# THE APP STORE MODEL FOR ENTERPRISE APPLICATION SOFTWARE

**Stefan Wenzel**

URN: urn:nbn:de:bvb:473-opus4-540277
DOI: https://doi.org/10.20378/irbo-54027

Diese Arbeit hat der Fakultät Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität als Dissertation vorgelegen.

Erstgutachter: Prof. Dr. Otto K. Ferstl
Zweitgutachter: Prof. Dr. Orestis Terzidis
Mitglied der Promotionskommission: Prof. Dr. Tim Weitzel

Tag der Disputation: 25.10.2018

# ACKNOWLEDGEMENTS

This dissertation was motivated and created in the context of practical challenges in the enterprise software industry. Over the years, many people - from academia, the business world, and my private environment - supported and contributed to bring this endeavor to a successful conclusion. I am profoundly grateful for this support and would like to give special thanks to the following persons.

First of all, I want to thank my doctoral supervisor, Prof. Dr. Otto K. Ferstl. He gave me the opportunity to pursue this dissertation as an external doctoral student allowing me to continue my professional career at SAP in parallel. He greatly helped me to maneuver this project through this setting with many challenges but also opportunities. His patient mentorship helped me to keep focus and persevere. Additionally, I would like to thank Prof. Dr. Orestis Terzidis for being part of my doctoral committee as second referee, for providing valuable feedback, and for being always approachable. My thanks go also to Prof. Dr. Tim Weitzel who was a member of my doctoral committee and gave me crucial guidance on research methodology and rigor.

Moreover, I would like to thank my co-authors for countless hours of discussion, late night conference calls, and last-minute activities to meet publication deadlines. Especially, I want to thank my co-author Dr. Francesco Novelli for the many great workshops and afternoons at SAP Research in Darmstadt. Likewise, I want to extend my thanks to my colleagues at SAP. For their professional inputs, expertise and advice. Special thanks go to Dr. Wolfgang Faisst who was my mentor from practice and who gave great inspirations to my research.

Very special thanks go to my parents, Renate and Kurt Wenzel. Without their support my entire career both academic and professional would not have been possible. Likewise, I would like to thank my brother, Dr. Andreas Wenzel. He has been an example to me and would always provide advice when needed.

Finally, my deepest respect and gratitude goes to my two wonderful kids, Philipp and Rosa, and my beloved wife, Carina. My kids deserve my thanks since they often had to forego my company for something, they did not really see any value in. Ultimately, my biggest thanks go to Carina, she has been paramount to success - not only, but also by always giving me the freedom to work on the thesis and by taking care of the family duties on her own – be it weekends, holidays, or late evenings.

Stefan Wenzel

# ZUSAMMENFASSUNG (GERMAN SUMMARY)

### Motivation und Problemstellung

App-Stores haben die Art und Weise wie Konsumentensoftware entwickelt, paketiert, vermarktet und konsumiert wird nachhaltig verändert. Auch im Segment für betriebswirtschaftliche Anwendungssoftware (kurz Unternehmenssoftware) arbeiten Unternehmen an einer Neukonfiguration der Software-Wertschöpfungskette: Etablierte Anbieter veröffentlichen Softwareplattformen zur Entwicklung komplementärer Anwendungen, ergänzt durch proprietäre elektronische Marktplätze, den sogenannten App-Stores, zur elektronischen Vermarktung und Distribution der Softwareprodukte.

Im engeren Sinne handelt es sich bei den App-Stores für Unternehmenssoftware um E-Commerce-Systeme zur Realisierung eines elektronischen Marktplatzes (Novelli & Wenzel 2013). Der betriebliche Einkauf von Unternehmenssoftware unterscheidet sich allerdings deutlich von dem im Konsumentenbereich: Lange Verkaufszyklen, komplexe Verhandlungen, sowie eine Vielzahl von Entscheidern und Beeinflussern charakterisieren diesen Prozess. Darüber hinaus ist die Unternehmenssoftware selbst meist deutlich komplexer. So existieren beispielsweise von einzelnen Produkten zahlreiche Varianten, Konfigurationen und Kombinationsmöglichkeiten, was den elektronischen Vertrieb vor zusätzliche Herausforderungen stellt (Wenzel et al. 2012).

Im weiteren Sinne betrachtet das App-Store-Modell nicht nur ein E-Commerce-System, sondern die gesamte Softwarewertschöpfungskette und infolgedessen auch die beteiligten Unternehmen, deren strategische Ausrichtung und Rolle am Markt, ihr Erlösmodell und ihre Art der Zusammenarbeit. Zudem begünstigt das App-Store-Modell spezielle Eigenschaften von Softwareprodukten: Zuvor monolithische Systeme werden nun in Form eines oder weniger Kernprodukte mit zusätzlichen komplementären Anwendungsmodulen dem Softwarekunden angeboten. Kern- und Komplementärprodukte basieren beide auf derselben technologischen Plattform. Das Kernprodukt als auch die technologische Plattform werden meist vom sogenannten Plattformanbieter, die komplementären Produkte von unabhängigen Softwareanbietern (ISV[1]) entwickelt und angeboten. Der Plattformanbieter kontrolliert zudem meist den App-Store, welcher als Katalysator für den zweiseitigen Markt fungiert und Softwarekunden und ISV zusammenbringt (Wenzel et al. 2012; Burkard et al. 2012).

Im App-Store münden somit nicht nur Softwarefunktionen im Sinne einer E-Commerce-Anwendung, sondern auch Regeln und Prozesse zur Entwicklung, zur Veröffentlichung, zur Vermarktung, zur Verteilung der Software, zum Preismodell und zu den zugehörigen

[1] Englisch: „Independent Software Vendor“.

finanziellen Transaktionen (Jansen & Bloemendal 2013). Folglich sollte das App-Store-Modell zur ganzheitlichen Erfassung als Geschäftsmodell verstanden werden.

Im Markt für Unternehmenssoftware unterscheiden sich die Strukturen, Anforderungen und technologischen Rahmenbedingung erheblich von denen im Konsumentensegment. Das einfache Replizieren und Übertragen der Konzepte aus diesem in den Unternehmenssoftwarekontext kann deshalb nicht die bevorzugte Lösung sein. Vielmehr sollten die spezifischen Anforderungen der Marktteilnehmer im Unternehmenssoftwaresegment analysiert werden, um daraufhin Lösungsvorschläge zu entwickeln wie die Softwarewertschöpfungskette und die nötigen Informationssysteme zu gestalten sind, um ein App-Store-Modell nachhaltig am Markt zu etablieren.

Die übergeordnete Forschungsfrage der vorliegenden Arbeit ist folglich: Wie ist das App-Store-Modell für Unternehmenssoftware zu gestalten? Zur Beantwortung der Forschungsfrage wird das Geschäftsmodellkonstrukt herangezogen. Das Geschäftsmodell im Verständnis dieser Arbeit besteht aus dem Werteversprechen für die betrachteten Akteure, der Wertearchitektur und dem Erlösmodell (Stähler 2002, pp.41–42). Die Analyse des App-Store-Modells legt den Schwerpunkt auf die Wertearchitektur. Diese wird zur Analyse in vier Teile zerlegt: die angebotsseitige Wertschöpfungskette (a), die nachfrageseitige Wertschöpfungskette (b), den App-Store für Unternehmenssoftware als Anwendungssystem (c) und Produkteigenschaften von Unternehmenssoftware für App-Stores (d).

**Methodik**

Die Forschungsfrage wird in zwei aufeinanderfolgenden Schritten mit jeweils unterschiedlicher aber komplementärer Methodik beantwortet. Die Wertearchitektur wird zunächst in den genannten vier Teilbereichen mit Hilfe von sieben international veröffentlichter Artikeln untersucht[2]. Hierzu werden empirisch-qualitative Methoden mit explorativer Zielsetzung angewendet. Die empirischen Untersuchungen fokussieren sich vornehmlich auf die Erfassung existierender Strukturen im Untersuchungsbereich, den mit dem App-Store-Modell verbundenen Zielsetzungen und Herausforderungen sowie der Erfassung und Analyse existierender Praxisansätze zur Lösung einzelner Problemstellungen.

Im zweiten Schritt führt die Einführungsschrift die erarbeiteten empirischen Ergebnisse mit Hilfe des Geschäftsmodellkonstrukts zu einer integrierten Sicht zusammen. Diese modellbasierte Integration geht über eine bloße Zusammenfassung der Einzelteile hinaus und ist ein eigenständiges Forschungsinkrement. Es folgt der gestaltungs- bzw. konstruktionsorientierten Methodik und hat zum Ergebnis ein Modellartefakt, welches auf

[2] Einer der sieben Artikel ist eine umfassende Erweiterung eines veröffentlichten Artikels. Dieser wird somit in der erweiterten Form zusammen mit der Einführungsschrift der Dissertation das erste Mal publiziert.

Basis der empirischen Erkenntnisse entwickelt wurde: das „App-Store-Modell für Unternehmenssoftware". Der gestaltungsorientierte Teil orientiert sich insbesondere am Verständnis von Frank und Becker (Frank 2009; Becker 2010).

**Aufbau und Ergebnisse**

Die Arbeit gliedert sich in fünf Hauptteile. Der erste Teil (*Part I*) beinhaltet die Einführungsschrift, die anderen vier Teile der Arbeit (*Part II-V*) gruppieren die wissenschaftlichen Einzelbeiträge im Sinne der kumulativen Dissertation.

Die Einführungsschrift (*Part I*) selbst erfüllt drei Kernfunktionen: Erstens bietet sie einen gesamthaften Überblick zum Thema, in dem es dieses grundsätzlich motiviert und herleitet. Relevante Begriffe werden definiert und abgegrenzt sowie verwandte Forschungsbereiche ausführlich dargestellt. Zudem wird ein gesamthafter Überblick zur Methodik und zu den wichtigsten Forschungsergebnissen der Einzelbeiträge gegeben. Zweitens hat die Einführungsschrift eine Komplementärfunktion. Sie ergänzt Einzelaspekte oder fügt neue hinzu, welche aufgrund von formalen Seitenlimitationen nicht in die Einzelbeiträge aufgenommen werden konnten. Die dritte und wichtigste ist eine Synthesis- und Integrationsfunktion. Ein eigenständiges Kapitel zur integrierten Sicht auf das App-Store-Modell bearbeitet die übergeordnete Forschungsfrage: Wie ist das App-Store-Modell für Unternehmenssoftware zu gestalten? Hierzu wird mit Hilfe des Geschäftsmodellrahmens ein Modellartefakt erstellt. Dabei werden die Erkenntnisse aus den anderen Kapiteln und den Einzelbeiträgen zusammengeführt, ergänzt und im Sinne der Modellkonstruktion interpretiert und zu einem eigenständigen Modellartefakt erweitert. So werden zunächst die möglichen Werteversprechen für Softwarekunden, ISV und Plattformanbieter entlang des Geschäftsmodellrahmens dargestellt. Im nächsten Schritt erfolgt die Entwicklung der Wertearchitektur. Hierbei werden sowohl die Geschäftsprozesse der Aufgabenebene modelliert, als auch die Aufgabenträgerebene in Form von Anforderungs- und Funktionskatalogen beschrieben. Zuletzt wird das Geschäftsmodellartefakt durch die Beschreibung des Erlösmodells vervollständigt.

Der zweite Teil der Arbeit (*Part II*) betrachtet die angebotsseitige Wertschöpfungskette. Hierbei werden zunächst die Strukturen von plattformbasierten Softwareökosystemen untersucht und die Kernelemente des Plattform-Service herausgearbeitet. Im nächsten Schritt werden die Rollen des ISV, des Plattformanbieters und des Softwarekunden abgegrenzt und definiert. Das Angebot des Plattformanbieters gegenüber den ISVs wird im Kontext von Platform-as-a-Service (PaaS) und dessen typischen Leistungsbestandteilen beschrieben. Darauf aufbauend werden die Zielsetzungen und der Wertschöpfungsprozess der ISVs im Softwareökosystem als auch die Methoden des Plattformanbieters zur Unterstützung der ISVs detailliert analysiert.

Die Ergebnisse zeigen, dass ISVs insbesondere vier Ziele verfolgen, wenn sie mit Plattformanbietern zusammenarbeiten: (1.) verbesserter Kundenzugang, (2.) bessere Befriedigung der Kundennachfrage, (3.) Integration der eigenen Lösung mit Kernlösungen der Plattformanbieter und (4.) Erweiterung der eigenen Geschäftstätigkeit. Zur Erreichung der Ziele sollte der Plattformanbieter neben softwaretechnischen Ressourcen (z.B. Entwicklungsumgebung, Infrastrukturdienste) insbesondere Wert legen auf eine gesamthafte Unterstützung und strukturierte Anleitung der ISVs entlang des für sie relevanten Teils der Softwarewertschöpfungskette (inklusive der Vermarktungs- und Vertriebsphasen).

Teil drei der Arbeit (*Part III*) betrachtet die nachfrageseitige Wertschöpfungskette und nimmt den Softwarekunden und den zu diesem gerichteten Teil des Plattformanbieters in den Fokus. Die bearbeiteten Kernbereiche umfassen die Analyse des Onlineeinkaufsprozesses des Softwarekunden für Unternehmenssoftware und die Untersuchung der Treiber und Barrieren zur Adoption des Onlineeinkaufskanals für Unternehmenssoftwarekäufe. Dabei werden mögliche Maßnahmen, sowohl informationstechnische als auch organisatorische, zur Überwindung der identifizierten Barrieren untersucht. Die Ergebnisse zeigen, dass drei Gruppen von Einflussfaktoren die Eignung des Onlinekanals zum Softwarekauf beeinflussen: (software-) lösungsbezogene Faktoren, transaktionsbasierte Faktoren und softwarekundenbezogene Faktoren.

Eine herausragende Stellung nehmen die lösungsbezogenen Faktoren ein, da diese nicht nur direkten Einfluss auf die Adoption des Onlinekanals haben, sondern auch die weiteren Determinanten beeinflussen. Es werden insgesamt acht lösungsbezogene Kriterien differenziert: Kritikalität, Liefermodell (Cloud/On-Premise), Evaluierbarkeit, Implementierungs- und Integrationsaufwand, Preisniveau, Lösungsumfang, Spezifität/Individualisierung und Nutzeranzahl. Als Konsequenz kann zusammengefasst werden, dass diese Aspekte der Softwarelösung maßgeblichen Einfluss auf die Eignung für den Onlineeinkauf haben und der angestrebte Vertriebskanal demnach bei der Gestaltung des Softwareproduktes berücksichtigt werden sollte. Darüber hinaus konnte gezeigt werden, dass sich die Eignung des Onlinekanals im Verlauf des Einkaufsprozesses verändert. Je nach Softwarekategorie ist es demnach sinnvoll den Onlinevertriebskanal in ein Multi-Kanal-System mit traditionellen Vertriebskanälen (z.B. personellem Direktvertrieb) zu integrieren.

Aufbauend auf den vorangegangenen Analysen zur angebotsseitigen und nachfrageseitigen Wertschöpfungskette, wird als nächstes der App-Store für Unternehmenssoftware als Anwendungssystem in Teil vier der Arbeit (*Part IV*) untersucht. Dieser Teil fokussiert insbesondere auf die Fragestellung nach den wichtigsten Nutzungsszenarien für App-Stores für Unternehmenssoftware und deren Anforderungen und Funktionen in Bezug auf die organisationalen Softwarebeschaffungsprozesse.

Es können grundsätzlich zwei unterschiedliche Arten von App-Stores für Unternehmenssoftware identifiziert werden: der öffentliche App-Store und der interne App-Store. Während der öffentliche App-Store zur betrieblichen Beschaffung von Unternehmenssoftware den gesamten Einkaufsprozess unterstützt, fokussiert sich der interne App-Store auf die Verteilung der Software an den Nutzer und das Management der Softwareanwendungen während des gesamten Nutzungszeitraums. Beide App-Store-Typen benötigen spezifische Funktionalitäten zur Unterstützung der betrieblichen Prozesse, welche sich deutlich von App-Stores für Konsumenten unterscheiden (z.B. Unterstützung eines komplexen Rechte- und Rollenmodells, betriebliche Rabatte, Angebotsanfrageprozesse). Aus Sicht der IT-Governance ist zudem eine Integration des öffentlichen und internen App-Stores zu empfehlen, um somit den Softwareeinkaufsprozess mit dem Softwareverteilungsprozess effizient und durchgängig zu verbinden.

Der letzte Teil der Arbeit, Teil fünf (*Part V*), untersucht detailliert die Eigenschaften von Unternehmenssoftware und ihre Eignung für das App-Store-Modell, insbesondere für den dazugehörigen Onlinevermarktungs- und -vertriebsprozess. Hierzu werden als Basis die im Kapitel zur nachfrageseitigen Wertschöpfungskette identifizierten lösungsbezogenen Determinanten zur Adoption des Onlineeinkaufskanals herangezogen. Darauf aufbauend wird die Frage evaluiert, welche Eigenschaften von Unternehmenssoftware sich am besten für das App-Store-Modell eignen und mit welchen Maßnahmen diese Eigenschaften bei Unternehmenssoftwareprodukten erreicht werden können. Es werden hierzu sowohl kommerzielle, funktionale als auch technische Aspekte von Softwareprodukten untersucht und mit Hilfe lösungsbezogener Determinanten bewertet. Sieben sogenannte App'ifizierungskriterien konnten identifiziert werden, welche die Eignung von Softwareprodukten für das App-Store-Modell verbessern und zum Teil erst ermöglichen, wie beispielsweise ein aufgabenorientierter Lösungsumfang sowie die Erprobbarkeit oder Verfügbarkeit eines Starterpakets. Es werden zahlreiche Maßnahmen vorgestellt, von denen einige auf existierende Softwareprodukte anwendbar sind, während andere bereits frühzeitig bei Planung und Gestaltung neuer Softwareprodukte berücksichtigt werden müssen.

**Beitrag zu Wissenschaft und Praxis**

Die Erfassung, Gestaltung und Analyse von technologiegetriebenen Geschäftsmodellen wird von Hess als besonders wichtige Aufgabe der Wirtschaftsinformatik hervorgehoben (Hess 2012, p.3). Die Dissertation folgt dieser Empfehlung indem sie das App-Store-Modell für Unternehmenssoftware erstmals vollständig erfasst und einen umfangreichen und detaillierten Lösungsraum modelliert und beschreibt.

Initiative und Inhalt der vorliegenden Dissertation sind stark aus der Praxis und den aktuellen Problemstellungen, Herausforderungen und Trends der Unternehmenssoftwarebranche motiviert. Um den komplexen sozio-technologischen Kontext von App-Stores für Unternehmenssoftware zu erfassen, mussten unterschiedlichste Forschungsfelder zusammengeführt werden. Deshalb ist, neben den Erkenntnissen in den Einzelbereichen, die integrative Natur der Dissertation selbst als wissenschaftlicher Beitrag zu verstehen. Die zahlreich hergestellten Verknüpfungen zwischen den sonst oft isoliert betrachteten Forschungsfeldern, wie beispielsweise zwischen Industriegütermarketing, Forschung zu Softwarearchitekturen und Softwareökosystemen, oder der Verbindung von Arbeiten zu Cloud-Computing und elektronischen Marktplätzen, ermöglichten völlig neue Erkenntnisse. Ein Beispiel hierfür ist die Erkenntnis, dass der präferierte Vertriebskanal eines Softwareproduktes schon sehr früh bei Entscheidungen im Software-Design- und Softwarearchitekturbereich berücksichtigt werden sollte. Aufbauend darauf konnten konkrete softwaretechnische Maßnahmen und Empfehlungen abgeleitet werden.

Darüber hinaus gibt die Arbeit detaillierte Einblicke in die tägliche Praxis und die Herausforderungen der Unternehmenssoftwarebranche, wie sie der akademischen Welt oftmals schwer zugänglich ist. Dies gilt gleichermaßen für die in der Arbeit einzeln behandelten Forschungsbereiche wie Softwareeinkauf, IT-Governance oder Softwareökosysteme (insbesondere Partnermanagement), als auch im Kontext der umfassenderen Geschäftsmodellforschung. Somit stellt die vorliegende Dissertation mit ihrem integrativen Charakter auch einen möglichen eigenständigen Forschungsgegenstand für weitere und aufbauende Arbeiten dar.

Die enge Zusammenarbeit mit Industriepartnern war essentiell für die Arbeit: Einerseits ermöglichten sie den Zugang zu Kernressourcen wie Experten, Systeme, und Dokumentations- und Informationsquellen zur Erhebung qualitativ empirischer Daten. Andererseits wurden die Forschungsergebnisse in Form von Expertenreviews und Praxisworkshops kontinuierlich zurückgespielt. Dabei konnten die Ergebnisse validiert und Feedback eingeholt werden. Dieser kontinuierliche Austausch mit der Praxis stellte somit auch die Relevanz der Arbeit sicher. Die kumulative Struktur der Thesis hat bei diesem Vorgehen sehr geholfen.

Die zahlreichen aus den Forschungsaktivitäten abgeleiteten Erkenntnisse und Empfehlungen können Praktikern bei vielen Problemstellungen Unterstützung bieten: Das umfassende Geschäftsmodellartefakt und die detaillierte Analyse der Aufgaben- und Aufgabenträgerebene leisten wertvolle Hilfestellung sowohl bei der Planung und Umsetzung eines App-Store-Modells, als auch bei der Analyse und Verbesserung bereits existierender Prozesse und Implementierungen.

Abschließend bleibt festzustellen, dass – trotz der Allgegenwärtigkeit von App-Stores im Konsumentenumfeld – das App-Store-Modell im Unternehmenskontext noch nicht vollständig umgesetzt ist. So wird die gesamte Softwarewertschöpfungskette des plattformbasierten Softwareökosystems trotz deutlicher Fortschritte in einzelnen Bereichen noch nicht durchgängig von integrierten Informationssystemen unterstützt. Die Arbeit wird so auch zukünftig dazu beitragen das App-Store-Modell konsequent für Unternehmenssoftware am Markt zu etablieren.

**Literaturverweise**

Becker, J., 2010. Prozess der gestaltungsorientierten Wirtschaftsinformatik. In H. Österle, R. Winter, & W. Brenner, eds. *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. pp. 13–17.

Burkard, C., Widjaja, T. & Buxmann, P., 2012. Software Ecosystems. *Business & Information Systems Engineering*, 4(1), pp.41–44.

Frank, U., 2009. Die Konstruktion möglicher Welten als Chance und Herausforderung der Wirtschaftsinformatik. In J. Becker, H. Krcmar, & B. Niehaves, eds. *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik.* Heidelberg: Physica-Verlag, pp. 167–180.

Giessmann, A., 2015. *Designing business models for cloud platforms*, Lausanne: Université de Lausanne.

Hess, T., 2012. *Geschäftsmodelle als Thema der Wirtschaftsinformatik - Arbeitsbericht 1/2012* T. Hess, ed., München: Ludwig-Maximilians-Universität München, Fakultät für Betriebswirtschaft, Institut für Wirtschaftsinformatik und Neue Medien.

Jansen, S. & Bloemendal, E., 2013. Defining App Stores: The Role of Curated Marketplaces in Software Ecosystems. In *Lecture Notes in Business Information Processing Volume 150*. pp. 195–206.

Novelli, F. & Wenzel, S., 2013. Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity? In *ICIS 2013 Proceedings*.

Stähler, P., 2002. *Geschäftsmodelle in der digitalen Ökonomie. Merkmale Strategien und Auswirkungen* 2nd ed., Josef Eul Verlag.

Wenzel, S. et al., 2012. New Sales and Buying Models in the Internet: App Store Model for Enterprise Application Software. In *MKWI 2012*. pp. 639–651.

# TABLE OF CONTENTS: THESIS OVERVIEW

# PART I: INTRODUCTORY PAPER

## Index of Abbreviations

| | |
|---|---|
| A2A | Administration-to-administration |
| A2B | Administration-to-business |
| A2C | Administration-to-consumer |
| aPaaS | Application based PaaS |
| ARE | Application runtime environment |
| ASP | Application Service Providing |
| B2A | Business-to-administration |
| B2B | Business-to-business |
| B2C | Business-to-consumer |
| B2E | Business-to-employee |
| BI | Business intelligence |
| BYOD | Bring-your-own-device |
| C2C | Consumer-to-consumer |
| CAQDAS | Computer aided qualitative data analysis software |
| CIM | Computer integrated manufacturing |
| CIO | Chief information officer |
| CRM | Customer relationship management |
| CxO | Chief-level executive |
| EAS | Enterprise application software |
| E-business | Electronic Business |
| E-commerce | Electronic Commerce |
| ERP | Enterprise resource planning |
| GT | Grounded theory |
| IaaS | Infrastructure-as-a-Service |
| IAS | Interaction schema |
| IDE | Integrated development environment |
| IDT | Innovation diffusion theory |

| | |
|---|---|
| IS | Information system |
| ISV | Independent software vendor |
| MAM | Mobile application management |
| MDM | Mobile device management |
| PaaS | Platform-as-a-Service |
| PLM | Product lifecycle management |
| RQ | Research question |
| SaaS | Software as a service |
| SCM | Supply chain management |
| SECO | Software ecosystem |
| SOA | Service-oriented architecture |
| SOM | Semantic object model |
| SRM | Supplier relationship management |
| TAM | Technology Acceptance Model |
| TBA | Theory of Planned Behavior |
| TCO | Total cost of ownership |
| TOE | Technology-Organization-Environment |
| TRA | Theory of Reasoned Action |
| UI | User interface |
| UTAUT | Unified Theory of Acceptance and Use of Technology |

# Index of Figures

# Index of Tables

# Table of Contents: Introductory Paper

# 1 Introduction

## 1.1 Motivation and Problem Description

App stores have gained a wide popularity in the segment of mobile consumer software and have ultimately changed the way consumer software is built, packaged, sold and delivered, as well as how they are bought and consumed. The huge amount of 100 billion apps downloaded from mobile, consumer-oriented app stores had already been reached by the end of 2013, with the Apple App Store and Google Play Store, the two most prominent sources, together accounting for approximately 90% of all downloads (Gartner 2013).

The growing diffusion of consumer app stores has also driven large providers of enterprise application software (EAS) to establish their own software platforms, offering complementary applications and services through their own proprietary app stores, such as the Salesforce.com AppExchange (Salesforce.com 2014) or the SAP Store (SAP, 2014; see also Appendix A). Technology analysts also believe that app stores have the potential to significantly improve the value chain for EAS and nominated the topic as a major IT trend in 2012 and 2013 (Pettey 2011; Pettey 2012; Finley 2012).

### 1.1.1 App Store as an Online Channel

An app store in the narrow sense is an online sales and distribution channel (or short online channel), as a set of organizational and technological means. It constitutes a centralized electronic commerce (e-commerce) infrastructure, i.e., an electronic marketplace, enabling software providers to market software applications online to software customers (i.e., an individual person or organizational entity). An app store serves the software customer throughout the buying process by providing information search, evaluation to purchase and software delivery, with minimum and possibly virtual and asynchronous human interaction (Novelli & Wenzel 2013).

Traditional, monolithic, and feature-rich EAS, such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM), are classified as investment goods and marketed to the central IT organization via a long-lasting, highly consultative, and personnel-intensive process (Wenzel et al. 2012). The entire buying process from information gathering, evaluating alternative products, negotiating on discounts and individual terms, to the actual purchase of an application can last from several months to years (Liao et al. 2007; Halingten & Verville 2002). This heavyweight buying process often does not, or only insufficiently, involve business stakeholders and users, and single requirements are dropped since they do not justify a business case (Wenzel et al. 2012). As a consequence, IT departments are perceived as inhibitors and an innovation bottleneck. Business departments and users have reacted by starting to help themselves and by subscribing to external offerings without permission by IT, or establishing workarounds based on spreadsheets. This phenomenon is also referred to as "shadow IT" (Jones et al. 2004; Berbner & Bechtold 2010, p.261).

As a result business departments ask for more and more influence in IT purchasing decisions, and the role of the chief information officer (CIO) and the IT department are questioned (Carr 2003; Rettig 2007; Vizard 2012). The rise of cloud computing has only intensified this development as it puts business stakeholders even more in the driver seat. In particular, Software-as-a-Service (SaaS) eliminates the need for local hardware, IT operations, and lengthy implementation projects, and therefore allows business users to directly interact with external providers (Armbrust et al. 2009; Jennings 2008). Business users and decision makers are ultimately also consumers and demand a similar convenience, user experience, and flexibility when searching, buying, and using software applications in their business environment as they do in their private context (Wenzel et al. 2012; Price et al. 2012).

App stores for EAS try to explicitly address these decentralized buying potentials and the demand for a simple buying experience with their high transparency and minimal entry barriers, and by replacing the long-lasting, personnel-intensive, and consultative buying process of EAS with an efficient, flexible, and in short, rather transactional one (Rackham & DeVincentis 1999; Wenzel et al. 2012).

However, it is disputable whether online software purchases are as compelling for organizations as they are for individuals. In recent years, adoption rates and the maturity of app stores for EAS remain at a low level (Böckle 2013): e.g., many applications cannot be bought and activated online, so users have to request a quote by filling in web forms before sales professionals then follow up offline. While in consumer software markets traditional channels have been largely replaced by online channels, offline channels based on intermediaries and sales professionals represent the dominant go-to-market model for EAS. Therefore, it is of interest to know what the drivers for and barriers against adopting an online channel for EAS acquisitions are.

Aside from lacking e-commerce capabilities and organizational issues, another reason might be that EAS itself is not ready to be sold online – applications are still too complex and have a monolithic structure – and these systems often have a history of 10-20 years and a go-to-market model which was tailored to fit a consultative, offline sales channel. Consequently, an investigation is needed to determine which product characteristics EAS needs to fulfil in order to suit online channels, i.e., app stores.

### 1.1.2 App Store as a Business Model

However, app stores for EAS not only digitize and automate the sales and distribution process but affect the entire software value chain, including the companies involved, their strategic focus, their collaboration, and their revenue model. While ultimately app stores also favor a different "design" of software applications (Wenzel et al. 2012; Burkard et al. 2012). Previous individual products can now be offered as systems with one core product and multiple complementary products. The provider of the core product together with Independent Software Vendors (ISV), who offer the complementary products, form a so-called Software Ecosystem (SECO) (Burkard et al. 2012), while core and complementary products are mostly based on a common technological platform. The app store is typically owned by the provider of the core product and the technological software platform and is referred to as a platform provider. This acts as a catalyst and enabler for the two-sided platform business by implementing and defining functions, rules and policies for information exchange, software development, product publication, distribution, and the related financial transactions (Jansen & Bloemendal 2013).

Holistically, the App Store Model for EAS should therefore be understood as a business model in the software industry. It is based on a two-sided platform business with ISVs on the one side, and software customers on the other. Furthermore, it includes a software platform and an app store owned by the platform provider. Therefore, the app store model describes a business model for the aforementioned SECO including the entire value network of platform provider and independent software vendors (cf. Figure 1).

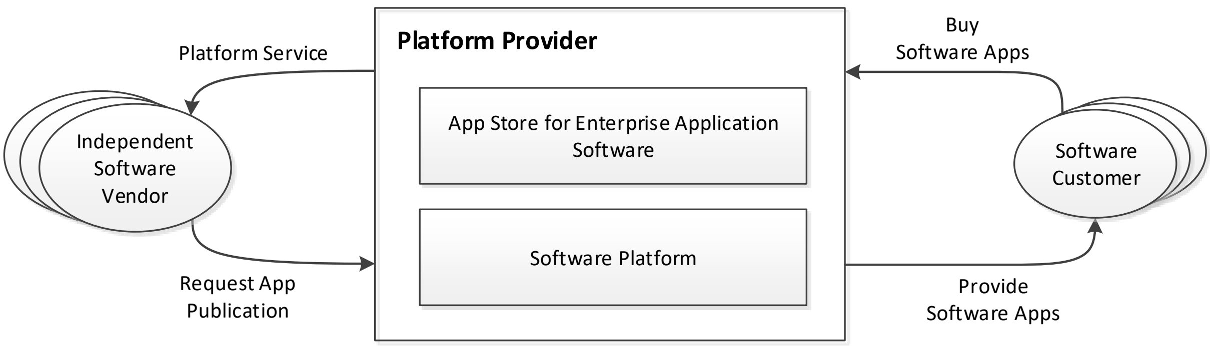


**Figure 1.** Simplified representation of the business system[1] of the app store model.

According to Stähler (Stähler 2002, pp.41–42; Stähler 2010) a business model describes the value proposition of all actors, the value architecture and the revenue model.

It is therefore of high practical but also academic interest to know how the app store model as business model is defined and especially the value creation and value chain configuration (value architecture), since this demonstrates the interrelation of digitization, technology and business concepts (Veit et al. 2014).

[1] The figure presented here uses the Semantic Object Model (SOM), i.e., the interaction schema (Ferstl & Sinz 1997). Transactions labelled with "E:" refer to enforcing transactions.

As described in the opening section the App Store Model for EAS as a business model describes a complex socio-technological scenario. Business models in general are a topic of strong interest for Information Systems (IS)[1] research, business administration and economic research, and for practitioners in the respective domain of the business model (Hess 2012, p.2 ff.). Before I set out the individual research objectives and research questions, I want to offer further arguments to support the relevance of the topic to the aforementioned scholars and especially IS research.

The topic at hand, can therefore be understood as interdisciplinary, applied research, which is not only typical for IS research, but also highly necessary in complex settings to fully comprehend the socio-technological context, as Banker and Kauffman conclude in their 50 year review of IS research (Banker & Kauffman 2004).

In recent years business models have been acknowledged as relevant topic for IS (Hess 2012, p.3; Osterwalder & Pigneur 2005; Burkhart et al. 2011). In their IS research agenda on business models, Veit et al. define four major fields of research (Veit et al. 2014):

1) business model as a fundamental concept and unit of analysis
2) IT support for developing and managing business models
3) business models in IT industries
4) digital business models.

The first topic, business model fundamentals, mainly study the definition of the concept and the related theory. The second one studies the process of business model creation and its lifecycle, whereas the third and fourth topics deal with the description, analysis, and design of business models. Among business models in the IT industry to which the app store model can be assigned, Veit et al. argue that those in the software industry are of special interest for IS research due to the unique characteristics of software products (e.g., low marginal cost) and the software market (e.g., high internationalization and strong network effects), but also since EAS itself is an IS topic. Furthermore software business models require a deep understanding of the market (Veit et al. 2014), and consequently an application-oriented research endeavor, such as the analysis of the App Store Model for EAS demands an interaction with the software industry, i.e., EAS companies and relevant business experts.

Business administration and economic scholars are typically interested in the subjects of strategic management, entrepreneurship and in business models (Hess 2012, p.6). Strategic management is of interest since business models can be viewed as an intermediary between

[1] While referring to IS research, I synonymously refer to both the international orientation of IS research and the orientation of its German sister discipline „Wirtschaftsinformatik“ (sometimes translated as Business Information Systems Engineering (Veit et al. 2014), but mostly as IS).

strategy and business processes (Al-Debei et al. 2008). While strategy focuses on the position of a company in the market and towards competition, business models draw attention to the value creation in the company and their value network (Veit et al. 2014). Entrepreneurship is of interest since business models can be used as a blueprint to plan future business, and as an analysis tool for entrepreneurial activity (George & Bock 2011).

However, in the dedicated context of the app store model it is mainly scholars of marketing management, (i.e., “app store as an online channel”), industrial marketing (i.e., organizational buying behavior), and IT adoption (i.e., adoption of an online channel, innovativeness of the software value chain, diffusion of IT) who study the related subjects.

At the time of writing and to the best of my knowledge, the App Store Model for EAS has hardly been investigated beyond the papers in this work. There are multiple related research subjects contributing and intersecting with this work, such as SECO, software platform research, or works on organizational buying behavior (cf. Chapter 2. for a comprehensive overview on related works). However, none of the works examine app stores in the domain of EAS and their impact on the EAS value chain comprehensively or in detail.

## 1.2 Research Objectives

The research objective of this thesis is to comprehensively capture and describe the App Store Model for EAS as a business model, in order to analyze, design, or improve app-store-based systems. The central research question of this thesis is therefore a design-oriented question: How to design the app store model for EAS? To answer this overarching question, the complexities of the problem have been broken down into multiple empiric questions analyzing the major objectives, existing structures, and challenges of the researched domain. Before I elaborate in more detail about the research objectives and questions, I prepone the definition and use of the business model concept to make the arguments more transparent and comprehensible.

A business model, like any model, is an abstraction (Al-Debei et al. 2008). The level of abstraction varies with the objectives of the business model. It can be an abstraction of one concrete existing business activity, of a future business, or it can group multiple business activities. Moreover, it can be used for planning a business or analyzing it (Stähler 2002, p.41 ff.). For this work the business model acts as a framework, which helps to derive and integrate the individual research questions and results, and to help better comprehend the interrelation between value creation and the disruptive nature of new technologies. Moreover, the framework also helps bridge the boundaries of the different academic disciplines and integrates aspects of IS and business or economic research. Finally, it provides a framework which offers practitioners easy access to the results of this research.

The following components of a business model are distinguished in this work (Stähler 2002, p.41 ff. Stähler 2010):

- the value proposition of the individual actors
- the value architecture
- the revenue model.

The value architecture will be at the core of this thesis (for a detailed elaboration of the importance of the value architecture, see section 2.1 on business models). Using the introduced business system illustration from Figure 1 we can further distinguish four sub-components within the value architecture of the app store model. The four sub-components as highlighted in Figure 2 represent four sub-research clusters which help to derive more concrete research questions and support the integration of the individual research results at the end of the thesis. In the following, I will present detailed research questions assigned to each sub-cluster.

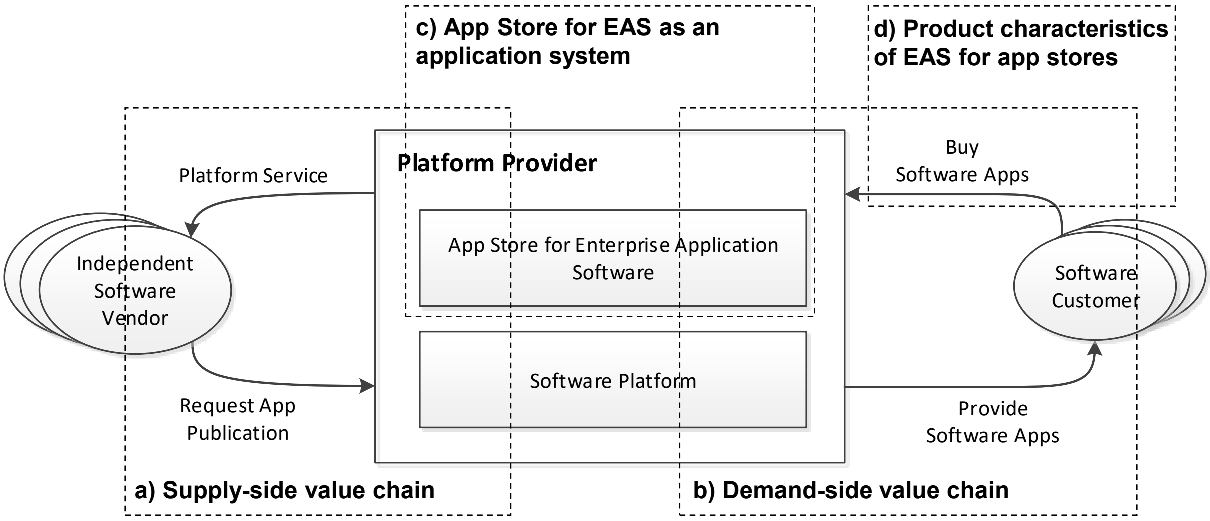


**Figure 2:** Research clusters delimited using the value architecture / business system

*a) Supply-side value chain*

The value chain or value creation process will be investigated in two parts. The first, the supply-side value chain, examines the business processes (incl. process phases, service transactions, supporting IT systems and tools, and organizational aspects) between ISVs and the platform provider, by addressing the following research questions (RQ):

- Why do ISVs cooperate with platform providers? (RQ_a.1)
- What are the key elements of the platform service? (RQ_a.2)
- How do ISVs define their value creation process within the SECO? (RQ_a.3)

- What are the platform provider's key enablers to support ISVs in reaching their goals? (RQ_a.4)

*b) Demand-side value chain*

The demand-side value chain evaluates the buying process of EAS via an online channel, i.e., the app store. Consequently, the business processes between software customers and the platform provider will be illuminated (incl. transactions, supporting IT systems, and organizational aspects), using the following research questions:

- What is the online buying process of an EAS customer? (RQ_b.1)
- What are the drivers and barriers of adopting an online channel for EAS acquisitions? (RQ_b.2)
- Which technical and organizational measures can be applied to overcome the adoption barriers of the online channel? (RQ_b.3)
- How can online sales channels for EAS be introduced and co-exist with existing, traditional "offline" sales channels? (RQ_b.4)

*c) App Store for EAS as an application system*

This cluster focuses on the app store as an application system, i.e., capabilities, features, and usage scenarios.

- What are the major usage scenarios, features, and capabilities of app stores for EAS recognizing the unique requirements of companies? (RQ_c.1)
- What are the different types of app stores for EAS and how can they be best used in the corporate context? (RQ_c.2)

*d) Product characteristics of EAS for app stores*

The last research cluster focuses on the suitability of EAS products for the app store model.

- Which characteristics of EAS products are best for the app store model? (RQ_d.1)
- Which measures help to improve the suitability for the app store model? (RQ_d.2)

Although this work aims to provide an integrated and holistic perspective on the app store model there are some aspects which are explicitly excluded from the investigation. First, the areas of application of EAS and its individual features and functions are not examined comprehensively. Functional domains, application scope, and design of software, do indeed play a role in the conducted case studies and are referenced in the frameworks developed, but these are mostly used to exemplify arguments. Furthermore, software architectures of EAS, such as Service Oriented Architectures (SOA), component-based architectures, software extensibility concepts, are not part of this investigation. I will however reference selected frameworks in the related works section. The main focus when investigating the EAS itself will be on product characteristics, which are characteristics relevant to the buying and adoption process of the software. Last, mainly economic examinations such as two-sided markets, network effects, and platform market structures are also seen as adjacent fields of research and are referenced in Chapter 2, but are not focal points of this work.

The design and requirements of a software platform as presented in Figure 2 will only be examined as part of the analysis on the supply-side value chain. As with software architectures on EAS, the design, the requirements, and the technical details of software platforms require dedicated studies to guarantee the topic can be captured adequately.

## 1.3 Structure of the Thesis

The thesis is structured in five parts. Figure 3 illustrates the structure using the previously introduced business system of the app store model.

**Part I: Introductory Paper**

The Introductory paper has a threefold function. First, it provides a holistic overview of the topic in that it fundamentally motivates the research from an academic and practical point of view, introduces basic definitions, essential theories and the major related works representing the state-of-the-art. Moreover, it describes all the methodologies applied, and highlights the results of the individual papers. Last, an overview of limitations, contributions, and future research will be provided.

Second, the introductory paper has a complementary function, in that it takes the freedom to elaborate dedicated aspects at greater length (e.g., methodology or theoretical foundation) or even to add aspects which had fallen victim to page limitations and the rigid focus of conference or journal papers.

Third and last, the introductory paper has an integration and synthesis function. The dedicated Chapter 5 “An Integrated Perspective on the App Store Model for Enterprise Application Software” will use the introduced business model framework and the structure as highlighted in Figure 2 to consistently present, combine, and integrate the individual results and show their connection across delimitated subjects. Chapter 5 represents its own dedicated research increment and moreover should be understood as a ‘model artefact’ according the notion of design-oriented IS research (for a comprehensive explanation of how to interpret this model artefact, refer to Chapter 5.1). Over years of research in a novel, fast changing, and complex socio-technological context such as this one, it is inevitable that terminologies evolve or are replaced, and new technical developments appear on the market which need to be recognized and captured. The integration chapter therefore also tries to harmonize the diverging aspects of the individual papers.

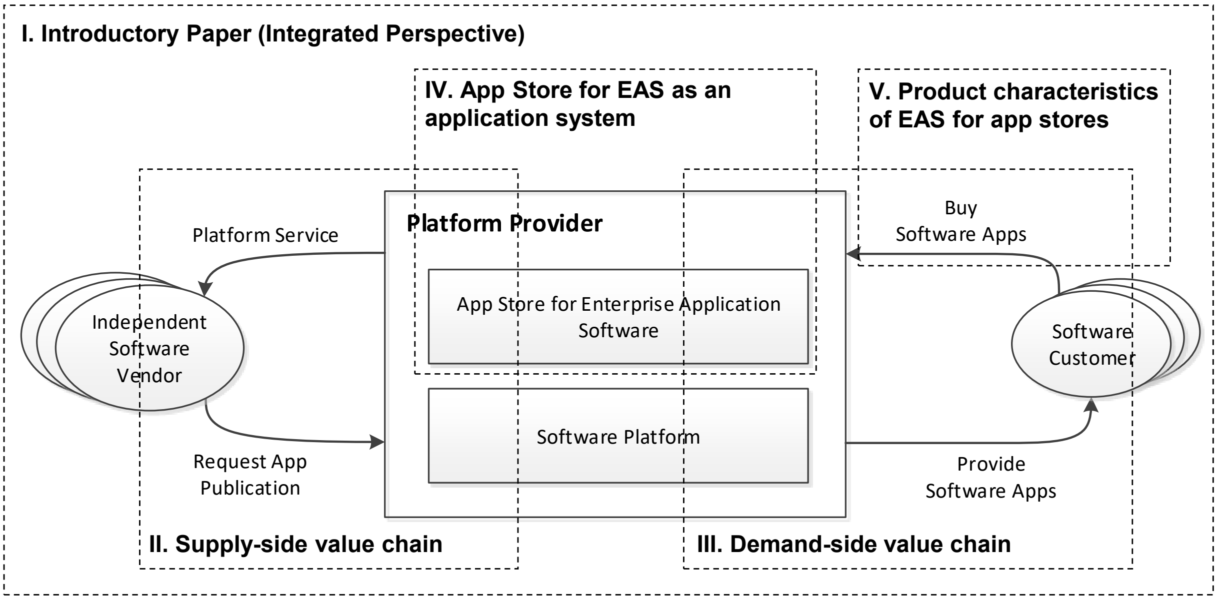


**Figure 3:** Structure of the thesis

Part II-V consist of the individual research papers.

## Part II: Supply-Side Value Chain

**Paper 1:** Daniel Beimborn, Thomas Miletzki, Stefan Wenzel: Platform-as-a-Service (PaaS). In: Business & Information Systems Engineering (2011:6), 381-384.

**Paper 2:** Theresa Rickmann, Stefan Wenzel, Kai Fischbach: Software Ecosystem Orchestration: The Perspective of Complementors. In: Proceedings of the Twentieth Americas Conference on Information Systems (AMCIS 2014), Savannah.

## Part III: Demand-Side Value Chain

**Paper 3:** Stefan Wenzel, Francesco Novelli, Christoph Burkard: Evaluating the App-Store Model for Enterprise Application Software and Related Services. Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI2013), Leipzig 2013.

**Paper 4:** Francesco Novelli, Stefan Wenzel: Adoption of an Online Sales Channel and “Appification” in the Enterprise Application Software Market: A Qualitative Study. Proceedings of the 21st European Conference on Information Systems (ECIS 2013), Utrecht 2013.

**Paper 5:** Francesco Novelli, Stefan Wenzel: Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity? Proceedings of the 34th International Conference on Information Systems (ICIS 2013), Milan 2013.

### Part IV: App Store for EAS as an application system

**Paper 6:** Stefan Wenzel: Comparing Public and Internal Enterprise App Stores: A Qualitative Case Study.

Paper 6 is the extended version of a conference paper (Wenzel 2014b):

> *Wenzel, Stefan: App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores. Proceedings of the 5th International Conference on Software Business (ICSOB 2014). Lecture Notes in Business Information Processing Volume 182, 2014, pp 227-242.*

The paper was awarded the "Best Paper Award" by the program committee of the ICSOB 2014. Since the extended version includes all the relevant aspects, the conference version was not included in this thesis. Details on the differences between the conference and the extended and revised version are provided in Appendix E.

### Part V: Product Characteristics of EAS for App Stores

**Paper 7:** Stefan Wenzel: App'ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications. Proceedings of the 17th International Conference on Business Information Systems (BIS 2014). Lecture Notes in Business Information Processing (LNBIP) Volume 176, 2014, pp 61-72.

Paper 7 was awarded the "Best Paper Award" of the BIS 2014 conference by the program committee.

### Additional Works in the Context of this Thesis

In the context of this research endeavor, over the past years I have contributed to additional research papers which have been created prior or in parallel to the works included in this dissertation. In part they had a significant influence on this dissertation or have used early research results of this thesis and applied those to adjacent fields. Some of the works are also integrated in the related literature section (cf. Chapter 2) and are introduced there at greater length. Since the interested reader might study selected additional aspects using these works, a comprehensive chronological list for reference is provided in Table 1.

**Table 1:** Additional papers in the context of this thesis

| Year | Citation | Reference |
|---|---|---|
| **2010** | (Wenzel et al. 2010) | Wenzel, S., Neumann, S., Faisst, W., & Bandulet, F. (2010). Electronic Business Services and their Role for Enterprise Software. In A. Benlian, T. Hess, & P. Buxmann (Eds.), Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungs-strukturen (pp. 75–91). Wiesbaden: Gabler. |
| | (Bandulet et al. 2010) | Bandulet, F., Faisst, W., Eggs, H., Otyepka, S., & Wenzel, S. (2010). Software-as-a-Service as Disruptive Innovation in the Enterprise Application Market. In A. Benlian, T. Hess, & P. Buxmann (Eds.), Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen (pp. 15–29). Wiesbaden: Gabler. |
| **2011** | (Müller et al. 2011) | Müller, G., Syring, A., Eggs, H., & Wenzel, S. (2011). Risikomanagement und Vertrauensbildung im Cloud Computing. Wirtschaftsinformatik & Management, 3(5), 66–73. |
| **2012** | (Wenzel et al. 2012) | Wenzel, S., Faisst, W., Burkard, C., & Buxmann, P. (2012). New Sales and Buying Models in the Internet: App Store Model for Enterprise Application Software. In MKWI 2012 (pp. 639–651). |
| **2014** | (Wenzel 2014a) | Wenzel, S. (2014). App'ification of Enterprise Software - Evaluating Mobile App Characteristics Enabling Online Purchase And Their Portability To Enterprise Application Software (pre-print). http://hdl.handle.net/10419/146785. |
| | (Wenzel 2014b) | Wenzel, S. (2014). App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores. In Proceedings of the 5th International Conference on Software Business (ICSOB 2014) - Lecture Notes in Business Information Processing Volume 182 (pp. 227–242), Springer. (Best Paper Award, see above “Paper 6”). |
| **2015** | (Giessmann et al. 2015) | Giessmann, A., Feurer, S., Schneider, T., & Wenzel, S. (2015). Consumers’ Preferences on Platform as a Service and SAP`s Engineering Response (pre-print). http://hdl.handle.net/10419/106129. |
| **2016** | (Nadj et al. 2016) | Nadj, M., Haeußler, F., Wenzel, S., & Maedche, A. (2016). The Smart Mobile Application Framework (SMAF) – Exploratory Evaluation in the Smart City Context. In Multikonferenz Wirtschaftsinformatik (MKWI) 2016 (pp. 1367–1378). |

# 2 Theoretical Foundations and Related Literature

This section introduces works and concepts which are either part of adjacent fields of research or research which has been used in this thesis (e.g., fundamental definitions or frameworks used). Figure 4 provides an overview of the presented streams of research in this chapter. The first objective is to introduce each of the topics to the reader to enable her or him to fully comprehend and assess the research results. Since the topic touches many disciplines it will not be possible to extensively describe each aspect and reference all the relevant research in each field. However, the second objective is to present major state-of-the-art works for each of the research areas and to provide entry points to more detailed studies for the interested reader. Last, this section intends to sharpen the research area of this thesis by reflecting and interpreting the App Store Model for EAS from the point of view of each of the related research areas.

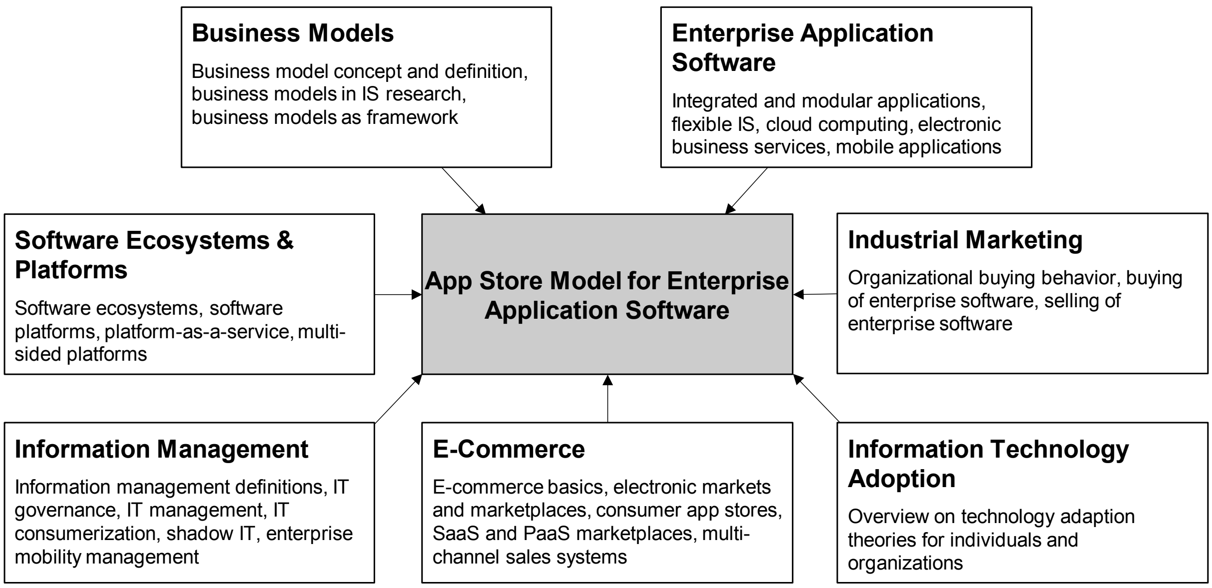


**Figure 4:** Adjacent fields of research (overview)

## 2.1 Business Models

The business model concept, as well as the definition favored in this thesis, have already been introduced in the introduction before to properly derive the motivation, research objectives and the thesis structure (cf. 1.1.2 and 1.2). This section will therefore contain some redundancies but also tries to provide more background and support regarding the reasons why the business model framework has been used in the way presented.

### 2.1.1 Business Model Concept in IS Research

Comprehensive reviews of business model literature with an IS focus have been carried out by Burkhart et al., Al-Debei et al., and Zott et al. (Burkhart et al. 2011; Al-Debei & Avison 2010;

Zott et al. 2011). Moreover the works from Veit et al. (Veit et al. 2014) and Hess (Hess 2012) are recommended for reviewing the status of business models as a dedicated research cluster in IS and regarding the related research agenda.

As highlighted by Hess, business models are a topic of high interest for IS research. While in the early phase of this line of research the focus was on finding a proper definition and classification, the focus has now shifted to capturing, designing, and analyzing technology driven business models (Hess 2012). Veit et al. define four different streams of research in IS on business models (Veit et al. 2014):

1) business model as a fundamental concept and unit of analysis
2) IT support for developing and managing business models
3) business models in IT industries
4) digital business models.

The app store model can be assigned to the third category “business models in the IT industry” with the special instance of enterprise software. While practical business model research in the enterprise software domain is rather rare, a good example in this domain is by Giessmann and Legner on PaaS business models (Giessmann & Legner 2013).

Moreover, Veit et al. and Hess both stress the fact that value creation and value architecture gain a major amount of attention in recent works on business models and are at the core of business models (Hess 2012; Veit et al. 2014; Zott et al. 2011). Consequently the value creation focus of business models differentiates them from pure strategic concepts and makes them the intermediary between strategy and business processes (Al-Debei et al. 2008; Veit et al. 2014). Moreover the value creation view (or value architecture) is preferred for research into the interrelation of digitization, technology, and business concepts (Veit et al. 2014). Hence, the works in this thesis will mainly focus on this aspect of the business model.

### 2.1.2 Business Model Definition

Table 2 lists some selected business model definitions (for a comprehensive overview refer to Al-Debei & Avison 2010).

**Table 2:** Overview of selected business model definitions

| Author(s) | Definition |
|---|---|
| (Amit & Zott 2001) | ▪ "The business model depicts the content, structure, and governance of transactions designed so as to create value through the exploitation of business opportunities." |
| (Chesbrough & Rosenbloom 2002) | "The functions of a business model are to:<br>▪ articulate the value proposition, i.e. the value created for users by the offering based on the technology;<br>▪ identify a market segment, i.e. the users to whom the technology is useful and for what purpose, and specify the revenue generation mechanism(s) for the firm;<br>▪ define the structure of the value chain within the firm required to create and distribute the offering, and determine the complementary assets needed to support the firm's position in this chain;<br>▪ estimate the cost structure and profit potential of producing the offering, given the value proposition and value chain structure chosen;<br>▪ describe the position of the firm within the value network linking suppliers and customers, including identification of potential complementors and competitors;<br>▪ formulate the competitive strategy by which the innovating firm will gain and hold advantage over rivals". |
| (Osterwalder & Pigneur 2005) | "A business model is a conceptual tool containing a set of objects, concepts and their relationships with the objective to express the business logic of a specific firm. Therefore we must consider which concepts and relationships allow a simplified description and representation of what value is provided to customers, how this is done and with which financial consequences." |
| (Stähler 2010; Stähler 2002) | ▪ Value Proposition: Which value does the company create for customers and partners?<br>▪ Value Architecture: How is the value created? Description of value chain (internal value architecture, external value architecture, actors and roles (market design), product model.<br>▪ Revenue model: Which revenues are generated from which sources? |
| (Timmers 1998) | ▪ "An architecture for the product, service and information flows, including a description of the various business actors and their roles; and<br>▪ a description of the potential benefits for the various business actors; and<br>▪ a description of the sources of revenues." |

As described earlier, in this thesis the business model concept recognizes three major components:

- the value proposition
- the value architecture
- the revenue model.

These three components are mainly derived from the definition of business models given by Stähler (Stähler 2002, p.41 ff. Stähler 2010). However, with altering terminology they can also be found in many other definitions of business models (compare definitions in Table 2 or (Al-Debei & Avison 2010). A stated above, the second aspect of the business models, namely the "value architecture" of the App Store model for EAS, has been chosen as a focus of this study. A specific aspect of concern of the business model is that which recognizes the particular context of IT innovations and constitutes the category necessary to derive concrete design recommendations for practitioners.

While Timmers remains rather vague with his definition of the value architecture and only refers to “[an] architecture for the product, service and information flows, including a description of the various business actors and their roles” (Timmers 1998), Stähler defines three further sub-elements: the product and market design, the internal value architecture, and the external value architecture. The market and product design defines which products and product bundles are offered to which market segments. Furthermore, it includes details on product characteristics and configuration. The internal and external value architecture describe the actual value creation process and enable product delivery to the customer. The internal value architecture specifies company internal resources, the configuration of the individual value creation activities (“value steps”), communication channels, the coordinating mechanisms, and a delimitation of value steps conducted within the company and the value steps sourced externally from partners in the value network. Last, the external value architecture includes the interface to customers, the interface to value partners and again communication channels and coordination mechanisms (Stähler 2010, pp.43–46; Stähler 2002).

The structure of the value architecture as presented by Stähler puts a single company in focus by strictly delimitating internal and external architecture. I believe that in the context of the app store model this structure would fall short, since value partners and also customers are viewed as black boxes, i.e., from an outside view. To fully comprehend the value creation process of EAS and its ultimate delivery to the end-user it is indispensable to take into consideration the view inside from value partners and customers, i.e., this study thus looks at the ecosystem (cf. section 2.3) instead of a single company. The sub-components of the value architecture have therefore been adopted to the purpose of this work. The following will be differentiated: the supply-side value chain, demand-side value chain, the app store for EAS as an application system, and product characteristics of EAS for app stores.

### 2.1.3 The Function of the Business Model Framework in this Thesis

With regards to the function of a business model, Stähler explains: “It is a model of an existing business or a planned future business. A model is always a simplification of the complex reality. It helps to understand the fundamentals of a business or to plan how a future business should look like” (Stähler 2010). Veit et al. further elaborate on this: “The business model is seen as a tool for depicting, innovating and evaluating business logics in startups and in existing organizations, especially in IT-enabled or digital industries. In IS research it is also used as a tool, as a unit of analysis and as a framework” (Veit et al. 2014).

The business model framework with its selected components serves multifold functions in this thesis. First, as outlined in the introduction it helps to provide a meaningful structure ensuring the topic is covered comprehensively and establishing the base from which the research questions are derived. Second, it helps in reflecting on the results of the individual works and ultimately integrating them into a consistent framework (see section 3.5 on the methodology of the model-based integration). Third, it supports the interdisciplinary research between IT and business related lines of research. And fourth, it is a practical framework which provides easy access for managers and experts using the results as reference or a guide to implement or improve real-world businesses. At the same time, it also provides a sufficient level of abstraction in order make the results available to a broader set of companies and contexts (but within the boundaries of the domain). And last the framework is also well accepted by academics, ensuring that the results contribute to both groups, practitioners and researchers.

## 2.2 Enterprise Application Software

This chapter will give an overview of the major characteristics of standardized, commercial EAS and will present details of the major trends of EAS, including modular and service-oriented software architectures, smart applications and highly flexible systems, cloud computing, and mobile enterprise applications. Further, the impact of EAS and the trends presented will be associated with the App Store Model for EAS.

### 2.2.1 Characteristics of Integrated Enterprise Application Software

As with any other software, EAS is a digital, hence an immaterial good. In theory EAS as digital good has a marginal cost close to zero: the majority of its cost occurs with the creation of the first version and the reproduction cost – copying software code – is negligible. In practice however there are some limitations to be considered. For EAS to be used productively there is a long-lasting selection, evaluation and buying process to be conducted, and thereafter companies procure implementation, customization, training, and support services (Buxmann et al. 2011, p.23).

Whereas commercial individual EAS is “made-to-order” and the commercial risk often lies with the software customer, the provider of standardized EAS take the full risk of commercial success. Their development cost for the first version are moreover considered as a “sunk cost” since they cannot be influenced afterwards (Buxmann et al. 2011, p.23).

From a procurement point of view EAS is classified as a capital good and the characteristics of organizational buying behavior apply (cf. section 2.4). From an information economics’ perspective EAS is mostly considered as an experience good (in contrast to search or credence goods). This means the value of the software, and its ultimate fit to the business requirements, can only fully be assessed after the application has been used in a real-world setting (Lehmann & Buxmann 2009; Nelson 1970). Some product aspects however may also be classified as a search (hardware requirements) or credence (security features) good. The ability to evaluate EAS and its fit to the business requirements during the buying decision process is an important research cluster for the app store model and will be investigated as part of Paper 3, 4, 5, and 7 (see Chapter III.1, III.2, III.3 and V.1).

From a corporate IS point of view EAS represents the automated part of the IS, or in other words it groups application systems supporting business processes by partly or fully automating business tasks (Ferstl & Sinz 2012, pp.480–485). Conceptually enterprise applications can be categorized into operative systems (including administrative and disposition systems), and planning and control systems. Planning and control systems are also referred to as business intelligence systems (BI), management information & support systems, reporting and analytical systems. Major targets of EAS are the meaningful automation of business tasks and the integrated information processing along the actual business processes, both intra- and inter-company (Mertens 2006, pp.5–14). Since, a complete integration is often too complicated in reality, certain cross-functional and cross-process integration clusters have emerged on the market and are offered as commercial, standardized EAS. The major examples are ERP, CRM, Supply Chain Management (SCM), Supplier Relationship Management (SRM), Product Lifecycle Management (PLM) and Computer Integrated Manufacturing (CIM) (Mertens 2006, p.7). Figure 5 illustrates the functional and process-related scope of these integration clusters. Alongside these typical EAS products there are many hybrid variants, complementing products (such as industry-specific add-ons or variants) and other specialized products available on the market.

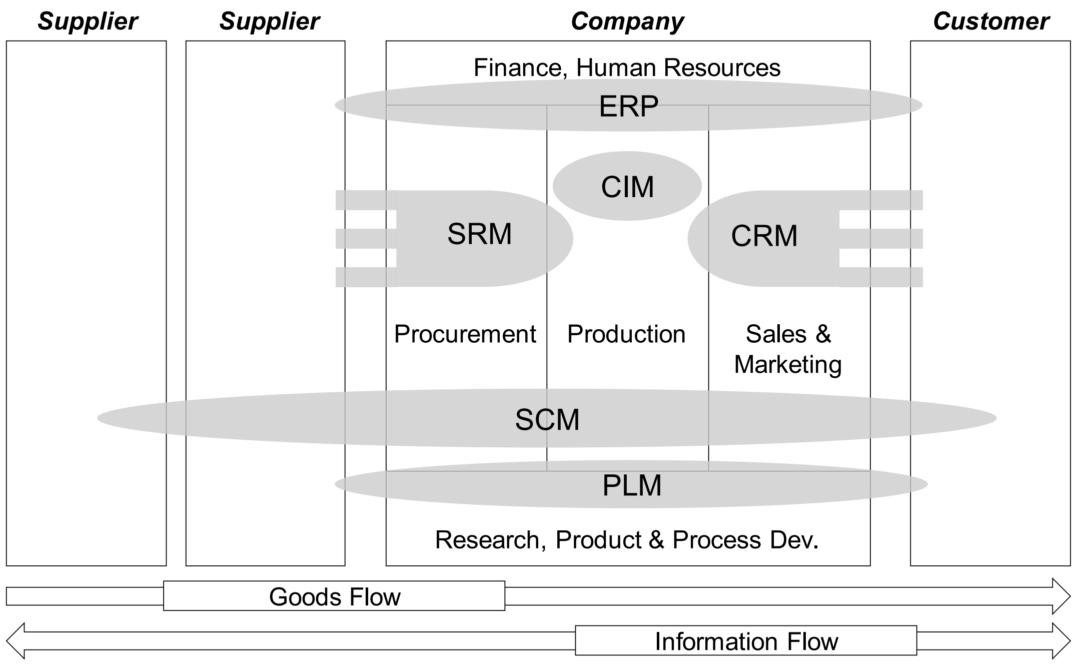


**Figure 5:** Scope of typical enterprise applications (developed from Kemper et al. 2010, p.7))

Since standardized, commercial EAS in contrast to custom software needs to meet the requirements of many, often heterogeneous companies (Buxmann et al. 2011, p.5), more and more features have been added over the years (Klaus et al. 2000; Enquist & Juell-Skielse 2010). Resulting in monolithic bundles with complex configuration logic and customization options. Davenport even refers to ERP systems as “mega-packages” to reflect their feature-rich and monolithic nature (Davenport 1996).

These software applications are typically deployed and operated on the customer’s premises (“on-premise”) and require a sophisticated technological stack (incl. hardware, system software, databases, and middleware, e.g., for integration and mobile device access) which in turn requires an extensive skill set on the customer side to implement, operate and support. Therefore, implementation, customization, and subsequent training of customer employees is time consuming and cost intensive (Aberdeen Group 2006). Furthermore, an entire service industry has been established that is specialized on training, consulting, customizing, and implementing EAS (Buxmann et al. 2011, p.20). Typically, the on-premise enterprise applications itself are charged by upfront, perpetual license (e.g., per user or per activated functionality) fees and additional recurring maintenance fees (Bontis & Chung 2000). The percentage of maintenance fees varies, depending on provider or service levels. However, typical annual fees are 20% of the license value (Buxmann et al. 2011, p.108). In the past ~10 years cloud computing has emerged as a new deployment model and is a serious alternative to the “on-premise” model (cf. 2.2.4).

### 2.2.2 Modular Applications and Service-Oriented Architectures

Despite extensive configuration options, large monolithic enterprise applications often need to be extended or complemented with custom developments to meet customer-specific requirements (Buxmann et al. 2011, p.9). In order to reduce the share of expensive, tailor-made software in companies, EAS providers unbundle the large application monoliths and provide more fine-granular, modular components. The individual modules should be complementary and combinable so that customer-specific requirements can be met with standardized software.

Figure 6 exemplifies the idea of modular EAS from a functional and scope perspective. From the core to the outer shells the functionality of the individual components becomes more company-specific (Lohmann et al. 2001). The inner area includes components (core or horizontal components) relevant to all or most industries such as “Finance & Accounting” or “Human Capital Management”. The next outer shell groups industry specific or so-called vertical components which include functionalities unique or specific to one industry, such as the industry of service providers. The third shell refers to sub-industries or “micro-verticals”, e.g., IT service providers. The outermost shell finally encapsulates the remaining customer-specific components – the so-called “last mile” (Faisst 2011). Ideally the components can be enhanced, exchanged, or updated without affecting other components.

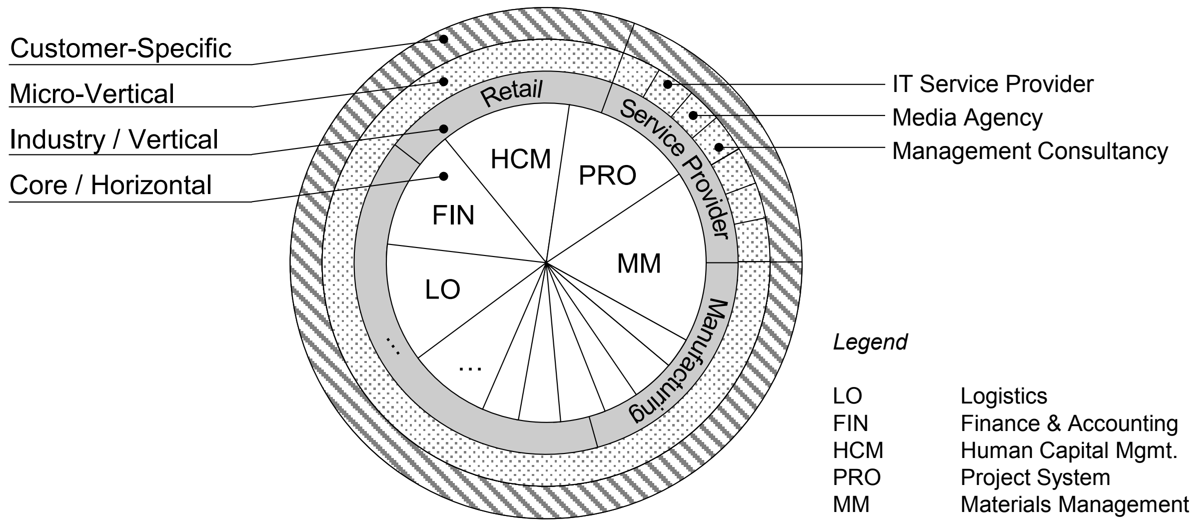


**Figure 6:** Core-shell model of enterprise application software (derived from Lohmann et al. 2001; Faisst 2011)

If different components are delivered by individual providers, benefits through specialization and division of tasks can be achieved. Hence, modular, component-based software is also a basis for SECOs for EAS and thus also a pre-requisite for the App Store Model for EAS (cf. section 1.1.2, section 2.7, Burkard et al. 2012).

In order to enable customer-individual application systems composed out of components from different providers, they also need to comply with common technical and business or content standards (Turowski 2014).

The basis of such modular systems is a so-called component-based architecture or its subsequent development, the service-oriented architecture (SOA). SOA can be understood as a technology independent software architecture pattern for building large, distributed, modular software systems. Applications consist of components which encapsulate functionality to serve a clearly delimited business task (Reinheimer et al. 2007; Becker et al. 2011). The application components provide their functionality to other application components via services. Application components in a SOA can take the role of a service provider or service consumer. Services are loosely coupled and stateless to ensure independence from service availability, location, and consumption; for example, stateless services can be consumed by multiple applications in parallel, without interfering with each other. The services are registered in a central service repository and a separate component, the service bus, orchestrates the communication between the services (Krafzig et al. 2005, pp.59–65).

One major target of SOA is to enable flexible and adaptable application systems which can be used in different contexts and which can be quickly adjusted to changing needs (Reinheimer et al. 2007; Becker et al. 2011). Modern enterprise applications based on SOA promise "modification-free" customizations by changing, removing, or adding modules. The components are either provided by the software vendor of the main solution, 3rd party providers, or by the in-house development of the software customer (Zencke & Eichin 2008).

A SOA is mostly implemented using web technologies, i.e., web services (Becker et al. 2011). A web services-based SOA is also a pre-requisite to enable IT-based "outtasking" by seamlessly integrating company external, electronic business services provided via the Internet (Wenzel et al. 2010, section 2.2.3). Moreover, SOA principles and especially the statelessness of application services is a condition of so-called multi-tenant cloud applications (cf. section 2.2.3, Armbrust et al. 2009).

Although much attention has been paid to the SOA topic in IS research, extensive literature reviews uncover a lack of business orientation in SOA research (Kontogiannis et al. 2008; Viering et al. 2009; Joachim 2011). Hence, the research on app store models for EAS could provide product and marketing insights to SOA researchers and help fill the gap on business orientation.

### 2.2.3 Flexibility of Information Systems and Smart Process Applications

Business process requirements are not only highly heterogeneous from company to company but they also change over time. Both researchers and practitioners have observed an increased demand for higher flexibility of business processes and corporate IS in general (Sinz et al. 2011, p.V; Wagner et al. 2011). Whereas in the past higher flexibility in business processes was often enabled by less automation and media breaks (Sinz et al. 2011, p.V), today corporations demand flexible EAS which support the digitization of these flexible processes. From a systems theory point of view, Wagner et al. define flexibility as "[…] the capability of a system to react to or anticipate system or environmental changes by adapting its structure and/or its behavior considering given objective" (Wagner et al. 2011).

The previously described modular and SOA paradigms are possible foundations to implement flexible IS. However, they do not solve a company's problem in flexibly identifying, acquiring, deploying or integrating a new application when a new process requirement or change arises. This is where the App Store Model for EAS tries to offer solutions to enable application enhancements at short notice. In such cases, the app store model can play an important role in enabling IS flexibility within corporations. At the same time IS flexibility is a pre-requisite for the app store model: new apps which are acquired ad-hoc need to integrate with existing applications, systems and business processes. Hence the governance structures in the company, the existing system landscape, but also the application acquired via the app store need to allow flexible changes to system behavior and/or structure. The governance aspects will be discussed in this thesis as part of Paper 6 (see Chapter IV.1). Requirements of the EAS app itself are part of Paper 7 (cf. Chapter V.1).

The ability of an application to adapt both its behavior and structure to its environment, be it other systems, users, or further environmental variables ("context-aware"), is at the core of a new application type referred to as "Smart Process App" (Bartels & Moore 2013). Nadj et al. have conceptualized this new application type (Nadj et al. 2016) and propose in their framework three "smart activities" and five "smart qualities". The activities sensing (1), decision making (incl. acting) (2), and learning (3) can be assessed along the smart qualities of autonomy, social-ability, proactivity, reactivity, and (energy) resource management. These applications promise – to a certain degree – flexible systems without the need for human intervention, by autonomously adapting its behavior or structure to achieve a given objective. For example, smart process apps may be used for different business process variants with different behavioral or structural needs. They may be used at different customer locations or in different settings and are resilient to ad-hoc environmental changes. The research field of smart process applications is still in its very early stages, but it is highly revealing in the context of the App Store Model for EAS since it addresses some fundamental barriers: new enterprise apps, despite their modular structure or service-orientation, still need to adapt to the customer specific context (e.g., business processes, tasks, IT landscape) and often still require manual, cumbersome setup, and configuration activities – activities which might become obsolete with smart process apps.

### 2.2.4 Cloud Computing

Cloud computing is one of the major IT trends for organizations in the past years. Gartner annually publishes the top 10 IT trends for organizations, and cloud computing, cloud architectures or personal cloud are a regular part of the top 10 trends for 2012, 2013, and 2014 (Pettey 2011; Pettey 2012; Rivera 2013). Also in the report on "Key Issues for IT Executives", cloud computing is ranked second among the top technology trends in 2011 and 2012 (Luftman & Derksen 2012). Yet there is no common accepted definition of cloud computing, though many authors describe similar aspects and characteristics.

One generic definition based on a literature review on cloud computing definitions is provided by Leimeister et al. It already includes many aspects of cloud computing:

> "An IT deployment model, based on virtualization, where resources, in terms of infrastructure, applications and data are deployed via the internet as a distributed service by one or several service providers. These services are scalable on demand and can be priced on a pay-per-use basis." (Leimeister et al. 2010).

Another attempt of a universal definition based on expert interviews is provided by Vaquero et al.:

> "Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs" (Vaquero et al. 2008).

The two definitions are not contradictory, but focus on slightly different aspects. To summarize, the cloud customer, according to both definitions, consumes "cloud resources" or cloud services via the internet. Since the services can be scaled as requested ("on demand" can be interpreted as real-time) the user typically pays on a use basis. Typically, three variants of cloud computing or cloud services are delimitated, as presented in Table 3.

**Table 3:** Three variants of cloud computing (Vaquero et al. 2008; Weinhardt et al. 2011; Weber 2009)

| # | Cloud Service Type (Short) | Description |
|---|---|---|
| 1 | Software-as-a-Service (SaaS) | SaaS provides software applications via the internet to end-users. Typically, many customers use one instance of a SaaS application (multi-tenancy). A SaaS provider can again use 3rd party PaaS or IaaS to provide his service. |
| 2 | Platform-as-a-Service (PaaS) | PaaS provides developers (individuals or companies, e.g. ISVs) software, hardware, and additional tools as a service to develop, test, deploy, and operate SaaS applications or even other types of applications (e.g., mobile apps). A PaaS provider can rely on 3rd party IaaS to provide his service. |
| 3 | Infrastructure-as-a-Service (IaaS) | IaaS provides basic IT resources as a service, such as computing, storage or networking capacity. |

In particular, SaaS and PaaS will be of interest for this thesis. PaaS will be examined in more detail in section 2.7 and is also a focal topic of the research of this thesis in Paper 1 and 2 (cf. section II.1 and II.2)

The major difference of SaaS applications versus traditional applications is illustrated by a combination of deployment model and pricing model in Figure 7. Traditional software is deployed on the customers' own technology stack, incl. hardware and system software (operating system, database, middleware etc.). The customer also operates the entire stack and is responsible for installing updates. From a pricing perspective, traditional software is charged with one-off license fees and recurring maintenance and support fees. On the contrary, SaaS is fully operated by the software provider. The end user typically consumes the software using a web browser and the customer company does not require any central IT infrastructure to use the software. The customer pays one subscription fee which typically includes the right to use the software, the operation and provisioning of the software, and also support and maintenance services. The two described scenarios are also available in hybrid variants, of which the so-called Application Service Providing (ASP) is the most famous and which was very popular in the 1990s (Buxmann et al. 2008; Bandulet et al. 2010).

| | | Pricing Options | |
|---|---|---|---|
| | | License | Subscription |
| Deployment Options | On-Premise | **Traditional Software** | **Hybrid** |
| | Hosted | **Hybrid (ASP)** | **Software -as-a-Service** |

**Figure 7:** Pricing and deployment of software (Bandulet et al. 2010)

One often mentioned characteristic of SaaS is the potential to realize economies of scale by operating the application on shared hardware and as part of a multi-tenant-architecture – which is a technology to use the same application for more than one customer. Customer specific data is separated and encapsulated from the application logic (Matros et al. 2010; Weber 2009, p.24). Another advantage for providers is a massive simplification in application maintenance. In contrast to traditional software, SaaS is ideally only available in the latest version and customers cannot choose to stay on an old version of the application. Consequently providers have to maintain only one software release (Föckeler 2010).

Many EAS providers believe SaaS will be the dominant software delivery and business model in the future and has the potential to disrupt the industry (Anding 2010). Bandulet et al. reviewed the most prominent SaaS providers in the EAS market using financial statements and found out that at the time of the study highly standardized applications for single, often non-critical, and rather isolated domains seem to be well suited for the SaaS model, such as salesforce automation and other parts of CRM (Salesforce.com, SugarCRM, RightNow), talent management as part of human resources (SuccessFactors, Taleo), or travel expense management (Concur). Whereas customer adoption of more integrated solutions covering typical ERP or SCM functionality is still at a low level (Bandulet et al. 2010). Recent announcements in the industry indicate large providers such as SAP or Microsoft also offer cloud variants of their core ERP solutions (IS-Report 2016b; IS-Report 2016a).

One of the major benefits for SaaS customers is the significant simplification with regards to technology but also the pricing model. SaaS customers do not need to worry about the underlying software stack or hardware, they simply need to evaluate service level agreements and the functionality provided. If the customers consume multiple SaaS applications from different providers it is also not relevant if the applications are built using the same technology, but only if they can communicate with each other, i.e., if they can be integrated. In that regard SaaS has the potential to fulfill the promise of SOA (cf. section 2.2.2 and Anding 2010). Standardized web technologies ensure (technical) interoperability between SaaS providers. This fact has greatly reduced the barriers for a market of diverse SaaS offerings and SaaS marketplaces where established. Salesforce.com is often referenced as a pioneer in this domain, who launched the AppExchange in 2006 (Burkard et al. 2011), even two years before Apple launched its consumer app store for mobile devices (Buxmann et al. 2011, p.74). SaaS marketplaces are in essence a variant of app stores for EAS as they are defined in this thesis. The SaaS marketplace in this context can define standards to also ensure semantic interoperability between the individual applications.

Major risks of SaaS for customers identified by Müller et al. include security, availability, and controllability (Müller et al. 2011). SaaS marketplaces can counter these risks by acting as a trusted party, by defining standards, and by acting as a regulatory body controlling adherence to these standards.

Another important aspect of SaaS is the replacement of high upfront cost for licenses and hardware with continuous subscription fees for a service. This allows customers to change providers without fearing a sunk cost. Hence, SaaS inherently reduces the risk of a false investment and provides higher strategic and operative flexibility (Benlian et al. 2010). Given the nature of SaaS applications, an IT department is not required to pre-finance the applications, since business units can use their budget to “pay-as-they-go”, nor is an IT department required to provide the technical infrastructure. This might explain the guerrilla adoption of SaaS applications by business users (Wenzel et al. 2012). This phenomenon is also referred to as “Shadow IT” and will be further discussed in Chapter 2.5 on Information Management.

SaaS characteristics (e.g., reduced pre-requisites on IT infrastructure, no upfront investment through ‘pay-as-you-go’ pricing) are major enablers for online purchasing on an app store for EAS. These characteristics will be further evaluated, especially in Paper 7 (cf. Chapter V.1)

### 2.2.5 Electronic Business Services

Beyond the previously introduced variants of cloud computing IaaS, PaaS, and SaaS, there are multiple other variants, partly coined by marketing departments. The inflationary use of “as-a-service” has also led to the acronym XaaS, “Everything-as-a-Service”. Noteworthy, however, is the concept of “Business-as-a-Service” (BaaS) (Rimal et al. 2009).

This concept in principle describes electronic business services offered via the internet. For the customer, a digital or electronic business service is similar to a SaaS offering – the difference is that an electronic business service is not only an application to be used by an end-user, but provides the execution of one or multiple business tasks or even entire business processes outside the company (e.g., processing of payrolls or credit scoring). The request of the business service and the delivery of the result of the business service is conducted electronically. The previously described reduction of barriers of integrating different cloud services, also enables easy consumption of electronic business services. Since electronic business services externalize the execution of business tasks and business processes their service in essence is business outtasking and outsourcing (Wenzel et al. 2010). In contrast to typical long-term planned outsourcing projects, electronic business services are modular, fine-granular, and are often requested ad-hoc or only once. Such use cases would not be worthwhile if they required expensive and long-lasting setup projects in terms of identifying a business service, setting up contracts, integrating it in EAS. Therefore the app store models for EAS can be the ideal breeding ground for these services by providing an efficient platform to search, buy, and consume such services (for a comprehensive analysis of electronic business services refer to (Wenzel et al. 2010)).

### 2.2.6 Mobile Enterprise Applications

Originating in consumer markets, a further major trend in EAS is mobile computing and the corresponding mobile applications or short mobile apps. Accordingly, Luftmann and Derksen rank mobile computing, app development and mobility among the top 10 technology trends for IT organizations in 2010, 2011, and 2012 (Luftman & Derksen 2012). Equally Gartner lists mobile applications among the top 10 technology trends for companies for 2012, 2013, and 2014 (Pettey 2011; Pettey 2012; Rivera 2013). Moreover, the SAP Store, the proprietary app store of SAP, listed 323 separate downloadable and purchasable mobile solutions (as of January 2015) either provided by SAP itself or its partners (SAP 2015a).

Typically, mobile applications for enterprises are not standalone applications as many of their consumer counterparts are. Instead they are connected to a backend system where the integrated business logic and the central data management reside. The mobile application therefore represents one user interface (UI) to a business application (Jin et al. 2014). Furthermore, the backend application typically supports integrated business processes in which the mobile app is only partially involved, mostly for one or a few business tasks. Jin et al. examined the market of enterprise mobile apps and evaluated in which industries, business areas mobile apps are available and which are adopted the most. From an industry perspective by far the top category for available apps was “multi-industry”, hence these apps were not industry-specific, followed by retail and consumer product industries. From a business area perspective, the leading categories were sales, customer service, and human resources, the first two being areas were employees are traditionally en route. The third, human resources, mainly includes employee-self-service apps (e.g., such as approval workflows for vacation or business travel). Not surprisingly, apps from this category have been adopted the most, as they are applicable for the entire staff and not just a small sub-group (Jin et al. 2014).

Though the term "app" already appeared in the context of cloud or web applications even before mobile apps hit the market (cf. "AppExchange" by Salesforce which was founded 2006, two years ahead of the launch of the Apple App Store, section 2.2.4) the term is strongly linked to mobile computing. However, there are few scholars who have tried to define what a mobile app is. In the context of how to turn web applications into mobile applications, Sodhi et al. highlight four characteristics of mobile apps design (Sodhi et al. 2012):

- "Sharp focus on delivering only specialized functionality.
- Workflow simplification, for instance, by leveraging device sensors.
- Architected to harness the client device's computing and sensing capabilities.
- Focused on tapping the greater user intimacy that the client devices enjoy".

The major difference between a mobile application versus a browser-based web application is thereafter in the design and scope of the application which is tailored to the specifics of the mobile use context, and the device characteristics, as well as the greater focus on hedonistic aspects as user experience (Sheng & Teo 2012). In addition, Jin et al. mention the different use scenario of mobile enterprise apps versus traditional desktop applications: unplanned, ad-hoc, and frequent use, while having a limited display size. Therefore they need to be simpler and highly focused with regards to functionalities supporting specific business tasks in contrast to complex feature collections as in traditional desktop applications (Jin et al. 2014).

Practitioners identify generic design patterns which can be applied to domains other than mobile, including traditional desktop applications (Kosner 2012). For instance, SAP SE, one of the major vendors for EAS, believes that design principles of mobile computing have the potential to redefine business processes and ultimately also EAS itself and has updated its design guidelines following design principles rooted in mobile, consumer apps for all types of applications (SAP 2013c; SAP 2013b). Moreover SAP provides more than 400 so called Fiori Apps as bundle on top of the SAP Business Suite (including mainly SAP ERP) (SAP 2015b). SAP Fiori apps are based on HTML5 and the "responsive user interface paradigm", so that the apps work on desktop, tablets, and smartphones (SAP 2013a).

The success and broad adoption of app stores for mobile apps in the consumer segment make these types of applications and products relevant to this research, since an evaluation needs to be carried out into whether the characteristics of mobile apps (especially commercial app products) are conducive for online purchase and distribution and if they can be applied to the domain of EAS, both mobile and non-mobile. This research question will be mainly addressed in Paper 7 (see Chapter V.1). The management and distribution of mobile applications within the company will be addressed in section 2.5.3.

## 2.3 Software Ecosystems and Platforms

In the software industry large players have started to follow a strategy known as "platform strategy", which is based on modularity, combined with innovation strategies involving third parties (Bosch 2012; Burkard et al. 2012; Hess 2015; Scholten 2011, p.15; Selander et al. 2013). These innovation strategies are also referred to as "value networks", "value co-creation", or "ecosystem innovation" (Ceccagnoli et al. 2012; Leimeister et al. 2010). The two elements combined are hence referred to as "platform-based ecosystem innovation" (Gawer & Cusumano 2013, p.1)

In striving to keep up with the pace of innovation as demanded by the market, and to meet the numerous heterogeneous requirements of different industries, especially in the domain of EAS, companies are limited if they rely on internal innovation capabilities only. Innovation in a network of partners is perceived as more effective and efficient for covering the demand for customizing and features (Kim et al. 2010; Iansiti & Levien 2004a; Bosch 2012). In other words, ecosystems have the potential to enhance the internal capabilities and capacities of a company (Iyer et al. 2007).

Aside from the concept of electronic marketplaces (cf. section 2.6.1), the App Store Model for EAS relies on a software platform and an associated software (platform) ecosystem.

### 2.3.1 Software Ecosystem

SECOs is a trending topic in IS and is studied from many different angles including software engineering, economic, social, and network theoretical perspective (cf. Barbosa & Alves 2011). However, since SECOs are a subset of business ecosystems they are not a complete novelty (see also Moore 1996). A business ecosystem is defined as "an economic community involving many companies working together to gain comparative advantages as a result of their symbiotic relationships" (Kim et al. 2010). The definition of such an ecosystem is very broad and does not restrict the form of collaboration among different companies. It therefore includes everything from traditional linear supplier-customer relations, via trade-networks including multi-level distributors or outsourcing relationships to interwoven co-innovation networks (Iansiti & Levien 2004a).

Jansen and Cusumano have based their definition of SECOs on existing ecosystem research in biology and economics, and existing IS definitions of SECOs: "A software ecosystem is a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts” (Jansen & Cusumano 2012).

**Classification of Software Ecosystems**

Jansen and Cusumano introduce four dimensions to further classify SECOs (see Figure 8). The light grey boxes highlight the variants which are relevant for the App Store Model for EAS in this thesis.

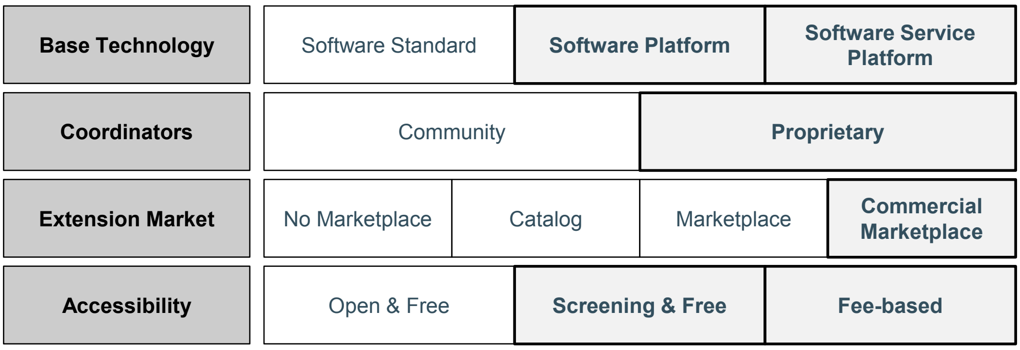


**Figure 8:** Classification of SECOs applied to the App Store Model for EAS (Jansen & Cusumano 2012)

First they assume that all SECOs are underpinned with a technology, and they distinguish software standards, software platforms, and software service platforms (Jansen & Cusumano 2012). A software standard, for example, can be a message protocol or file format, while software platforms are software systems themselves (e.g., operating systems, core business applications, or a stack of reuse software modules) and are comprehensively discussed in the next section 2.3.2. A software service platform is a software platform which exposes their services and tools online. In this thesis, I refer to this type of base technology as platform-as-a-service (PaaS). The concept of PaaS is briefly reviewed in section 2.3.3 and furthermore, represents a distinct research artefact as part of this thesis in Chapter II.1. SECOs that are relevant to the App Store Model for EAS are always based on a software platform or software service platform (i.e., PaaS). Some authors refer to these types of SECOs as software platform ecosystems (Jansen & Bloemendal 2013; Evans et al. 2006; Schreieck et al. 2016).

The coordinators of a SECO define rules, methods, standards, and tools used by the ecosystem to create value. An ecosystem can either be coordinated by a community (e.g., often applicable in open source projects) or by a company, which is mostly also the owner of the base technology (Jansen & Cusumano 2012). For the App Store Model for EAS in this thesis, only proprietary coordinators are relevant, since most EAS is commercial software and the ecosystems evolve around proprietary commercial technology (e.g., Apple iOS represents a proprietary commercial technology).

The dimension extension market defines whether there is a marketplace coordinating the distribution of software applications created by the ecosystem to the software customers. In case there is no coordinating marketplace the individual companies market their applications themselves. The first central mechanism is a simple catalog, listing the available software applications of an ecosystem but not supporting the actual procurement or download. A marketplace without commercial functions supports the distribution and is mainly used for free software applications. A commercial marketplace supports commercial transactions and the owner typically receives a commission fee from the seller for providing the marketplace and its services (Jansen & Cusumano 2012). The App Store Model for EAS is inherently based on a commercial electronic marketplace (cf. section 2.6.1).

Accessibility determines who can participate in an ecosystem and under which conditions. Jansen and Cusumano have identified three distinct categories. In case there are no barriers anyone can participate in the ecosystem without having to pay any fees. This is the case in many open source-based ecosystems. Even if these ecosystems are free and basically accept any contribution, there might still be conditions and regulations under which complementary solutions may be created or published (for example see GNU General Public License Free Software Foundation 2016). To ensure quality control, screening of complementors and/or their complementary software solutions without asking a fee is one level of accessibility control. Last, in addition to screening there can be a fee (either a pure entry fee, commission fee or other forms of fees) (Jansen & Cusumano 2012). For the App Store Model for EAS, complementors will be screened and mostly a fee is required. However, options where complementors may freely contribute are equally applicable. The complete restriction-free open design will in most cases not fit the commercial nature of EAS.

The objectives of software ecosystem are described and structured by Popp and Meyer (Popp & Meyer 2010, p.131 ff.). They identify five goal categories pursued when establishing an ecosystem-based strategy: financial, customer related, product related, network effect related, or market related goals.

### Roles in Software Ecosystems

The actors in SECOs have different roles and fulfill different tasks in the overall value creation. The literature on SECOs uses many different terms for partially overlapping roles. Table 4 lists the most commonly used terms in ecosystem and SECO research for the different roles of the ecosystem actors and their respective tasks

**Table 4:** Roles and respective tasks of actors in SECOs

| Roles used in literature | Terms used in this thesis | Key tasks |
|---|---|---|
| ▪ Platform sponsor and platform provider (Eisenmann et al. 2009)<br>▪ Keystone (Iansiti & Levien 2004b)<br>▪ Hub (Burkard et al. 2012; Kude et al. 2012)<br>▪ Coordinator (Jansen & Cusumano 2012)<br>▪ Orchestrator (Jansen et al. 2009; Angeren et al. 2013)<br>▪ Platform or Ecosystem Leader (Gawer & Cusumano 2008; Jansen & Cusumano 2012)<br>▪ Platform owner (Boudreau 2010; Ceccagnoli et al. 2012; Scholten 2011) | Platform provider | ▪ Develop base or platform technologies, products, and services (Gawer & Cusumano 2013)<br>▪ Define technology, rules, components, and accessibility of platform (Eisenmann et al. 2009)<br>▪ Provide platform product and services (Burkard et al. 2012)<br>▪ Mediate transactions between demand and supply on the platform (Eisenmann et al. 2009; Eisenmann et al. 2006))<br>▪ Create and distribute value (Iansiti & Levien 2004b)<br>▪ Ensure ecosystem health (Iansiti & Levien 2004b)<br>▪ Coordinate ecosystem (Jansen & Cusumano 2012)<br>▪ Curate marketplace (Jansen & Bloemendal 2013) |
| ▪ Complementors (Boudreau 2008; Eisenmann et al. 2006; Yoffie & Kwak 2006)<br>▪ Spoke (Kude et al. 2012)<br>▪ Ecosystem follower (Jansen et al. 2009)<br>▪ Independent software vendor (ISV) (Ceccagnoli et al. 2012; Evans et al. 2006)<br>▪ Partner (Angeren et al. 2013)<br>▪ Niche player (Iansiti & Levien 2004b; Burkard et al. 2012)<br>▪ Supply-side (platform) user (Eisenmann et al. 2009)<br>▪ Platform consumer (Giessmann et al. 2012)<br>▪ Third-party or application developer (Giessmann & Legner 2013; Selander et al. 2013; Heitkoetter et al. 2012) | ISV or complementor | ▪ Develop components & complements (Manikas & Hansen 2013)<br>▪ Provide functionality required by the customers (Manikas & Hansen 2013)<br>▪ Create value to the ecosystem (Iansiti & Levien 2004b)<br>▪ Develop niche solutions (Iansiti & Levien 2004b)<br>▪ Sell solutions to customers (Burkard et al. 2011) |
| ▪ Demand-side user (Burkard et al. 2012)<br>▪ (Software) Customer (Manikas & Hansen 2013) | Software Customer | Purchases or obtains and uses a complete or partial product of the software ecosystem (Manikas & Hansen 2013) |

Table 4 groups the actors identified in the literature in three major categories. Some individual authors, however, differentiate further roles. For example, Eisenmann et al. differentiate the role of platform sponsor and provider (Eisenmann et al. 2009), whereas in this thesis the terms are used synonymously. Manikas and Hansen differentiate between the actor creating a niche or complementary solution and the one selling it, such as a Value-added reseller (Manikas & Hansen 2013) – this role is not considered explicitly since in the app store model channel intermediaries play a secondary role. For the purpose of the App Store Model for EAS we focused on three major roles: the platform provider, the complementor or ISV (used synonymously), and the software customer. With regards to the tasks and the objectives of the different roles existing literature is very vague and abstract. This thesis will contribute to achieving greater clarity about the concrete tasks and objectives of each of the roles. The complementors and the platform provider's supply-side activities will be analyzed in detail in Chapter II.1. The software customer and the platform provider's demand-side tasks are the focus of Part III: Demand-Side Value Chain.

**Benefits and Challenges of a Software Ecosystem**

The potential benefits and challenges of a software-ecosystem-based strategy are the basis on which to derive the value proposition of a software-ecosystem-based business model. Bosch (cf. Bosch 2012) and Barbosa and Alves (Barbosa & Alves 2011) have identified potential benefits for companies to innovate via SECOs over innovating within the boundaries of a single firm:

- increased value of the core product to existing and new customers;
- increased "stickiness" of the product/platform (switching cost increase for customers if they have multiple products of the same platform);
- higher pace of innovation, higher speed to market;
- decreased cost of innovation and development: incremental innovations (components) developed by multiple companies;
- preferred opportunity to internalize selected components developed by ecosystem partners into the core product or platform;
- shared product maintenance and customer support among ecosystem partners.

Barbosa & Alves have further identified a set of challenges and limitations of SECO strategies:

- platform architecture challenges: interface stability, feature evolution management, security, reliability;
- finding a compromise between openness of ecosystem versus ensuring standards;
- coping with heterogeneity in the ecosystem (e.g., software licenses);
- establishing relationships among and management of ecosystem actors;

- compensating for the loss of competitive advantage and differentiation by giving up parts of the value chain to the ecosystem;
- establishing a comprehensive landscape of tools and services to foster the co-creation value chain (development, sales, support, social interaction, decision making, and governance).

### 2.3.2 Software Platform

The software platform is the underpinning technology of a software platform ecosystem, as previously outlined. The term software platform or platform, however, is used in many contexts and in IS research there is no common definition. Hence, I first present several common definitions of platforms.

**Software Platform Definition**

- Evans et al. take a very lean and technical view: "[…] a software platform [is] a software program that makes services available to other software programs through Application Programming Interfaces (APIs)" (Evans et al. 2006).
- Hess defines software platforms as software systems composed of a stable core and enhancing software modules or applications (Hess 2015).
- Buxmann et al. differentiate between product and industry platforms: Product platforms enable the efficient creation of products and service via reuse of already created modules. Industry platforms allow the creation of complementary products or services by third parties from within the industry (Buxmann et al. 2011, p.189).
- Gawer and Cusomano define platforms as "[…] products, services or technologies developed by one or more firms, and which serve as foundations upon which a larger number of firms can build further complementary innovations, in the form of specific products, related services or component technologies" (Gawer & Cusumano 2013).
- Boudreau includes tools and rules as part of his platform definition: "The platform consists of those elements that are used in common or reused across implementations. A platform may include physical components, tools and rules to facilitate development, a collection of technical standards to support interoperability, or any combination of these things" (Boudreau 2010).
- Eisenmann et al. provide a very broad definition of platform: "The platform encompasses the set of components and rules employed in common in most user transactions. Components include hardware, software, and service modules, along with an architecture that specifies how they fit together. Rules are used to coordinate network participants' activities. They include standards that ensure compatibility among different components, protocols that govern information exchange, policies that constrain user behavior, and

contracts that specify terms of trade and the rights and responsibilities of network participants" (Eisenmann et al. 2009).

All definitions given are complementary and can be applied to the App Store Model for EAS. The definitions by Evans et al. and Hess mostly address the architectural aspects of modular software. Modularity in EAS, as an enabler of the app store model, has already been introduced in section 2.2.2. Buxmann introduces two platform categories, whereas the product platform definition also displays large similarities with the modular software paradigm and its objectives (e.g., reuse of modules), the industry platform widens the scope to multiple organizations and introduces the ecosystem aspect. The App Store Model for EAS can clearly be assigned to the industry platform variant under this definition. The definition by Gawer and Cusomano is very similar to the industry platform definition but is more precise with regards to the different roles in co-creation. One company creates platform products, services, technologies and multiple other companies create the complements.

The last two definitions provided by Boudreau and Eisenmann et al. are broader than the previous ones. They not only cover product or service components as core or complementary ones, but encompass all components incl. hardware, software, services, tools, and rules, incl. standards, protocols, policies, terms, and contracts, which are defined to support or regulate the creation and trade of platform or complementary products and the use of the platform components. The app store model in the light of this definition includes components and rules for both ISVs and software customers. The components for ISVs, for example, would also entail development tools to create complementary software products and a developer or partner program which defines the "rules" of complementary product creation, sales, and support. The perspective of ISVs in the app store model is investigated comprehensively in Chapter 5.4.1 and II.2.

Schreieck et al. have analyzed research on platform ecosystems but have not provided their own definition of platforms. Instead they identified two different perspectives on platforms in the existing literature: the technical-oriented and a market-oriented perspective. Whereas the technical-oriented perspective focuses on software or hardware components to be reused in a larger system, the market-oriented perspective focuses on an intermediary connecting two market sides via a marketplace or community (Schreieck et al. 2016). The App Store Model for EAS covers both the market as well as the technical-oriented view.

**Platform Design Options**

Burkard et al. have identified three major design decisions a platform provider has to take: the management of the complementors, the openness of the platform, and the revenue model (Burkard et al. 2012). The management of complementors will be a topic of this thesis in Chapter II.1 where the lifecycle of a complementor on a platform will be investigated.

The openness of a platform deals with how restricted the access to the platform is in terms of using the platform and its services, developing and selling complementary solutions. Eisenmann et al. have made this design question focal topic of their research "Opening Platforms: How, When and Why?" (Eisenmann et al. 2009). They differentiate between vertical and horizontal openness, where vertical openness is the openness of the platform towards the complementors. According to Eisenmann et al. a platform provider needs to make decisions in the following areas when 'opening vertically':

- What are the rules of backward compatibility when upgrading the platform?
- Shall complementors be exclusively on the platform (forbid so called multi-homing)?
- Shall the platform grant category exclusivity for selected lighthouse complementors (i.e., allowing a company to cover a certain segment or functionality exclusively)?
- Which complementary solutions should be acquired and absorbed by the platform?

Horizontal openness refers to opening the platform to other mostly competitive platforms. Horizontal openness or interoperability between two platforms allow users of the two platforms to transact with each other (Eisenmann et al. 2009).

The third design area for a platform provider according to Burkard et al. is the revenue model. From the platform provider's perspective, he needs to decide if he charges both sides, that is the complementors and the software customers (cf. Figure 9). If he charges the complementors, there are three common options on software platforms according to Giessmann et al. (Giessmann et al. 2012):

- a 'revenue share' on the revenue complementors make with the software customers;
- a flat subscription fee for platform or ecosystem access and the services provided;
- a 'pay per use' model for services provided by the platform;
- any combination of the above.

### 2.3.3 Platform-as-a-Service

PaaS represents one layer of the cloud business model stack (cf. Table 3 in section 2.2.4). It is the "provision of a complete platform, i.e., hardware AND software as a service to develop and to provide SaaS solutions […]" (Grohmann 2009). This means the platform concept as previously introduced is developed further in that it raises the claim to completeness (from development to provisioning) and productization so that it can be consumed 'as a service'. Consequently, it evolves from a strategic option for a product company to a primary standalone business model for a platform provider. Initially PaaS emerged only in the context of the development of cloud applications but is nowadays equally used to create all kinds of software applications (e.g., mobile apps) (cf. section 2.2.4). Giessmann et al. have (Giessmann et al.

2012) identified the ten important aspects of a PaaS offering complementors will assess when deciding on a PaaS offering:

- development environment
- test environment
- monitoring (of application operations)
- mobile access to platform services
- community features
- market penetration (= market share of platform)
- pricing
- marketplace functionalities (= app store)
- payment handling by marketplace
- migration among PaaS offerings.

Since PaaS is the ideal instance of a software platform offering for the app store model, the PaaS concept will be comprehensively introduced in this thesis in Chapter II.1. The requirements above will be reviewed and consolidated in a comprehensive PaaS stack. Moreover, it will be differentiated between pure PaaS and so called “application-based” PaaS (aPaaS) offerings where PaaS is underpinned by a core application.

### 2.3.4 Multi-Sided Platform

As previously explained, the software platform in the app store model serves two distinct types of organizations: the ISV and the software customer. In doing so it makes the platform a so-called two-sided or more generally a multi-sided platform: “Software platforms are inherently multisided. They usually serve distinct groups of customers, who benefit from having each other on the same platform. Application Programming Interfaces (APIs) forge the crucial relationship between application developers and end users. The developer can benefit from using APIs when she can sell the resulting software to users who have those APIs on their computing devices” (Evans et al. 2006, p.39). Typically, the multi-sided platform creates value by intermediating demand and supply, i.e., by facilitating the transactions of demand and supply platform customers. In contrast to traditional linear supply chains, two-sided platforms facilitate tri-angular relationships (Scholten 2011, p.23; Eisenmann et al. 2007) (see Figure 9).

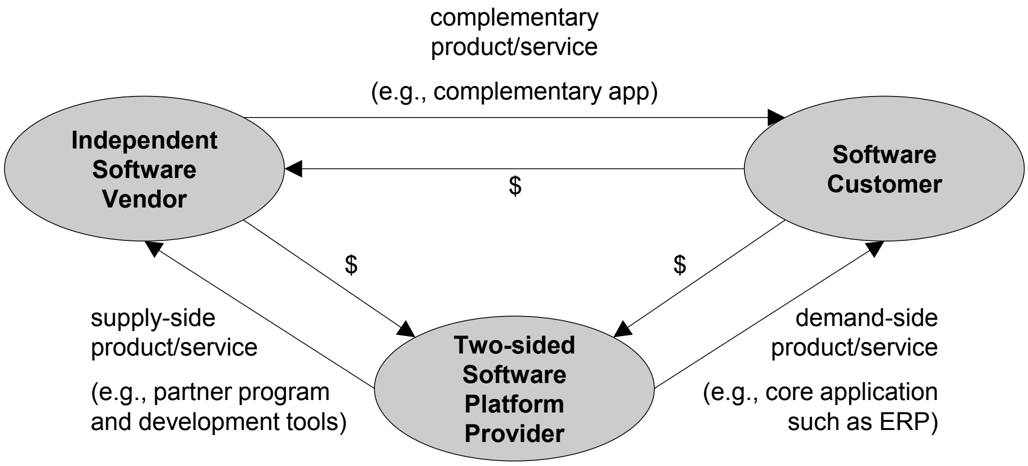


**Figure 9:** Tri-angular relationship in two-sided platforms (cf. Eisenmann et al. 2009, p.4)

The two-sided platform provider is able to sell and charge two distinct product or service portfolios to two distinct markets (Scholten 2011, p.24). Hence in economics this framework is also referred to as a two- or multi-sided market. The following characteristics apply to two-sided platforms (Evans et al. 2006, p.53ff.):

- The platform addresses two distinct customer groups.
- The platform generates value by connecting the two different groups.
- Indirect network effects occur on the platform.

Platforms in common are associated with network effects. Generally one speaks of a network effect if the benefit of a product or service increases with the number of customers / users (Burkard et al. 2012). In multi-sided platforms two types of network effects are distinguished (see Figure 10): direct (or ‘same-side’) and indirect (or ‘cross-side’) network effects.

Direct network effects are present if the value of the platform for one platform customer group increases with the number of customers of the same group, whereas indirect network effects occur when the benefit of one platform customer group increases with number of the other platform customer group (Burkard et al. 2012; Eisenmann et al. 2009; Eisenmann et al. 2007). Both network effects are often present on multi-sided platforms, however, only cross-side network effects are a defining characteristic.

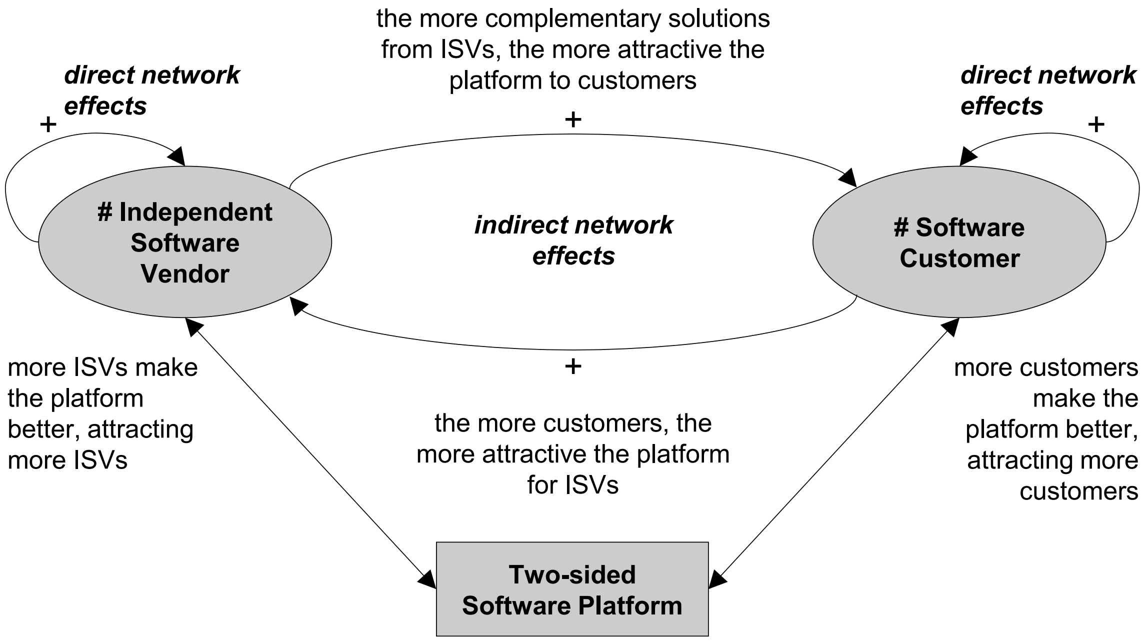


**Figure 10:** Simplified diagram illustrating direct and indirect network effects on two-sided platforms

An example of direct network effects are communication standards: the more users use e-mail technology the more valuable the technology is to both new and existing users. Indirect network effects in the case of the app store model means the app store and the platform are more valuable to software customers the more complementary apps are available (provided by ISVs), and vice versa, it is more valuable to ISVs the more software customers have the platform core product to procure complementary software apps via the app store.

For the platform provider the knowledge about the in-direct network effects are crucial especially in the early stages of a platform business when numbers of one or both platform customer groups are low. The problem of lacking either of the customer groups is also referred to as the “chicken and egg problem”. To overcome this issue, software platform providers may choose to kickstart the platform by offering a number of their own complementary solutions to attract software customers first and then attract additional ISVs onto the platform (Burkard et al. 2011; Burkard et al. 2012).

## 2.4 Industrial Marketing

Purchasing software can also be examined from a marketing, and in the case of EAS, from an industrial marketing perspective. Therefore, section 2.4.1 presents basic concepts of organizational buying behavior, before in section 2.4.2 existing research on the procurement of EAS is introduced.

### 2.4.1 Characteristics of Organizational Buying Behavior

Organizational buying differs in many aspects from consumer buying behavior. Table 5 contrasts typical characteristics of consumer versus organizational buying (see also Backhaus & Voeth 2011, pp.35–37; Kotler 2003, p.215 ff. Foscht & Swoboda 2007). In practice, however these stereotypical classifications are softened up and should only be perceived as a tendency. For example technologies, such as purchasing marketplaces (Ariba 2015) for highly standardized products also enable companies to easily switch suppliers or make ad-hoc buying decisions for orders with small volumes. Moreover, organizational procurement is not always centralized and conducted by the purchasing department, but many individuals conduct purchase decisions – either as part of a corporate program where employees can use corporate credit cards without approval up to a certain budget or they just act in non-compliance (Karjalainen et al. 2009). In such a case it cannot be excluded that employees behave like consumers even in their work context and make buying decisions including for emotional and subjective reasons.

**Table 5:** Typical characteristics of consumer versus organizational buying

| Category | Consumer buying | Organizational buying |
|---|---|---|
| Products | ▪ simple products and services<br>▪ highly standardized<br>▪ low priced or low quantities | ▪ often complex products or services<br>▪ tendency to request customization<br>▪ high price or high quantity |
| Customer | ▪ anonymous market<br>▪ many, small customers | ▪ transparent market<br>▪ few, large customers |
| Buying decision / process | ▪ decisions are strongly emotional and subjective<br>▪ individual need often occurs ad-hoc<br>▪ decisions are less formalized and have a low process orientation<br>▪ pre-dominantly individual decisions | ▪ buying decisions are highly objective and based on facts<br>▪ demand is planned and known ahead<br>▪ decisions are formalized and process oriented<br>▪ pre-dominantly group decisions |
| Customer relationship | ▪ short-term relationships to providers<br>▪ low switching barriers between providers<br>▪ customers are often not known to the provider<br>▪ customer competence is of minor importance for products | ▪ long-term relationships to providers<br>▪ switching suppliers bears higher risks<br>▪ customers are well known to providers<br>▪ customer competence is important and spread among multiple persons |

Backhaus and Voeth conclude that organizational buying behavior is characterized by a multi-personnel problem-solving and decision process with an active information-seeking behavior and frequent interactions (Backhaus & Voeth 2011, p.37). The individual buying process however differs and depends on multiple parameters and conditions. Figure 11 provides an overview of influencing factors regarding individual buying behavior.

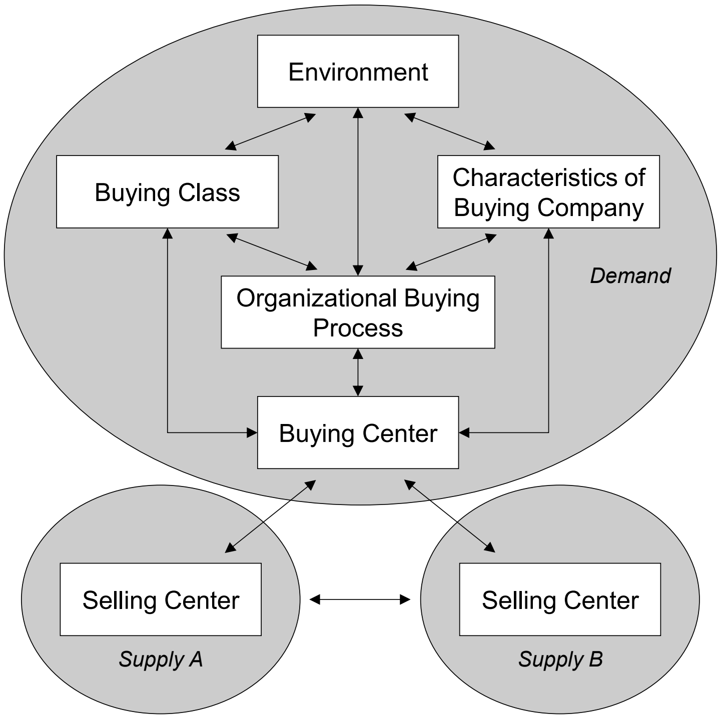


**Figure 11:** Influential factors on organizational buying behavior according to (Backhaus & Voeth 2011, p.38)

The buying class defines the type of the investment and the reason. Robinson et al. first investigated this aspect and introduced the "buyclass[1] / buygrid" framework to structure organizational buying situations (Robinson et al. 1967, p.25). It is based on three dimensions: the newness of the problem, information requirements, and consideration of alternatives, and proposes three distinct buying classes (i.e., patterns of purchase behavior), the new task, the modified rebuy, and the straight rebuy.

Webster and Wind mainly view organizational buying as a decision-making process (short: buying process) and propose a phase model to structure the buying process in five phases: identification of need, establishment of specifications, identification of alternatives, evaluation of alternatives, and selection of suppliers. Backhaus and Voeth provide an overview of the numerous other phase models of the organizational buying process in industrial marketing literature (Backhaus & Voeth 2011, p.43).

The buying process is carried out by the so-called buying center. It is the virtual set of all the people involved in the buying decision and varies from a few people to dozens depending on the size and scope of the targeted investment and organizational factors such as structure and size of the company (Foscht & Swoboda 2007, p.250). There is, furthermore, a positive correlation between buying center size and length of buying process (Backhaus & Voeth 2011, p.37). Webster and Wind defined five distinct buying center roles. These roles have different

[1] "buyclass" and "buying class" are used synonymously.

tasks and authorities in the decision-making process (Webster & Wind 1972). Though in practice, one person can represent multiple roles or many persons can carry one role (Johnston & Bonoma 1981).

The *user* is the person who should use the targeted product and often initiates the acquisition. Moreover, he often has deep product and context knowledge and significantly determines the acceptance of the new investment, which gives him a key role in the buying process. The *buyer* has the task of preparing and concluding the purchase agreement as well as negotiating special conditions and discounts. Typically, this role is conducted by employees from the purchasing department. The *influencer* is formally not involved in the buying process, but can decisively, directly or indirectly, influence the purchasing decisions, e.g., by providing relevant information or minimum requirements, in that he is restricting or extending the list of alternatives. This role is performed by very different people in the organization, such as IT-experts, financial advisors, or even external consultants. The *decider* has the authority to choose from different buying alternatives. For significant strategic investments CxOs are often the ultimate deciders, whereas small recurring investments are decided upon by representatives from the purchasing department. Last, the *gatekeeper* is a group of people who control the information flow into and within the buying center, in that they indirectly influence the buying decision by either forwarding selective information to people or by restricting the spread of certain details. Gatekeeper can be assistants to other roles in the buying center or even company externals who pursue their own interests (Webster & Wind 1972). Kotler adds the roles of *initiator* and *approver* to the initial five proposed by Webster and Wind (Kotler 2003, p.221). More recent works investigate the change of structures within the buying center in different buying situations and in different phases of the buying process (Järvi & Munnukka 2009; Lewin & Donthu 2005).

Since many roles within buying centers are only informal and the buying center size, structure and actual setting changes from buying situation to buying situation it is extremely challenging for e-commerce-based sales channels to deal with buying centers, e.g., to implement differentiated functions targeting the needs for different buying center roles or just to identify the members of a buying center (Winkelmann 2008, p.363).

Many studies of organizational buying behavior have researched the variables influencing the buying decision process described here and have provided integrated models to explain the behavior. In their model, Webster and Wind identified four major variables: individual, social, organizational, and environmental (Webster & Wind 1972). Sheth proposed an integrated model for an industrial buying behavior and modelled the buying process, buying center, product characteristics, company characteristics, situational factors, and their interrelationships. His particular contribution in this context is the aspect of differentiating autonomous versus joint decisions and in modeling the conflict resolution process in joint decisions (Sheth 1973; Sheth 1996). A comprehensive literature review by Johnston and Lewin on concepts in organizational buying behavior and influencing factors on the buying decision revealed that the most researched constructs influencing the buying decision are organization, group, participants, process, seller, and information (Johnston & Lewin 1996).

The frameworks of organizational buying behavior have guided the thinking and structure while analyzing the demand-side value chain of the app store model in this thesis. The concepts of buying situations, buying process, and buying center have been used in particular in Papers 3, 4, and 5 (cf. Part III: Demand-Side Value Chain).

### 2.4.2 Organizational Buying of Enterprise Application Software

Only a few researchers have applied the generic theory of organizational buying behavior to the domain of enterprise software. Using the organizational buying framework from Webster and Wind (Webster & Wind 1972), Verville and Halington qualitatively researched the purchase of ERP systems using four case studies. In an initial publication they focused on factors influencing the buying decision using the categories of environmental, organizational, interpersonal/social, and individual and researched factors relevant to buying ERP solutions within these categories. Major findings included the importance of users' opinions about the purchase as well as the project management skills applied during the acquisition process (Halingten & Verville 2002). In a second publication, Verville and Halington proposed a six-stage model of the buying process of ERP systems, as shown in Figure 12 (Verville & Halingten 2003).

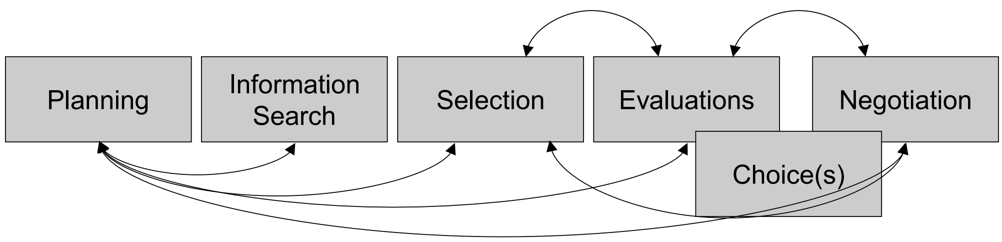


**Figure 12:** Six-stage process model of buying ERP solutions by (Verville & Halingten 2003)

The activities conducted and their sequence varies in reality from company to company, and from buying situation to buying situation. Therefore, the constructs identified by Verville and Halington should instead be understood as process cluster or phases within the overall buying process instead of single activities. The lines between the stages in Figure 12 indicate the iterative nature of the process. Among those persons involved in the buying process, Verville and Halington differentiate between the acquisition team and authorities external to the acquisition team. The process phases planning, information search, selection, evaluation and negotiation are conducted iteratively by the acquisition team, whereas the choice phase includes the external authorities. The information search phase supplies information to all the other phases (Verville & Halingten 2003).

Loebecke et al. examine the factors influencing the purchase decision of enterprise software (they use the term “Business-to-Business (B2B) software”). Loebbecke et al. identify information-related drivers and feature-related drivers as major impact factors on the buying decision. In detail these drivers are (Loebbecke et al. 2010):

- information-related drivers: customer references, expert recommendations, demonstration team presentations;
- feature-related drivers: price performance, functionality, sales team service.

Last, Palanisamy et al. in their empirical study confirm the influence of mainly five factors on the buying decision process (Palanisamy et al. 2010):

- enterprise systems strategy and performance
- business process re-engineering and adaptability
- management commitment and users’ buy-in
- single vendor integrated solution
- consultants, team-location, and vendor’s financing.

The six-stage model of the buying process ERP solutions served as the basis for the buying process investigations in this thesis. Verville and Halington developed the model using case studies from 1995-1997 (Verville & Halingten 2003), and at that time personal direct sales channels were the dominant form of sales channels. In order to investigate the buying process using an online sales channel such as the app store for EAS, the individual phases had to be slightly adjusted. The buying process in the context of app stores for EAS is mainly investigated in Papers 3, 4, and 5 (see Chapter III.1, III.2, III.3)

The various factors influencing the actual decision for or against an EAS product help us to understand what information needs to be provided throughout the buying process and where the levers are. However, in this thesis it was of great importance to investigate the factors, i.e., drivers and barriers, influencing the adoption of an online channel (app store) to purchase EAS. Though both research topics are very closely related, the factors identified, e.g., by Loebbecke or Palanisamy et al., were not directly used in the models developed (Loebbecke et al. 2010; Palanisamy et al. 2010). However, they provided a good groundwork and means of understanding the impacts of the buying situation.

### 2.4.3 Organizational Selling of Enterprise Application Software

While the previous two sections dealt with organizational buying, this section will introduce concepts and strategies on the sales and distribution side. According to Voeth and Backhaus, sales and distribution strategies deal with the supply of downstream companies in the value chain with goods or services and include distribution logistic activities as well as acquisition activities (Buxmann et al. 2011, p.87). In the case of EAS as a digital good, distribution logistic related activities are of minor importance and will be discussed as part of e-commerce technologies in Chapter 2.6. For the purpose of this thesis, two main aspects should be introduced with regards to acquisition activities: design of the sales system and design of sales activities. Though the concepts presented in this section are mainly generally accepted I will present them with the EAS domain in mind.

**Design of the Sales System**

The design of a sales system mainly deals with decisions on the sales organization and sales channels. The sales organization, especially for large software vendors needs to be structured in order to tailor the sales and acquisition activities to the needs of specific customer segments. The following criteria are mostly used for this purpose (Buxmann et al. 2011, pp.87–89):

- regions (e.g., continents, set of countries, countries, states)
- industries (e.g., professional services, automotive)
- products (e.g., SCM, CRM, ERP)
- existing and new customers
- company size (with regards to employees and/or revenue).

Regional sales units have the advantage of customer proximity, whereas industry sales units are beneficial if industry knowledge is required. Product-oriented sales organizations are mainly used if the software portfolio is rather broad and deep functional expertise is needed during sales activities. The division between existing and new customers is made due to the different nature of personal sales styles when acquiring new customers (“hunting”) or serving existing customers (“farming”) (Buxmann et al. 2011, p.88). Last, the company size is important since it often determines the average deal size and the overall effort required to interact with the customer (Homburg et al. 2012, p.96).

Another important subject of a sales system is the design of the sales channel. There are two major dimensions determining the sales channel. The first determines whether the software vendor interacts *directly* with the software customer or *indirectly* via a so-called channel partner. There are different forms of channel partners depending on their role and contribution. The objectives of establishing an indirect sales channel via partners are to increase proximity from a customer perspective, customer loyalty and to cover a larger market share. Another target is to improve sales cost structures if the expected volume per customer is low. Since EAS is typically complex, software companies establish dedicated programs to train and certify their channel partners (Buxmann et al. 2011, p.89; Kittlaus & Clough 2009, pp.109–110). The downside is the software vendor loses the direct customer contact and might not receive the feedback he needs to react to changing demands in the market.

The other dimension is the sales form. Three forms are mainly distinguished from personal, remote personal, and impersonal sales (Winkelmann 2008, pp.36–39). In the software business these three generic forms are usually implemented through three concrete sales channels (Kittlaus & Clough 2009, pp.104–108):

- *Field sales* is a personal sales channel, mainly used for large customers with high sales volume. A typical role for field sales staff is the key account manager, who ensures frictionless implementation of purchased licenses and products. He further supports the customer in licensing management and maintains contacts and major relationships. The key account manager further tries to "up- and cross-sell" complementary products.
- *Telesales* is a remote personal sales channel. It is performed by a sales staff typically via phone or tele conference solutions (incl. screen sharing). It is often used for smaller accounts, or if many customers need to be reached in a short period of time.
- *Online sales*[1] is an impersonal sales channel leveraging e-commerce technology. At this point online sales are mostly used for consumer products. A major advantage is the possibility of closely intertwining marketing initiatives (e.g, online ads or branding campaigns) with online sales activities (e.g., app stores).

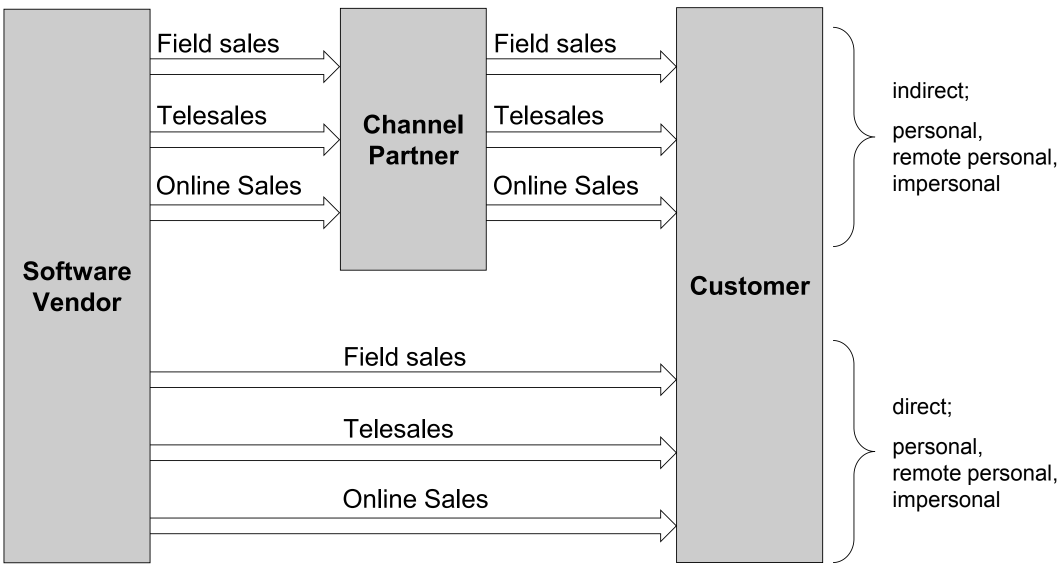


**Figure 13:** Sales channel variants

Kittlaus and Clough state that online sales channels would be less appropriate for complex software products in terms of functional and license structure (Kittlaus & Clough 2009, p.108).

[1] Sometimes the terms electronic sales, digital sales, or internet sales are also used. In this thesis these terms are all used synonymously.

Which software characteristics can make online sales of EAS more appropriate, is discussed as part of Paper 7 (see Chapter V.1)

The combination of direct/indirect sales and the variants of sales form for a software vendor is presented in Figure 13. The sales channel configuration of the indirect channel including channel partners needs to be described using two channel sections. The first section is between the vendor and the partner, and second one between the vendor and the end-customer. In principle both channel sections can have either of the defined sales forms, resulting in nine possible configuration variants.

Balancing opportunities and risks, the software vendor needs to find and define the right channel mix for his product portfolio. He may very well combine a multitude of different channels: e.g., direct field sales channel for the large customers and most complex products (ERP), and indirect field sales via partner for smaller customers or products only targeted at small companies (dedicated ERP for small and midsized companies), and last the online sales channel for complementary simple products, such as additional analytical reports (Buxmann et al. 2011, p.90; Kittlaus & Clough 2009, pp.110–111). Moreover a software vendor may choose to sell a single product via only one channel, a so-called single-channel-system, or via multiple channels in parallel, a multi-channel system (Buxmann et al. 2011, pp.91–92). With multi-channel-systems software vendors try to yield synergies, leveraging the benefits of individual channels while avoiding their downsides. The challenge in the design of a multi-channel system is therefore to allow the selected channels to complement each other while avoiding channel conflicts or cannibalization (Buxmann et al. 2011, p.92; Kittlaus & Clough 2009, p.111).

Further basic concepts for multi-channel sales systems including an online channel are discussed in the next chapter in 2.6.2 and are also the focus of Paper 5 (see Chapter III.3) where I will investigate how an online channel and a personal field sales channel can be combined without cannibalization.

**Design of Sales Activities**

The sales activities can be structured using a selling process. The selling process should ideally mirror the organizational buying process from the vendor perspective. Buxmann et al. present a comprehensive reference process including multiple phases (Buxmann et al. 2011, pp.98–103):

- direct marketing / or quote request
- lead creation, feasibility
- prepare/make initial presentation
- prepare/present solution proposal

- prepare/conduct analysis workshop
- prepare/make solution presentation
- quote creation and presentation
- negotiation
- internal handover
- customer kick-off.

The process as presented by Buxmann et al. is already very specific and tailored to a personal sales channel. Homburg et al. define a more general sales process using five phases: customer acquisition, bid preparation, order processing, service delivery, continued support (Homburg et al. 2012).

Most process models for selling enterprise software cited in the literature are based on a personal direct sales model, and therefore they were not used to analyze the demand-side value chain in this thesis. Instead, generic process phases were used to research the buying behavior and the functionality an app store needs to provide along the buying process (cf. Paper 3, 4, 5, 6).

Equivalent to the buying center, the software vendor forms a sales team, the so-called selling center. It includes all those persons from the selling firm involved in the selling/buying process of the respective customer (Puri 1992; Backhaus & Voeth 2011, p.104 ff.). Typically, there is a (key) account manager responsible for the sales opportunity. He is responsible for the relationships and interactions to the customer and coordinates the selling center. In addition, experts from different functional departments of the vendor are involved. Examples are representatives from the product, service, and support department respectively (Puri 1992). These experts are often called “pre-sales” (Kittlaus & Clough 2009, p.108). Aside from content expertise, the selling center should also be composed by considering the customer hierarchies involved on the buying side, so that communication and negotiations can be done at eye level. The sales tactic of explicitly considering hierarchies is referred to as multi-level selling (Kleinaltenkamp & Saab 2009, p.130).

Another method to structure and tailor sales activities is the hunter/farmer model. Hunters typically recruit new customers, whereas farmers, retain customers and try to up- and cross-sell products and increase the vendor’s share of wallet at the customer organization (Homburg et al. 2012, p.99). Therefore, the hunter/farmer model can either be used to structure the sales organization or to define specific sales activities.

Another method of defining appropriate sales activities is to analyze the nature of the sales opportunity, i.e., the nature of the customers buying situation. In order to succeed, the sales activities need to fit the customer’s buying situation. For this purpose, Rackham and

DeVincentis contrast transactional versus consultative sales. The transactional sales model is focused on closing the deal with a short time horizon, whereas the focus of the consultative sales model focuses on solving the problems of the customer, consulting him throughout the buying process, and typically has a longer time horizon (Rackham & DeVincentis 1999, p.79).

Wenzel et al. used the concept of transactional and consultative sales models and combined it with the buy classes from Robinson (Robinson et al. 1967) and the buying center from Webster and Wind (Webster & Wind 1972) to create a framework for selling and buying situations for EAS (cf. Figure 14).

**Transactional Sales Model** ⟷ **Consultative Sales Model**

| | | | | |
|---|---|---|---|---|
| **What** | **Price** | Low | Medium | High |
| | **Risk** | Low | Medium | High |
| | **Specificity** | Generic | … | Specific |
| | **Complexity** | Simple | … | Complex |
| **Who** | **User** | Single User | Department | Corporate |
| | **Buying Authority** | Individual | Local Buying Center | Corporate Buying Center |
| | **Level of Expertise** | Low | Medium | High |
| **How** | **Buying Class** | Straight Re-Buy | Modified Re-Buy | Initial Purchase |
| | **Ability to Evaluate** | Specification | Experience | Trust |
| | **Implementation Type** | Instant Usage | Activities / Workflow | Project |
| | **Deployment** | On Demand | On Device | On Premise |

Low **Transaction Cost** High

**Figure 14:** Classification of selling / buying situations for EAS (Wenzel et al. 2012)

They argue that a customer's buying situation is defined by "what" is being bought, "who" is conducting the purchase, and "how" it is being bought. The conclusion of their analysis is that traditional EAS is better suited for consultative sales, however online sales channels (such as electronic marketplaces or app stores) should instead support the transactional sales model, since they mainly support standardized and limited interaction patterns. To cope with this contradiction and to still make EAS more suitable for online sales, they propose three strategies (Wenzel et al. 2012):

- "reduction of the complexity of the customer's buying situation;
- new e-commerce technologies for consultative sales patterns;
- integration of electronic marketplaces into multi-channel sales systems."

Since their framework lacks empirical evidence, it should be understood as a hypothetical proposal. In this thesis, these three proposed strategies have partly motivated specific, individual papers. In Paper 3 and 4 (cf. Chapter III.1, III.2) the buying situation will be qualitatively evaluated in more detail to identify the barriers to and drivers of adopting an online channel for buying EAS. These analyses include the evaluation of the buying process, the EAS itself, but also new e-commerce technologies acting as drivers for adoption. Paper 7 (cf. Chapter V.1) analyzes EAS and derives recommendations to reduce complexity and barriers inherent with the software product itself.

Paper 6 (see Chapter IV.1) analyses app stores for EAS and their e-commerce capabilities in detail and how they cope with typical issues while buying EAS online.

Last, Paper 5 (cf. Chapter III.3) investigates the combination of an online sales channel for EAS with a personal direct sales channel into a multi-channel sales system and derives concrete design recommendations.

## 2.5 Information Management

From a software customer perspective, app stores for EAS are an e-commerce system to purchase software applications, to distribute them within the organization, or to download and consume applications, and to manage them to a certain extent. Depending on the definition of the various concepts and terms these tasks are part of corporate information management, IT governance, or IT management. In Chapter 2.5.1, I will present selected definitions and frameworks from this topic area. Thereafter in Chapter 2.5.2, research describing the current phenomenon of IT consumerization and shadow IT and their effects on the corporate IS are elicited. New technologies and applications supporting the management of consumer technologies in the enterprise are then reviewed in Chapter 2.5.3, including mobile device management (MDM), mobile application management (MAM), and internal enterprise app stores.

### 2.5.1 Information Management, IT Governance, and IT Management

According to Ferstl and Sinz information management is defined by conducting the tasks designing and governing the corporate IS, and operating the application systems and IT infrastructure. The holistic design and governance of the corporate IS includes both, the definition of the IS tasks and business processes, and the definition of the assigned resources, i.e., application systems, IT infrastructure and personnel. The responsibility for the automated part of the IS is thereafter described by the term IT governance (Ferstl & Sinz 2013, pp.445–464). The automated part of the IS includes all application systems and IT infrastructure, whereas the non-automated part of the IS includes the personnel working with the “object” information in a company (Ferstl & Sinz 2013, pp.6–7).

According to Krcmar, information management has the target to best utilize the resource information in accordance with the company targets (Krcmar 2015, p.10) and is defined by his model of information management (cf. Figure 15).

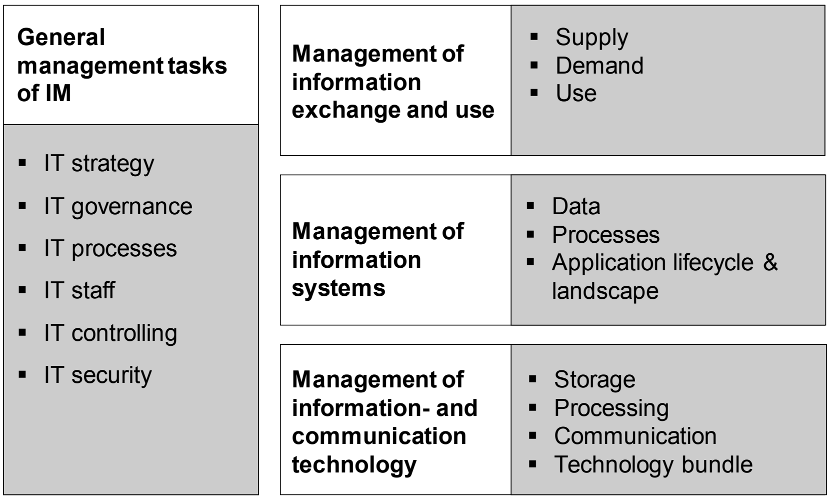


**Figure 15:** Model of information management according to (Krcmar 2015, p.10)

Which software applications are purchased is according to the model by Krcmar part of the management of information systems. However, general acquisition policies on IT investments and purchasing process standards are defined as part of the general management tasks of IM.

Similarly to Ferstl and Sinz, Krcmar defines IM as a very holistic concept and the tasks as part of the company's general management (Ferstl & Sinz 2013, p.446; Krcmar 2015, p.10). The structure of IM tasks, their assignment to resources, and the principles guiding their execution are defined by the IT governance in Krcmar's model (Krcmar 2015, p.11).

Meyer et al. see IT governance as a partially overlapping but also as a new and complementary concept to IM. They argue that IM concepts are based on a traditional role model of a corporate IT department providing IT infrastructure, services, and applications to other parts of the organization. Therefore, IM focuses on the management of planning, development, and operation of IS. In contrast, IT governance focuses more on the processes of sourcing, delivery, support, monitoring, and control of IT (Meyer et al. 2003).

COBIT is a comprehensive, best-practice framework for the governance and management of IT (available in the 5$^{th}$ edition (ISACA 2012)). The current framework is based on five core principles (ISACA 2012, pp.13–14): Meeting stakeholder needs (principle 1), covering the enterprise end-to-end (principle 2), applying a single integrated framework (principle 3), enabling a holistic approach (principle 4), separating governance from management (principle 5). De Heas et al. have extensively analyzed the framework and emphasize the significance of value creation and risk management through the use of IT and the importance of strategic business and IT alignment (cf. principle 1) (De Haes et al. 2013). Moreover, COBIT distinguishes between governance and management of IT (cf. principle 5). IT governance defines processes to evaluate stakeholder needs, to direct by giving priorities and making decisions, and to monitor performance, compliance and progress. Orthogonal to these governance processes, COBIT specifies processes for IT management. Four groups of IT management process are defined (ISACA 2012, pp.32–33):

- “align, plan, organize
- build, acquire, implement
- deliver, service, support
- monitor, evaluate, assess.”

IT governance according to Weill is defined “[…] as specifying the framework for decision rights and accountabilities to encourage desirable behavior in the use of IT” and he goes on to say “[…] IT Governance is not about what specific decisions are made. That is management. Rather, governance is about systematically determining who makes each type of decision (a decision right), who has input to a decision (an input right), and how these people (or groups) are held accountable for their role” (Weill 2004). He defines five distinct areas of IT decision needs:

- *IT principles:* how to use IT in the company;
- *IT architecture:* includes technology choices, policies & rules of how to use IT, and roadmaps of future IT enabled business;
- *IT infrastructure strategies:* includes strategies for fundamental IT infrastructure and shared IT base services, such as help desk or network services;
- *business application needs:* specifies the business need of purchased or internally developed application systems;
- *IT investment and prioritization:* decisions about IT budget size and investment approvals.

Weill proposes six governance archetypes companies can choose from to make the major decisions from the five aforementioned areas. The archetypes are defined by which group has either the decision rights or input rights to make a decision. Weill considers three groups to define the archetypes: chief-level executives (CxO), corporate or business unit IT, and business unit leaders or process owner (cf. Table 6). For example, the archetype “IT monarchy” describes a situation where the major decisions are made by executives of the corporate IT or business unit IT department.

**Table 6:** IT governance archetypes by Weill (Weill 2004)

| Archetypes / decision or input is made by: | CxO Level Executives | Corporate IT or Business IT | Business Unit Leader or Process Owner |
|---|---|---|---|
| Business Monarchy | ✓ | | |
| IT Monarchy | | ✓ | |
| Feudal | | | ✓ |
| Federal | ✓ | ✓ | ✓ |
| | ✓ | | ✓ |
| IT Duopoly | ✓ | ✓ | |
| | | ✓ | ✓ |
| Anarchy | | | |

It is not the intention to provide a further general definition of IM, IT governance, or IT management in this work, but only to make use of selected aspects of the presented concepts relevant to the app store model. Therefore, IM is understood as a superordinate concept defining the tasks and resources of a corporate IS in accordance with the corporate goals.

Meyer et al. (Meyer et al. 2003) and the COBIT framework (ISACA 2012) introduced the process aspect of IT governance or IT management respectively. In particular, the aspects of sourcing/acquiring, delivery, monitoring, and control are of interest in the context of app stores being used by software customers to purchase EAS. Moreover, companies may use app stores as a means to implement IT governance policies and rules.

The interesting aspect about Weills work for this thesis is that he introduces business versus IT decision makers (versus top management), and also in part the aspect of central versus decentral decision-making authority. Furthermore, he introduces the decision area of “business application needs” which includes the decisions of selecting the relevant applications.

The aspects of IT governance (or IT management) will be used in Paper 6 (cf. Chapter IV.1) to study how app stores can help to implement certain governance strategies, e.g., certain decision-making patterns including business stakeholders and IT representatives.

### 2.5.2 IT Consumerization and Shadow IT

The broad use of consumer IT in everyday life has also changed how business users perceive their IT in the work context. Employees are today much more knowledgeable with respect to IT use, and demand the same user experience, such as simple try and buy processes, ease of use, and appealing design from their work IT, as they do from their private apps and devices (Price et al. 2012; Weiß & Leimeister 2012b). The trend of new technologies first finding broad adoption in consumer markets and the subsequent diffusion or even uncontrolled infiltration into companies is referred to as “IT consumerization” (Weiß & Leimeister 2012b). In particular, cloud services (e.g., SaaS) and mobile devices and apps find their way into the workplaces of today's employees (Ingalsbe et al. 2011).

A survey from 2011 among 4017 employees by Harris et al. documents the scale of this phenomenon:

- 29% stated they would sometimes, and 23% often or very often, use personal IT (software or hardware) for work-related activities.
- 36% agree or strongly agree with the statement that they do not worry about their organization's IT policies and just use the technologies they need for the work.
- 45% find personal hardware devices and software applications more useful than the ones provided by the company.

It is ultimately the dissatisfaction of employees with technology and applications provided by corporate IT which fuels the uncontrolled infusion of technologies into the workplace without permission (Weiß & Leimeister 2012b). Among the positive effects of IT consumerization, Harris et al. have identified an increased innovativeness of the organization, a higher employee productivity, and higher employee satisfaction (Harris et al. 2012). Niehaves et al. even derive a theoretical model using qualitative research methods to show the potential of IT consumerization on increasing employees' individual work performance (Niehaves et al. 2013).

Uncontrolled IT consumerization, however, leads to shadow IT (Jones et al. 2004; Berbner & Bechtold 2010, p.261) and presents severe risks to corporate information management. In this context Harris et al. mention data security, reliability, performance, dependability, connectivity, and availability. Weiß & Leimeister holistically examine the effects of IT consumerization using the four categories in the Information Management Model from Krcmar (Krcmar 2010) as summarized in Table 7.

**Table 7:** Impact of IT consumerization on tasks of information management (based on Weiß & Leimeister 2012)

| # | Category | Impacts |
|---|---|---|
| 1 | Management of Information Exchange | ▪ new information source with easier access<br>▪ new, so far uncovered information channels<br>▪ increased information demand<br>▪ increased amount of information to be managed<br>▪ challenge to assure information quality and reduce amount of information |
| 2 | Management of Information Systems | ▪ temporary network absence of consumer devices<br>▪ need to adjust business processes<br>▪ shorter application lifecycles<br>▪ beta applications<br>▪ licensing issues (privately acquired licenses)<br>▪ increased technology heterogeneity<br>▪ disintegration of classic system architectures |
| 3 | Management of Information and Communication Technology | ▪ new devices (i.e., mobile) and operating systems (e.g., iOS, Android) as integral part of system landscape<br>▪ growing data storage demand<br>▪ growing demand in network bandwidth (mobile networks, WLAN)<br>▪ decreased predictability of upcoming technologies due to short innovation cycles<br>▪ less stable and guaranteed support and maintenance periods<br>▪ risks of data compliance and data protection require new technical solutions and IT policies |
| 4 | Managerial Tasks for Information Management | ▪ new software and hardware procurement processes (decentral procurement by business users)<br>▪ bring-your-own-device (BYOD) initiatives<br>▪ support of IT via employee-self-service<br>▪ corporate IT increasingly defined bottom-up (driven by employees)<br>▪ corporate IT needs to flexibly integrate new applications and technology on short notice (ad-hoc)<br>▪ partially replacing proactive management with reactive capabilities (IT governance)<br>▪ limited control w/r to IT security demands new policies and technologies |

In order to avoid the downsides while exploiting the potentials, IT consumerization should be actively managed. Harris et al. introduce six management strategies to deal with IT consumerization, which are illustrated in Figure 16. The Laissez-Faire strategy in principle does not restrict the use of external IT, whereas the Authoritarian strategy does not allow consumer technology beyond normal IT acquisition policies. The so-called middle ground strategies, however, try to find a balance between controlling IT and providing freedom of choice in order to harness the benefits of new technologies (Harris et al. 2012).

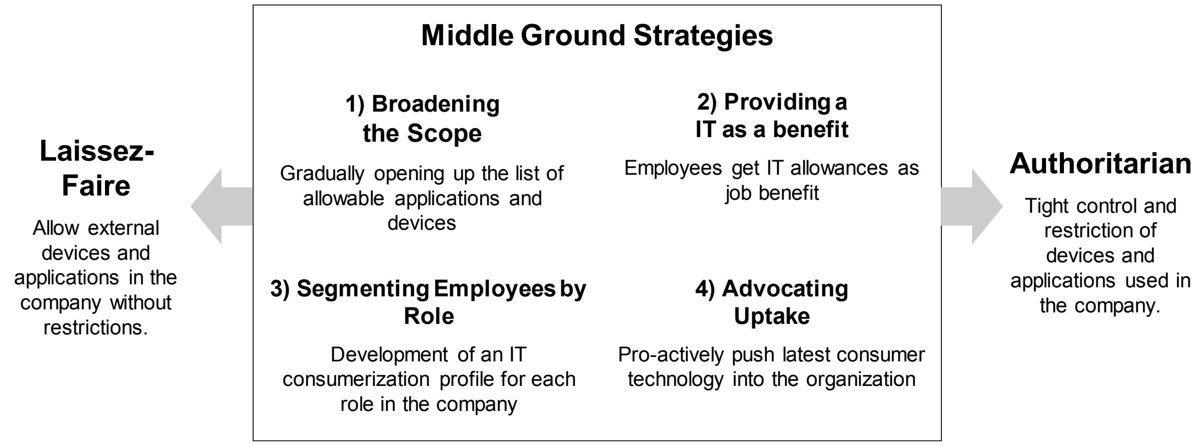


**Figure 16:** Management Strategies for IT Consumerization according to (Harris et al. 2012)

IT Consumerization is an important related research subject, since app stores for EAS are a technology to provide new ways of searching, evaluating, buying, distributing, and managing applications to the organization. In that they are a means of counteracting the risks of IT consumerization and shadow IT, as Beimborn and Palitza argue (Beimborn & Palitza 2013).

### 2.5.3 Enterprise Mobility Management

Today's mobile devices, such as smartphones and tablets were originally designed with the consumer in mind. Naturally, at the time when companies started to show interest in this technology the devices and respective operating systems had hardly any features required by corporate IT departments (Weiß & Leimeister 2012a). Moreover, companies started to implement programs to allow co-use of mobile devices for private and professional purposes. Using private devices in the work-context is called bring-your-own-device (BYOD) and is an alternative to company-owned devices (French et al. 2014). To ensure employees adhere to IT policies and security rules, companies demand solutions for enterprise mobility management (Ortbach et al. 2014).

Mobile device management (MDM) describes a central application enabling the management of mobile devices in the company, and MDM often provides dedicated features to support BYOD programs (Ortbach et al. 2014). It can be used to control and monitor employee devices (Lee et al. 2013). Typical features of MDM applications are presented in Table 8.

**Table 8:** Typical functionality of mobile device management systems (Lee et al. 2013)

| Functional Category | Typical Features |
|---|---|
| Monitor | ▪ GPS data<br>▪ phone state<br>▪ device status<br>▪ network connectivity<br>▪ text, voice, and data usage<br>▪ app usage |
| Control | ▪ disabling hardware features (Bluetooth, GPS, camera etc.)<br>▪ restricting not-approved app installations<br>▪ restricting app permissions (e.g., access to address book)<br>▪ pushing selected apps to devices<br>▪ remotely erase data on device<br>▪ remotely disable device |

Whereas MDM focuses on the management of the entire device with the metaphor of a "fleet management" (Beimborn & Palitza 2013), mobile application management (MAM) systems provide enhanced capabilities to manage the entire lifecycle of mobile applications including development, procurement, distribution, configuration, update and removal of an app (Beimborn & Palitza 2013). According to Crook, MAM complements MDM systems (Crook 2013). Table 9 summarizes key features of a MAM system.

**Table 9:** Typical functionality of mobile application management systems (Crook 2013)

| Functional Category | Typical Features |
|---|---|
| Application life cycle management | ▪ distribute apps by group or policy<br>▪ automatic updates<br>▪ app version management<br>▪ app end-of-life management<br>▪ detailed app analytics |
| Security management and policy control | ▪ application white and black listing<br>▪ data encryption per application<br>▪ enforce, restrict user authentication per app<br>▪ configuration of virtual private networks for individual apps<br>▪ disable or enable data storage, offline access, document sharing<br>▪ erase single applications remotely |

If MAM systems are provided as standalone applications, independent from MDM solutions, they are often referred to as enterprise app stores (Crook 2013; Beimborn & Palitza 2013). To avoid confusion regarding terminology, these app stores are referred to as "internal enterprise app stores" in this thesis. Internal because they are under full control of the respective company in contrast to what I will refer to as "public enterprise app store" where the owner is a platform or software provider. In contrast to MAM, standalone internal enterprise app stores also provide their functionality to non-mobile applications, such as browser applications, desktop applications, or digital content.

Hess et al. and Beimborn and Palitza have analyzed available internal enterprise app stores and identified the following typical characteristics and features (Hess et al. 2012; Beimborn & Palitza 2013):

- user interface imitating popular consumer app stores to browse, search, try or evaluate, download, update, and manage personal app portfolio
- social and communication features (e.g., share applications, review or rate applications)
- support of app procurement processes, including connectivity to public enterprise app stores
- license management of commercial apps
- separation of private and work data (e.g., deployment of secure data container)
- centrally trigger installation of apps
- centrally remove apps from single or all devices
- centrally update apps
- UI to manage app catalog content and to apply corporate identity design (e.g., logo, colors) to the internal enterprise app store
- user access to app catalog via mobile device or desktop.

The benefits of such a system are multi-faceted and are comprehensively analyzed by Beimborn and Palitza (who developed a benefits framework), Crook, and Drake (Beimborn & Palitza 2013; Crook 2013; Drake 2012). These benefits can be summarized as follows:

- support for IT governance and compliance (reduce shadow IT, provide data privacy on employee devices, transparent app license management);
- customized end-to-end app management (in-house app development, app procurement, app distribution, app version management, and app removal – all based on end-user segmentation);
- reduction of cost for app management and BYOD programs (central management for all apps used in the company, reduction of maintenance effort for apps);
- improved user experience by leveraging highly usable interfaces and custom corporate branding for managing business apps;
- providing IT department detailed insights into app usage of employees.

Internal enterprise app stores are part of the demand-side value chain of the app store model (cf. Chapter 1.3) and mainly support company-internal processes. Paper 6 (cf. Chapter IV.1) analyzes an existing internal enterprise app store in detail, and shows how it can be used to implement certain IT governance strategies and be best integrated into the overall demand-side value chain of the app store model.

## 2.6 Electronic Commerce

In this section I will first introduce basic concepts of e-commerce (Chapter 2.6.1) and then classify the App Store Model for EAS along the introduced dimensions. Thereafter dedicated e-commerce subjects and app store concepts will be reviewed in more detail.

### 2.6.1 Fundamental Concepts of Electronic Commerce

E-commerce describes the information and communication technological facilitation of coordinating and transacting goods, products or services exchange, i.e., (trading) between different actors via electronic networks (see also Schoder 2013a; Turban 2008, p.4). E-commerce is differentiated from electronic business (e-business) in that it is defined as a subset of e-business and where the latter includes transactions within the firm (Laudon et al. 2010, p.36f.; Gersch 2015).

**Electronic Markets and Marketplaces**

E-commerce systems also enable the creation of so-called electronic markets and marketplaces. A market in general is defined by meeting demand and supply for a product or service (Domschke & Scholl 2000, p.176; Malone et al. 1987). An electronic market can hence be defined as a market that is facilitated by the means and technologies of e-commerce. Malone et al. predicted that IT will "[…] lead to an overall shift toward proportionately more use of markets - rather than hierarchies - to coordinate economic activity." Mainly due to the fact that IT reduces the transaction cost (i.e., the coordination cost) of matching supply and demand (Malone et al. 1987). An electronic market according to Schmid therefore ideally supports all, and at least the single phases (phases: initiation, contracting, enforcing) of a market transaction between the demand and supply actors (Schmid 1993; Schmid 1997).

Furthermore, the term marketplace originally refers to a physical location where buyers and sellers meet to trade goods or services ("Agora" the marketplace of ancient Greek serves as prototype (Schmid 1997)). Hence, an electronic marketplace is a delimitable and virtual space created by organizational, and information and communication technological means to realize an electronic market where the buyer and sellers, or their electronic agents meet to conduct market transactions (Schmid 1997). Bakos simply defines an electronic marketplace as the "electronic market system" (Bakos 1997). This system consists of business processes, human actors, and the e-commerce application systems used to form an inter-organizational IS connecting buyers and sellers to trade goods or services electronically. Since electronic marketplaces digitize single or all phases of the market transaction (see also Schoder 2013b;

Schmid 1993), they typically enable higher number of participants and new forms of coordination. While fully distributed systems (also in the sense of ownership) exist, the commercially available electronic marketplaces (e.g., Ebay) are mostly systems owned and operated by a delimited organization, an intermediary (Schmid 1993). This organization can then define and enforce policies, standards, and entry barriers to participate in the marketplace (cf. Ferstl 2012, p.76f. & 336).

**Classification of E-Commerce Systems**

E-commerce systems can be categorized using various factors (Laudon et al. 2010, p.575; Schoder 2013a; Wenzel et al. 2010):

- *Different types of actors*: The most common ones are business to consumer (B2C), business to business (B2B). Others variants are business to employee (B2E), consumer to consumer (C2C). Sometimes administration is named as separate actor type and forms the following categories: administration to administration (A2A), administration to business (A2B), administration to consumer (A2C).
- *Transaction-oriented phases*: Initiation (e.g., information gathering, evaluation), contracting (e.g., quote, negotiation), enforcement (e.g., electronic delivery or delivery support, payment), after sales (e.g., learning, support), administration (e.g., contract, user and privileges management).
- *Types of goods*: Physical goods, digital goods, services.
- *The form of intermediation supported*: Match maker, market maker, agent (sell-side), agent (buy-side), or broker.
- *Relationship type*: One-to-one (one supplier, one buyer), one-to-many (one supplier, multiple buyers), many-to-one (multiple-supplier, one buyer), many-to-many (multiple suppliers, multiple buyers).
- *Type of e-commerce application system*: e.g., online catalog, online shop, electronic mall, electronic marketplace.

The morphological box in Figure 17 shows the dimensions introduced to categorize e-commerce systems. The highlighted light-grey boxes indicate the values which may apply to possible instances of the App Store Model for EAS.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Types of actors** | **B2B** | B2C | **B2E** | C2C | C2C | A2A | A2C | A2B |
| **Transaction phases** | **Initiation** | **Contracting** | **Enforcement** | **After Sales** | **Administration** | | | |
| **Types of goods** | Physical goods | **Digital goods** | **Services** | | | | | |
| **Relationship type** | One-to-one | One-to-many | Many-to-one | **Many-to-many** | | | | |
| **Form of intermediation** | **Match maker** | Market maker | Agent (buy-side) | **Agent (sell-side)** | Broker | | | |
| **Type of e-commerce application system** | Online catalog | **Online shop** | Electronic mall | Electronic auction house | **Electronic marketplace** | … | | |

**Figure 17:** E-commerce classification of the App Store Model for EAS (morphological box)

Typically, app stores for EAS are classified as B2B systems where ISVs market applications to software customers. The software customers are software consuming companies. However, there might be policies established to also allow the employee to be the organizational unit of relevance to the app store and the ISVs. In such a scenario the app store for EAS can be classified as a B2E (cf. Huang et al. 2004) system where individual employees have decision and buying authority to equip themselves with productivity applications. Hence, the buying and marketing capabilities in such a case are tailored to individuals (see also Paper 6, IV.1). Whereas for typical B2B situations organizational buying behavior applies, with multiple individuals involved in a buying decision (cf. Chapter 2.4).

E-commerce systems are relevant for the app store model with regards to two areas: The supply-side and the demand-side (see Figure 18). For both the demand-side and the supply-side, app store systems typically support all transaction phases related to trading EAS apps – however the level of automation is typically higher on the demand-side since the number of buyers (software customers) and number of transactions is higher than the number of ISVs and transactions on the supply-side. It is important to note that not all processes and application systems relevant to the app store model can be classified as e-commerce system. The provisioning of the platform service to ISVs as well as processes within the platform provider and the application systems supporting these processes are part of the app store model, however they go beyond the scope of typical e-commerce systems and should hence be classified as “e-business”.

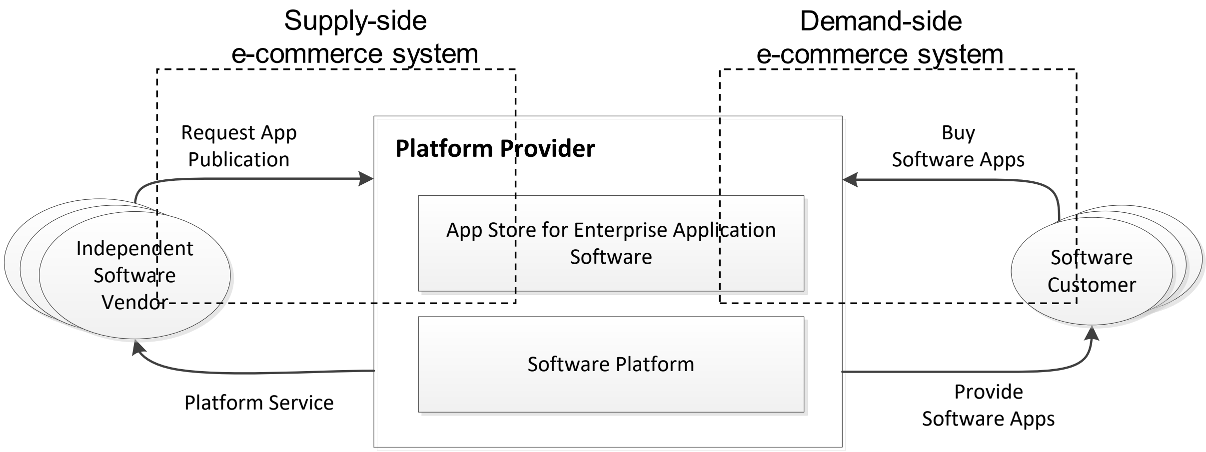


**Figure 18:** E-commerce systems involved in the App Store Model for EAS

The goods traded on app stores are software applications which are considered digital goods. Often software applications provide digital access to services and the borders between products or goods and services blurs (see also “electronic business services” in Chapter 2.2.5) or the software applications are provisioned as a cloud service (see Chapter 2.2.4). Furthermore, app stores for EAS may also market electronic or traditional IT services complementing software applications (e.g., training, implementation, customization, support). Hence, the type of goods also includes the services category.

App stores for EAS connect multiple ISVs with multiple software customers (“many-to-many”). The intermediation pattern used in the App Store Model for EAS mostly classifies as sell-side agent model. In a sell-side agent model the platform provider receives a mandate to “negotiate” (i.e., mandate to sell) on behalf of the ISV to a software customer. The ISV pays the platform provider for this service. The legal contract for the sale of a software application is established between the software customer and the ISV, the platform provider has separate frame agreements with the ISV and the software customer regulating the terms of the app store. In rare cases or for app stores with a low level of automation and standardization the pattern match maker may also apply. In such a case the app store would only support the “initiating” phase of a transaction. The other transaction phases would then be conducted directly between ISV and the software customer without the support of the app store and the platform provider. While other forms of intermediation are theoretically viable options (e.g., platform provider as market maker), my market research has shown that the sell-side agent and match maker seem to be the pre-dominant and most favored models for app stores for EAS.

The type of application system implementing the e-commerce scenario can be grouped into the usual categories available in the market. The app store model in general is mostly categorized as an electronic marketplace implementing a two-sided electronic market (Jansen & Bloemendal 2013; Wenzel et al. 2012). The platform provider, however, often not only acts as curator of a marketplace but also owns its own portfolio of EAS products (also referred to as “platform core products” opposed to “complementary products” from ISVs). In such a case the app store system instead also acts as an online shop or online sales channel in a direct relationship with software customers (see also Part III: Demand-Side Value Chain, which focuses mostly on the app store as a sales or buying channel for EAS).

### 2.6.2 Software-as-a-Service and Platform-as-a-Service Marketplaces

With the rise of cloud computing and especially the SaaS and PaaS business models, the IS community has again paid attention to electronic marketplaces (Giessmann 2015, p.25 ff. Price et al. 2012; Giessmann et al. 2015).

The study by Price et al. looks at both the theoretical view on how SaaS marketplaces can best facilitate the market transaction, and how they are used and perceived in practice. From a theoretical perspective SaaS marketplaces reduce transaction cost, and especially the seek and matching cost by (Price et al. 2012):

- “attractive presentation and efficient organization of information;
- assembly of offers that appeal to customers;
- information substitutes to increase customer trust: certification, partners with reputations, contractual guarantees;
- communities that provide ratings and reviews.”

Price et al. further state that the main attempt of SaaS marketplaces would be to connect SaaS providers (= ISVs) with as many software customers as possible. Moreover the SaaS marketplaces would route the software customers on to the SaaS provider directly after matching, or they would also hold the “billing relationship” (Price et al. 2012).

Burkard et al. investigated five SaaS/PaaS marketplaces with regards to supply-side (= vendor-side) characteristics. In particular, they examined the following characteristics by automatically collecting publically available data (Burkard et al. 2011):

- composition of ISVs (number, size, role)
- entry barriers
- stability (history, health)
- pricing
- multi-homing (an ISV offers his solution on more than one marketplace).

Though in this work they can only present preliminary empirical data, some interesting aspects can already be shown. Hence, four of the five examined marketplaces were also used by the marketplace owners to market their in-house applications. And the majority of ISVs only hosted a single application on the marketplace. Furthermore it is noteworthy that they found that a significant number of vendors (70 at the time of their publication) offered their SaaS solution on more than one marketplace ("multi-homing").

However, the vision that SaaS applications can be easily marketed and sold electronically, simply due to the fact that they do not require perpetual upfront licenses and on-premise IT or technology stacks, seemed a bit too optimistic: A multi-case study among different SaaS vendors revealed that electronic sales or the internet only play a minor role in sales and marketing of SaaS solutions (Tyrväinen & Selin 2011). The outcome of this study showed that the major sales channel remains a direct personal sales channel with selective support of the internet limited to marketing communication. Customer acquisition cost in this regard accounts for the largest part followed by customer retention activities (avoiding "churn"). Also the study by Price et al. came to similar conclusions: "Overall, we have discovered discrepancies between the theoretical view of an electronic marketplace (particularly SaaS marketplaces) and the real world. The appeal of such marketplaces in for business is still limited […]" (Price et al. 2012).

While the need for more efficiency in the SaaS area should be given, especially since the average transaction volume of a SaaS deal as well as the lifetime value of SaaS customer is smaller than average in the software industry as another study by Luoma et al. revealed (Luoma et al. 2012). Besides, the study by Giessmann et al. researched the preferences of ISVs with regards to PaaS capabilities, and the majority of respondents highlighted the importance of a marketplace as part of the PaaS offering and 68% even stated that they would expect the marketplace to also handle payment processes (Giessmann et al. 2015).

While first implementations exist on the market, the mentioned studies also showed that the existing SaaS marketplaces cannot live up to the vision. Hence, research could help by not only evaluating the economic, theoretical opportunities, or by empirically recording the current market situation, but also by coming up with detailed solutions and design options on how to make the app store model more viable.

### 2.6.3 Consumer App Stores

The next form of electronic marketplace in the software business appeared with the rise of mobile computing in the consumer segment, the app stores (coined by the name of the Apple App Store for iOS devices). Though academic studies in this area are rare, two well-structured works could be identified which help us achieve a better understanding of the topic. First, Holzer and Ondrus structured the phenomenon at a time when the mobile application market was more heterogeneous and the dominance of Apple and Google was not that distinctive. They also refer to "application portals" instead of app stores. They analyze the mobile application market by using three different actors, the developer, the consumer, and the platform provider. The developer receives development tools from the platform provider and in turn publishes the applications to an application portal. The consumer downloads the application from the application portal and pays in return. The application portal forwards the payment to the developer (Holzer & Ondrus 2011).

Moreover, they analyze different patterns of platform integration using three components: the developer tools, the application portal, and the mobile device. Depending on which components are controlled by the platform provider different integration patterns can be derived (see Figure 19). At the time of their analysis all models had real world use cases – however they already saw a strong tendency towards the full integration model, since the platform provider has more opportunities to balance the two-sided market and subsidize one element with another (e.g., giving away the device for less, but charging more on the developer side) (Holzer & Ondrus 2011).

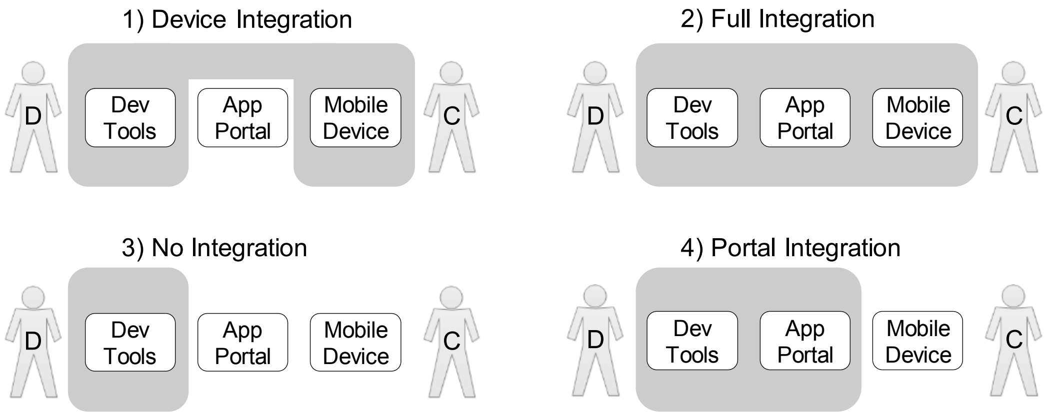


**Figure 19:** Different types of platform integration (Holzer & Ondrus 2011)

Another good overview of the topic of consumer app stores was published by Jansen and Bloemendal. In their work they first define an app store as follows (Jansen & Bloemendal 2013):

> "[An app store is a] online curated marketplace that allows developers to sell and distribute their products to actors within one or more multi-sided software platform ecosystems." The further argue, that "an app store can be seen as a catalyst in […] a software ecosystem. The services it offers are part of the common platform of the ecosystem and it can have a pivotal role in creating the common market. App stores allow developers to monetize their software and bring consumers new functionality. A successful app store is beneficial to the success of a software ecosystem".

In their study they conducted a multi-case analysis using the most prominent consumer app stores (incl. Apple App Store, Google Play Store, Amazon App Store). Based on their empirical findings they proposed a conceptual model for app stores illustrated in Figure 20.

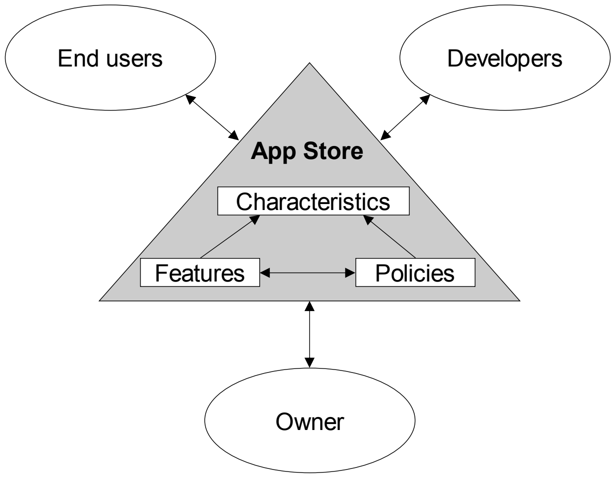


**Figure 20:** Conceptual model of an app store (Jansen & Bloemendal 2013)

The conceptual model consists of three types of actors, the app store owner, the developers, and the end users. Developers can use the app store for publication, end users use the app store to find apps and buy them. The owner builds and governs the app store and receives a revenue share from the developer. The features of the app store refer to capabilities of the software system, the policies are mainly rules and regulations, and the characteristics are aspects of the app store which cannot directly be influence by the app store owner, such as number of apps, number of developers, quality of apps (Jansen & Bloemendal 2013).

The core features they identified studying the consumer app stores are listed in Table 10.

**Table 10:** Core app store features (Jansen & Bloemendal 2013)

| Core Feature | Feature Description |
|---|---|
| app categories | apps are listed in categories and subcategories |
| app listing | apps are listed with full description, images, etc. |
| app lists | apps are listed, e.g. top selling lists or latest additions |
| dev app management | developers can manage their apps in a developer console |
| dev transaction list | developers can manage their transactions |
| distribution integration | distribution and installation happens through platform |
| featured apps | apps can be featured to receive more attention |
| free revenue model | apps can be offered for free |
| paid revenue model | apps can be sold |
| pay out methods | number of pay out methods |
| payment methods | number of payment methods |
| platform comp. filter | apps have information on their platform compatibility |
| ratings | apps can be rated by the user |
| reviews | users can read and write reviews of an app |
| search | users can search for apps using search keywords |

Moreover, they list comprehensive policies identified on the various app stores: e.g., users can only post a review if they have purchased the app.

Comparing the underlying concepts of SaaS marketplaces and consumer app stores, reveals that large similarities exist (i.e., business models, intermediation models, supported market transactions). However, what it is already obvious is that consumer app stores introduce a much higher level of process standardization, and electronic support of the respective market transactions. Whereas SaaS marketplace mainly took over the matchmaking between SaaS provider and software customer (and only optionally other transactions), the consumer app stores per definition also support beyond that the application delivery and the payment processes. The challenge to achieve a similar level of standardization and automation in the business segment as in the consumer segment is one of the motivations of this thesis. The EAS industry differs greatly from consumer software segments – corporate buying procedures, software complexity, or simply the large number of application users – are just some complexity drivers that need to be coped with when designing a similar model in the enterprise world.

**Enterprise App Stores**

Since the rise of mobile devices and mobile apps this technology has also diffused into companies. The concept of enterprise app stores or internal app stores (as I refer to them in this thesis) has already been discussed as part of the previous section on Information Management (cf. Chapter 2.5.3). However, it is important to note that the previously described consumer app stores classify as an electronic marketplace realizing a two-sided market by facilitating business transactions between developers and customers (i.e., consumers). Contrariwise, internal (enterprise) app stores have been built with the purpose of (mobile) application management mainly as an enhancement to MDM. In this thesis the internal app stores will be discussed as an opportunity to "extend" the public app-store-related processes into the software customer organization (see also Chapter 5.4.2 and IV.1 on "internal app store").

### 2.6.4 Multi-Channel Sales System

In previous sections the concept of different sales channels was introduced (see Figure 13). Three sales channel variants have been differentiated, the personal (field sales), the remote personal (tele sales), and the impersonal (online sales) sales channel. According to Heinemann, one of the success factors of e-commerce systems is to establish a "supplement and support channel strategy" (Heinemann 2010, p.90). More precisely the strategy proposes designing a multichannel sales system in such a way that the online channel can fully play to its strengths, while at the same time the multichannel sales system compensates for the online channel's weaknesses (Schögel et al. 2004).

In designing such a multichannel sales system, it must be decided whether to separate the sales channels or integrate them. Between the two extremes of fully interdependent and integrated sales channels on the one hand, and fully separated and autonomous sales channels on the other, a variety of combinations is possible (Cespedes & Corey 1990; Schögel et al. 2004).

In a multichannel sales system with separated channels every channel needs to fully cover all sales tasks. The separation can be achieved, for example, by clearly assigning dedicated products or entire product portfolios to a certain channel. Another option can be to assign specific customer segments to a certain channel (Schögel et al. 2004).

In the case of integrating the sales channels, the sales tasks are fulfilled by the sales system as an integrated unit, i.e., a shared distribution of tasks among the channels (Schögel et al. 2004). A major target of an integrated multichannel system is to avoid channel conflicts such as channel cannibalization and to let the channels best cross-fertilize. For a comprehensive analysis of sales cannibalization in the IT industry the work by Novelli can be recommended (Novelli 2015). One typical instance of an integrated multichannel system is when customers use different channels at different stages of the buying cycle or process since they prefer a degree of personal consultation. Switching channels throughout the sales process is also referred to as channel hopping (Heinemann 2010, p.162; Schögel et al. 2004). Channel hopping can be designed in multiple ways – either through dedicated "exit" points where channel hopping paths are strictly predefined or by switching the channels which is designed more freely.

The challenge in designing a multichannel sales system is to identify the weaknesses of a channel, and in our case specifically the weaknesses of the online sales channel, and to find strategies in coping with these weaknesses. Since EAS is perceived as highly complex and the traditional sales channel is a consultative personal sales channels, multichannel sales systems complementing the online channel is a highly relevant option which should be considered in order to overcome barriers of the app store in the EAS domain. Paper 5, in Chapter III.3, investigates this problem cluster as a central research topic.

## 2.7 Information Technology Adoption

The app store for EAS represents an IT innovation for software customers in that it digitizes the buying and distribution process of EAS. Consequently, innovation management, or more precisely the adoption of IT innovations, is a relevant research cluster. In the case of technological innovations, research mostly refers briefly to technology adoption or technology acceptance (see Venkatesh et al. 2003).

Rogers defines innovation as “an idea, practice, or object that is perceived as new by an individual or other unit of adoption” (Rogers 1983, p.11). The app store in this context can be understood as a combination of practices and objects to be used by an organization (as the unit of adoption) for conducting the buying process for EAS. Rogers further defines adoption as “a decision to make full use of an innovation as the best course of action available” (Rogers 1983). Consequently, theories of technology adoption deal with the decision to use a technology for an intended action or task. They therefore propose models with mostly one dependent variable (e.g., the positive decision to use or the intention to use a technology) and multiple independent variables influencing the dependent variable. Some authors differentiate between the intention to use (“behavioral intention” (Venkatesh et al. 2003)) as the predictor of the ‘actual use of a technology’ as two dependent variables in their models (Venkatesh et al. 2003; Ajzen 1991). Subsequently, in the context of the app store for EAS, adoption means the organization positively decides to make use and then makes actual use of this technology for the buying process of EAS.

### 2.7.1 Overview of Technology Adoption Theories Used in IS Research

Adoption research was primarily studied by psychology or social psychology scholars investigating human behavior. They focused on cognitive phenomena influencing individuals’ intention to perform a certain action (e.g., adopt an innovation). These models have been widely applied to other contexts, including technology and IS research, and they have been adapted and developed further to better suit these contexts. Table 11 gives an overview of ten important and widely applied adoption theories in the IS domain. Eight theories are applicable to individuals whereas two can be applied to the organizational level. The table further summarizes the core constructs used in these models and identifies the originating research scholar.

**Table 11:** Overview of major technology acceptance theories (adapted and extended from Venkatesh et al. 2003)

| Theories of Technology Adoption | Core Constructs (explanation) | Original Research Scholar | Unit of Analysis |
|---|---|---|---|
| Theory of Reasoned Action (TRA) (Fishbein & Ajzen 1975) | ▪ attitude towards behavior (negative or positive feelings about the target behavior)<br>▪ subjective norm (perceived social pressure about the target behavior) | Social Psychology | Individuals |
| Technology Acceptance Model (TAM, derived from TRA) (Davis 1989) | ▪ perceived usefulness (perception about how a system can support a person’s job)<br>▪ perceived ease of use (perception about how effort-free the use of a system would be) | IS | Individuals |

| **Theories of Technology Adoption** | **Core Constructs (explanation)** | **Original Research Scholar** | **Unit of Analysis** |
|---|---|---|---|
| Technology Acceptance Model 2 (derived from TAM) (Venkatesh et al. 2000) | ▪ perceived usefulness<br>▪ perceived ease of use<br>▪ determinants of perceived usefulness (subjective norm, image, job relevance, output quality, result demonstrability)<br>▪ experience<br>▪ voluntariness | IS | Individuals |
| Technology Acceptance Model 3 (derived from TAM) (Venkatesh & Bala 2008) | ▪ perceived usefulness<br>▪ perceived ease of use<br>▪ determinants of perceived usefulness (subjective norm, image, job relevance, output quality, result demonstrability)<br>▪ determinants of perceived ease of use (Computer self-efficacy, perception of external control, computer anxiety, computer playfulness, perceived enjoyment, objective usability)<br>▪ experience<br>▪ voluntariness | IS | Individuals |
| Motivational Model (Davis et al. 1992) | ▪ extrinsic motivation (perform an activity with the perception to gain additional value other than completing the activity)<br>▪ intrinsic motivation (perform an activity without promise of additional value other than completing the activity) | Psychology | Individuals |
| Theory of Planned Behavior (TBA, derived from TRA) (Ajzen 1991) | ▪ attitude toward behavior<br>▪ subjective norm<br>▪ perceived behavioral control (degree to which the activity to be performed is perceived as easy or difficult) | Social Psychology | Individuals |
| Model of PC Utilization (Thompson et al. 1991) | ▪ job-fit (degree to which an individual thinks technology improves job performance)<br>▪ complexity (extent to which an innovation is perceived as difficult to comprehend & use)<br>▪ long-term consequences (future benefits of using the technology)<br>▪ affect towards use (positive or negative feelings or emotions w/r to using a technology)<br>▪ social factors (influence of the social context of an individual)<br>▪ facilitating conditions (objective factors or benefits which drive the adoption of a technology) | IS | Individuals |

| Theories of Technology Adoption | Core Constructs (explanation) | Original Research Scholar | Unit of Analysis |
|---|---|---|---|
| Diffusion of Innovations Theory (DOI) (Rogers 1983) | ▪ relative advantage (extent to which an innovation is better than the existing solution to be replaced)<br>▪ compatibility (the fit to values, past experiences, and needs of adopters)<br>▪ complexity<br>▪ trialability (extent to which an innovation can be tested)<br>▪ observability ("the degree to which the results of an innovation are visible to others") | Sociology | Individuals & organizations |
| Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh et al. 2003) | ▪ performance expectancy ("the degree to which an individual believes that using the system will help him or her to attain gains in job performance")<br>▪ effort expectancy ("degree of ease associated with the use of the system")<br>▪ social influence<br>▪ facilitating conditions<br>▪ gender<br>▪ age<br>▪ experience<br>▪ voluntariness of use | ▪ IS | Individuals |
| Technology-Organization-Environment Framework (TOE) (DePietro et al. 1990; Baker 2012) | ▪ environmental context (incl. industry structure, support by technology service providers, regulatory environment)<br>▪ organizational context ("characteristics and resources of the firm, including linking structures between employees, intra-firm communication processes, firm size, and the amount of slack resources")<br>▪ technology context ("technologies that are already in use at the firm as well as those that are available in the marketplace but not currently in use" ) | ▪ Innovation Mgmt. | Organizations |

### 2.7.2 Individual Technology Adoption Theories

A thorough and comprehensive review of theories of individual adoption was conducted by Venkatesh et al. in their paper "User Acceptance of Information Technology: Towards a Unified View" (Venkatesh et al. 2003). They evaluate eight theories of individual technology adoption and derived their own Unified Theory of Acceptance and Use of Technology (UTAUT) by applying the eight existing theories to a common data set, and combining the findings into a unified view (cf. Figure 21).

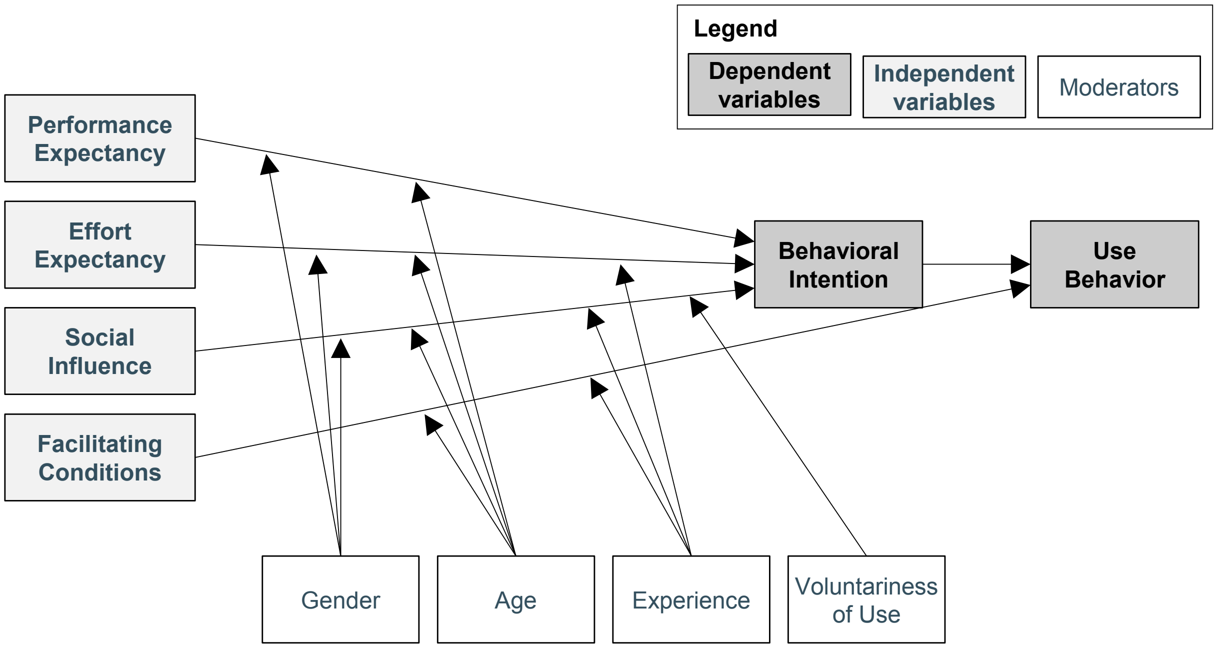


**Figure 21:** UTAUT model (Venkatesh et al. 2003)

The theory consists of two dependent variables. First, the dependent variable 'behavioral intention' is directly influenced by the independent variables 'performance expectancy', 'effort expectancy', and 'social influence'. The second dependent variable 'use behavior' is influenced by the 'behavioral intention' variable and by the fourth independent variable 'facilitating conditions'. The model further includes four moderator variables influencing the prefix and strength of the relationship between the independent variables and the dependent variables: gender, age, experience, and voluntariness of use.

### 2.7.3 Organizational Technology Adoption Theories

The two available theories applicable to organizational adoption have been reviewed by Oliveira and Martins (Oliveira & Martins 2011): the Diffusion of Innovation theory (DOI) by Rogers (Rogers 1983) and the Technology-Organization-Environment framework by DePietro et al. [1] (DePietro et al. 1990).

#### Diffusion of Innovations Theory

The DOI theory is a study of the diffusion of innovations, i.e., "the process by which an innovation is communicated through certain channels over time among the members of a social system" (Rogers 1983, p.4) and tries to "explain and predict rates of innovation adoptions" (Lyytinen & Damsgaard 2001). The "social system" can be individuals, organizations, or a group of individuals (Rogers 1983, p.24). According to the DOI theory, the

[1] The TOE framework is sometimes referenced as Tornatzky and Fleischers' framework (see Baker 2012; Oliveira & Martins 2011). However, they were only the editors of the book "The processes of technological innovation" (1990). The authors of the book section which introduced the TOE framework was written by: Rocco DePietro, Edith Wiarda, and Mitchell Fleischer.

decision of an innovation can be described using a process with five phases: knowledge, persuasion, decision, implementation, and confirmation (Rogers 1983, p.20). The rate of adoption as an important variable in the DOI theory is defined by Rogers as “the relative speed with which an innovation is adopted by members of a social system. It is generally measured as the number of individuals who adopt a new idea in a specified period. So rate of adoption is a numerical indicant of the steepness of the adoption curve for an innovation” (Rogers 1983, p.232). The rate of adoption as an dependent variable is influenced by multiple variables, as shown in Figure 22.

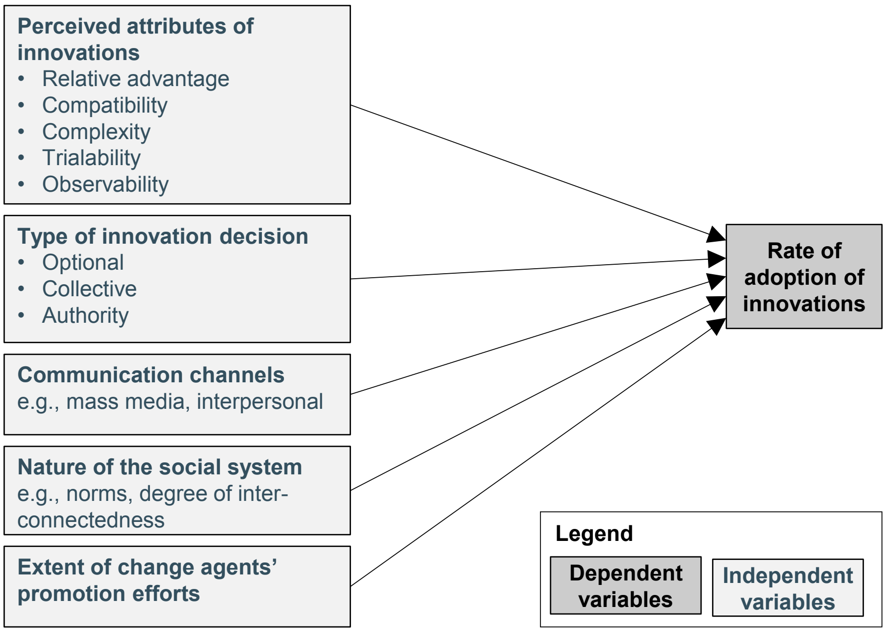


**Figure 22:** Modeling the rate of adoption of innovations (Rogers 1983, p.233)

The ‘type of innovation decision’ can have three values. The optional decision is made by an individual independent of other members of the social system. The other two decision types ‘collective’ and ‘authority’ can be applied to organizational decisions. Collective decisions are made by multiple individuals of a social system (e.g., organization) by finding a “consensus”. Authority decisions are made by relatively few members of a social system (Rogers 1983, p.347). Rogers suggests that the more people involved in a decision to adopt an innovation the slower the adoption. Hence he proposes, as one option to speed up the adoption rate, changing the decision type from collective (all members of the organizational unit) to authoritarian (few members of the unit) (Rogers 1983, p.233).

Furthermore, Rogers introduces five adopter categories individuals can be assigned to based on their innovativeness: innovators, early adopters, early majority, late majority, laggards. The relative time an individual needs to adapt a new technology determines his innovativeness. The distribution of the five adopter categories within the population follows a normal distribution according to Rogers (Rogers 1983, p.247). The categories may also be used for other units of adoption, such as groups or organizations (see Rogers 1983, p.245). As Rogers, however, himself concludes, organizational adoption is much more complicated (Rogers 1983, p.348). To better understand organizational innovativeness and the process of innovation in organizations he introduces two additional elements of the DOI theory on the level of organizations. First, he models organizational innovativeness using the variables depicted in Figure 23. The figure in addition shows the prefix of the relation of the independent variables to the dependent variable (positive “+” or negative “-“).

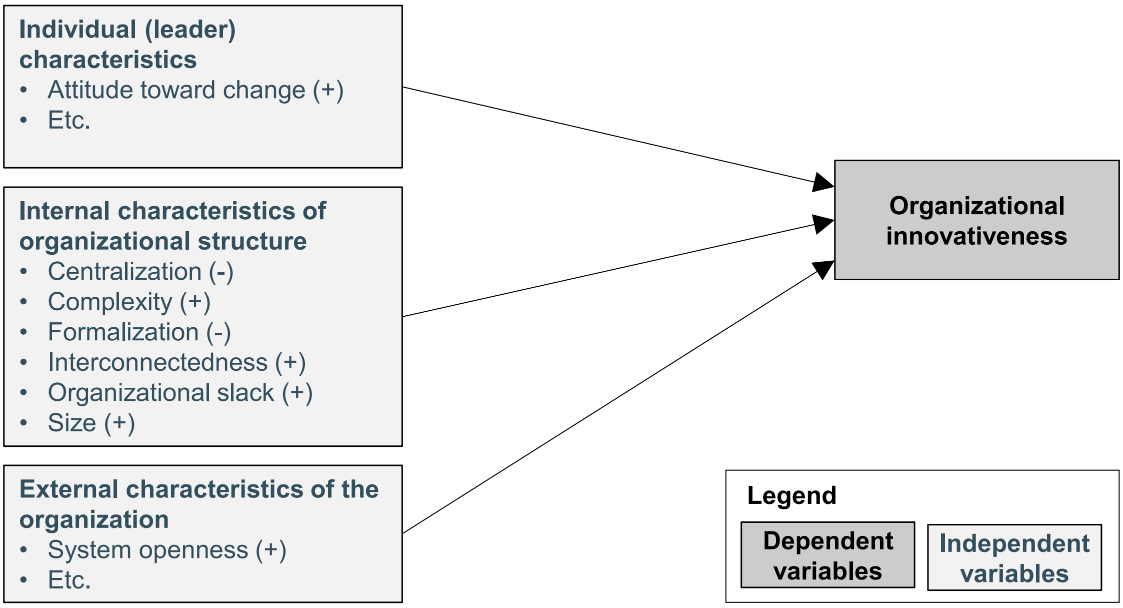


**Figure 23:** Modeling the organizational innovativeness (Rogers 1983, p.360)

Moreover, he enhances his previously introduced five-phased innovation process to reflect the organizational context. Thus, the innovation process in organizations has five sequential stages grouped in two phases. The initiation phase starts with

- agenda setting (identifying needs), followed by
- matching (mapping needs to available innovations), and
- leading to a decision to adopt.

Thereafter the implementation phase follows with the stages

- redefining / restructuring (adjusting either the innovation or the organizational context to fit the specific situation),
- clarifying (innovation is put into regular use in the organization), and
- routinizing (innovation is becoming an integral element of the organization).

**Technology-Organization-Environment Framework**

The TOE framework identifies three key elements influencing the decision to adopt and implement an innovation: the technological context, the organizational context and the environmental context (DePietro et al. 1990). Figure 24 illustrates the three contexts of the TOE framework. The *technological context* includes technologies available within the company and those available at the market but not yet in use at the company. The *organizational context* refers to company resources and its structural characteristics, incl. communication processes, company size, slack resources, and linking structures among employees. Last, the *environmental context* includes government regulations, characteristics of the industry and the respective market structure (e.g., competitors), as well as companies which may offer technology, infrastructure or service support to the adopting company (Baker 2012; Oliveira & Martins 2011; Dedrick & West 2004)

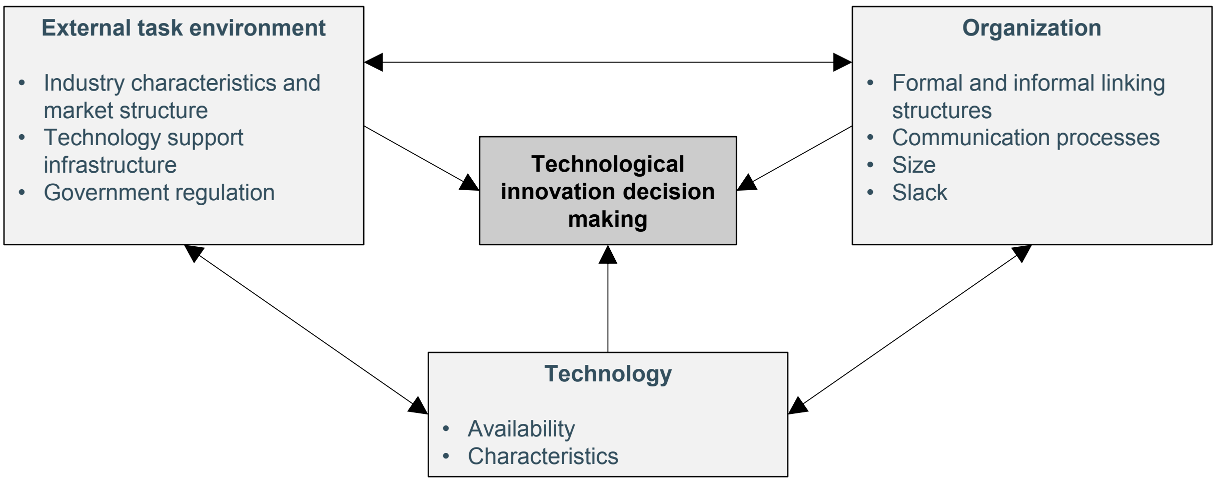


**Figure 24:** The Technology-Organization-Environment framework (DePietro et al. 1990)

### 2.7.4 Assessment of Existing Theories to be used in this Research

While evaluating existing research models to be used in the context of this thesis to try to support our research objectives, i.e., to study the adoption of an online channel for buying EAS and providing design recommendations (e.g., requirements, processes, standards), we have identified several shortcomings which are supported by the literature.

The first shortcoming of the existing acceptance models is that the majority of theories focus on individual acceptance or adoption. The “app store” as an online channel for EAS, however, is relevant to organizations. While organizations themselves are made up of individuals, one could argue that these models can also explain the adoption of an online channel by a buying organization or a virtual buying center. Since the individuals in the buying center of an organization however have different roles with different objectives and who interact with each other, and furthermore because the adoption of the app store for EAS needs to be made by either the entire organization or at least a meaningful subset (e.g., the buying center of a department with full power of attorney), the prediction about individual acceptance could still not be translated easily to the adoption of this innovation on an organizational level. Moreover as Rogers states, “in many cases, an individual cannot adopt a new idea until an organization has previously adopted” (Rogers 1983, p.347), which means that the organizational adoption might even be a hard pre-requisite for individual adoption. Venkatesh et al., who have reviewed all the major theories relating to individual adoption, state: “The technologies that have been studied in many of the model development and comparison studies have been relatively simple, individual-oriented information technologies as opposed to more complex and sophisticated organizational technologies” (Venkatesh et al. 2003). Rogers also reflects on existing research on organizational innovation adoption, and states that early research in this area has tried to apply theories of individual adoption to the organizational level. However due to this oversimplification and not considering the organizational complexities, these studies have very little explanatory power (Rogers 1983, p.355; Gross et al. 1971, p.22)

A further issue in applying one of the theories examined in this study was that the explanatory power was very weak with regards to our objectives, i.e., they did not support our target of giving design recommendations to practitioners seeking to digitize their online buying and distribution process for EAS in the app store model. Despite the fact that most theories are very generic, they neither reflect the context of EAS, the context of e-commerce systems in the B2B context, or the combination of both contexts. Venkatesh et al. formulates the issue as follows: “Despite the ability of the existing models to predict intention and usage, current theoretical perspectives on individual acceptance are notably weak in providing prescriptive guidance to designers.” They continue by proposing that future research focus on “the causal antecedents” of the “cognitive phenomena” included in UTAUT and explicitly name “system characteristics” as an important area (Venkatesh et al. 2003).

The DOI theory as one of two frameworks applicable to organizational adoption has been widely applied due to its generic nature. However, we see this as its major disadvantage. The example of applying DOI theory to the adoption of EDI technology confirmed this conjecture: “Due to the inattention to [the technological] features DOI models could not explain EDI adoptions. Instead, we observed that the diffusion ‘factors’ had to be changed radically due to the complex and networked nature of the technology” (Lyytinen & Damsgaard 2001).

Last, the TOE framework showed that the three context categories introduced are in line with empirical studies that apply the TOE framework as Baker highlights. However, for each of the three categories, researchers had to identify and use unique factors for each specific technology studied (Baker 2012). Dedrick and West even offer a conclusion regarding the TOE framework: “the so-called TOE framework […] is little more than a taxonomy for categorizing variables, and does not represent an integrated conceptual framework or a well-developed theory" (Dedrick & West 2004).

We finally decided not to directly apply any of the existing models to our research. Instead we proceeded with an exploratory, qualitative approach to uncover the structures and unique factors of the research context at hand. Specifically, we designed our research method to identify both drivers of and barriers to organizational adoption which could then be translated into recommendations for designing processes and systems to support the app store for EAS adoption in real-world contexts (see also Chapter 3.3 on the respective methodology).

# 3 Methodology

In this Chapter the chosen research strategies, designs, and methodologies will be presented. First an overview will be given of all research artefacts (Chapter 3.1), how they are grouped, how they relate to each other, and the epistemological orientation of the overall research strategy will also be introduced. Afterwards the major part, the empirical research will be further classified (Chapter 3.2). In Chapter 3.3 the methodology of the papers using a cross-sectional design is presented in detail. Chapter 3.4 presents the respective information for the papers using a case study design. Last, Chapter 3.5 introduces the methodological background of the design-oriented artefact introduced with this introductory paper in Chapter 5.

## 3.1 Overview of Research Artefacts, Research Methods, and Epistemology

Figure 25 gives an overview of the individual research artefacts as already introduced in Chapter 1.3 using a different form of representation.

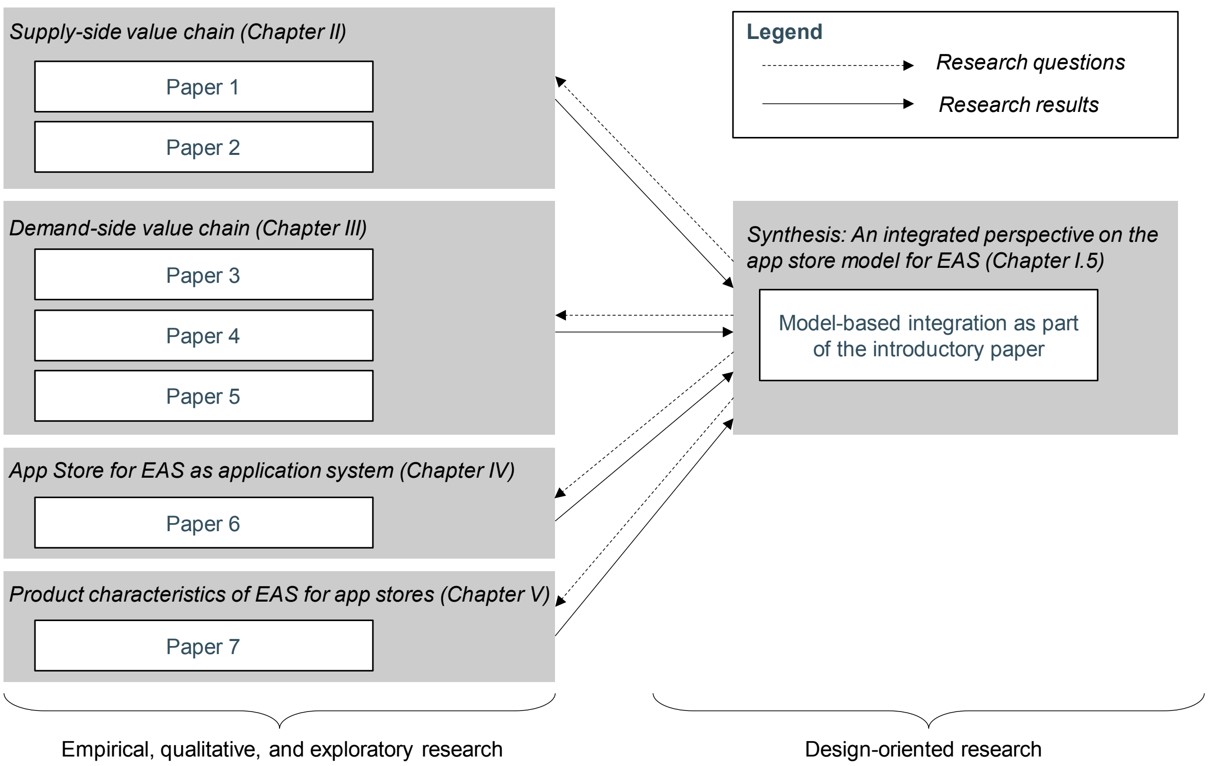


**Figure 25:** Research artefacts, methodological classification, and relationships among research artefacts

The illustration groups the artefacts along two dimensions and puts them in relation to each other. First, five different content-related groups are delimited – the supply-side value chain, the demand-side value chain, app store for EAS as an application system, product characteristics of EAS for app stores, and the integrated perspective on the App Store Model for EAS. Paper 1-7 are assigned to the first four groups and contribute their individual research result to the integrated perspective. The relationship between the integrated perspective and the other sections, however, is bi-directional in that the integrated perspective of the app store model was used to derive the first four groups and the respective research questions as introduced and elaborated in the introduction chapter (cf. Chapter 1.1.2, 1.2, and 1.3).

The second dimension groups the artefacts with regards to their methodological orientation. Whereas Papers 1-7, the majority of the thesis, constitute empirical, qualitative research with an exploratory purpose, the synthesis research artefact in Chapter 5 can be assigned to the design-oriented research stream. Consequently, the overall methodology of the thesis could be interpreted as a "mixed method", combining behavioral and design-oriented elements. Following the classification designed by Huysmans and De Bruyn, who analyzed the relation between design-oriented and behavioral research, the present work would be defined as "BEHAVIORAL → design" (Huysmans & De Bruyn 2013). Following their use of syntax, the capitalized behavioral element indicates which part dominates the overall orientation and represents the main part, while the arrow indicates the pacing of the elements (and the flow of the research results).

The epistemological orientation of the thesis follows the ideas of what is generally referred to as constructivism,[1] in contrast to positivist research (ontological naive realism). Hence, following a constructivist paradigm means recognizing that knowledge cannot be obtained fully objectively and there is always a subjective influence. There is a context relationship of the subject with reality (cf. Niehaves & Becker 2006; Ferstl & Sinz 2013, p.135 ff.). The epistemological orientation has largely influenced the selection of the research methods (i.e., favoring qualitative and design-oriented strategies over positivist, quantitative research strategies) and the nature of the generated research results (ideographic versus nomothetic) (compare Ponterotto 2005; Creswell 2003, p.6 ff. Zelewski 2014). Moreover, the epistemological orientation is important when assessing the quality, i.e., the rigor of the research artefacts (e.g., generalizability of research results).

## 3.2 Classification of Empirical Research

The research methodological taxonomy as presented in Table 12 is derived from Bryman & Bell and classifies the empirical works in this thesis, i.e., Papers 1-7, using five categories: research process, logic, purpose, design, and method.

The *research process* defines whether to follow an empirical, i.e., qualitative or a quantitative approach or a design-oriented one. Quantitative research is preferable to test theory, thus it follows a deductive *research logic*, whereas qualitative research is mostly used to develop

[1] Often also called "constructionism".

theory, hence it follows an inductive logic (Bryman & Bell 2011, p.27). Moreover, the *purpose of a study* can be descriptive, exploratory, explanatory / confirmatory, predictive[1], or design-oriented[2] (A. Hevner et al. 2004; Wilson 2010, pp.102–106; Collis & Hussey 2009, pp.4–8). App stores for EAS are a rather new socio-technological context with limited practical data available, and as a result a dominant design of the value chain or app stores has not established itself in the enterprise market as it has in the consumer software segment. This is the reason why mainly inductive, exploratory, qualitative research strategies have been chosen to investigate this novel context in which categories are still in the making and where organizational issues are closely linked with IS-related aspects. (Darke et al. 1998; Bryman & Bell 2011, p.26 ff.).

The *research design* is a framework that guides the execution of a research method and the analysis of the data. In this thesis various variants of case studies and cross-sectional designs have been used. Cross-sectional designs are used to investigate a specific population (e.g., all business managers buying EAS) at a single point in time. Case studies investigate a single case or a few cases (in the case of multiple or comparative case studies) to reveal in-depth features or characteristics (Bryman & Bell 2011, p.68).

*Research methods* describe the instruments and techniques to collect the data, analyze the data, and develop theories or concepts. Multiple methods have been used in this thesis that are individually selected and tailored to the individual research questions and settings. In particular, expert interviews combined with qualitative data analysis, analysis of public documents and sources, the evaluation of application systems, and model creation and analysis were the methods chosen. Since qualitative research offers plenty of choices and freedom with regards to the configuration of the research process and the available methods it is necessary to thoroughly plan and document each individual research endeavor.

[1] "Predictive" research is only mentioned by Collis and Hussey (Collis & Hussey 2009, p.5).

[2] "Design-oriented" was not mentioned as a category by the referenced business research literature since it is not typically present in behavioral, empirical research. Design science and behavioral science (quantitative or qualitative empiric research) are seen as two distinct, yet compatible research paradigms (Österle et al. 2010; A. Hevner et al. 2004; Huysmans & De Bruyn 2013). The category "purpose" seemed most appropriate for integrating design science aspects into this classification for a mixed method approach.

**Table 12:** Research classification of individual research artefacts

| Paper | Title | Research classification |
|---|---|---|
| 1 | Platform-as-a-Service | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory / descriptive<br>▪ **Design:** multiple mini case studies<br>▪ **Method:** analysis of public documents |
| 2 | Software Ecosystem Orchestration: The Perspective of Complementors | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory<br>▪ **Design:** case study<br>▪ **Method:** semi-structured expert interviews, qualitative data analysis |
| 3 | Evaluating the App-Store Model for Enterprise Application Software and Related Services. | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory<br>▪ **Design:** cross-sectional design<br>▪ **Method:** semi-structured expert interviews, qualitative data analysis |
| 4 | Adoption of an online sales channel and “appification” in the enterprise application software market: a qualitative study. | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory<br>▪ **Design:** cross-sectional design<br>▪ **Method:** semi-structured expert interviews, qualitative data analysis |
| 5 | Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity? | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory<br>▪ **Design:** cross-sectional design<br>▪ **Method:** semi-structured expert interviews and qualitative data analysis |
| 6 | Comparing Public and Internal Enterprise App Stores: A Qualitative Case Study. | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory, design-oriented<br>▪ **Design:** comparative case study<br>▪ **Method:** review of public documents, expert interviews, application system evaluation, model creation & analysis |
| 7 | App’ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications. | ▪ **Process:** qualitative<br>▪ **Logic:** inductive<br>▪ **Purpose:** exploratory<br>▪ **Design:** multiple case study<br>▪ **Method:** review of public documents, focus group |

However, it is necessary to mention that the classification presented in Table 7 is only an approximation. For example, the individual research, even if inductive and qualitative, might use existing theory as part of the process (e.g., Paper 3, 4, 5), or the results are used to *predict* an outcome if certain design recommendations are put in place even though the nature of the results are *exploratory* or *descriptive* (Paper 7). Furthermore, the individual research processes have been carefully designed and tailored to meet the research objectives – a classification as presented in Table 12 can therefore only give a rough orientation in this regard.

## 3.3 Qualitative Cross-Sectional Research

Cross-sectional designs analyze cases at a single point in time. Though cross-sectional designs are often automatically associated with a quantitative strategy, they are also common in the case of qualitative interviewing and data analysis (Bryman & Bell 2011, p.57).

Qualitative cross-sectional research with a largely nomothetic[1], inductive approach has been used for the papers 3, 4, and 5 (cf. sections III.1, III.2, and III.3) to examine the demand-side value chain. We have relied on semi-structured interviews and qualitative content analysis in the respective studies. Moreover, all three papers share a common research process and also largely share a common data base. Figure 26 illustrates the simplified research process using three phases, the interview design, data collection, and data analysis and theory building.

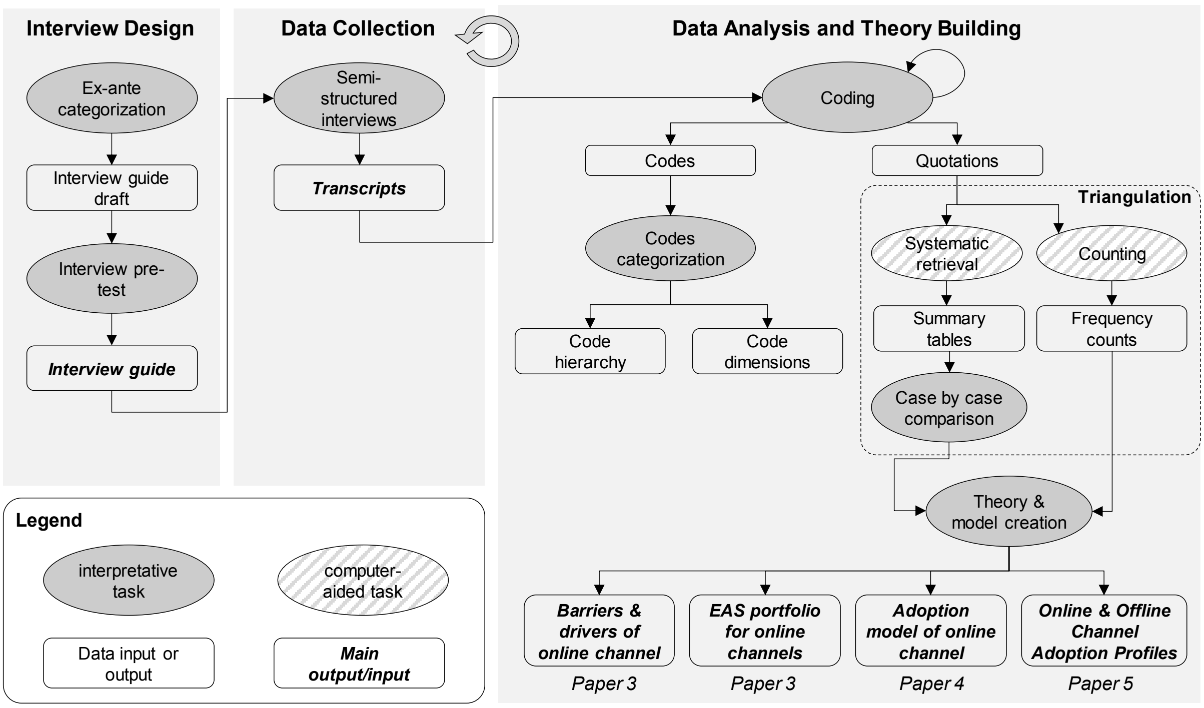


**Figure 26:** Common simplified research process of Paper 3, 4, and 5

### 3.3.1 Interview Design

During the interview design phase, we initially compiled a set of questions based on deductive categorization from the relevant literature. In particular, we considered research from the following areas: SECOs and platforms, organizational buying, enterprise software buying, e-commerce, and IT adoption theories (compare Chapter 2). Based on these findings we created a draft interview guide that was then tested at the trade fair CeBIT in Hannover in March 2012. At the fair, ten candidates were recruited and interviewed for approximately 30 minutes each. Four candidates represented customer companies for EAS and six were from EAS providers

[1] Some results of this research also classify as idiographic.

or resellers. Based on their input we could revise the interview guide by adding, removing, rephrasing, and re-ordering questions. Moreover, we developed graphical illustrations of key concepts which were shared with the interview candidates at the beginning of the interview. The revised interview guide was then thoroughly reviewed using an experienced senior sales executive for EAS, and afterwards minor adjusted were made in terms of wording and presentation.

The guide consists out of four parts. The initial part includes typical profile questions (e.g., questions about job role, industry experience) and questions about personal experience with app stores and online channels to purchase software, both in a professional and private context. Moreover, prior to starting with the main parts of the interview, it includes the graphical representations of the buying process and a stylized product portfolio (detailed explanation, see below). The second part walks the interviewee through the buying decision process for EAS. During the individual buying process phases, the interviewee is asked to first describe this phase as it is conducted today in his organization, and then to imagine conducting the phase using an online channel, such as an app store for EAS (to identify the benefits, drawbacks). Furthermore, each phase was reviewed from the perspective of different EAS products and their individual product characteristics (see stylized product portfolio below). Thereafter, the third section focuses on the factors influencing the decision for choosing an online or offline channel. The last part is then used to ask the interviewee if aspects have been overlooked and should be included. Overall, the questions in the interview guide were largely open-ended to allow the interviewees to elaborate on certain topics in more detail, to provide new insights, and to add their personal view of the discussed topic.

**Graphical schematizations**

During the preliminary interviews we noticed that the wide range of products offered in the EAS market and the variety of buying decision processes in companies, and sometimes even the different uses of terminology led to confusion or misunderstandings between the interviewer and interviewee. To establish a common understanding, we have therefore included graphical illustrations of the buying decision process and developed a stylized EAS product portfolio to effectively address the questions in the interview guide. Moreover, we noticed that product characteristics seem to influence the choice of channel.

The buying process illustration (see Figure 27) was derived from the ERP buying process proposed by Verville and Halington (Verville & Halingten 2003 and Chapter 2.4.2). The process we used comprised five phases: problem recognition, information search, evaluation of alternatives, negotiation & purchase, and aftersales.

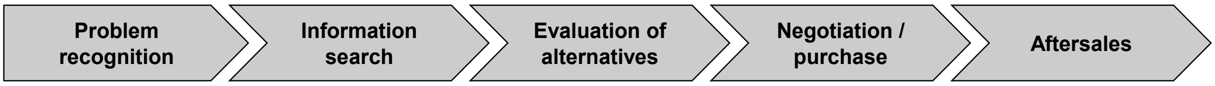


**Figure 27:** Illustration of the buying process phases introduced to the interviewees

The phases of the buying process typically cover the following activities:

- *Problem recognition:* The customer gains awareness of an opportunity, need or threat which can be dealt with by acquiring an EAS product.
- *Information search:* The customer acquires information material about all potential providers and solutions.
- *Evaluation of alternatives:* The customer evaluates the individual solutions, e.g., by ranking selected solutions and providers using pre-defined criteria.
- *Negotiation & purchase:* The customer finalizes the terms of the transaction, stipulates contracts, and executes the purchase.
- *Aftersales:* The customer purchases additional goods related to already acquired solutions (e.g., service level enhancements, additional licenses)

The stylized EAS product portfolio (see Figure 28) includes four types of solutions: core solutions, on-top solutions, usage enhancements, and EAS-related IT services. Core solutions are either company-wide systems such as ERP or systems dedicated to one organizational or functional area, such as SRM or CRM. One decisive characteristic of core solutions is that they are mostly standalone, i.e., they do not necessarily rely on other solutions in order to be used. On-top solutions are software applications or just single components adding additional functionality, or alternative front-ends to core solutions. Usage enhancements are products purchased in the aftersales phase. They enhance existing solutions without touching the application code or application configuration. This includes, for example, user licenses, usage contingents, service level agreements with the provider. Last, IT services include professional services related with the respective solution, such as implementation, customization, data migration, or training services.

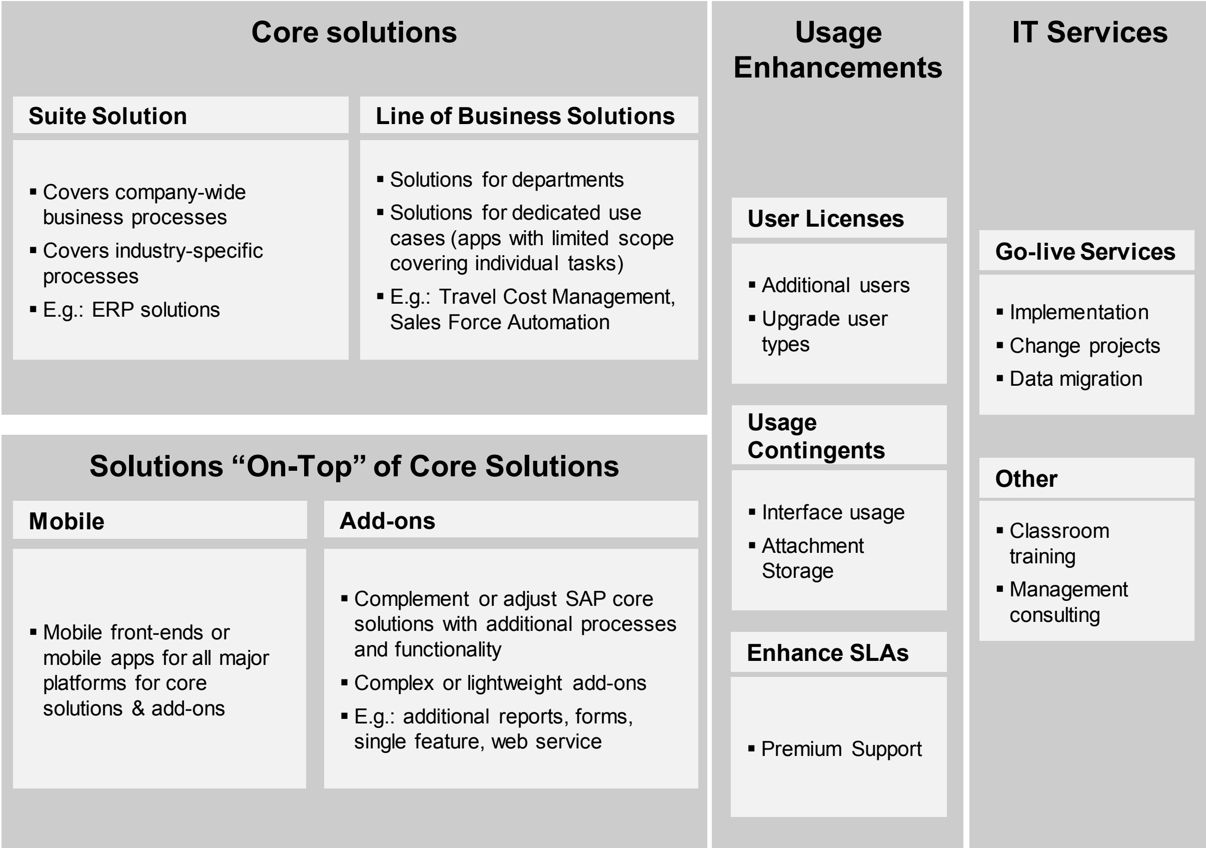


**Figure 28:** Stylized product portfolio used in the interview guide

### 3.3.2 Data Collection

To recruit the interview candidates, we used convenience and snowball sampling. This allowed us to make direct use of our professional network[1] within the EAS ecosystem (both convenience and snowball). Our approach to sampling can be classified as theoretical sampling, as used in grounded theory (GT) and that Corbin and Strauss argue is more appropriate to qualitative research than statistical or probability sampling (Corbin & Strauss 1990).

Recruited interview candidates were also asked to provide further relevant contacts from their own network (snowball sampling). The candidates were either approached telephonically or via e-mail. As indicated in Figure 26 (by the circular arrow), the research process was not conducted in a rigidly linear approach, but the data collection and data analysis part passed through multiple rounds. This approach followed the concept by Glaser and Strauss that seeks to reach a sufficient level of saturation in qualitative research (Glaser & Strauss 1967, pp.61–62). Table 13 gives an overview on all interviews conducted and used for data analysis (not

[1] The researchers’ professional networks covered SAP as a major EAS provider, the SAP partner network, the SAP research community, additional Germany-based enterprise software vendors, multiple universities and IS university chairs incl. the University of Bamberg, TU Darmstadt, Karlsruhe Institute of Technology, University of Freiburg, and the University of Mannheim.

including the interviews conducted for designing and testing the interview guide). Three rounds of interviews can be distinguished: First round including interview 1-4, second round including interviews 5-12, and third round including interviews 13-16. Paper 3 is based on the data collected from interviews 1-12, whereas papers 4 and 5 are based on all 16 interviews.

We have carefully screened the profiles of all interview candidates and applied the following minimum selection criteria:

- active participation in the buying process of EAS, at least once;
- sufficient level of experience, i.e., senior level with a minimum of five years job experience.

Moreover, we ensured a highly diversified set of organizational contexts with regards to industry, company size, role of the company, career level, and job role. The company role determines the relationship of the company towards EAS. We interviewed candidates from customer companies, multiple EAS vendors, and partner companies (service, sales, or solution partners) from large EAS vendors. Apart from interview 12, all companies operated globally. All candidates declared that they were familiar with organizational software purchases and have participated in their organization’s software buying processes in one or more roles (e.g., decider, influencer, or buyer). The sales executives were all highly experienced and had responsibilities as account executives for multiple customers throughout their respective careers.

**Table 13:** Sample details

| # | Personal profile | | Organizational profile | | | Date | Sampling | Paper |
|---|---|---|---|---|---|---|---|---|
| | Level | Job role | Company role | Industry | Size* | | | |
| 1 | Senior | Sales exec. | EAS vendor | IT/software | LE | March 2012 | Snowball | 3,4,5 |
| 2 | Manager | Sales exec. | EAS vendor | IT/software | LE | March 2012 | Snowball | 3,4,5 |
| 3 | Senior | Sales exec. | EAS vendor | IT/software | LE | March 2012 | Snowball | 3,4,5 |
| 4 | Manager | Sales exec. | EAS vendor | IT/Software | LE | April 2012 | Snowball | 3,4,5 |
| 5 | Manager | Line of business | Customer | Manufacturin g | LE | April 2012 | Snowball | 3,4,5 |
| 6 | Manager | Line of business | Customer | Manufacturin g | LE | May 2012 | Snowball | 3,4,5 |
| 7 | Executive manager | CxO | Partner of EAS vendor | IT/software | SME | May 2012 | Snowball | 3,4,5 |
| 8 | Manager | IT | Partner of EAS vendor | IT/software | LE | May 2012 | Convenience | 3,4,5 |
| 9 | Manager | IT | Customer | Retail | LE | May 2012 | Snowball | 3,4,5 |
| 10 | Senior | Line of business | Partner of EAS vendor | Management consulting | SME | May 2012 | Convenience | 3,4,5 |
| 11 | Manager | IT | Customer | Finance | LE | June 2012 | Snowball | 3,4,5 |
| 12 | Manager | IT | Customer | Public administratio n | LE | June 2012 | Snowball | 3,4,5 |
| 13 | Executive manager | CxO | EAS vendor | IT/software | SME | August 2012 | Convenience | 4,5 |
| 14 | Manager | IT | Customer | Tele-communicat. | LE | August 2012 | Convenience | 4,5 |
| 15 | Manager | IT | Customer | Finance | SME | August 2012 | Convenience | 4,5 |
| 16 | Manager | Line of business | Customer | Manufacturin g | LE | August 2012 | Convenience | 4,5 |

**Abbreviations: SME = small or medium sized enterprise; LE = large enterprise.*

The large heterogeneity of the sample ensured that different views about the issues were captured. Furthermore, the high professional seniority of the interview candidates (13 candidates were managers or executives), the good mix between candidates from business departments, IT departments, and the board room, ensured that we gained holistic insights based on rich and multi-year experiences. Since we also included sales executives we broadened our sample, since they not only provided their personal view but shared experiences and views from current and previous customers. In contrast to quantitative research, many qualitative researchers prefer a high diversity in a sample, since the prime target is not to find “an average”, or “the majority” equipped with statistical significance, but a broad spectrum of ideas, concepts, or opinions to uncover structures and concepts. Although sample size should not be a criteria, but instead saturation with regards to emerging concepts, qualitative researchers often propose samples of 12-26 persons (Luborsky & Rubinstein 1995). In our case, we already experienced a high saturation after analyzing the interviews of the first 12 candidates, and thereafter only minor new insights and structures could be found and the models main concepts only stabilized (increasing the frequency counts of certain relationships or arguments).

The interviews were conducted in 2012, between March and September and were either conducted in person (7 out of 16) or via phone (9 out of 16). In the case of phone interviews, the graphical representations were either shared via a conference system or via an electronic document prior to the interview. The conversations took place in German, the mother tongue of all the participants. All interview candidates were informed about the purpose of the interview, which is part of a research project with a subsequent publication of results, and all agreed to digitally record the interviews. Moreover, the candidates were guaranteed personal and organizational anonymity. Each interview lasted about an hour. Around two thirds of the interview time was used for the major interview parts two and three. In total 939 minutes of interviews were recorded. Thereafter full transcripts were created totaling approximately 111,000 words.

### 3.3.3 Data Analysis and Theory Building

**Qualitative Data Analysis**

During the data analysis, our chosen technique to analyze the transcripts was coding. Whereas multiple researchers are often only a means to divide labor, especially in the data analysis of this research, it was a necessary element of the research process (also supporting reliability and trustworthiness, cf. Chapter 3.3.4). Throughout the analysis process the researchers created notes and memos to track methodological issues (e.g., coding issues) or conceptual developments (newly emerging relevant variables or relationships). Moreover, all main interpretative tasks were performed in multiple full-day workshops together.

Initially three researchers coded the same transcripts in parallel. The codes of the individual researchers where then compared, grouped, and mapped to create a common codebook. The codebook included 62 standardized codes. Thereafter, all interviews were coded by the researchers, whereas each coded transcript was reviewed at least by one more researcher to ensure completeness of coding and a common understanding of the codes used and the coding techniques. Questions arising among the researchers with regards to coding decisions were discussed and resolved during the repeated workshops. The coding process as well as subsequent code analysis was supported by computer aided qualitative data analysis software (CAQDAS), i.e., “Atlas.ti”, or simply by using spreadsheets.

During the iterative coding process we applied multiple different coding techniques (cf. Table 14). They can be grouped in first cycle coding and second cycle coding methods. First cycle coding is done during the initial round and mostly codes the obvious and direct aspects, whereas second cycle coding applies codes to already coded data but of a more analytical and interpretative nature (Saldaña 2009, p.45).

**Table 14:** Coding techniques used for paper 3, 4, and 5

| Cycle | Coding method | Code example | Quotation example |
|---|---|---|---|
| 1 | Attribute | PROFILE INFO: ORGANIZATIONAL ROLE | *"I am in global sourcing […] and I do strategic projects on how to further develop our sourcing department."* |
| 1 | Structural + subcoding | PROCESS PHASE: INFORMATION SEARCH | *"I believe information search somewhat depends on the need of explanation of the respective products."* |
| 1 | Descriptive + subcoding | SOLUTION DEPLOYMENT \| ON-PREMISE | *"For an on-premise solution, one probably still has the traditional mindset and will raise a request directly to the company."* |
| 1 | Values | BARRIER OF ADOPTION | *"On-premise rather via the traditional channels because the initial investment is higher and the scope is more complex. This does not fit very well with the electronic channel."* |
| 2 | Simulta-neous | PRODUCT TYPE: CORE \| PRICE | *"Yes, the price does have a significant impact, especially when we talk about large and expensive solutions like ERP."* |
| 2 | Magnitude | NUMBER OF USERS \| EFFECT DIRECTION (-) | *"If I had to buy 2000 Windows licenses, I would definitely try to reach out to the sales person and negotiate the price."* |
| 2 | Evaluation | IMPLEMENTATION EFFORT \| BARRIER OF ADOPTION \| EFFECT DIRECTION (–) | *"If you must customize the software highly, you will have intense personal contact and this will not work in an automated fashion."* |

* "|" separates multiple codes applied to the same datum; ":" separates a code category label (e.g., process phase) from the specific code instance in the quotation (e.g., information search)

*Attribute coding* was used to describe the basic characteristics of the interviewee and his organization. *Structural coding* annotates the text with the structural elements used in the interview, such as the buying process phases or EAS product types (e.g., core solution). *Descriptive coding* is the major technique used to analyze the content and topics of the transcript. *Values coding* in general reflects the interviewees' opinion or beliefs. In our context we mainly characterized whether the interviewee perceives a certain characteristic as a barrier or a driver for the adoption of an online sales channel. *Simultaneous coding* is the use of two or more codes on the text passage. Simultaneous coding was applied in different forms, either by simply using different codes on the same text or by applying a code hierarchy including a category label, and an instance label. For second cycle coding methods we applied magnitude and evaluation coding. *Magnitude coding* assigns a symbol or alphanumerical code to an existing code. In our study we introduced the magnitude code "effect direction" to indicate if an incremental unit of the label has a positive (“+”) or negative (“-“) effect on the adoption of an online channel. *Evaluation coding* as a second cycle method has a similar function as value coding. Whereas value coding labels rather than directly quoting, evaluation coding includes the interpretation of text passages and reflects the attitudes and judgments of the interviewee that is hidden in or between statements (Saldaña 2009, pp.45–101).

In total more than 1500 quotations were coded using the techniques presented above, and these formed the basis for the subsequent analysis. Descriptive, value/evaluation, and magnitude codes were categorized using coding hierarchies and code dimensions (i.e., a scale representing code properties, such as quantity) (Gibbs 2007, p.76). Furthermore, the codes where used to systematically compile quotations and create summary tables. The summary tables allowed an in-depth analysis of the content and case-by-case comparisons (see Figure 26).

To support the interpretative tasks during the analysis phase, we also applied counting techniques. Counting involves quantifying non-numerical qualitative data (Hannah & Lautsch 2011). There was, and remains, quite a controversial discussion among qualitative researchers as to whether numbers should be used at all in qualitative research; these arose mainly for philosophical and political reasons that originated from the qualitative versus quantitative debate. Maxwell (Maxwell 2010) and Hannah and Lautsch (Hannah & Lautsch 2011) comprehensively review and trace this controversy. However, they both conclude that there are numerous reasons for the purposeful application of counting. Maxwell argues for four major benefits of counting (Maxwell 2010):

- support of internal generalizability and validity (not generalizability across different settings, but within the studied setting);
- to characterize the diversity of identified concepts (beliefs, attitudes etc.) in the qualitative material (e.g., to avoid the typical bias to seek for uniform concepts);
- identify patterns which are not apparent in the qualitative data complementing the interpretations of the researcher or even patterns which are not apparent to the interviewee;
- provide evidence for researchers interpretations, “to counter claims that you have simply cherry-picked your data for instances that support these interpretations” (Maxwell 2010).

Hannah and Lautsch (Hannah & Lautsch 2011) distinguish four purposes of counting:

- *Autonomous counting to* produce standalone numerical research findings.
- *Supplementary counting to* enable researchers to generate additional, new insights.
- *Corroborative counting* to verify qualitative interpretations. Is used in a triangulation approach to come to the same result using two independent processes.

- *Credentialing counting* increasing the confidence in the researcher's findings by documenting the characteristics of the data set (e.g., size, number of sources) and by providing evidence of the analytical honesty of the researchers (i.e., either representativeness of findings or objectivity of the coding process).

Following the logic of Hannah and Lautsch (Hannah & Lautsch 2011) we have applied supplementary, corroborative, and credential counting in our study. By using computer-aided proximity analysis (analyzing the similarity among quotations) we were able to identify new patterns in the form of relevant code co-occurrences (supplementary counting). However, the frequency counts of code co-occurrences should not be misunderstood, in that they are not results in terms of identified correlations with statistical significance (as in quantitative research). This is because, first, the study used non-random sampling, and second because the code co-occurrences were not automatically used as a result but were only interpreted as an additional hint for possible new findings. In other words, we qualitatively analyzed all numerical findings and only thereafter decided upon its relevance.

Furthermore, we also combined counting and interpretative analysis in a triangulation approach to address the same research question using both methods (corroborative counting). Corroborative counting, however, was always seen as a subordinate process and followed the interpretative task of the researcher so as to avoid frequency counts influencing or biasing the interpretative task.

Last, we employed counting to select a subset of cases for further interpretative analysis, and in particular we used the number of code occurrences or co-occurrences. We are convinced a complex topic can only be analyzed and interpreted faithfully and comprehensively if sufficient material is available. For example, we used counting to pre-select the potential interdependencies among the identified adoption factors. This way we ensured that the cases and interpretations were grounded in the data and not just subjectively selected by the researchers, thus supporting the objectivity of the research process (credentialing counting).

**Model Creation**

The data which has been analyzed and prepared was afterwards interpreted and translated into multiple tables and diagrams. At the core of all the results generated is the "adoption model of an online channel for EAS". The model creation of this artefact followed the example of Ferstl and Sinz (cf. Ferstl & Sinz 2013, p.135 ff.)

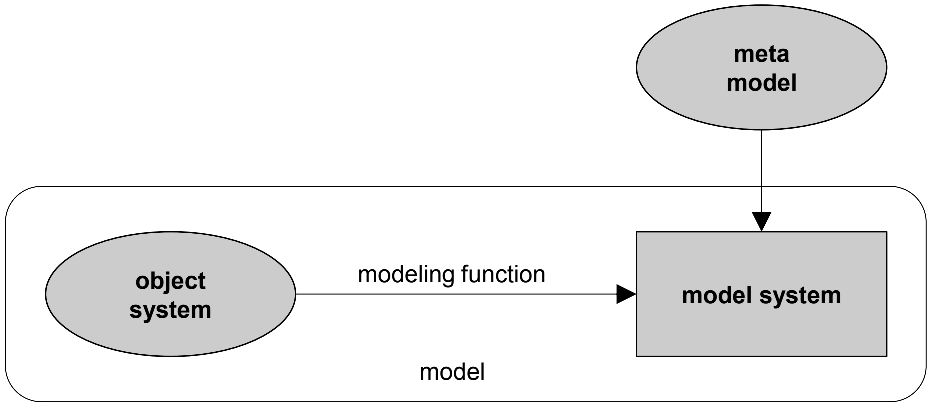


**Figure 29:** Model, object system, model system, model mapping function according to (Ferstl & Sinz 2013, p.135)

Thereafter a model is defined as a 3-tuple consisting of an object system, a model system, and a model mapping function (short: modeling function) mapping the object system to the model system (see Figure 29). This model definition is underpinned by a constructivist modeling view as depicted in Figure 30. Accordingly, the object system as a subset of the reality is delimited by the subject through perception and interpretation. The subject itself has a contextual relationship to the reality. The perception and interpretation of the subject is influenced by the objectives he or she pursues with the model. On the basis of the model objectives and the interpretation, the model system is created with the help of a so-called meta model and a metaphor. Meta model and metaphor aim to reduce the individual subjectivity but also support consistency within a team of researchers. The meta model defines the available model components and the relationships among the model components. The metaphor is a description of the modeler's, i.e., the subject's viewpoint or perspective used, as a basis for capturing the object system and which is transferred to the model system (Ferstl & Sinz 2013, p.135 ff.).

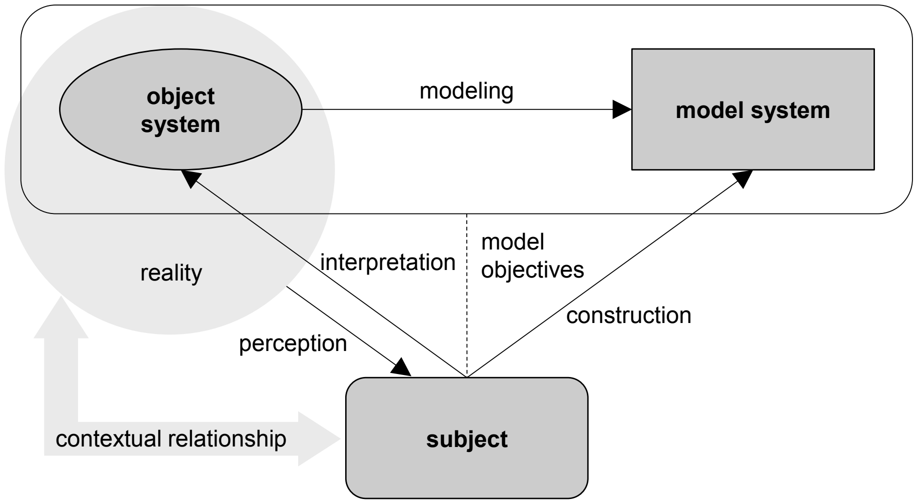


**Figure 30:** Constructivist modeling view (Ferstl & Sinz 2013, p.136)

This understanding of models is applied to the analysis of the online channel adoption of EAS customers. The outcome is a semi-formal adoption model including barriers and drivers of adoption. The *model objective* is described as follows: Identify the determinants of adopting an online channel for EAS from the software customers' perspective. The *metaphor* used is: The adoption of an innovation is influenced by determinants, which are either drivers or barriers. The *meta model*[1] used is illustrated in Figure 31 and was created on the basis of a causal diagram. Hence, the model consists of factors and relationships. The factors are either influencing factors or outcome factors. There is exactly one outcome factor and any number of influencing factors. Two types of directional relationships are possible: 'from influencing factor to influencing factor' and 'from influencing factor to outcome factor'. The relationships can either have a positive ('+') or negative ('-') prefix determining the effect of the relationship. The semantic of a relationship between two factors (x, y) can be interpreted as follows: an incremental unit of factor x has a positive or negative effect on factor y. The model is semi-formal since neither the unit of the factors nor the strength of the effects is accurately defined. The modeling was hence a multi-step procedure including the selection of interview candidates, conducting the interviews, recording, transcribing, and analyzing the data and ultimately interpreting the analysis results, and applying the meta model.

[1] The meta model was created on the basis of the meta-meta-model defined by Ferstl and Sinz (Ferstl & Sinz 2013, p.139).

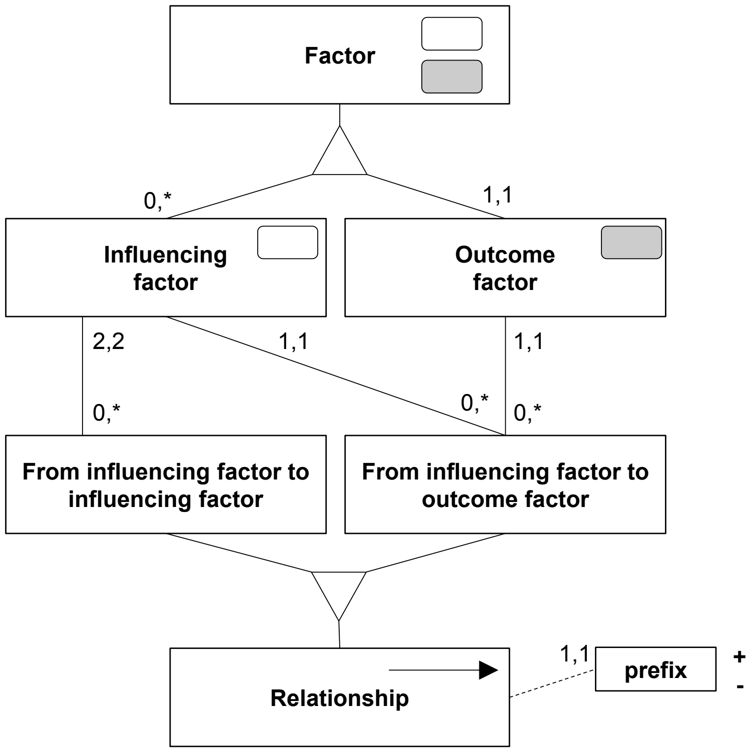


**Figure 31:** Meta model used to construct the adoption model for an online channel for EAS

From the analysis, multiple results could be derived which were published in three papers (Papers 3, 4, and 5; cf. Figure 26). The major ones are (cf. Chapter 4.2 and Part III: Demand-Side Value Chain):

- Paper 3: Barriers and drivers as determinants for the adoption of an online channel for EAS;
- Paper 3: Appropriateness or suitability of the different EAS products for an online channel;
- Paper 4: Adoption model for an online channel for EAS (incl. interrelations among adoption factors and "app'ification" patterns);
- Paper 5: Adoption profiles of online and offline channels for EAS along the buying process; (incl. integration of online and offline channels).

### 3.3.4 Ensuring Quality of Qualitative Cross-Sectional Research

To assess qualitative research, academic researchers have proposed two different approaches. The first approach is to use criteria traditionally applied to quantitative research, in particular reliability and validity, and to adapt these to the qualitative paradigm. The second one applies criteria uniquely defined for qualitative research (Bryman & Bell 2011, p.394 ff.). Both approaches have been considered throughout the planning and execution of our research and will be reviewed and reflected upon in this section.

**Quality assessment based on adapted criteria from the quantitative tradition**

The criteria originating from quantitative research, which have been adapted to qualitative works are the following (Bryman & Bell 2011, p.395):

- *External reliability* refers to the possibility of replicating a study.
- *Internal reliability* is used to express whether the research team members are consistent in what they observe and interpret.
- *Internal validity* is the degree to which the observations match the theories developed. Internal validity confronts questions like ‘Are the theories able to cover the observations comprehensively?’, ‘Can the conclusions and causalities be trusted?’
- *External validity* is the degree to which the results of a study can be generalized. Can the results be transferred to other contexts?

The *external reliability* (possibility of replicating) of our research is rather low. This, however, is very typical for qualitative research since the setting in which the data is retrieved is unique. Our research, for instance, was influenced by the organizational context in which it was conducted (e.g., the selection of interview candidates via the professional network of the research team). Despite this, we made transparency a number one priority by extensively documenting our research methodology which was tailored to the objectives of our study. This allows third parties to comprehend our approach step-by-step and examine it for logical or conceptual issues.

The research team consisted of three researchers in the early phase and two researchers in the later phases. The *internal reliability* among the team members was ensured by using multiple measures from the start. The team was structured according to professional project standards incl. a common project plan, a project charter, a shared network drive with standardized folder structure, a common literature base, regular weekly meetings with minutes, and finally a series of day-long workshops. In particular, interpretative tasks were performed together in workshops and key activities were thoroughly planned and aligned: e.g., a common codebook was used and a common CAQDAS using a common database to perform coding and qualitative analysis.

Strengthening of *internal validity* was also fostered throughout the research process. The interview guide was developed via multiple iterations and reviewed with experts. The full interview transcripts were created by researchers’ assistants who were all familiar with the topic and native German speakers (the interviews were conducted in German). To avoid coming to the wrong conclusions based on the results of a single analysis technique and the subjectivity involved in interpretative tasks, we applied triangulation: e.g., we preceded purely qualitative analysis tasks with counting to prevent so-called “cherry picking” (cf. Maxwell 2010).

According to Marshall and Rossman, no qualitative research fulfils the criteria of *external validity* (generalizability) (Marshall & Rossman 2006, p.42). Our research can also not fulfil this criteria in the probabilistic sense. This is, first of all, due to our non-random sampling design, and second due to the small sampling size. As comprehensively outlined in Chapter 3.3.2 on data collection, the quality of the sample for qualitative research should instead be rated with regards to the saturation of emerging concepts and variety within the sample, and the abstraction of concrete findings to general concepts to extend the theory's reach (Yin 2013, pp.36–37). But we also believe our sample to be equally adequate with regards to size and representativeness. According to Luborsky and Rubinstein's guide, our sample size lies in the range of what is commonly accepted in qualitative research (Luborsky & Rubinstein 1995, p.105). The strength of our sample with regards to representativeness lies in the high variety of individual and organizational profiles as well as the good fit of the selected individuals to our research objectives. This is further supported by the fact that four of our interview candidates were highly experienced sales representatives covering a breadth beyond the boundaries of a single company. In conclusion, we are confident that our results can be transferred to other settings within the boundaries of the EAS domain. Settings beyond EAS might also benefit from our findings. However, every case should be cautiously analyzed before applying our results or procedures. Last, the nature of exploratory, inductive research is that it often provides 'early theoretical findings' which should be further developed, applied and validated in consecutive works. The same applies to the results of this study.

**Quality assessment based on native qualitative research criteria**

Guba and Lincoln have proposed alternative criteria for qualitative research (Guba & Lincoln 1994; Bryman & Bell 2011, p.395). The two primary criteria are *trustworthiness* and *authenticity*. Trustworthiness in turn entails four secondary criteria:

- *Credibility* corresponds with internal validity.
- *Transferability* corresponds with external validity.
- *Dependability* corresponds with external reliability.
- *Confirmability* corresponds with objectivity.

The four criteria of trustworthiness have their counterparts in the quantitative tradition (Guba & Lincoln 1994). Hence the assessment of 'Credibility', 'Transferability', and 'Dependability' equals the description given above concerning the aspects of 'internal validity', 'external validity', and 'external reliability'. The confirmability criteria corresponds to the objectivity criteria used in many quantitative studies as an important aspect of rigor (Guba 1981). Objectivity is the absence of subjective influences derived from the researchers on the results and interpretations, and describes the idea of a neutral, distanced observer. The ideal of an objective investigator originates from the positivistic tradition. However, we believe complete objectivity is not possible. Furthermore, as outlined above in Chapter 3.3.3 (see Figure 30) we pursue a constructivist based view which states that the researcher always has a contextual relationship with reality and pursues certain objectives with the research. As a compromise, Bryman and Bell proposed reducing subjectivity by keeping personal values and theoretical inclinations out of the research process (Bryman & Bell 2011, p.395). Consequently, the objectivity of our study can be criticized due to the fact that all researchers were affiliated with a large software vendor. The affiliated private organization following commercial interests could be accused of holding political interests in the research outcome. However, none of the researchers had received guidance or orders regarding the research process or was asked to hold back or alter certain results. Moreover, to reduce individual subjectivity to a minimum we applied a common metaphor, and strictly followed the outlined research process which in turn followed academically respected and well-accepted research standards.

Instead of trying to achieve the ideal of objectivity, Guba proposes striving for “data and interpretational confirmability”. Moreover, he proposes two elements a researcher should include in order to achieve this: the application of triangulation and the practice of reflexivity. Triangulation is understood by Guba as a high variety in data, methods, and perspectives (Guba 1981). Thereafter we have applied triangulation in many ways: we employed a high variety in our data sample (e.g., different organization sizes, different buying center roles, different professional levels, different industries), we applied multiple methods on the same issue during qualitative data analysis (e.g., qualitative content analysis was combined with counting), and we used the perspectives of three researchers in interpretative tasks. Practicing reflexivity was achieved by reflecting within the team as well as with external experts. Within the team we used the technique of ‘memo’ taking assisted by the common CAQDAS. Memo taking helped us understand the individual interpretations (e.g., on emerging concepts) of researchers within the team and throughout the process. Certain artefacts have also been iteratively reviewed and discussed with external experts, such as the interview guide or the graphical schematizations used in the interview guide.

The second primary criteria is authenticity, which is broken down into a number of secondary criteria, too (Guba & Lincoln 1994; Bryman & Bell 2011, p.398):

- *Fairness* is the degree to which the study represents views from different members of the social setting.
- *Ontological authenticity* is the degree to which the study helps us gain additional insights into the members of the (socio-technological) setting.
- *Educative authenticity* is the extent to which members of the setting better understand perspectives of other members of the setting through the research.
- *Catalytic authenticity* is the degree to which the research has spurred the members of the setting to take action.
- *Tactical authenticity* is the degree to which the research has helped and empowered the members of the setting to take action.

The fairness of our research was achieved by thoroughly ensuring a high variety in our sample of interview candidates (see examples above). We also believe that the insights we have generated are both new and of high value to software customers, platform providers, and independent software vendors, and hence both ontological and educative authenticity should be ensured. Last, we are confident catalytic and tactical authenticity are achieved: the research project was initiated and conducted in the organizational environment of a large software vendor acting as a platform provider in our terminology. The results were regularly and promptly presented both to practitioners closely related with the topic and within academia via highly renowned conferences that helped to further extend the results. The incremental nature of the study and their prompt publication and dissemination via conferences ensure that the results are presented when they are still relevant in an environment which is changing at an accelerating pace.

## 3.4 Case Study Research

Whereas the previously described qualitative cross-sectional studies shared a common research process and largely shared a common, yet evolving, data set, the research processes for the case studies in this thesis are individual, using individual data sources and with research methodologies tailored to the individual case study. Therefore, I will first briefly describe the general characteristics of case study research and thereafter present the unique aspects of the individual case studies featured in this thesis.

Case studies represent one of the most popular research designs in IS research (Wilde et al. 2007). Typically the case study design is chosen in complex scenarios or in situations where little empirical data exists and the structures of the phenomenon are still unclear (Bryman & Bell 2011, pp.59–61; Wilde et al. 2007). Moreover case study designs are chosen to examine contemporary cases where it is difficult to isolate the unit of analysis from the surrounding system, and therefore the object of interest is investigated in its natural context (Dubé & Paré 2003). Though case studies can equally be used in conjunction with a quantitative or a qualitative strategy (Bryman & Bell 2011, p.60), all case studies in this thesis are qualitative with an inductive logic and exploratory purpose combined with descriptive and design-oriented elements. Yin proposes using the case study design if the research question is a “how” or “why” question and whenever that question requires a detailed and in-depth examination of a case (Yin 2013, p.4). Therefore the researcher is interested in the unique features of a case and the results are mostly idiographic (Bryman & Bell 2011, p.60). Consequently, case studies are well suited for the new socio-technological context of app stores for EAS and SECOs, and the presented exploratory research questions (see Chapter 1.2).

The typical case study analyzes one single case in-depth. Yin proposes five different rationales for a single case study (Yin 2013, pp.51–53):

- *Critical case* to test or better understand an existing hypothesis or theory. In this case the research is often explanatory.
- *Unusual or extreme case* deviating from the usual case. Often used in clinical studies, these cases may also provide insights into the normal case.
- *Common case* is used to better understand the processes and structures in everyday cases or situations.
- *Revelatory case* is used to uncover the structures of a phenomenon which has not or hardly been studied and were access to data is difficult.
- *Longitudinal case* investigates a situation changing over time and the way it alters.

However, case studies are not restricted to only one case. Another form is the so-called multiple-case study. In addition to the rationales provided by Yin for single-case studies, multiple-case studies are carried out mainly for three purposes. First, they allow comparisons between cases and thus may explain a certain phenomenon better when two or more cases are contrasted or are put in relation (also called *comparative* case study). Moreover, contrasting cases may allow for the identification of unique (*unusual* or *extreme*) and *common* features across multiple cases (Bryman & Bell 2011, p.63). Second, studying multiple cases is a form of literal or theoretical replication similar to replications done in experiments. Following this logic, a second case may either duplicate findings (literal replication) or reveal contrasting results (theoretical replication) (Yin 2013, p.54). Third, and in particular for exploratory research, selecting multiple-cases allows the *variety* of findings and options to be broadened (Dubé & Paré 2003). In IS research the latter reason may play an important role, since multiple, alternative capabilities can equally solve a single problem. In such a case the research result is not the one and single “truth”, but represents one or more adequate solutions to a problem in line with predefined objectives.

The question might arise regarding what the difference between multiple-case studies and cross-sectional designs is (especially if both are qualitative and exploratory). The major difference between both is the focus of the research. For cross-sectional designs the researcher targets revealing nomothetic findings, i.e., more generic results for a targeted population, whereas the case study researcher is focused on idiographic findings, i.e., unique features of the selected cases (Bryman & Bell 2011, p.63). As Bryman and Bell argue, many studies do have the characteristics of more than one research design (Bryman & Bell 2011, p.60), and obviously, a pre-dominant cross-sectional study may also identify idiographic features, especially since studies very often include more than one research question.

Another way to distinguish between case studies or multiple-case studies and cross-sectional designs is by achieving clarity regarding the unit of analysis. The definition of the unit of analysis is equal to the definition of the case in a study (Yin 2013, p.30). Typical units of analyses in the social sciences are individuals, groups, organizations, or societies (Bryman & Bell 2011). As Yin states, however, the unit of analysis can be any event or entity (Yin 2013, p.30), such as business processes, programs, decisions, or software applications. *Paper 2*, for example, is a qualitative, exploratory study using semi-structured expert interviews as its data collection method. However, the unit of analysis, i.e., the case, is not the individual interviewed expert, but the development partner program of a single large software vendor, whose ecosystem represents the common context. *Paper 3* also uses semi-structured interviews to collect the data. In this circumstance, the selected interview candidates are the unit of analysis and originate from different contexts, i.e., multiple companies. Moreover, the research question is targeted at largely nomothetic results, i.e., what the barriers and drivers are for the adoption of an online channel for EAS (cf. Table 15 ).

**Table 15:** Equal methods, different research designs: case study versus cross-sectional design

| **Paper** | Paper 2 | Paper 3 |
|---|---|---|
| **Title** | Software Ecosystem Orchestration: The Perspective of Complementors | Evaluating the App-Store Model for Enterprise Application Software and Related Services |
| **Method** | semi-structured interviews, qualitative data analysis | semi-structured interviews, qualitative data analysis |
| **Unit of Analysis** | program; i.e., the development partner program of a single software vendor | Individuals; i.e., experts involved in the buying process of enterprise software |
| **Focus** | largely ideographic | largely nomothetic |
| **Research Design** | case study | cross-sectional |

Table 16 gives an overview of the papers using case study designs in this thesis and classifies them according to the introduced concepts of case study type, rationale, replication logic and the unit of analysis.

**Table 16:** Overview of papers using case study designs

| Paper | Title | Case Study Type | Rationale | Replication | Unit of Analysis |
|---|---|---|---|---|---|
| 1 | Platform-as-a-Service | multiple-, mini-case study | comparative, common | literal & theoretical | service offering; i.e., PaaS offering |
| 2 | Software Ecosystem Orchestration: The Perspective of Complementors | single-case study (with two embedded units of analysis) | revelatory, common | literal & theoretical | program; i.e., the development partner program of a single software vendor |
| 6 | Comparing Public and Internal Enterprise App Stores: A Qualitative Case Study. | multiple- / comparative-case study | comparative, common | literal & theoretical | application; i.e., internal and public enterprise app stores |
| 7 | App'ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications. | multiple-case study | comparative, common, unusual / extreme, variety | literal & theoretical | Application; i.e., big data business application |

*Paper 1* studies PaaS as a business model and market offering. Therefore, two cases are studied in addition to a literature review to derive the theory (in particular the PaaS technology stack, roles in the PaaS ecosystem, and the revenue model of PaaS). Since the two studied cases represent two different variants of PaaS (pure PaaS and application-based PaaS) they are used equally for literal and theoretical replication. The addition of "mini" was added since the scope, in terms of breadth and depth of the studied cases, was limited, and the research process was simplified compared to the other case studies in this thesis.

*Paper 2* studies the development partner program in a proprietary SECO. Since the unit of analysis is the development partner program of a single software vendor, i.e., SAP, the case study is classified as a single case study with one case study context. However, the program studied comes in two variants, one variant for partners developing mobile EAS and one variant for partners developing cloud-based EAS. According to Yin, these two variants can be seen as two embedded units of analysis of a single case and can therefore be used for literal and theoretical replication (Yin 2013, p.46). Furthermore, data from within proprietary real-world SECOs, especially in the EAS segment, are difficult to obtain and hardly studied (Manikas & Hansen 2013). The rationale of the study can therefore be classified as revelatory. Since SAP is one of the largest vendors for EAS the rationale is also to identify common patterns in the domain of EAS.

*Paper 6* examines two app stores for EAS, one internal app store and one public app store. The rationale for this multiple-case study is a comparative one to distinguish these two types of app stores, but also to present the common features and capabilities of each type. The two cases are used for both literal and theoretical replication.

*Paper 7* studies eight different enterprise applications and identifies measures to make EAS products "ready" for sales via app stores. The individual cases have been selected to uncover common and unique features and measures. Moreover the multiple cases are used to contrast (comparative) individual aspects, but especially to increase the variety of measures.

In general the research process of the case studies (as in most empirical studies) can be structured in four phases: research design, data collection, data analysis, and the composition phase (Yin 2013, p.5). All case studies in this thesis are inductive, hence the composition phase is mainly a theory building phase and includes the preparation of the case description. However, the actual research processes of the four case studies in this thesis were individually designed to target the research objectives.

### 3.4.1 Research Process Paper 1

*Paper 1 "Platform as a Service (PaaS)"* (cf. Chapter II.1) was published as a "catchword" article in the journal Business & Information Systems Engineering. The catchword section is designed for emerging and trending topics in the area of IS (AIS 2015). Typically, the papers are shorter and have a stronger relation to topics originating from practice than other sections in this journal. Therefore, the research process was designed to explicitly serve the purpose of this type of paper.

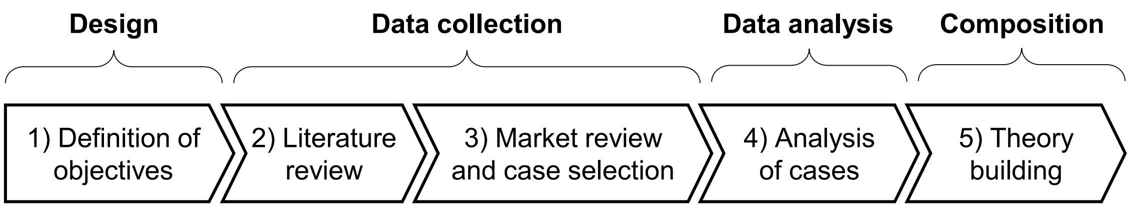


**Figure 32:** Research process of Paper 1

First the objectives of the research were defined. In particular they included the exploration of the elements of PaaS, its variants, the different stakeholders and roles as well as the identification of IT foundations enabling this independent business model and market offering. The literature review was restricted to the few available works at that time and provided a definition and basic understanding of the topic. Subsequently, we did a market analysis to identify the most important PaaS providers. The works from business analysts served as a starting point (Dubey et al. 2008) and were complemented by our own online investigations and recommendations from our professional network in the EAS industry.

During the data analysis phase we evaluated individual PaaS offerings by reviewing the publically available online material and created a comparison based on important PaaS characteristics[1] (see the overview of the reviewed PaaS offerings in Appendix B) (e.g., type of PaaS offering, revenue model, application domain). Two examples from this detailed comparison were then studied in more detail, one pure PaaS and one application-based PaaS (aPaaS). The cases have been selected for common yet different approaches to PaaS. Based on the provider comparison, the detailed review of the two selected cases, and the literature review, we then developed our concept which included three elements: the PaaS stack, a role model of the PaaS ecosystem, and the PaaS revenue model. The results were also questioned and reviewed by industry experts prior to publication.

### 3.4.2 Research Process Paper 2

The research process of *Paper 2 "Software Ecosystem Orchestration: The Perspective of Complementors"* (cf. Chapter II.1) was guided by GT. In short GT can be defined as "[…] the discovery of theory from data – systematically obtained and analyzed […]" (Glaser & Strauss 2012, p.1). Hence, GT is underpinned by two main principles. First, theory is developed bottom-up, solely from the retrieved data, without using a priori defined hypotheses. Second, the approach is iterative, meaning data analysis, data collection, and theory building are conducted either in multiple rounds or in parallel, re-enforcing each other (Bryman & Bell 2011, p.576). GT is therefore well suited for the exploratory, inductive research objective, and moreover also very well suited for the revelatory rationale of the case study at hand.

Corbin and Strauss propose eleven procedures which should be considered when designing a GT-based research process (Corbin & Strauss 1990):

1) data collection and analysis are interrelated processes.
2) concepts are the basic unit of analysis.
3) categories must be developed and related.
4) sampling in GT proceeds on theoretical grounds.
5) analysis makes use of constant comparisons.
6) patterns and variations must be accounted for.
7) process must be built into the theory.
8) writing theoretical memos is an integral part of doing GT.

[1] This comparison was initially published in the form of an online appendix (http://isdl.uni-bamberg.de/paas/appendix.pdf. Unfortunately, the online appendix is no longer accessible. Therefore the PaaS provider overview has been updated and can be reviewed in Appendix B.

9) hypotheses about relationships among categories should be developed and verified as much as possible during the research process.

10) a grounded theorist need not work alone.

11) broader structural conditions must be analyzed.

In the following I will walk through the research process as presented in Figure 33 and reference the procedures applied using the following notation: "*GT procedure 1*", "*GT procedure 2*" etc.

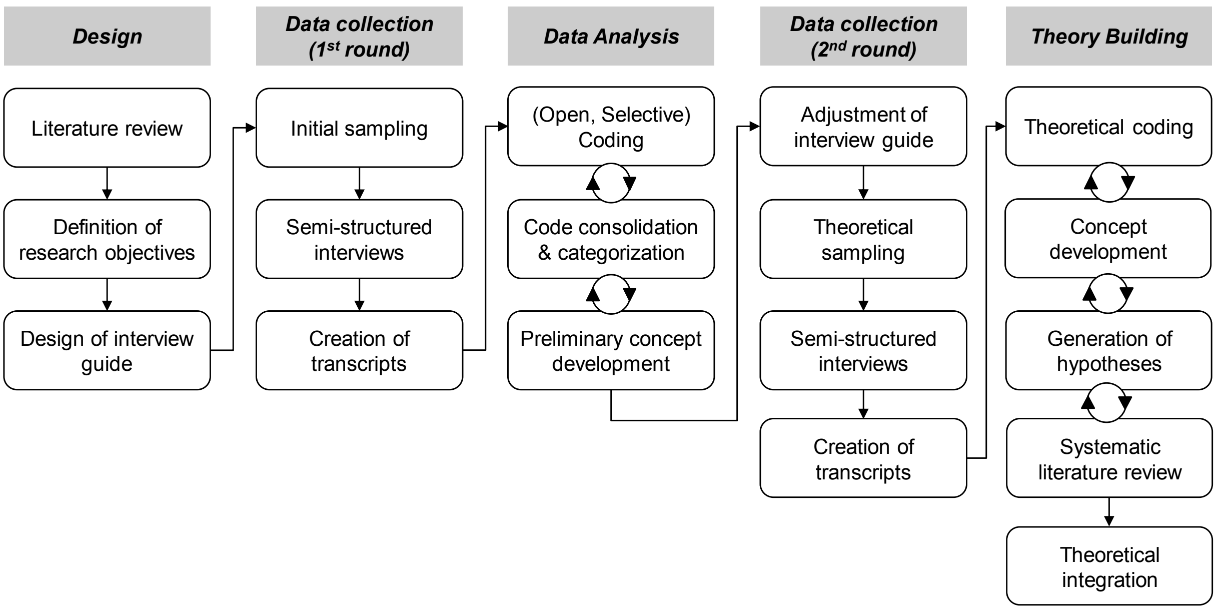


**Figure 33:** Research process of Paper 2 (simplified).

In the design phase we conducted a preliminary literature review to detail the research objectives and to define the initial semi-structured interview guide. It should be mentioned, however, that the literature at this point was not used to generate initial hypotheses or to help identify concepts but only to focus the research and to structure the interview. The introductory part of the guide begins with profile questions about the person and her or his company. The main part uses a simple model of the development partner lifecycle with four stages as defined by the software vendor's partner program ("Get Started", "Become a Partner", "Build", and "Publish & Sell"). This lifecycle model allowed us to make use of a concept already known to the interview candidates and helped us avoid misunderstandings by serving as common anchor points. The interview guide was tested with a partner manager of the software vendor who had played a leading role in the design of the development partner program from the beginning.

As previously mentioned, data collection and data analysis are done iteratively in GT. In our research process we implemented this aspect in two ways (*GT procedure 1*). First, the data collection was carried out in two rounds following the principles of theoretical sampling (cf. Chapter 3.3.2; *GT procedure 4*). Second, within the super-ordinate two rounds, data collection and data analysis were conducted in parallel, i.e., we started to analyze the interview once the first transcripts were available and could therefore already react on early findings in subsequent interviews by e.g., adding additional questions or stressing certain aspects. For the first round of interviews we reviewed an extract of the partner database of SAP and applied multiple selection criteria to ensure the candidate sample was appropriate with regards to their properties and variety (see principles of theoretical sampling; Corbin & Strauss 1990). The criteria included:

- company is signed up in the researched developer program (property aspect);
- company has already developed an application or is currently developing an application as part of the program (property aspect);
- interview candidate is fully knowledgeable about the entire program lifecycle (property aspect);
- company origin (variety aspect);
- company size (variety aspect);
- job role (variety aspect).

Since the investigated developer program was only officially launched approximately six months prior to our study, the companies fulfilling our criteria were somewhat limited. For the first interview round we were able to identify 20 candidates, i.e., companies with multiple contacts each. We approached the desired persons via e-mail, and contacts from eight companies agreed to an interview (four additional people in the second round). The interviews were done over the phone, either in English or German. The candidates were guaranteed anonymity. The interviews lasted between 40 and 60 minutes each, were fully transcribed by the researchers, and accounted for 480 minutes of recordings and 123 pages of transcript (for both rounds or 12 interviews in total).

Coding is at the heart of data analyses in GT. We applied the three coding techniques (*GT procedure 2)* as proposed by Urquhart: open coding, selective coding, and theoretical coding (Urquhart 2013, pp.9–10). Initially open coding was applied to the first four interview transcripts line-by-line to transform the raw data into analyzable data. Thereafter the open codes were consolidated and categorized to more abstract concepts (*GT procedure 3*). These concepts included the category process (*GT procedure 7*), tools, theoretical concepts and the core concept ("perceived helpfulness to achieve own goals"). Interviews 5-8 were then selectively coded with the major target to reach saturation. However, through constant comparisons (*GT procedure 5*) and the iterative nature of the process new open codes were also included, categories were revisited, and in case previous patterns proved to be wrong, codes, code categories, or concepts were adjusted, dropped, or rearranged (*GT procedure 6*). Previously coded transcripts had to be re-coded accordingly. The coding, conceptualization, and categorization process was accompanied with memo writing (*GT procedure 8*) to keep track of initial ideas, categories, tendencies, patterns, and also to support communication among the researchers. From a tool perspective, the coding process, memo writing, and analysis was supported by the CAQDAS-tool Atlas.ti and the mind map application Mindjet. At least two of the three participating researchers closely collaborated in all phases of the research, and in particular during coding and data analysis the participating researchers reinforced each other or challenged individual interpretations to avoid subjective biases (*GT procedure 10*).

The second round of data collection included four additional interviews (eight were invited) and followed the same procedures as the initial round. The interview guide was adjusted for this round to put more focus on the concepts of interest. For example, in this round we emphasized the aspect of interrelations between the concepts. Moreover the adjusted guide included more questions on the partnership in general (such as expectations, motivation, and targets of a partner with the development program), hence analyzing the broader surrounding and environmental conditions (*GT procedure 11*).

Theoretical coding was the first step of theory building. It is targeted at uncovering relationships between concepts (*GT procedure 9*) and integrating the concepts to develop a conceptual framework to combine the individual findings in one theory (Urquhart et al. 2010). To support this process the memos and mind maps created in the previous steps were used to extract a hypothesis and iteratively develop the final framework. A general recommendation in qualitative research is to use multiple sources of evidence as a form of triangulation to counteract the risk of subjective biases (Dubé & Paré 2003). Therefore, have in addition to the GT-based process, we conducted a systematic literature review. Urquhart also strongly suggests relating the framework developed using GT with the existing literature (Urquhart 2013, p.136 ff.). The literature review followed the recommendations by Webster and Watson (Webster & Watson 2002) and included the following sources:

- top ten IS journals according to the “MIS Journal Ranking” (AIS 2013a)
- selected journals of the “Senior Scholars’ Basket of Journals” (AIS 2013b)
- major IS conferences
- Thomson Reuters Web of Knowledge
- EBSCOhost databases.

We used 21 keywords and conducted a title and abstract screening to identify the relevant research papers. Works from three fields of research were then reviewed: SECOs, platform economy, and classic alliances. The literature was subsequently used to discuss the developed framework in relation to existing theories. This review, for example, confirmed that the lifecycle view which is one important concept in the developed framework is not researched at all in SECO literature, and complements the state of knowledge. Moreover, it confirmed identified relationships and concepts, and was also used to adjust the terminology and order of the framework's representation. In total 12 top IS journals and two databases were searched using the keywords.

Last, we discussed our results, and particularly the framework, in a focus group of seven experts from the SAP's partner management, who were all closely involved with the partner development program. The focus group was used as a form of "reality check", to gauge whether the findings were compatible with practical day-to-day experiences, but also to see if the findings and the developed framework could be used to better analyze issues and improve the program in the future.

### 3.4.3 Research Process Paper 6

Research *paper 6 "App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores"* is available in two versions. The first one was published at the 5th International Conference on Software Business in 2014. Due to the rigid page limitations of the conference template, I prepared an extended version of the paper which is published with this thesis (cf. Chapter IV.1).

Paper 6 is a qualitative study using inductive logic. Three research objectives have been postulated, whereas two follow an exploratory and one a design-oriented purpose:

- Which scenarios are supported by public and which by internal enterprise app stores (exploratory)?
- How can enterprise app stores leverage the opportunities of IT consumerization while avoiding its drawbacks, such as shadow IT? How do these models support the targets of IT governance (exploratory)?
- How can both public and internal enterprise app stores be used in combination (design-oriented)?

The combination of behavioral methods with design science methods is typical in IS research. Huysmans and De Bruyn evaluated mixed methods approaches and have defined the different types mixing design science with empirical and behavioral methods (Huysmans & De Bruyn 2013). Applying their findings, this research can be classified as "BEHAVIORAL → design". "Behavioral" here refers to qualitative or quantitative empirical research and "design" refers to design-oriented research. The upper case determines which element is dominant or defines the overarching "theoretical drive" of the study (in this paper this is the qualitative research process). The arrow determines the direction of the sequence. This also means that the design element is based on the result of the preceding behavioral, qualitative results. When assessing the rigor of this study the focus should be on the dominant element, i.e., the qualitative part. (Huysmans & De Bruyn 2013).

The case study examines two cases, i.e., two different app store types: the internal and public enterprise app store. Because the individual cases are investigated with two objectives, first to describe the systems' major capabilities as each represents a typical instance of its class, and second, with the objective of contrasting them, the case study rationale is both common and comparative (cf. Table 16).

To achieve the threefold objective, the following methods have mainly been used: review of public documents, expert interviews, evaluation of application systems, and model creation and analysis. The research process can be structured in four phases as depicted in Figure 34.

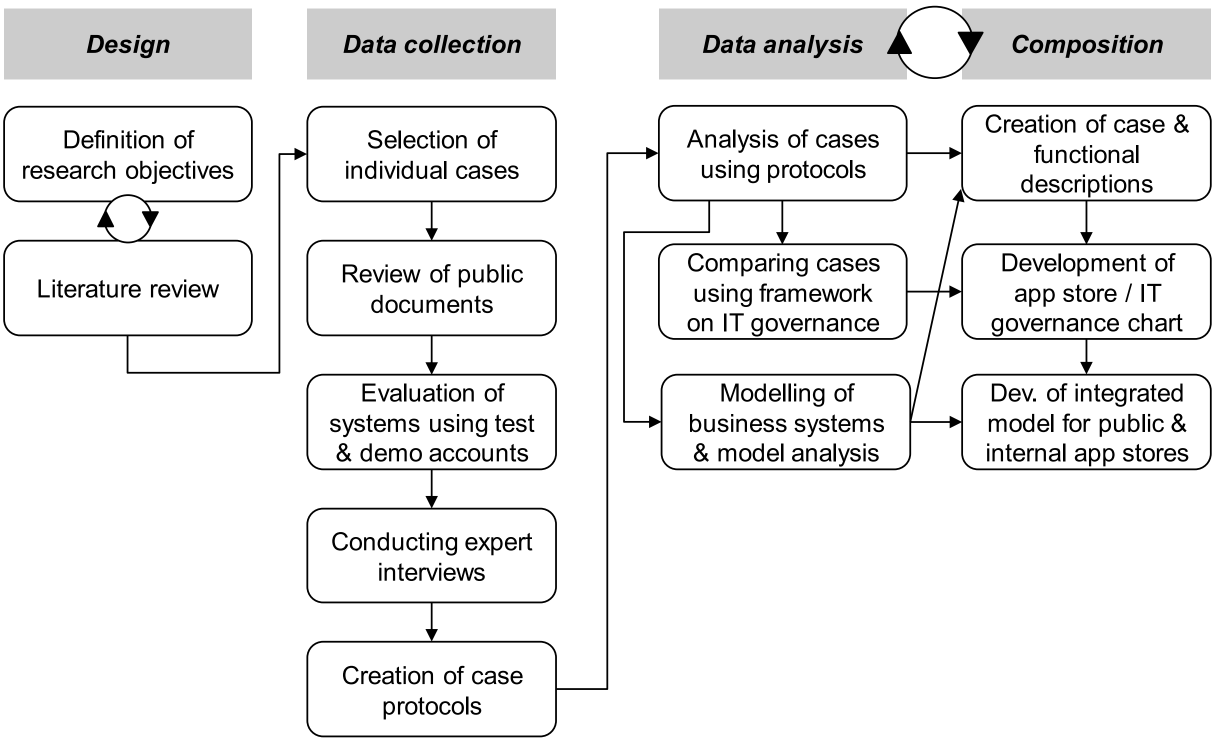


**Figure 34:** Research process of paper 6

In the design phase, the definition of the research objectives was made in close alignment with the available literature on enterprise app stores. In particular, the literature has revealed the importance of enterprise app stores in the context of IT governance (and the trend of IT consumerization) which influenced the overall direction of the paper, and thus one additional research objective was added.

In order to identify the relevant cases for the study, market research was conducted and a shortlist of available internal and public app stores created. The possible cases in the shortlist were reviewed and assessed based on their suitability for the research objectives. The shortlist including the results of the assessment are available in Appendix C.

SAP has been chosen as a common organizational context for both cases since it fulfilled several criteria:

- SAP is one of the largest EAS vendors and its systems are very representative instances in their respective class.
- SAP operates a public enterprise app store, the SAP Store, and offers an internal enterprise app store, the SAP Enterprise Store.
- SAP owns multiple software and development platforms and provides standardized, packaged PaaS offerings, allowing ISVs to develop and market their own enterprise applications.
- SAP has MDM and MAM as part of their enterprise mobility portfolio.
- We had direct access to experts within SAP and in the SAP ecosystem working with the two app store systems
- We could get access to both application systems.
- There was sufficient publically available documentation.

To collect the relevant data, we first reviewed the publically available documentation. Subsequently we were provided with system access: a test account to the public SAP Store and access to a demo system of the SAP Enterprise Store. This allowed us to analyze the respective functionality hands-on. Moreover, we were able to interview two product managers at the SAP Store and SAP Enterprise Store. The interviews focused on confirming assumptions and discussing topics which were beyond what the public documentation and system evaluation could reveal. The interviews lasted approximately 45 minutes, were conducted in person and notes were taken. The findings from all three data sources (documentation, system review, and interview) were consolidated in two case study protocols that were designed to record the data in an un-interpreted manner yet follow similar structures in order to make comparisons easier in the data analysis phase.

The data analysis phase and the composition phase were not conducted sequentially, but iteratively, mainly due to the fact that three research results can be delimited. The data analysis started with the review and evaluation of the case study protocols and this analysis also served as a basis for applying the IT governance model from the literature to the findings. Moreover the case analysis and protocols were also used to develop business system models for each of the cases and analyze these models to gain additional knowledge.

The respective business systems were modeled and analyzed using the Semantic Object Model (SOM), in particular, the so-called interaction schema (IAS) as part of the SOM framework. A detailed description of the SOM framework can be reviewed in (Ferstl & Sinz 1997; Ferstl & Sinz 2013, p.194 ff.). In short, the IAS can be used to structurally analyze the business system in focus, i.e., the business objects and business processes within a business system. Business processes again are investigated with regards to their service relationships among business objects and with regards to the coordination mechanisms of business objects. The term 'service' in this context refers to actual services, products, or payments. SOM delimits four types of coordination: the non-hierarchical "negotiation principle", the hierarchical "feedback-control principle", and the hierarchical and non-hierarchical "coordination via targets" (Ferstl & Sinz 1997; Ferstl & Sinz 2013, p.205 ff.).

The analysis of the case protocols and the business system models developed were combined in the case descriptions which consisted of three elements respectively: First, the business system models, with all the relevant business objects and transactions among them. Second, the catalogs of the app stores were analyzed (which enterprise apps are supported or included). Third, a detailed functional analysis was provided on the two app stores.

The second result is a chart showing the potential influence of the respective app store on the corporate IT governance. This was derived by applying the governance model from the literature to the functional descriptions and the business system models.

Last, an IAS model was developed that integrated the two use cases of internal and public enterprise app stores. The objectives of the integration were to maximize the potential benefits w/r to the IT governance objectives. The developed model was reviewed regarding its feasibility and value-add with the product managers interviewed beforehand. The creation of this artefact followed the principles of design-oriented research as outlined by Becker (Becker 2010) and the constructivist model view as previously introduced in Chapter 3.3.3 (see also Figure 29 and Figure 30).

### 3.4.4 Research Process Paper 7

*Paper 7 "App'ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications"* (cf. Chapter V.1) is the last research artefact in this thesis. It is a qualitative, inductive study with exploratory purpose, and uses a multiple-case study design and provides a focus on ideographic and comparative results by studying business applications (the unit of analysis) within the subdomain of analytical 'big data business applications'. The purpose of the study is to research business applications and their readiness for the app store model and by doing so identify methods and measures which can be applied to other applications. The specific domain has been chosen since it represents a new breed of applications without a long history. Whereas traditional business applications, such as ERP systems, were tailored to the traditional B2B sales model, new applications should recognize new go-to-market models as the digital sales and distribution model proposed by the app store model. The common organizational case study context was SAP. As one of the largest vendors for enterprise software SAP fulfilled four important criteria for conducting the envisioned research. Hence, SAP

- owns an enterprise app store, the SAP Store,
- provides, a software platform and infrastructure to develop and operate big data business applications, the SAP HANA cloud platform,
- released a number of HANA-based big data business applications already, and
- also has partners releasing HANA-based big data applications via the application developer partner program.

The multiple-case study rationale is to uncover common, comparative but also unique (i.e., unusual/extreme) measures to improve the app store readiness. The replication logic follows both a literal and a theoretical approach. The research process is structured in four phases as illustrated in Figure 35.

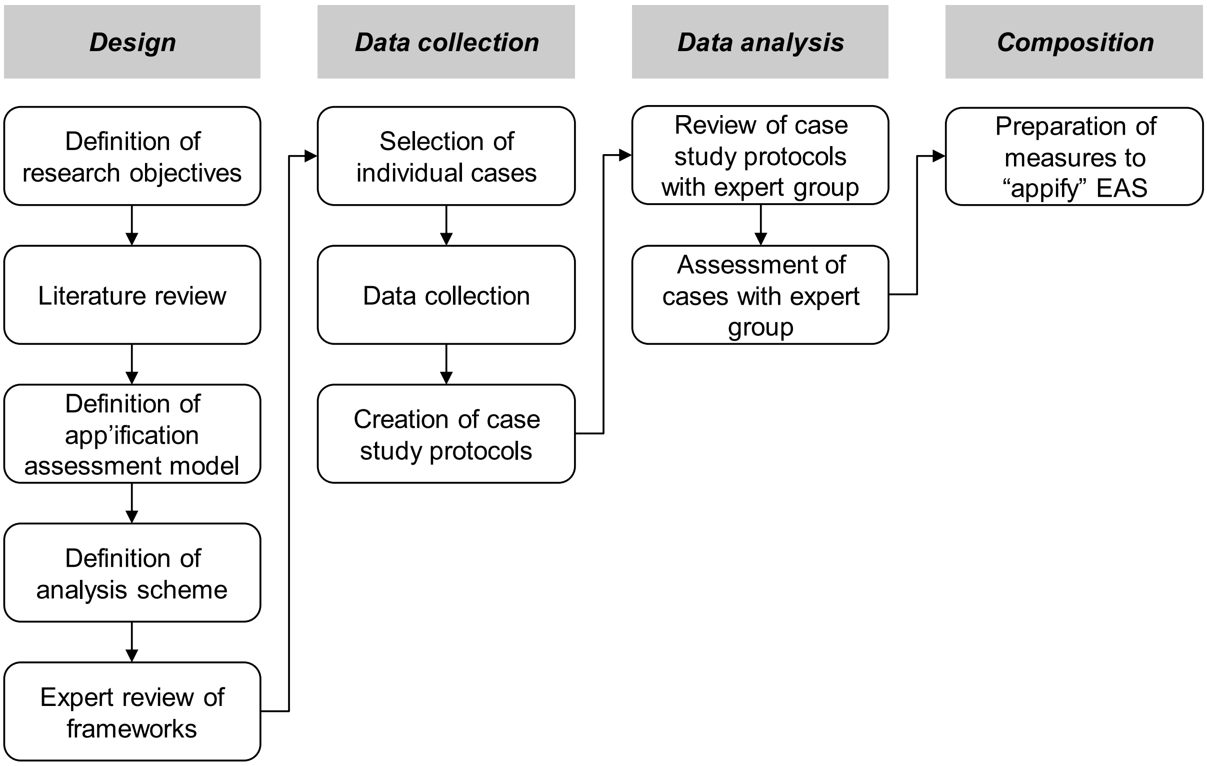


**Figure 35:** Research process paper 7

In the *design phase*, after the definition of the research objectives, a literature review was conducted to identify studies within the research field at hand. Since no study could be identified which would address the postulated research objectives directly, the literature review focused on adjacent research providing elements supporting the analysis. In particular, two frameworks were derived from literature supporting the research process (see Figure 36). First, an assessment model was derived based on which the readiness of a business application could be assessed using seven criteria, criteria that was based mainly on the works in Part III of this thesis, on studies from technology-acceptance and adoption theory (see Chapter 2.7) and the works of Verville and Halingten on the buying process of enterprise software (see Chapter 2.4.2) (Verville & Halingten 2003). To ensure the multiple case studies could be replicated and the data captured comprehensively, a standardized analysis scheme was developed. Furthermore, the analysis scheme ensured that the data collected for the individual cases could be compared by unifying the different terminologies, levels of information, and variety due to different data sources. Last, this approach also ensured that the researcher's subjectivity was limited. The analysis scheme was created based on the business model definition by Stähler (Stähler 2002) and adopted to the purpose of this study. Hence the scheme entails three views with sub-categories in each view (in total 14 subcategories).

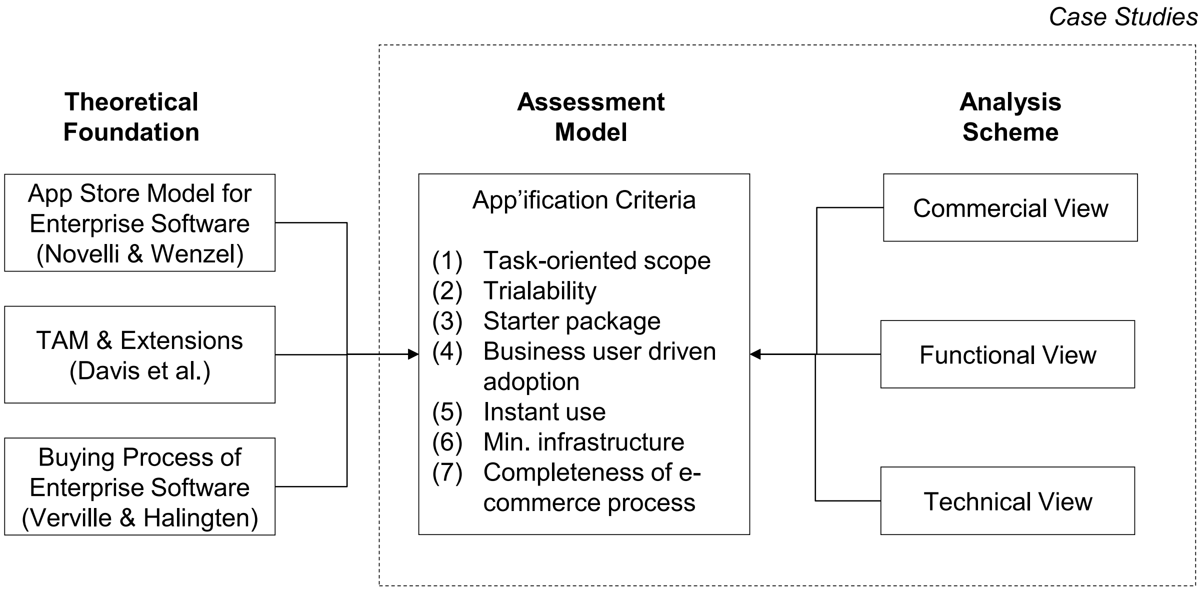


**Figure 36:** Frameworks used in Paper 7

The subcategories were also related to the individual 'app'ification criteria' in the assessment model to support the assessment of individual application characteristics and their influence on the app store readiness. The design phase was concluded with a review of the postulated frameworks in an expert group consisting of five people, incl. two product managers, a software architect, a sales manager, and an e-commerce specialist.

The *data collection phase* starts with the selection of the individual which was carried out in two sequential steps. First a shortlist had to be created. The shortlist was created by applying three criteria: the application is published on the SAP Store and/or the SAP HANA Marketplace (a department of the SAP Store focusing on HANA-based applications), the application classifies as big data, real-time analytical or high-performance applications, and the application is a proper business application and not just an infrastructure product or toolkit for developers.

In total 1400 solutions available on the SAP Store and the HANA Marketplace were reduced to 36 that met the three criteria. The 36 shortlisted applications were screened in more detail to consider the following aspects: applications are typical instances for their kind (representative aspect), applications represent the breadth of big data applications (variety aspect), and, last, sufficient publically available material was available to retrieve the necessary data. Ultimately five applications were selected (four from SAP, one by an SAP partner). The data for the five cases was collected mainly by using multiple different online resources (e.g., the SAP Store, the SAP help portal, SAP community network). The data was recorded in case protocols which were structured according to the three views and 14 subcategories of the analysis scheme. In total, 56 pages of case protocols were created.

The *data analysis* was carried out in two consecutive workshops (~120 minutes each) with the expert group introduced in the design phase. First the individual cases were introduced and reviewed with the group to ensure the data captured was comprehensive and interpretative tasks (e.g., categorizations and early conclusions) corresponded with the common understanding of the group. In a second workshop, the assessment of the individual cases using the assessment model and the case protocols was carried out. The results were directly transcribed during the workshop.

In a last step the workshop results were prepared in a comprehensive overview on measures to make the application app-store-ready.

### 3.4.5 Ensuring Quality in Case Study Research

As outlined above, the presented case studies each have a different purpose and a very specific research process. Their common denominator is the exploratory and inductive nature. The academic discussion on how to best assess the rigor of qualitative research as introduced in Chapter 3.3.4 also largely applies to qualitative case study designs. It is, however, even more commonly accepted that the traditional criteria originating from the quantitative tradition are not suitable for case studies and ought to be adjusted accordingly (Bryman & Bell 2011, p.61).

Yin proposes four tests in order to assess the quality of case study research: the construct validity, internal validity, external validity, and reliability. Moreover Yin proposes multiple tactics to improve the rigor of case studies in these four categories (Yin 2013, p.33). The tactics proposed by Yin have been reviewed and enhanced by Gibbert and Ruigrok (Gibbert & Ruigrok 2010). Table 17 lists Yin's four quality criteria, the tactics he, Gibbert and Ruigrok propose, and the phase of the research process in which the tactic should be best applied. Moreover the table indicates which of the tactics have been used in which case study paper.

**Table 17:** Tactics to improve quality of case studies (derived from Yin 2013, p.33; Gibbert & Ruigrok 2010)

| Criteria | Tactic | Phase | Paper 1 | Paper 2 | Paper 6 | Paper 7 |
|---|---|---|---|---|---|---|
| Construct validity | Use multiple sources of evidence / triangulation | Data collection | (✓) | ✓ | ✓ | ✓ |
| | Establish chain of evidence | Data collection | (✓) | ✓ | ✓ | ✓ |
| | Have key informants review draft case study report | Com-position | ✓ | ✓ | ✓ | ✓ |
| Internal validity | Do pattern matching | Data analysis | ✓ | ✓ | × | × |
| | Do explanation building | Data analysis | × | ✓ | × | × |
| | Do time-series analysis | Data analysis | × | × | × | × |
| | Formulate a clear research framework | Design | × | ✓ | ✓ | ✓ |
| | Apply theoretical triangulation | Multiple | × | ✓ | ✓ | ✓ |
| External validity | Use replication logic in multiple-case studies | Design | ✓ | ✓ | ✓ | ✓ |
| | Provide clear rationale for case selection | Data collection | ✓ | ✓ | ✓ | ✓ |
| Reliability | Detailing the research process and procedures | Design | ✓ | ✓ | ✓ | ✓ |
| | Use case study protocol | Data collection | ✓ | ✓ | ✓ | ✓ |
| | Use case study database | Data collection | (✓) | ✓ | × | × |

*Construct validity* describes the extent to which a study researches what it claims. Since Yin's epistemological view is rather positivistic, his view on construct validity is also closely linked with the ideal of full objectivity (Gibbert & Ruigrok 2010). As stated earlier, the epistemological view of this study is largely constructivist (see Chapter 3.3.3 & 3.3.4), hence the target pursued with construct validity is instead to reduce subjective influences as much as possible. Three tactics are proposed by Yin. First, it is proposed that triangulation be applied during data collection that relies on more than one source of data or data type. In all of the case studies we have applied some form of triangulation (e.g., expert interviews + documentation, or different groups of interviewees, different sources for documentation). The second tactic is referred to as "establish chain of evidence," to let third parties reconstruct how the researchers made their conclusions from raw data to theoretical conclusions (Gibbert & Ruigrok 2010). We have applied this by thoroughly detailing the research process, either directly in the respective paper or with complementary explanations and material in this introductory paper. Gibbert and Ruigrok further stress the importance of making the data collection process explicit (Gibbert & Ruigrok 2010) – to satisfy this need we have very comprehensively described how the individual cases have been selected and which criteria have been applied to shortlist and ultimately select the cases (e.g., Appendix B and Appendix C). Last we have always referenced intermediate constructs or models and used well accepted data analysis procedures (e.g., coding, Grounded Theory) and made use of data analysis software, i.e., CAQDAS, were appropriate to support the chain of evidence as proposed by Gibbert and Ruigrok (Gibbert & Ruigrok 2010). Since our research was embedded in the organizational context of a large software vendor, we made extensive use of the third proposed tactic and used expert input and reviews as much as possible – not only in the composition phase of the research but at multiple points including the design phase, data collection and data analysis.

To ensure the second criteria, *internal validity*, in total five tactics were identified. Yin stresses that internal validity from his epistemological view is mainly relevant to explanatory research (Yin 2013, p.47). Still, we believe that exploratory research should also make an effort to ensure the results provided 'can be trusted' and the conclusions made are on solid grounds. The first tactic, pattern matching, proposes to compare the findings of the case study with findings from previous studies or findings from studies with different contexts (Gibbert & Ruigrok 2010). Since all studies have a rather exploratory nature, existing studies which provided patterns to compare with were limited, except for Paper 1 and Paper 2 where we matched our findings with findings from the literature in the areas were studies were available. In general, we believe that pattern matching is not very well suited for exploratory research with a novel and complex socio-technological context. Explanation building was partly applied in Paper 2 as part of the Grounded Theory process (iteratively developing the concepts and relationships). It has to be noted that Yin states that this tactic is mainly relevant to explanatory case studies (Yin 2013, p.47). Time series analysis was not chosen as a tactic in any of our studies, since we did not intend to study the change of phenomena through time, and we believe that this tactic is not appropriate given our research objectives. The fourth tactic is to formulate a clear research framework which is then strictly followed. We have done this in three papers (Paper 2: Grounded Theory, Paper 6: IT governance framework and SOM model analysis, Paper 7: analysis scheme and assessment Model). Last, theoretical triangulation is proposed as a tactic, which means applying "different theoretical lenses or bodies of literature", either during data collection, analysis or in interpreting the results (Gibbert & Ruigrok 2010). Paper 2 triangulated by applying the findings generated via Grounded theory and the ones identified via literature research. In paper 7, theoretical triangulation was an inherent part of the methodology since the research framework included the analysis scheme and assessment model which integrated different lenses. Paper 6 also investigated the cases from different standpoints, namely from both the business system view and the IT governance view.

*External validity* (generalizability) in qualitative research and especially for case studies is a field of intense discussions (Bryman & Bell 2011, p.61; Gummesson 2007). Instead of applying the meaning of statistical generalizability, which means generalization of observations of a population, the literature proposes applying analytical generalizability, i.e., the process of observation to theory (Gibbert & Ruigrok 2010; Eisenhardt 1989). To improve studies in this regard two tactics were proposed, applying replication logic and providing a clear rational for the case selection. All case study designs have been designed to include some form of literal or theoretical replication. Either via multiple-case study designs (Paper 1 and 7), comparative design (Paper 6), or by having so-called embedded cases (Paper 2) (Yin 2013, p.46). The second tactic has also been applied throughout the case studies. Paper 2 applies theoretical sampling as part of Grounded Theory and for Paper 1, 6, and 7 the case selection process, criteria and rational have been explicitly and comprehensively provided.

The last quality criteria proposed is *reliability* (especially external reliability, i.e., possibility of replicating results). Since most qualitative and also most case studies have a low external reliability, the target of this criteria is to provide as much transparency as possible to allow others to assess the research process on consistency and conceptual or logical errors. Therefore, we put a lot of effort in all the case studies to comprehensively document the research process and applied methods, either in the published paper directly or due to page limitations in the publishing templates, via complementary material in this introductory paper. Next, for all case studies case protocols were used which also formed the basis for case analysis and case descriptions. Last, for Paper 2 a case database was used (via CAQDAS "Atlas.ti") which included all the interview transcripts, codings, and memos. For Paper 1 a small database was generated of all the evaluated PaaS offerings which stored the major characteristics of each PaaS offering.

## 3.5 Model-Based Integration of Research Results

The previous explanations in Chapter 3.2, 3.3, and 3.4 dealt with the methodology of the individual papers of this thesis, the empirical research artefacts. The Introductory paper and especially Chapter 5 "An Integrated Perspective on the App Store Model for Enterprise Application Software" go beyond a typical summary of the individual findings – instead Chapter 5 should be understood as a dedicated research artefact (IT or model artefact) representing the overall synthesis and integration of the individual research results using the business model framework and the structure of the value architecture as introduced in Chapter 1.1.2.

### 3.5.1 Methodological Classification

From a methodological point of view the model-based integration as an artefact can be understood as design-oriented research, and business models are among the well accepted artefacts in design-oriented IS research[1] (Frank 2012). The corresponding research process leading to the artefact corresponds with the overarching research process of this thesis and can be structured according to Becker's proposal on the ideal structure of design-oriented research in four phases: analysis, design, evaluation, and diffusion (Becker 2010). This overarching research process as illustrated in Figure 37 embraces the individual Papers 1-7 and the model-based integration (highlighted with a grey box). The four phases have a sequential direction, but have been conducted iteratively – both within the process of a single research paper and similarly across the individual ones. The research process moreover combines the mainly empirical, qualitative research with the design-oriented research in one consistent and coherent process where the empirical findings form the basis for design recommendations, which are again integrated into the artefact using a pre-defined business model framework (Huysmans & De Bruyn 2013). The integration of the empirical findings is facilitated by the app store model itself. The high-level business system model of the App Store Model for EAS which was introduced in Chapter 1.2 serves as a basis for deriving the research objectives resulting in the presented Papers 1-7.

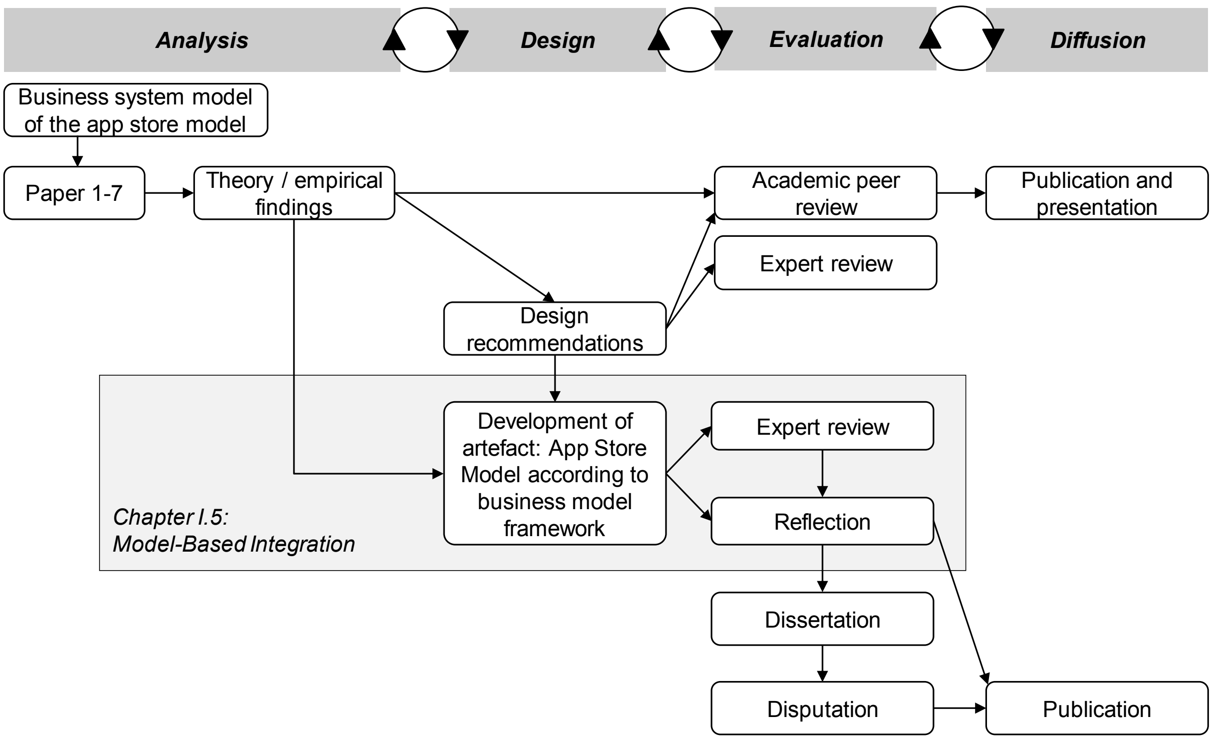


**Figure 37:** Research process leading to the model-based artefact (simplified)

The analysis phase of this research process is consequently covered by the analysis of the business system of EAS app stores and the individual research papers uncovering the

[1] In this context, IS research and the German *Wirtschaftsinformatik* are used synonymously, neglecting the historical and philosophical differences of the two scholars. For a comparison of the similarities and differences see (Wigand 2012; Robra-Bissantz 2012).

behavior and structure in this socio-technological environment using empirical qualitative and largely exploratory research. The results of these studies are theories describing the phenomena of the App Store Model for EAS in its various aspects, e.g., influencing factors of a given problem. In most papers the consequences of the theoretical findings have been discussed and where appropriate design recommendations (e.g., process recommendations or system recommendations) given. These recommendations were promptly presented to and discussed with experts within SAP, the EAS vendor, which provided the organizational grounding for a large portion of this research.

Design recommendations have been assigned in this model to the design phase according to Becker's classification, since they already represent a first step of transforming empirical findings via goal-oriented interpretation into usable constructs. An important role in the evaluation of a design-oriented research process, according to Becker, is academic assessment (peer review) (Becker 2010). In our case, the empirical papers were peer-reviewed and thereafter published and presented to the academic community (in the case of conference papers).

The construction of the artefact itself can be understood as a construction problem. The construction problem is interpreted as a special instance of a general research problem following the definition by Ferstl (Ferstl 1979, p.43 ff. see also Sinz 2010). Thus, a research problem is defined as follows:

- A research situation consists of a research problem and research procedure.
- A research problem is defined by research goals and a research object.
- By applying appropriate research procedures research results are generated.

Accordingly, the construction problem is defined in the following way:

- A construction problem is a special research problem.
- The research object is a non-existing system.
- The behavior of the system is postulated.
- The research goal intends to find a system structure which implements the required system behavior.
- The research result is a solution space which may entail multiple options to implement the structure to achieve the behavior.
- Solution space may be constrained by certain structural provisions or limitations.

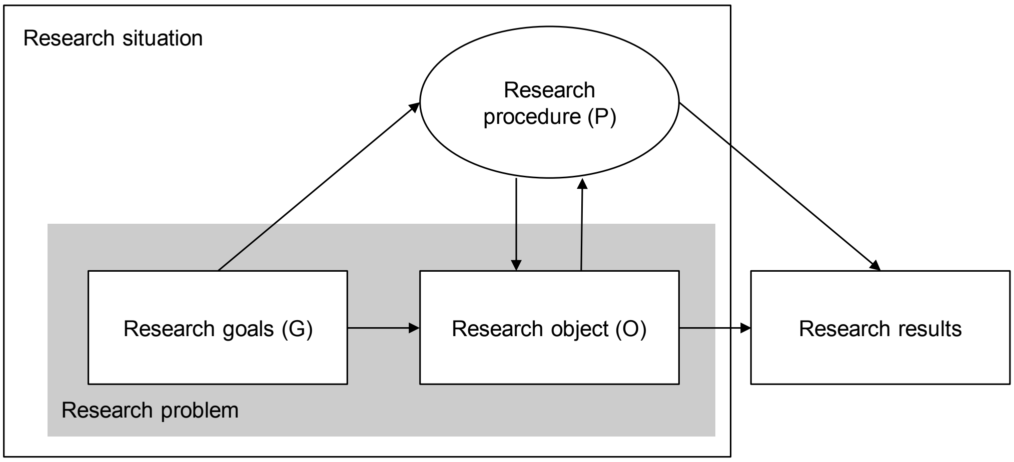


**Figure 38:** Research situation according to Ferstl (Ferstl 1979, p.43 ff.)

While the research object, in our case the business system, does not fully exist in its targeted state, parts of it have already been implemented and have also changed over time and throughout this research. Hence, the claim that the structure of the business system is not available must be limited. Instead it can be said that the targeted structure is not fully implemented but parts have already been realized in many different software ecosystems. To cater for this, the concept of the research situation will be adapted to Ferstl's "model-based research situation," as depicted in Figure 39 (Ferstl 1979, p.80).

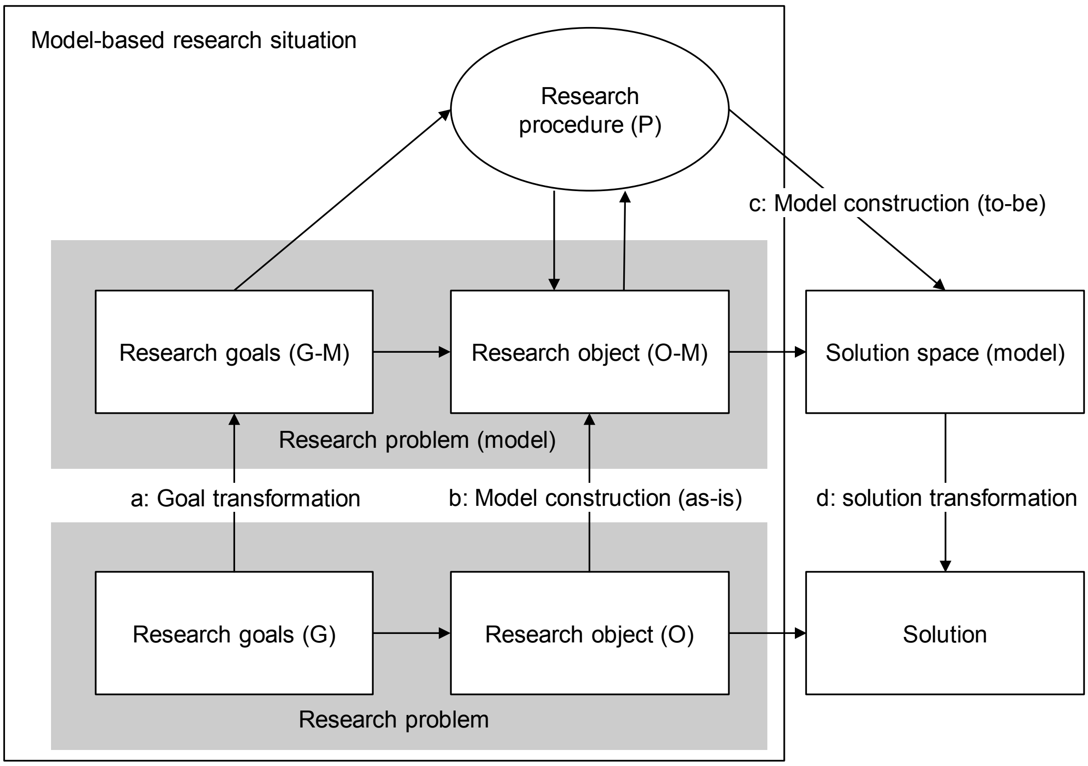


**Figure 39:** Model-based research situation according to Ferstl (Ferstl 1979, p.80)

Applying the definition of a construction problem and the model-based research situation framework to the creation of the app store model artefact we can conclude:

- The research object (O) is represented by multiple business systems as they exist and with incomplete or inadequate structure and behavior.
- The construction of the app store model cannot be done on the so-called object level (real world). Hence, models (in the widest sense) have to be created (b) to research the existing structures and to-be behaviors of the business system (as-is models). The exploratory findings and theories generated by Paper 1-7 represent the research object (O-M) on model level.
- The goals on object level (G) are also transformed (a) into research goals on the model level (G-M) and consist of constructing a model artefact representing the solution space and the “to-be” state. The solution space is a model describing the possible structures which, if implemented, can provide the value propositions postulated for the respective actors of the app store model (value proposition is part of the business model definition and hence part of the app store model).
- The research procedure interprets and transforms (c) the individual empirical theories and findings (Paper 1-7) into a common model framework based on the business model framework and architecture of value creation as introduced in Chapter 1.1.2.
- The solution space (model), the App Store Model for EAS, does not represent an “as-is” model, but a “to-be” model (Sinz 2010). Moreover, it represents a solution space covering multiple options to implement the business system structure complying with the given behavior.
- In order to transfer the solution on model level into reality (object level), the solution again has to be transformed (d). This transformation happens during implementation where the guidance from the model needs to be interpreted and adapted to the real-world context.

### 3.5.2 Structure of Model Artefact

The artefact created is first introduced with a definition of “app stores for EAS” and “the App Store model for EAS” including several assumptions which constrain and guide the systematical derivation of the model elements.

The business model structure with its three elements, the value proposition, value architecture, and revenue model, is used to structure the model artefact (Figure 40). The value architecture which represents the major section is further broken down in three parts: the supply-side value chain, the demand-side value chain, and the EAS product characteristics part.

The supply-side and demand-side value chain elements are structured on two levels, the task level and the resource level. The task level analysis presents the major business processes in the form of transactions between business objects. As the foundation of the analysis and representation of the task-level, the SOM methodology is applied. In particular, the structure oriented IAS is used to present the business processes. A short introduction to this methodology is given in Chapter 3.4.3. For a comprehensive introduction please refer to the works from Ferstl and Sinz (Ferstl & Sinz 1997; Ferstl & Sinz 2013, p.194 ff.). For the detailed transaction analysis between the platform provider and the ISV or the software customer respectively this thesis foregoes the graphical representation of an IAS. Instead the transactions are presented in a tabular overview.

The resource level analysis consists of structured function and capability lists of the identified personal and application system resources. The task and resource views are integrated by assigning the respective resources to the individual transactions on the task level.

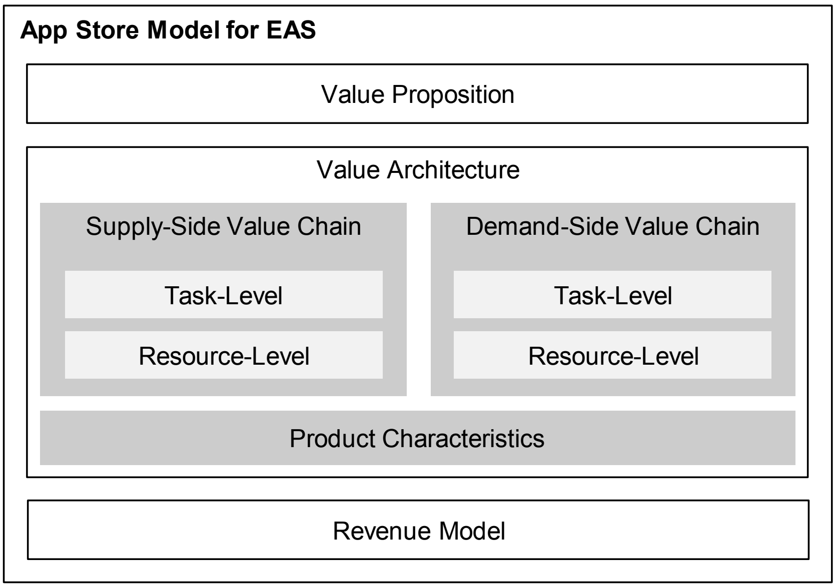


**Figure 40:** Structure of the model artefact

### 3.5.3 Interpretation of Results and Quality Assurance

It should, however, be mentioned that the transaction analysis using the IAS and the subsequent detailed transaction analysis in the tabular overview are not “fully complete and fully consistent”: in particular, not all transactions mentioned are disaggregated to the same level of detail, nor are coordinating mechanisms uncovered for every service transaction. For many “enforcing” transactions presented a concrete “contracting” or “initiating” transaction simply would not add any value since the overall transaction granularity is still too coarse or abstract. Furthermore, the graphical representation in the form of an IAS and the subsequent detailed tabular transaction analysis are on different aggregation levels and not all transactions in the tabular overview can be found in the IAS. The two views, the IAS and the tabular overview, pursue different goals. While the IAS represents the major actors, business objects and the major transactions between the objects, the tabular overview of the transactions provides a structured access to the detailed level where a graphical representation of all transactions would simply not have been very readable within the boundaries of this thesis template.

The limitations on consistency and completeness equally apply to the resource level. The presented functions and capabilities focus on the identified critical aspects of the business processes. Basic requirements which would also apply to any other e-commerce system or application system (e.g., multi-language or mobile user interface) have not been the focus of the analysis. Moreover, only those elements have been included in the models which had either proper backing in the empirical findings and theories of the seven research papers, which could be simply derived from the same, or are backed by findings from the relevant literature. Literature however was only used in rare instances where the papers in this thesis could not provide enough substance to cover the model elements.

Consequently, the models presented should only be understood as a starting point or frame which is used to derive a concrete model for a particular instance in the real world, and in favor of simplicity and clarity certain aspects have been skipped. Following the argument set out by Frank, the research result can also be described using his concept of a “constructed possible world”. This world includes IT artefacts, i.e., the app store and the software platform, and the corresponding “system of action”, such as process models (Frank 2009). The goal of the app store model is to guide implementation in order to reach the objectives of the respective actors (platform provider, ISV, and software customer). The advantage of this approach is to describe interrelations and dependencies between organizational and technological issues beyond the borders of an individual research artefact. However, this constructed world does not exist and cannot be simulated due to its social-technological complexity (Frank 2009). Therefore, as a whole, it cannot be validated or invalidated using established empirical methods, and instead should be seen as a result of inductive conceptualization providing a framework to derive design templates in the researched domain (Wilde et al. 2007). As Frank argues, the construction of possible worlds cannot always be assessed using a claim for truth, since these worlds are constructed with objectives in mind (Frank 2009). The business model concept itself provides an element for the construction objectives: the value proposition. Therefore, the contents of the App Store Model for EAS should best be evaluated in relation to their “appropriateness” and “usefulness” in reaching the postulated objectives (A. R. Hevner et al. 2004; Becker & Pfeiffer 2006; Frank 2009). The usefulness and appropriateness is assessed by confronting and discussing the results with selected experts and by reflecting on the results with regards to the value propositions and to the practicability of the model elements. The results of the expert review and the reflection are presented in Chapter 5.6.

Ultimately the dissertation evaluation and disputation of this thesis can also be seen as further steps in evaluating the artefact following the argument set out by Becker (Becker 2010) and succeeded by the publication of the results (cf. Figure 37).

# 4 Main Research Results of Individual Research Papers

Beyond this introductory paper, the cumulative dissertation includes seven individual research papers. While Chapter 1 holistically motivates the thesis and its research objectives and provides an overview of the structure, Chapter 2 lays the foundations by introducing relevant fields of research and their major theories and concepts. Next, Chapter 3 presents very comprehensively the research methodology used. The current Chapter consequently focuses on the major results of the individual research papers without repeating the methodological, foundational or motivational aspects. Thereafter, Chapter 5 uses the individual results to integrate them according to the previously defined business model framework and derives a standalone IT or model artefact.

The individual papers will be presented in the sequence, and structure as introduced in Chapter 1.3:

- supply-side value chain: Papers 1 & 2
- demand-side value chain: Papers 3, 4, 5
- app store for EAS as an application system: Paper 6
- product characteristics of EAS for app stores: Paper 7.

## 4.1 Supply-Side Value Chain

The supply-side value chain cluster contains two papers (Paper 1 & 2) and researches the business processes (incl. process phases, service transactions, supporting IT systems and tools, and organizational aspects) between ISVs and the platform provider. In particular, the following research questions will be addressed (cf. Chapter 1.2):

- Why do ISVs cooperate with platform providers? (RQ_a.1)
- What are the key elements of the platform service? (RQ_a.2)
- How do ISVs define their value creation process within the SECO? (RQ_a.3)
- What are the platform provider's key enablers for supporting ISVs in reaching their goals? (RQ_a.4)

### 4.1.1 Paper 1: Platform-as-a-Service (PaaS)

Though Paper 1 can be assigned to the supply-side value chain cluster of the app store model's business system, it also lays the foundation of the overall thesis. This is because it provides some fundamental definitions and frameworks which serve as a basis for the app store model as a whole and the subsequent papers.

First, PaaS is introduced as an independent service and business model in the cloud stack, combining hardware and software technologies to provide other companies, i.e., ISVs, with the means to develop and provide SaaS solutions. The technologies are bundled and offered as a service.

Two types of PaaS can be differentiated: the pure PaaS and the application-based PaaS. The application-based PaaS is defined around a core application (e.g., ERP application) which allows ISVs to enhance this core product with add-ons. The pure PaaS is not based on a core application and is a standalone service. The applications built with pure PaaS are consequently mostly not add-ons but standalone applications.

Three actor roles are differentiated in the PaaS ecosystem[1]: the PaaS provider, the ISV, and the software customer. The PaaS provider owns the platform and offers the platform service as a dedicated product to the market. In the case of aPaaS, the PaaS provider is also a software application firm. ISVs develop, operate, and market applications (either add-ons or standalone) using the PaaS offering. Last, the software customers buy and consume the software applications; often it remains invisible to software customers and software users that applications are built and operated in a PaaS environment.

Next, the PaaS stack is introduced differentiating the core components of PaaS (or aPaaS) and value-added services (see Figure 41).

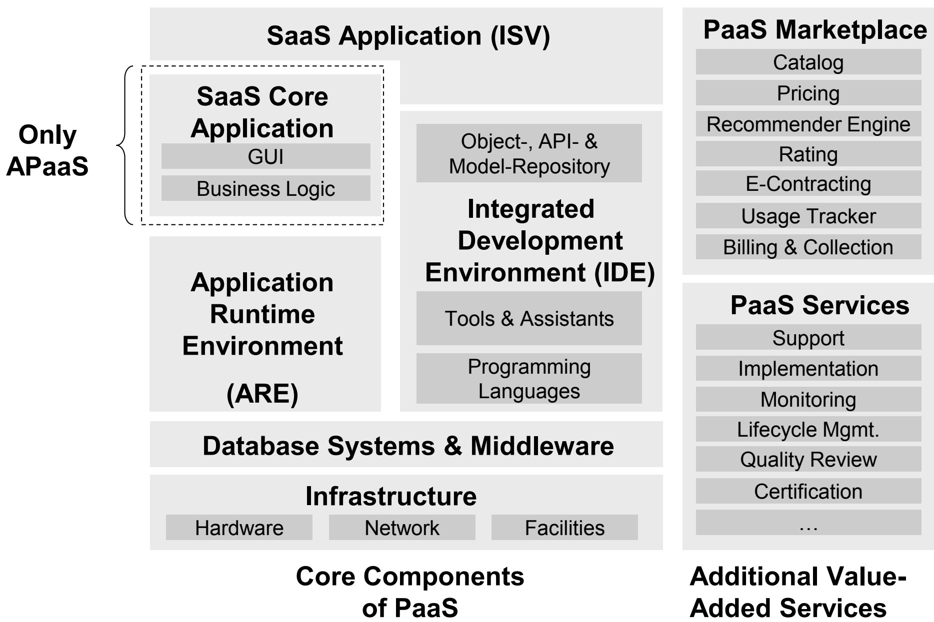


**Figure 41:** The PaaS stack (Paper 1) (developed from Dubey et al. 2008; Gillett et al. 2008)

[1] PaaS ecosystem is a special instance of a software platform ecosystem as introduced in paragraph 2.3.

Core components include: an infrastructure, the database and required middleware, the application runtime environment (ARE), the integrated development environment (IDE) and in case of aPaaS, the core application. Additional services can be part of the service offering. A central one is the PaaS marketplace which is equivalent to the App Store for EAS in this thesis. Moreover, a multitude of additional services might be provided to attract ISVs and differentiate them from other PaaS offerings, or to ensure certain standards on the platform.

Last, the paper introduces monetization, service, and support options for PaaS and the applications built with it. In the case of the aPaaS, the PaaS provides the core application and incl. the respective add-on to the customer. In addition, it provides support services. The PaaS provider receives money from the software customer for the core application and a revenue share from the ISV for the add-on (cf. Figure 42).

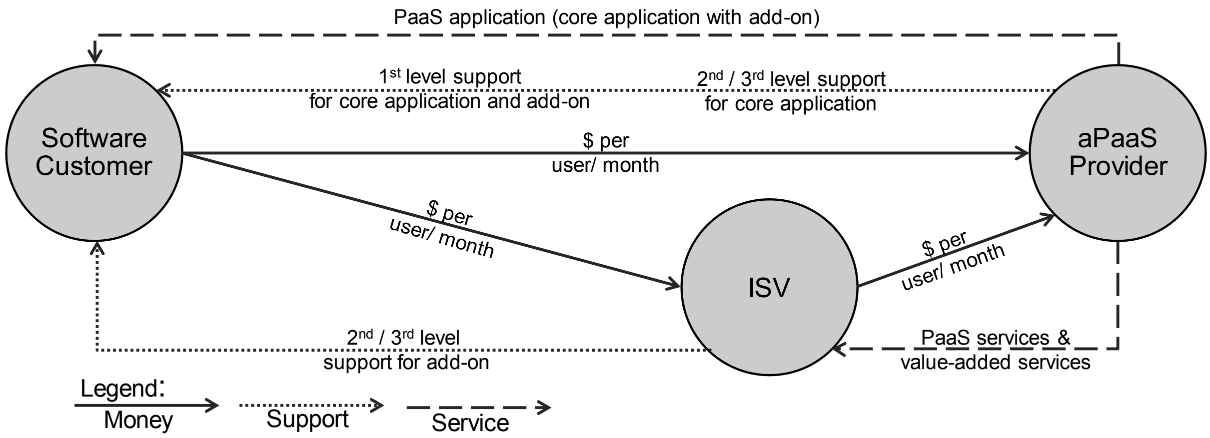


**Figure 42:** Flows of money and services in the aPaaS scenario (Paper 1)

Beyond the basic definitions of a platform-based ecosystem with the respective actors, the paper addresses the research question RQ_a.2, the key elements of the platform service.

As a closing remark on this paper, at the time of its creation we classified PaaS as standardized platform services for SaaS applications only. In accordance for example with Giessmann and Legner, we interpret PaaS more broadly in that it covers all kinds of applications incl. SaaS apps, traditional on-Premise offerings, mobile apps and any combination of these variants (Giessmann & Legner 2013).

#### 4.1.2 Paper 2: Software Ecosystem Orchestration: The Perspective of Complementors

Paper 2 enhances Paper 1 on the supply-side by addressing three research questions:

- Why do ISVs[1] cooperate with platform providers? (RQ_a.1)
- How do ISVs define their value creation process as part of a software ecosystem? (RQ_a.3)
- What are the platform's key enablers for supporting ISVs in reaching their goals? (RQ_a.4)

All three research questions are addressed by a unified partner framework which represents the integrated result of the paper as shown in Figure 43. It is referred to as a partner management framework since ISVs are partners from the platform provider's perspective and the framework can be used as a foundation to shape, manage, and adjust a platform provider's ecosystem partner management.

The framework includes all developed concepts: the *goals* of the ISVs pursued as part of the software platform ecosystem, the conceptual *enablers* to reach these goals, and the *instruments* which are concrete implementations of the enablers. Moreover, it includes the *effects* describing the ISV's perception of how well the platform and the partnership help to achieve the goals. The effects and additional *influencers* in turn ultimately influence the *intentions*, i.e., whether an ISV intends to partner with a platform provider or intends to continue an existing partnership. Moreover these concepts take effect along the ISVs lifecycle in the partnership. Ultimately, with the *intents* element it also represents an adoption model explaining why ISVs partner with a platform provider or join the respective platform ecosystem.

[1] In the paper, the role "ISV" is denoted as "complementor" and the role "platform provider" as "platform sponsor". To maintain consistency with the terminology of this introductory paper the names of the respective roles have been adjusted. The meaning of the roles however is not affected, since they are perceived as synonyms in the scope of this thesis.

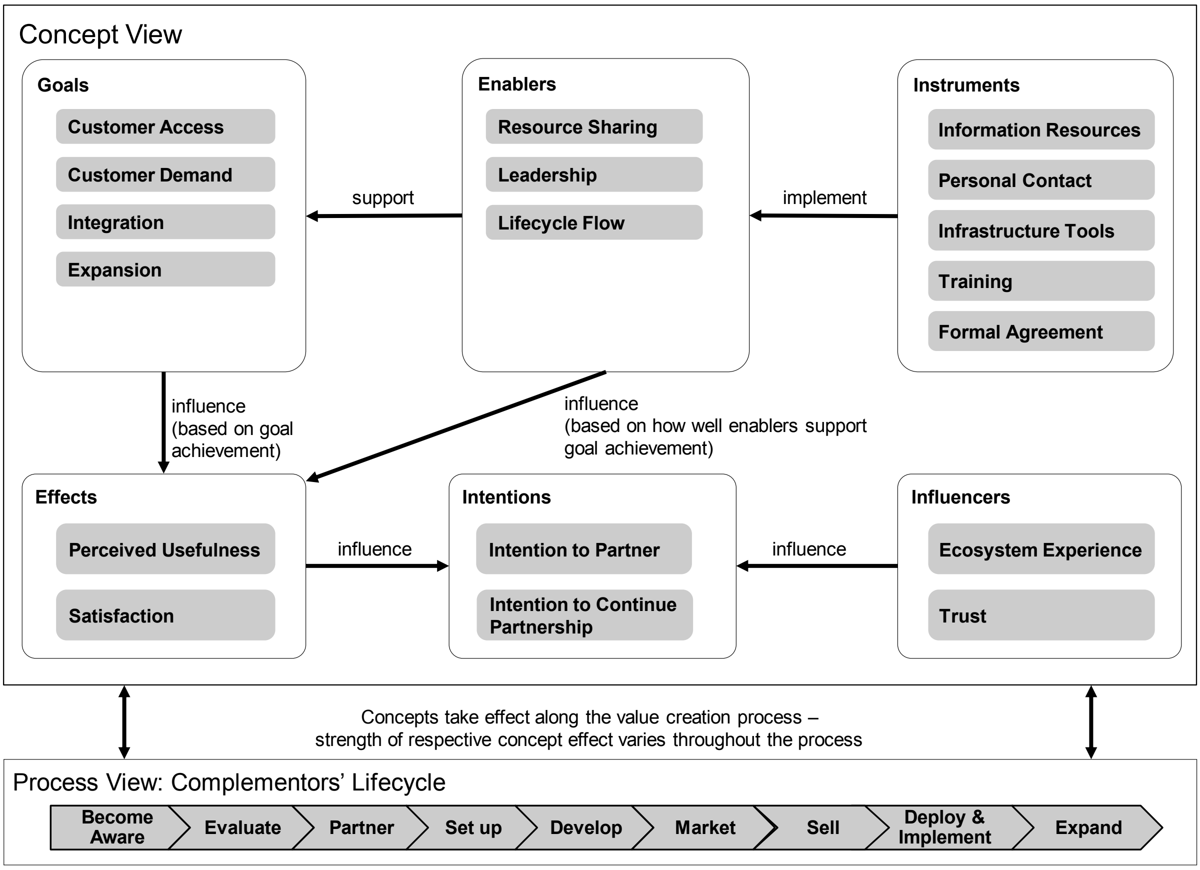


**Figure 43:** Partner management framework (Paper 2)

Four goals could be extracted from the data. The first goal pursued is customer access, and in particular, access to those customers who are not part of the existing customer base or who are hard to reach by increasing their visibility and trustworthiness via the partnership to the platform provider. The second goal is to better meet customer demand by following the guidance of the platform provider (e.g., technological guidance) and by improving their capabilities with the platform (e.g., quality, time-to-market, ease-of-use). The integration goal is mainly related to the core application of a platform provider. To be able to develop enhancements to platform core products ISVs often prefer or require a partnership as part of a program with the platform provider. Last, ISVs pursue an expansion of their existing business into new segments (e.g., new functional areas, new industries, or new business models after all).

The platform provider can support the ISV in achieving its goals by using so-called enablers. The enablers are implemented by concrete instruments, and one concrete instrument might be assigned to more than one enabler. The first enabler, resource sharing, includes everything from information resources, infrastructure or personal resources. The second enabler, leadership by the platform provider, refers to the future of the platform including technological roadmap, coverage of customer segments as well as go-to-market strategies. Leadership defines the outer boundaries for ISVs in which they define their sub-segment and niche product. Third, the lifecycle flow, is an enabler that supports the ISV with respective tools throughout the lifecycle, from development to testing, marketing, sales, and support. Important instruments supporting this enabler are the development environment and the app store. The five instruments in Figure 43 are only the instrument categories. For a full list of all the instruments identified see Appendix C.

The enabler (and with them also the instruments) together with the goals influence the effects. The effects have two variables, the perceived usefulness, which refers to the degree to which the ISV believes the partnership will support him in achieving his goals, and satisfaction which refers to the actual target achievement and his assessment of the partnership's contribution. The effects in turn influence the partner's decision to join a platform or continue the platform partnership. Moreover, two additional concepts (influencers) haven an effect on the partner's intention toward the partnership decision: trust and experience with the platform or its provider.

The partner lifecycle as a separate construct introduces nine different stages which describe the co-determined business process between the ISV and the platform provider. First, prospective ISVs are attracted by a platform offering (*become aware*), then they *evaluate* the platform offering. In case they decide to join the platform, contractual *partnering* has to be carried out. Afterwards the ISVs need to setup the development environment as well as the organizational necessities (e.g., contact person, support staff). The *development* stage ends with a certification which approves the application and checks multiple quality criteria. In the *marketing* and *selling* stage an application is offered to the market and sales transactions are conducted ideally using the app store platform. After the application is sold it needs to be *deployed and implemented* at the customer. The *expand* stage is an option to use the platform in other ways than originally intended, add additional technology pillars or use the platform to access new markets. Different stages of the lifecycle are repetitively conducted, whereas others might only be conducted once.

## 4.2 Demand-Side Value Chain

This section consists of three consecutive papers (Papers 3, 4, and 5) which respectively enhance the results of its predecessor. In particular, they look at the app store as an online sales channel (from the platform provider's and ISV's perspective) or buying channel (from the software customer's point of view). The following research questions have been postulated in the introduction to be answered by this research cluster (cf. Chapter 1.2):

- What is the online buying process of an EAS customer? (RQ_b.1)
- What are the drivers of and barriers to adopting an online channel for EAS acquisitions? (RQ_b.2)

- Which technical and organizational measures can be applied to overcome the adoption barriers of the online channel? (RQ_b.3)
- How can online sales channels for EAS be introduced and co-exist with existing, traditional “offline” sales channels? (RQ_b.4)

### 4.2.1 Paper 3: Evaluating the App-Store Model for Enterprise Application Software and Related Services

The target of Paper 3 is to identify barriers and drivers as determinants for the adoption of an online channel for EAS. As part of the methodology and the interview guide two constructs have been developed, the EAS portfolio and the software buying process phases. While on the one hand, these constructs are instruments of the research method in the scope of this paper, they are also important (intermediate) results directly addressing parts of the postulated research questions. Moreover, they are fundamental for the subsequent analysis and the following papers in this cluster.

The EAS portfolio classifies different types of enterprise software products and related IT services and includes the categories: core solutions (line of business solution, suite solution), on-top solutions (mobile, add-ons), usage enhancements (user licenses, usage contingences, service level agreement enhancements), and IT services (see Chapter 3.3.1 and Figure 28).

The buying process derived from the literature and validated by experts consists of five phases (see Chapter 3.3.1 and Figure 27):

- *Problem recognition:* The customer gains awareness of an opportunity, need or threat which can be dealt with by acquiring an EAS product.
- *Information search:* The customer acquires information material about all potential providers and solutions.
- *Evaluation of alternatives:* The customer evaluates the individual solutions, e.g., by ranking selected solutions and providers using pre-defined criteria.
- *Negotiation & purchase:* The customer finalizes the terms of the transaction, stipulates contracts, and executes the purchase.
- *After sales:* The customer purchases additional goods related to already acquired solutions (e.g., service level enhancements, additional licenses).

The determinants for the adoption of an online channel for EAS have been looked at from multiple perspectives. The determinants – irrespective of whether they are a barrier or a driver of the adoption – which could be extracted from the data are listed in Table 18.

**Table 18:** Determinants influencing the software customer adoption of the online channel

| Category | Determinant |
|---|---|
| Solution | ▪ solution complexity<br>▪ solution criticality<br>▪ solution evaluability<br>▪ solution scope<br>▪ solution pricing<br>▪ solution specificity<br>▪ integration & implementation effort<br>▪ solution deployment<br>▪ number of end-users |
| Relationship / (Transaction) | ▪ provider's reputation & trustworthiness<br>▪ existing relationships with the provider<br>▪ contracts in place with the provider |
| Software customer | ▪ prior experience with solution<br>▪ prior experience with channel<br>▪ IT competences<br>▪ IT governance & procurement<br>▪ innovativeness |

The determinants can be grouped into three categories, the solution-related, relationship-related[1], and (software) customer-related factors. These factors have then been qualitatively analyzed along the five-phased buying process, resulting in concrete barriers and drivers of adopting an online channel for buying EAS in a dedicated phase of the buying process. While some determinants had an effect as a barrier or driver throughout the buying process (e.g., existing relationship with the provider) others were only relevant for a single or a few phases (e.g., high volumes of end-users as barrier was only relevant in the negotiation and purchase phase). While the qualitative tables generated can directly act as design recommendations to improve the technological or organizational means in order to improve the online channel adoption, some aggregate conclusions can be drawn as well.

The problem recognition, information search and after-sales phases seem to be highly suitable for an online channel, whereas the evaluation and the negotiation and purchase phases only seem to work for simple solutions. From the product portfolio point of view core solutions seem to better suite personal offline sales, while on-top solutions and usage enhancements seem to be a good fit for the online channel. The results for services were mixed and no clear tendency could be derived.

Ultimately, three areas of managerial implications and related recommendations were derived for platform providers or EAS vendors who want to establish an app store as an online sales channel: First, the role of the online channel as part of the greater sales system needs to be defined. For simple solutions the app store can act as a primary sales channel throughout the buying process, while for more complex products, the app store can serve as complementary

[1] In the subsequent papers, this category is named "transaction" related attributes.

channel to a personal sales channel. Second, EAS products have to be adjusted to better suite the online channel. Traditional EAS products are tailored for the personal sales channel. These are often characterized by a large scope, high complexity, on-premise deployments, and complex upfront pricing models, which all act as barriers to adoption. Third and last, traditional e-commerce capabilities of consumer oriented app stores need to be enhanced to meet the enterprises' unique requirements (e.g., corporate buying roles, or approval workflows).

The paper addresses the postulated research question RQ_b.1, by defining and detailing the online buying process. Moreover, it significantly contributes to RQ_b.2, by identifying the determinants of the adoption of an online channel and by analyzing their effect on the adoption along the buying process. The qualitative analysis and the derived managerial implications also contribute to RQ_b.3, since the described effects of the factors during the buying process can easily be translated into measures to overcome the barriers or to strengthen driver effects.

#### 4.2.2 Paper 4: Adoption of an Online Sales Channel and "Appification" in the Enterprise Application Software Market: A Qualitative Study.

The determinants for adoption of an online channel for EAS in Paper 3 have been further formalized in Paper 4. First, this is based on additional data and through continued qualitative data analysis and iterative conceptualization activities. The identified determinants in Paper 3 could be extended to be either clear drivers of or barriers to adoption. Some adjustments in naming, grouping as well as some additional determinants have been added.

The first major result of this paper is therefore the adoption model for an online channel in the EAS market as depicted in Figure 44. The figure shows a simple causal graph which groups the determinants in three groups, the customer attributes, the solution attributes, and the transaction attributes. Hence, the model includes the direct effects of the factors on the adoption decision and can be interpreted as follows: one incremental unit of the factor x, has negative (barrier) or positive (driver) effect on the intention to use the online channel for EAS buying. Moreover, the major indirect effects, i.e., the relationships between the factors have been considered in the model.

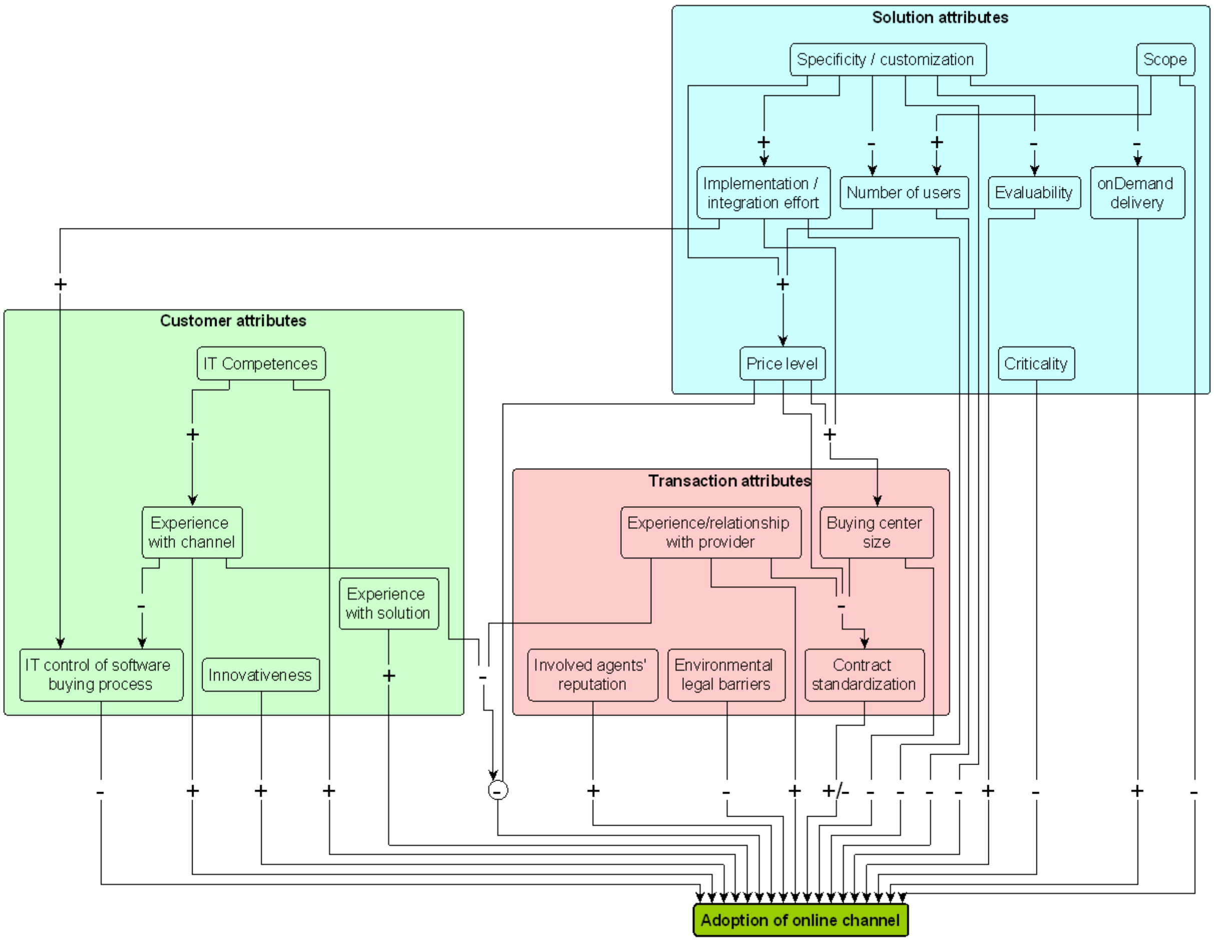


**Figure 44:** Adoption model for an online channel for EAS (Paper 4)

The solution attributes of the adoption model have then been applied to the EAS portfolio classification resulting in an assessment view allowing us to predict the suitability of the respective portfolio items for the online channel. Table 19 shows the result: the respective portfolio items are first valued with regards to the individual characteristics (either high, medium, low) and then whether the respective value has a positive or negative impact on the online adoption. Since the portfolio categories are still relatively coarse, not all factors can be assigned a clear value. For example, both core and on-top solutions can be either cloud or on-premise deployed applications. In case solutions characteristics have a clear tendency towards online suitability it can also be called “app’ified” to express its suitability towards the app store model. Hence the solution factors are also referred to as “app’ification criteria”. The prediction of the suitability of the portfolio items towards the online channel was tested by analyzing the available data with regards to direct causal links. The test confirmed the prediction except for the IT services. The direct causal analyzes resulted in a neutral result versus a negative result in the assessment via the solution factors – this difference may however also lie in the high variety of available IT services.

In conclusion, the further analysis undertaken in this paper revealed the high determinant impact of the solution factors on online adoption – not only directly but also indirectly. Hence, the managerial recommendations from Paper 3 can be repeated. At best, product designers and engineers should consider the targeted sales channel, and ensure app'ification criteria are already met in the early stages of a new product development in case the app store is the chosen means of sales and distribution. Moreover, Paper 4 now provides a toolset to both assess and improve the suitability of EAS for the app store model.

**Table 19:** EAS portfolio and app'ification attributes (Paper 4)

| | **Portfolio items** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Solution Factors** | **Core Sol.** | | **On-Top Sol.** | | **IT Services** | | **Usage Enhanc.** | |
| Criticality | high | (–) | low | (+) | mid/high | (–) | Low | (+) |
| Cloud (C) / OnPremise (OP) | C / OD | | C / OD | | N/A | | N/A | |
| Evaluability | low | (–) | high | (+) | low | (–) | High | (+) |
| Integr./impl. effort | high | (–) | mid | | N/A | | Low | (+) |
| Price level | high | (–) | low/mid | (+) | high | (–) | low/mid | (+) |
| Scope | high | (–) | low | (+) | low | (+) | N/A | |
| Specificity / customization | high | (–) | high | (–) | high | (–) | N/A | |
| Number of users | high | (–) | mid | | N/A | | N/A | |
| ***Overall tendency*** | | **(–)** | | **(+)** | | **(–)** | | **(+)** |

Paper 4 comprehensively addresses RQ_b.2, by providing the adoption model. Moreover, by mapping the solution factors to the EAS portfolio it further contributes to RQ_b.3 in that the suitability of certain EAS product categories are assessed. This tool expands the provider's scope of action in defining dedicated sales strategies for certain product categories and provides engineers a benchmark which they should already consider during the definition and development of new products.

### 4.2.3 Paper 5: Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity?

In addition to the previous two papers, Paper 5 includes the personal sales channel, i.e., the offline channel, in the analysis. All previously developed models and analysis are repeated with two adoption decisions for buying EAS, either using the online channel or the offline channel. Table 20 lists the factors, the description and their respective effect on the online or offline channel adoption. Not surprisingly, most factors have the opposite effect on the offline channel adoption compared to the effect on the online channel adoption, except for three cases which require a further detailed analysis since they may hint at the possibility that the factor acts at some other level, e.g., effect the software purchase in general and not the channel choice. These cases, however, also uncovered the strength of the qualitative research approach, since we identified well-grounded reasons for all three cases in the qualitative data:

- "The reputation and trustworthiness of the involved agents" drives the adoption of both channels and is therefore a determining factor of the vendor selection decision. However, the offline sales channel is also better suited for building trust and reputation as opposed to the online channel. Hence, a higher level of trust and reputation is required prior to conducting the buying process online versus the offline channel.
- While "prior experiences with the provider" might be beneficial for the online channel, personal contacts from these earlier relationships are the preferred means of communication, hence the offline channel is preferred in cases where relationships already exist.
- A certain level of "contract standardization" is a pre-requisite for the online channel, too much standardization may drive software customers towards the offline channel since they demand a certain level of customization.

**Table 20:** Factors influencing the channel adoption decision (Paper 5)

| Factor | | Description | Impact on… | |
|---|---|---|---|---|
| | | | **online adoption** | **offline adoption** |
| Solution attributes | Criticality | Importance of the supported business processes for the organization | Barrier | Driver |
| | Evaluability | Extent and ease of evaluating the solution relying on the online channel's capabilities | Driver | Barrier |
| | Implementation/integration effort | Effort (in terms of time and financial investments) needed to have the application wholly implemented and integrated with pre-existent systems as needed | Barrier | Driver |
| | On-demand delivery | Possibility to deliver the application on-demand | Driver | Barrier |
| | Price level | Price of the purchased application | Barrier | Driver |
| | Scope | Breadth and depth of the supported functionalities. | Barrier | Driver |
| | Specificity / customization | The degree to which the supported functionalities are peculiar to a specific organizational domain or need to be adapt to it | Barrier | Driver |
| | Number of end-users | End-users to which the application is delivered | Barrier | Driver |
| Customer attributes | Innovativeness | Customer's attitude towards innovation and technology | Driver | Barrier |
| | IT competences | Availability of in-house IT know-how and personnel | Driver | Barrier |
| | IT control over the buying process | Level of control exerted by the IT personnel on software purchase decisions | Barrier | Driver |
| | Prior experience with the online channel | Past experience with a similar channel | Driver | Barrier |
| | Prior experience with the solution | Past experience with a similar solution | Driver | Barrier |
| Transaction attributes | Involved agents' reputation | Reputation and trustworthiness of the involved agents (vendor, channel provider, etc.) | *Driver* | *Driver* |
| | Buying center size | Number of people playing a role in the software purchase decision | Barrier | Driver |
| | Prior experience and relationship with the provider | Past experience and pre-existent relationships with the channel provider | *Mixed* | *Driver* |
| | Contracts standardization | Level of standardization of the contracts formalizing the software purchase | *Mixed* | *Barrier* |
| | Online purchase legal barriers | Breadth and depth of environmental legal requirements to be fulfilled in the online software purchase | Barrier | Driver |

Paper 5 moreover analyzes the importance of the individual factors in the five phases of the buying process. The results, however, only give an indication, since the non-quantitative research methods do not provide statistically significant results.

Applying the process view and the portfolio view to the two channels results in the channel adoption profiles shown in Figure 44. It shows four panels indicating the software customer's preference towards the online or offline channel, in panel (a) across all solution types, in panel (b) for core solutions, panel (c) for on-top solutions, and panel (d) for IT services.

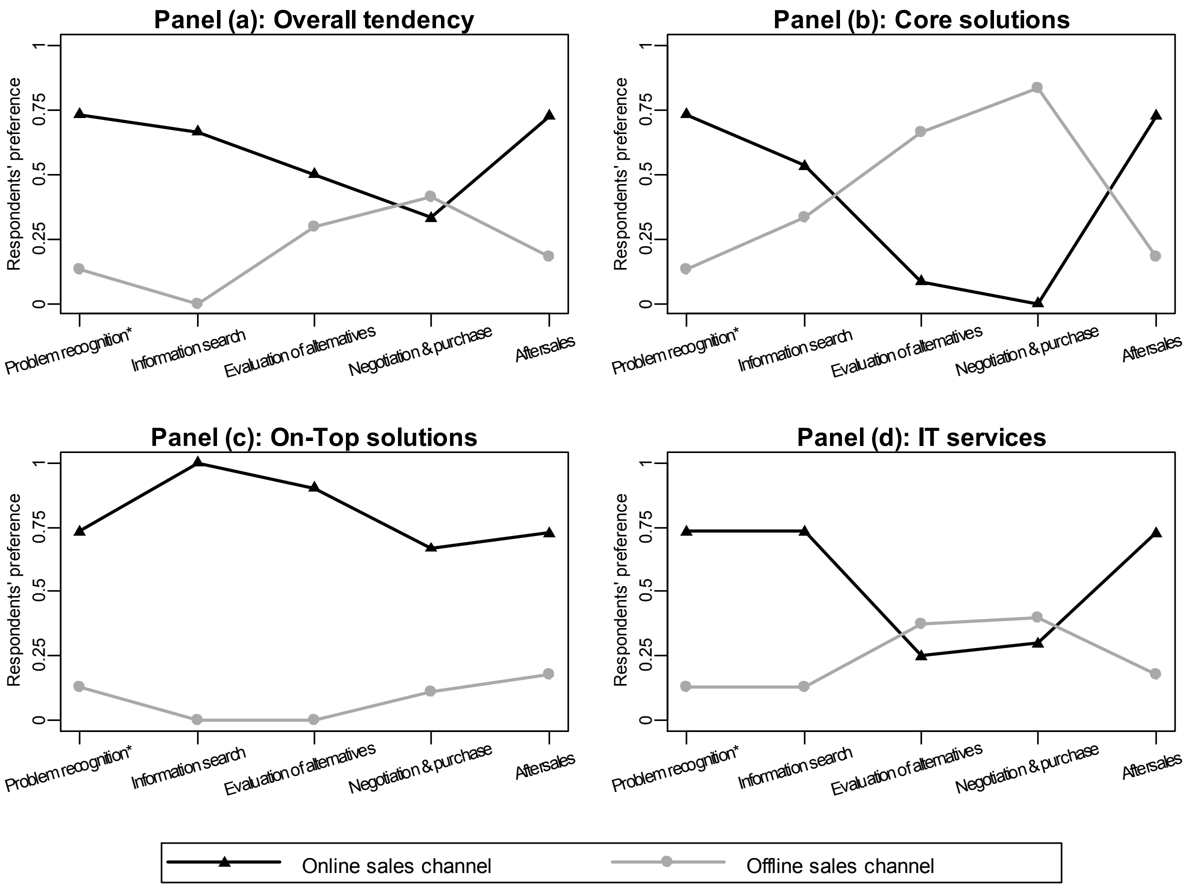


**Figure 45:** Channel preference of EAS products and services (Paper 5)

These panels consequently are also the basis for deriving design recommendations for a multichannel sales system. Generally, we can conclude that the online channel is a highly relevant and in many aspects the preferred channel choice for EAS customers. The recommendations in short are:

- An integrated sales channel system is proposed with individual channel strategies for each EAS portfolio item.
- Core solutions should be equally present on the online channel in the early phases of the buying process as in the offline channel. Dedicated handover points should be defined towards the offline channel to satisfy the customer demand for more consultation and

personal assistance in the buying decision (e.g., handover points via quote request, demo request).

- For on-top solutions an exclusive online channel should be preferred, largely replacing the offline channel. Pre-requisites are a highly standardized contractual model and minimum implementation and configuration efforts after the purchase. However, dedicated exit path in the negotiation & purchase phase towards the offline channel should be considered at least for premium customers to reduce drop-off rates.
- IT services can be initiated online, however, the high customization needs and the involvement of human resources demand a more consultative sales process. Hence the online channel might focus on highly standardized service bundles and the offline channel can take over in the evaluation and negotiation & purchase phase for the more complex services.
- A separate channel strategy should be pursued for the after-sales phase across the entire EAS portfolio, especially because in this phase often simple transactions are conducted and existing products are merely enhanced. Even if the initial purchase was conducted offline the customer should be handed over to the online channel for after purchase inquiries.

Paper 5 clearly addresses RQ_b.4 by also analyzing the models with regards to the offline channel adoption. It also contributes to RQ_b.3, by deriving further insights to the online channel in the presence of an offline channel and by providing design recommendations to implement a multichannel sales system based on the channel preferences of the customer with regards to the phase in the buying process and the type of solution in focus.

## 4.3 App Store for Enterprise Application Software as an Application System

While the previous papers have either looked at the supply- or demand-side value chain as a whole, this chapter focuses on the app store for EAS as an application system, its capabilities and features as well as its area of operation. Two detailed normative research questions have been introduced which will be covered by Paper 6 (cf. Chapter 1.2):

- What are the major usage scenarios, features, and capabilities of app stores for EAS recognizing the unique requirements of companies? (RQ_c.1)
- What are the different types of app stores for EAS and how to best use them in the corporate context? (RQ_c.2)

### 4.3.1 Paper 6: Comparing Public and Internal Enterprise App Stores: A Qualitative Study

Paper 6 selects two cases to study the two prevailing types of app stores for EAS, which are referred to as the public enterprise app store and the internal enterprise app store in this paper.

Based on the two superordinate research questions, three specific targets have been derived:

- Which scenarios (incl. features and capabilities) are supported by public and by internal enterprise app stores respectively?
- How can enterprise app stores better involve the business (user or department) while at the same time ensure IT governance targets (i.e., IT control) are met?
- How can both public and internal enterprise app stores be best used in combination?

The first of these three questions is addressed by comprehensively studying and presenting the two cases, the SAP Store (public enterprise app store) and the SAP Enterprise Store (internal enterprise app store). The cases study three aspects of the app stores respectively:

- Business system analysis: What is the business and business process context in which the application is used?
- Catalog management / offering: Which applications are supported?
- Functional analysis: What are the major capabilities and features of the respective systems?

**Business System Models**

The business systems have been modeled and analyzed using the SOM interaction scheme modeling methodology (cf. Chapter 3.4.3 on more methodological details). The two business systems are shown in Figure 46 and Figure 47

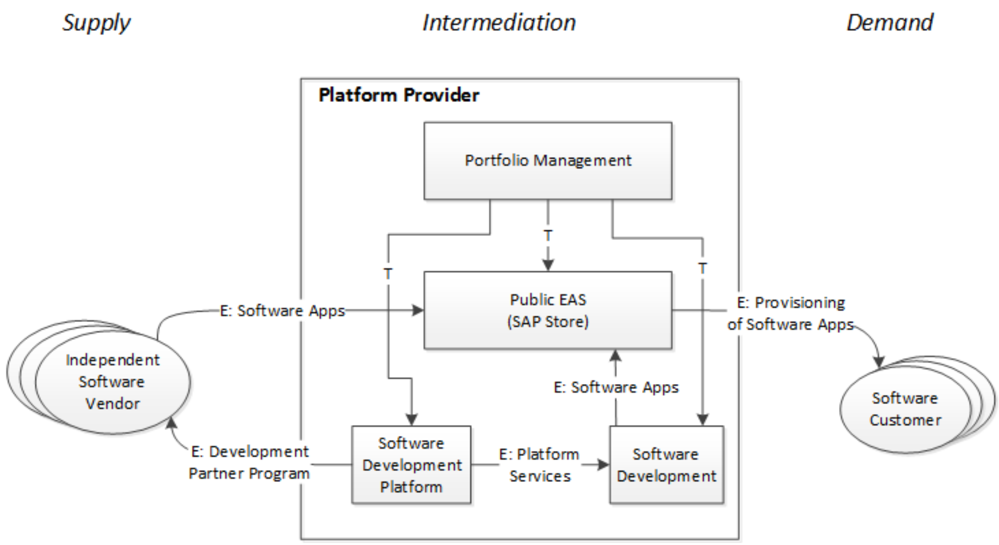


**Figure 46:** Business system model of the SAP Store, the public enterprise store (Paper 6)

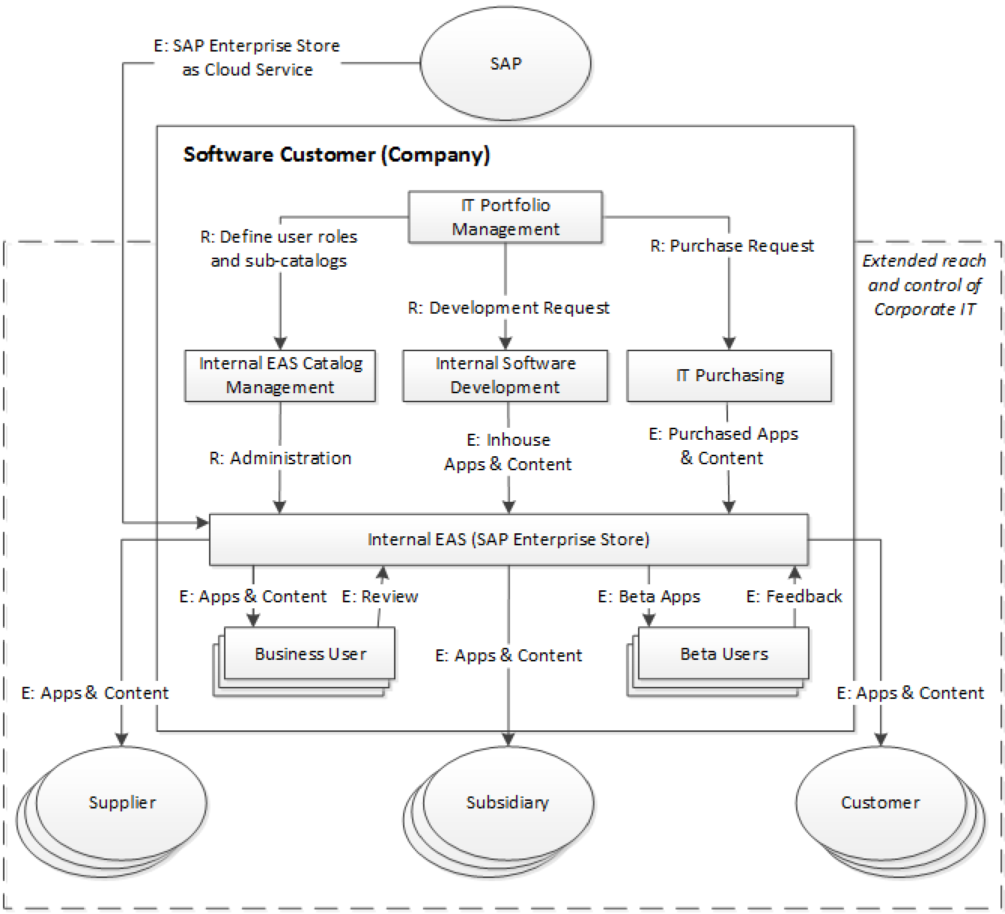


**Figure 47:** Business system model of the SAP Enterprise Store, the internal enterprise app store (Paper 6)

### Catalog Management

The catalog analysis revealed the public enterprise app store studied here has a very broad portfolio of business applications, both from the (platform) provider and from ISVs. All deployment models (cloud, on-premise, mobile apps, desktop) are covered and range from productivity applications focusing on the individual user and complex integrated enterprise applications with a sophisticated backend technology stack. The internal app store in contrast focuses on applications for individuals or only parts of a greater application which are relevant to the individual, such as the frontend or the mobile app accessing the backend application. Moreover, the internal app store can handle use licenses and e-learnings. While the catalog is managed by the platform provider in case of the public app store, the catalog and which applications are listed is fully defined and managed by the software customer in case of the internal app store.

**Functional Analysis**

The functional analysis evaluated the app stores along similar categories to ensure better comparability. However, the categories had to be adjusted slightly to acknowledge the different capabilities and usage scenarios of the respective app store.

In short, the public store provides capabilities to conduct the full buying process, from problem recognition to purchase and delivery. Moreover, an advanced user management is provided to recognize different corporate roles with different buying permissions. The store also supports a quote request and quote accepting process in case instant purchases are not feasible (high price products or products with more complex licensing structure). Hence, the public enterprise app store focuses on external sourcing and supports the entire software customer's organizational buying process. To do so, the public store implements unique requirements supporting the complexity of corporate buying situations.

The internal store focuses on the adoption of applications for and by individual business users within the scope of one company and its ecosystem (e.g., suppliers). Moreover, it supports the management of these applications by an IT department. Product characteristics of EAS for app stores. As a special capability, the internal store also supports the company-internal IT innovation process by providing capabilities to upload and distribute "apps in development" and "beta apps", and give developers the opportunity to receive ratings and detailed feedback on their applications. In conclusion one can state that internal enterprise app stores extend the concept of "software catalogs" by enhanced monitoring, application management and user experience simplifying the software adoption process for business users.

**Enterprise App Stores and IT Governance**

To address the second specific target of the paper, how enterprise app stores can better support business users and departments and ensure IT control, an IT Governance framework has been used and assessed with the findings from the case studies. The result is displayed in Table 21.

**Table 21:** Comparison of key capabilities of the public versus internal EAS model with focus on business involvement and IT control along the IT governance process (Paper 6)

| IT Gov. Process | SAP Store (public Enterprise App Store) | SAP Enterprise Store (internal Enterprise App Store) |
|---|---|---|
| Sourcing | ▪ Business can identify, gather information about, and try new business applications<br>▪ IT defines buyers and proactively invites business reps to participate during external sourcing<br>▪ IT can enable selected business reps to make purchases | ▪ Early involvement of business users in in-house development projects (internal sourcing) |
| Delivery | ▪ Instant delivery of software can accelerate delivery process | ▪ Business users select and consume apps in a self-service mode using a consumer-friendly app catalog<br>▪ Provide apps to ecosystem<br>▪ Secure and instant delivery to user devices |
| Support | - | ▪ Can be used to distribute updates of applications<br>▪ Distribute e-learnings |
| Monitoring | ▪ IT can monitor all purchases on the EAS via a central order view | ▪ Monitor app usage, downloads, reviews, ratings<br>▪ License monitoring |
| Control | ▪ Define buyer roles<br>▪ Prevent business users from buying non-authorized app-lications | ▪ Define target groups for applications (who can access which apps)<br>▪ Fully define catalog content and visual styling of app store |

## Proposed Corporate Use Scenario

Last, the paper addresses the third more design-oriented target, how the two models can be best used in the corporate context. A combined use is proposed in which the public store focuses on the external sourcing of EAS and the internal store on company internal distribution and management of EAS.

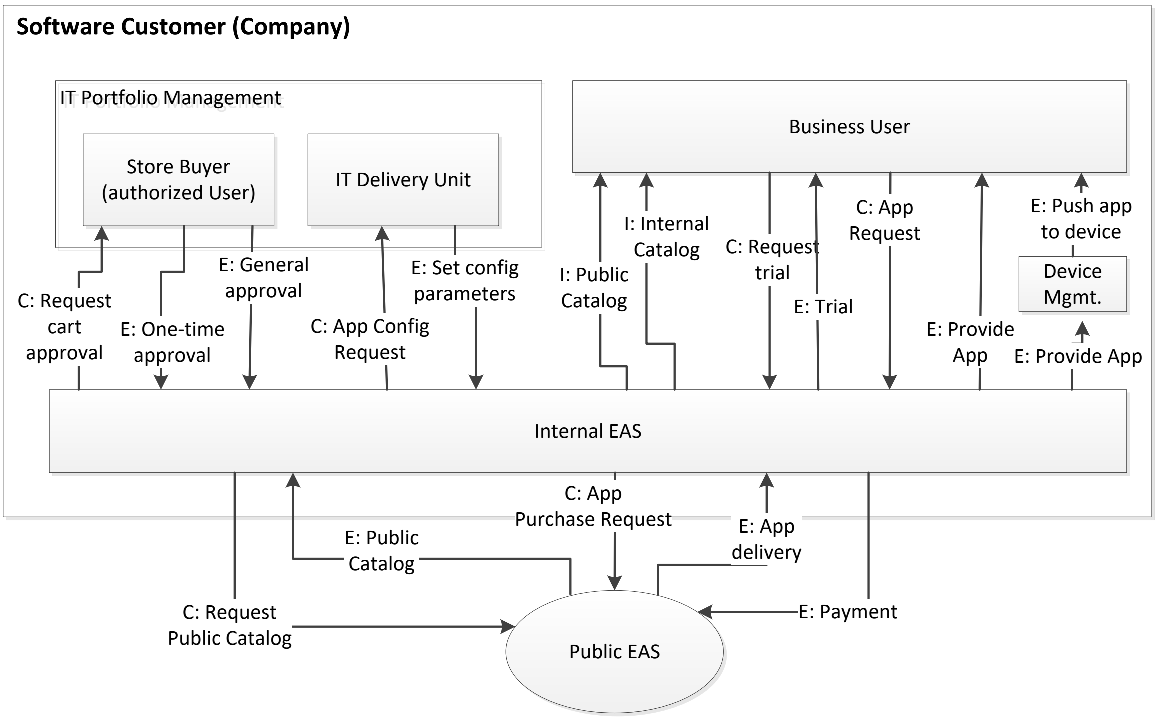


**Figure 48:** Public and internal enterprise app store in an integrated scenario (Paper 6)

Furthermore, to best leverage the combined use, a scenario is modeled (using the SOM interaction scheme) and described integrating the public and the internal enterprise app stores as shown in Figure 48.

In summary, this paper gives a unique access to two different enterprise app store types and provides a comprehensive and detailed analysis of the usage scenarios and capabilities (RQ_c.1). Moreover, their impact on IT governance processes has been assessed and discussed, which formed the basis for providing recommendations on how to best use the two systems in the corporate context (RQ_c.2).

## 4.4 Product Characteristics of Enterprise Application Software for App Stores

The last research cluster focuses on the suitability of EAS products for the app store model.

- Which characteristics of EAS products are best for the app store model? (RQ_d.1)
- Which measures help to improve the suitability for the app store model? (RQ_d.2)

### 4.4.1 Paper 7: App'ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications

One of the major findings of Papers 3, 4, and 5 was the decisive importance of product characteristics of EAS for the adoption of an enterprise app store by the software customer. Consequently, this paper builds on the previous findings and further elicits not only what the characteristics of a solution are, but also how these can be achieved by studying five enterprise applications intensively in this multiple case study.

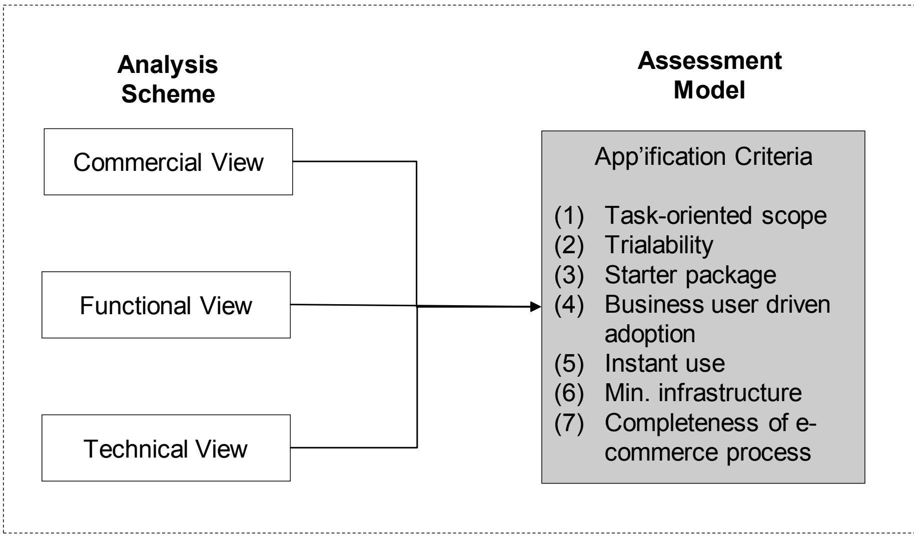


**Figure 49:** Framework of research enterprise application with regards to their suitability for the app store model (Paper 7)

The first intermediate building block in reaching the target is a framework which helps to analyze individual applications, as shown in Figure 49. The framework consists of two elements. The analysis scheme defines three views to investigate an application with 15 sub-categories. The second element is the assessment model, rating an application on its suitability for the app store model using the app'ification criteria. The app'ification criteria are derived from the solution-related determinants from Paper 3-5 and represent the ideal instances of these dimensions to support the app store model combined with a business driven adoption. The analysis scheme helps to structure the information about an application and then rate the application using the assessment model.

The next major result of this paper is the knowledge gained from studying the five applications in-depth and identifying how these applications achieved the respective criteria. A summary of the findings is shown in Table 22.

**Table 22:** Summary of findings to improve suitability for the app store (Paper 7)

| # | App'ification criteria | Possible measure(s) |
|---|---|---|
| 1 | Task-oriented scope | ▪ design applications for single task or single business role, apply user centric design paradigms during application development<br>▪ design applications to allow partial use scenarios<br>▪ dis-aggregate application monoliths in smaller applications with enhancements or add-ons |
| 2 | Trialability | ▪ provide limited free versions (time or functionality restrictions)<br>▪ provide pre-configured cloud-based trial systems with sample data for complex solutions (shared or personal)<br>▪ provide customized trials with special requirements<br>▪ mobile apps with demo mode (no backend connection)<br>▪ video tutorials and guided video tours<br>▪ screenshot walkthrough<br>▪ click-through-demo<br>▪ screen-cast<br>▪ combination of various to satisfy the different needs for evaluation |
| 3 | Starter package | ▪ user-based pricing with low starter volumes<br>▪ subscription-based pricing<br>▪ bundling of different pricing elements<br>▪ "onion pricing" combining software, service, operations, and support in a one-dimensional pricing scheme tailored to the target audience to allow incremental adoption of the solution over time (start small and grow with the needs) |
| 4 | Business user driven adoption | ▪ consumer-oriented user interface design<br>▪ provide business users access to information material and trials without special permissions |
| 5 | Instant use | ▪ define minimal use scenario without customization need<br>▪ provide standardized service offerings to implement or configure and customize solution ("packaged implementation")<br>▪ offer remote service and support session to get application up and running<br>▪ provide best practice templates and tool-based wizards with business language to customize solution<br>▪ provide activation and access link directly after purchase<br>▪ automatically create first user based on buyer information |
| 6 | Minimal infrastructure pre-requisits | ▪ provide cloud versions or deployments of the application<br>▪ allow standalone mode even if system requires integration to legacy system by e.g., implementing manual data upload or pre-configured cloud-based integration services.<br>▪ use existing IaaS offerings to enable cloud deployment for legacy on-premise applications |
| 7 | Completeness of e-commerce process | ▪ provide comprehensive online accessible material to satisfy different information needs (value propositions, functional or user oriented documentation, technical information)<br>▪ access to trials online<br>▪ enable online purchase and delivery / activation for at least the minimum starter package |

The paper uses the previous research to derive the app'ification criteria which can then be used to assess whether an EAS product is suited to being sold via an online channel with a focus on business users or departments (RQ_d.1). The in-depth case studies revealed numerous and alternative measures which can be applied to make applications ready for online sales (RQ_d.2).

# 5 An Integrated Perspective on the App Store Model for Enterprise Application Software

This chapter presents the model artefact having integrated the previously presented research results in the form of a business model framework with a focus on the value architecture. First, Chapter 5.1, introduces several aspects the reader should consider while reading this chapter and using its results. Paragraph 5.2 then starts by deriving definitions for the app store for EAS and the App Store Model for EAS. These definitions are the basis for the model framework presented in the form of the value proposition (5.3), the value architecture (5.4), and the revenue model (5.5). Using the structured feedback received from the consulted experts the appropriateness and usefulness of the presented model artefact is reflected on and discussed in paragraph 5.6.

## 5.1 Interpretation of the Model Artefact

The purpose of this chapter, the model artefact, is to give design, tactical and operational recommendations based on the empirical findings of the papers 1-7. The recommendations represent one possible solution space or “constructed possible world” (Frank 2009) which I could faithfully derive from the findings. It explicitly leaves out possible options and variants, which might equally fit the respective problem statements, but for which I have not found any backing in the previous studies or the related literature that was introduced, or which could not be properly integrated into the larger picture.

While I have tried presenting both the methodology as well as the empirical findings in detail, the development and derivation of the design recommendations (i.e., the model artefact) is a translation step which is highly influenced by the subject, i.e. me, my context, my knowledge and my professional experience. Hence, subjectivity cannot fully be avoided, and it will not be possible to fully root back every design recommendation element to a single empirical finding. The topic at hand is simply too complex and interrelated to keep things completely separate. The modeling act as such consequently and necessarily involves a certain level of subjective creativity which cannot be expatiated. Last, the topic covers a large breadth and touches on areas which nowadays have established their own extensive research clusters with possibly many high-class publications per year (e.g., software ecosystems). Consequently, with regards to comprehensiveness the recommendations do not claim absolute completeness.

Furthermore, as described earlier, the App Store Model for EAS is a business model for software platform ecosystems. Nevertheless, most recommendations are directed or are most useful to the platform provider role who either wants to establish an app store model or evaluate and improve an existing one. Dedicated aspects, however, are also addressed to

ISVs or software customers – these recommendations can be either used directly by these two groups or mediated by the platform provider.

Last, as already elaborated in the methodology section (cf. 3.5.3), the recommendations and model elements are not ready blueprints for companies. With regards to the level of information granularity, the business model framework can be localized between “strategic concepts” and “business process models”. Hence, compared to business blueprints or concrete business process models it has a higher level of abstraction and focuses more on the unique aspects of the business model at hand and leaves out necessary but often basic aspects which need to be considered in order to make it work in the real world: E.g., the app store capabilities described do not include all the e-commerce functionality needed, but only focus on the specific aspects relevant in the EAS context.

Considering the presented assumptions, I define the following objectives for the model artefact:

- The artefact defines the value proposition, the value architecture, and the revenue model of the App Store Model for EAS.
- The value architecture represents the core of the solution space to be developed, in the form of semi-formal model elements and design recommendations, which are capable of delivering the postulated value propositions for the respective actors and the revenue model.
- The solution space is defining the structure in the form of the required business processes (i.e., business objects and transactions) and resources as well as descriptive recommendations.
- The model artefact should be abstract enough to serve beyond the scope of a single company or real-world context.
- It aims to help substantiate strategic concepts and serve as a basis for deriving more concrete business process models and application blueprints or requirement specifications.
- The model artefact should be practicable, i.e., it should be comprehensive, comprehensible, and detailed enough to help practitioners establish, enhance, or adjust an App Store Model for EAS in practice.

The expert review and reflection conducted in Chapter 5.6 will assess the model artefact regarding its usefulness and appropriateness in relation to these objectives.

## 5.2 Definitions Used for the Model Artefact

Though the definitions introduced hereinafter may also be used beyond the scope of this chapter, they have been especially developed to serve as model assumptions to the presented model artefact.

First, I introduce a definition of the app store for EAS. The definition uses and enhances the definition of Jansen & Bloemendal (see also Chapter 2.6.2) (Jansen & Bloemendal 2013):

> *An app store for enterprise application software is an online curated marketplace that allows independent software vendors to market, sell, and distribute their products to software customers within a software platform ecosystem for enterprise application software.*

Building on this, I define the App Store Model for EAS:

> *The app store model for enterprise application software is a business model described by the value proposition, the value architecture, and the revenue model.*

Furthermore, the model as presented in this chapter has the following boundary conditions:

- *a) Three actors are differentiated, the platform provider, the independent software vendor, and the software customer;*
- *b) a technical software platform and an app store are technological prerequisites;*
- *c) the platform provider owns and provides the services of the software platform and the app store;*
- *d) moreover, the platform provider offers one or more core products to software customers;*
- *e) ISVs use the software platform or platform services to develop complementary software products;*
- *f) ISVs use the app store to market, sell, and distribute these products to software customers;*
- *g) software customers are companies or corporate entities with adequate buying authority;*
- *h) software products marketed to software customers classify as enterprise application software.*

Whilst most of the boundary conditions should not require additional explanations, condition (d) might be perceived as optional at first. I have added this as a condition, however, since platform providers without core products would differ greatly with regards to their business rationale on establishing an app store model: In particular, they would not have an established EAS product, in turn they would not have a reputation with regards to this product segment at all towards software customers and the software ecosystem, especially the ISVs, could not profit from this reputation in joining the platform. Although an App Store Model for EAS with a platform provider without core products may equally be a valid and sustainable economic option, most of the research conducted in this thesis is related to platform providers with core products. Consequently, the theories developed and recommendations derived are based on this ecosystem variant, and due to its decisive impact on the overall model it should be included as a constraining model assumption.

Condition (g) has been added since one of the rationales to foster the app store model was to better involve business stakeholders in the software acquisition process and provide them with an option to flexibly and quickly satisfy IT or application demands. Hence, the software customer need not only be a company as a whole or the representing procurement officer, but also small organizational units down to the individual business user. A pre-requisite is that they have been granted certain modes of authority.

As a last remark, I have not added any condition with regards to the deployment model of the EAS (e.g., mobile, cloud, on-premise). However, consistent expert feedback throughout the entire research of this thesis suggests that the dominant deployment model for EAS will be a cloud-based model. Though the proposed business processes and resource capabilities should also mostly work for on-premise based software, some might be better tailored to a cloud-based application stack.

## 5.3 Value Proposition

The value proposition as an element of the business model framework suggested by Stähler, describes the benefits and values gained by either customers or value partners from the business model (Stähler 2010). The benefits or value propositions for the focal company are typically not or only considered as revenue related, and reflected on as part of the revenue model. The App Store Model for EAS has three stakeholders. While the focal company of this business model is the platform provider, I argue that this business model provides a concept for the entire software platform ecosystem. Also for the platform provider, the app store model proposes more than just revenue related value gains. Hence, the value proposition should be defined for all actors of the App Store Model for EAS, the ISV, and the software customer, *and* the platform provider.

### 5.3.1 Value Proposition for Independent Software Vendors

The value proposition for ISVs has been derived from the findings in Paper 1, 2, 6 (cf. Chapter II.1, II.2, and IV.1), and complemented with findings from the works by Giessmann and Legner (Giessmann & Legner 2013). Five distinct value proposition for ISVs can be differentiated for joining a software ecosystem according to the App Store Model for EAS:

a) enhance customer access and market reach (cf. Paper 1, Paper 2);

b) better meet customer demand (effectiveness) (cf. Paper 2);

c) improve efficiency and realize cost savings, i.e., development, operations, marketing, and sales (cf. Paper1, Paper 2, Paper 6, (Giessmann & Legner 2013));

d) gain access to unique resources and 'core product' APIs (cf. Paper 2, (Giessmann & Legner 2013));

e) expand business by additional lines of business (cf. Paper 2).

Value proposition (a) promises the ISV access to new customers or entire market segments (e.g., new countries, large enterprises) which were not reachable before. It may also include access to the platform provider's so-called (protected) installed base, that is, his existing customers for the core products. The platform moreover helps him gain additional visibility, and a positive reputation of the platform provider can have a positive impact on the ISV's credibility and trustworthiness.

Proposition (b) is backed by the idea that the ISV can benefit from the platform providers' guidance in strategic investment areas and technology (e.g., certain industries, big data, mobile apps, cloud). According to this proposition, the platform provider has a better market overview and intelligence, and can hence make more informed decisions about future investment areas. Moreover, the ISVs as part of the platform gain access to a broad community of software customers and potential customers, and dedicated knowledge resources of the platform provider which can provide additional insights into software needs and requirements. Altogether this helps to more accurately meet customer demands and thus reduce waste in the value chain.

The platform and the app store act as a catalyst, as they provide resources to support the core business process of the ISV. Hence, they improve the overall efficiency of the development, marketing, and sales activities of the ISV (proposition c). The catalyst function of the app store and the platform goes back to the specialization within the value chain. The platform provider can invest a multiple of resources supporting the development, operations, marketing and sales processes compared to the individual ISV. Instead of taking care of the platform technology stack, e-commerce (app store) system or other supporting functions (e.g., billing and collection) themselves, the ISV can outsource these functions to the platform provider and focus on its core capabilities which are: capturing domain knowledge and developing business applications. Moreover, the use of the platform can significantly improve the time-to-market since the ISV value chain can be simplified significantly.

Next, the resources owned and maintained by the platform provider are often not only more cost efficient for the ISV when used via the platform service but often it is also the only way to access such resources (d). Platform providers with highly successful core products often protect APIs which would allow the enhancement of the core products with proprietary standards and associated terms. The only access to these APIs is then via a fee-based platform service. But even if the APIs are legally and technically accessible without a platform subscription, software customers often demand a certification of applications enhancing core products. These application certifications are again only available for platform members. Last, platform providers often also invest in unique technology which provides a competitive advantage to the ISV's application (e.g., in-memory database).

Companies also join software platform ecosystems since they intend to grow their business into new business segments or entirely new business models. Especially in the enterprise software domain, many small companies have focused on a services business model (implementation, integration, and customization) acting in a highly local manner and covering the so-called "last mile" to the customer (cf. 2.2.2 "core shell model). These service-and geographically locally-oriented business models originate from the "on-premise world" where applications were built on complex technology stacks operated at the customer leading to a high demand for locally available expertise to operate and customize the applications. With the rise of cloud computing these business models are endangered and companies are looking for new, alternative opportunities. One of these opportunities seems to be the software product business. Platform providers with productized platform service offerings present a compelling opportunity with low entry barriers to such businesses. ISVs can learn best practices regarding software product business or learn about new industries, technologies, etc. from the knowledge resources offered by the platform.

### 5.3.2 Value Proposition for Software Customers

For software customers, six value propositions are differentiated. The propositions have been derived from all the papers in this thesis and, additionally, arguments have been confirmed using the works by Giessmann and Legner (Giessmann & Legner 2013) and Wenzel et al. (Wenzel et al. 2012):

a) better meet software needs with standardized enterprise applications (Paper 6, 7, (Wenzel et al. 2012));
b) meet software demands quicker and more flexibly (Paper 7, (Wenzel et al. 2012));
c) better involvement of business stakeholders (Paper 6, 7, (Wenzel et al. 2012));
d) transparent and better IT and application standards and service levels (Paper 1, 6);
e) faster and more efficient buying process (Paper 3, 4, 5, (Wenzel et al. 2012));
f) less complexity in software supplier and software license management (Paper 3, 4, 5).

With the app store model software customers get transparent access to a broad range of enterprise apps beyond a single vendor via a single store front. They can transparently browse these applications following a standardized catalog taxonomy and access compatibility information. The apps from ISVs enhance and complement core products from the platform provider. The fine granular structure of the complementary apps allows software customers to compile an application stack with standardized products that better meet their needs compared to single monolithic applications (proposition a). To adjust these monoliths, often costly custom solutions have been developed by service providers (see paragraph. 2.2).

Moreover, the app store model promises to meet demands quicker and more flexibly (b) compared to the long-lasting buying cycles in traditional EAS acquisitions (Wenzel et al. 2012). A positive influence on how flexibly and quick customer demands can be met is the granular structure of complementary apps. The lower granularity reduces the entry barrier for single investments and along with this the individual risk and switching cost for an application. Compared to large application monoliths covering hundreds of use cases, a more granular and focused application allows software customers to justify investments with a simple value-cost analysis. Moreover, the app store infrastructure mostly allows access to demos or trials which can be quickly evaluated by real end-users further positively influencing the buying process duration and the associated risk. The digital and instant buying process combined with cloud-based pricing models also allows peak demands for software services and applications to be covered flexibly without risking acquiring “shelf-ware”.

The seamless and integrated digital software buying and deployment processes via the app store allow shorter buying process cycle times and consequently a better time-to-value ratio (c). The e-commerce capabilities reduce the need for in-person meetings typically conducted in enterprise buying situations, and the information required to make the buying decision is rather “pulled” from the buying center members versus the “push” mode prevalent in traditional personal sales models. The pull model allows for more focused, personalized information flows and less information redundancy, which in turn leads again to a more efficient buying decision process. The seamless and consumer-grade app store interfaces allow all buying center members to make use of the app store infrastructure. In case personal consultancy is required the app store should tightly integrate with the personal sales channel, i.e., handover to personal sales agents. This way the two channels, app store and personal agent, cross-fertilize each other, focusing on their individual strength and resulting in a more effective and efficient sales approach.

The app store model also promises to improve the involvement of business stakeholders in the buying decision (d). The app store is typically publically available, and business users can inform themselves and evaluate available applications. With adequate permissions management, app stores can even be used by business users to conduct buying activities without a dedicated involvement of IT personal. The app store for EAS at the same time promises to ensure corporate IT governance regulations and thus avoid shadow IT.

The terms of the app store and the software platform the platform provider can enforce uniform application standards (e.g., support SLA, availability SLA) across the app portfolio of the ecosystem. Customers relying on the applications from this ecosystem can then be sure that standardized minimum SLAs apply for all applications. This increases transparency, and simplifies SLA monitoring, comparability, risk management, as well as SLA enforcement (proposition e). Moreover, the platform provider can act as an additional “insurance” (trusted third party) in case of SLA violations by one of the ISVs.

Last, proposition (f), promises less complexity in the overall process of software supplier management. The unified store front of the app store and the capabilities of managing and monitoring application contracts, licenses and usage agreements in a central place simplifies the management of the software suppliers. Furthermore, the enhancement or termination of contracts follows a standardized uniform process for every ISV. In case the app store provides billing and payment services, the software customer can benefit from a single billing statement with all cost items in one overview, and utilizing a single payment transaction to settle the liabilities with all accounts. The higher transparency and uniform supplier and contract management should ultimately also support software customers in reducing waste and “shelf-ware” as reflected in their IT expenses.

### 5.3.3 Value Proposition for Platform Providers

The five value gains promised by the app store model for platform providers have been extracted from Papers 1, 2, and 7 (cf. Chapter II.1, II.2, and V.1), complemented with the paper by Wenzel et al. (Wenzel et al. 2012), and triangulated with the works by Popp and Meyer (Popp & Meyer 2010, p.132 ff.):

a) create new sources of revenue (Paper 1, (Popp & Meyer 2010));
b) expand addressable market and market penetration (Paper 1, Paper 2, (Popp & Meyer 2010));
c) increase marketing & sales efficiency and effectiveness (Paper 7, (Wenzel et al. 2012; Popp & Meyer 2010, p.133);
d) increase development efficiency and effectiveness (Paper 1, (Popp & Meyer 2010);
e) benefit from network effects (Paper 1, (Popp & Meyer 2010)).

The first and probably most convincing value proposition is the opportunity to create new sources of revenue (a). There are two major streams of additional revenue related to the app store model: first the revenue generated by providing a platform service, and second in obtaining a revenue share from the app sales from ISVs. Further details of this value proposition are given in the revenue model in Chapter 5.5.

As for the ISV, the app store model is also an opportunity for the platform provider to increase his market reach and market penetration (proposition b). The platform provider can simply not cover all the business requirements with his core products. Complementary apps from ISVs can cover additional niches and especially long-tail requirements. Complementary apps may also reduce the need for custom applications developed for software customers and thus reduce the total cost of ownership (TCO) of core products for software customers. Complementary apps, however, can not only cover niches or long-tail requirements in already addressed segments, but can also open up entirely new ones (e.g., countries, industries) and thus provide the breeding ground for the platform providers' core products. Moreover, the market reach is further influenced by the app store as an electronic sales and marketing channel: it can approach much more software customers, i.e., companies and users with lower cost compared to a personal sales channel, and consequently create more sales opportunities for the entire ecosystem. By also addressing the business users and departments with the app store channel, platform providers further establish a "new route" into customer organizations.

The latter two arguments also pay off for the next value proposition. The app store increases the efficiency (time and cost) as well as the effectiveness of the platform provider's sales processes (proposition c). The app store as e-commerce system represents a sales channel with a high level of automation for the platform provider. The marginal costs are consequently very low compared to the marginal costs of a personal sales channel with an increase of sales costs that is close to linear. Moreover, the high level of automation on the provider side reduces the time to market. The effectiveness is improved by leveraging the app store as part of a multi-channel sales system in which every sales channel (personal, remote personal, app store / online sales channel) focuses on its individual strengths and cross-fertilizes the other channels. Last, the effectiveness is further influenced since the app store channel can highly personalize its marketing activates down to a user level, leveraging today's recommendation engines and context-sensitive technologies, while the app store approach can address customer expectation towards enterprise software buying with a "consumer grade ease of use" and "instant value generation" (Wenzel et al. 2012).

Aside from a more efficient sales and marketing process, an app store model based software ecosystem also promises a more efficient and effective product development. Platform providers concentrate on core applications covering "horizontal" functionalities used by many customers, and ISVs provide "vertical" functionality required by certain industries or niches. Outsourcing the long-tail to ISVs reduces the scope and complexity of the core products and thus reduces the cost associated with requirement management, application maintenance, etc. Moreover, leveraging ISVs to capture specific customer, or industry knowledge further reduces research cost. Effectiveness is addressed at the same time, as ISVs which maintain close and local relationships to customers can better address specific demands than a centralized large software provider. In critical areas it might also make sense to pro-actively co-innovate with ISVs to fill certain functionality gaps. Last, the app store model approach also improves the utilization of platform resources when shared with ISVs (e.g., cloud infrastructure) and thus improves the cost structure of large technology investments.

The two-sided platform business resulting from the App Store Model for EAS is characterized by the so-called network effects. The network effects can be used to explain a large portion of the attractiveness of software platform ecosystems. In particular, indirect network effects as they occur in two-sided platforms are both a challenge and a potential benefit for platform providers: the more ISVs the more attractive they are to software customers, while the greater the number of software customers the more attractive the platform to ISVs (cf. Chapter 2.3.4). When starting such a platform either customers or ISVs are often low in number. When the network effects come into force they also increase the “platform stickiness” (Popp & Meyer 2010, p.134). In essence, the stickiness can be interpreted as switching cost of either ISVs or software customers to a different platform. This stickiness can therefore represent a significant competitive advantage.

## 5.4 Value Architecture

The value architecture is sub-divided into three further sections, the supply-side value chain (5.4.1), the demand-side value chain (5.4.2), and the characteristics of enterprise software products for the app store model (5.4.3). The app store as an application system which was a dedicated research cluster in the empirical part of this thesis (cf. 1.3, 4.3) has been integrated into the supply- and demand-side value chain sections. The two sections are further divided in a task-level analysis and resource-level analysis. The task-level focuses on the business processes and business objects, while the resource-level identifies personnel resources and application system or machine resources. The app store as an application system is hence included in the resource perspective of the value chain analyses.

The descriptive explanations[1] in this chapter have to be restricted to selected results for considerations of the length of the thesis. Hence I have chosen only those aspects which I believe are critical to the presented model and which help the reader comprehend the whole. Consequently, parts of the results in this section, i.e., detailed tabular overviews, have been moved to the Appendix and are referenced at the appropriate position in the text. This will also improve readability. However, the outsourced overviews are no less important, and represent an equal part of the results, and as such ought to be considered when applying the results in practice and when assessing the usefulness and appropriateness of the artefact.

[1] Major constructs, such as business objects, business process phases etc. will be highlighted using italics. Moreover, when referring to named business objects in this section, upper case will be used.

### 5.4.1 Supply-Side Value Chain

The supply-side value chain models have been developed using Paper 1 and Paper 2 (cf. sections II.1 and II.2). In some cases the raw material (e.g., interview transcripts) had to be revisited in order to derive certain aspects. The recommendations in this section translate the value proposition of the ISV and parts of the platform provider into a possible solution structure represented by business process model elements on the task level, and functionality and capability descriptions on the resource level. Aside from the value proposition, these structural elements are furthermore developed using the mainly behavioral-oriented model proposed in Paper 2, which represents an adoption model of ISVs for the software platform ecosystem, and as such introduces important determinants which ought to be translated into respective business processes or resource capabilities. Paper 1 contributes both partly behavioral elements, but also basic structures which could be reused in the development of the supply-side recommendations and model elements.

#### 5.4.1.1 Supply-Side Value Chain: Task Level

The IAS presented in Figure 50 represents a simplified possible solution to model the business processes of the supply-side value chain of the app store model with the *Platform Provider*, such as the so-called “universe of discourse” with its subordinate business objects represented by rectangular shapes and the *ISV* and the *Software Customer* as “environmental business objects” represented by oval shapes (Ferstl & Sinz 1997) . The model is structure-oriented and includes the major business objects, the service business transactions (directed arrows with label “E”) and the related coordinating business transactions (directed arrows with label “T”, “I”, or “C”).

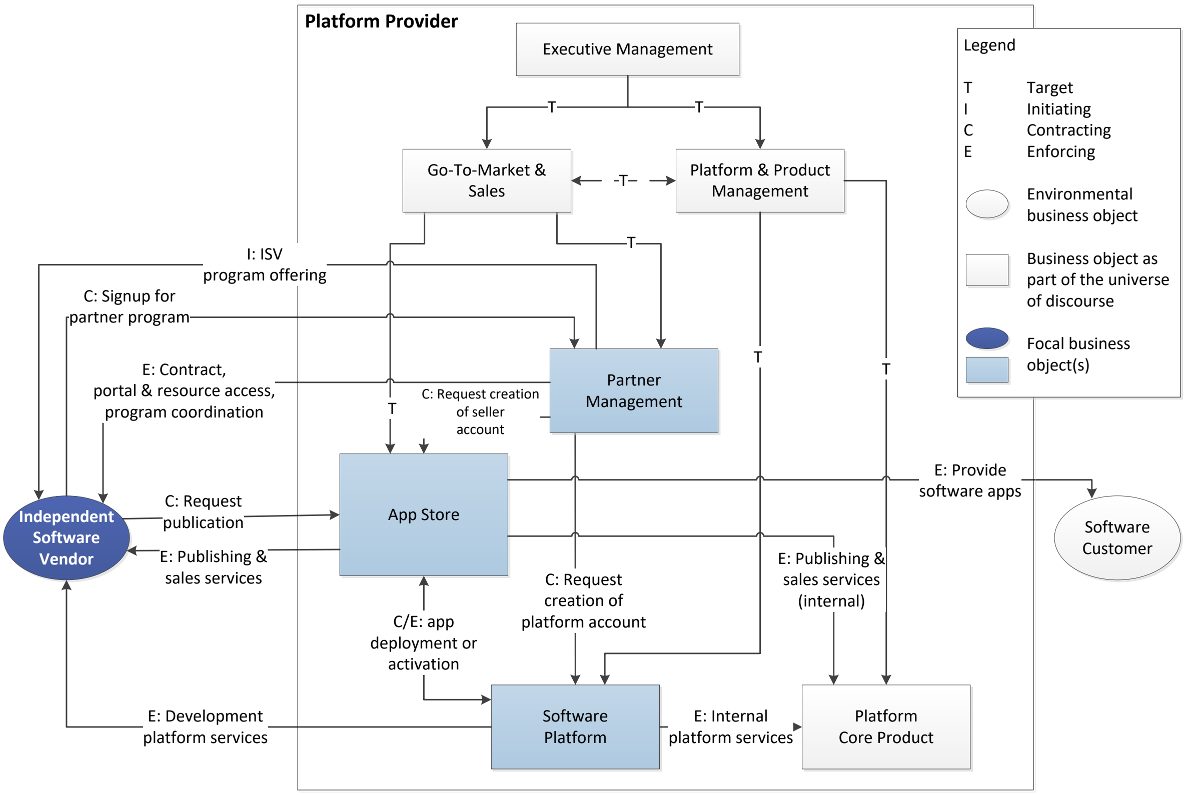


**Figure 50:** Supply-side SOM interaction scheme (simplified)

The *Executive Management* may be the company management or the management of a business unit. The next hierarchical level is decomposed into *Go-To-Market & Sales* and the *Platform & Product Management*. Both are coordinated by the executive management via targets. To achieve their goals an optional target alignment is indicated and expresses the need for close collaboration of *Go-to-Market & Sales* and *Platform & Product*.

The *Go-To-Market & Sales* is modeled with two subordinate objects, the *App Store* and the *Partner Management.* The *Platform & Product Management* in turn coordinate the two objects *Software Platform* and *Platform Core Products*. The business objects as part of the universe of discourse and their description, i.e., the description of their responsible task cluster is listed in Table 23. The descriptions therein are mainly focused with regards to the supply-side task responsibilities. The three most important objects interacting with the ISV on the supply-side value chain within the Platform Provider are *the Partner Management*, *the App Store*, and *the Software Platform* and are therefore highlighted in light blue in Figure 50.

**Table 23:** Business objects of the platform provider (supply-side focus)

| Business object | Description (task cluster) |
|---|---|
| Executive Management | ▪ define and adjusts company targets and reviews target achievement<br>▪ assign sub targets to Goto-Market/Sales and Platform & Product Mgmt. |
| Platform & Product Management | ▪ define market segments covered by core products & the ones to be addressed by the platform<br>▪ define product and platform portfolio<br>▪ define product and platform targets |
| Go-To-Market & Sales | ▪ responsible for marketing and sales activities<br>▪ closely aligns targets with product & platform management<br>▪ define targets for the app store<br>▪ define targets for partner management<br>▪ entails personal/remote personal sales |
| **Partner Management** | ▪ attract new and retain and grow existing partners / ISVs ("partner sales")<br>▪ define details on ISV program<br>▪ inform and enable ISV partners<br>▪ support ISV partners with coordination throughout their lifecycle<br>▪ administrative function towards ISVs (i.e., contract management) |
| **Software Platform** | ▪ define, built, support, and operate software platform components<br>▪ delivers software platform services to company internal and external customers |
| **App Store** | ▪ provides electronic marketing, sales, and distribution services to internal and external customers<br>▪ defines app store capabilities and implements e-commerce functions<br>▪ curates and manages app catalog<br>▪ operates app store and provides related services<br>▪ consults internal and external units on sales and marketing via the app store<br>▪ app payment & settlement (billing and collection) |
| Platform Core Products | ▪ define, design, & develop enterprise applications to be marketed to software customers directly<br>▪ request software platform services to build the applications<br>▪ request app store services to market, sell, and distribute<br>▪ contribute API which can enhance the core products to the software platform |

The *Partner Management* has the responsibility to attract new ISV partners onto the platform and retain and grow existing ones. This function may also be referred to as "partner sales" and may be associated with various targets, e.g., regarding the number of new partners, the type of new partners (from a specific segment), the number of applications contributed to the platform or the revenue generated with partner apps. Beyond the pure "sales" activities the *Partner Management* also defines the details (i.e., terms, service elements, pricing) of the productized platform service, the "ISV program". The service elements are partly delivered by the *Partner Management* itself, but the major parts are delivered by the *App Store* and the *Software Platform*. The services delivered by *Partner Management* to the ISV entail information and enablement services (e.g., on the program, infrastructure, tools), coordination services throughout the ISV's lifecycle on the platform, as well as, administrative functions such as contract and relationship management. *Platform Provider* internally the *Partner Management* requests the inclusion of new ISVs into the service delivery of the *App Store* and the *Software Platform.*

The *Software Platform* defines, builds or develops, supports, and operates the components of the technical software platform. Moreover the *Software Platform* delivers the software platform services to internal (the *Platform Core Product*) and external customers (i.e., the *ISV*). As such the software platform services entail the provisioning of all required infrastructure and tools as well as providing in-depth enablement services and support in case of issues during the development of apps. The software platform also oversees the operation of applications, i.e., cloud applications.

The *App Store* is the electronic sales agent for internal (*Platform Core Products*) and external (*ISVs*) customers. From an ISV perspective the *App Store* takes over intermediary functions and provides him with "on-behalf" marketing, sales, and distribution services. To provide these services, the e-commerce capabilities are defined, implemented, and operated by the *App Store*. Moreover, the app store is the curator of the app catalog ensuring the defined standards are complied with. The *App Store* reporting to *Go-To-Market & Sales* receives also revenue targets related to the online sales channel. Last, the *App Store* provides expert enablement sessions to internal and external customers on how to best position, market, and sell applications via the app store online channel. To deliver the apps to software customers after the purchase, the app store requests dedicated services from the *Software Platform* internally (e.g., activation or deployment of cloud apps). Last, the *App Store* also takes care of billing towards *Software Customers* and *ISVs,* collecting the payments from both sides and distributing the earned revenues to the respective *ISVs*.

**Inside the Independent Software Vendor**

One aspect with regards to the *ISV* organization should be addressed with is not covered by the IAS shown in Figure 50, and where a view inside the *ISV* organization is needed (cf. Figure 51). During several interviews with *ISV* representatives we received feedback that the "product business" was a challenge for the *ISV's* organization. Many of the companies joining EAS platforms originate from service businesses which are not used to maintaining a software application product over a longer period of time. Instead they are used to develop custom solutions on a "time and material" basis. Since there are manifold differences between these two types of business models, it is recommended that the *ISV* bundle their product related activities and establish a dedicated product division with product managers and product development.

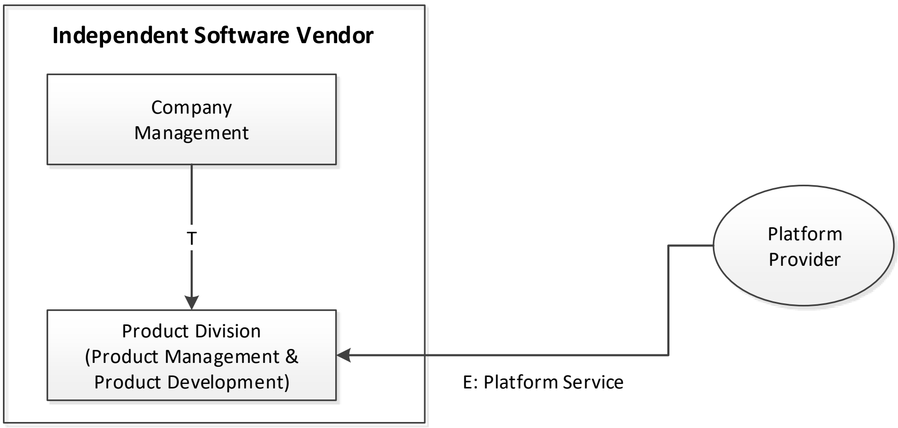


**Figure 51:** Inside the ISV organization (SOM interaction scheme)

The transactions between the *Platform Provider* and the *ISV* have been further analyzed and broken down. In total 25 transactions have been decomposed and grouped in “ISV Lifecycle Phases” (cf. Figure 52). The complete tabular overview is presented in Appendix F. On the level of the detailed transactions the resources needed have been identified and also assigned to the respective transactions.

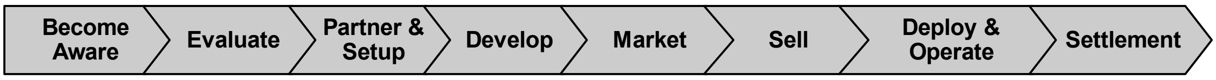


**Figure 52:** ISV lifecycle phases

#### 5.4.1.2 Supply-Side Value Chain: Resource Level

Based on the previously described transaction analysis, the resources, their capabilities and functions have been identified, and assigned to the respective business objects and transactions. The detailed assignment of resources to the respective transactions is presented in Appendix F.

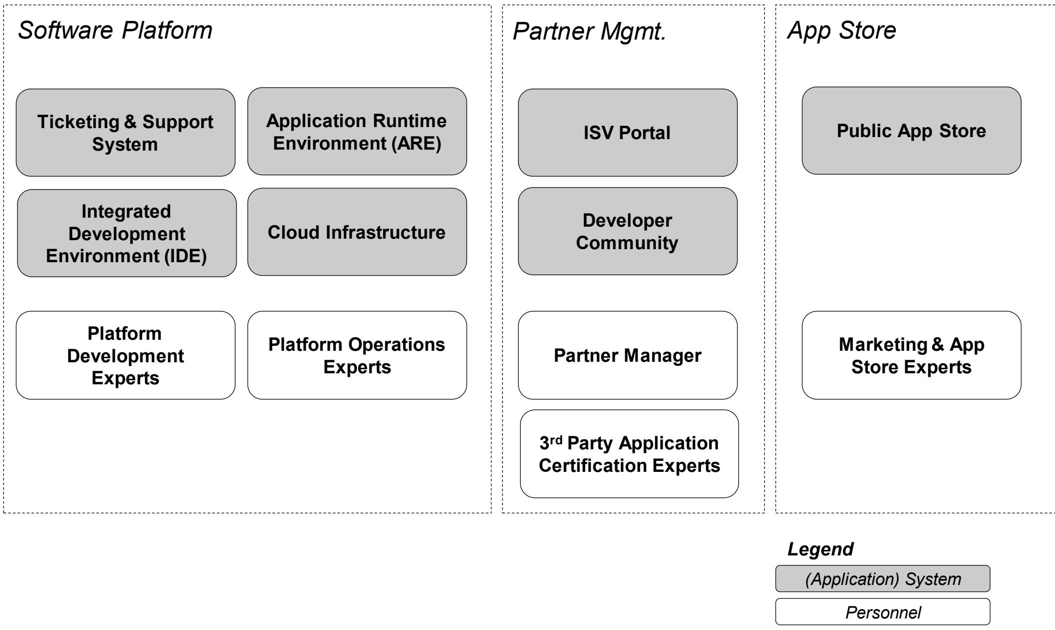


**Figure 53:** Resources of the platform provider (supply-side relevant)

Figure 53 shows the different supply-side relevant resources of the platform provider and assigns them to the previously introduced provider internal business objects *Software Platform*, *Partner Management*, and *App Store.* The resources are either personnel resources or application systems. In the case of the *Cloud Infrastructure*, the non-personnel resource also includes hardware. The detailed function and capability lists of these resources is shown in Appendix F. In the following I will briefly present their major function and capabilities only.

The *Ticketing & Support System* allows the creation, monitor, and manage support or service tickets in case of errors or other issues. The system is mainly used to operate and coordinate an application support incident involving, software customer, platform provider, and ISV. It can be also used in case of incidents with the software platform where the ISV requests support.

The *Integrated Development Environment (IDE)* entails and encapsulates everything the ISV needs to develop his applications. Including a graphical editor, programming language interpreters, compilers, libraries, version control, source code management, but also tools for testing, transport, and component integration.

The *Application Runtime Environment (ARE)*, includes all system components which allow an application to be executed. Moreover, it provides tools for a high level application management, such as multi-tenancy management and tools to ensure non-functional application standards (e.g., security, reliability, and scalability).

The *Cloud Infrastructure* represents the lowest level of system infrastructure bundling all hardware and low level system management applications (system software, databases, middleware, operations and monitoring systems, disaster recovery, file hosting).

Two types of experts are relevant for the *Software Platform,* the *Platform Development Experts* and the *Platform Operations Experts.* The first group supports the ISV or internal customer throughout the development process of an application and provides enablement, training, and support services on an expert or very technical level.

The *ISV Portal* is used by the *Partner Management* as an information and interaction hub with the *ISVs.* It is a web portal with a public and protected or restricted access. The public part of the portal is mainly used to attract new *ISVs* and to satisfy their needs in that phase. The restricted part of the portal is used for existing partners providing them with information and tool access throughout their lifecycle (from development to sales and operations of an app). Beyond information, the portal is also used to automate standardized request processes (e.g., registering for an expert session, or requesting certification services).

Aside from the *ISV Portal*, the *Partner Management* also operates a *Developer Community*. This community is used to provide ISVs the opportunity to help themselves and connect with other experts in the software ecosystem. The community platform includes wikis, forums, and micro blogs.

The *Partner Manager* is a personnel resource acting as both sales agent towards *ISVs* and central contact person for all program related inquiries for the ISV. He provides support and guidance mostly remote-personal via phone or conference and has fixed ISVs assigned which he or she is responsible for.

The *3rd Party Application Certification Expert* is a specialized personnel resource dedicated to the process of certifying the *ISV* applications. They conduct the actual certification and provide certification training or consultations upfront to the *ISVs*.

The *App Store* leverages two resources to provide its services to *ISVs,* the *Public App Store* itself, the e-commerce application system, and a team of *Marketing & App Store Experts.* For *ISVs* the *Public App Store* provides a typical “seller interface” as part of electronic marketplaces. This includes capabilities to upload or edit an application to the store catalog, a (sales) order management, sales statistics, administrative functions, as well as payment and billing capabilities. Moreover, the *Public App Store* is also used to get an overview of the fees of the ISV program and related services.

The *Marketing & App Store Experts* support and consult ISVs on the sales and marketing via the *Public App Store.* Besides, when an *ISV* uploads an application they review the application marketing and catalog content and approve the final publication of the listing. The expert team also provides dedicated marketing and sales services associated with the *App Store* but which require personnel to be conducted (e.g., ISV co-branding, or curating a featured apps list)

### 5.4.2 Demand-Side Value Chain

The demand-side value chain models and recommendations have been derived using the findings from Papers 3, 4, 5, and 6 (cf. III.1, III.2, III.3, and IV.1). In particular, the structure which is developed by means of business process models (task level) and the proposed resource functionalities and capabilities, address the behavioral determinants for the adoption of the app store model by software customers introduced in the Papers 3 to 5. The drivers and barriers were grouped in customer-related, solution-related, and transaction-related determinants. At the same time the recommendations provide a possible solution for realizing the value propositions postulated previously. The results from Paper 6 already represent a preliminary step towards the solution space which is taken up, enhanced and completed with this section.

#### 5.4.2.1 Demand-Side Value Chain: Task Level

Figure 54 presents the simplified SOM IAS[1] with the platform provider as the universe of discourse focusing on the demand-side value chain. The IAS for the demand-side is compatible with the IAS of the supply-side (cf. Figure 50). However, the decomposition focus is on uncovering the major business objects and transactions relevant to the demand-side value chain. Hence, the supply-side business objects and transactions remain on a higher aggregation level to improve the representation of the graph and reduce complexity. The same applies, but vice versa, to the IAS graph presented on the supply-side (e.g., *Platform & Product* business object is further decomposed on the supply-side into *Core Products* and *Software Platform*).

[1] see also short explanation of the SOM IAS graph in 5.4.1

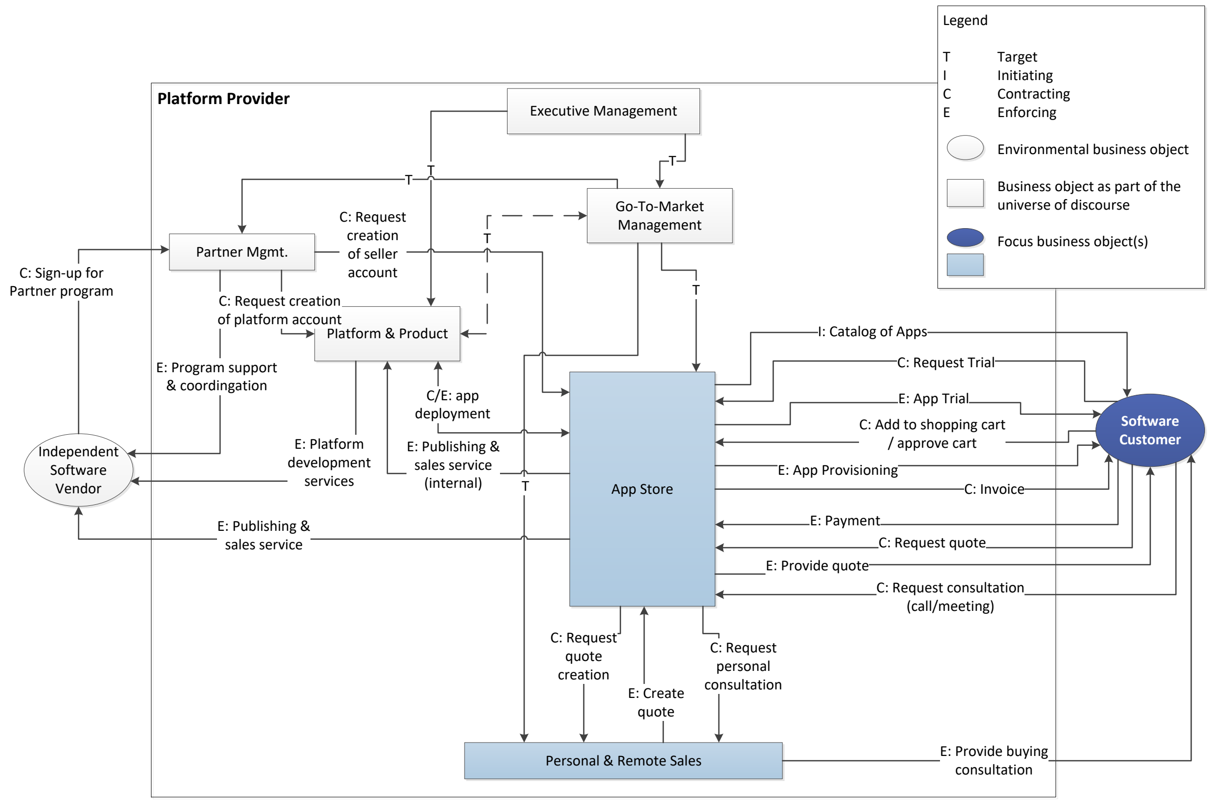


**Figure 54:** Demand-side SOM interaction scheme (simplified)

The description of the *Executive Management* provided on the supply-side model also stays valid for the demand-side (please refer to 5.4.1).

*Product & Platform* receiving targets from the *Executive Management* is an aggregate business object which entails the three business objects, Platform & Product Management, Core Products, and Software Platform introduced in the supply-side model. The description of these three business objects also applies to the *Product & Platform* object (cf. 5.4.1).

The go-to-market unit has been more detailed than in the supply-side IAS and consists of the *Go-To-Market Management* receiving targets from the *Executive Management* and coordinating the *Partner Management*, the *App Store,* and the *Remote & Personal Sales* with targets. An overview of the business objects with the description of their task responsibility focusing on the demand-side is presented in Table 24.

The major business objects within the *Platform Provider* directly involved in transactions with the *Software Customer* on the demand-side value chain are the *App Store* and the *Personal & Remote Sales* which are highlighted in light blue in Figure 54.

**Table 24:** Business objects of the platform provider (demand-side focus)

| **Business object** | **Description (task cluster)** |
|---|---|
| Executive Management | ▪ define and adjusts company targets and reviews target achievement<br>▪ assign sub targets to business objects |
| Go-To-Market Management | ▪ sales and marketing management<br>▪ closely aligns targets with product & platform (management)<br>▪ define targets for the app store<br>▪ define targets for partner management<br>▪ defines targets for sales and remote personal sales |
| **App store** | ▪ provides the online store front to software customers and presents the entire portfolio of applications of the software platform ecosystem<br>▪ supports the software customer throughout the buying process (problem recognition, information, evaluation, purchase, provisioning, and settlement)<br>▪ defines app store capabilities and implements e-commerce functions<br>▪ curates and manages app catalog<br>▪ operates app store and provides related services<br>▪ provides electronic "on-behalf" marketing, sales, and distribution services to internal product units and ISVs |
| **Personal & Remote Personal Sales** | ▪ consultative sales approach<br>▪ sells complex, high-price applications<br>▪ supports app store sales with consultative services (e.g., calls, meetings)<br>▪ creates quotes |
| Partner Management | ▪ define details on development partner program<br>▪ attract new partners ("partner sales")<br>▪ inform partners<br>▪ support partners with coordination throughout their lifecycle<br>▪ administrative function towards partners (i.e., contract management) |
| Platform & Product | ▪ define built, and operate software platform and core products<br>▪ provide platform services to internal and external customers (incl. ISV)<br>▪ deploy and operate (cloud apps) ISV applications for software customers<br>▪ provide first level support for application issues for software customers |

The *App Store* is the central business object in the demand-side value chain. Its tasks entail the collective marketing, sales, and distribution activities towards *Software Customers*. First and foremost, the *App Store* provides the customer with an e-commerce system, an "online store front", including a catalog with all the applications of the ecosystem (precisely those which are chosen to be marketed via the *App Store).* The e-commerce system supports the *Software Customer* throughout the buying process which is best divided in six phases as depicted in Figure 55.


**Figure 55:** Software customer buying process phases

The *App Store* covers the following tasks in the six process phases:

- In the *Problem Recognition* phase the *App Store* provides contextual product or application recommendations to users in order to create awareness and educate users on solutions for their demands.
- In the *Information Search* phase, the *App Store* provides different levels and types of information within the application description to satisfy the different information needs of the *Software Customer's* buying center members, ideally in such a way that additional personal consulting is not needed. Especially for complex EAS, multifold measures need to be undertaken to achieve this.
- The *Evaluation* phase is highly critical, since all aspects of the intended purchase need to be evaluated, first and foremost of the application's capabilities. Application trials are the means to best meet these needs. However, also beyond the functionalities, the *App Store* needs to provide means to ensure the application is compatible with the *Software Customer's* system and application landscape, ensure the timing until the application can be productive fits the customer's schedule (keyword implementation and service planning), and that the pricing is in line with a budget. These various aspects are mostly evaluated by a group of individuals and every key person on the *Software Customer* side needs to be involved in order to guide the customer towards a positive buying decision.
- Beyond the typical shopping cart and checkout features, the *Purchase* phase should provide capabilities to comply with corporate procurement regulations, e.g., respect corporate buying authorities, or provide information and document exchange with and to existing customer procurement systems.
- In the *Software Provisioning* phase, the application is delivered to the buyer or users, and depending on the application type different provisioning processes have to be covered. For cloud applications the *App Store* requests deployment and activation of the application instance from the *Platform & Product* business object. Moreover, the applications operations service provided by the *Platform Provider* incl. support processes to the *Software Customer* in order use the application productively, is also assigned to this phase. Also the later introduced "Internal App Store" service provided to the *Software Customer* is part of *Software Provisioning* in the larger sense.
- In the *Settlement & Aftersales* Phase the invoicing and payment processes will be conducted. Moreover, the *App Store* provides processes to adjust existing contracts for already procured applications (e.g., license termination or enhancements).

*Personal & Remote Sales* often represent an alternative separate sales channel. The configuration of the overall sales system of the *Platform Provider* depends on many different aspects and is not elaborated further in this chapter. However, in the context of the online channel materialized by the *App Store*, the *Personal & Remote Sales* acts as a supporting business object. One of the major marketing and sales targets is to convert prospective customers from one status to the next one until a customer ultimately buys the product. Hence the possibility that the customer "drops off" must be avoided. In case the *Software Customer* has a need for additional consultation which cannot be covered by the *App Store*, the *Personal & Remote Sales* should take over, either by providing personal consultation or by providing a quote based process to satisfy the need for non-standard contracts or terms. The involvement of *Personal & Remote Sales* is coordinated by the *App Store* via pre-defined "exit points" in the business process (e.g., request consultation, request quote).

The transactions between the *Platform Provider* and the *Software Customer* have been, as with the supply-side analysis, further detailed and analyzed. In total 39 transactions have been decomposed and grouped in the buying process phases as introduced previously (Figure 55). The tabular overview is presented in Appendix G and also includes an assignment of the resources which will be introduced later in this chapter (see 5.4.2.2).

**Inside the Software Customer**

By focusing on the *Software Customer* we can investigate further aspects which were to be addressed by the App Store Model for EAS, i.e., to better involve business stakeholders while respecting the IT and procurement guidelines of the company. Hence the *Public App Store* should recognize different corporate app store roles at different levels of permissions, and support adequate approval processes. A simple approval process with two different user roles is depicted in Figure 56. The *Business User* can access browse, evaluate apps, and put them in the shopping cart. To complete the checkout, the *Store Buyer* is contacted for approval. Once he or she has provided the same, the application is provisioned to the *Business User.*

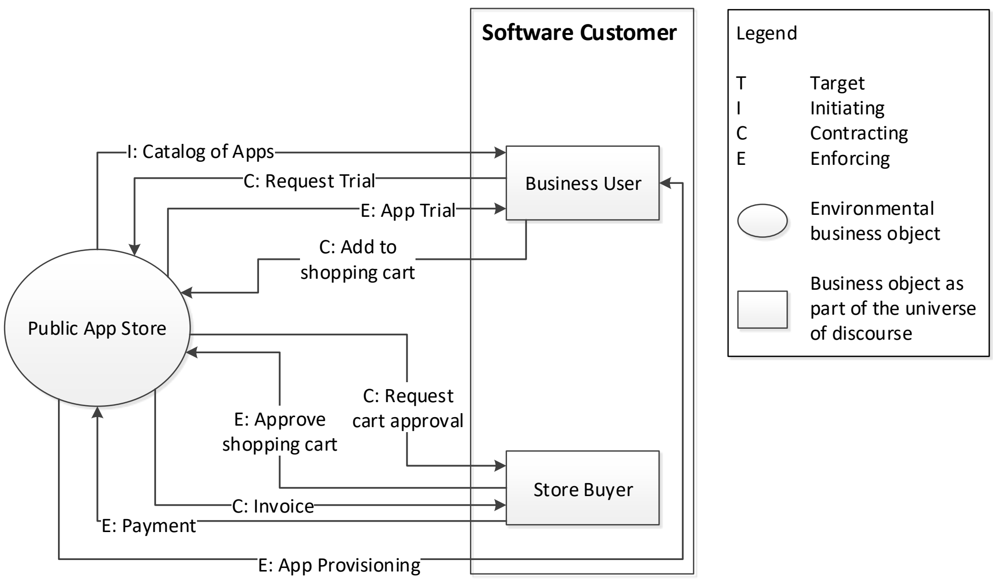


**Figure 56:** Approval process with the public app store (SOM IAS)

An optional enhancement to the *Public App Store* represents the concept of the *Internal App Store.* Whereas the *Public App Store* focuses on the procurement of new software “into” the portfolio of a company the *Internal App Store* introduces standardized business processes to mainly distribute personal business applications to *Business Users.* Figure 57 shows a possible use of an *Internal App Store* within a *Software Customer* organization. The *Internal App Store* can hence be understood as an extension of the business processes defined by the *Public App Store* into the *Software Customer* organization standardizing, and digitizing (automating) the *Software Customer* internal business processes to distribute and manage personal applications internally. The *Internal App Store* is provided as a cloud service by the *Platform Provider* and fully controlled and managed by the *IT Portfolio Management.* The *IT Portfolio Management* coordinates three subordinate business objects, the *Internal App Store Administration,* the *Internal Software Development,* and the *IT Purchasing.* The *Internal App Store Administration* administers and configures the *Internal App Store* according to the guidelines from the IT Portfolio Management. (e.g., which user roles with which permissions are to be maintained, and which catalogs with which apps are visible to which user groups). The *Internal App Store* catalog can be filled with any kind of application, standardized applications, applications from *3rd Party Vendors*, from the *Platform Provider*, or internally developed applications. Hence the *Internal Software Development* can also upload their apps to the *Internal App Store*. For standardized enterprise software, the *IT Purchasing* receives a request to procure business applications from the *IT Portfolio Management.* The desired applications are either procured from the *Platform Provider* or a *3rd Party Vendor,* and then uploaded to the *Internal App Store*.

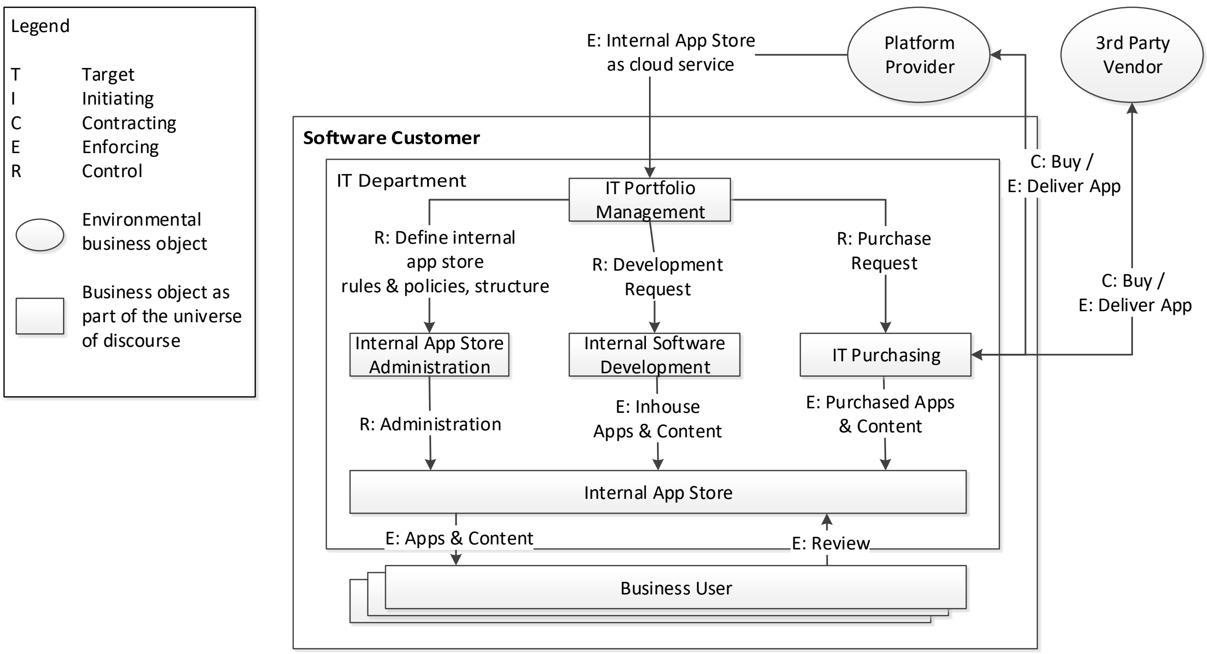


**Figure 57:** Internal App Store (simplified SOM IAS)

Ultimately the *Business Users* depending on their role can access the *Internal App Store* and are provided with a role specific catalog. They can select apps which are then distributed to the respective devices (mobile or desktop) or simply activated in the case of cloud applications. The *Business Users* can also provide application reviews to provide feedback to the *IT Department* and other users on the usefulness of the applications.

The concept of *Internal App Stores* supports the business processes following those of the public app store.

Figure 58 shows the business processes supported by *Public* and *Internal App Stores* in the form of sequential process phases. After the "external sourcing", using the *Public App Store*, is completed, the *Software Customer's IT Department* configures the app for internal use (e.g., app customization and creation of policy files) and publishes (*Internal Publication*) the app on the *Internal App Store.* Afterwards employees or business users use the *Internal App Store* to inform themselves about and evaluate personal applications. The *Business User* can then select and download apps*.* Also pushing distribution via existing MDM infrastructure is an option which should be available for configuration on certain apps. In the *Maintenance & Monitoring* phase, standardized processes and capabilities ensure updates are distributed or retired apps are delisted from the catalog and deactivated (or uninstalled) so that they can't be used any longer. Moreover, app downloads and usage can be monitored by the *IT Department.* With these business processes the *Internal App Store* provides more flexibility to business users while implementing control mechanisms to support the objectives of IT Governance.

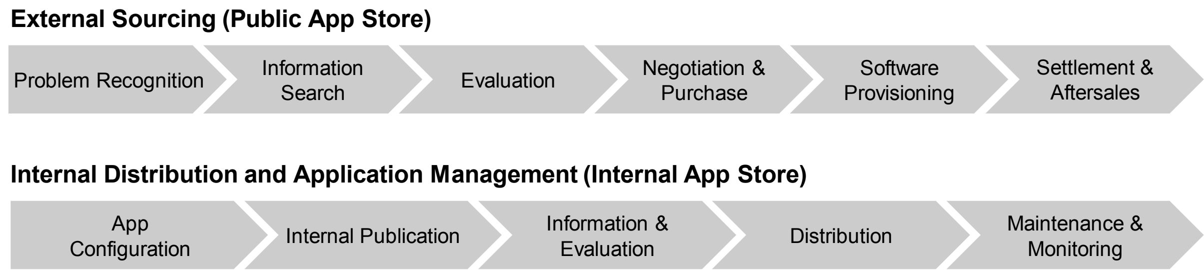


**Figure 58:** Business processes supported by public and internal app stores

To fully digitize and standardize the acquisition and the delivery down to the individual business user and to automate control aspects of IT Governance, the business processes of the *Public* and *Internal App Store* should be tightly integrated. A design proposal for this aspect is already recommended as part of Paper 6 (cf. IV.1) and is therefore not repeated here.

#### 5.4.2.2 Demand-Side Value Chain: Resource Level

As with the supply-side analysis, based on the transaction analysis and models, the resources, their capabilities and functions have been derived and matched to the respective business objects. The assignment of the resources to the individual demand-side business transactions and the detailed capability and function lists are presented in Appendix G.

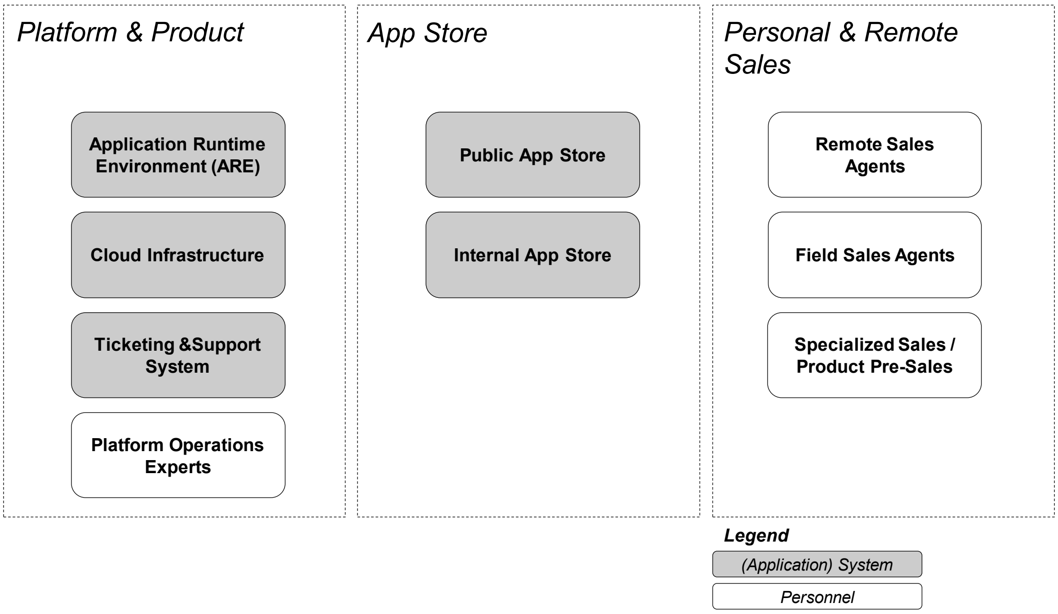


**Figure 59:** Resources of the platform provider (demand-side relevant)

Figure 59 depicts the resources relevant to the demand-side within the *Platform Provider.* Three business objects are delimited at this decomposition level to which the individual resources have been assigned: the *Platform & Product,* the *App Store*, and the *Personal & Remote Sales.* The *Internal App Store* represents a special case, since it is a resource mainly used for tasks within the *Software Customer* organization. However, since it is provided as a service from the *Platform Provider* it has been also listed as one if its resources. It will be described at the end of this section.

The *ARE* and the *Cloud Infrastructure* are used to operate cloud applications for the *Software Customer.* Besides, the *Cloud Infrastructure* also hosts files which need to be downloaded.

The *Ticketing & Support System* is used by the *Software Customer* to raise application support incidents either during the implementation or productive usage of an application. First level support is then provided by *Platform Operations Experts.* In case they cannot help, further resources are contacted (i.e., product specialists or developers from the *Platform Provider* or *ISV*).

The *Public App Store* is the central resource on the demand-side and supports the entire customer buying process. Compared to consumer app stores it needs to cope with specifics from corporate buying and more complex software applications. The capabilities are described following the six buying process phases and one administration category:

- *Administration:* The *Public App Store* needs to provide an account management on the level of companies which also allows the maintenance of frame contracts incl. company-level discount schemes and SLAs. Moreover, corporate user management needs to ensure that every user on the store are assigned to an account (company-level). The user management should support different user roles and permissions, e.g., users with buying authority and other users (e.g., business users) without buying permissions.
- *Problem Recognition:* Product recommenders should proactively provide app recommendations to users based on their context (e.g., work context, company-context, and current trends in the customer industry). Capabilities at this stage pro-actively push information to the users. Dedicated in-product recommendations, for example, recommend product add-ons or enhancements directly in the users' work context.
- *Information Search:* In this stage users are actively "pulling" and seeking information about applications. The catalog allows multiple access ways, and structures the applications in multiple dimensions to browse, search, and filter the portfolio. The products / applications are presented with different levels of information to satisfy the different information needs of different user types (e.g., end user, manager, IT staff, and controller). The product presentation hence includes textual information and multi-media content, e.g., to showcase product previews or videos). Embedded chat capabilities let users connect with *Remote Sales Agents* to clarify basic questions with regards to the *Public App Stores* store front or the product portfolio. For more complex inquiries the *Software Customer* can also request a call-back to elaborate a problem via phone or video conferencing with a *Remote Sales Agent.* Moreover, for complex core products, regular online webinars or live Q&As should be offered to satisfy detailed information needs about product capabilities. The store should provide an online registration tool for this purpose.
- *Evaluation:* The *Public App Store* provides the option to request application trials. Trials are real systems, but their use is limited in some ways (e.g., time, functionality, number of users). Additionally, a trial provisioning process needs to be supported (e.g., activate trial account, request trial deployment from the cloud infrastructure). Complementary to a trial is the demo. For the most part, the demo application is not the "real" system. The *Public App Store* provides a demo environment which can host demos with guiding information and controlled interactivity to allow users to also evaluate complex applications quickly. For complex applications dedicated scoping wizards allow the configuration of the required functionality and at the same time assess individual app capabilities. In case applications are add-ons to core products, a compatibility check can evaluate online whether all pre-requisites are fulfilled to use the add-on. Additional updates or technical adjustments might be necessary which the *Software Customer* needs to consider during the *Evaluation* phase. Complex applications also require a dedicated implementation phase to be introduced

productively. Service planning assistants as part of the *Public App Store* automatically generate an implementation plan with a project plan and a proposal on packaged service offerings. The results of the service planning, compatibility check or the scoping wizard can be shared with colleagues for review and feedback. In case further clarifications are needed, the *Public App Store* can be used to book on-site workshops with *Field Sales Agents* or *Product Pre-Sales*.

- *Negotiation & Purchase:* The pricing engine of the Public App Store recognizes different kinds of discounts (volume, bundle, or customer individual discount schemes) and applies country specific taxation. Moreover, complex products require a multi-dimensional pricing configuration on the different service elements of a product (e.g., number of users, capability levels, SLA levels). Furthermore, existing frame contracts may additionally impact the pricing or even inherent product elements such as SLAs. Hence the *Public App Store* needs to check and validate the existing contracts and their dependencies to the selected products and apply the relevant elements. A shopping cart supports the selection of multiple products from multiple vendors, while at the same time the shopping cart is a means of confirming a previous pricing or solution configuration. Different approval processes should be supported depending on the *Software Customer's* buying policies. Either online approval workflows using the different corporate buying roles or by transferring the shopping cart to the *Software Customer's* procurement system for approval. Once the shopping cart is approved, the checkout process completes the purchase and the related order documents are created and the buyer user and connected procurement systems are notified. As an alternative to the standard buying processes, a dedicated electronic quote process should be supported. For dedicated products or shopping carts, quotes can be requested with the option to describe the reason for the quote request and the desired adjustment (e.g., price, SLA, terms). *Remote Sales Agents,* receive the quote request and can electronically respond. *Software Customers* can review, share, accept or reject the respective quotes in a quote inbox.
- *Software Provisioning* depends on the type of product and the technology. An interface to the *Cloud Infrastructure* triggers the delivery processes. For cloud-based applications the *Public App Store* requests the deployment of the cloud app and generation of the initial user. The information (link to the application) and user information is collected and distributed to the *Software Customer.* In the case of on-premise applications, the generation of download links is requested and download permissions are assigned to the buyer. For personal applications, direct downloads to the device (desktop or mobile) need to be supported. Aside from the actual software delivery, delivery notification mails are sent and contractual and license data is updated. For complex products, an implementation guide is generated based on the selected scope and existing system landscape which

helps the customer to perform the necessary activities. The guide is then distributed with the delivery notification mails.

- *Settlement & Aftersales:* To commercially manage the purchased goods from the *Public App Store* a purchase order overview lists the different purchases with all products linked, a contract management overview focuses on the terms and conditions and SLAs and their validity and the invoice overview lists all invoices and their payment status. As payment, either corporate credit cards or traditional bank transfers should be supported. The license management gives an overview of the purchased licenses and the licenses in use. It further allows the purchasing or subscribing of additional licenses of already purchased products or terminates license subscriptions that comply with the defined termination periods.

Typically the *Personal & Remote Sales* form a dedicated sales channel. In this thesis only their support function towards the *Public App Store* is considered. Three different sales agent roles are delimited: the *Remote Sales* Agent, the *Field Sales Agent,* and the *Specialized Sales or Product Pre-Sales* Agent. The interaction with these agents is facilitated by the *Public App Store* with respective request functionalities as described above.

The *Remote Sales Agent* supports the customer remotely using phone, chat, or conferencing tools. Typically, *Remote Sales Agent* covers basic questions with regards to the product portfolio or the buying process on the *Public App Store*. Moreover, *Remote Sales Agents* can create quotes upon request or support with administrative tasks (e.g., assignment of user permissions).

*Field Sales Agents* are travelling sales agents and conduct on-site customer meetings. Usually, *Field Sales Agents* are senior agents with a sound and comprehensive overview of the product and service portfolio. They too create quotations and may also close contracts offline in case the customer chooses not to use the *Public App Store.*

*Specialized Sales or Product Pre-Sales* are asked to support customers either remotely or on-site on specific product questions which need to be clarified in order to make a purchase decision. Hence, they have deep product knowledge and are typically responsible only for a single or a few products. *Specialized Sales* is a role to be covered by both the *Platform Provider* (for core products) and by *ISV* for their products.

The capabilities of the *Internal App Store* is grouped in five sub-categories:

- *Setup & Administration* capabilities allow the visual appearance to be adjusted to reflect a corporate identity (e.g., colors, logo). Most importantly, however, is the configuration of the catalog (definition of catalog categories, catalog variants) and the definition of user roles and permissions.
- *App Configuration & Publication:* A user interface allows the maintenance of new apps in the *Internal App Store* including an editor to maintain the presentation content (e.g.,

description, screenshots, video files), meta-data (e.g., catalog categories, compatibility), as well as maintaining available licenses or user seats. Additionally, delivery relevant application configuration files can be uploaded which need to be distributed to the users (e.g., policy files). Once maintained, a workflow supports the publication request and the approval of publication.

- *Information Search & Evaluation:* Similarly, to the *Public App Store* the *Internal App Store* provides user discovery features to browse, search, and filter the application catalog. The presentation of the app includes textual information, screenshots, app ratings and reviews as well as device compatibility information and system pre-requisites.
- *Distribution:* Mobile or desktop apps are instantly delivered to the respective device. More complex applications are provided via download links with further information in installation. For cloud or web apps the user is generated and activated. In case individual licenses are used by the app consumption, the license or subscription seat is assigned to the respective business user. In case an MDM system is in place, the *Internal App Store* can also trigger the delivery via the MDM system and exchanges the required information via adequate system interfaces.
- *Maintenance & Monitoring*: The *Internal App Store* can be used for app version and update management and to automatically distribute updates or notify users on updates. Moreover, functionality to manage app retirements and replacements should be available. From the commercial perspective the license overview shows the available and used user licenses or subscription seats, and also notifies the administrator in case certain products run out of licenses. Statistics on user downloads and app usage further provide information to the administrator.

### 5.4.3 Product Characteristics for the App Store Model

The product model of a business model as defined by Stähler typically describes the service elements of a product or product bundle the focal company wants to produce for its customers (Stähler 2002, p.43). The product model in the case of the App Store Model for EAS, however, does not define the actual functionality of the products, but looks at non-functional product characteristics of EAS. These characteristics influence whether a software customer is willing to purchase the respective product using an online channel, i.e., the app store. In particular, the behavioral models from Papers 3, 4, and 5 uncovered the importance of the actual product characteristics on the software customer's decision to use the app store as the buying channel (see also Figure 44 for the "solution" related determinants). The previous section has proposed business processes and functionalities to overcome or cope with "negative" product characteristics (e.g., scope wizards for products with a large scope), contrariwise the methods proposed in this section attempt to improve the actual non-functional product aspects thus making the barriers obsolete or reducing their negative effects. Paper 7 (cf. Chapter V.1) has already presented multifold options in this regard, hence this section will further generalize the concepts and highlight the most important ones.

Before looking at the product characteristics in more detail, I want to offer some preliminary remarks on the overall structure of the product portfolio in the App Store Model for EAS. As already introduced previously, the portfolio of products consists of one or more “core products” and many “complementary products”. The core products are developed by the platform provider, and the complementary products by the ISVs. From a functional perspective this delimitation ideally follows the idea of the core-shell model introduced in Chapter 2.2.2: The core product or application covers the functionality most customers require, and the complementary applications focus on more specific functionality and needs. While this is the guiding idea of the product positioning, it should not serve to derive general technical dependencies or contractual obligations: for example, ISVs can also provide standalone applications which do not require a customer to own one of the platform provider’s core applications, and at the same time the platform provider might also develop apps with a very specific niche functionality or add-ons to his core product.

Independent of being a core or a complementary product, the five most important enterprise application characteristics coping with the barriers of choosing the app store to buy EAS are the following:

- granular application scope
- trialability
- starter package
- minimum infrastructure
- instant use.

These characteristics are also referred to as “app’ification patterns”. The metaphor app’ification comes from applying characteristics associated with consumer-oriented “apps” available in “mobile consumer app stores” to other domains or contexts (e.g., Kosner 2012). In this thesis I use the metaphor to describe patterns making EAS products ready for the app store model.

Beyond these five product patterns, Paper 7 has investigated further characteristics: business user driven adoption and e-commerce completeness. “E-commerce completeness” mostly refers to the capabilities of the app store based business processes and capabilities, and was already covered by the previous section on the demand-side value chain. The aspects investigated with the “business user driven adoption” criteria partially affect the actual functionality of an application which was explicitly excluded from discussion at this point since the decision regarding which applications are developed to serve market needs are beyond the scope of the App Store Model for EAS. Another aspect reviewed there is the application user interface simplicity or attractiveness. Notwithstanding that an appealing user interface may be a convincing aspect in the adoption decision of an application, with regards to the channel adoption decision it is merely relevant for the evaluability of an application and is discussed as part of the trialability pattern.

**Table 25:** Overview app'ification patterns and their rationale

| # | Pattern | Determinants of adoption | Reasons |
|---|---|---|---|
| 1 | Granular scope | Scope, number of users, (evaluability, price level, customization /implementation effort) | ▪ reduction of overall complexity and heterogeneity of application<br>▪ reduction of associated buying center size / involved decision makers<br>▪ less overall risk and lower entry barrier |
| 2 | Trialability | Evaluability | ▪ allow to assess the applications by the actual business users<br>▪ allow customer to evaluate with his data and context<br>▪ enable evaluation of complex scenarios which cannot be evaluated fully by descriptive feature list<br>▪ reduce the need for trust in descriptive information<br>▪ reduces need for consultancy (personal) |
| 3 | Starter package | Price level | ▪ reduce price complexity and reduce the risk for hidden cost<br>▪ reduce entry barriers for new customers with lower prices and less need for consultancy |
| 4 | Minimum infrastructure | On demand delivery | ▪ less upfront investments<br>▪ less company internal and system landscape related cross-dependencies |
| 5 | Instant use | Customization / implementation/ integration effort | ▪ better "time to value" for the customer<br>▪ avoiding risks of long-lasting expensive implementations |

Table 25 lists the five product patterns and summarizes their respective rationale, first by listing the respective solution-related adoption barrier from the adoption model introduced with Papers 4 & 5 (cf. Figure 44) and complemented with additional arguments and explanations. In Appendix H, all concrete measures for implementing the five different patterns are listed. Moreover, which of the three product facets are affected by the measures (commercial, functional, technical product facet) are categorized. In the following, the five patterns and selected important measures will be briefly described.

*Granular scope* is potentially the pattern with the most influence, since it not only directly affects the adoption decision according to the adoption model, but also indirectly other solution-related determinants. Though the adoption model has only clearly revealed the influence on the number of users, there are logical reasons why others are influenced too. A more granular scope reduces the overall scope to be evaluated by the user and hence should simplify evaluability. Applications with lesser and more granular scope are also typically priced lower and, moreover, if there is less scope or less use cases approached by an application, there should also be less effort or need to customize the respective functionality. I conclude that a granular scope is in general associated with a lower risk, and the application complexity and heterogeneity with regards to its requirements is reduced. If there is less scope covered, the business users involved should consequently also be reduced which influences the size of the buying center and the number of decision makers.

Measures for achieving a more granular application scope or making an existing one more granular can be allocated to the functional product facet and should be applied by job roles such as product designers, product managers, or product architects. First, a user centric design allows an application to be designed and developed from a single user role's perspective and hence results in an application were only one user role within a company is addressed. Assuming one user role within a company has similar requirements compared to the requirements towards an application with multiple roles involved, this should reduce buying decision complexity. Similarly, a task-oriented design defines the scope of an application around a single or few connected business tasks. Another option is to allow so-called partial use scenarios of an application leaving other parts of the application untouched or inactive. Since a partial use scenario is not simply given in complex integrated applications it requires dedicated enablement. Last, existing applications can be dis-aggregated or "incremented", meaning the application's core use case is separated from all other functionality which is encapsulated in enhancements or add-ons.

The *trialability* pattern copes with the evaluability determinant of the adoption model. If enabled it allows a business end-user to directly evaluate the application. Moreover, customers can assess the application using their own data and also try more complex scenarios. Overall trialability reduces the need for trust in the provider's promises and propositions or the need for personal consultancy. The trial strategy should be tailored to the product and needs in order to recognize the application scope and complexity. Typically to provide these trials, all major facets of a product need to cooperate: the commercially responsible persons, the function-oriented, and technical ones. Regardless of this, different levels of trials should be provided – this can be also referred to as the "trial cascade". At the lower end of the trial cascade are *screenshots* and *demo videos*. Though they are not typical trials, they allow the users to become familiar with the application's key features while seeing the application "in action". The next level, the *guided tour* is a very limited system in which the user can interactively experience the application and the major use cases by following guides or guiding hints on the user interface. The *shared trial* is a fully functional system, which the user, however, shares with users from other companies. A *private trial* is hence a trial which is only available to one user or users of one company. This trial may even be used for some productive cases. The *custom trial* may be an option for very complex solutions where the provider has customized certain aspects of the trial for a single customer. Since this includes a high degree of effort, it is potentially not used very broadly. Last, a free edition of an application is also an option to provide trialability. Free editions allow productive use but are limited in some way (e.g., time, scope, number of users).

The *starter package* addresses the price level determinant from the adoption level. Moreover, it also reduces complexity, since the starter package bundles everything the customer needs to use the application productively. Combined, the starter package is a means to significantly reduce the entry barrier for a product and the need for consultancy. A starter package should bundle everything the customer needs, this potentially includes other products or services (e.g., implementation service, support service). Besides, the different elements should be included in a single price to reduce complexity and consulting need associated with multi-dimensional pricing schemes. A subscription and user or usage-based pricing moreover reduces barriers connected with large upfront investments.

The *minimum infrastructure* pattern reduces the need for additional internal infrastructure approvals or entire teams to be involved or even setup (to procure, setup, maintain and operate the infrastructure). It correlates with the on-demand delivery determinant of the adoption model and is allocated with the technical product facet. Consequently, a cloud-based version or an application according to SaaS addresses this aspect. However, enterprise applications often cannot be operated fully isolated since they rely on data from other systems and thus ought to be integrated. Therefore, integration middleware should also be provided as a cloud service. An existing legacy on-premise application may be provided combined with an IaaS to avoid on-premise infrastructure and bundled as one offering. In case on-premise hardware cannot be avoided (e.g., machine control applications), the application can be bundled with the hardware in the form of a so-called “appliance”. Maintenance can be provided remotely with the adequate infrastructure connected to the appliance.

Last, the *instant use* pattern concerns the integration and customization effort and affects both the functional and technical product facets. It goes without saying that applications which need no or minimal implementation and customization effort have a better time-to-value ratio and involve less risk. Measures include automating implementation and customization tasks as much as possible. Another option is to provide pre-configured systems or configuration templates covering the “best practices” or typical use cases. Moreover, business-language oriented configuration guides reduce the need for technical staff and specialized consultants. A major hurdle getting business applications up and running is data migration and integration. Hence wizards or migration assistants should support this step, and if such support is needed it should be provided flexibly via remote infrastructure.

## 5.5 Revenue Model

The revenue model analyzes the sources of revenue of a business model. These revenues are then contrasted with the cost of the value architecture and determine the margin structure of the App Store Model for EAS (Stähler 2002, p.47). Four distinct revenue streams have been derived mainly using the results from Paper 1 (cf. II.1) and confirmed by work by Giessmann and Popp and Meyer (Giessmann 2015, p.43; Popp & Meyer 2010, p.134) and presented in Figure 60. It has to be noted that the "technical" flow of the payments may differ from the conceptual revenue flows, and the two aspects should not be mixed. For example, even if the complementary apps can be paid using an "app store collection service" from the software customer to the platform provider, the revenue stream still originates from the software customer and flows to the ISV.

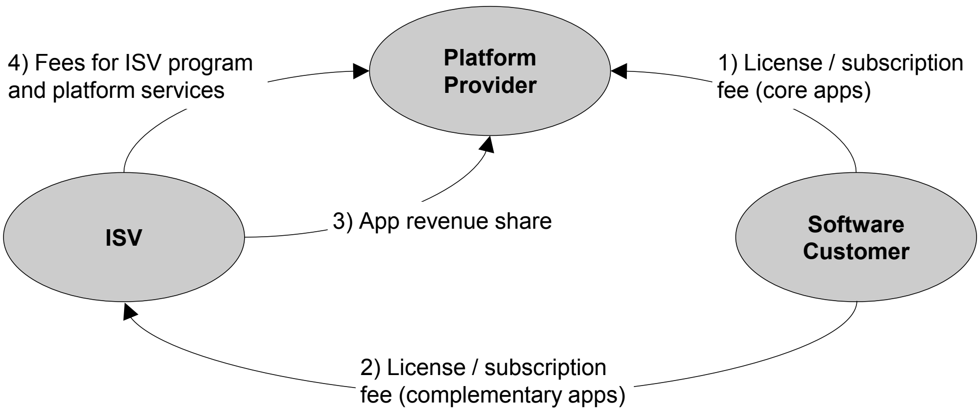


**Figure 60:** Revenue streams of the App Store Model for EAS

*1) License or subscription fee for core apps*: The platform provider receives a fee for its core products from the software customer. The fee is charged for the right to use the application plus additional associated services. These may include application support, updates, or hosting services. The price structure for this fee can vary in practice. Typical models are a monthly subscription fee per user seat for cloud-based SaaS applications or a so-called one-off upfront licenses plus annual support fees as a percentage of the license volume. This revenue stream is not newly created, assuming the platform provider is a vendor for EAS also without the app store model. But, as elaborated in the value proposition, the app store model provides the potential to increase the revenue for core products (cf. 5.3.3).

*2) License or subscription fee for complementary apps*: The fees for the complementary apps are the source of revenue for the ISV originating from the software customer. Its rationale and structure is similar to that for the core apps and its structure can also take on different forms in practice. In the light of the App Store Model for EAS, however, it has to be mentioned that a standardization of pricing structures can have a positive influence on the acceptance of the app store as an online channel. Hence the platform provider may decide to enforce certain pricing models for complementary apps to simplify pricing structures and thus provide more transparency and comparability to software customers. Also this revenue stream is not newly generated by the app store model, assuming the ISV would also develop and market software applications without being a member of the ISV program or the ecosystem, the App Store Model for EAS has the potential to increase this revenue stream for ISVs (cf. 5.3.1)

*3) App revenue share:* The revenue share is usually paid as a percentage of the fees for the complementary apps from the ISV to the platform provider. The fee is mainly reasoned by the intermediation service provided by the platform provider (on-behalf marketing, sales, and distribution via the app store). This revenue stream is a new one established with the App Store Model for EAS.

*4) Fees for ISV program and platform services* are paid from the ISV to the platform provider for the services bundled as part of the ISV program. Typically, the program covers access to technologies and resources. Certain services however may be charged additionally, e.g., such as app certification or the on-behalf operations of the ISV apps by the platform provider in the cloud model. Whereas program fees typically occur on a regular monthly or annual basis with a fixed price tag, additional services are often usage-based or have fixed price tags. This revenue stream is also newly generated by the App Store Model for EAS, since the platform and its services are an integral part of the app store model as previously defined.

If the costs and revenues are opposed, a revenue-related decision logic can be derived for the platform provider and the ISV. This logic is illustrated in Figure 61 using the introduced revenue streams and simple cost categories.

**Platform Provider**

| Cost | | Revenue |
|---|---|---|
| cost of platform | | ∆ license revenue (core apps) |
| + cost of app store | | + ISV program & platform fees |
| | | + revenue share |
| ∑ cost of app store model for EAS | < | ∑ revenue of app store model for EAS |

**ISV**

| Cost | | Revenue |
|---|---|---|
| cost of ISV program fees | | ∆ license revenue (complementary apps) |
| + cost of platform service fees | | |
| + ∆ app development, operations, & sales | | |
| ∑ cost of app store model for EAS | < | ∑ revenue of app store model for EAS |

**Figure 61:** Costs and revenues of the App Store Model for EAS contrasted

Consequently, from a revenue perspective the App Store Model for EAS is beneficial for the platform provider if the summed up revenues generated by the app store model exceed the costs arising from the same. In particular, on the cost side, the platform provider needs to finance the business processes and resources associated with the platform and the app store (simplified categories), while on the revenue side he needs to consider the delta (potential increase) in license revenue from the core apps plus the revenues from the revenue share and the program and platform fees. The ISV, in turn, incurs additional costs from the ISV program and the consumed platform services. However, he may also benefit from higher development, operations, and sales efficiency via the platform and the app store and thus decrease the cost of development, operations, and sales. Additionally, the ISV may benefit from incremental revenue for his complementary apps.

While the ISV relies on a single source of revenue to maintain his business, the platform provider receives revenue streams from the two sides of the two-sided platform. As introduced in Chapter 2.3.4 on multi-sided platforms, the platform provider needs to balance the two sides of the platform, since they cross-influence each other via the so-called network effects, which can be both an opportunity and a threat to the platform provider and the ecosystem – assuming the objective is to establish a sustainable ecosystem.

With the three potential revenue streams, multi-fold design options can be derived which provide the platform provider with options to influence the attractiveness of either side or to pursue certain strategies. For example, the platform provider may choose not to charge for the ISV program and only rely on the app revenue share to further reduce entry barriers for new ISVs. Vice versa, in case too many ISVs are on the platform or ISVs of questionable quality or intentions, raising the entry barriers via fees on the ISV program might only attract those ISVs with serious business interests. Cross-financing the one platform-side with the other is another interesting option. In case new customers for core products are the major goal, the platform provider may choose to reduce the price for the same and bet on new customers, and subsequently higher sales numbers for complementary apps and thus a higher revenue share. Obviously, these strategies may be adjusted carefully over time to suit the ecosystem's respective situation.

## 5.6 Expert Review and Reflection

The expert review and subsequent reflection is an inherent part of the design-oriented research methodology for this section (cf. Chapter 3.5 and 5.1), and is part of the evaluation phase of the research process as defined by (Becker 2010) (see also Chapter 3.5.1). The objective of the expert review is to assess the created model artefact with regards to its usefulness and appropriateness, and to derive options for future enhancements or revisions of the model artefact.

The criteria of usefulness and appropriateness has been further interpreted in the context of this thesis to allow the experts to provide precise feedback and thereafter offer a structured and focused reflection. Accordingly, usefulness and appropriateness can be best assessed along the objectives of the model artefact (cf. Chapter 5.1):

- The artefact defines the value proposition, the value architecture, and the revenue model of the App Store Model for EAS.
- The value architecture represents the core of the solution space to be developed, in the form of semi-formal model elements and design recommendations, which are capable of delivering the postulated value propositions for the respective actors and the revenue model.
- The solution space defines the structure in the form of the required business processes (i.e., business objects and transactions) and resources as well as descriptive recommendations.
- The model artefact should be abstract enough to serve beyond the scope of a single company or real-world context.
- It aims to help substantiate strategic concepts and serve as a basis for deriving more concrete business process models and application blueprints or requirement specifications.
- The model artefact should be practicable, i.e., it should be comprehensive, comprehensible, and detailed enough to help practitioners establish, enhance, or adjust an App Store Model for EAS in practice.

### 5.6.1 Expert Review Procedure

To conduct the expert review, a review presentation document including the artefact documentation was created (71 slides). In total five individual expert review sessions where conducted. The candidates for the expert review were selected based on their domain expertise in the area of EAS and app stores, or similar platforms for the same. They are all long-term practitioners, with four of them having an academic background (i.e., Ph.D. or other doctoral degree). At the time of the expert review, four experts were employed at SAP SE and one expert at Hybris AG (acquired by SAP SE, but operated as an independent business unit (SAP SE 2013)). Table 26 provides an overview of the professional profile of the candidates. Column two indicates their overall professional experience and their professional experience in the software industry (in brackets). Furthermore, their job title is shown, their function, and a short description of their job responsibilities.

**Table 26:** Overview expert profiles invited for review

| # | Years (SW) | Job Title | Job Function | Job Responsibilities |
|---|---|---|---|---|
| **1** | 18 (18) | Head of Digital Platform | Manager, Expert | Definition of software requirements for the SAP Digital Platform. SAP Digital Platform is the information system supporting the direct online sales and marketing processes for a selected portfolio of enterprise application software or software technology. |
| **2** | 14 (11) | Vice President, Platform Partner Services and Business Models (Platform Partner Ecosystem) | Manager, Expert | ▪ definition of solution partner models/programs, frameworks, contracts, and business models<br>▪ definition partner services and SLAs<br>▪ roll-out of new programs, services, or program updates |
| **3** | 16 (10) | Global Lead Platform Ecosystem Business Model | Expert | ▪ definition of partner contracts, partner program pricing and packaging<br>▪ contract negotiations |
| **4** | 19 (12) | Former (relevant) position: Head of SAP Store Solution Management<br><br>Current: Head of Business Unit (SAP Exchange Media) | Manager, Expert | w/r to position "Head of SAP Store Solution Management":<br>▪ definition of requirements to enable the SAP Store market and sell SAP solutions online<br>▪ coordination of development of the SAP Store<br>▪ curating and management of the SAP Store product portfolio |
| **5** | 12 (12) | Product Manager & Product Owner for Hybris, i.e., Hybris-as-a-Service Marketplace | Expert | ▪ definition of software requirements for a micro-services marketplace<br>▪ co-ordination of software development teams |

The presentation document was electronically sent to the experts before the interview to allow them to familiarize themselves with the comprehensive documentation. All invited experts accepted the invitation. The review sessions were initially scheduled for 90 minutes each. Two sessions were extended ad-hoc to approximately 120 minutes to allow sufficient time for feedback and discussion. The review was structured in three parts:

- introduction (planned 10 min)
- presentation of results (planned 60 min)
- feedback questionnaire (planned 20 min).

The introduction part (17 of 71 slides) made the experts familiar with the objective of the review and the overall context of the research. Moreover, the expert review procedure was explained including a guarantee of individual anonymity. Last, the experts were asked to provide information on their professional profile and a confirmation that this data, including all the feedback they provide, may be used as part of academic publications even if e.g., job titles or job responsibilities might be used to identify the respective person (all experts agreed).

The presentation of the results (50 of 71 slides) started with a brief overview of the research process leading to the model artefact as well as an introduction to several important definitions, pre-assumptions, and the structure of the model artefact. The model artefact was presented following along the three main components: value proposition, value architecture, and revenue model. The majority of time was spent on the value architecture part which was further clustered into the parts “Supply-Side Value Chain”, “Demand-Side Value Chain”, and “EAS Product Characteristics”. The value chain parts were further clustered in a task-level and resource-level analysis, as well as graphical schematizations combining the two.

The last part (4 of 71 slides) of the review was the feedback questionnaire where the experts were asked to provide structured feedback on the presented model artefact (see Appendix I for the full questionnaire). In total 14 statements in three groups were created to best assess the objectives of the model artefact, i.e., its appropriateness and usefulness. The statements were reflected on and the experts asked for qualitative feedback to the statements with regard to the presented content. The qualitative feedback was accompanied with a rating from 1 (fully disagree) to 5 (fully agree). The rating was not meant to be used in a statistical sense but to first of all help to calibrate and interpret the qualitative feedback, both the feedback given by one expert and also to calibrate the feedback given across the experts. Parts of the questionnaire could be filled during the interview. Every expert, however, took the time to walk through the questions again afterwards and make adjustments or provide additional notes and feedback electronically.

### 5.6.2 Results and Reflection

Table 27 shows the feedback questionnaire, i.e., the criteria to be assessed and the individual ratings by the experts. Since the numeric feedback was not to be assessed in terms of statistical significance, but rather to help calibrate the provided qualitative feedback and to give direction towards the overall appropriateness and usefulness, an interpretation of the numbers is not individually offered here, but only selective reference made to them as part of the qualitative feedback review.

**Table 27:** Feedback questionnaire and expert ratings

| # | Criteria to be assessed | Experts | | | | |
|---|---|---|---|---|---|---|
| | | #1 | #2 | #3 | #4 | #5 |
| | The design, tactical and operational recommendations ... | Rating 1-5 (5=fully agree; 1=fully disagree) | | | | |
| **Function of the app store model as business model** | | | | | | |
| 1 | ... help to establish an app store model in the enterprise application software (EAS) segment. | 5 | 5 | 4 | 5 | 4 |
| 2 | ... help to assess and improve an existing app store model in the EAS segment. | 4 | 4 | 4 | 4 | 5 |
| 3 | ... help to substantiate strategic concepts for an app store model in the EAS segment. | 5 | 4 | 4 | 5 | 5 |
| 4 | ... are a good starting point and help to derive more concrete business process models, service definitions, and functional blueprints. | 5 | 5 | 5 | 4 | 5 |
| 5 | ... address the presented value propositions for the different stakeholders adequately. | 4 | 3 | 4 | 4 | 5 |
| 6 | ... are concrete enough to help practitioners to derive meaningful action. | 5 | 5 | 5 | 5 | 3 |
| **Reach of model** | | | | | | |
| 7 | ... are abstract enough to serve different companies within the boundaries of the defined EAS segment. | 5 | 5 | 5 | 5 | 5 |
| 8 | ... are of high value for companies acting as platform provider. | 5 | 5 | 4 | 5 | 5 |
| 9 | ... are of high value for companies acting as ISV (independent software vendor) | 4 | 3 | 4 | 5 | 4 |
| 10 | ... are of high value for companies acting as software customer. | 5 | 4 | 3 | 5 | 4 |
| **Overall impressions** | | | | | | |
| 11 | ... overall are of high value. | 5 | 5 | 4 | 5 | 5 |
| 12 | ... comprehensive in the defined domain. | 5 | 4 | 5 | 5 | 4 |
| 13 | ... are well structured. | 5 | 5 | 5 | 5 | 5 |
| 14 | ... are easy to comprehend. | 4 | 5 | 3 | 4 | 3 |

Due to the large amount of feedback given, the focus here will be on the most relevant aspects and those aspects providing proper insight (e.g., leaving out comments such as “very good”, “good list and scheme”). As the experts provided the feedback in their own words (partially using keywords, partially using colloquial expressions or in German) they are reformulated here or grouped from individual statements where meaningful.

**Function of the app store model as business model**

The first set of statements tests if the model is useful for practitioners to establish, assess, or improve an app store in practice, in terms of deriving meaningful action or preparing detailed concepts and blueprints. The areas of improvement in this category are (the number in brackets indicates the number of experts who have mentioned an aspect related to this area):

a) incremental plan (3);

b) implementation guidance (3);

c) benchmark example with existing app store for EAS (1);

d) relevance of individual value propositions (3).

Three experts mentioned the need to stronger "increment" (a) the model elements to help them prioritize while making an implementation plan. This may also include a classification according to "must-haves", "should-haves", and "optional" elements. More specifically one expert wished to have an incremental plan with a "scope" to start with and then further options to enhance it. Related to these aspects was the next group of feedback provided, which was very much related to transferring the model elements into implementation blueprints (b). Hence, more guidance on the various alternative options in form of a questionnaire was mentioned as one possible enhancement. Another expert mentioned that the model describes very well the "what" – but for selected aspects a more detailed view on "how" to implement an aspect would certainly help. A third area of improvement was related to how to use the model to assess an existing implementation (c). In this context the expert proposed to provide an example "benchmark analysis" to an existing app store. Last, with regards to the presented value propositions, it was mentioned that in reality only a subset of the value propositions will apply to a certain company and that it may be helpful to further cluster the value propositions according to various dimensions to support strategic people in focusing on the right elements (e.g., revenue enabling, efficiency related, competitive differentiation).

The major strengths of the model in this category were:

a) model bridges strategy and implementation (2);

b) model is a collection of best practices (2);

c) demystifies the notion of app store (1);

d) model is comprehensive and has the right level of detail (5);

e) model structure and presentation (4).

On the upside, the experts mentioned the good fit of the model between strategy and implementation (a). To support this, one expert argued, that any digital strategy as well as any implementation of an app store for EAS needs to have proprietary elements in order to differentiate it from the competition, since in a global and connected world "platforms" will be faced with the "winner takes all effect". Consequently, the proposed model can serve as a guide to implement such a proprietary differentiation, and to set realistic goals. Another one mentioned that the model will help to avoid blind spots during implementation and would be a very good starting point to structure the project and create implementation checklists. Similar comments have been made with regards to the quality of the design recommendations and model elements (b). The model was perceived as a reference model and the described processes and requirements as industry best practices which allow companies individual adaptations.

Two experts also mentioned that the presented work would help to “demystify” the notion of app stores (c). Many managers would only “fixate on the Apple App Store”, and ignore that consumer app stores “cannot be simply transferred to the EAS segment”. All experts highlighted the comprehensiveness of the model and also the good level of detail (d). One expert explained that the level of detail is “fully appropriate” and that even more detail would be counterproductive and not be usable in practice because every implementation needs to consider different business and technological pre-requisites. Last, the experts also mentioned the structure of the presented material (e). The clusters were positively highlighted that divided the model between the supply- and demand-side value chain and then further into the task- and resource levels. Additionally, the use of different forms of representation was mentioned positively (i.e., graphical diagrams as well as tabular overviews).

**Reach of model**

The next section of statements tested the reach of the model, i.e., how valuable it was beyond a single company and whether it was useful for all observed roles in the software ecosystem (platform provider, software customer, and ISV).

Some feedback given in this section is redundant to the feedback provided in the first section. Hence only new aspects are listed here. The areas of improvement with regards to reach are:

a) value for ISV can mainly be assessed by the platform capabilities provided, and the number of software customers on the app store (1);
b) options on task sharing between ISV and platform provider (1);
c) alternative intermediation models not considered (1);
d) value for software customers limited (2).

The first point (a) is a statement rather than a recommendation for improvement. Accordingly, the expert stated, that the platform selection decision made by an ISV is mainly determined by two factors: first, the platform capabilities and second, the number of customers accessing the app store who in turn are the potential customers for the ISV. While the first aspect is addressed by the presented model and may help the ISV to assess existing platforms, the latter would be beyond such a model according to the expert. The second aspect mentioned referred to the delimitation of tasks assigned to the ISV and tasks assigned to the platform provider (b). The expert mentioned that there are multifold design options to split certain business processes among platform provider and the ISV. One potential enhancement of the model could be to focus on this aspect provide alternative setups. The third topic (c), focused on the revenue model, stating that there would be additional viable options which could be added and further evaluated – in particular he stated the “platform provider as reseller” model should be added and further researched in addition to the agent model. Last (d), two experts mentioned that the value of the model itself for software customers was limited since they would mainly be interested in the capabilities of the application and the ease of buying and not in how this is achieved.

The positive highlights in this sections are:

a) model can be effectively transferred across companies (3);

b) model can accelerate the design of the platform provider’s organization (2);

c) model helps ISVs to select the right platform (2);

d) research on internal app store for software customer (3);

e) app’ification model for the software customer (2).

With regards to transferability (a), the experts mentioned that the comprehensive model has aspects which might not be relevant for all companies, but at the same time this also supports the applicability of the model across the EAS domain beyond the scope of single companies. Two experts stressed the importance of the model to the platform provider (b), and mentioned that since he needs to define the largest part with regards to the app store model, it can “greatly accelerate the design work needed to build up such an organization”. The value for the ISV (c) was mainly seen in having a framework supporting him to assess and select the right platform. The software customer benefits, according to three mentions, from the research on the internal app store (d) and how it can help with IT governance processes. Moreover, the software customer will also benefit from the recommendations made as part of the proposed “app’ification criteria” (e). Explicitly, the experts mentioned, focused scope, trials, and seamless deployment.

**Overall impression**

The last section only uncovered two new aspects with regards to improvements:

a) model language and model scope (3);

b) options for competitive differentiation (1).

First, three experts mentioned that parts of the presented model require significant time to comprehend (a) – especially if one is not familiar with the model language or has no IS background (or modeling background). One expert suggested different forms of representation for different audiences. The second aspect mentioned was that the model should explicitly allow for differentiation (b). While there would be elements a company should explicitly “not reinvent” and just follow the “best practices”, there are other areas which are more suited for differentiation.

**Reflection**

Enlarged upon here are selected aspects from the area of improvements. First, there were the mentions of providing an incremental plan with classifications of what is a “must”, and what are only “options for enhancing” an implementation of the app store model. While the desire for guidance in this direction is fully understandable, the data which was available did not allow such a model structure and I believe it will also be hard to achieve the same in future studies: the prioritization of selected elements largely depends on the point of departure of the focal company – what are the business pre-requisites, what are the technological capabilities, what is the company’s business strategy? The app store model for EAS bridges strategy and actual business process models, hence a real prioritization and incremental plan can only be created against a real world background. From a researcher’s point of view, one proposal would be to soften the question to ‘what is the minimum scope’ a platform provider should implement as part of a first project phase to launch a viable app store for EAS? Such a query could potentially be answered by a future study, for example, by evaluating existing app stores with regards to their common capabilities.

Another aspect mentioned was the desire for more detailed implementation guidance and, furthermore, descriptions of how to implement certain processes. Important to mention at this point is one argument raised by another expert who highlighted that he believed the model had the right detail and that any further detail would not help in practice since it could not consider the necessary real-world context. Moreover, when translating the model to the real world the respective company is asked to add their proprietary aspects in order to differentiate against competitors. While I agree with these arguments, I also agree that single model elements allow for more detailed analysis and could be researched in future studies (e.g., consultative sales tools).

Future research could also take care of the suggestion to apply the proposed model to existing app stores for “benchmarking”. While I explicitly mentioned that the presented model could be used for assessing existing app store models on the market, to create a comprehensible benchmark the model would require some simplification and compression to be a practicable framework for benchmarking existing app stores. Moreover, I would encourage researchers investigating in this area to consider developing a multi-step maturity model instead of a simple static benchmark.

Other aspects mentioned such as alternative revenue models could be added, while options for alternative setups between ISV and platform provider are valid and offer further opportunities for future research. The model artefact at hand, as mentioned before, does not claim to be complete in the sense that it entails all possible and viable options. Included here are those aspects which were sufficiently grounded in the data which was available through either original empirical research or literature. Based on this data the model represents a solution space which can however of course be enhanced, challenged or revised by future works (with regards to interpretation of the model, see also Chapter 5.1).

# 6 Contribution and Limitations

In this chapter the multi-fold contributions to theory and practice will first be reviewed (6.1). Limitations and options for future research are then summarized in Chapter 6.2

## 6.1 Contribution to Theory and Practice

### 6.1.1 Contribution to Theory

The research at hand is an application-oriented research, strongly motivated by practice, from the trends and challenges within the IT and enterprise software industry. However, to grasp the socio-technological phenomenon of the App Store Model for EAS, multiple fields of research have been leveraged and merged in this work. Consequently, the cross-sectional and integrative nature of this research should itself be understood as a contribution, bridging the boundaries of single fields of research. For example, the combination of industrial marketing with software ecosystem and platform research has not been researched in such a scope before. This aspect has been explicitly acknowledged only recently by Giessmann who referenced two papers of this thesis in this regard (Giessmann 2015, p.25).

Overall, this work can be assigned to business model research. As mentioned earlier, “business models in the IT industry” have been highlighted as one of four dedicated streams in business model research (Veit et al. 2014). Moreover, Hess stresses the importance of capturing, designing, and analyzing technology-driven business models as opposed to finding business model definitions or classification taxonomies (Hess 2012). This thesis follows these recommendations by analyzing, capturing, and designing the App Store Model for EAS for the first time. Overall business model research in this domain is rare. Hence the exploratory approach in this thesis opens access to insights into the enterprise software industry which are often not available to the academic community in the depth and breadth presented here. Concretely, Chapter 5 with its design-oriented approach deriving the App Store Model for EAS based on the empirical findings of seven distinct and mostly exploratory research artefacts, is a very comprehensive model artefact which focuses on the value architecture. Therefore, the presented model can also be an object of further analysis to discuss the individual proposed aspects or to motivate confirmative repetitive works.

Looking more closely at the individual research clusters and their impact, Paper 1 defined the concept of PaaS as a dedicated business model in the IT industry accompanied by two practical mini-case studies at a time when the concept started to take shape in practice. As a foundational contribution, this work will serve as a reference to many subsequent works in this field.

Paper 2 investigates software platform ecosystems from the complementors', i.e., ISVs', perspective. Most studies in software ecosystem research are strategy-oriented, and thus practice-oriented studies providing insights into practical partner management tactics and determinants are rarely available and even less so in the enterprise software domain (see also Manikas & Hansen 2013; Iansiti & Levien 2004b; Selander et al. 2013; Ghazawneh & Henfridsson 2013). Consequently, this work has conceptualized a framework grounded in practice modelling the determinants of partners joining or retaining a software platform ecosystem. The framework can also be seen as an application-oriented adoption model and this way also contributes to research on IT adoption for organizations.

On the demand-side of the App Store Model for EAS, Papers 3, 4, and 5 incrementally research the buying process of EAS from a software customer's organization on the condition of using an online, e-commerce-based, buying channel: the app store. Since app stores have rarely been studied, these works contribute an adoption model for the channel choice for EAS. This is an application-oriented contribution to the IT adoption field for organizations, but equally a contribution to research on industrial marketing interested in determinants within the buying process of EAS goods using online channels.

Paper 6 then introduces the App Store model for EAS as an application system, which is investigated in relation to IT governance and consumerization of IT, as part of information management research. It uncovers the potential of the app store to support the targets of IT governance frameworks and counter the problems of IT consumerization in organizations.

Paper 7 researches the characteristics of enterprise software which enable the procurement using an online channel such as an app store. Very typically, research on enterprise software has mostly focused on the architectural and technical level, proposing concepts such as SOA or micro-services. However, research should not stop there, since further knowledge is necessary to make software a commercially ready product. And knowledge about how to make EAS ready for online channels or the app store has not been available either to theory or practice. Therefore, this paper makes a unique contribution to the aspect of commercialization of EAS products for online sales and marketing in the domain of enterprise software research. It has also been clearly shown that commercialization and software design should not be looked at fully independently but that these aspects need to be designed simultaneously in order to best meet the necessary requirements.

Beyond providing an application-oriented business model artefact, Chapter 5 also contributes to e-commerce research by introducing multi-fold e-commerce technologies and measures in the EAS domain. The proposed multi-channel approach in Paper 5 can also be assigned to the e-commerce related results.

Last, the methodological orientation of the Germany-based “Wirtschaftsinformatik” (IS) has undergone a broad discussion in the past years. Whereas international IS research scholars have focused on applying empirical research methods, German IS scholars have traditionally focused on design-oriented works. Some argue in favor of the empirical tradition, others in favor of the design-oriented one (as an example of this discussion see Wigand 2012; Frank 2012; Robra-Bissantz 2012; Sinz 2010). The methods selected and combined in this thesis also contribute to the mediation of both opinions in that they try to show that it is not a question of “either/or” but it is rather a “both/and” – a viewpoint which I believe will prevail in the long-run. The methods chosen need to suit the endeavor at hand. I have explained in great detail the reasons why a qualitative and exploratory strategy has been chosen for the empirical works to best grasp the new and complex socio-technological phenomenon. At the same time, I believe application-oriented research, as IS research clearly is, cannot stop there, but needs to contribute to translating the empirical findings into usable artefacts. Hence the empirical research has been embedded in a design-oriented frame with is completed with this introductory paper and especially Chapter 5. Therefore, I hope this work may also serve as a positive example for the discussion about the methodological orientation of the German and international IS disciplines alike.

### 6.1.2 Contribution to Practice

Industry partners provided access to a broad range of resources such as experts, systems, or documentation. Beyond providing the sources for the empiric research, they also have been treated as partners throughout the research process: in exchange for access to resources and information vice versa the research results have been presented by conducting dedicated expert reviews and workshops which were enabled by preparing the results with a language tailored to managers and practitioners. The cumulative structure of the thesis has helped to timely loop back the results in digestible increments to both, the academic community via conferences and journals, and to practitioners. Moreover, the feedback given could be incorporated and ensured the relevance of the individual works and was ultimately also an integral element of the overarching research process (cf. Chapter 5.1). In the following I highlight the elements most relevant to practitioners.

Paper 2 models the determinants of ISVs or partners towards their intention to join a software platform ecosystem. The framework is most useful to a partner management within a platform provider’s organization. They can derive concrete actions to retain partners or acquire new ones and improve the overall attractiveness of or satisfaction with the platform and related program and services. The framework, however, may also be useful to ISVs to better understand and calibrate their role within the ecosystem, and to help them decide if joining a platform is beneficial and if so which platform is best to choose.

Paper 3, investigates the EAS product portfolio and the buying process via an app store and identifies major barriers and drivers for conducting the purchase via the online channel. Paper 4, enhances and integrates the findings of Paper 3 in an adoption model, and Paper 5, adds the perspective of offline channels and integrates the adoption model into a multi-channel adoption framework. The findings of all the papers can be used to either establish and design a new online channel or app store, or to improve an existing one, by counteracting the barriers and promoting the drivers with adequate e-commerce technologies or other means. Moreover, the results can support better assigning of the different EAS products to either offline or online sales channels, or the qualitative findings can be used to make a root cause analysis in the case of issues. One of the findings of Papers 3 to 5 is, furthermore, that product or solution related determinants have an outstanding impact on channel decision choice. This aspect is further researched in Paper 7, which proposes the so-called “app’ification pattern” to make EAS products ready for app stores. Obviously, product designers, product managers, or product marketers can use these findings to better evaluate whether a product should be sold via an app store or to introduce guidelines for developing new apps for the online channel.

Paper 6 can be used by software customers who want to understand the benefit of using public or internal app stores in their company. Here the work proposes practices to make best use of the two app store variants for supporting company-internal IT governance objectives while providing more flexibility to business departments and users in acquiring and choosing business applications.

Ultimately, this introductory paper and Chapter 5 constitute a detailed business model artefact which can serve as a reference to platform providers for establishing an App Store Model for EAS or for reflecting on and calibrating an existing one. The recommendations can be used to derive concrete business process models, functional blueprints or specifications, or even job profiles. The model artefact can also be leveraged by ISVs when deciding to join a platform by providing them with a guide to best assess the capabilities of existing providers.

## 6.2 Limitations and Future Research

Quality aspects of the presented research have already been discussed in detail as part of the respective methodology descriptions. For quality aspects of the qualitative cross-sectional research, refer to Chapter 3.3.4, while for the case studies in this thesis see 3.4.5, and for the design-oriented model artefact in Chapter 5, quality aspects have been discussed in 3.5.3. Furthermore, Chapter 5.6 includes an expert review of the model artefact, highlighting areas for improvement from a practitioner’s point of view and also reflecting on these suggestions and discussing options for potential future research. Moreover, in sections 3.1 and 3.2 the overall classification of the research was introduced and the epistemological orientation (constructivist) clarified. The arguments presented in these cited sections will not be repeated at this point, but instead a general reflection will follow on the limitations of the work to motivate future research.

The models which have been conceptualized based on qualitative data are mainly of an exploratory nature. While the sampling conducted followed the guidelines for qualitative studies (e.g., saturation, heterogeneity, appropriateness), the overall number of objects studied was limited. Moreover, the studies have always researched one aspect of the value-chain at a time. Hence, the empirical findings were uncovered against the background of the state of technology during the research time frame. While the behavioral models have tried to abstract from concrete technologies as much as possible, close interdependencies cannot be denied. Similarly, the developed artefact proposes measures against the background of what is possible at the time of writing, or, with goodwill, in the foreseeable future, a timeframe without further definition. Consequently, in a fast-paced environment such as the IT industry, where boundary conditions may change, it is not known how stable the findings will be over time. Furthermore, the qualitative research objects have mainly been chosen in the greater environment of SAP. Though I would argue SAP can serve as a highly representative context within the EAS industry, it is a limitation which should be addressed by future research. Therefore, I encourage repetitive studies throughout the value chain of the app store model to either confirm, enhance, or reject the findings of this thesis.

In general, my decision to favor a qualitative, exploratory research strategy might be a source of criticism. Some scholars even argue that exploratory research cannot be regarded as completed research as it only represents “preliminary” findings laying out the structures which should to be confirmed by consecutive studies which follow a more confirmative nature (Bryman & Bell 2011, p.62). While I do not want to discourage quantitative strategies for confirming or rejecting certain findings[1], a few aspects should be considered: The contemporary and complex socio-technological phenomenon studied in this thesis in large part does not or only in a limited way allow a proper quantitative design to comprehensively cover all constructs. First, this is because it simply would not be economically feasible to invest the effort in modeling all relevant objects, collecting statistically sufficient data and deriving the significant and relevant results, and second because the population of the unit of analysis is often not large enough to do the same. While, last, the quantitative findings have a tendency to favor the statistical mean and thus lose the important information on the edges. Following this argument, one can conclude that case studies and qualitative studies in general are more

[1] I myself have investigated the presented app’ification patterns using a quantitative approach in a working paper, see (Wenzel 2014a). Ultimately I decided not to proceed on this path since the knowledge which could possibly be gained using the quantitative methods was very limited given the constraints in terms of time and resources. In particular, creating a sample large enough to generate statistically significant results on the interesting aspects was a major challenge.

suited to coping with and recognizing the complexity of a phenomenon than quantitative studies (for a comprehensive reflection of this topic see Gummesson 2007).

Another point of criticism very often repeated in relation to qualitative studies in general and even more so with case studies is the issue of generalizability (external validity) of findings (Bryman & Bell 2011, p.61; Gummesson 2007). As elaborated in the methodology section (cf. 3.4), case study designs are chosen when the unit of analysis is closely embedded in its context and boundaries of the context, and the unit of analysis or the phenomenon cannot be clearly delimited (Yin 2013). Consequently, transferring a finding from one object of investigation to another would also mean that not only would the parameters of the object need to be controlled in order to predict the outcome but also the parameters of the object's context. In practice, when facing highly complex socio-technological contexts, this is mostly not possible. However, the case study designs are also not chosen with the intention of deriving models that are capable of predicting outcomes with a statistical probability. They are instead chosen to identify possible structures or measures that can be applied to a certain problem. These measures, however, should never be chosen without reflection. To illustrate my argument, I refer to the example of the case study conducted in Paper 7 (cf. Chapter 3.4.4 & V.1). The intention was to identify how measures of readiness of a business application for sales and distribution via an app store can be increased. It is obvious that applications can differ in many aspects (price, scope, purpose, technology, etc.). Hence, the measures identified for one application might not be transferable to another application. Moreover, one strategy could have been to only include those measures that have proven to work for at least *n* number of applications. Despite the fact that this would still not create statistical significance we would also lose the particular measures which appeared only once. Consequently, I argue that generalizability should not only be rated in case studies by the replications applied (theoretical or literal) but also by how well a phenomenon's context, i.e., pre-requisites and related constructs, are explicitly described and thus ultimately allow, for example, a practitioner to properly reflect on a potential measure as to whether it might also suit his specific new context. I therefore also recommend that future repetitive attempts investigate the App Store Model for EAS using qualitative research strategies.

Further criticism may be addressed at the design-oriented model artefact developed in this introductory paper. One aspect may be the business model framework that I have chosen. The framework by Stähler (Stähler 2010) consists of three major business model elements. While this also corresponds to other business model frameworks, for example the one by Timmers (Timmers 1998), other business model frameworks delimit more elements, such as the framework by Osterwalder and Pigneur (Osterwalder & Pigneur 2005). As highlighted by both Veit et al. and Hess, the unique aspect of the business model framework is the value architecture (Veit et al. 2014; Hess 2012) which is also the focus of the developed artefact in this thesis. Hence, I chose the framework by Stähler since he also recognizes the value architecture as the central element of the business model framework.

Furthermore, researchers may ask why I have chosen to follow the very generic and abstract guidance offered by Becker (Becker 2010) as opposed to more precise frameworks on design-oriented research, such as (Peffers et al. 2007). The artefact I have developed integrates a broad range of research artefacts deriving a business model framework and including the description of the value chain of an entire industry sector. The model artefact as a whole is hence not a “prototype-able” software artefact, but rather an artefact which corresponds to the idea of a “constructed possible world” as mentioned by Frank (Frank 2009). Consequently, the framework that was needed should provide sufficient degrees of freedom to act as a frame for the thesis as a whole. The approach proposed by Becker seemed most appropriate for me to reach my objectives.

Despite that parts of my recommendations have already found their way into practice by the means I have elaborated in the section on contributions to practice (6.1.2), I encourage future design-oriented research to further detail selected elements of my model artefact and conduct dedicated studies involving the prototyping of certain elements or by testing my recommendations in practice together with partners from practice. For example, developing and evaluating consultative sales tools such as online wizards to define and select the scope of complex solutions using business language is a challenging endeavor which may serve as its own dedicated research artefact which needs to be closely designed together with actual users.

Moreover, while approaches to more granular applications using open, lightweight interface technologies (e.g., REST-based APIs) are nowadays widely practiced, the flexible integration of single apps into a greater landscape of applications automating entire business tasks remains a practical challenge. However, this would be a significant enabler to a wider adoption of the app store model for EAS, to flexibly buy and use business applications. While early concepts exist, to design such highly flexible systems (Sinz et al. 2011), e.g., via so-called smart process apps (Nadj et al. 2016) which intelligently recognize their context and environment and can adapt their structure and behavior accordingly, practical concepts to make this a reality are still absent or rare and further research is needed.

# 7 Conclusion

The objective of this thesis was to comprehensively capture, structure, and analyze the phenomenon of app stores for EAS. Moreover, based on the gathered knowledge the objective was to create a design proposal which can help practitioners to design, analyze, or adjust app stores for EAS. The topic touches on many different aspects of the EAS value chain and in that can be considered an integrating subject. Therefore, the business model construct has been selected as a frame with the focus on the value architecture. The analysis and the design of such new and IT-driven business models and the associated information systems is at the core of IS research.

The results first capture the requirements of the different stakeholders towards app stores for EAS, including platform providers, ISVs, and software customers. In particular, they uniquely present the specifics of the EAS industry towards app stores from first hand empirical qualitative data. These results have then been used to design a model artefact combining the insights into a consistent framework proposing process and resource requirements to design an end-to-end app store for EAS. The results provide manifold opportunities for academic exploitation and practitioners can benefit from detailed and practicable recommendations. The combination of empirical qualitative research methods and design-oriented methods has been the approach of choice, and the combination of the two has proved to be highly compatible and complementary.

When I first engaged with the topic from a research perspective in 2010 I believed the market would very soon and broadly adopt the concepts of electronic marketplaces in the EAS segment and the entire value chain would be highly automated through integrated information systems. Now, at the time of writing in 2017, I conclude that in this area progress has been made but not to the degree I had expected. Single aspects have greatly evolved, for example, the maturity and availability of PaaS offerings for EAS, and also marketplaces and App Stores for EAS today have a much better user experience. A seamless integration of the individual systems of the EAS value chain from supply-side platforms to demand-side app stores for EAS, however, has still not been achieved. Partially, I could also observe major setbacks where relaunched app stores for EAS dropped capabilities for payment support or even direct ordering. Therefore, I am confident, that the integrating nature of the present work will also be of great value to practitioners in the future.

## References

AIS, 2015. About This Journal: Business & Information Systems Engineering. Available at: http://aisel.aisnet.org/bise/about.html [Accessed February 14, 2015].

AIS, 2013a. MIS Journal Rankings. Available at: http://aisnet.org/?JournalRankings [Accessed November 20, 2015].

AIS, 2013b. Senior Scholars' Basket of Journals. Available at: http://aisnet.org/?SeniorScholarBasket [Accessed November 29, 2013].

Ajzen, I., 1991. The Theory of Planned Behavior. *Organizational Behavior And Human Decision Processes*, 50(2), pp.179–211.

Al-Debei, M., El-Haddadeh, R. & Avison, D., 2008. Defining the business model in the new world of digital business. In *Proceedings of the Fourteenth Americas Conference on Information Systems (AMCIS)*.

Al-Debei, M.M. & Avison, D., 2010. Developing a unified framework of the business model concept. *European Journal of Information Systems*, 19(3), pp.359–376.

Amit, R. & Zott, C., 2001. Value Creation in E-Business. *Strategic Management Journal*, 22, pp.439–520.

Anding, M., 2010. SaaS: A Love-Hate Relationship for Enterprise Software Vendors. In A. Benlian, T. Hess, & P. Buxmann, eds. *Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen*. Wiesbaden: Gabler, pp. 43–56.

Angeren, J. van, Kabbedijk, J. & Popp, K., 2013. Managing software ecosystems through partnering. In M. A. Jansen, Slinger Cusumano & S. Brinkkemper, eds. *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Edward Elgar Publishing, p. 344.

Ariba, 2015. Solutions for Buyers and Suppliers. Available at: www.ariba.com [Accessed January 22, 2015].

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I. & Zaharia, M., 2009. *Above the Clouds: A Berkeley View of Cloud Computing*, Electrical Engineering and Computer Sciences University of California at Berkeley.

Backhaus, K. & Voeth, M., 2011. *Industriegütermarketing*,

Baker, J., 2012. The Technology – Organization – Environment Framework. In Y. K. Dwivedi, M. R. Wade, & S. L. Schneberger, eds. *Information Systems Theory: Explaining and Predicting Our Digital Society*. Springer New York, pp. 231–245.

Bakos, J.Y., 1997. Reducing Buyer Search Costs: Implications for Electronic Marketplaces. *Management Science*, 43(12), pp.1676–1692.

Bandulet, F., Faisst, W., Eggs, H., Otyepka, S. & Wenzel, S., 2010. Software-as-a-Service as Disruptive Innovation in the Enterprise Application Market. In A. Benlian, T. Hess, & P. Buxmann, eds. *Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen*. Wiesbaden: Gabler, pp. 15–29.

Banker, R.D. & Kauffman, R.J., 2004. 50th Anniversary Article: The Evolution of Research on Information Systems: A Fiftieth-Year Survey of the Literature in Management Science. *Management Science*, 50(3), pp.281–298.

Barbosa, O. & Alves, C., 2011. A Systematic Mapping Study on Software Ecosystems.

*Proceedings of the Workshop on Software Ecosystems 2011*, pp.15–26.

Bartels, A. & Moore, C., 2013. Smart Process Applications. *The Forrester Wave, Q2 2013*.

Becker, A., Widjaja, T. & Buxmann, P., 2011. Value Potentials and Challenges of Service-Oriented Architectures. *Business & Information Systems Engineering*, 3(4), pp.199–210.

Becker, J., 2010. Prozess der gestaltungsorientierten Wirtschaftsinformatik. In H. Österle, R. Winter, & W. Brenner, eds. *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. pp. 13–17.

Becker, J. & Pfeiffer, D., 2006. Beziehungen zwischen behavioristischer und konstruktionsorientierter Forschung in der Wirtschaftsinformatik. In S. Zelewski & N. Akca, eds. *Fortschritt in den Wirtschaftswissenschaften: Wissenschaftstheoretische Grundlagen und exemplarische Anwendungen*. Wiesbaden: DUV, pp. 1–17.

Beimborn, D. & Palitza, M., 2013. Enterprise App Stores for Mobile Applications - Development of a Benefits Framework. In *AMCIS 2013 Proceedings*.

Benlian, A., Hess, T. & Buxmann, P., 2010. Chancen und Risiken des Einsatzes von SaaS - die Sicht der Anwender. In A. Benlian, T. Hess, & P. Buxmann, eds. *Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen*. Wiesbaden: Gabler, pp. 173–187.

Berbner, R. & Bechtold, J., 2010. Innovationsmanagement als elementarer Bestandteil des IT-Managements. In *Innovatives IT-Management*.

Böckle, R., 2013. B2B App Stores - Anbieter im Vergleich. *Computerwoche*, 49(49), pp.14–21.

Bosch, J., 2012. Software ecosystems: Taking software development beyond the boundaries of the organization. *Journal of Systems and Software*, 85(7), pp.1453–1454.

Boudreau, K., 2010. Open Platform Strategies and Innovation: Granting Access vs. Devolving Control. *Management Science*, 56(10), pp.1849–1872.

Boudreau, K.J., 2008. Too Many Complementors? *SSRN Electronic Journal*.

Bryman, A. & Bell, E., 2011. *Business Research Methods*, Oxford University Press, USA.

Burkard, C., Draisbach, T., Widjaja, T. & Buxmann, P., 2011. Software ecosystems: Vendor-sided characteristics of online marketplaces. In *Informatik 2011, 41. Jahrestagung der Gesellschaft für Informatik, Lecture Notes in Informatics, Volume P192*. pp. 192–205.

Burkard, C., Widjaja, T. & Buxmann, P., 2012. Software Ecosystems. *Business & Information Systems Engineering*, 4(1), pp.41–44.

Burkhart, T., Krumeich, J., Werth, D. & Loos, P., 2011. Analyzing the Business Model Concept-A Comprehensive Classification of Literature. In *ICIS*. pp. 1–19.

Buxmann, P., Diefenbach, H. & Hess, T., 2011. *Die Softwareindustrie*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Buxmann, P., Hess, T. & Lehmann, S., 2008. Software as a Service. *WIRTSCHAFTSINFORMATIK*, 50(6), pp.500–503.

Carr, N., 2003. IT doesn’t matter. *Harvard business review*.

Ceccagnoli, M., Forman, C., Huang, P. & Wu, D.J., 2012. Cocreation of Value in a Platform Ecosystem: The Case of Enterprise Software. *MIS Quarterly: Management Information Systems*, 36(1), pp.263–290.

Cespedes, F. V. & Corey, E.R., 1990. Managing Multiple Channels. *Business Horizons*, 33(4), pp.67–77.

Chesbrough, H. & Rosenbloom, R.S., 2002. The role of the business model in capturing value from innovation : evidence from Xerox Corporation ’ s technology spin-off companies. *Industrial and Corporate Change*, 11(3), pp.529–555.

Collis, J. & Hussey, R., 2009. *Business research: A practical guide for undergraduate and postgraduate students* 3rd ed., Palgrave Macmillan.

Corbin, J.M. & Strauss, A., 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13, pp.3–21. Available at: http://web.b.ebscohost.com/ehost/pdfviewer/pdfviewer?sid=591044ca-66e7-4cb4-9873-ff4fb5ce2ba2%40sessionmgr113&vid=0&hid=102 [Accessed February 12, 2015].

Creswell, J.W., 2003. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* 2nd ed., Thousand Oaks, CA: Sage.

Crook, S.K., 2013. *The Role of the Corporate App Store in the Mobile Enterprise*, IDC.

Darke, P., Shanks, G. & Broadbent, M., 1998. Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal*, 8(4), pp.273–289.

Davis, F.D., 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), pp.319–340.

Davis, F.D., Bagozzi, R.P. & Warshaw, P.R., 1992. Extrinsic and Intrinsic Motivation to Use Computers in the Workplace. *Journal of Applied Social Psychology*, 22(14), pp.1111–1132.

Dedrick, J. & West, J., 2004. An Exploratory Study into Open Source Platform Adoption. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS’04)*, 8(C), p.80265b.

DePietro, R., Wiarda, E. & Fleischer, M., 1990. The context for change: Organization, technology and environment. In L. G. Tornatzky & M. Fleischer, eds. *The processes of technological innovation*. Lexington (Massachusetts): Lexington Books, pp. 151–175.

Domschke, W. & Scholl, A., 2000. *Grundlagen der Betriebswirtschaftslehre: eine Einführung aus entscheidungsorientierter Sicht*, Springer.

Drake, S.D., 2012. *Mobile Enterprise App Stores : Open for Business*, IDC.

Dubé, L. & Paré, G., 2003. Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations. *MIS Quarterly*, 27, pp.597–635.

Dubey, A., Mohiuddin, J. & Baijal, A., 2008. Emerging platform wars in enterprise software.

Eisenhardt, K.M., 1989. Building Theories from Case Study Research. *Academy of Management Review*, 14(4), pp.532–550.

Eisenmann, T., Parker, G. & Van Alstyne, M., 2007. Platform Networks – Core Concepts. Executive Summary.

Eisenmann, T., Parker, G. & Alstyne, M.W.V.M. Van, 2006. Strategies for two-sided markets. *Harvard business review*, 84(10), p.12.

Eisenmann, T.R., Parker, G. & Van Alstyne, M., 2009. Opening Platforms: How, When and Why? In *Platforms, Markets and Innovation*. pp. 131–162.

Evans, D.S., Hagiu, A. & Schmalensee, R., 2006. *How Software Platforms Drive Innovation and Transform Industries*, London: MIT Press.

Faisst, W., 2011. Die nächste Generation der Unternehmens-Software am Beispiel von SAP Business ByDesign. *Wirtschaftsinformatik & Management*, 3(4), pp.24–31.

Ferstl, O.K., 2012. E-Commerce Systeme. *Lectore Notes University Bamberg*.

Ferstl, O.K., 1979. *Konstruktion und Analyse von Simulationsmodellen*, Hain.

Ferstl, O.K. & Sinz, E.J., 2013. *Grundlagen der Wirtschaftsinformatik* 7th Editio., Oldenbourg Verlag.

Ferstl, O.K. & Sinz, E.J., 1997. Modeling of Business Systems Using the Semantic Object Model (SOM) - A Methodological Framework -. In P. Bernus, K. Mertins, & G. Schmidt, eds. *Handbook on Architectures of Information Systems. International Handbooks on Information Systems.* Berlin: Springer.

Finley, K., 2012. Trends 2012: The State of Enterprise App Stores. *Silicon Angle*. Available at: http://siliconangle.com/blog/2012/02/02/trends-2012-the-state-of-enterprise-app-stores/ [Accessed December 30, 2014].

Fishbein, M. & Ajzen, I., 1975. *Belief, attitude, intention and behavior: an introduction to theory and research*, Addison-Wesley.

Föckeler, C., 2010. Neue Anbieterstrategien: Wie Salesforce.com den Software-Markt umkrempelt. In A. Benlian, T. Hess, & P. Buxmann, eds. *Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen*. Wiesbaden: Gabler, pp. 109–124.

Foscht, T. & Swoboda, B., 2007. *Käuferverhalten: Grundlagen - Perspektiven - Anwendungen* 3rd ed., Gabler Verlag.

Frank, U., 2009. Die Konstruktion möglicher Welten als Chance und Herausforderung der Wirtschaftsinformatik. In J. Becker, H. Krcmar, & B. Niehaves, eds. *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik.* Heidelberg: Physica-Verlag, pp. 167–180.

Frank, U., 2012. Konstruktionsorientierter Forschungsansatz. *Enzyklopaedie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/uebergreifendes/Forschung-in-WI/Konstruktionsorientierter-Forschungsansatz [Accessed January 30, 2016].

Free Software Foundation, 2016. GNU General Public License. Available at: https://www.gnu.org/licenses/ [Accessed July 30, 2016].

French, A., Guo, C. & Shim, J.P., 2014. Current Status, Issues, and Future of Bring Your Own Device (BYOD). *Communications of the Association for Information Systems*, 35(1).

Gartner, 2013. Gartner Says Mobile App Stores Will See Annual Downloads Reach 102 Billion in 2013. Available at: http://www.gartner.com/newsroom/id/2592315 [Accessed December 22, 2014].

Gawer, A. & Cusumano, M., 2013. Industry Platform and Ecosystem Innovation. *Journal of Product Innovation and Management*, 31(3), pp.417–433.

Gawer, A. & Cusumano, M.A., 2008. How Companies Become Platform Leaders. *MIT Sloan Management Review*, 49(2), pp.28–35.

George, G. & Bock, A., 2011. The business model in practice and its implications for entrepreneurship research. *Entrepreneurship theory and practice*, 35(1), pp.83–111.

Gersch, M., 2015. Electronic Business. *Enzyklopädie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/informationssysteme/crm-scm-und-electronic-business/Electronic-Business [Accessed March 20, 2016].

Ghazawneh, A. & Henfridsson, O., 2013. Balancing platform control and external contribution in third-party development: The boundary resources model. *Information Systems Journal*, 23(2), pp.173–192.

Gibbert, M. & Ruigrok, W., 2010. The "'What'" and "'How'" of Case Study Rigor Three Strategies Based on Published Work. *Organizational Research Methods*, pp.1–28.

Gibbs, G.R., 2007. *Analysing Qualitative Data*, Los Angeles: SAGE Publications.

Giessmann, A., 2015. *Designing business models for cloud platforms*, Lausanne: Université de Lausanne.

Giessmann, A., Feurer, S., Schneider, T. & Wenzel, S., 2015. Consumers' Preferences on Platform as a Service and SAP`s Engineering Response. *SAP Research*. Available at: http://hdl.handle.net/10419/106129 [Accessed January 12, 2015].

Giessmann, A., Feurer, S., Schneider, T. & Wenzel, S., 2012. Consumers' Preferences on Platform as a Service and SAP`s Engineering Response. *SAP Research - Whitepaper*. Available at: http://scn.sap.com/docs/DOC-42360 [Accessed January 3, 2015].

Giessmann, A. & Legner, C., 2013. Designing Business Models for Platform as a Service: Towards a Design Theory. In *ICIS*. pp. 1–10.

Gillett, F.E., Staten, J., Brown, E.G. & Lee, C., 2008. *Future View: The New Tech Ecosystems Of Cloud, Cloud Services, And Cloud Computing*, Forrester Research.

Glaser, B.G. & Strauss, A.L., 1967. *The Discovery of Grounded Strategies for Qualitative Research* 7th ed.,

Glaser, B.G. & Strauss, A.L., 2012. *The Discovery of Grounded Theory: Strategies for Qualitative Research* 7th ed., Transaction Publishers.

Grohmann, W., 2009. *Von der Software zum Service: ASP-Software on Demand- Software-as-a-Service- Cloud Computing ; Neue Formen der Software-Nutzung*, H. K. P. Consulting.

Gross, N.C., Giacquinta, J.B. & Bernstein, M., 1971. *Implementing organizational innovations: a sociological analysis of planned educational change*, Harper and Row.

Guba, E.G., 1981. Criteria for assessing the trustworthiness of naturalistic inquiries. *ECTJ*, 29(2), pp.75–91.

Guba, E.G. & Lincoln, Y.S., 1994. Competing Paradigms in Qualitative Research. In N. K. Denzin & Y. S. Lincoln, eds. *Handbook of qualitative research*. Thousand Oaks, CA: Sage, pp. 105–117.

Gummesson, E., 2007. Case study research and network theory: birds of a feather. *Qualitative Research in Organizations and Management: An International Journal*, 2(3), pp.226–248.

De Haes, S., Van Grembergen, W. & Debreceny, R.S., 2013. COBIT 5 and Enterprise Governance of Information Technology: Building Blocks and Research Opportunities. *Journal of Information Systems*, 27(1), pp.307–324.

Halingten, A. & Verville, J.C., 2002. A qualitative study of the influencing factors on the decision process for acquiring ERP software. *Qualitative Market Research: An International Journal*, 5(3), pp.188–198.

Hannah, D.R. & Lautsch, B. a., 2011. Counting in Qualitative Research: Why to Conduct it, When to Avoid it, and When to Closet it. *Journal of Management Inquiry*, 20(1), pp.14–22.

Harris, J., Ives, B. & Junglas, I., 2012. IT consumerization: when gadgets turn into enterprise IT tools. *MIS Quarterly Executive*, 11(3), pp.99–112.

Heinemann, G., 2010. *Der neue Online-Handel: Erfolgsfaktoren und Best Practices* 3rd ed., Wiesbaden: Gabler.

Heitkoetter, H., Hildebrand, K. & Usener, C., 2012. Mobile Platforms as Two-sided Markets. In *AMCIS 2012 Proceedings*.

Hess, B., Sutanto, J., Ameling, M. & Reischach, F. von, 2012. A Business-to-Business Perspective on Mobile Application Stores. In *2012 International Conference on Mobile Business*.

Hess, T., 2012. *Geschäftsmodelle als Thema der Wirtschaftsinformatik - Arbeitsbericht 1/2012* T. Hess, ed., München: Ludwig-Maximilians-Universität München, Fakultät für Betriebswirtschaft, Institut für Wirtschaftsinformatik und Neue Medien.

Hess, T., 2015. Plattformstrategie. *Enzyklopädie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Integration-und-Migration-von-IT-Systemen/plattformstrategie/index.html?searchterm=plat [Accessed February 3, 2016].

Hevner, A., March, S., Park, J. & Ram, S., 2004. Design science in information systems research. *MIS quarterly*, 28(1), pp.75–105.

Hevner, A.R., March, S.T., Park, J. & Ram, S., 2004. Design Science in Information Systems Research A. R. Hevner & S. Chatterjee, eds. *MIS Quarterly*, 28(1), pp.75–105.

Holzer, A. & Ondrus, J., 2011. Mobile application market: A developer's perspective. *Telematics and Informatics*, 28(1), pp.22–31.

Homburg, C., Schäfer, H. & Schneider, J., 2012. *Sales Excellence*, Berlin, Heidelberg: Springer.

Huang, J.-H., Yang, C., Jin, B.-H. & Chiu, H., 2004. Measuring satisfaction with business-to-employee systems. *Computers in Human Behavior*, 20, pp.17–35.

Huysmans, P. & De Bruyn, P., 2013. A Mixed Methods Approach To Combining Behavioral and Design Research Methods in Information Systems Research. In *European Conference of Information Systems*. p. 12.

Iansiti, M. & Levien, R., 2004a. Strategy as Ecology. *Harvard Business Review*, 82(3).

Iansiti, M. & Levien, R., 2004b. *The Keystone Advantage: What the new Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*, Boston, MA: Harvard Business School Press.

Ingalsbe, J., Shoemaker, D. & Mead, N., 2011. Threat Modeling the Cloud Computing, Mobile Device Toting, Consumerized Enterprise – an overview of considerations. In *AMCIS 2011 Proceedings*.

IS-Report, 2016a. Microsofts Business-Solutions-Chef forciert die Cloud. *IS Report*. Available at: http://www.isreport.de/allgemein/exklusiv-interview-microsofts-business-solutions-chef-forciert-die-cloud/ [Accessed July 11, 2016].

IS-Report, 2016b. S/4 Hana löst SAP Business Suite ab. *IS Report*. Available at: http://www.isreport.de/news/sap-revolutioniert-kernprodukt-business-suite/ [Accessed July 11, 2016].

ISACA, 2012. COBIT 5: A Business Framework for the Governance and Management of Enterprise IT.

Iyer, B., Lee, C.-H. & Venkatraman, N., 2007. Monitoring Platform Emergence: Guidelines from Software Networks. *Communications of the Association for Information Systems*, 19, pp.1–13.

Jansen, S. & Bloemendal, E., 2013. Defining App Stores: The Role of Curated Marketplaces in Software Ecosystems. In *Lecture Notes in Business Information Processing Volume 150*. pp. 195–206.

Jansen, S. & Cusumano, M.M.A., 2012. Defining software ecosystems: a survey of software platforms and business network governance. In *Proceedings of the International Workshop on Software Ecosystems 2012.* pp. 13–28.

Jansen, S., Finkelstein, A. & Brinkkemper, S., 2009. A sense of community: A research agenda for software ecosystems. In *2009 31st International Conference on Software Engineering - Companion Volume*. pp. 187–190.

Järvi, P. & Munnukka, J., 2009. The dynamics and characteristics of buying centre networks: A qualitative study on Finnish firms. *Marketing Intelligence & Planning*, 27, pp.439–457.

Jennings, C., 2008. The end of the IT department – is it in the cloud? *Computerweekly.com*. Available at: http://www.computerweekly.com/feature/The-end-of-the-IT-department-is-it-in-the-cloud [Accessed January 1, 2013].

Jin, L., Deng, X. & Lea, B.-R., 2014. An Explorative Study of Mobile Apps in the Enterprise. In *Proceedings of the Twentieth Americas Conference on Information Systems (AMCIS 2014)*. pp. 1–13.

Joachim, N., 2011. A literature review of research on service-oriented architectures (SOA): characteristics, adoption determinants, governance mechanisms, and business impact. In *AMCIS 2011 Proceedings*.

Johnston, W.J. & Bonoma, T. V, 1981. The Buying Center: Structure and Interaction Patterns. *Journal of Marketing*, 45, pp.143–156. Available at: http://web.ebscohost.com/ehost/detail?hid=17&sid=140e3e67-920f-45b7-8ba1-a4ddc44b2dad@sessionmgr11&vid=9&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ==#db=bth&AN=4997578 [Accessed January 22, 2015].

Johnston, W.J. & Lewin, J.E., 1996. Organizational buying behavior: Toward an integrative framework. *Journal of Business Research*, 35(1), pp.1–15.

Jones, D., Behrens, S., Jamieson, K. & Tansley, E., 2004. The Rise and Fall of a Shadow System: Lessons for Enterprise System Implementation. In *ACIS 2004 Proceedings*.

Karjalainen, K., Kemppainen, K. & Van Raaij, E.M., 2009. Non-compliant work behaviour in purchasing: An exploration of reasons behind maverick buying. *Journal of Business Ethics*, 85(2), pp.245–261.

Kemper, H.-G., Baars, H. & Mehanna, W., 2010. *Business Intelligence – Grundlagen und praktische Anwendungen*, Wiesbaden: Vieweg+Teubner.

Kim, H., Lee, J.-N. & Han, J., 2010. The Role of IT in Business Ecosystems. *Communications of the ACM*, 53(5), pp.151–156.

Kittlaus, H.B. & Clough, P.N., 2009. *Software product management and pricing: Key success factors for software organizations*, Berlin, Heidelberg: Springer.

Kleinaltenkamp, M. & Saab, S., 2009. *Technischer Vertrieb: Eine praxisorientierte Einführung in das Business-to-Business-Marketing*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Kontogiannis, K., Lewis, G. a. & Smith, D.B., 2008. A research agenda for service-oriented architecture. In *Proceedings of the 2nd International Workshop on Systems Development in SOA Environments - SDSOA '08*.

Kosner, A.W., 2012. The Appification Of Everything Will Transform The World's 360 Million Web Sites. *Forbes*. Available at: http://www.forbes.com/sites/anthonykosner/2012/12/16/forecast-2013-the-appification-of-everything-will-turn-the-web-into-an-app-o-verse/ [Accessed October 7, 2013].

Kotler, P., 2003. *Marketing Management* 11th ed., Prentice Hall.

Krafzig, D., Banke, K. & Slama, D., 2005. *Enterprise SOA: Service-oriented Architecture Best Practices*,

Krcmar, H., 2015. *Einführung in das Informationsmanagement*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Krcmar, H., 2010. *Informationsmanagement* 5th ed., Springer.

Kude, T., Dibbern, J. & Heinzl, A., 2012. Why Do Complementors Participate? An Analysis of Partnership Networks in the Enterprise Software Industry. *IEEE Transactions on Engineering Management*, 59(2), pp.250–265.

Laudon, K.C., Laudon, J.P. & Schoder, D., 2010. *Wirtschaftsinformatik: Eine Einführung*, Pearson Deutschland GmbH.

Lee, J., Crossler, R. & Warkentin, M., 2013. Implications of Monitoring Mechanisms on Bring Your Own Device (BYOD) Adoption. *ICIS 2013 Proceedings*.

Lehmann, S. & Buxmann, P., 2009. Preisstrategien von Softwareanbietern. *Business and Information Systems Engineering*, 51(6), pp.519–529.

Leimeister, S., Böhm, M., Riedl, C. & Krcmar, H., 2010. *The Business Perspective of Cloud Computing: Actors, Roles and Value Networks*,

Lewin, J.E. & Donthu, N., 2005. The influence of purchase situation on buying center structure and involvement: a select meta-analysis of organizational buying behavior research. *Journal of Business Research*, 58(10), pp.1381–1390.

Liao, X., Li, Y. & Lu, B., 2007. A model for selecting an ERP system based on linguistic information processing. *Information Systems*, 32(7), pp.1005–1017.

Loebbecke, C., Weiss, T., Powell, P. & Krooss, C., 2010. Drivers of B2B software purchase decisions. *Frontiers in Artificial Intelligence and Applications*, 212, pp.535–546.

Lohmann, M., Schmitzer, B. & Mertens, P., 2001. Kern-Schalen-Modell mit Fokus auf E-Commerce. In *3. Workshop "Komponentenorientierte betriebliche Anwendungssysteme."*

Luborsky, M.R. & Rubinstein, R.L., 1995. Sampling in Qualitative Research: Rationale, Issues, and Methods. *Research on aging*, 17(1), pp.89–113.

Luftman, J. & Derksen, B., 2012. Key issues for IT executives 2012: Doing More with Less. *MIS Quarterly Executive*, 11(4), pp.207–218.

Luoma, E., Rönkkö, M. & Tyrväinen, P., 2012. Current Software-as-a-Service Business Models: Evidence from Finland. In *ICSOB 2012 - Lecture Notes in Business Information Processing*. pp. 181–194.

Lyytinen, K. & Damsgaard, J., 2001. What's Wrong with the Diffusion of Innovation Theory? In M. A. Ardis & B. L. Marcolin, eds. *Diffusing Software Product and Process Innovations*. Boston, MA: Springer US, pp. 173–190.

Malone, T.W., Yates, J. & Benjamin, R.I., 1987. Electronic Markets and Electronic Hierarchies. *Communications of the ACM*, 30(6), pp.484–497.

Manikas, K. & Hansen, K.M., 2013. Software ecosystems – A systematic literature review.

*Journal of Systems and Software*, 86(5), pp.1294–1306.

Marshall, C. & Rossman, G.B., 2006. *Designing Qualitative Research* Fourth., Thousand Oaks: Sage Publications.

Matros, R., Rietze, C. & Eymann, T., 2010. SaaS und Unternehmenserfolg: Erfolgskategorien für die Praxis. In A. Benlian, T. Hess, & P. Buxmann, eds. *Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen*. Wiesbaden: Gabler, pp. 239–254.

Maxwell, J. a., 2010. Using Numbers in Qualitative Research. *Qualitative Inquiry*, 16(6), pp.475–482.

Meyer, M., Zarnekow, R. & Kolbe, L.M., 2003. IT-Governance - Begriff, Status quo und Bedeutung. *Business & Information Systems Engineering*, 45(4), pp.445–448.

Moore, J.F., 1996. *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*, HarperBusiness.

Müller, G., Syring, A., Eggs, H. & Wenzel, S., 2011. Risikomanagement und Vertrauensbildung im Cloud Computing. *Wirtschaftsinformatik & Management*, 3(5), pp.66–73.

Nadj, M., Haeußler, F., Wenzel, S. & Maedche, A., 2016. The Smart Mobile Application Framework (SMAF) – Exploratory Evaluation in the Smart City Context. In *Multikonferenz Wirtschaftsinformatik (MKWI) 2016*. pp. 1367–1378.

Nelson, P., 1970. Information and Consumer Behavior. *Journal of Political Economy*, 78(2), pp.311–329.

Niehaves, B. & Becker, J., 2006. Epistemological Perspectives on Design Science in IS Research. In *AMCIS 2006 Proceedings*. pp. 3579–3589.

Niehaves, B., Köffer, S. & Ortbach, K., 2013. The Effect of Private IT Use on Work Performance-Towards an IT Consumerization Theory. *Wirtschaftinformatik Proceedings 2013*, pp.39–53.

Novelli, F., 2015. Detection and Measurement of Sales Cannibalization in Information Technology Markets. , pp.1–185.

Novelli, F. & Wenzel, S., 2013. Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity? In *ICIS 2013 Proceedings*.

Oliveira, T. & Martins, M., 2011. Literature Review of Information Technology Adoption Models at Firm Level. *The Electronic Journal Information Systems Evaluation*, 14(1), pp.110–121.

Ortbach, K., Brockmann, T. & Stieglitz, S., 2014. Drivers for the Adoption of Mobile Device Management in Organizations. *ECIS 2014 Proceedings*.

Österle, H., Winter, R. & Brenner, W., 2010. *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*,

Osterwalder, A. & Pigneur, Y., 2005. Clarifying business models: Origins, present, and future of the concept. *Communications of the Association for Information Systems*, 16, pp.1–25.

Palanisamy, R., Verville, J., Bernadas, C. & Taskin, N., 2010. An empirical study on the influences on the acquisition of enterprise software decisions: A practitioner's perspective. *Journal of Enterprise Information Management*, 23(5), pp.610–639.

Peffers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management*

*Information Systems*, 24(3), pp.45–78.

Pettey, C., 2011. Gartner Identifies the Top 10 Strategic Technologies for 2012. *Gartner*. Available at: http://www.gartner.com/newsroom/id/1826214 [Accessed November 19, 2013].

Pettey, C., 2012. Gartner Identifies the Top 10 Strategic Technology Trends for 2013. *Gartner*. Available at: http://www.gartner.com/newsroom/id/2209615 [Accessed November 20, 2013].

Ponterotto, J.G., 2005. Qualitative research in counseling psychology: A primer on research paradigms and philosophy of science. *Journal of Counseling Psychology*, 52(2), pp.126–136.

Popp, K.M. & Meyer, R., 2010. *Profit from software ecosystems: business models, ecosystems and partnerships in the software industry*, Books on Demand.

Price, M., Tamm, G. & Stantchev, V., 2012. SaaS marketplaces: Visions from theory and experience from practice. In *19th ITS Biennial Conference, Bangkok 2012*. International Telecommunications Society (ITS).

Puri, S.J., 1992. Industrial Vendors' Selling Center: Implications for Sales Management. *Journal of Business & Industrial Marketing*, 7(3), pp.59–69.

Rackham, N. & DeVincentis, J., 1999. *Rethinking the Sales Force: Redefining Selling to Create and Capture Customer Value*, McGraw Hill Professional.

Reinheimer, S., Lang, F., Purucker, J. & Brügmann, H., 2007. 10 Antworten zu SOA. *HMD - Praxis der Wirtschaftsinformatik*, 253(253), pp.7–17.

Rettig, C., 2007. The Trouble With Enterprise Software. *MIT Sloan Management Review*. Available at: http://sloanreview.mit.edu/article/the-trouble-with-enterprise-software/ [Accessed January 5, 2015].

Rimal, B.P., Choi, E. & Lumb, I., 2009. A Taxonomy and Survey of Cloud Computing Systems. In *2009 Fifth International Joint Conference on INC, IMS and IDC*. Ieee, pp. 44–51.

Rivera, J., 2013. Gartner Identifies the Top 10 Strategic Technology Trends for 2014. *Gartner*. Available at: http://www.gartner.com/newsroom/id/2603623 [Accessed November 20, 2013].

Robinson, P.J., Faris, C.W. & Wind, Y., 1967. *Industrial Buying and Creative Marketing*, Allyn and Bacon.

Robra-Bissantz, S., 2012. Empirische Forschung in der Wirtschaftsinformatik. *Enzyklopädie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/uebergreifendes/Forschung-in-WI/Empirische-Forschung-in-der-Wirtschaftsinformatik [Accessed August 10, 2016].

Rogers, E., 1983. *Diffusion of innovations* 3rd ed.,

Rymer, J.R., 2009. Platform-as-a-Service is Here : How to Sift through the Options.

Saldaña, J., 2009. *The Coding Manual for Qualitative Researchers* 2nd Editio., SAGE Publications.

Salesforce.com, 2014. AppExchange. Available at: https://appexchange.salesforce.com/ [Accessed December 23, 2014].

SAP, 2015a. Mobile Solutions. *SAP Store*. Available at: https://store.sap.com/sap/cpa/ui/resources/store/html/Solutions.html?pcntry=&sap-language=EN&catID=MOB&_cp_id=id-1421334997881-0 [Accessed January 15, 2015].

SAP, 2013a. SAP Fiori – Architecture overview. Available at: http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/e03be70d-68fb-3010-c6a0-ff56889c2c81?QuickLink=index&overridelayout=true&58652073478233.

SAP, 2015b. SAP Fiori App Library. Available at: https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/#/detailInfo [Accessed January 15, 2015].

SAP, 2014. SAP Store. Available at: www.sapstore.com [Accessed December 27, 2014].

SAP, 2013b. SAP User Experience Strategy. *SAP*, (October). Available at: http://www.sapdesignguild.org/User_Experience_Strategy_2013_10.pdf.

SAP, 2013c. Simple UI for SAP Applications: SAP Fiori Improves User Experience. Available at: http://en.sap.info/sap-fiori-improves-user-experience/98278 [Accessed November 1, 2013].

SAP SE, 2013. SAP Completes Acquisition of hybris to Help Businesses Deliver Seamless Commerce Experience to Customers. *news.sap.com*. Available at: http://news.sap.com/sap-completes-acquisition-of-hybris-to-help-businesses-deliver-seamless-commerce-experience-to-customers/ [Accessed February 21, 2017].

Schmid, B.F., 1993. Elektronische Märkte. *Wirtschaftsinformatik*, 35(5), pp.465–480.

Schmid, B.F., 1997. Requirements for Electronic Markets Architecture. *Electronic Markets*, 7(1), pp.3–9.

Schoder, D., 2013a. Electronic Commerce. *Enzyklopädie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/informationssysteme/crm-scm-und-electronic-business/Electronic-Business/Electronic-Commerce [Accessed January 21, 2016].

Schoder, D., 2013b. Elektronischer Marktplatz. *Enzyklopädie der Wirtschaftsinformatik*.

Schögel, M., Sauer, A. & Schmidt, I., 2004. Multichannel-Management—Vielfalt in der Distribution. In O. Merx & C. Bachem, eds. *Multichannel-Marketing-Handbuch*. Springer, pp. 1–27.

Scholten, S., 2011. *Platform-based innovation management : a framework to manage open innovation in two-sided platform businesses*, Stuttgart: Universität Stuttgart.

Schreieck, M., Wiesche, M. & Krcmar, H., 2016. Design and Governance of Platform Ecosystems – Key Concepts and Issues for Future Research. In *European Conference of Information Systems (ECIS 2016)*.

Selander, L., Henfridsson, O. & Svahn, F., 2013. Capability search and redeem across digital ecosystems. *Journal of Information Technology*, 28(3), pp.183–197.

Sheng, M.L. & Teo, T.S.H., 2012. Product attributes and brand equity in the mobile domain: The mediating role of customer experience. *International Journal of Information Management*, 32(2), pp.139–146.

Sheth, J.N., 1973. A Model of Industrial Buyer Behavior. *Journal of Marketing*, 37(4), pp.50–56.

Sheth, J.N., 1996. Organizational buying behavior: past performance and future expectations. *Journal of Business Industrial Marketing*, 11(3/4), pp.7–24.

Sinz, E.J., 2010. Konstruktionsforschung in der Wirtschaftsinformatik: Was sind die Erkenntnisziele gestaltungsorientierter Wirtschaftsinformatik-Forschung? In H. Österle, R. Winter, & W. Brenner, eds. *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. pp. 27–33.

Sinz, E.J., Bartmann, D., Bodendorf, F. & Ferstl, O.K., 2011. *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*, University of Bamberg Press.

Sodhi, B., Agrawal, A. & Prabhakar, T. V., 2012. Appification of Web Applications: Architectural Aspects. In *1st IEEE International Conference on Communications in China Workshops (ICCC)*. IEEE.

Stähler, P., 2010. Business Models as a unit of Analysis for Strategizing. Available at: http://www.scribd.com/doc/34770740/Business-Models-as-a-unit-of-Analysis-for-Strategizing.

Stähler, P., 2002. *Geschäftsmodelle in der digitalen Ökonomie. Merkmale Strategien und Auswirkungen* 2nd ed., Josef Eul Verlag.

Thompson, R.L., Higgins, C.A. & Howell, J.M., 1991. Personal computing: Toward a conceptual model of utilization. *MIS Quarterly*, 15(1), pp.124–143.

Timmers, P., 1998. Business models for electronic markets. *Electronic markets*, 8(2), pp.3–8.

Turban, E., 2008. *Electronic Commerce 2008: A Managerial Perspective*, Prentice Hall.

Turowski, K., 2014. Komponentenbasierte Softwareentwicklung. *Enzyklopaedie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Softwarearchitektur/Architekturentwicklung/Komponentenbasierte-Softwareentwicklung/index.html?searchterm=komponenten [Accessed January 12, 2015].

Tyrväinen, P. & Selin, J., 2011. How to sell SaaS: A model for main factors of marketing and selling software-as-a-service. In *ICSOB 2011 - Lecture Notes in Business Information Processing*. pp. 2–16.

Urquhart, C., 2013. *Grounded Theory for Qualitative Research: A Practical Guide*, SAGE Publications.

Urquhart, C., Lehmann, H. & Myers, M.D., 2010. Putting the “theory” back into grounded theory: Guidelines for grounded theory studies in information systems. *Information Systems Journal*, 20, pp.357–381.

Vaquero, L.M., Rodero-Merino, L., Caceres, J. & Lindner, M., 2008. A break in the clouds. *ACM SIGCOMM Computer Communication Review*, 39, pp.50–55.

Veit, D., Clemons, E., Benlian, A., Buxmann, P., Hess, T., Spann, M., Kundisch, D., Leimeister, J.M. & Loos, P., 2014. *Business Models - An Information Systems Research Agenda*,

Venkatesh, V. & Bala, H., 2008. Technology Acceptance Model 3 and a Research Agenda on Interventions. *Decision Sciences*, 39, pp.273–315.

Venkatesh, V., Davis, F.D., Venkatesh, V. & Davis, F.D., 2000. A Theoretical Extension of the Technology Acceptance Model : Four Longitudinal Field Studies. *Management Science*, 46(2), pp.186–204.

Venkatesh, V., Morris, M.G., Davis, G.B. & Davis, F.D., 2003. User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3), pp.425–478.

Verville, J. & Halingten, A., 2003. A six-stage model of the buying process for ERP software. *Industrial Marketing Management*, 32(7), pp.585–594.

Viering, G., Legner, C. & Ahlemann, F., 2009. The (Lacking) Business Perspective on SOA-Critical Themes in SOA Research. In *Wirtschaftinformatik Proceedings 2009*.

Vizard, M., 2012. CIOs Struggle With Relevance of Role to Business. *CIO Insight*. Available at: http://www.cioinsight.com/it-management/cios-struggle-with-relevance-of-role-to-business [Accessed November 1, 2013].

Wagner, D., Suchan, C., Leunig, B. & Frank, J., 2011. Towards the Analysis of Information Systems Flexibility: Proposition of a Method. In *Proceedings of the 10th International Conference on Wirtschaftsinformatik (WI)*.

Weber, M., 2009. Cloud Computing-Evolution in der Technik, Revolution im Business. *BITKOM Report*.

Webster, F.E. & Wind, Y., 1972. A General Model for Understanding Organizational Buying Behavior. *The Journal of Marketing*, 36(2), pp.12–19.

Webster, J. & Watson, R., 2002. Analyzing the past to prepare for the future: Writing a literature review. *Management Information Systems Quarterly*, 26(2).

Weill, P., 2004. Don't just lead, govern: How top-performing firms govern IT. *MIS Quarterly Executive*, 3(1), pp.1–17.

Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W. & Stoesser, J., 2011. *Cloud Computing – A Classification, Business Models and Research Directions*,

Weiß, F. & Leimeister, J.M., 2012a. Consumerization. *Business & Information Systems Engineering*, 4(6), pp.363–366.

Weiß, F. & Leimeister, J.M., 2012b. Consumerization - IT Innovations from the Consumer Market as a Challenge for Corporate IT. *Business & Information Systems Engineering*, 4(6), pp.363–366.

Wenzel, S., 2014a. *App'ification of Enterprise Software - Evaluating Mobile App Characteristics Enabling Online Purchase And Their Portability To Enterprise Application Software*,

Wenzel, S., 2014b. App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores. In *Proceedings of the 5th International Conference on Software Business (ICSOB 2014) - Lecture Notes in Business Information Processing Volume 182*. Springer, pp. 227–242.

Wenzel, S., Faisst, W., Burkard, C. & Buxmann, P., 2012. New Sales and Buying Models in the Internet: App Store Model for Enterprise Application Software. In *MKWI 2012*. pp. 639–651.

Wenzel, S., Neumann, S., Faisst, W. & Bandulet, F., 2010. Electronic Business Services and their Role for Enterprise Software. In A. Benlian, T. Hess, & P. Buxmann, eds. *Software-as-a-Service - Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen*. Wiesbaden: Gabler, pp. 75–91.

Wigand, R., 2012. Information Systems. *Enzyklopaedie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/uebergreifendes/Kerndisziplinen/Information-Systems/index.html [Accessed January 30, 2016].

Wilde, T., Hess, T. & Kfm, D., 2007. Forschungsmethoden der Wirtschaftsinformatik Eine empirische Untersuchung. , 49, pp.280–287.

Wilson, J., 2010. *Essentials of Business Research: A Guide to Doing Your Research Project*,

Winkelmann, P., 2008. *Marketing und Vertrieb* 5th ed., München: Oldenbourg.

Yin, R.K., 2013. *Case Study Research: Design and Methods (Applied Social Research Methods)*, SAGE Publications, Inc.

Yoffie, D.B. & Kwak, M., 2006. With Friends Like These: The Art of Managing

Complementors. *Havard Business Review*, 84(9), pp.88–98.

Zelewski, S., 2014. Wissenschaftstheorie. *Enzyklopädie der Wirtschaftsinformatik*. Available at: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/uebergreifendes/Forschung-in-WI/Wissenschaftstheorie/index.html?searchterm=konstruktivis [Accessed July 21, 2016].

Zencke, P. & Eichin, R., 2008. SAP Business ByDesign. *Business & Information Systems Engineering*, 50(1), pp.47–51.

Zott, C., Amit, R. & Massa, L., 2011. The Business Model: Recent Developments and Future Research. *Journal of Management*, 37(4), pp.1019–1042.

# Appendix A. App Stores for Enterprise Application Software

**Table 28: Overview selected app stores for enterprise application software**

| # | Company | App Store | URL | Vendors | Launch Year | Application Type | Comment |
|---|---|---|---|---|---|---|---|
| 1 | Salesforce.com | AppExchange | https://appexchange.salesforce.com | Salesforce.com, Partners | 2006 | SaaS | |
| 2 | SugarCRM | SugarExchange | https://sugarexchange.sugarcrm.com | SugarCRM, Partners | 2006 | SaaS | |
| 3 | Microsoft | Microsoft Pinpoint | http://pinpoint.microsoft.com | Microsoft, Partners | 2008 | On-premise, SaaS | retired in 2016 |
| 4 | SAP | SAP Ecohub | http://ecohub.sap.com | Partners | 2008 | On-premise, SaaS | retired in 2011, redirects to the SAP Store |
| 5 | Zoho | Zoho Creator Marketplace | https://marketplace.zoho.com | Zoho, Partners | 2008 | SaaS | |
| 6 | Netsuite | SuiteApp.com | http://suiteapp.com | Partners | 2009 | SaaS | |
| 7 | Google | Google Apps Marketplace | https://apps.google.com/marketplace | Google, Partners | 2010 | SaaS | |
| 8 | SAP | SAP Store | https://www.sapstore.com | SAP, Partners | 2011 | On-premise, SaaS | |
| 9 | Amazon | AWS Marketplace | https://aws.amazon.com/marketplace | Amazon, Partners | 2012 | SaaS, (On-premise) | on-premise apps via virtualized environment |
| 10 | Telekom | Business Marketplace | https://apps.telekomcloud.com | Partners | 2012 | SaaS | |
| 11 | Intuit | Intuit Quickbooks Apps Store | https://apps.intuit.com | Intuit, Partners | 2013 | On-premise, SaaS | |
| 12 | Microsoft | Office Store | https://store.office.com | Microsoft, Partners | 2013 | On-premise, SaaS | |
| 13 | Sage | Sage Partner Solution Source | http://sagepss.com | Partners | 2013 | On-premise, SaaS | |
| 14 | SAP | SAP HANA Marketplace | http://marketplace.saphana.com | SAP, Partners | 2013 | On-premise, SaaS | retired in 2016 |
| 15 | SAP | SAP App Center | https://www.sapappcenter.com | Partners | 2015 | On-premise, SaaS | |
| 16 | Software AG | Digital Marketplace | https://marketplace.softwareag.com | Software AG, Partners | 2015 | On-premise, SaaS | |
| 17 | Microsoft | Microsoft App Source | https://appsource.microsoft.com | Microsoft, Partners | 2016 | SaaS | |

The table was updated on the 17th of October 2016.

# Appendix B. Paper 1: Reviewed Platform-as-a-Service Offerings

**Table 29:** Overview of PaaS offerings reviewed for Paper 1

| # | Vendor | PaaS Offering | URL (updated August 2016) | PaaS Type | IDE | App Store | App Store URL | Billing Service |
|---|---|---|---|---|---|---|---|---|
| 1 | Amazon | Amazon WebServices | https://aws.amazon.com | pure PaaS | IDE neutral | AWS Marketplace (launched 2012) | https://aws.amazon.com/marketplace | (✓) |
| 2 | AdventNet | Zoho Creator | https://www.zoho.com/creator/ | aPaaS | proprietary IDE | Zoho Creator Marketplace | https://marketplace.zoho.com/ | (✓) |
| 3 | Bungee Labs | BungeeConnect | http://www.bungeelabs.com/ | pure PaaS | based on Eclipse | ✗ | ✗ | ✗ |
| 4 | Caspio | Caspio | https://www.caspio.com/ | pure PaaS | visual proprietary IDE (4GL) | ✗ | ✗ | ✗ |
| 5 | Daptiv | Daptiv Dynamic Applications | http://www.daptiv.com/products.htm | aPaaS | proprietary IDE | ✗ | ✗ | ✗ |
| 6 | Google | Google AppEngine / Google Cloud Platform | https://cloud.google.com/appengine/ | pure PaaS + aPaaS | IDE neutral | Google Apps Marketplace | https://apps.google.com/marketplace/ | ✓ |
| 7 | Intuit | Intuit Developer Group | http://developer.intuit.com | aPaaS | proprietary IDE | Intuit Quickbooks Apps Store | https://apps.intuit.com/ | ✓ |
| 8 | Magic Software | Magic Software XPA Application Platform | https://www.magicsoftware.com/magic-xpa-application-platform | pure PaaS | visual proprietary IDE (4GL) | ✗ | ✗ | ✗ |
| 9 | Micro Focus | Micro Focus Enterprise Cloud Services | https://www.microfocus.com/ | pure PaaS | IDE neutral | ✗ | ✗ | ✗ |
| 10 | Microsoft | Microsoft Azure Services Platform | https://azure.microsoft.com/ | pure PaaS + aPaaS | IDE neutral | Microsoft App Source | https://appsource.microsoft.com/ | (✓) |
| 11 | Microsoft | Microsoft SharePoint Online | https://products.office.com/en-us/sharepoint/sharepoint-online-collaboration-software | aPaaS | proprietary IDE | Office Store | https://store.office.com/ | (✓) |
| 12 | OrangeScape Technologies | OrangeScape PaaS Platform for Enterprises | http://www.orangescape.com/paas/platform-as-a-service/ | pure PaaS | visual proprietary IDE (4GL) | ✗ | ✗ | ✗ |
| 13 | Salesforce.com | Force.com | https://developer.salesforce.com/platform/force.com | pure PaaS + aPaaS | proprietary IDE | AppExchange | https://appexchange.salesforce.com/ | ✓ |
| 14 | SAP | SAP PartnerEdge Program for Application Development | https://www.sapappsdevelopmentpartnercenter.com/ | pure PaaS + aPaaS | IDE neutral | SAP App Center | https://www.sapappcenter.com/ | ✓ |
| 15 | Software AG | Agile Apps Cloud (previously Longjump) | http://www.softwareag.com/agileappscloud/ | aPaaS | visual proprietary IDE (4GL) | Digital Marketplace | https://marketplace.softwareag.com/ | ✗ |

Table 29 provides an overview of PaaS offerings. The list provides information on the vendor, the PaaS specific offering, the PaaS type, the Integrated Development Environment (IDE), the corresponding app store of the platform, and whether there is a billing service available. “aPaaS” in the Table refers to application-based PaaS, and “IDE neutral” means there is support for a variety of IDEs and tools.

The review of the PaaS offerings was initially conducted between October 2009 and January 2010 and used as a basis for Paper 1 (see Chapter II.1). The provider selection has been inspired by the analyst report by Rymer in 2009 and was extended by my own research efforts (Rymer 2009). The overview was originally published online under “http://isdl.uni-bamberg.de/paas/appendix.pdf” but has subsequently been taken offline.

Table 29 is based on the original table but the entries were checked and updated in August 2016. Compared to the original list I have not added new providers or offerings. However I have updated the offerings’ names, URLs, and all other entries in case of changes to provide valuable and relevant information beyond documentary purposes. Compared to the original review only one offering was discontinued and removed from the list (i.e., Stax networks was acquired by Cloudbees Inc.). The product-focused SAP Business ByDesign partner program was replaced by a SAP-company-wide partner program covering a multitude of technologies, deployment models and SAP products. In general, it can be stated that the original offerings are still present with their capabilities, but have been enhanced and differentiated with a multitude of options making a clear classification more complicated.

# Appendix C. Paper 2: Instrument Details

Due to the rigid page limitations of the publication template, Paper 2 only included the instrument categories, with selected instruments as examples. Therefore, this appendix lists the instrument categories with all instruments which could be extracted from the qualitative data. The category “infrastructure tools” in the paper can be further broken down into the categories “development tools” and “marketing tools”. The term tool is used rather broadly and entails elements which also include manual activities (e.g., architecture review) or a combination of manual activities and software tools (e.g., certification).

**Table 30:** Instrument categories and instruments

| Instrument categories | Instruments |
|---|---|
| Information Resources | ▪ developer community<br>▪ business process expert community<br>▪ technical documentation<br>▪ expert material (best practice case studies)<br>▪ sample code / applications<br>▪ newsletter<br>▪ partner portal (restricted to partners only) |
| Personal Contact | ▪ dedicated partner manager<br>▪ platform or partner conferences<br>▪ regular Q&A calls and info-sessions<br>▪ support hotline<br>▪ contact person in partner country or language |
| Development Tools (Infrastructure Tools) | ▪ comprehensive development environment<br>▪ test environment<br>▪ operations monitoring<br>▪ performance monitoring<br>▪ technology trials<br>▪ installation routines<br>▪ migration wizard (migrate from other platforms)<br>▪ service and support infrastructure<br>▪ individual architecture reviews<br>▪ code checks<br>▪ certification |
| Marketing Tools (Infrastructure Tools) | ▪ joint marketing and sales initiatives<br>▪ joint press releases<br>▪ app store, app marketplace<br>▪ billing & collection services / payment handling<br>▪ app challenges<br>▪ partnership logo<br>▪ certification logo<br>▪ product marketing guide (“how to sell via app store”) |
| Training | ▪ certification for developers<br>▪ online trainings<br>▪ classroom trainings<br>▪ code jam |
| Formal Agreement | ▪ partner contract<br>▪ embedded licensing<br>▪ software licensing in general |

# Appendix D. Paper 6: Case Selection

**Table 31: Assessment of cases for Paper 6**

| App Store Type | Provider | App Store | URL | Access to public documentation | System access (e.g., trial, demo) | Access to subject matter experts | Representativeness of system | Overall |
|---|---|---|---|---|---|---|---|---|
| Internal | App47.com | App47.com | http://www.app47.com/ | ◑ | ○ | ○ | ● | ◔ |
| Internal | OpenPeak | OpenPeak Openshop | https://www.openpeak.com/ | ◑ | ○ | ○ | ◑ | ◔ |
| Internal | Salesforce.com | Private AppExchange | https://appexchange.salesforce.com/listingDetail?listingId=a0N3000000B4VlZEAV | ◑ | ◑ | ○ | ● | ◑ |
| Internal | SAP | SAP Enterprise Store | http://help.sap.com/ses | ◑ | ● | ● | ● | ● |
| Internal | Symantec | Symantec App Center | http://www.symantec.com/connect/groups/symantec-app-center | ◑ | ○ | ○ | ◑ | ◔ |
| Public | Amazon | AWS Marketplace | https://aws.amazon.com/marketplace | ◑ | ● | ◑ | ● | ◕ |
| Public | Fujitsu | Fujitsu Cloud Store | http://cloudstore.ts.fujitsu.com/ | ◑ | ○ | ○ | ○ | ○ |
| Public | Google | Google Apps Marketplace | https://apps.google.com/marketplace/ | ● | ● | ◑ | ● | ● |
| Public | Intuit | Intuit Quickbooks Apps Store | https://apps.intuit.com/ | ◑ | ◑ | ○ | ◑ | ◔ |
| Public | Microsoft | Microsoft Pinpoint | http://pinpoint.microsoft.com | ◑ | ◑ | ○ | ◑ | ◔ |
| Public | Netsuite | SuiteApp.com | http://suiteapp.com/ | ◑ | ◑ | ○ | ◑ | ◔ |
| Public | Oracle | Oracle Store | https://shop.oracle.com/ | ◑ | ◑ | ○ | ○ | ◔ |
| Public | Salesforce.com | AppExchange | https://appexchange.salesforce.com/ | ● | ● | ○ | ● | ◕ |
| Public | SAP | SAP Store | www.sapstore.com | ◑ | ● | ● | ● | ● |
| Public | SugarCRM | SugarExchange | https://sugarexchange.sugarcrm.com/ | ◑ | ◑ | ○ | ◕ | ◑ |
| Public | Telekom | Business Marketplace | https://cloud.telekom.de/ | ◑ | ◑ | ○ | ○ | ◔ |
| Public | Zoho | Zoho Creator Marketplace | https://marketplace.zoho.com/ | ◑ | ◑ | ○ | ◑ | ◔ |

Table 31 shows the assessment carried out for Paper 6 to identify the cases used for the case study. In total five internal and elven public app stores were shortlisted and reviewed. The criteria were chosen to identify those app stores for which the most data could be retrieved and which app store could be classified as a typical representation of its kind. The criteria ‘access to subject matter experts’ as well as ‘system access’ was rated from the researcher’s subjective perspective, or in other words whether I was able to gain access to relevant experts via my personal and professional network, and how I could rate the likelihood of gaining access to a system in order to evaluate the application “hands-on”. The harvey balls in the table should be understood as follows: no (empty/white ball), low, medium, high, very high (full/black ball).

# Appendix E. Paper 6: Revised and Enhanced Conference Paper

Paper 6 (see Chapter IV.1), which is part of this thesis, is an extended version of a conference paper (Wenzel 2014), which was awarded with the “Best Paper Award” by the program committee. The citation of the original conference version reads as follows:

> *Wenzel, Stefan.: App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores. Proceedings of the 5th International Conference on Software Business. Lecture Notes in Business Information Processing Volume 182, 2014, pp 227-242.*

An extended version has been created to overcome the limitations of the conference template, as I felt those limitations also led to shortcomings with regards to readability and the adequate presentation of methodology and results. Moreover, despite having received the best paper award, I also received very useful feedback as part of the reviews and following the presentation of the paper. Hence I decided to develop an improved and extended version of the paper which is included in this thesis.

Since, the extended version includes all the relevant aspects of the conference version, the latter one was not included in this thesis. In compliance with the conference guidelines, a revised version of the published paper can only be published again in case the revised version contains at least 30% new material. The following list proves that the extended version is in full compliance with these regulations.

**Content-Related Changes**

Following is a summary of which aspects have been enhanced in the extended version:

- I have adjusted and simplified the title to “Comparing Public and Internal Enterprise App Stores: A Qualitative Case Study”.
- The research objectives in Chapter 1 have been sharpened, especially by formulating three explicit research questions instead of only describing the targets.
- The Chapter “1 Introduction” has been further enhanced to reflect new content in the paper and to better describe how the postulated research questions will be addressed.
- Chapter 2 “Related Work” has been enhanced; in particular I have re-written and significantly enhanced the section the “IT Governance and IT Management.
- The methodology and research process (Chapter 3) has been significantly enhanced by analyzing the two cases using the “Semantic Object Model (SOM). In detail, the SOM-

Modeling methodology was used to analyze the business systems of EASs and replaced the vague concept of value systems by Porter with a systematic approach. So-called interaction schemas (an element of the SOM methodology) were used to model the business systems and the universe of discourse of the respective EAS. This helped to better convey the service relationships and coordination transactions as well as the business objects (stakeholders, roles and resources) involved in these transactions.

- The introduced methodology to analyze business systems was applied to the individual case studies. Therefore, Chapter 4 "Presentation of cases" was significantly enhanced by the aforementioned interaction schemes and a more in-depth analysis of the business systems, involved business objects and transactions. The entire analysis is now presented in greater detail.
- Chapter 5 "Comparison of Public versus Internal EAS" has also been significantly enhanced. More detail is provided with regards to the decentralized adoption of corporate IT and the influence of public and internal EAS on the IT Governance and IT Management processes "Sourcing, Delivery, Support, Monitoring, and Control".
- Furthermore, a comparison of a traditional software buying process versus an "app store oriented process" has been introduced.
- Chapter 6 "Integration of Public and Internal EAS" has been newly introduced. It includes the earlier "proposal on combining public and internal EAS" which was enhanced by a discussion of the problems posed by the combination of public and internal EAS. Furthermore, a proposal of how to integrate public and internal EAS was added. The integration scenario includes a model presented as SOM interaction scheme and a discussion of the most important requirements and pre-requisites to realize such an integration scenario.
- Last, Chapter 7 was significantly revised and enhanced. It now includes a more thorough reflection on limitations and future research, and additionally the conclusion reflects on the initially stated research questions.

**Changes in "Numbers"**

- Paper length (excluding references and abstract): ~8200 words in extended version versus ~6000 words in conference version (increase of ~36%).
- Paper length in pages: 22 (extended version) versus 15 pages (conference version) using the original conference template. Increase of ~46%.
- The conference version included 5 figures and 1 table. The extended version now includes 7 figures and 2 tables.

# Appendix F. Chapter 5: Supply-Side Value Chain: Transaction and Resource Analysis

### Supply-Side Transaction Analysis and Resource Support

Table 32 lists the result of the detailed transaction analysis on the supply-side. The first column "ID (Type)" provides a transaction ID where the "S" stands for supply-side, the first digit for a phase in the ISV lifecycle, the second digit for the individual transaction. The type of transaction is classified in brackets in the same column (I= initiating, C=contracting, e=enforcing). The transactions have been further grouped in "ISV Lifecycle Phases" providing an additional structure. The last column "Direction" indicates the direction of the transaction, either originating from the *Platform Provider* directed to the *ISV* or vice versa. The respective transactions have not been assigned to *Platform Provider* internal business objects at this level, since more complex company-internal coordination and compound service delivery transactions might be involved. This level of detail will also be part of the concretization of the model when implementing or deriving business process model instances.

The transaction overview focuses on the service transactions between the *ISV* and the *Platform Provider* as well as the major coordinating transactions, which are mostly dedicated "requests" for service provided by the *Platform Provider*. Moreover, every transaction has been assigned one or more resources supporting the transaction. In case there is more than one resource assigned, the transaction is either still on an aggregate level and may need further decompositions to uncover sub-transactions, or it indicates an option space where different resources might be assigned to the respective task. E.g., depending on the desired level of automation, "program information" (transaction S2.2) provided to the ISV may be delivered via the *ISV Portal* or in-person via a *Partner Manager*, or by both in semi-automated. This level of detail is not covered or recommended by the presented model artefact but needs to be decided on when transferring the model into practice and deriving concrete business process models and blueprints.

**Table 32: Overview supply-side transactions and resource support**

| ID (Type) | ISV Lifecycle Phase | Transaction | Resource Support (Platform Provider) | Direction |
|---|---|---|---|---|
| S1.1 (I) | Become Aware | Provide ISV program marketing material | ISV Portal | PP→ISV |
| S2.1 (C) | Evaluate | Register for information service | ISV Portal | ISV→PP |
| S2.2 (E) | Evaluate | Provide detailed program information | ISV Portal, Partner Manager | PP→ISV |
| S2.3 (C) | Evaluate | Request access to trial | ISV Portal | ISV→PP |
| S2.4 (E) | Evaluate | Provide platform & technology trials | IDE, ARE, Cloud Infrastructure, ISV Portal | PP→ISV |
| S3.1 (C) | Partner & Setup | Sign-up for developer program | ISV Portal | ISV→PP |
| S3.2 (E) | Partner & Setup | Provide contract, systems & resource access | ISV Portal, Partner Manager | PP→ISV |

| ID (Type) | ISV Lifecycle Phase | Transaction | Resource Support (Platform Provider) | Direction |
|---|---|---|---|---|
| S3.3 (C) | Partner & Setup | Create platform account settings & dev. user | ISV Portal, IDE | ISV→PP |
| S3.4 (E) | Partner & Setup | Provide IDE/SDK setup guidance & support | ISV Portal, Partner Manager, Platform Dev. Experts | PP→ISV |
| S4.1 (E) | Develop | Provide and operate IDE and required systems | Cloud Infrastructure, IDE, ARE | PP→ISV |
| S4.2 (E) | Develop | Provide development support & enablement | ISV Portal, Dev. Community, Platform Dev. Experts | PP→ISV |
| S4.3 (C) | Develop | Request app review & certification | ISV Portal, Certification Experts | ISV→PP |
| S4.4 (E) | Develop | Provide app certification | Certification Experts, ISV Portal | PP→ISV |
| S5.1 (I) | Market | Provide app store guidelines & best practices | App Store, ISV Portal | PP→ISV |
| S5.2 (C) | Market | Request app listing | App Store, Partner Manager, App Store Experts | ISV→PP |
| S5.3 (E) | Market | Review & Publish app on app store | App Store, Marketing & App Store Experts | PP→ISV |
| S5.4 (E) | Market | Provide value-add marketing & promotion services | ISV Portal, Marketing & App Store Experts | PP→ISV |
| S6.1 (E) | Sell | Provide app sales service | App Store, Marketing & App Store Experts | PP→ISV |
| S6.2 (E) | Sell | Provide billing & collection service | App Store, Sales & Marketing Experts | PP→ISV |
| S7.1 (E) | Deploy & Operate | Provide deployment service to software customer | App Store, ARE, Cloud Infrastructure | PP→ISV |
| S7.2 (E) | Deploy & Operate | Operate ISV application on-behalf (for cloud apps only) | ARE, Cloud Infr., Platform Dev. & Ops Experts | PP→ISV |
| S7.3 (C) | Deploy & Operate | Request support | Ticketing & Support System, Dev. & Ops Experts | PP→ISV |
| S7.4 (E) | Deploy & Operate | Provide implementation or application support | Ticketing & Support System, IDE | ISV→PP |
| S8.1 (C) | Settlement | Provide billing overview & payment request | App Store | PP→ISV |
| S8.2 (E) | Settlement | Pay revenue share and platform services fees | App Store, Sales & Marketing Experts | ISV→PP |

### Supply-Side Resource Analysis: Functions and Capabilities

Table 33 lists all identified personnel and application system resources of the platform provider relevant to the modeled supply-side business processes. The functions and capabilities have been derived and proposed based on the business processes and more precisely the tasks to be performed by the business objects as part of the transactions (see Table 32). They only reflect the critical requirements towards the resources and are only a starting point for a detailed requirements specification and blueprinting process. For personnel resources the described capabilities represent critical elements to be included in the respective job profiles.

**Table 33: Overview resources functions and capabilities (supply-side)**

| Resource | Type | Major capabilities / functions / job profiles |
|---|---|---|
| ISV Portal | System | ▪ web portal with public access and protected access / access for registered users<br>▪ user management and self-registration<br>▪ content-management system supporting web content: incl. text, video streaming, audio streaming, document hosting and downloads<br>▪ newsletter distribution<br>▪ web forms to send structured requests (e.g., sign-up for partnership)<br>▪ register for e-learning, live expert sessions |
| Developer Community | System | ▪ wiki system<br>▪ forum system<br>▪ micro blogs |
| Ticketing & Support System | System | ▪ create support tickets<br>▪ track ticket status<br>▪ assign tickets to users |
| IDE | System | ▪ SDK / libraries / programing languages<br>▪ editor, compiler, debugger<br>▪ version control and source code management<br>▪ test management and test automation (e.g., unit test, performance test, UI test)<br>▪ transport and component integration |
| ARE | System | ▪ all system components to allow an application to be executed<br>▪ application management and application monitoring (ISV level)<br>▪ multi-tenancy management<br>▪ reliability / availability components (e.g., redundant systems)<br>▪ scalability (automatically assign additional resources)<br>▪ security (semantic security checks) |
| Cloud Infrastructure | System | ▪ system management (low level, to be accessed by platform provider only)<br>▪ security (technical security checks / low level, e.g., DDOS)<br>▪ hardware components (CPU & storage) or their virtualized equivalents<br>▪ system software, databases, and middleware components<br>▪ operations and monitoring systems (e.g., health checks)<br>▪ backup<br>▪ disaster recovery<br>▪ file hosting / distribution |
| Partner Manager | Personnel | ▪ central contact person for all ISV program related inquiries<br>▪ consulting on ISV program scope and options<br>▪ consulting on all contractual and commercial aspects related to the program<br>▪ provides information, support, guidance and issue resolution<br>▪ local language, local time zone<br>▪ can be contacted by phone<br>▪ provides dedicated services (e.g., Q&A sessions)<br>▪ fixed assignment to partner over period of time |
| Platform Development & Operations Experts | Personnel | ▪ technical expert (developer, development architect, cloud operations expert) on all IDE, ARE, cloud infrastructure, or other technology related issues<br>▪ provide expert sessions on selected topics (trainings, e-learnings)<br>▪ provide developer certification trainings<br>▪ engage in the community (forum monitoring, micro-blogging on trending topics)<br>▪ support on technical issues and inquiries (e.g., tickets on platform components)<br>▪ collect technical and platform requirements |
| 3rd Party Application Certification Experts | Personnel | ▪ consults on app certification guidelines<br>▪ reviews applications during certification process<br>▪ provides feedback to ISV in case of identified issues with resolution proposals |
| Marketing & App Store Experts | Personnel | ▪ consults on app store listing, sales, marketing and billing related matters<br>▪ reviews and approve app store listing content (partner logo, description, pricing scheme, terms and conditions etc.)<br>▪ provide and coordinate dedicated marketing and sales services (e.g., co-branding, featured apps) |

| Resource | Type | Major capabilities / functions / job profiles |
|---|---|---|
| Public App Store<br><br>(Supply-side / ISV-related capabilities only) | System | ▪ upload / edit app listing<br>▪ message inbox (quote requests, contact requests)<br>▪ sales order management<br>▪ sales statistics<br>▪ platform contract & platform services consumption overview (commercial view)<br>▪ billing & payment console<br>▪ account and user management |

# Appendix G. Chapter 5: Demand-Side Value Chain: Transaction and Resource Analysis

### Demand-Side Transaction Analysis and Resource Support

Table 34 lists the result of the detailed transaction analysis on the demand-side. The first column “ID (Type)” provides a transaction ID where the “D” stands for demand-side, the first digit for a phase in the customer buying process, the second digit for the individual transaction. The type of the transaction is classified in brackets in the same column (I= initiating, C=contracting, e=enforcing). The transactions have been further grouped in “Buying Process Phases” providing an additional structure. The last column “Direction” indicates the direction of the transaction, either originating from the *Platform Provider* directed to the *Software Customer* or vice versa.

As already explained for the transaction analysis on the supply-side (see Appendix F), the respective transactions have not been assigned to *Platform Provider* internal business objects. Moreover, the transaction overview focuses on the service transactions between the *Software Customer* and the *Platform Provider* as well as the major coordinating transactions, which are mostly dedicated “requests” for services provided by the *Platform Provider*. Furthermore, every transaction has been assigned one or more resources supporting the transaction.

**Table 34: Overview demand-side transactions and resource support**

| ID (Type) | Buy Process Phase | Transaction | Resource Support (Platform Provider) | Direction |
|---|---|---|---|---|
| D1.1 (I) | Problem Rec. | Provide context-sensitive product recommendations | App Store | PP→SC |
| D1.2 (I) | Problem Rec. | Push recommendations to users | App Store | PP→SC |
| D1.3 (I) | Problem Rec. | In-product recommendations | App Store | PP→SC |
| D1.4 (I) | Problem Rec. | Provide “awareness” material (e.g., success stories) | App Store | PP→SC |
| D2.1 (I) | Info. Search | Provide app catalog and catalog discovery | App Store | PP→SC |
| D2.2 (I) | Info. Search | Provide app information content (e.g., text, docs, pics, videos) | App Store | PP→SC |
| D2.3 (I) | Info. Search | Provide reviews and user ratings | App Store | PP→SC |
| D2.4 (C) | Info. Search | Request remote buying support and consultancy | App Store | SC→PP |
| D2.5 (E) | Info. Search | Provide remote consultancy | Remote Sales Agents | PP→SC |
| D2.6 (C) | Info. Search | Register for live online session (expert session / webinar) | App Store | SC→PP |
| D2.7 (E) | Info. Search | Provide live online session | Specialized or product pre-sales | PP→SC |
| D3.1 (C) | Evaluation | Request demo / trial | App Store | SC→PP |
| D3.2 (E) | Evaluation | Provide demo / trial | App Store, ARE, Cloud Infr. | PP→SC |
| D3.3 (I) | Evaluation | Provide guide /assistance for complex applications and solution bundles (incl. app + service) | App Store | PP→SC |
| D3.4 (C) | Evaluation | Request system landscape compatibility check | App Store | SC→PP |
| D3.5 (E) | Evaluation | Provide compatibility check | App Store, ARE, Cloud Infr. | PP→SC |
| D3.6 (I) | Evaluation | Provide project & service plan evaluation (for complex apps) | App Store | PP→SC |
| D3.7 (I) | Evaluation | Provide group evaluation capability | App Store | PP→SC |

| ID (Type) | Buy Process Phase | Transaction | Resource Support (Platform Provider) | Direction |
|---|---|---|---|---|
| D3.8 (C) | Evaluation | Request remote or on-site product consultancy | App Store | SC→PP |
| D3.9 (E) | Evaluation | Provide product consultancy service | Field Sales Agents, Specialized or product pre-sales | PP→SC |
| D4.1 (I) | Neg.&Purchase | Provide pricing configuration (multi-dimensional pricing, service pricing, discount schemes, taxation) | App Store | PP→SC |
| D4.2 (C) | Neg.&Purchase | Add to shopping cart | App Store | SC→PP |
| D4.3 (C) | Neg.&Purchase | Generate approval request for 3rd party procurement system | App Store | SC→PP |
| D4.4 (C) | Neg.&Purchase | Approve shopping cart & buy | App Store | SC→PP |
| D4.5 (C) | Neg.&Purchase | Request quotation | App Store | SC→PP |
| D4.6 (E) | Neg.&Purchase | Provide quotation | Remote Sales Agents | PP→SC |
| D4.7 (C) | Neg.&Purchase | Accept quotation & add to cart | App Store | SC→PP |
| D5.1 (E) | SW Provisioning | Provide download links and instructions (on-premise SW) | App Store, Cloud Infr. | PP→SC |
| D5.2 (E) | SW Provisioning | Provide application link and user activation note (cloud SW) | App Store, Cloud Inf., ARE | PP→SC |
| D5.3 (E) | SW Provisioning | Provide download to device (personal software) | App Store, Cloud Infr. | PP→SC |
| D5.4 (E) | SW Provisioning | Provide application service (cloud) | Cloud Inf., ARE | PP→SC |
| D5.5 (C) | SW Provisioning | Request application support | Ticketing & Support System | SC→PP |
| D5.6 (E) | SW Provisioning | Provide support | Platform Operations Experts | PP→SC |
| D5.7 (E) | SW Provisioning | Provide Internal App Store “as-a-Service” to software customers | Internal App Store, Cloud Inf., ARE | PP→SC |
| D6.1 (C) | Settlement & Aftersales | Provide billing overview & payment request | App Store | PP→SC |
| D6.2 (E) | Settlement & Aftersales | Trigger payment | App Store | SC→PP |
| D6.3 (I) | Settlement & Aftersales | Provide contract & licensing overview (status & usage) | App Store | PP→SC |
| D6.4 (C) | Settlement & Aftersales | Terminate / enhance existing contracts / license agreements | App Store | SC→PP |
| D6.5 (E) | Settlement & Aftersales | Update license agreements & contracts | App Store | PP→SC |

**Demand-Side Resource Analysis: Functions and Capabilities**

Table 35 lists all the identified personnel and application system resources of the platform provider relevant to the modeled demand-side business processes. The functions and capabilities have been derived and proposed based on the business processes and more precisely the tasks to be performed by the business objects as part of the transactions (see Table 34). They only reflect the critical requirements towards the resources and are only a starting point for a detailed requirements specification and blueprinting process. For personnel resources the described capabilities represent critical elements to be included in the respective job profiles.

The functions and capabilities of the systems *Public App Store* and *Internal App Store* have been further structured. For the *Public App Store,* the functional categories are oriented towards the six buying process phases (cf. Figure 55) used for the detailed transaction analysis plus an administration category. For the *Internal App Store* the categories are aligned with the process phases introduced for internal distribution and management of apps (cf. Figure 58)

**Table 35: Overview resources functions and capabilities (demand-side)**

| Resource | Type | Major capabilities / functions / job profiles |
|---|---|---|
| ARE | System | ▪ all system components to allow an application to be executed<br>▪ system management and operations monitoring (ISV level)<br>▪ multi-tenancy management<br>▪ reliability / availability components (e.g., redundant systems)<br>▪ scalability (automatically assign additional resources)<br>▪ security (semantic security checks) |
| Cloud Infrastructure | System | ▪ system management (low level, to be accessed by platform provider only)<br>▪ security (technical security checks / low level, e.g., DDOS)<br>▪ hardware components (CPU & storage) or their virtualized equivalents<br>▪ system software, databases, and middleware components<br>▪ operations and monitoring systems (e.g., health checks)<br>▪ backup<br>▪ disaster recovery<br>▪ file hosting / distribution |
| Ticketing & Support System | System | ▪ create support tickets<br>▪ track ticket status<br>▪ assign tickets to users |
| Platform Operations Experts | Personnel | ▪ provide software customer support (i.e., first / second level support)<br>▪ interact with ISV on application issues which cannot be resolved in first/second level support<br>▪ resolve issues w/r to application hosting and operations |
| Remote Sales Agent | Personnel | ▪ provides remote support on buying software via the app store<br>▪ uses multiple remote channels to provide support: phone, chat, online conferencing, co-browsing<br>▪ is familiar with the app store processes and capabilities<br>▪ has an overview on the app store portfolio<br>▪ can answer basic questions w/r to application capabilities and scope<br>▪ creates quotations<br>▪ resolves issues with regards to app store administration (e.g., user or account settings) |
| Field Sales Agent | Personnel | ▪ conducts on-site customer meetings<br>▪ pursues a consultative sales approach matching complex customer needs to the product and service portfolio<br>▪ supports the online buying process with dedicated customer consulting needs<br>▪ is familiar with the entire product and service portfolio of the platform provider (both online and offline portfolio)<br>▪ creates quotations<br>▪ closes contracts offline |
| Specialized Sales or Product Pre-Sales Agent | Personnel | ▪ conducts remote and on-site customer meetings<br>▪ supports either remote or field sales agents in case special knowledge is required<br>▪ has deep product expertise for few selected products<br>▪ specialized sales agents are either employed at the platform provider in case of platform core applications or employed with the ISV. |
| **Public App Store** | | |
| Public App Store: Administration | System | ▪ corporate user management / invite users<br>▪ user roles (anonymous user w/o company assignment, corporate business user, corporate buyer)<br>▪ customer account management with corporate discount schemes, frame contracts<br>▪ multi-language<br>▪ multi-country support |

| Resource | Type | Major capabilities / functions / job profiles |
|---|---|---|
| Public App Store: Problem Recognition | System | ▪ product recommendation engine (using user profiles, company profile, buying history)<br>▪ push recommendations to users beyond store boundaries: e.g., newsletter, social media<br>▪ “featured” apps / “popular” apps<br>▪ quick app presentation / app highlights<br>▪ problem oriented app representation (customer success stories, customer references, best practices)<br>▪ in-product / embedded app catalog: provide view on or access to app store from within core applications. Provide filtered view to app catalog with apps fitting the current user context. |
| Public App Store: Information Search | System | ▪ catalog of apps with multiple catalog categories<br>▪ discovery of apps: search, browse, filter app catalog<br>▪ detailed product presentation (textual information, pictures / screenshots, videos, documents)<br>▪ application reviews and ratings<br>▪ technical information on prerequisites, compatibility with other solutions<br>▪ embedded chat with remote sales agent<br>▪ request contact (call back or meeting request) with sales agents<br>▪ app store co-browsing / guided browsing with remote sales agent<br>▪ register for online webinars / live sessions |
| Public App Store: Evaluation | System | ▪ trial system request & provisioning (management of time-restricted trials)<br>▪ demo request & demo provisioning (demo environment: guided demo tours with user instructions)<br>▪ application & solution configuration assistants for complex applications (“scoping wizard”)<br>▪ automatic application compatibility check with customer system landscape (provide information on missing components, or dedicated updates to enable add-ons)<br>▪ service and project planning assistant (generate standardized service proposals and project plans as well as implementation guides for complex solutions)<br>▪ invite colleagues to review selected applications selections and planning material (service plans, application configurations)<br>▪ request / book on-site product workshops (booking tool) |
| Public App Store: Negotiation & Purchase | System | ▪ pricing engine (recognizing volume, bundle, or customer discount schemes, country specific taxation)<br>▪ multi-dimensional pricing configurator (ability to select different price relevant product attributes or alternative product variants, subscription versus license option)<br>▪ contract validation / recognize frame agreements (customer individual SLAs, or discounts from frame agreements)<br>▪ shopping cart capability to compile bundle of applications and services from multiple vendors<br>▪ request shopping cart approval capability (send request to authorized buyer user)<br>▪ transfer shopping cart to customer SRM / procurement system via open standard interface<br>▪ maintain customer internal IDs (cost center, purchase order IDs during checkout)<br>▪ checkout and order creation (notification to buyer user and procurement system)<br>▪ request quote for single applications or entire shopping carts with additional custom requirements<br>▪ quote inbox with review, sharing, accept / reject capability |
| Public App Store: Software Provisioning | System | ▪ generation of download links and assignment of download permissions to users<br>▪ request application / system / user activation from platform or 3rd party applications<br>▪ direct download to device (mobile, desktop)<br>▪ compile delivery notification<br>▪ update customer license and contract data<br>▪ implementation guide generator (create implementation guide with steps to be performed by the customer) |

| Resource | Type | Major capabilities / functions / job profiles |
|---|---|---|
| Public App Store: Settlement & Aftersales | System | ▪ purchase order overview<br>▪ online contract management (review agreements and terms and conditions and contract validity)<br>▪ license & usage cockpit (see available licenses, user subscriptions, used licenses versus open seats)<br>▪ add/reduce licenses or subscription seats via license cockpit / recognize already purchased products<br>▪ invoice and payment overview<br>▪ maintain & adjust payment preferences<br>▪ support for corporate credit cards |
| **Internal App Store** | | |
| Internal App Store: Setup and Administration | System | ▪ integration with corporate user identity management systems (e.g., LDAP)<br>▪ adjust visual appearance and style (incl. selection of corporate colors and logo)<br>▪ definition of catalog categories<br>▪ definition of catalog variants (sub-catalogs for dedicated user roles)<br>▪ definition of user roles and permissions |
| Internal App Store: App Configuration & Publication | System | ▪ enter new application incl. publishing content (description, screenshots, meta data)<br>▪ upload application delivery relevant content (application core, installation files, configuration, delivery, or policy files)<br>▪ maintain available licenses / user seats<br>▪ request publication<br>▪ approve / reject publication |
| Internal App Store: Information Search & Evaluation | System | ▪ catalog discovery: browse, search, filter catalog<br>▪ app presentation: view app details, view app ratings<br>▪ view app multi-media content<br>▪ device compatibility information and system pre-requisites |
| Internal App Store: Distribution | System | ▪ mobile or desktop app: Instant delivery to device (mobile or desktop)<br>▪ complex application: generation of download package and assignment of download permissions<br>▪ cloud / web app: user role activation (request user activation from 3rd party system)<br>▪ generation of download / installation e-mail notification<br>▪ trigger delivery with MDM system or 3rd party delivery system<br>▪ assign subscription seat or license to user |
| Internal App Store: Maintenance & Monitoring | System | ▪ allow users to provide app ratings and reviews<br>▪ view app download and usage statistics<br>▪ version and update management: trigger automatic updates or update notifications to app users<br>▪ license and subscription overview and status<br>▪ notify administrator when user licenses are low for dedicated application<br>▪ retirement management / app replacement rules |

# Appendix H. Chapter 5: App'ification Measures

Table 36 lists the proposed measures for implementing the respective app'ification pattern, i.e., the criteria coping with the solution-related adoption barriers of the app store for EAS.

**Table 36:** Measures to implement on app'ification patterns

| Pattern | Measures | Product Facet |
|---|---|---|
| Granular scope | ▪ user centric design: application for a single user role<br>▪ task-oriented design: support only single or few business tasks<br>▪ partial use: allow applications to be used partially (e.g., single task or user role)<br>▪ dis-aggregation / "incrementing": focus application on its core use case and encapsulate further functionality in application enhancements | Functional |
| Trialability | ▪ establish trial cascade: different levels of application "trials"<br>▪ screenshots: provides first impression of key functionalities highlighted using the user interface<br>▪ demo video: provides overview of key use case(s) guiding through the application with contextual information provided<br>▪ guided tours: user can interactively experience the application in a limited scope, major use cases can be introduced step by step using on-screen guides or UI hints<br>▪ shared trial: pre-configured system which is used by many users from different companies, only for trial purposes, data may be visible by other users<br>▪ private trial: standardized and pre-configured system which is solely provided to one user, data is not shared, may be used to deeply evaluate single aspects, very limited productive use possible<br>▪ customized trial: trial system which is configured according to customer specification, customer data may be uploaded to assess critical customer scenarios<br>▪ free edition: application can be used productively with certain limitations (functionality, period of time, number of users) | Commercial / Functional / Technical |
| Starter package | ▪ standardized contract<br>▪ bundles: include everything customer needs to start (e.g., services, application functionality)<br>▪ onion-based pricing and bundling: product elements include, service elements + functionality, different packages are modeled along the adoption path (e.g., from individual use via departmental to corporate-wide use)<br>▪ subscription-based pricing<br>▪ user- or consumption-based pricing with low minimum order quantity | Commercial / (Functional) |
| Minimum infrastructure | ▪ cloud-based version<br>▪ options for private or public cloud<br>▪ bundle with IaaS offering or provide hosted/virtual environment<br>▪ provide cloud-based integration services or hub with pre-configured integration scenarios<br>▪ ship as appliance (hardware, system software, remote maintenance) | Technical |
| Instant use | ▪ automation of implementation, customization, and configuration tasks<br>▪ pre-configured systems or configuration templates for typical use case (no customization)<br>▪ business level customization or configuration guides<br>▪ data integration & migration: enable/allow manual data integration or upload ("decoupled" scenario), provide data migration assistant (data cleansing, data mapping, data completion)<br>▪ remote implementation / configuration support | Technical / Functional |

The product facet refers to which aspect of the product is affected by the measures. This can be used to identify the responsible job roles to implement such measures. For example, measures to realize a granular scope affect the *functional* product facet. Hence, this aspect should already be considered during the product design phases from product designers, product architects, or product managers. While the commercial aspects should instead be worked on by those responsible for business development, and the technical aspects by software architects or development experts.

# Appendix I. Chapter 5: Expert Feedback Questionnaire

The following questionnaire was discussed with the experts as part of the expert review (cf. Chapter 5.6).

**Table 37:** Questionnaire for expert review

| # | Criteria to be assessed | **Rating 1-5** (5=fully agree; 1=fully disagree) | **Qualitative feedback** (How to improve? What is missing? Where is additional research needed?) |
|---|---|---|---|
| | The design, tactical and operational recommendations ... | | |
| **Function of the app store model as business model** | | | |
| 1 | ... help to establish an app store model in the enterprise application software (EAS) segment. | | |
| 2 | ... help to assess and improve an existing app store model in the EAS segment. | | |
| 3 | ... help to substantiate strategic concepts for an app store model in the EAS segment. | | |
| 4 | ... are a good starting point and help to derive more concrete business process models, service definitions, and functional blueprints. | | |
| 5 | ... address the presented value propositions for the different stakeholders adequately. | | |
| 6 | ... are concrete enough to help practitioners to derive meaningful action. | | |
| **Reach of model** | | | |
| 7 | ... Are abstract enough to serve different companies within the boundaries of the defined EAS segment. | | |
| 8 | ... are of high value for companies acting as platform provider. | | |
| 9 | ... are of high value for companies acting as ISV (independent software vendor) | | |
| 10 | ... are of high value for companies acting as software customer. | | |
| **Overall impression** | | | |
| 11 | ... overall are of high value. | | |
| 12 | ... comprehensive in the defined domain. | | |
| 13 | ... are well structured. | | |
| 14 | ... are easy to comprehend. | | |

# PART II: SUPPLY-SIDE VALUE CHAIN

## II.1 Paper 1: Platform-as-a-Service (PaaS)

# PLATFORM-AS-A-SERVICE (PAAS)

**Daniel Beimborn**

University of Bamberg

**Thomas Miletzki**

University of Bamberg

**Stefan Wenzel**

SAP AG

University of Bamberg

Business & Information Systems Engineering (2011), 3(6), pp.381–384.

DOI: 10.1007/s12599-011-0183-3

Available at: https://aisel.aisnet.org/bise/vol3/iss6/6

## II.2 Paper 2: Software Ecosystem Orchestration: The Perspective of Complementors

# SOFTWARE ECOSYSTEM ORCHESTRATION: THE PERSPECTIVE OF COMPLEMENTORS

**Theresa Rickmann**
University of Bamberg

**Stefan Wenzel**
SAP AG
University of Bamberg

**Kai Fischbach**
University of Bamberg

Proceedings of the 20th Americas Conference on Information Systems (AMCIS2014), Savannah, 2014.

Available at: https://aisel.aisnet.org/amcis2014/e-Business/GeneralPresentations/7

# PART III: DEMAND-SIDE VALUE CHAIN

## III.1 Paper 3: Evaluating the App-Store Model for Enterprise Application Software and Related Services

# EVALUATING THE APP-STORE MODEL FOR ENTERPRISE APPLICATION SOFTWARE AND RELATED SERVICES

**Stefan Wenzel**
SAP AG
University of Bamberg

**Francesco Novelli**
SAP AG, SAP Research
TU Darmstadt

**Christoph Burkard**
SAP AG, SAP Research
TU Darmstadt

Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI2013), Leipzig, 2013.

Available at: https://aisel.aisnet.org/wi2013/88/

## III.2 Paper 4: Adoption of an Online Sales Channel and “Appification” in the Enterprise Application Software Market: A Qualitative Study

# ADOPTION OF AN ONLINE SALES CHANNEL AND “APPIFICATION” IN THE ENTERPRISE APPLICATION SOFTWARE MARKET: A QUALITATIVE STUDY

**Novelli, Francesco**[1]
SAP AG, SAP Research
TU Darmstadt

**Stefan Wenzel**[1]
SAP AG
University of Bamberg

Proceedings of the 21st European Conference on Information Systems (ECIS2013), Utrecht 2013.

Available at: https://aisel.aisnet.org/ecis2013_cr/200

[1] Authors contributed equally to the publication and are listed alphabetically.

## III.3 Paper 5: Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity?

# ONLINE AND OFFLINE SALES CHANNELS FOR ENTERPRISE SOFTWARE: CANNIBALIZATION OR COMPLEMENTARITY?

**Francesco Novelli**[1]
TU Darmstadt
SAP AG, SAP Research

**Stefan Wenzel**[1]
SAP AG
University of Bamberg

Proceedings of the 34th International Conference on Information Systems (ICIS2013), Milan 2013.

Available at: https://aisel.aisnet.org/icis2013/proceedings/EBusiness/12/

[1] Authors have contributed equally to the publication and are listed alphabetically.

# PART IV: APP STORE FOR ENTERPRISE APPLICATION SOFTWARE AS AN APPLICATION SYSTEM

## IV.1 Paper 6: Comparing Public and Internal Enterprise App Stores: A Qualitative Case Study

# COMPARING PUBLIC AND INTERNAL ENTERPRISE APP STORES: A QUALITATIVE CASE STUDY

**Stefan Wenzel**
SAP AG
University of Bamberg

*Unpublished, revised and extended version.*

Previous conference version was awarded the "ICSOB 2014 Best Paper Award" and was published as "App Store Models for Enterprise Software: A Comparative Case Study of Public versus Internal Enterprise App Stores" in:

Proceedings of the 5th International Conference on Software Business (ICSOB2014) and part of the Lecture Notes in Business Information Processing book series (LNBIP, volume 182).

## Abstract

Mobile app stores have changed the way consumers discover and buy private software. Employees and end users in companies expect a similar experience and flexibility from their corporate IT. Enterprise software vendors (ESVs) therefore create new modular applications and establish their own versions of app stores for companies. Two models have appeared on the market: the public and the internal enterprise app store (EAS). Public EASs are managed and operated by large ESVs serving them both as sales and distribution channels for their software and the software built by their ecosystem. Internal EASs are managed and operated by corporate IT departments to distribute applications to company-internal users. We conduct a comparative case study of one public and one internal EAS and derive recommendations for corporate use to better meet the expectations of today's business stakeholders.



### Keywords

app store, enterprise application software, business applications, app'ification, enterprise app store, IT governance, IT consumerization

# 1 Introduction

Enterprise application software, such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM), is traditionally sold via a highly consultative, personnel-intensive process (Wenzel et al. 2013). A customer's software acquisition process is usually governed by a central IT department (Meyer et al. 2003). Buying cycles of several months up to a number of years are still widely common (Liao et al. 2007; Halingten & Verville 2002) and significant resources are tied up on the sales and buying sides. This is a costly process for both the enterprise software vendor (ESV) and the customer.

Moreover, from an innovation perspective ESVs struggle with early market adoption of newly introduced software products. It is widely accepted in the innovation management literature that a new product or invention is only classified as an innovation if it is adopted by a customer (Edison et al. 2013). The innovativeness of a software company should therefore not only be measured in terms of "time-to-market", but also in terms of "time from availability to adoption". This laborious go-to-market model also has consequences for the software buying company's internal innovation process: business units often cannot justify the business case for single requirements and IT departments are overextended by consolidating the different needs in the organization or are tied up with operating complex IT landscapes, resulting in innovation bottlenecks (Lamendola 2001; Gunasekaran et al. 2001).

The relationship between business users and IT departments is further complicated by a trend referred to as "consumerization of IT" (Weiß & Leimeister 2012). Consumer technologies such as smartphones and app stores are pervasive in many people's lives. Hence, business users are nowadays much more knowledgeable and sensitive towards technology in general, but also towards corporate IT and information systems (IS) (Niehaves et al. 2013). Business users ask for IT solutions, with consumer-grade usability, supporting ad-hoc use cases and request a stronger involvement in the software selection process or want to directly select the software they use themselves. Since IT departments, and available enterprise software and the related go-to-market process cannot comply with these requirements, the role of the CIO or the IT department comes into question (Carr 2003; Vizard 2012). Another consequence is the rise of "shadow IT" (Jones et al. 2004; Beimborn & Palitza 2013): business users circumvent corporate IT rules and use their private devices and applications without permission in their day-to-day work.

ESVs seem to have recognized the described multifaceted dilemma: they are building new, modular ("app-like") applications with consumer-oriented user interfaces (SAP 2013e) and pursue new go-to-market and software distribution models by introducing their own version of app stores[1] for companies, trying to reproduce the success of app stores in the consumer market. These B2B online sales channels not only shift the software acquisition and distribution process from the “offline” to the “online” world, but also promise a change in the enterprise software adoption paradigm: they favor a business-driven bottom-up approach over the traditional IT-driven top-down approach (Niehaves et al. 2013). These EASs are referred to as “public EASs” and are usually managed by a software provider.

With mobile apps entering the enterprise and the need for mobile application management (MAM), another form of EAS has emerged: the internal EAS[2]. In contrast to the public EAS, the internal EAS is managed by the individual software customer company.

By introducing EASs, researchers expect to effectively counter the problems arising with IT consumerization and shadow IT by satisfying the needs of business users, while gaining back control of the IT used in the company (Beimborn & Palitza 2013). Software vendors hope to benefit from the app store model with a reduction in cost of sales, increased reach to business users and an acceleration of adoption rates of new products (Wenzel et al. 2012). However, software customers seem to be reluctant, and adoption rates of EASs are still low (Böckle 2013; Novelli & Wenzel 2013a). One reason might be the uncertainty of how to best use these new models in the corporate context, i.e., how to integrate enterprise app stores into “sourcing, delivery and support” of corporate IT services (Meyer et al. 2003).

Therefore, the research objective of this study is to investigate the two prevailing models of EASs: the public and the internal EAS. The individual use cases are evaluated from a software customer’s perspective, the differences of each model are highlighted and the respective consequences are discussed. Furthermore, we propose combining the two models to leverage the advantages of both EASs. Consequently, our research questions can be formulated as follows:

1) Which scenarios are supported by public EASs and which by internal EASs?
2) How can EASs leverage the opportunities of IT consumerization while avoiding its drawbacks, such as shadow IT?
3) How can both public and internal EASs be used in combination?

To answer the research questions, an explorative, qualitative research strategy has been chosen, using an idiographic and comparative case study design (Bryman & Bell 2011). SAP

[1] Examples are: SAP Store (SAP 2014), Salesforce.com AppExchange (Salesforce.com 2014), Microsoft Pinpoint (Microsoft 2013), Amazon AWS Marketplace (Amazon.com 2013), Google Apps Marketplace (Google 2013).
[2] Examples are: SAP Enterprise Store (SAP 2013c), Symantec App Center (Symantec 2013), App47 (App47.com 2013), Salesforce.com Private AppExchange (Salesforce.com 2013), OpenPeak Openshop (OpenPeak 2013)

serves as the organizational case study context, since it provides a public EAS, the SAP Store (SAP 2014), and an internal EAS, the SAP Enterprise Store (SAP 2013c). This article targets IS researchers interested in the under-investigated topic of EASs and their implications towards corporate IT processes or IT Governance, as well as companies evaluating the use of EASs or those looking for new ways to provide corporate IT to business units and users.

The article is structured as follows; after a comprehensive literature review on related fields of research (Chapter 2), the research methodology is presented (Chapter 3). Chapter 4 presents the two cases; the public SAP Store and the internal SAP Enterprise Store in detail. Thereafter the individual findings are compared and discussed in the context of IT governance and IT management (Chapter 5). Based on the outcome of the case study, a framework is derived concerning how to combine and integrate the two models (Chapter 6). The work concludes with a discussion on limitations of the study, future research, and a summary of findings (Chapter 7).

## 2 Related Work

In this section contributions from multiple streams of research are presented which help to explain the novel socio-technological context of EASs or to assess it. The works are at the crossroads of IS research and Industrial Marketing.

**IT Consumerization**

As described previously, IT consumerization refers to the trend that IT innovations are adopted first by consumers and subsequently diffused into enterprise segments (Niehaves et al. 2013). The widespread use of consumer technologies lets business users rate corporate IT and IS with the eyes of a consumer: simple and visually appealing user interfaces, instant or ad-hoc use, and self-determined selection of software are exemplary expectations of business users towards corporate IT. These expectations are often not addressed with today's corporate IS and governance models (Niehaves et al. 2013). Therefore the diffusion of consumer technology into companies is often driven by individual employees and not controlled or permitted by the IT department ("infiltration") (Weiß & Leimeister 2012). To provide evidence of this phenomenon, Harris et al. (2012) present a survey of 4017 employees: 52% responded that they would at least sometimes use their personal consumer devices for work-related activities, 36% stated that they would not worry about the IT policies in place and just use the technology they need to perform their work, and 45% agreed with the statement that private devices and software applications are more useful than the ones provided by corporate IT (Harris et al. 2012). Furthermore, Harris et al. (2012) identify three major benefits of IT consumerization for companies: increased innovativeness, productivity, and employee satisfaction. However, if IT consumerization is not managed actively in the company it leads to shadow IT (Beimborn & Palitza 2013) and risks typical IT targets such as data security, reliability, and integrity (Harris et al. 2012). Therefore, it is proposed to actively manage IT consumerization or to introduce new governance models such as “Bring your own device” (BYOD, (Weiß & Leimeister 2012)). Beimborn and Palitza (2013) mention EASs as a promising option to manage consumerization and counteract shadow IT (Beimborn & Palitza 2013).

**IT Governance and IT Management**

Meyer et al. (2003) define IT governance as “a set of principles, practices, and measures to ensure corporate targets are met with the IT used, while resources are used responsibly and risks are adequately monitored” (translated from original German version, (Meyer et al. 2003)). From a process perspective, they mention sourcing, delivery, support, monitoring, and control as key activities of IT governance (Meyer et al. 2003).

COBIT is a best-practice framework for the governance of enterprise IT. It is now available in its fifth edition (ISACA 2012). Heas et al., who have analyzed COBIT 5 in great detail, emphasize the importance of risk and value management for Enterprise Governance of IT (De Haes et al. 2013). COBIT differentiates between the governance and management of IT: IT governance processes are defined to evaluate stakeholder needs, to set directions by prioritizing and making decisions, and to monitor performance, compliance and progress. Based on these governance processes, COBIT defines processes for the management of IT. These are structured in three groups: a) “align, plan, organize”, b) “build, acquire, implement”, and c) “deliver, service, support” (De Haes et al. 2013; ISACA 2012).

Weill (2004) defines IT governance “[…] as specifying the framework for decision rights and accountabilities to encourage desirable behavior in the use of IT” (Weill 2004). He further examines the different fields of IT decision needs and presents different IT governance archetypes. The five most important IT decision needs are in the areas of IT principles, IT architecture, IT infrastructure strategies, business application needs, and IT investment. The IT governance archetypes are defined by “who makes each type of decision, who has input to a decision, and how these people are held accountable for their role”. According to Weill, the different decision roles can be assigned to C-level executives, corporate IT, and business units or process owners (Weill 2004).

EASs overlap with the competencies defined by the above definitions of IT governance / IT Management (COBIT), hence it will be important to discuss the consequences of EAS use in the governance and management of IT: Do EASs support the management of risks and value of IT (cf. COBIT)? How can EASs be used to implement different role models between business and IT decision makers? We will therefore analyze the impact of EASs on IT governance focusing on aspects of sourcing, delivery, support, monitoring, and control as well the respective involvement of IT and business personnel in each of these processes.

### Organizational Buying Behavior and Enterprise Software Acquisitions

From an Industrial Marketing point of view, software procurement is an instance of organizational buying and an EAS can be defined as "a set of organizational and technological means constituting a centralized infrastructure serving a (individual or organizational) software consumer throughout the buying process" (Novelli & Wenzel 2013b). Robinson et al. (1967) developed a framework to identify organizational buying situations and introduced three "buying classes": new task, modified rebuy, and straight rebuy (Robinson et al. 1967). According to Webster and Wind (1972), the buying process is carried out by a buying center – the set of all the individuals from the buying organization taking on a role in the decision process (typical roles are: influencer, decider, user) (Webster & Wind 1972) and involve different organizational units, such as the IT department, business units, the purchasing department, or workers council. The vendor in turn compiles a "selling center" (Puri 1992) to approach the different interests of the customer stakeholders.

Based on the early works in organizational buying behavior, researchers have investigated factors of influence in the organizational buying process: for example, Sheth (1973, 1996) distinguished individual, environmental, and group-organizational aspects (Sheth 1973; Sheth 1996). Few authors have researched organizational buying in the context of software purchases. Based on Webster and Wind (1972), Halingten and Verville (2002; 2003) studied the purchase of ERP systems and classified influencing factors grouped into environmental, organizational, interpersonal, and individual factors. In addition, they analyzed the acquisition process and defined the phases planning, information search, selection, evaluation, choice and negotiation (Halingten & Verville 2002; Verville & Halingten 2003).

### Technology Adoption and Acceptance

An enterprise app store itself can be seen as an IT innovation. Hence, technology adoption and acceptance is an important field of research to investigate the use of EASs by companies and end users. The Technology–Organization–Environment framework (TOE) states that innovation adoption decisions by organizations are influenced by the technological, organizational, and environmental context (DePietro et al. 1990). The theory most widely used in IS research to study adoption decisions by individuals is the Technology Acceptance Model (TAM) (Davis 1989). Its main proposal is that two independent variables influence the individual's intention to use a specific technology: perceived usefulness and perceived ease of use.

### Software Platforms and Ecosystems

Multiple publications in software platform and ecosystem research mention that an "online marketplace" is at the heart of software platform offerings (e.g., Platform-as-a-Service, PaaS) (Beimborn et al. 2011; Giessmann & Stanoevska 2012; Scholten 2011). PaaS as an independent market offering constitutes hardware, software, and service components in order to enable independent software vendors (ISVs) to develop and provide software solutions to customers (Beimborn et al. 2011). The marketplace or the “platform store” is mainly evaluated in its role of an intermediary (i.e., cybermediary) to market the software products developed by ISVs on top of the software platform to customers (Scholten 2011). Giessmann et al. define such a marketplace as follows: “The PaaS provider maintains a marketplace where customers can buy software components. The marketplace can offer provisions for software requests […]” (Giessmann & Stanoevska 2012). The PaaS provider therefore offers both the software platform to develop software components and the online marketplace to market and distribute them. Moreover, the PaaS provider often offers its own software components via the marketplace (Beimborn et al. 2011).

The notion of a PaaS marketplace is largely equivalent to that of a public EAS. However, it is not necessarily required for the public EAS provider to also offer a software development platform to ISVs.

**Enterprise App Stores**

The EAS model as such has only been the focus of a few scholars so far, though business and technology analysts (e.g., Gartner, IDC) regularly rank it among the top strategic IT trends (Pettey 2012; Pettey 2011; Drake 2012).

Novelli and Wenzel (2013) have qualitatively researched the app store model for enterprise software (i.e., public EAS) and identified drivers and barriers with regards to the organizational adoption of an EAS for different types of enterprise software (core solutions, on-top solutions, usage enhancements, IT services) (Novelli & Wenzel 2013a). Moreover, they have coined the term “app’ification” in the context of enterprise software referring to “app’ified” software if an application is suited for online sales and distribution due to its characteristics such as focused scope, trial availability, starter package, or instant deployment (cf. (Wenzel 2014)). Furthermore, they provided recommendations on how to best integrate EASs with traditional, “offline”, direct sales channels (Novelli & Wenzel 2013b).

Beimborn and Palitza (2013) have investigated the benefits of internal EASs (Beimborn & Palitza 2013): they define an internal EAS as a software system for providing the functions of MAM. MAM complements Mobile Device Management (MDM) in corporations by managing the lifecycle of mobile apps, including development, procurement, distribution, configuration, update, and removal. They have identified the benefits of using an internal EAS in the areas of IT compliance, app lifecycle management, and total cost of ownership (TCO). Internal EASs as described in their study originate from the mobile app segment; however, they are not restricted to this domain and can be applied to other types of software applications as well.

Neither of the works elicited the different use cases and capabilities of public and internal EASs in detail, nor did they contrast the two models.

# 3 Methodology and Research Process

In this study we have opted for an explorative, qualitative research strategy and have chosen an ideographic and comparative case study design (Bryman & Bell 2011; Darke et al. 1998). Exploratory research is preferred in novel contexts where only limited empirical data is available or the structures and themes of the studied phenomenon are still unclear. A case study design is used to investigate the unique characteristics of one or multiple cases (i.e., ideographic) in contrast to cross-sectional designs, where the focus lies on nomothetic, generic findings. Comparative designs embody the logic of comparison: they intend to better explain a phenomenon when two or more cases are selected and contrasted or put in relation (Bryman & Bell 2011). EASs are a novel socio-technological context and come in two distinct models. By intensively studying internal and public EASs, we try to reveal the most important features, processes, and consequences for software customers. Further, the study design allows us to compare the two models and to derive additional knowledge which would not have been possible in two unrelated studies of the two EAS models. As an organizational case study context, SAP has been chosen. SAP fulfilled multiple criteria at the same time: SAP is one of the largest vendors of enterprise software; it operates a public EAS, the SAP Store (SAP 2014), and offers an internal EAS (SAP 2013c), the SAP Enterprise Store. Moreover SAP provides software development platforms and related PaaS offerings to enable ISVs to develop and market their own applications (SAP 2013b); Last, SAP has a comprehensive MDM and MAM portfolio to manage enterprise mobility within companies (SAP 2013a). Figure 1 illustrates the sequential research process of this study.

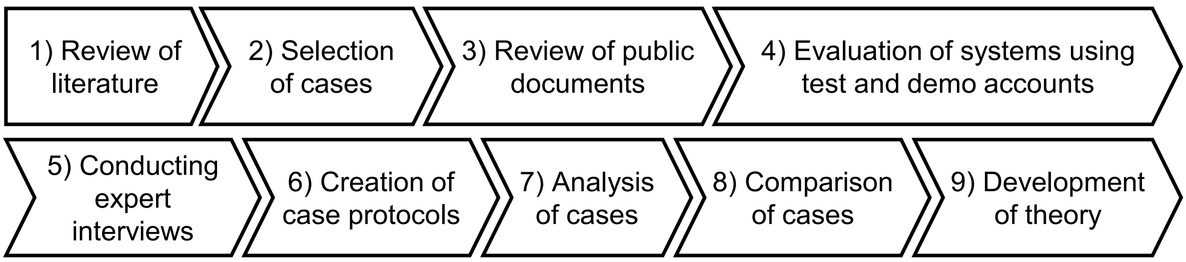


**Figure 1:** Simplified, sequential research process

After a literature review and case selection, publicly available material was reviewed (online documentation: the SAP Store Support Guide (SAP 2013d) and the SAP Enterprise Store help documentation (SAP 2013c)). Subsequently, we were provided with a test account for the SAP Store (SAP 2014) and a demo system of the SAP Enterprise Store to analyze the functionality of the respective systems. Last, two experts from the product management of the SAP Store and the SAP Enterprise Store were interviewed. During the interviews we mainly tried to confirm assumptions and discuss open questions which could not be fully answered by the public material and or test systems. The interviews lasted 45 minutes each and notes were taken during the interviews. All findings (public material, test systems, interviews) were recorded and integrated in case study protocols. They formed the basis for the comparison of the two cases and were used to derive implications.

The case studies of public and internal EAS are divided in three categories: First, the structure of their respective business systems is analyzed to identify all relevant stakeholders and to uncover their roles and relationships. The business system is modeled and analyzed using the Semantic Object Model (SOM), i.e., the so-called interaction schema (for detailed description of the SOM modeling framework refer to (Ferstl & Sinz 1997; Ferstl & Sinz 2006)). The interaction schema is used to structurally analyze the business objects and business processes of a business system. Business processes again are analyzed with regard to their service delivery (the term service includes services, products and payments) and with regards to the coordination of business objects. The SOM framework distinguishes three main coordination principles: the non-hierarchical “negotiation principle”, the hierarchical “feedback control” principle, and the hierarchical “coordination via targets” (Ferstl & Sinz 1997; Ferstl & Sinz 2006). Second, the catalogs are reviewed, i.e., which enterprise applications are supported by the respective EAS. Third, a detailed functional analysis is conducted. Although, we tried to keep the categories in the functional analysis equal, they partially differ. This is due to the fact, that the internal and public EAS essentially are two different models and support partially different processes.

To compare public and internal EAS, we use the following IT Governance and Management processes (cf. section 2 and  (Meyer et al. 2003)): IT sourcing, delivery, support, monitoring and control. The special focus of this comparison is how public and internal EAS help to involve business stakeholders in the respective processes while ensuring IT control (cf. section 2 and (Weill 2004)).

As a final step we propose the combined use of internal and public EAS and derive requirements and pre-requisites for an integrated scenario.

# 4 Presentation of Cases

In this section the two cases, SAP Store and SAP Enterprise Store, are presented (cf. Figure 2). First, the business system is modeled and analyzed. Second, the supported applications (Catalog Management) are analyzed, and third, a functional analysis is performed.

**Figure 2:** Screenshot SAP Store (left) and SAP Enterprise Store (right, demo system)

## 4.1 Case: SAP Store

**Business System Analysis.** Figure 3 shows the business system model of the SAP Store using a SOM interaction schema (cf. section 3). The three main roles in the model are represented as business objects: the ISV, the platform provider (SAP) and the software customer. The platform provider object is further decomposed into the public EAS (the SAP Store), the software development platform, the internal software development unit, and a portfolio management unit. For better readability the interaction schema focuses on service delivery transactions (labeled with “e:” for enforcing transaction, (Ferstl & Sinz 1997)) and leaves out the coordination transactions with SAP external business objects. The software development platform provides ISVs the SAP Application Developer Partner Program (SAP 2013b). Essentially this program can be seen as a PaaS offering: as part of this program, ISVs have access to all relevant enabling and technological services in order to develop software applications using SAP’s technological platforms and to provide their applications on the SAP Store using it as a global, online sales and distribution channel. SAP as platform provider is operating the SAP Store and fulfills the tasks of an intermediary and broker: it provides the app store infrastructure, maintains the public store catalog, certifies and assures the quality of the applications, and provisions the software to the software customer. SAP collects a revenue share from the ISVs and thus participates in each sales transaction. Furthermore, SAP’s own development unit provides software applications via the SAP Store and uses it as an online sales channel, too. The software customer uses the SAP Store as an online, self-service environment to conduct software acquisitions. As a result, the SAP Store provides the software application to the customer. The SAP internal units SAP Store, software development and software development platform are aligned by a common portfolio management. The portfolio management controls the autonomous units software development, software development platform and public EAS via target settings (transactions labelled with “T”) ((Ferstl & Sinz 2006, p.204 ff.)).

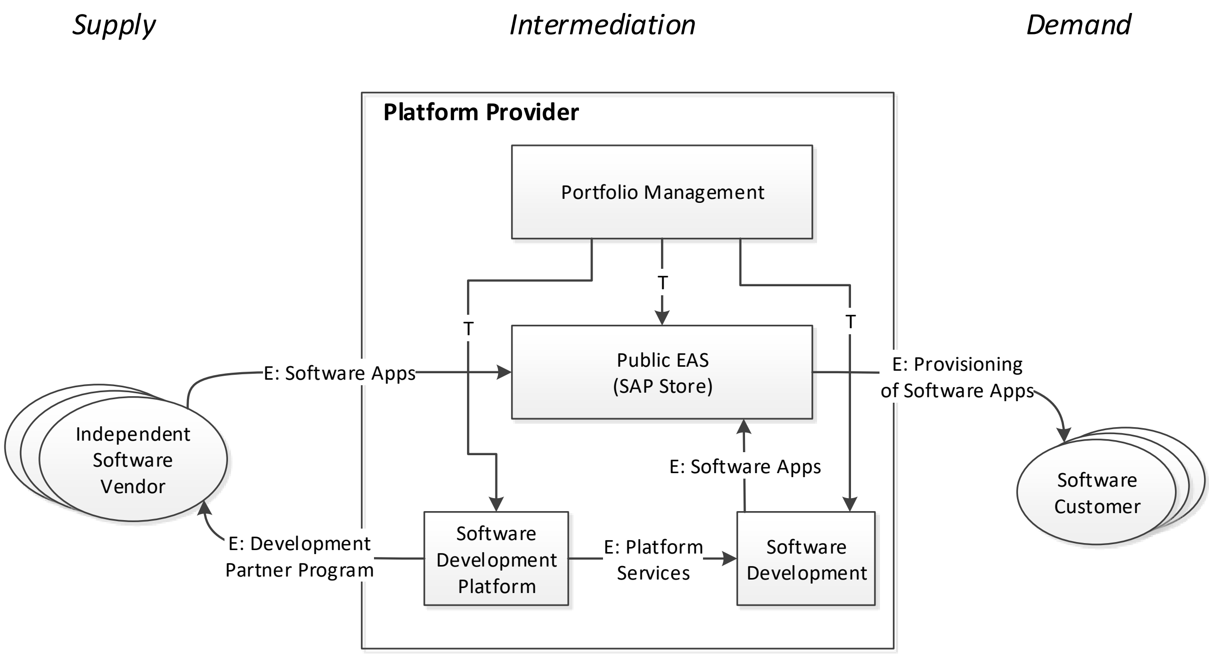


**Figure 3:** Business system model of the SAP Store 1 (SOM interaction schema)

The interaction between the SAP Store and the software customer has been further analyzed and modelled separately in Figure 4. For this reason, the transaction "E: Provisioning of Software Apps" and the software customer object in Figure 3 have been further decomposed. In contrast to the model in Figure 3, the SAP Store has been modeled as an environmental object (represented as oval) and the software customer as an object belonging to the universe of discourse.

For a dedicated purchase transaction, typically two distinct user roles are involved. The SAP Store user is a registered user and the typical user role of a business user. The SAP Store initiates the transaction (labeled with "I:") by providing the app catalog to the SAP Store user including descriptions and other information content. The SAP Store user can request a trial version of a desired application (contracting transaction labeled with "C:") or directly add it to the shopping cart. Once a SAP Store user has "checked out", the respective shopping cart is sent to the SAP Store Buyer user for approval. This is a user role with purchasing authority on the SAP Store. Typically this role is assigned to a person in the IT department, purchasing department or to a dedicated person in a business unit (there can be multiple SAP Store Buyer users). After the SAP Store Buyer has approved the cart, the SAP Store triggers the provisioning of the application to the SAP Store user. The invoice is sent to the SAP Store Buyer who is also responsible for the payment process (different payment methods are supported).

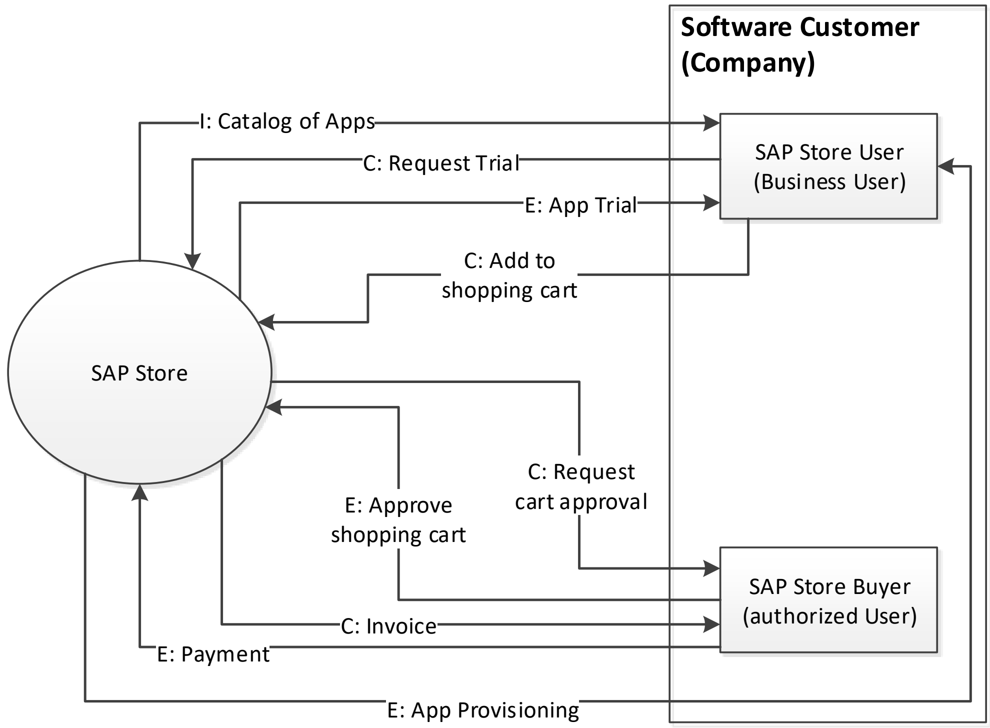


**Figure 4:** Business system model of the SAP Store 2 (SOM interaction schema)

### Catalog Management / Offering

At the time that the data for this study was collected (August to November 2013), the SAP Store offered approximately 1200 solutions. This number differs slightly depending on the country selected. About half of the solutions are provided by SAP and the other half are provided by around 400 SAP partners (ISVs). The solutions span all industries, business areas, deployment models (on-premise, cloud, mobile devices, desktop), and range from simple to highly complex applications.

### Functional Analysis

The SAP Store supports the entire buying process from information search, evaluation, negotiating and buying, and delivery (derived from Verville and Halingten; cf. (Verville & Halingten 2003; Novelli & Wenzel 2013b)). Below, the functionality will be evaluated for each process step separately complemented with non-process-related functional categories.

### Information Search

The SAP Store catalog is organized by multiple categories: industry, line of business, or categories for special topics, such as “Analytics” or “Cloud Solutions”. In addition to browsing, filtering, and searching the catalog, there is a recommender providing personalized recommendations to the user. Three main screens support the phase of identifying a solution and gathering information. The SAP Store homepage highlights selected solutions and gives personal recommendations to the user. The catalog view is a typical list view: it shows search results and provides filters. Key information such as short descriptions or price are visible in this view and the key transactions can be accessed (e.g., add to cart). The solution view provides detailed information on a solution, e.g., functional, technical, and pricing details, introduction videos, customer success stories, and screenshots.

**Evaluation**

The SAP Store provides several dedicated features to evaluate solutions in more detail. Most solutions come with a trial version or demo mode. Simple solutions (e.g., which can be installed on the desktop) or which are operated in the cloud often have trials with most functionality enabled and integrated sample data. These trials are often limited to a certain period (e.g., 30 days). Other solutions come in a free version and a paid version. The free version can be downloaded and used without time limitations and can serve for evaluation or simple use cases.

Mobile enterprise apps can be downloaded for free (only the backend components need to be purchased) and include a demo mode to try the app. System landscape requirements can be evaluated with the so-called "compatibility check". This feature analyzes the customer's system landscape and assesses prerequisites of the selected solution. The compatibility check informs the user if additional components need to be acquired and installed first. The SAP Store provides a selection of social features which may also be used to better evaluate a solution. Users can write reviews for purchased solutions and provide ratings which are visible to other customers. Further, users can recommend a solution to colleagues by using established social networks such as LinkedIn or simply e-mail.

**Negotiation and Purchase**

If the user decides to purchase a solution, he can add it to the shopping cart. The shopping cart shows detailed price information (e.g., fixed and recurring fees) and the quantity of items can be changed in this screen. As a payment option, the user can choose between invoice and credit card. The SAP Store also recognizes corporate customer or volume discounts. Once the user proceeds, the review screen is shown: in this screen the terms and conditions of the selected solutions need to be reviewed and agreed to. Before submitting the order, the user can also enter an internal "Purchase Order ID" for correct booking of the order with his company's purchasing system.

**Delivery**

The delivery of the solution depends on the deployment model. On-premise server solutions, mobile apps, or desktop solutions are delivered via direct file downloads (a download link is sent via e-mail to the buyer); cloud solutions are simply activated and the buyer receives a link with login instructions. The invoice is sent to the buyer in a separate e-mail.

### Users and Permissions

There are multiple predefined user roles in the SAP Store with different permissions. A "Guest User" is an unregistered user on the SAP Store. He can freely browse the entire catalog and can even download free applications or demos. A "SAP Store User" is a registered user associated with a company. A SAP Store User can fully browse the catalog, use social features (review solutions, recommend solutions to colleagues), and buy a few selected solutions using his credit card (mainly personal solutions). The "SAP Store Buyer" has the additional permission to purchase all solutions available on the SAP Store on behalf of his associated company (invoice or credit card). Last, the SAP Store Super User can invite users to the SAP Store and pre-register them with his company. Further, he can assign and revoke SAP Store roles to the individual users.

### Administration and Support

The "SAP Store Account" section allows users to access user management, review the details of past SAP Store orders, and view and accept quotes. A dedicated support page helps users with questions. Further, SAP provides phone numbers, an e-mail address, and a live chat if users have questions with regards to the SAP Store or a desired solution.

### Complex Buying Scenarios

The SAP Store is a self-service buying and e-commerce platform. However, there are dedicated features to use the SAP Store in combination with a traditional offline, direct sales channel, e.g., a SAP sales representative. Most solutions on the SAP Store also provide a "Contact Me" button to ask for the involvement of a salesperson or to provide a "request a quote" transaction. A salesperson then creates this quote offline and loads it back up to the SAP Store where customer users can review and accept (or decline) the quote.

## 4.2 Case: SAP Enterprise Store

### Business System Analysis

The SAP Enterprise Store is an internal EAS. It is available for purchase as a single product or can be acquired as part of the SAP Mobile Secure suite (SAP, 2013). In contrast to the business system analysis of the public SAP Store, the analysis of the SAP Enterprise Store represents a "good practice" customer setting; individual companies may use only parts of the described processes or have adapted individual elements. The Business System is modeled in Figure 5: The SAP Enterprise Store is hosted by SAP and provided to the software customer as a cloud solution. The software customer has been decomposed into the following business objects:

The IT portfolio management administrates and manages all IT requirements and therefore interacts with business units (not modelled in Figure 5). For software acquisitions, the IT portfolio management sends a purchase request to the IT purchasing department. In the event that the required application needs to be developed in-house a development request is sent to the internal development unit. Furthermore, the IT portfolio management defines which applications can be accessed by which user group or role and defines the structure of the catalog. These instructions are addressed to the internal EAS management. The internal EAS management administrates the internal EAS. This includes managing the application behavior, the visual appearance, user and content management, activating and deactivating applications as well as creating reports usage reports. The applications and the respective digital content such as descriptions or app screenshots are provided either by the internal software development unit for applications or content developed in-house or by the IT purchasing group for acquired applications or content. The internal EAS management is responsible for the ultimate publication of an application and the correct assignment to catalog categories.

Business users (i.e., employees) can discover and evaluate applications and content, and can download, activate, or consume them. They can further provide reviews of downloaded items which is valuable feedback for the IT portfolio management or in-house developers. The SAP Enterprise Store further supports a co-innovation process in which in-house developers can provide early versions of their applications to a selected group of “beta users” to receive feedback before an official release.

Furthermore, the SAP Enterprise Store can be used to position selected content (e.g., via sub-catalogs) not only to company-internal consumers, but also to its ecosystem, such as suppliers, partners, subsidiaries, or customers. This way it can be used to expand the reach of corporate IT beyond the borders of the company (cf. Figure 5). The SAP Enterprise Store can be partially integrated with the SAP Store: the catalog of the internal EAS can be prefilled with selected solutions from the public EAS. Further, it integrates with SAP Afaria, an MDM solution. In this case the SAP Enterprise Store replaces the App Catalog as part of Afaria, which is a rudimentary version of an internal EAS.

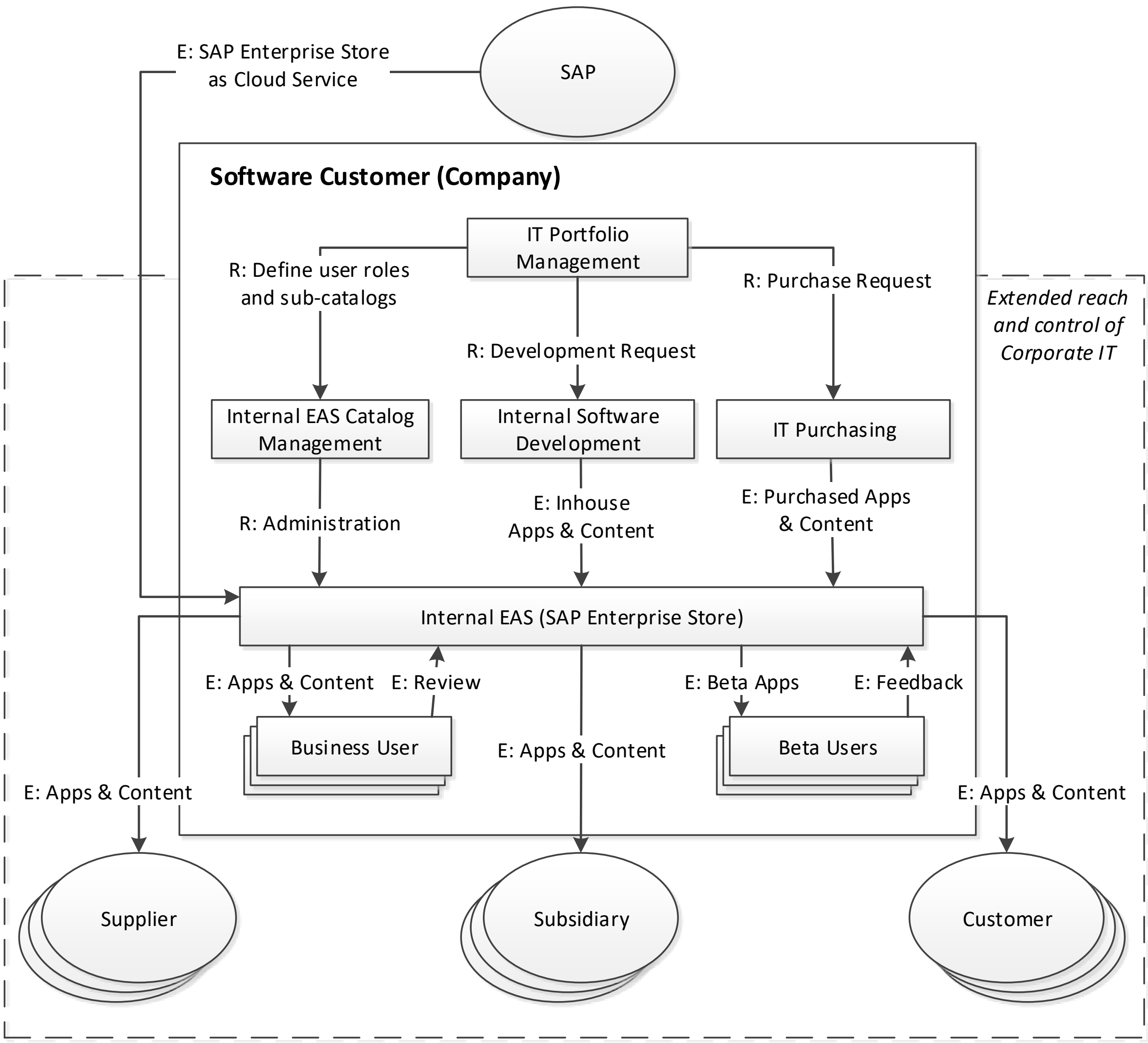


**Figure 5:** Business system model of the SAP Enterprise Store (SOM interaction schema)

## Catalog Management / Offering

The catalog is fully managed by the customer's IT department (IT portfolio management and internal EAS management) and can be used to host different kinds of digital content: mobile apps, desktop apps, web apps, or e-learnings. The applications and content provided target individuals, i.e., end users. Hence, the SAP Enterprise Store is not meant to distribute an entire server application, but it would in this case provide the user-related interface, which could be an app or even just a link and license activation. The SAP Enterprise Store can host solutions which are available in the SAP Store, applications developed in-house, or any third-party application. The categories to browse the catalog can also be freely configured by the administrator.

### Functional Analysis

The functional analysis is presented along slightly different categories than the SAP Store as the focus lies on the adoption of single applications by individual users and the management of these applications by an IT department.

#### Information Search

The SAP Enterprise Store is similarly organized to the SAP Store and has a comparable look and feel. It can be accessed via the browser or apps available for mobile devices. On the homepage the user can enter the catalog by categories (defined by the company), search for applications, or select recommendations. In the list view, the user can filter the results using multiple criteria (e.g., device, line of business). In comparison to the SAP Store the list view shows more details with regards to device compatibility (e.g., phone, tablet). The solution view displays details of the solution (screenshots, videos, functionality, and technical details). In contrast to the SAP Store, there are no dedicated pricing or commercial details but more details on device compatibility. The available user reviews can further be filtered with regards to a device-specific version. The download or deployment of an app can be triggered in the list view and the solution view.

#### Evaluation, Negotiation, and Purchase

To evaluate a solution, the user can view screenshots, videos, or ratings and reviews. Dedicated features to access trials or demos are not available. Once a user decides to use a solution he can download it or trigger the delivery. A shopping cart or dedicated buying process is not supported.

#### Delivery

The delivery depends on the deployment model of the app and on which device the SAP Enterprise Store is accessed. For mobile apps the installation of an app is directly triggered if the catalog is accessed from the mobile device itself. Where a mobile app is chosen from the desktop version of the catalog, an e-mail including the installation link is sent to the user’s e-mail address. For desktop apps or other digital content (e.g., e-learnings) a file download is triggered. For web apps login credentials and activation links are sent to the user’s e-mail address. The SAP Enterprise Store can also be integrated with SAP Afaria, the MDM solution of SAP. In this case, a selected app is handed over to the MDM system for device deployment. In the MDM scenario the SAP Enterprise Store can also be used to actively and centrally distribute selected applications to end-users. This way the “pull” and “push” distribution of applications can be combined.

**Company-Internal IT Innovation Process and Social Features**

The SAP Enterprise Store cannot only be used to distribute productive applications but also allows in-house developers to upload their own apps, even if these apps are still in development. This is supported by a dedicated process and status management: if an in-house developer uploads an application he can mark it as “In Development”, “Beta”, or “Productive”. Before the app is published, the administrator needs to review and approve it first. Eligible users can then review and try these non-productive apps and provide feedback via the social features of the SAP Enterprise Store. In general there are features to review and rate applications and this way give feedback to the IT department.

**Users and Permissions**

The SAP Enterprise Store integrates with corporate identity systems and recognizes users in the intranet. Out of the box there are only two roles: a standard user who can fully browse and download apps and the administrator role. The administrator can manage the catalog, view usage statistics, adjust the visual appearance, create catalog categories, assign users to roles, or even restrict certain catalog content to a group of users or roles. The standard roles can be enhanced with additional roles such as developers (eligible to upload apps), beta users (employees who have access to non-productive apps), or dedicated roles for suppliers or customer companies who only have access to a sub-set of applications.

**Administration**

The administration of the SAP Enterprise Store is grouped into two categories: Setup and Statistics. Setup includes general settings, visual style settings (e.g., apply corporate branding), catalog categories, app management (publish, edit, retire, approve/decline new apps), and user and role management. The Statistics view shows detailed usage information of the SAP Enterprise Store such as number of downloads or uploads and can also be used for license management.

# 5 Comparison Public versus Internal EAS

In this section we will discuss how the two EAS models presented help to resolve the initially stated dilemma: How can EASs leverage the opportunities of IT consumerization while avoiding its downsides, such as shadow IT? Or in other words,

how to involve business units and users during sourcing, delivery and support of business applications while providing mechanisms of monitoring and control to the IT organization to ensure guidelines of IT governance and IT management are followed and objectives are met. A stronger involvement of business users in the identification, evaluation and selection (sourcing) of software applications can also be described as bottom-up, decentral or heterarchical adoption, in contrast to the more traditional top-down, central and hierarchical adoption of software applications in a company. In general, it must be said that neither EAS model predefines who in the organization takes over which decision-making role in the investigated areas of the IT governance processes (cf. Weill, (Weill 2004)). Companies themselves still need to define how EASs are implemented and used in the organization, i.e., who takes over which part in decision processes. Table 1 presents the capabilities of the two EAS models with regards to business involvement and IT control in the main tasks of IT governance (cf. (Meyer et al. 2003)).

**Table 1:** Comparison of key capabilities of the public versus internal EAS model with focus on business involvement and IT control along the IT governance process (cf. (Meyer et al. 2003))

| IT Gov. Process | SAP Store (public EAS) | SAP Enterprise Store (internal EAS) |
|---|---|---|
| Sourcing | ▪ Business can identify, gather information about, and try new business applications<br>▪ IT defines buyers and proactively invites business reps to participate during external sourcing<br>▪ IT can enable selected business reps to make purchases | ▪ Early involvement of business users in in-house development projects (internal sourcing) |
| Delivery | ▪ Instant delivery of software can accelerate delivery process | ▪ Business users select and consume apps in a self-service mode using a consumer-friendly app catalog<br>▪ Provide apps to ecosystem<br>▪ Secure and instant delivery to user devices |
| Support | | ▪ Internal EAS can be used to distribute updates of applications<br>▪ Distribute e-learnings |
| Monitoring | ▪ IT can monitor all purchases on the EAS via a central order view | ▪ Monitor app usage, downloads, reviews, ratings<br>▪ License monitoring |
| Control | ▪ Define buyer roles<br>▪ Prevent business users from buying non-authorized app-lications | ▪ Define target groups for applications (who can access which apps)<br>▪ Fully define catalog content and visual styling of EAS |

In summary, the public EAS studied focuses on external sourcing, i.e., the acquisition, of new enterprise software. It can be used to tightly involve business users during the identification and evaluation of applications. Dedicated capabilities are provided to enable organizational buying (e.g., buyer roles, corporate discount, quotations, compatibility checks). Still, the features provided allow IT departments to control which applications are purchased, and buying permissions can be managed actively: The SAP Store Super user would typically be assigned to a representative of the IT department whereas the SAP Store buyer could either be one or more IT employees or even selected representative from business units. The self-service paradigm of the public EAS has the potential to increase efficiency during external sourcing activities and thus shorten buying cycles, and ultimately accelerate the introduction of new enterprise software applications. A potential shift in responsibilities during is software buying process (cf. (Novelli & Wenzel 2013a)) is illustrated in Table 2.

**Table 2:** Illustrative comparison of responsibilities (IT department and business unit representatives) of a traditional software buying process versus a buying process facilitated by Public EAS

| | | **Problem recogn.** | **Info Search** | **Evaluate / Trial** | **Purchase** | **Configure & Distribute** |
|---|---|---|---|---|---|---|
| **Traditional process** | *Driver* | Business | IT | IT | IT | IT |
| | *Support* | - | Business | Business | Business | Business |
| **Process facilitated by Public EAS** | *Driver* | Business | Business | Business | Business | IT |
| | *Support* | - | IT | IT | IT | Business |

The internal EAS model focuses on the distribution of personal applications to employees and the ecosystem. Thereby, it encourages an employee- or rather user-driven pull model in favor of a push model. IT departments can fully control the available content in the EAS catalog. The breadth of supported types of content (web, mobile and desktop apps, and other digital content) allows the internal EAS to be the single source for employees' app needs. The roles concept can further be used to target specific user groups: to pre-filter the available applications and even to position selected applications to the ecosystem. Another noteworthy capability is to actively support the internal innovation process, for example, by publishing beta apps to a selected group of users in order to receive early feedback. Last, the monitoring capabilities of the internal EAS inform the IT department about which applications are used and which are not, and help to establish a more accurate and compliant license management and also to ultimately improve the IT portfolio management by assessing user adoption rates of applications.

Whether the EAS models will be accepted by business users can be analyzed in terms of "perceived usefulness and ease of use" (cf. (Davis 1989)) . Perceived usefulness will be rated first and foremost by the applications provided by the EAS. Whereas the public EAS catalog is controlled by an external provider, the internal EAS catalog is filled by the IT department. In both cases it is important to offer a broad assortment of attractive applications. The second determinant "perceived ease of use" will be rated according to how convenient software acquisition or adoption processes can be conducted using an EAS. Both studied EASs are using a design very similar to consumer app stores or well-known e-commerce sites. Hence, it is safe to state that most users will be familiar with the interaction patterns of the researched EASs.

## 6 Integration of Public and Internal EAS

Our case study has shown that both models support the involvement of business users during software selection and adoption processes, though they focus on different parts of the overarching process. Figure 6 shows a potential combined use of public and internal EASs for a given company along the software adoption process.

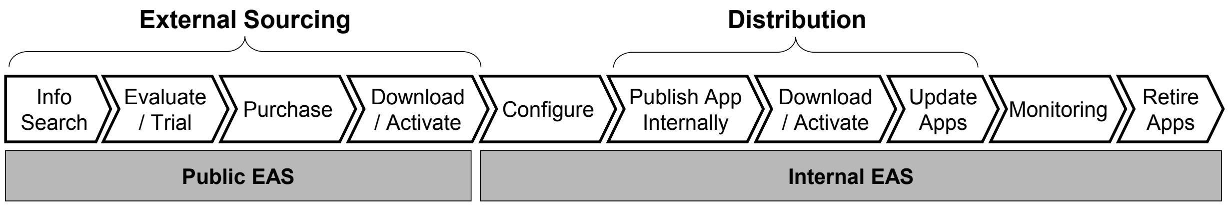


**Figure 6:** Combined use of public and internal EASs

The public EAS can be used for external sourcing of new enterprise software and for actively involving selected business representatives during the acquisition process. After configuration of the acquired software for corporate use and defining the target user groups, the internal EAS provides employees with a consumer-like experience to discover and consume (i.e., download/activate) corporate applications. Further, the internal EAS helps IT departments to manage and monitor applications along their firm-internal lifecycle.

However, the combined use of public and internal EAS as two distinct applications or services poses several problems. First, the two EAS are provided using two different domains and two distinct catalogs. Business users looking for applications would therefore need to browse the two distinct EASs separately. Furthermore, the user management is not aligned: the public and internal EAS might have different user roles with different permissions. Ultimately, this requires a more complex governance model and would lead to an inferior user experience: a single employee would require two different users and the company would need to define and apply two different terms of use. Practically this could be resolved by restricting the public EAS to few users, e.g., members of the IT portfolio management and defining the internal EAS as central application store for all employees. Obviously, this would limit the benefits of the public EAS as described earlier.

To maximize the benefits of both EAS models while avoiding the described complexity we propose an integrated scenario. The model presented in Figure 7 illustrates a possible integration of public and internal EAS using a SOM interaction schema (note, not all aspects of public or internal EAS are presented in Figure 7, as the model focuses on the main aspects relevant for the integration).

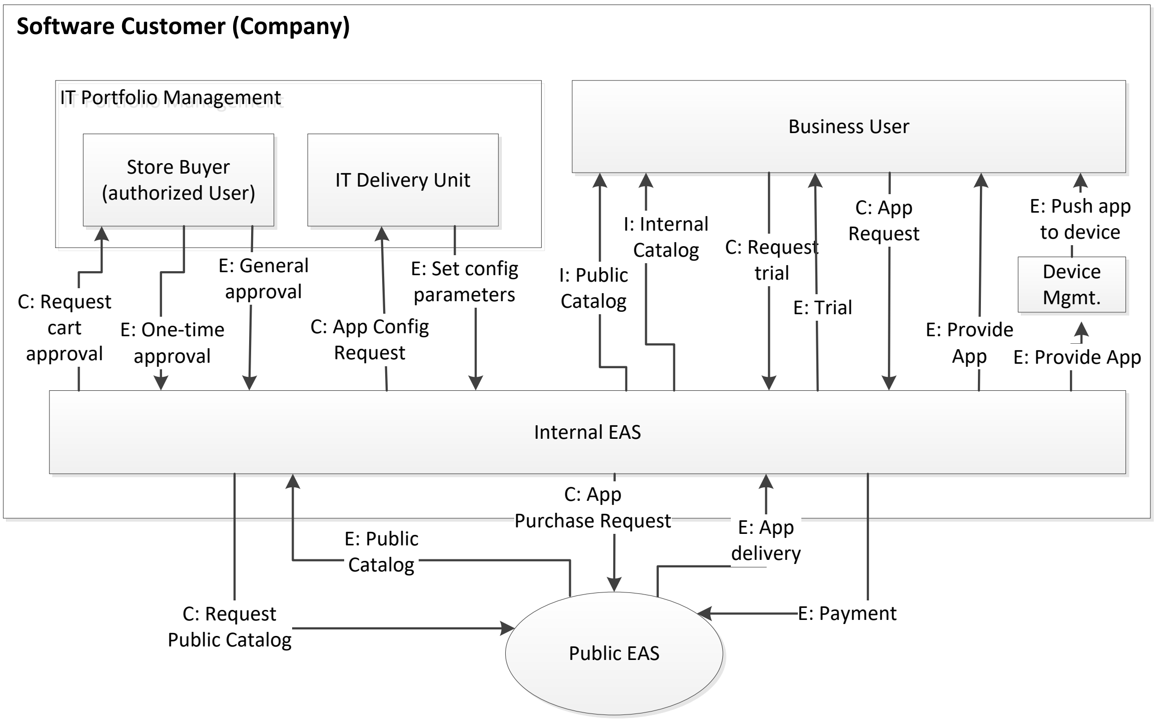


**Figure 7:** Public and internal EAS in an integrated scenario (SOM interaction schema)

The integrated scenario proposes the internal EAS as single frontend for corporate users. A business user who is registered in the internal EAS can browse both the internal catalog as well as the catalog of the public EAS via one user interface. The individual catalog entries might still be visually different so that a user can recognize whether an application is already “approved and configured” by the internal IT department and ready for direct consumption or whether it still needs approval. In order to make the public catalog accessible via the internal EAS both systems need to be integrated, e.g., via a standardized interface such as the Open Catalog Interface (OCI; (SAP n.d.)).

If a business user requests an app from the internal catalog, the application will be directly provisioned, either via download to the selected device or via a device management system. If the internal EAS is integrated with MDM solutions it can also be used to “broadcast” selected apps or content to user devices.

In case the application originates from the public catalog a workflow will be triggered to the authorized store buyer user to approve the shopping cart (Alternatively, the user management of the internal EAS could define dedicated roles that also allow the direct purchase of selected applications by business users). Typically an authorized store buyer would be a representative of the IT portfolio management or IT purchasing group. Such an approval request could be handled in two ways: either the application purchase is approved individually or the application is generally approved and thereafter automatically transferred to the internal catalog so that other users can access it without requesting approval. Before the application is provisioned to the business user many applications require a setup or configuration to work in the company landscape (e.g., integration with identity systems or setting up backend connectivity). Therefore, after purchase approval the internal EAS would post a configuration request to an IT delivery employee who conducts the configuration of the application (e.g., via a configuration console).

The communication between internal and public EAS would need to be fully automated after initial setup. Transactions such as “C: App Purchase Request”, “E: App Delivery” and “E: Payment” would be handled between the internal EAS and public EAS automatically based on the initial setup and the communication protocols. If end-user involvement is required in a transaction, the user will interact with the internal EAS.

# 7 Limitations, Future Research and Conclusion

**Limitations and Future Research**

In this study we used an explorative, qualitative research strategy and a case study design. Generalizability of qualitative research in general, but especially of case studies, is low (Bryman & Bell 2011). Furthermore, we did not involve experts from customer companies as we could not recruit adequate candidates at the time of this study (the SAP Enterprise Store was released in late Summer 2013). Future studies (qualitative or quantitative) may research EAS models in the actual customer context and analyze other EASs available on the market. Reliability, i.e., being able to reproduce the results with the methods that were chosen, has been considered by detailing the research process as much as possible and by applying systematic methods investigating the cases, such as the Semantic Object Model to analyze the respective business systems and the universe of discourse. Furthermore, objections may be raised with regards to the validity of our proposed integration of public and internal EAS since it was not systematically reviewed with experts. Therefore we plan to further evaluate and detail this integration model in a consecutive research project.

**Conclusion**

We studied public and internal EAS models by researching two cases, the SAP Store and SAP Enterprise Store. The capabilities of each EAS were analyzed in detail and scenarios for practical use were presented and discussed (cf. research question 1). Special focus was placed on how the two EAS models help to involve business users during key IT governance processes, especially the sourcing and delivery of IT, while keeping control of IT (cf. research question 2). Moreover, we presented the combined use of public and internal EASs and highlighted problems which might occur in practice. To resolve the issues of combined use and leverage the benefits of both internal and public EAS, finally we proposed an integrated scenario of internal and public EAS and presented a potential realization of such a setup by modelling the transactions which are relevant for the integration and by discussing the major pre-requisites. As a concluding remark, we believe EASs will spur the digitization of the software acquisition and distribution process and will enable faster adoptions of IT innovations and a stronger involvement of business stakeholders.

## References

Amazon.com, 2013. Amazon AWS Marketplace. Available at: https://aws.amazon.com/marketplace [Accessed November 6, 2013].

App47.com, 2013. App47. Available at: http://www.app47.com/ [Accessed December 3, 2013].

Beimborn, D., Miletzki, T. & Wenzel, S., 2011. Platform as a Service (PaaS). *Business & Information Systems Engineering*, 3(6), pp.381–384.

Beimborn, D. & Palitza, M., 2013. Enterprise App Stores for Mobile Applications - Development of a Benefits Framework. In *AMCIS 2013 Proceedings*.

Böckle, R., 2013. B2B App Stores - Anbieter im Vergleich. *Computerwoche*, 49(49), pp.14–21.

Bryman, A. & Bell, E., 2011. *Business Research Methods*, Oxford University Press, USA.

Carr, N., 2003. IT doesn't matter. *Harvard business review*.

Darke, P., Shanks, G. & Broadbent, M., 1998. Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal*, 8(4), pp.273–289.

Davis, F.D., 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), pp.319–340.

DePietro, R., Wiarda, E. & Fleischer, M., 1990. The context for change: Organization, technology and environment. In L. G. Tornatzky & M. Fleischer, eds. *The processes of technological innovation*. Lexington (Massachusetts): Lexington Books, pp. 151–175.

Drake, S.D., 2012. *Mobile Enterprise App Stores : Open for Business*, IDC.

Edison, H., bin Ali, N. & Torkar, R., 2013. Towards innovation measurement in the software industry. *Journal of Systems and Software*, 86(5), pp.1390–1407.

Ferstl, O.K. & Sinz, E.J., 2006. *Grundlagen der Wirtschaftsinformatik, Band 1* 5. Auflage.,

Ferstl, O.K. & Sinz, E.J., 1997. Modeling of Business Systems Using the Semantic Object Model (SOM) - A Methodological Framework -. In P. Bernus, K. Mertins, & G. Schmidt, eds. *Handbook on Architectures of Information Systems. International Handbooks on Information Systems.* Berlin: Springer.

Giessmann, A. & Stanoevska, K., 2012. Platform as a Service–A Conjoint Study on Consumers' Preferences. , pp.1–20.

Google, 2013. Google Apps Marketplace. Available at: https://www.google.com/enterprise/marketplace/ [Accessed November 6, 2013].

Gunasekaran, a., Love, P.E.D., Rahimi, F. & Miele, R., 2001. A model for investment justification in information technology projects. *International Journal of Information Management*, 21(5), pp.349–364.

De Haes, S., Van Grembergen, W. & Debreceny, R.S., 2013. COBIT 5 and Enterprise Governance of Information Technology: Building Blocks and Research Opportunities. *Journal of Information Systems*, 27(1), pp.307–324.

Halingten, A. & Verville, J.C., 2002. A qualitative study of the influencing factors on the decision process for acquiring ERP software. *Qualitative Market Research: An International Journal*, 5(3), pp.188–198.

Harris, J., Ives, B. & Junglas, I., 2012. IT consumerization: when gadgets turn into enterprise IT tools. *MIS Quarterly Executive*, 11(3), pp.99–112.

ISACA, 2012. COBIT 5: A Business Framework for the Governance and Management of Enterprise IT.

Jones, D., Behrens, S., Jamieson, K. & Tansley, E., 2004. The Rise and Fall of a Shadow System: Lessons for Enterprise System Implementation. In *ACIS 2004 Proceedings*.

Lamendola, M., 2001. Justifying Software Purchases. *Electrical Construction and Maintenance*. Available at: http://ecmweb.com/content/justifying-software-purchases [Accessed August 9, 2013].

Liao, X., Li, Y. & Lu, B., 2007. A model for selecting an ERP system based on linguistic information processing. *Information Systems*, 32(7), pp.1005–1017.

Meyer, M., Zarnekow, R. & Kolbe, L.M., 2003. IT-Governance - Begriff, Status quo und Bedeutung. *Business & Information Systems Engineering*, 45(4), pp.445–448.

Microsoft, 2013. Microsoft Pinpoint. Available at: http://pinpoint.microsoft.com/ [Accessed November 6, 2013].

Niehaves, B., Köffer, S. & Ortbach, K., 2013. The Effect of Private IT Use on Work Performance-Towards an IT Consumerization Theory. *Wirtschaftinformatik Proceedings 2013*, pp.39–53.

Novelli, F. & Wenzel, S., 2013a. Adoption of an Online Sales Channel and “Appification” in the Enterprise Application Software Market. In *ECIS 2013 Proceedings*.

Novelli, F. & Wenzel, S., 2013b. Online and Offline Sales Channels for Enterprise Software: Cannibalization or Complementarity? In *ICIS 2013 Proceedings*.

OpenPeak, 2013. OpenShop. Available at: http://www.openpeak.com/OpenShop.html [Accessed December 3, 2013].

Pettey, C., 2011. Gartner Identifies the Top 10 Strategic Technologies for 2012. *Gartner*. Available at: http://www.gartner.com/newsroom/id/1826214 [Accessed November 19, 2013].

Pettey, C., 2012. Gartner Identifies the Top 10 Strategic Technology Trends for 2013. *Gartner*. Available at: http://www.gartner.com/newsroom/id/2209615 [Accessed November 20, 2013].

Puri, S.J., 1992. Industrial Vendors’ Selling Center: Implications for Sales Management. *Journal of Business & Industrial Marketing*, 7(3), pp.59–69.

Robinson, P.J., Faris, C.W. & Wind, Y., 1967. *Industrial Buying and Creative Marketing*, Allyn and Bacon.

Salesforce.com, 2014. AppExchange. Available at: https://appexchange.salesforce.com/ [Accessed December 23, 2014].

Salesforce.com, 2013. Private AppExchange. Available at: https://appexchange.salesforce.com/listingDetail?listingId=a0N3000000B4VIZEAV [Accessed December 3, 2013].

SAP, 2013a. SAP.COM - Secure Apps. Available at: www.sap.com/pc/tech/mobile/software/solutions/device-management/secure-apps.html [Accessed December 1, 2013].

SAP, 2013b. SAP Application Development Partner Center. Available at: www.sapadpc.com [Accessed November 8, 2013].

SAP, SAP HELP - Open Catalog Interface. Available at: https://help.sap.com/saphelp_dimp50/helpdata/en/5d/0eba5fa0084247acbb0c04cede18e8/content.htm [Accessed October 13, 2014].

SAP, 2013c. SAP Help - SAP Enterprise Store. Available at: http://help.sap.com/ses [Accessed December 1, 2013].

SAP, 2014. SAP Store. Available at: www.sapstore.com [Accessed December 27, 2014].

SAP, 2013d. SAP Store Support. *www.sapstore.com*. Available at: https://store.sap.com/sap/cp/ui/resources/store/html/CustomerSupport.html [Accessed December 10, 2013].

SAP, 2013e. Simple UI for SAP Applications: SAP Fiori Improves User Experience. Available at: http://en.sap.info/sap-fiori-improves-user-experience/98278 [Accessed November 1, 2013].

Scholten, S., 2011. *Platform-based innovation management : a framework to manage open innovation in two-sided platform businesses*, Stuttgart: Universität Stuttgart.

Sheth, J.N., 1973. A Model of Industrial Buyer Behavior. *Journal of Marketing*, 37(4), pp.50–56.

Sheth, J.N., 1996. Organizational buying behavior: past performance and future expectations. *Journal of Business Industrial Marketing*, 11(3/4), pp.7–24.

Symantec, 2013. Symantec App Center. Available at: http://www.symantec.com/mobility/products [Accessed December 3, 2013].

Verville, J. & Halingten, A., 2003. A six-stage model of the buying process for ERP software. *Industrial Marketing Management*, 32(7), pp.585–594.

Vizard, M., 2012. CIOs Struggle With Relevance of Role to Business. *CIO Insight*. Available at: http://www.cioinsight.com/it-management/cios-struggle-with-relevance-of-role-to-business [Accessed November 1, 2013].

Webster, F.E. & Wind, Y., 1972. A General Model for Understanding Organizational Buying Behavior. *The Journal of Marketing*, 36(2), pp.12–19.

Weill, P., 2004. Don't just lead, govern: How top-performing firms govern IT. *MIS Quarterly Executive*, 3(1), pp.1–17.

Weiß, F. & Leimeister, J.M., 2012. Consumerization - IT Innovations from the Consumer Market as a Challenge for Corporate IT. *Business & Information Systems Engineering*, 4(6), pp.363–366.

Wenzel, S., 2014. App'ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications. *Business Information Systems Conference 2014/ Lecture Notes in Business Information Processing*, 176, pp.61–72.

Wenzel, S., Faisst, W., Burkard, C. & Buxmann, P., 2012. New Sales and Buying Models in the Internet: App Store Model for Enterprise Application Software. In *MKWI 2012*. pp. 639–651.

Wenzel, S., Novelli, F. & Burkard, C., 2013. Evaluating the App-Store Model for Enterprise Application Software and Related Services. In *Wirtschaftsinformatik Proceedings 2013*.

# PART V: PRODUCT CHARACTERISTICS OF ENTERPRISE APPLICATION SOFTWARE FOR APP STORES

## V.1 Paper 7: App‘ification of Enterprise Software: A Multiple-Case Study of Big Data Business Applications

# APP‘IFICATION OF ENTERPRISE SOFTWARE:

# A MULTIPLE-CASE STUDY OF BIG DATA BUSINESS APPLICATIONS

**Stefan Wenzel**

SAP AG

University of Bamberg

*BIS 2014 Best Paper Award.*

Business Information Systems Conference 2014 (BIS2014) and part of the Lecture Notes in Business Information Processing book series (LNBIP, volume 176).

Available at: https://doi.org/10.1007/978-3-319-06695-0_6