



# Managing the Agile Process of Human-Centred Design and Software Development

Peter Forbrig<sup>1</sup> and Michael Herczeg<sup>2</sup>

<sup>1</sup>University of Rostock, Institute of Computer Science  
Albert-Einstein-Str. 22, 18051 Rostock  
[peter.forbrig@uni-rostock.de](mailto:peter.forbrig@uni-rostock.de)

<sup>2</sup>University of Lübeck, Institute of Multimedial and Interactive Systems  
Ratzeburger Allee 160, 23562 Lübeck  
[herczeg@imis.uni-luebeck.de](mailto:herczeg@imis.uni-luebeck.de)

**Abstract.** The paper provides and discusses a life cycle model for agile software development methods like SCRUM. The process model integrates the idea of Human-Centred Design (HCD) in the agile approach. Based on the discussion of existing life cycle models and known process patterns, a process model is discussed that combines the advantages of HCD and SCRUM. The SCRUM process model was used in the discussion to be as concrete as possible. However, the suggested approach is applicable to other agile process models as well.

## 1 Introduction

It is generally accepted that software should be developed in an iterative way because otherwise it is difficult to capture all requirements in a sufficient way. Even that the classical waterfall model has some advantages for the definition of milestones it is agreed, that the spiral model of Boehm and the agile software development together with formative evaluation methods provide better software quality in the sense of usability and user experience.

## 2 Related Work

Agile software development has become popular since the late 1990s. It became popular because many projects failed because it took too much time from finalised requirements specification to first tests of the developed system. The approach was consolidated by a manifesto at the beginning of ten 2000s [1] that characterises the agile idea by twelve main principles that have to be followed.

Figure 1 gives additionally an overview of influences to agile software development. It presents approaches from planning, analysis, design, build, test deploy and review that found their way into agile development methods.

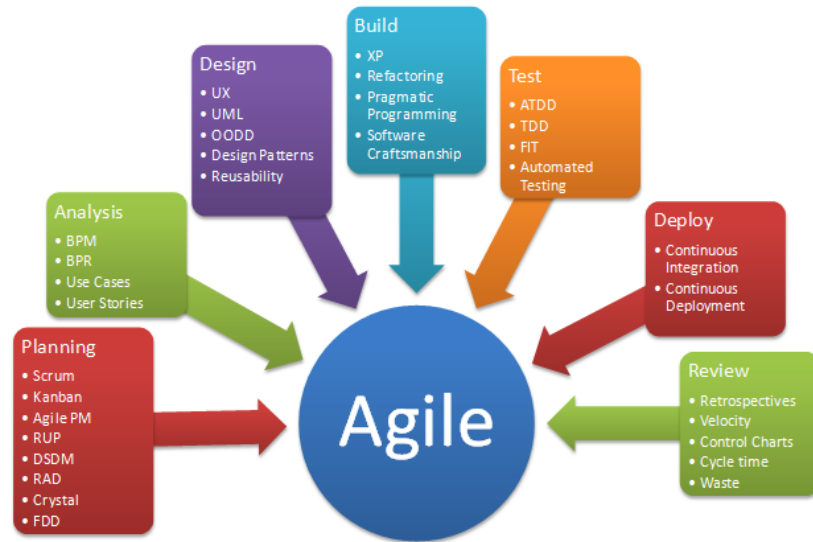


Figure 1. Agile influences (from <http://blog.kiandra.com.au/wp-content/uploads/2013/08/Agile-21.png>)

Currently, one of the most popular agile software development approaches is SCRUM, which will be revisited in the following paragraph.

## 2.1 Agile Software Development - SCRUM

To make things more concrete we focus on one specific agile development method. One of the most popular methods is SCRUM [9]. Therefore we will focus on the related process models and the corresponding terminology.

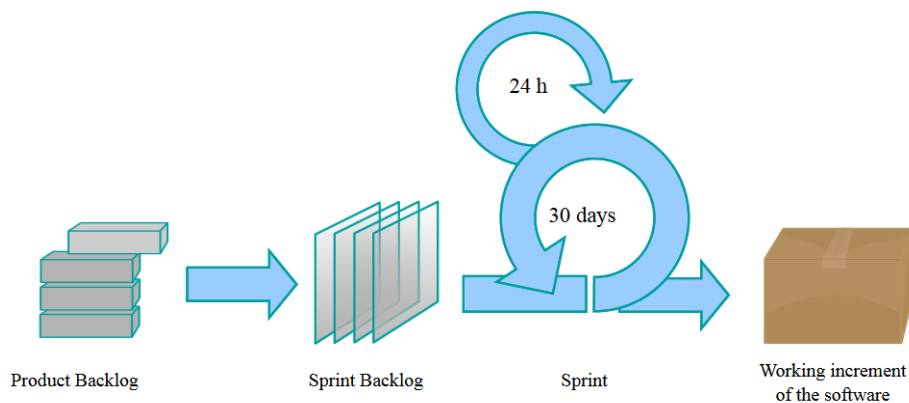


Figure 2. The scrum process (from [http://en.wikipedia.org/wiki/Scrum\\_software\\_development](http://en.wikipedia.org/wiki/Scrum_software_development))

Figure 2 presents the main process model for SCRUM. Based on the specified product backlog collecting all requirements for the product, a sprint backlog is selected that contains all requirements that have to be implemented during the forthcoming sprint. The result is an increment of the necessary software.

## 2.2 Human-Centred Design (HCD)

In the same way that SCRUM is popular for software engineering experts, HCD is popular for usability and user experience experts. One of the main reasons for its success is that context of use, the requirements of the users and the evaluation of design solutions play an important role. User requirements are more important than technical features that software engineers might derive. Users get what they really want.

The HCD process has been standardised by ISO 9241-210. Figure 3 gives a visual overview of the corresponding process model.

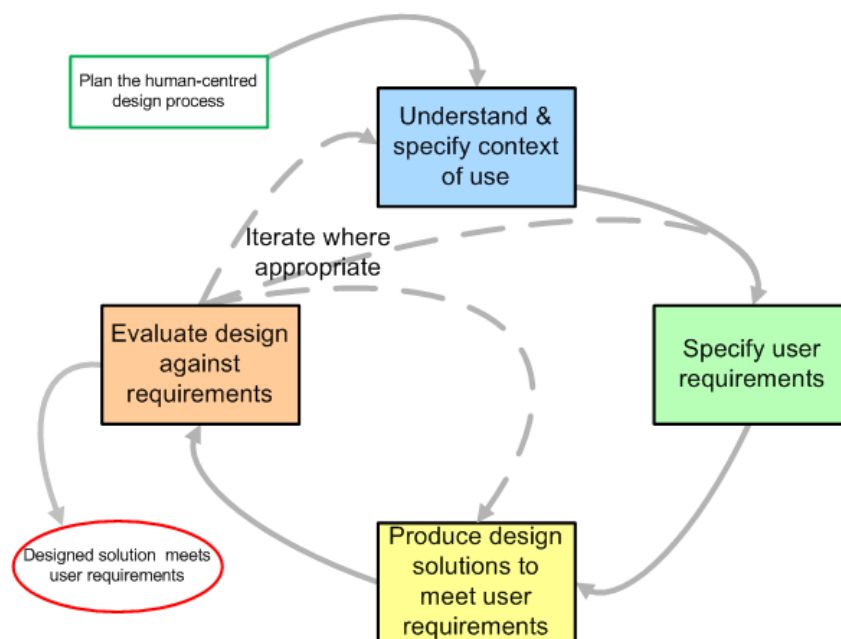


Figure 3. The design process from ISO 9241-210– Human-centred design process (from <https://thestandardinteractiondesignprocess.wordpress.com/> )

At the University of Lübeck a Usability Engineering Repository (UsER) was developed [7]. It is a framework, platform and repository for an integrated and modular development of human-machine systems with the special scope of usability engineering.



An integrated model for HCD and agile software development was developed in [8] and will be presented in the following paragraph.

### 2.3 Approaches for Combining HCD and SCRUM

It was already mentioned that a combined approach for HCD and SCRUM would be appreciated by a lot of software developers because such a unified approach would give the chance to combine the advantages of both approaches. Paelke et al. [6] published the following process model and called it “Agile UCD-Process”. However, it is more a UCD-Process with agile software development. Nevertheless, it is a very interesting starting point.

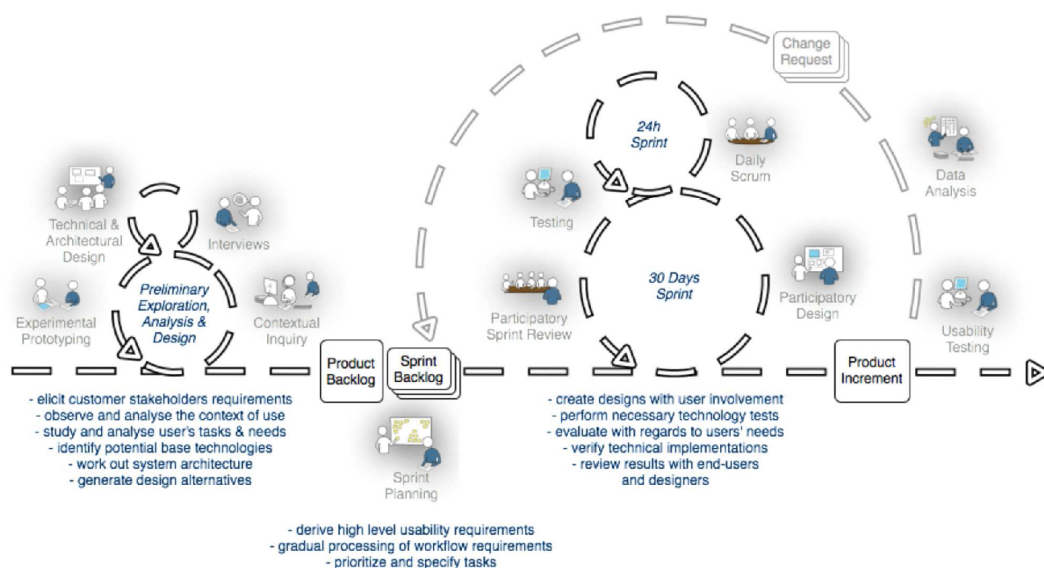


Figure 5. Agile User-Centred Design Process (Paelke et al. [5])

From our point of view it is interesting to start with UCD activities. However, the requirements elicitation is only loosely coupled with the software development process. A stronger coupling was suggested by Paul et al. [9] and is presented by the following Figure.

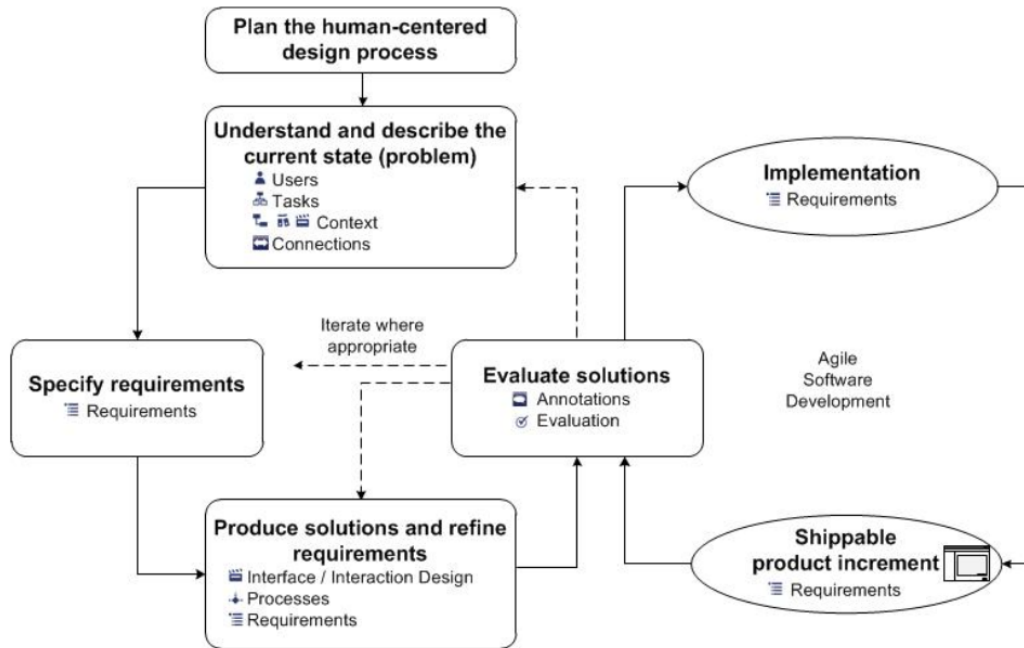


Figure 6. Extended User-Centred Design Process (Paul et al. [9])

According to this approach all created software influences the HCD process. U-SCRUM [11] is an approach with similar goals. It suggests to integrate usability aspects in SCRUM by having two product owners and to use the concept of personas for communication. However, the process is only roughly described by necessary artifacts.

Sinah et al [10] provide a practitioner perspective to the problem. They analysed that “*agile teams suffer from the additional challenge of lack of management support to UCD activities that occurred to participants*”. They mention that a clear roadmap for combining both approaches is missing and that parallel tracks might help. We try to go one step forward to such a roadmap by presenting a process model in paragraph 2.5.

## 2.4 Process Patterns

Patterns were introduced to the computer science community by the “Gang of Four” [3]. Their collection of Design Patterns was a great success for reusing knowledge of solutions for reoccurring problems. In the meantime patterns for different domains like tasks, workflows or user interfaces were discovered and published. Sutherland et al. [13] published nine patterns related to SCRUM.

We will discuss with the pattern “*Yesterday’s weather*” only one of them. This pattern says: “*In most cases, the number of Estimation Points completed in the last Sprint is the most reliable predictor of how many Estimation Points will be completed in the next Sprint*”. The other patterns are of similar kind

and do not say a lot about the SCRUM process model. However, Bertholdo et al. [2] analysed projects and reports and provided patterns for agile software development that ask for consequences of the process model. The first pattern is called “*Sprint Zero*” and attacks the following problem: “Missing the big picture of the system in the beginning ...”. It suggests the following solution: “*A short Sprint before the code implementation to define a broader view of the product, general goals, to roughly plan the next sprints and to define design principles*”. This process model of Figure 5 can be seen as one variant of the consequences of this pattern.

Another pattern is called “*One Sprint Ahead*”. It is related to the following problem: “*Making the development and the UX team synchronized, that both can collaborate and provide input to the development workflow*”. There is the following suggested solution: “*The UX team works at least one iteration ahead of the development team*”.

The pattern “*Parallel Tracks*” provides the following solution: “*UX or usability team work in a parallel track with the development team in order to synchronise their activities*”.

The suggested process model of Figure 6 is designed according to the solution of this pattern.

### 3 Updated Process Model for SCRUM

The result of our studies in projects studies and literature research is the following process model in Figure 7.

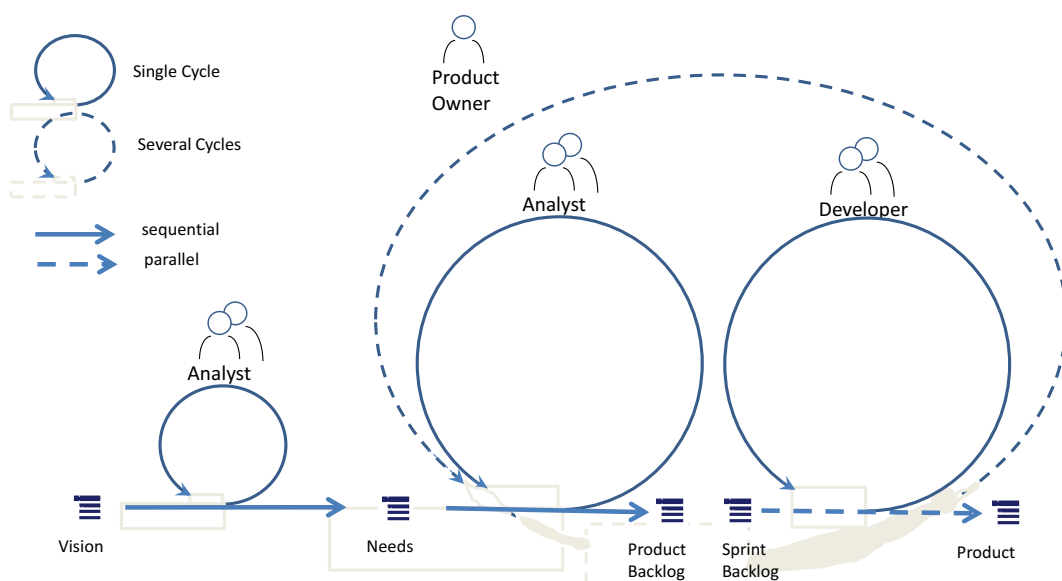


Figure 7. Suggested Human-Centred Design Process for SCRUM



requirements under development. The current state and context of use is analysed and represented by different models. Requirements are specified afterwards before possible solutions are produced and later evaluated. The identified requirements are collected and specified as product backlog.

We do not focus on the SCRUM development cycle because our process model is independent from details of these models. More or less each agile development cycle can be supported.

## 4 Summary and Outlook

The paper discussed the two widely accepted process models for Human-Centred Design and Agile Development and shows how to combine them beneficially. Additionally, there was a focus on patterns that were published in different scientific papers. They encapsulate some knowledge about the integration of usability aspects in agile software development and provided arguments for presenting an own extended process model for the integrated approach of Human-Centred Design and SCRUM. Although the terminology of SCRUM was used the process model can be used in conjunction with any other agile development process. The provided solution is based on the “*Zero Sprint Pattern*”, “*One Sprint Ahead Pattern*” and the “*Parallel Track Pattern*”. It can be combined with the idea of having UX specialist as product owners and having UX specialists in the development team.

It might make sense to study the organisation of teams with the specific roles in more detail in the future. One consequence could be that all people acting in the development process should take part in the process of identifying needs on the project’s vision.

## References

- [1] Agile Manifesto, <http://agilemanifesto.org/>, last visited June 4th 2015.
- [2] Bertholdo, A.P.O., da Silva, T.S., de Melo, C.O., Kon, F. and Silveira, M.S. Agile Usability Patterns for UCD Early Stages, In Marcus A. (Ed.) DUXU 2014 Part I; LNCS 8517, pp. 33 – 44.
- [3] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994

- [4] Herczeg, M., Kammler, M., Mentler, T. and Roenspieß, A. The Usability Engineering Repository UsER for the Development of Task- and Event-based Human-Machine-Interfaces, In: Narayanan, S. (Ed.). 12th IFAC, IFIP, IFORS, IEA Symposium on Analysis, Design and Evaluation of Human-Machine-Systems. Las Vegas : International Federation of Automatic Control. pp. 483-490
- [5] Memmel, T., Gundelsweiler, F. and Reiterer, H. 2007. Agile human-centered software engineering. In Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1 (BCS-HCI '07), Vol. 1. British Computer Society, Swinton, UK, UK, 167-175.
- [6] Paelke, V. and Nebe, K. Integrating Agile Methods for Mixed Reality Design Space Exploration. In Proceedings of the 7th ACM conference on Designing interactive systems (DIS '08). ACM, New York, NY, USA, 240-249. <http://doi.acm.org/10.1145/1394445.1394471>
- [7] Paul, M., Roenspieß A., Mentler T., Herczeg M. (2014). The Usability Engineering Repository (UsER). In Hasselbring, W & Ehmke, N C (Eds.) Software Engineering 2014 - Fachtagung des GI-Fachbereichs Softwaretechnik, 25.-28. Februar 2014, Kiel. Gesellschaft für Informatik e.V. (GI). 113-118
- [8] Paul, M. Systemgestützte Integration des Usability-Engineerings in den Software-Entwicklungsprozess, PhD Thesis, University of Lübeck, 2015.
- [9] Rising, L. and Janoff, S. N. "The Scrum Software Development Process for Small Teams", IEEE Software, vol.17, no. 4, pp. 26-32, July/August 2000, doi:10.1109/52.854065
- [10] Salah, D., Paige, R. and Cairns, P. A Practitioner Perspective on Integrating Agile and User Centred Design, Proceedings of the 28th International BCS Human Computer Interaction Conference (HCI 2014),
- [11] Singh, M. U-SCRUM: An agile methodology for promoting usability, Integrating usability engineering and agile software development: A literature review. In Proc. AGILE 2009, IEEE Press, pp. 555-560.
- [12] Sohaib, O. and Khan, K, Integrating usability engineering and agile software development: A literature review. In Proc. International Conference on Computer design and Applications (ICCD), Volume 2, p. 32 – 38, 2010.
- [13] Sutherland, J. Harrison, N. and Riddle, J. Teams That Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams. In Proc. HICSS 2014, pp. 4722-4728.