



# Gemeinsame mentale Modelle in der agilen Softwareentwicklung: Ein Ansatz zur Erstellung von Gestaltungsempfehlungen für „gute“ erfahrungsspezifische User Stories

Daniel Hallmann<sup>1</sup>, Ute Schmid<sup>2</sup> und Rüdiger von der Weth<sup>3</sup>

**Abstract:** User Stories sind ein probates Mittel zum Austausch von Anforderungen in der agilen Softwareentwicklung. Die Anwendung ist jedoch kritisch, da sich immer wieder Fehler in der Beschreibung einzelner Formularfelder wie den Akzeptanzkriterien einschleichen. Mit diesen fehlerhaften User Stories müssen Personen mit individuellen Erfahrungen ein gemeinsames mentales Modell konstruieren. Häufig ist der Aufbau jedoch nicht möglich und es kommt zu negativen Reaktionen wie Frustration bei fehlenden Erfahrungen, langen Diskussionen bei unterschiedlichen Vorstellungen und teuren Nacharbeiten bei missverstandenen Inhalten. Aufgrund dieser Effekte sollen mit einer Ex-post-Analyse, einer Feldstudie und einem Experiment die Zusammenhänge von fehlerhaften User Stories, individuellen Erfahrungen und gemeinsamen mentalen Modellen untersucht werden. Mit dieser Sicht auf die Verarbeitung von User Stories wird das Forschungsinteresse für kognitionspsychologische Prozesse in der agilen Softwareentwicklung unterstützt. Außerdem liefert die Arbeit für die Praxis Gestaltungsempfehlungen für „gute“ erfahrungsspezifische User Stories. Mit dieser Strategie sollen frühzeitig die Kosten in der agilen Softwareentwicklung reduziert werden.

**Keywords:** Agile Softwareentwicklung, Qualität von User Stories, Kognitive Psychologie, Erfahrungen, Gemeinsame Mentale Modelle

## 1 Einleitung

Für den Austausch von Anforderungen zwischen einem Product Owner, dem Interessenvertreter des Kunden und den Entwicklern nehmen User Stories in der agilen Softwareentwicklung eine zentrale Position ein. Bei der Erstellung durch den Product Owner schleichen sich jedoch immer wieder Fehler ein. Danach werden die *Felder* im Formular: *Titel* - Als *Rolle* möchte ich *Ziel*, um *Nutzen* - *Akzeptanzkriterien* [Co04] entweder (i) falsch ausgefüllt oder (ii) gar nicht angegeben [Gi10, Ka11, Kl13, NS10, Vi11, Wa12] Neben einer falschen Beschreibung von Titel, Rolle und Ziel fehlen bei einigen User Stories gänzlich die Angabe des Nutzens und der Akzeptanzkriterien. Außerdem ist es in der agilen Softwareentwicklung nicht unüblich, dass Entwickler mit unterschiedli-

---

<sup>1</sup> Otto-Friedrich-Universität Bamberg, Lehrstuhl für Kognitive Systeme, An der Weberei 5, 96047 Bamberg, daniel.hallmann@uni-bamberg.de

<sup>2</sup> Otto-Friedrich-Universität Bamberg, Lehrstuhl für Kognitive Systeme, An der Weberei 5, 96047 Bamberg, ute.schmid@uni-bamberg.de

<sup>3</sup> Hochschule für Technik und Wirtschaft Dresden, Professur Betriebswirtschaftslehre / Personalwirtschaft und Arbeitswissenschaften, Friedrich-List-Platz 1, 01069 Dresden, weth@htw-dresden.de

chen technischen und kundenspezifischen Erfahrungen zusammenarbeiten [Bo81, Es07]. Die Kenntnisse sind dabei abhängig vom persönlichen Werdegang und variieren bei Fragen zu Entwicklungswerkzeugen oder zum Fachbereich des Kunden.

Trotz dieser mangelhaften Qualität und individuellen Erfahrungen der Entwickler müssen sich der Product Owner und die Programmierer auf ein gemeinsames Bild der Anforderung festlegen. Häufig ist eine Einigung problematisch, da bei fehlerhaften User Stories beispielsweise neue Entwickler in Softwareprojekten mit Unverständnis reagieren. Aussagen wie:

„Damit kann ich nichts anfangen“ (O-Ton eines Entwicklers)

zeigen den Unmut bezüglich verminderter Qualität sowie unzureichender Erfahrungen. Hinzu kommen möglicherweise teure Nacharbeiten bei missverstandenen Inhalten und lange Diskussionen bei ungleichen Vorstellungen. Qualitativ schlechte User Stories und falsch eingeschätzte Erfahrungen können damit nicht nur Frustration hervorrufen, sondern sich auch erheblich auf die Wirtschaftlichkeit des Softwareprojektes für den Kunden durch einen erhöhten Korrektur- und Abstimmungsaufwand auswirken.

Ziel der Arbeit ist es die Zusammenhänge zu verstehen, die zwischen fehlerhaften User Stories, individuellen Erfahrungen sowie dem gemeinsamen Bild der Anforderung bestehen. Die Auswertung der Ergebnisse soll zur Erstellung von Gestaltungsempfehlungen für „gute“ erfahrungsspezifische User Stories genutzt werden. Dazu wird im Folgenden die Literatur erörtert, Punkt 3 beschreibt die Ziele, den Inhalt und die Voraussetzungen von Gestaltungsempfehlungen, Punkt 4 erläutert den Forschungsplan, Punkt 5 geht auf den Stand der Arbeit ein und Punkt 6 stellt den wissenschaftlichen und praktischen Nutzen vor.

## 2 Bisherige Arbeiten

Empirische Untersuchungen zu User Stories im Bereich der agilen Softwareentwicklung konzentrieren sich auf die Analyse von Praktiken und Problemstellungen zu Anforderungsdokumenten in Projektteams [dAO05, SRP09, RCB10, SK13]. Parallel werden in theoretischen Abhandlungen neue Techniken für die Verwendung von User Stories ausgearbeitet [MBA09, CD13].

Sowohl die empirischen wie auch die theoretischen Untersuchungen betrachten User Stories auf einer organisatorischen und technischen Ebene. Auf den kognitionspsychologischen Verstehensprozess zwischen dem Product Owner und den Entwicklern wird dagegen nicht eingegangen. Unbeachtet bleibt der Prozess zum Aufbau eines gemeinsamen mentalen Modelles [CCS93] zu der User Story, mit deren Hilfe die Personen eine kollektive Beschreibung und Erwartungshaltung der Aufgabe formulieren. Außerdem fehlen Erkenntnisse zu Einflussgrößen auf den Prozessverlauf. Danach ist die beiderseitige Wirkung von mangelhaften Formularfeldern in der User Story wie beispielsweise

fehlerhafte Akzeptanzkriterien sowie individuelle Erfahrungen wie etwa technische Programmierkenntnisse und kundenspezifische Produktionsabläufe auf das gemeinsame mentale Modell unbekannt. Um diese Forschungslücke zu schließen, wird daher folgende Forschungsfrage gestellt:

Wie beeinflussen:

- (i) die *Qualität* einer User Story  $US_1$  und
- (ii) individuelle *Erfahrungen* der Entwickler
- (iii) das *gemeinsame mentale Modell* von  $US_1$  zwischen dem Product Owner und den Entwicklern?

### 3 Idee

Nach dem ersten Lesen einer User Story durch die Entwickler sollte sich zwischen dem Product Owner und den Entwicklern ein gemeinsames Bild zum Inhalt der Anforderung einstellen. Ein falsch ausgefülltes Formular und unbeachtete Erfahrungen bei den Entwicklern dürfen nicht mehr zu Frustration, einer differenzierten Vorstellung der Aufgabe und weiterführend zu teuren Nacharbeiten und zeitraubenden Diskussionen führen. Vielmehr soll ein Product Owner, als Autor von User Stories, aktiv den gemeinsamen Verstehensprozess zum Inhalt einer User Story positiv beeinflussen können. Dazu braucht es Hilfestellungen für einen gezielten Eingriff in den Vorgang mit Aussicht auf eine Verbesserung.

Ein Vorgehen kann hier die Erstellung von Gestaltungsempfehlungen für „gute“ erfahrungsspezifische User Stories sein. Die Hinweise sollen sich dabei an der Wirkung von Fehlern und individuellen Erfahrungen auf das gemeinsame mentale Modell orientieren. Die Idee beinhaltet zwei Gestaltungsdimensionen: (i) Eine qualitative Stabilisierung berücksichtigt die Einhaltung der formalen Kriterien. Alle Formularfelder sollen „vorhanden“, „korrekt“ und „aktuell“ geschrieben sein. (ii) Eine erfahrungsspezifische Individualisierung umfasst die Anpassung des Detailgrads der User Story anhand vorhandener Erfahrungen. Ziel ist die Erstellung einer optimierten User Story, bei der die Entwickler nicht mehr durch Fehler irritiert und mit dem Inhalt über- oder unterfordert sind. Qualitativ stabilisierte und individualisierte User Stories sind damit eine wichtige Voraussetzung für ein gemeinsames mentales Modell zwischen dem Product Owner und den Entwicklern.

Für den Aufbau der Empfehlungen müssen der Inhalt und die Struktur sowie die Zusammenhänge der Fehler und die individuellen Erfahrungen bekannt sein. Dazu gehören eine fertige Kategorisierung der Problemstellen in den Formularfeldern und ein Konzept zu unterschiedlichen Ausprägungen von Erfahrungen der Entwickler sowie eine inhaltliche Klärung der wechselseitigen Beziehungen. Der Detailbetrachtung von fehlerhaften User Stories und Erfahrungen von Entwicklern mit beiderseitiger Wirkung muss eine Sicht auf die Effekte von Fehlern in User Stories und individuellen Erfahrungen auf das gemeinsame mentale Modell folgen.

## 4 Forschungsplan

Ein Strukturgleichungsmodell [BS10] formt die Basis für das Forschungsmodell. Es beschreibt die Beziehungen zwischen den latenten und nicht direkt beobachtbaren Variablen Qualität der User Story, Erfahrungen der Entwickler und gemeinsames mentales Modell. Im Strukturgleichungsmodell werden auch die Beziehungen der latenten Variablen zu den messbaren Indikatoren definiert. Folgende Schritte zeigen den Weg für die Bearbeitung der Forschungsfrage anhand des Strukturgleichungsmodells. Eine Ex-post-Analyse, eine Feldstudie und ein Experiment bilden den empirischen Kern der Arbeit.

- Themenfindung, Aufstellen der Forschungsfrage und Überführung der Fragestellung in ein Strukturgleichungsmodell im 1. Jahr.
- Indikatorbildung für das Strukturgleichungsmodell auf Basis einer Ex-post-Analyse und einer Studie der COCOMO-Ansätze [Bo81, BR87, Bo95] im 2. Jahr.
- Erhebung von empirischen Daten für die Modellauswertung in einer Feldstudie in Softwareprojekten [LWW01] im 3. Jahr.
- Betrachtung des Strukturgleichungsmodells unter der Kontrolle von Störgrößen über ein Experiment [HSE09] im 4. Jahr.
- Konzeption der Gestaltungsempfehlungen für User Stories und Abgabe der Dissertationsschrift im 5. Jahr.

## 5 Stand der Arbeit

Die Forschungsarbeit befindet sich im 2. Jahr. Abgeschlossen sind die Themenfindung, die Formulierung der Forschungsfrage und die Übersetzung der Forschungsfrage in ein Strukturgleichungsmodell. Das im Strukturgleichungsmodell enthaltene Strukturmodell besteht aus den latenten Variablen Qualität der User Stories, Erfahrungen der Entwickler und dem gemeinsamen mentalen Modell zwischen Product Owner und Entwicklern.

Aktuell in Bearbeitung befindet sich die Konzeption des Messmodells für das Strukturgleichungsmodell. Dazu werden manifeste Indikatoren für die latenten Variablen bestimmt. Geplant ist die Durchführung einer Ex-post-Analyse auf Basis von abgeschlossenen User Stories. Ergänzt wird die Operationalisierung der latenten Variablen, um eine Studie mit Schwerpunkt auf die COCOMO-Modelle [Bo81, BR87, Bo95] im Licht der agilen Softwareentwicklung.

Es folgen Arbeiten zur Auswertung des Strukturgleichungsmodells. Das Vorgehen umfasst die Erhebung der messbaren Indikatoren in einer Feldstudie in unterschiedlichen Softwareprojekten. Möglich wird damit die Berechnung der Korrelationen im Strukturgleichungsmodell. Den Abschluss bildet eine experimentelle Phase. Der Schritt dient der objektiven Überprüfung der Kausalbeziehungen unter Kontrolle von Störvariablen.

## 6 Nutzen

Die Evaluation der Ergebnisse präsentiert wissenschaftliche Erkenntnisse über Zusammenhänge von fehlerhaften User Stories, individuellen Erfahrungen und gemeinsamen mentalen Modellen. Mit dieser Sicht auf die Verarbeitung von User Stories wird das Forschungsinteresse für kognitionspsychologische Prozesse in der agilen Softwareentwicklung [SRP09] unterstützt.

Unter Einhaltung der Gestaltungsanweisungen in der Praxis ergeben sich motivationale Verbesserungen und zeitliche sowie finanzielle Einsparungen für die agile Softwareentwicklung. Bei der Vermeidung von Irritation durch Fehler und der Abschaffung von Unter- und Überforderung wird eine motivierte Arbeitseinstellung bei Entwicklern gefördert. Außerdem wirken die Anweisungen gegen unbemerkte Abweichungen im gemeinsamen Bild zur User Story im Projektteam, was eine Senkung von teuren Nacharbeiten bereits in einer frühen Phase des Entwicklungsprozesses ermöglicht. Hinzu kommt mit der Reduzierung von langen Diskussionen eine Einsparung von Entwicklungszeit, was sich ebenfalls positiv auf die Wirtschaftlichkeit auswirkt.

## Literaturverzeichnis

- [Bo81] Boehm, B. W.: *Software Engineering Economics*. Prentice-Hall, N.J., 1981.
- [Bo95] Boehm, B. et al.: *Cost Models for Future Software Life Cycle Processes: COCOMO 2.0*. In *Annals of Software Engineering*, 1995, 1; S. 57–94.
- [BR87] Boehm, B.; Royce, W.: *Ada COCOMO and the Ada Process Model*. In *Proceedings. Third COCOMO Users Group Meeting*, SEI, 1987; S. 1–34.
- [BS10] Bortz, J.; Schuster, C.: *Statistik für Human- und Sozialwissenschaftler*. Springer Berlin Heidelberg, Heidelberg, 2010.
- [CCS93] Converse, S.; Cannon-Bowers, J. A.; Salas, E.: *Shared Mental Models in Expert Team Decision Making*. In *Individual and Group Decision Making: Current Issues*, 1993; S. 221–246.
- [CD13] Chugh, N.; Dev Mishra, A.: *Assimilation of Four Layered Approach to NFR in Agile Requirement Engineering*. In *International Journal of Computer Applications*, 2013, 78; S. 25–28.
- [Co04] Cohn, M.: *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional, 2004.
- [dAO05] de Souza, Sergio Cozzetti B; Anquetil, N.; Oliveira, K. M. de: *A Study of the Documentation Essential to Software Maintenance: Proceedings of the 23rd Annual International Conference on Design of Communication: Documenting & Designing for Pervasive Information*. ACM, New York, NY, USA, 2005; S. 68–75.
- [Es07] Espinosa, J. A. et al.: *Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development*. In *Organization Science*, 2007, S. 613–630.

- [Gi10] Gill, W.: Better Scrum User Stories: Save the Solution for the Spec. <http://williamgill.de/2010/12/16/better-scrum-user-stories-save-the-solution-for-the-spec/>, 14.04.2016.
- [HSE09] Hussy, W.; Schreier, M.; Echterhoff, G.: *Forschungsmethoden in Psychologie und Sozialwissenschaften für Bachelor*. Springer-Verlag, 2009.
- [Ka11] Kaczor, K.: 5 Common Mistakes We Make Writing User Stories. <http://www.scrumalliance.org/community/articles/2011/august/5-common-mistakes-we-make-writing-user-stories>, 14.04.2016.
- [Kl13] Klement, A.: The Problem with User Stories and What's Better. <http://alanklement.blogspot.de/2013/03/the-problem-with-user-stories-and-whats.html>, 14.04.2016.
- [LWW01] Levesque, L. L.; Wilson, J. M.; Wholey, D. R.: Cognitive Divergence and Shared Mental Models in Software Development Project Teams. In *Journal of Organizational Behavior*, 2001, 22; S. 135–144.
- [MBA09] Miranda, E.; Bourque, P.; Abran, A.: Sizing User Stories Using Paired Comparisons. In *Information and Software Technology*, 2009, 51; S. 1327–1337.
- [NS10] Nazzaro, W.; Suscheck, C.: New to User Stories? <http://www.scrumalliance.org/community/articles/2010/april/new-to-user-stories>, 14.04.2016.
- [RCB10] Ramesh, B.; Cao, L.; Baskerville, R.: Agile Requirements Engineering Practices and Challenges: An Empirical Study. In *Information Systems Journal*, 2010, S. 449–480.
- [SK13] Stettina, C. J.; Kroon, E.: Is There an Agile Handover? An Empirical Study of Documentation and Project Handover Practices Across Agile Software Teams: 19th ICE & IEEEITMC International Conference, The Hague, Netherlands, 2013; S. 1–12.
- [SRP09] Sharp, H.; Robinson, H.; Petre, M.: The Role of Physical Artefacts in Agile Software Development: Two Complementary Perspectives. In *Interacting with Computers*, 2009, 21; S. 108–116.
- [Vi11] Vin: User Stories: Focus on the Problem Not the Solution. <http://brainslink.com/2011/02/user-stories-must-focus-on-the-problem-not-the-solution/>, 14.04.2016.
- [Wa12] Wake, B.: “As a Developer...” Is Not a User Story. <http://www.industriallogic.com/blog/as-a-developer-is-not-a-user-story/>, 14.04.2016.