

Mobile Scrum

Stephan Krusche, Tobias Konsek

krusche@in.tum.de, tobias.konsek@mytum.de

Abstract: In this paper we discuss whether mobile applications can support software developers to improve their efficiency and communication. We propose the idea of Mobile Scrum, a native and integrated mobile application that supports scrum teams in their activities and fits into their environment. Team members can use it anytime and anywhere to communicate and share knowledge within the team.

Mobile Scrum provides a lightweight, yet easy and intuitive to use interface compared to existing web-based and desktop applications. With guides and templates it helps to prevent typical problems when applying Scrum. Furthermore it improves the consistency of scrum artifacts and enables easy access to important information.

1 Introduction

In 1986, Takeuchi and Nonaka introduced the idea of agile methods for product development. [TN86] They focused on the change from linear to non linear and adaptive processes to be more flexible in a "fast-paced fiercely competitive world of new product development". This idea was adapted to software engineering by Schwaber in 1995. [S⁺95] He introduced Scrum as a development process to respond to change and minimize risks and shows how Scrum can be applied in software engineering projects.

Since then, the technical preconditions have significantly changed. Mobile devices, that are becoming increasingly popular [Mob12], replace desktop computers in many situations. The access to information is possible anywhere and anytime, and the computational power of mobile devices comes close to desktop computers, facilitating productive work with these devices. The increasing integration of mobile services and applications allows the usage of context-sensitive data to provide additional features, like e.g. location-based services.

With mobile devices, human-computer interaction has become usable and has been integrated into everyday life. Ubiquitous computing describes this progress and is considered as an advancement from the old desktop paradigm. The goal is to create "machines that fit the human environment instead of forcing humans to enter theirs." [Wei99] Integrated tools enhance existing processes and do not "hinder the workflow or frustrate users". [YP04]

With these trends the question arises whether we can create mobile applications and services that improve the software engineering process. In this paper we want to investigate one case, the integration of mobility into lightweight, agile projects that apply Scrum. We describe a mobile application, called Mobile Scrum, that can improve efficiency and com-

munication within a team and integrates tightly into developers and managers workflows. We think that a mobile application can help to avoid typical mistakes that often occur when applying Scrum, especially with unexperienced teams. Maintaining communication artifacts like e.g. the product backlog in a consistent way can also be simplified with mobile tool support.

The paper is organized as follows. Section 2 describes typical problems that can occur when applying Scrum in more detail. It further explains why an integrated and native mobile application is helpful to overcome some of these problems. In the third section we show related work in the area of software engineering on mobile devices and how this paper relates to it. Section 4 clarifies why usability is our main design goal. It further shows two scenarios in which Mobile Scrum can be used to improve communication and consistency. In section 5 we summarize the advantages and describe possible future work.

2 Motivation

Some typical mistakes often occur in Scrum projects. Kniberg, a certified scrum trainer and member of the Scrum Alliance, teaches "10 Ways to Screw Up with Scrum and XP". [Kni08] One problem he observed in many projects, is missing information about the velocity of teams. In these cases planning needs more time and is not as efficient as possible. Estimations for the amount of user stories, that can be completed during one sprint, are not accurate. Mobile Scrum shows the velocity in burn down charts which are refreshed automatically with every change in the system. Thereby, the progress is transparent at any time and planning becomes easier.

Kniberg also observed that the product backlog is often not maintained correctly during the project which leads to inconsistencies. If the product owner does not prioritize, it is hard to decide which user stories will be implemented during the sprint. If the size of a user story is too large, the team cannot estimate its effort correctly and might not complete it in one sprint which is one principle of agile planning. [Coh06]

We think that a mobile application reduces the time of planning and supports the product owner in creating and maintaining the product backlog. For managing user stories Mobile Scrum follows the "As a user, I want" template that "helps the product owner prioritize". [Coh08] This structure includes the type of user, the goal and the reason of the user story. With this approach Mobile Scrum can simplify and improve planning and provides clear insights to the roadmap and business plan that are missing in many projects.

Sutherland, one of the inventors of Scrum, describes seven ways to fail with Scrum. [Sut93] One failure point he mentions is that important information in the daily scrum meeting, i.e. status, impediments and promises, is often not communicated by each participant. Another problem is that the daily scrum meeting often takes too much time, i.e. more than the standard fifteen-minute time box, because the whole team discusses too long on issues that can also be solved in smaller groups. We think that a mobile application can guide the team to follow the principles of the daily scrum meeting more accurately.

Without using dedicated computer applications, the mentioned problems can hardly be

solved. Information can easily get lost, e.g. when a sticky note on a physical taskboard disappears or is not preserved after the end of the sprint. Even with a web based or desktop based application, it is difficult and time-consuming to store important information about tasks, impediments or backlog items in a consistent way. We observed that meeting minutes are often inserted into a tool after the meeting, which causes additional effort that can be avoided. Subjective impressions of different team members might lead to information loss, e.g. if the minute taker omits information that is important to other team members. With Mobile Scrum, notes can be collected and visualized directly during the daily scrum meeting to prevent these problems.

Another problem is that existing web based applications for Scrum (e.g. Greenhopper¹) are not optimized to be used on mobile devices, i.e no native mobile application exists. When using a web application on a mobile device, offline usage is limited, sensors and actuators of mobile devices cannot be used and the performance of web pages is too slow for a satisfying user experience. Some existing tools are just adopted to support Scrum and do not fulfill the users expectations concerning Scrum. For instance, only a few of the investigated existing web applications include a taskboard, one of the central Scrum artifacts that should be used to visualize sprint progress. [Coh09]

3 Related work

”Software tools and environments that are designed to enhance productivity” during the software engineering process, by automating tasks or making them more enjoyable, are called computer-aided software engineering (CASE) tools. Reiss separates CASE tools into categories like program editors, design editors or testing aids and concludes that a future category of case tools is needed that ”provide assistance in bookkeeping and managing the process”. [Rei96] Meanwhile management tools are available for web-based and desktop-based platforms. A native mobile application that integrates into the developers workflows is currently missing.

Tillman, a researcher at Microsoft who explores the *Future of Software Engineering on Mobile Devices*², states ”the world is experiencing a technology shift”. Mobile devices are becoming increasingly popular in work environments and present several challenges for user interface design, especially when going beyond consumer tasks. In [TMdH⁺12] he presents TouchDevelop, a CASE tool within a mobile application that can be used to develop mobile applications directly on the mobile device. Thereby, TouchDevelop ”embraces the form factor and input capabilities of such mobile devices”.

Overall, mobile CASE tools are quite an unexplored research topic with few publications. With Mobile Scrum we want to contribute to this research area by investigating the usefulness of collaborative issue- and task management on mobile devices to support the software engineering process.

¹<http://www.atlassian.com/software/greenhopper>

²<http://www.cs.sun.ac.za/colloquia/future-of-software-engineering-on-mobile-devices>

4 Solution

Increasing mobility allows software development teams to work anywhere and anytime. This can enhance the individual freedom of team members and can help them to improve their efficiency and productivity, but it is only possible if they are able to work on their mobile device and if work artifacts are synchronized on a central server. Mobile Scrum applications synchronize data with an issue tracking server³ as shown in figure 1 when the mobile device is online. Thus, teams benefit from a fast information distribution and improved collaboration. This would not be possible with a physical taskboard, especially if the team is not able to work in the same office. Furthermore, existing desktop or web based applications are not able to provide the same level of mobility.

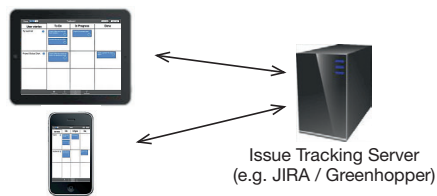


Figure 1: Collaborative support of Mobile Scrum synchronizing with an issue tracking server

There are certain challenges that need to be addressed when developing a mobile application to support Scrum. According to the Agile Manifesto [BBvB⁺01], software development needs to be lightweight and collaborative. Therefore usability and user experience of Mobile Scrum are decisive factors.

Nielsen defines five key attributes of a usable system: learnability, efficiency, memorability, error handling and satisfaction. [NH93] We support these aspects in the following way. Mobile Scrum integrates tightly with the main Scrum workflows and artifacts: Product backlog, sprint planning, taskboard (with daily scrum support) and sprint review. This allows the user to get familiar with the application quickly even if he only has little knowledge of Scrum (learnability). It also avoids overloading of the user interface which leads to an easy to use and productive application (efficiency). Mobile Scrum is developed as a native mobile application and provides well known standard platform controls and interactions like gestures (learnability, memorability, satisfaction). With the "As a user, I want" template for user stories, and with user guides (e.g. during the Daily Scrum Meeting), Mobile Scrum also reduces typical mistakes when performing Scrum (error handling).

Another challenge on mobile devices is the limited space compared to desktop computers. It is important to meet the users expectations with a mobile application that should be used in everyday work-life. Mapping these users expectations is described by Norman and Draper. [ND86] In Mobile Scrum, the user model represents how the user understands and applies the scrum process. The (user) interface model is influenced by the system model, i.e. the implementation of the workflows in Mobile Scrum, and by the design model that represents the understanding of the developers how Mobile Scrum should be

³This server is integrated into the existing environment, e.g. as plugin into JIRA / Greenhopper.

used. Testing Mobile Scrum with real users is important to determine whether design, user and interface models match with each other.

In the following, we describe two typical scenarios for Mobile Scrum. The first one focuses on supporting planned daily scrum meetings. The second one shows a typical mobile usage of the application in an unplanned situation.

4.1 Scenario: Daily scrum with a flat screen

By offering guided steps and visualization of current issues during the daily scrum, common mistakes - described in section 2 - can be prevented. At the beginning of the meeting, the scrum master Bob opens Mobile Scrum and mirrors his iPad wirelessly to a flat screen⁴ as shown in figure 2.

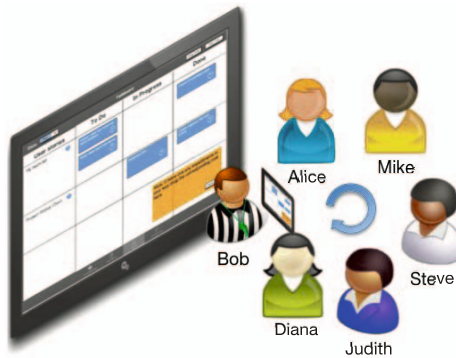


Figure 2: Daily scrum meeting with Mobile Scrum connected to a flat screen

Bob opens the taskboard view in Mobile Scrum and touches the Start Daily button. He confirms that all five developers participate in the meeting and chooses to use the guide because the team is not experienced with Scrum. He turns on the alert option to get notified when one participant takes more than three minutes. This helps the team to avoid long discussions in the fifteen minutes time-box.

The daily scrum starts and Bob passes the iPad to the first developer Alice. The taskboard adjusts automatically, to show all tasks that are assigned to Alice and all yet unassigned tasks. Alice answers the three questions presented to her at the bottom of the screen, starting with what she did since yesterday. While explaining to the team that she finished the *Dashboard View*, she adjusts the taskboard by moving the corresponding item from the *In Progress* to the *Done* column. Then she promises that she will work on the task *My Team Filter* until tomorrow and drags the task from the *To Do* to the *In Progress* column.

Finally, she reports an impediment that blocks her from solving another task. She drags the task *Query online users from Server* to the impediments area on the right bottom of

⁴This flat screen replaces the traditional physical taskboard that shows the progress of the sprint.

the screen and says "Server hardware seems to be broken". Mobile Scrum recognizes her voice and records the impediment. Then Alice passes the iPad to the next participant Mike, whose turn it is now. This procedure is repeated until every team member has reported his status. Then the daily scrum is automatically finished and Mobile Scrum shows the final taskboard.

After the meeting, Bob solves the technical issue reported by Alice by installing a new hardware component. Then he selects it on the taskboard and marks it as solved. Alice is notified and can continue to work on the server connection task.

4.2 Scenario: Mobile Scrum on the go

The product owner Steve is walking home from work through the park. He suddenly has an idea of a feature for the product. He opens Mobile Scrum on his smartphone. The application recognizes by using GPS information that Steve is moving and activates voice commands. Steve says "New feature Dashboard for next release". The application adds the new feature to the product backlog into the section for the next release and sets the title. Steve can now provide additional information. He says "High priority" and the application adjusts the priority of the feature. The new feature is automatically synchronized to the server. The scrum master Bob, who is preparing the sprint planning meeting for tomorrow, is notified about the new feature on his mobile device.

5 Conclusion

In this paper we discussed whether mobile applications can support software development by investigating mobile scenarios in Scrum. We introduced the idea of a mobile, native and integrated application, which we call Mobile Scrum. Existing scrum applications are mostly web based and not implemented natively for mobile devices which leads to usability problems. With a mobile application, it is possible to access information anytime and anywhere. This leads to new mobile scenarios and further decreases the need for desktop computer to access important project information.

Mobile Scrum processes information and data changes automatically in the background and provides software developers an easy and consistent access to relevant information including progress and impediments. The application is designed to guide inexperienced teams through activities like daily scrum meetings step by step. This helps to avoid typical mistakes that often occur when performing Scrum.

Our goal is to implement a first prototype of Mobile Scrum in order to conduct usability tests with users applying it in their work activities. We plan to perform a case study to analyze the usage of Mobile Scrum in real working environments and we want to show that software development processes can be improved with integrated mobile applications like Mobile Scrum. Furthermore we want to investigate whether Mobile Scrum supports distributed teams in which communication problems occur more often. [SVBP07]

The mobile impact on Scrum itself is another interesting research topic. We think that discussions in small unplanned and informal meetings (e.g. at the coffee machine) can now be minuted to increase the information sharing within the team. The question arises if Mobile Scrum can be further enhanced to meet the needs of distributed mobile users. Remote daily scrums might be possible in an easier way if Mobile Scrum additionally offers video conferences in the future. It also might improve the communication of part time developers, if they cannot meet each day.

References

- [BBvB⁺01] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. Manifesto for Agile Software Development. 2001. <http://agilemanifesto.org>.
- [Coh06] M. Cohn. *Agile estimating and planning*. Prentice Hall, 2006.
- [Coh08] M. Cohn. Advantages of the "As a user, I want" user story template, 2008. <http://www.mountangoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template>.
- [Coh09] M. Cohn. *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional, 2009.
- [Kni08] H. Kniberg. 10 ways to screw up with Scrum and XP. 2008.
- [Mob12] MobiThinking. Global mobile statistics - Part A: Mobile subscribers; handset market share; mobile operators, 2012. <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a>.
- [ND86] D.A. Norman and S.W. Draper. *User centered system design; new perspectives on human-computer interaction*. L. Erlbaum Associates Inc., 1986.
- [NH93] J. Nielsen and J.A.T. Hackos. *Usability Engineering*. Academic press San Diego, 1993.
- [Rei96] S Reiss. Software tools and environments. *ACM Computing Surveys*, 1996.
- [S⁺95] K. Schwaber et al. Scrum development process. In *Proceedings of the Workshop on Business Object Design and Implementation at OOPSLA*, 1995.
- [Sut93] J. Sutherland. 7 Ways to Fail with Scrum! 1993.
- [SVBP07] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov. Distributed Scrum: Agile Project Management with Outsourced Development Teams. In *HICSS*, 2007.
- [TMdH⁺12] N. Tillmann, M. Moskal, J. de Halleux, M. Fahndrich, and S. Burckhardt. TouchDevelop: App Development on Mobile Devices. 2012.
- [TN86] H. Takeuchi and I. Nonaka. The new new product development game. *Harvard business review*, 1986.
- [Wei99] Mark Weiser. The computer for the 21st century. *Sigmobile Mob. Comput. Commun.*, 1999.
- [YP04] J. York and P. Pendharkar. Human-computer interaction issues for mobile computing in a variable work context. *International Journal of Human-Computer Studies*, 2004.

