

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/270684220>

User Feedback in Mobile Development

Conference Paper · October 2014

DOI: 10.1145/2688412.2688420

CITATIONS

17

READS

887

2 authors:



Stephan Krusche

Technische Universität München

76 PUBLICATIONS 976 CITATIONS

SEE PROFILE



Bernd Bruegge

Technische Universität München

311 PUBLICATIONS 5,396 CITATIONS

SEE PROFILE

User Feedback in Mobile Development

Stephan Krusche

Technische Universität München
krusche@in.tum.de

Bernd Bruegge

Technische Universität München
bruegge@in.tum.de

Abstract

Developers need to obtain feedback early to build applications that fit to the users needs. In this paper we show how the combination of two approaches enables developers to continuously improve usability and user experience of mobile applications. The Tornado model is a light-weight scenario-based approach for producing executable prototypes. Rugby, an agile process model based on Scrum, allows the developer to continuously deliver these prototypes at any time during a sprint to obtain feedback.

Categories and Subject Descriptors D.2.9 [Software Engineering]: Management—Programming teams, Software configuration management, Software process models

Keywords Agile Methods, Release Management, Feedback, Continuous Delivery, Executable Prototypes, Communication Models, User Involvement

1. Introduction

With the incorporation of agile methods, software engineering has gained more flexibility, efficiency and speed. Feedback plays a more and more important role in the development process. Teams need to incorporate the opinions of end users as early as possible to build great applications. Mobile applications usually require high usability and user experience and developer need to perform usability testing to check the usefulness of their application.

2. Tornado

Tornado is a light-weight scenario-based design approach that emphasizes the use of informal models for the interaction between developers and users. [1] It is used in innovation projects where problem statements are formulated as

visionary scenarios and where requirements and technologies change often. In such cases, clients typically want developers to explore multiple ideas before they decide how vague requirements are realized. The Tornado process starts with visionary scenarios (wide in analysis) funneling down to demo scenarios (narrow in implementation) and relies on early and regular delivery of executable prototypes (touch-points) as well as user feedback (updrafts), see Fig. 1.

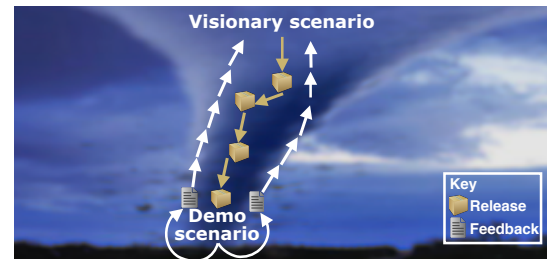


Figure 1. Tornado model (adapted from [1])

To get an early grasp of the user model, developers focus on low fidelity prototypes. Researcher (e.g. [6]) have shown that unpolished user interfaces receive more feedback from end users than polished ones. An example for the evolution from rough sketches (left) over low-fidelity prototypes to the delivered application (right) is shown in Fig. 2.



Figure 2. Exemplary evolution of the user interface [2]

Low fidelity prototypes focus on getting feedback about the user interface as early as possible. They are cheap to produce, easy to change and allow the rapid production of alternatives enabling the client to explore design possibilities and reformulate the initial requirements. [6] Using executable prototypes, developers are able to test the usability of the application with end users early in the design process.

Based on these tests they receive more feedback than they would receive when using the final, polished user interface. The feedback helps to adapt the visionary scenarios.

3. Rugby

Rugby is an agile process model based on Scrum [7] and the Unified Process [3].¹ Self-organizing teams develop iteratively in project-based organizations and apply two new workflows, release and feedback management. Fig. 3 shows Rugby's release management workflow that starts each time a *Developer* pushes source code to the *Version Control Server* (1), leading to a new build in the *Continuous Integration Server* (2) that informs the *Developer* about the build status (3). The *Release Manager* releases successful builds (4) that are uploaded to the *Continuous Delivery Server* (5), which notifies the *User* about the new release (6). The *User* downloads the release (7) onto his mobile *Device* and provides feedback (8) that is automatically uploaded (9) and forwarded to the *Issue Tracker* (10), which notifies the *Developer* (11).

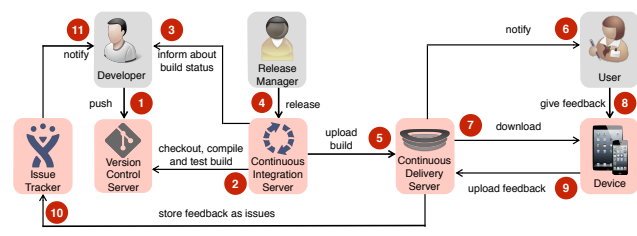


Figure 3. Rugby's continuous delivery workflow [4]

Rugby uses executable prototypes as basis for communication between developers and users. Developers can hardly discuss user interface issues without an executable prototype running on a mobile device. Therefore, Rugby allows releases during sprints, whenever feedback is required or when a client requests it. Fig. 4 illustrates four types of releases supported by Rugby.

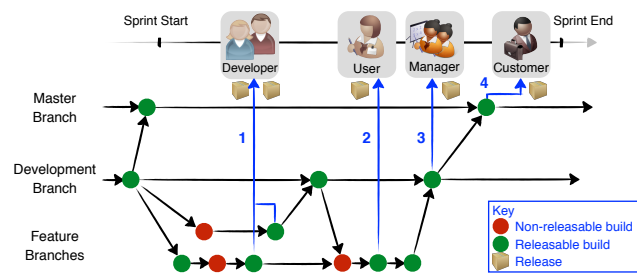


Figure 4. Event-based delivery in Rugby [4]

Developers use feature branch releases in meetings to demonstrate development status to their team members (1). This improves the quality of the communication in meetings, in particular, it shortens the time required to discuss implementation details. They can also use releases from feature

¹ The term rugby was first used by Takeuchi and Nonaka [8].

branches to obtain feedback from users about the development status of the feature (2). Managers can receive releases from the development branch to track finished features (3). Master branch releases are time-based at the end of a sprint, similar to Scrum's product increments, and are automatically produced for sprint review meetings (4). Branches increase the flexibility, because developers can use internal releases to test software and promote releases to external users.

Rugby includes a feedback management workflow. Depending on the change request, developers initiate different workflows. Fig. 5 shows four usage scenarios to deal with user feedback. Developers categorize each feedback according to its type and handle it in different workflows: feature requests in the analysis workflow, design requests in the design workflow (F3) and bug reports in the implementation workflow (F2). During a sprint, they can also decide to resist feedback and move it to the product backlog (F1).

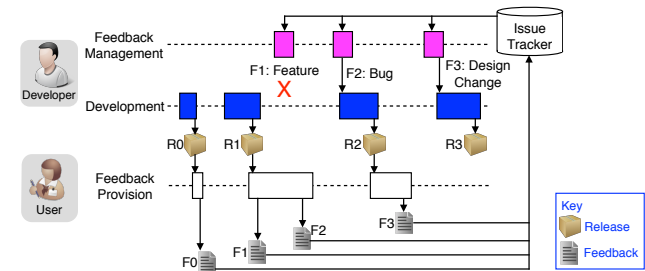


Figure 5. Rugby's feedback workflow (adapted from [5])

4. Conclusion

In this paper we described how two approaches, Rugby and Tornado, help software developers to obtain feedback early and continuously. Feedback is especially important in the development of mobile applications, where usability and user experience play an important role.

References

- [1] B. Bruegge, S. Krusche, and M. Wagner. Teaching Tornado. In *EduSymp*. ACM, 2012.
- [2] D. Dzvoniar, S. Krusche, and L. Alperowitz. Real Projects with Informal Models. In *EduSymp*, 2014.
- [3] P. Kruchten. *The Rational Unified Process*. Addison-Wesley, 2004.
- [4] S. Krusche and L. Alperowitz. Introduction of Continuous Delivery in Multi-Customer Project Courses. In *ICSE*, 2014.
- [5] S. Krusche, L. Alperowitz, B. Bruegge, and M. Wagner. Rugby: An Agile Process Model Based on Continuous Delivery. In *RCoSE*, 2014.
- [6] J. Rudd, K. Stern, and S. Isensee. Low vs. high-fidelity prototyping debate. *Interactions*, 1996.
- [7] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002.
- [8] H. Takeuchi and I. Nonaka. The new new product development game. *Harvard business review*, 1986.