Enabling Focused Software Quality Assurance in Agile Software Development Processes for Mobile Applications using Text and Usage Mining Methods

Michael Bauer[2], Kateryna Sergieieva[1], Gerrit Meixner[1], Konstantin Holl[2] and Frank Elberzhager[2]
[1] *Usability and Interaction Technology Laboratory (UniTyLab), Heilbronn University,*
Max-Planck-Straße 39, 74081 Heilbronn, Germany
[2] *Fraunhofer Institute for Experimental Software Engineering (IESE), Information Systems Quality Assurance,*
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

Keywords: Software Maintenance Tools, Agile Software Development, Data Mining, Data Analytics, Mobile Computing.

Abstract: The acceptance of mobile applications that support services, such as navigation guidance or travel management, is highly dependent on the set of features implemented and the application quality. Information about the acceptance can be gained quickly by collecting user feedback like explicit textual reviews given by the application users or their implicitly given usage data. Considering an approach that bases on developing a minimal set of functions in order to realize a minimal viable product (MVP), it is possible to place a product with a short time to market. In this paper the Opti4Apps approach is presented. It aims at a focused quality assurance as part of the MVP development, which enables and expands the benefits of MVP by providing a semi-automated feedback elicitation, analysis and processing framework. The Opti4Apps process adjusts to mobile development habits including an additional automated feedback and usage data processing infrastructure.

1 INTRODUCTION

Mobile applications have attained an increasing importance in daily use. In 2015, about 1640 million smart phones and tablets as opposed to about 276 million notebooks and desktop computers were sold (Yahoo, 2016). Among all smart phone users e.g. in the U.S. in 2013, 62 percent used mobile application and among them 74 percent used productivity applications (Nielsen, 2013). The sales numbers of mobile devices and the possibility to reach many users lead to more software development service providers entering the mobile application market. Scrum[1] is widely adopted as an agile software development process but there has been a noticeable shift in attention toward lean software development although these methods have much in common, including short iterative life cycles, quick and frequent feedback from customers, and constant learning (Wang et al., 2012). In lean software development the use of a MVP is propagated. A MVP is a product with just enough features and minimal characteristics to gather validated learning in continued development (Robinson, 2016).

The posing challenges are how to collect and process costumer respectively user feedback inside a continuous delivery (CD) environment and how to gain knowledge to enhance the quality of mobile applications including usability issues of future developments without the enlargement of delivery times. To address this, a focused quality assurance approach is needed which supports the MVP-based development of mobile applications, e.g. budget restrictions and short development periods. Hence, we aim to establish an optimization for mobile application (short: Opti4Apps) tailored to MVP-development using a semi-automated framework. The framework consists of feedback and usage elicitation, followed by analysing the data and load processing the analysis outcome.

2 STATE OF THE ART

CD defines a deployment pipeline which enables lean inadvertent error prevention (Fitzgerald and Stol, 2014) (Humble et al., 2006). This deployment

[1] https://www.scrum.org

Bauer M., Sergieieva K. and Meixner G.
Enabling Focused Software Quality Assurance in Agile Software Development Processes for Mobile Applications using Text and Usage Mining Methods.
DOI: 10.5220/0006224801280132
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 128-132
ISBN: 978-989-758-229-5

pipeline comprises visibility including all aspects of the delivery system like building, deploying, testing, and releasing which are visible to every member of the team to promote collaboration. This supplies feedback as team members learn of problems when they occur so that they are able to fix them as quickly as possible. It includes permanent deployment through a fully automated process to install and release any version of the software to any environment (Duvall, 2012). Beyond CD, the principle of development and operations (DevOps) is a development methodology that bridges the gap between software development and IT operations, emphasizes collaboration, also emphasizes continuous delivery and quality assurance (Jabbari et al., 2016). Lean software development defines seven principles (Poppendieck and Poppendieck, 2003):

- *Eliminate waste*: Everything not adding value to the customer is waste and has to be eliminated.
- *Amplify learning*: Software development based on iterations empowers a continuous learning process.
- *Decide as late as possible*: Delaying decisions as much as possible until they can be made based on facts.
- *Deliver as fast as possible*: A sooner delivery of the product without major defects enables immediate feedback which can be incorporated into the next iteration.
- *Empower the team*: The development team takes care of what actions might be taken and suggests improvements.
- *Build integrity in*: The product must not look like a collection of small solutions, but as one integral system.
- *See the whole*: Not sub-optimizing to get the details right, but considering the system as a whole empowers to adapt to project changes.

The paradigms of lean development, DevOps and CD depend on fast deployment to gain knowledge about the product. Therefore automation and constant monitoring of the deployment pipeline is required (Hasselbring, 2015). Regarding automating the deployment pipeline many tools already exist. Jenkins[2] can be used for automated builds, unit- and acceptance tests, static and dynamic code analysis by using a common code basis from a version control system (VCS) like Git[3]. JMeter[4] can be used for load and performance tests. Puppet[5] is a DevOps tool, that treats infrastructure as code (IaC), collaboration between development, operation, information security and networking. Beside this, web analytic tools like Google Analytics[6] and Piwik[7] are also used in mobile applications.

Despite of all those automation tools, some tasks are still accomplished manually, especially in the area of usability. The ISO 9241-210 defines a human-centred design (HCD) process where the user is involved in every step of the process and provides feedback constantly (ISO, 2011). Common methods are used during this process, e.g. interviews, heuristic evaluation, usability tests, and are defined in ISO/TR 16982 and are executed by usability experts manually (ISO, 2002). Feedback can be elicited with applications like iRequire (Seyff et al., 2010) and ConTexter (Gärtner and Schneider, 2012) where large groups of users are able to describe requirements and requests. Providers like UserVoice[8] combine the feedback of an application with a bug tracker. These solutions assume that stakeholders have the intrinsic motivation to contribute their problem descriptions. Appsee's[9] mobile analytics platform provides in-depth analysis of users' behavior. It provides dwell times on screens, as well as touch heat maps, recordings of the usage, navigation diagrams and tracking of clicks and other events. This usage data can be acquired in a similar way to classify usability and failure patterns. Based on the classification, a focused quality assurance method for mobile applications increases the effectiveness of finding failures of mobile applications during testing (Holl and Vieira, 2015).

3 VISION

The Opti4Apps approach is based on a semi-/automated elicitation and analysis of mobile application data. On the one hand data is collected within the mobile device, e.g. from sensors, user interactions, or device-resident sources like log files. On the other hand data is collected from software stores, e.g. Google Play Store[10] or App Store[11], and user forums where explicit feedback is given. The idea is to identify failure patterns using data mining and natural language processing methods in a mobile application.

[2] https://jenkins.io

[3] https://git-scm.com

[4] http://jmeter.apache.org

[5] https://puppet.com

[6] https://www.google.de/analytics/index.html

[7] https://piwik.org

[8] https://www.uservoice.com/

[9] https://www.appsee.com/

[10] https://play.google.com/store

[11] https://itunes.apple.com

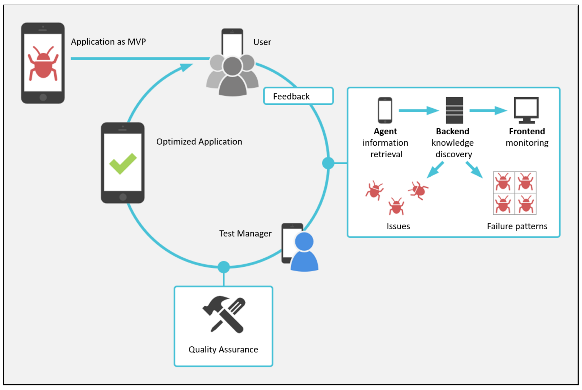

Figure 1: Overview of the Opti4Apps approach.

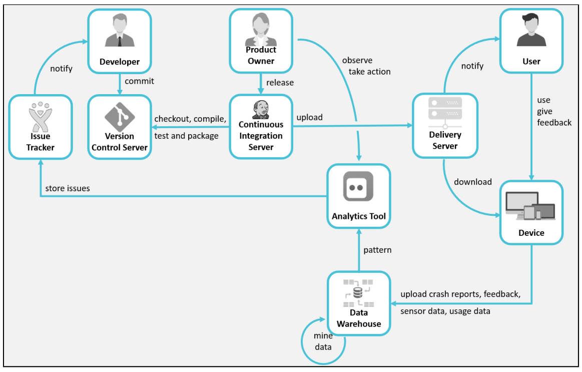

Figure 2: Overview of the Opti4Apps process.

Analysing these patterns is intended to speed the development process, allowing problems to be identified and rectified quicker. Thus, it can provide a baseline for the effective further development as well as the focused quality assurance. This can mean using the new knowledge to guide the test manager during the inspection of the requirements specification as well as the testing of the mobile application. It can also mean identifying user groups that have problems with a specific task within the application. This can be useful information for a usability engineer. Figure 1 shows an overview of the Opti4Apps approach. In the first iteration the mobile application is delivered to the users as a MVP. An agent collects feedback and sends it to the Opti4Apps backend. The frontend presents the processed information. The test manager uses this information to plan and execute quality assurance tasks. After that an optimized version of the mobile application is delivered to the users again.

The recognition of the Opti4Apps depends on the degree it is encouraging mobile development habits. It is also important not to enlarge the tasks of the development teams with additional configuration overhead. That is why the knowledge discovery has to be automated. As already mentioned in the introduction, agile development is widely adopted. Figure 2 describes a process that encapsulates an automated feedback cycle and knowledge discovery infrastructure for mobile applications. Following the DevOps methodology it should be provided as IaC. It is based on a CD approach presented in the research of Krusche and Alperowitz (Krusche and Alperowitz, 2014). In software development, requirements are stored in an issue tracker like JIRA[12]. The developers who are responsible for the requirements write code and commit it to a VCS like Git. A continuous integration server performs checkouts of the latest version and runs automated builds, tests, and generates reports. A product owner releases a version to a delivery server. This can either be software repositories like the Google Play Store, the App Store or self-hosted repositories. Sometimes services like TestFlight[13] and HockeyApp[14] are used for beta testing purposes. The user downloads the mobile application to his device from the delivery server. The usage data is collected by an agent attached to the mobile application. It is spatiotemporal time series data containing a set of measured points. The measurements can be divided in device specific data like the android version, sensor data like GPS location, and application specific data like event names. All that data is stored in a data warehouse. Clickstreams are extracted from the usage data and used for classification and clustering of tasks and user groups. Data from the development tools is also captured in the data warehouse to ensure traceability of a failure pattern to issues, versions, or even development branches. A combination of data mining methods like outlier analysis, clustering, and classification are used to identify failure and usage patterns. In addition to that explicit feedback from users is scraped from the delivery server respectively the Google Play Store, the App Store, and user forums. Natural language processing methods like topic models are used to identify additional failure patterns. If e.g. a group of words like failure, problem, etc. occurs often with words like network a resulting topic may indicate network problems. The frontend of the approach is an analytics tool which is observed by the product owner. Based on the analytics, the product owner takes action for the next iteration. The analytics tool helps the product owner to make decisions based on data and consult the right expert, e.g. a usability engineer or a test manager. Findings can also be stored to the issue tracker for improvements in the next iteration of the development.

[12] https://de.atlassian.com/software/jira/agile

[13] https://developer.apple.com/testflight

[14] https://hockeyapp.net/

4 CONCLUSIONS

Opti4Apps implements and extends the benefits of an MVP development by developing and using a framework based on the automatic elicitation and analysis of feedback as well as using an effective and efficient quality assurance methodology. In contrast to classical development this enables quicker delivery times and early feedback from users. With Opti4Apps this feedback is automatically processed in a systematic way to identify failure patterns. Those patterns define connections of typical failure causes with their common impacts. In addition to that using cluster analysis of failure patterns the approach is capable of finding user groups with similar problems. In future work finding user groups and segmentation of user tasks by analyzing clickstream data will be done. One challenge here is to label and to distinguish tasks. To solve this it is planned to use data fusion between development tools and clickstream data to interconnect e.g. issue, class, and function names with user interaction event data collected from an agent inside the mobile. Insights about efficiency and effectiveness on tasks also allow conclusions about the usability of the mobile application. Another challenge is to identify user groups based on the clickstream data when no other user data is available. A promising idea is to group users based on behavioral pattens within the application. In parallel developing a concept to protect the privacy of end users is mandatory. With Opti4Apps, lean principles can be applied to the software development of mobile applications. The framework supports amplified learning by iterations including immediate and structured feedback. The outlined process makes fast deliveries possible and enables late decisions based on reasonable data. Concerning this, the elicitation of relevant information with respect to the user's behavior should be automated as much as possible. Further more, the feedback and insights from one development can be reused in parallel or subsequent development.

ACKNOWLEDGEMENTS

The research described in this paper was performed in the project Opti4Apps[15] - Optimization for mobile applications (grant no. 01FJ15133) of the German Federal Ministry of Education and Research.

[15] http://www.opti4apps.de

REFERENCES

Duvall (2012). Continuous delivery. http://www.dccia.ua.es/dccia/inf/asignaturas/MADS/lecturas/10_Continuous_Delivery_Dzone_Refcardz.pdf. Accessed on 2016-11-7.

Fitzgerald, B. and Stol, K.-J. (2014). Continuous software engineering and beyond: Trends and challenges. In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, RCoSE 2014, pages 1–9, New York, NY, USA. ACM.

Gärtner, S. and Schneider, K. (2012). A Method for Prioritizing End-user Feedback for Requirements Engineering. In *Proceedings of the 5th International Workshop on Co-operative and Human Aspects of Software Engineering*, CHASE '12, pages 47–49, Piscataway, NJ, USA. IEEE Press.

Hasselbring (2015). Devops: Softwarearchitektur an der schnittstelle zwischen entwicklung und betrieb. In *GI-Fachtagung Architekturen 2015*.

Holl, K. and Vieira, V. (2015). Focused quality assurance of mobile applications: Evaluation of a failure pattern classification. In *41st Euromicro Conference on Software Engineering and Advanced Applications, EUROMICRO-SEAA 2015, Madeira, Portugal, August 26-28, 2015*, pages 349–356.

Humble, J., Read, C., and North, D. (2006). The deployment production line. In *Proceedings of the Conference on AGILE 2006*, AGILE '06, pages 113–118, Washington, DC, USA. IEEE Computer Society.

ISO (2002). *Ergonomics of human-system interaction - Usability methods supporting human-centred design; ISO/TR 16982:2002*. ISO International Organization for Standardization.

ISO (2011). *Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems; ISO 9241-210:2010*. ISO International Organization for Standardization.

Jabbari, R., bin Ali, N., Petersen, K., and Tanveer, B. (2016). What is devops?: A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016*, XP '16 Workshops, pages 12:1–12:11, New York, NY, USA. ACM.

Krusche, S. and Alperowitz, L. (2014). Introduction of continuous delivery in multi-customer project courses. In *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, pages 335–343, New York, NY, USA. ACM.

Nielsen (2013). The mobile consumer: A global snapshot. http://www.nielsen.com/content/dam/corporate/uk/en/documents/Mobile-Consumer-Report-2013.pdf. Accessed on 2016-11-02.

Poppendieck, M. and Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley.

Robinson, F. (2016). Minimum viable product. http://www.syncdev.com/minimum-viable-product/. Accessed on 2016-11-02.

Seyff, N., Graf, F., and Maiden, N. (2010). Using mobile RE tools to give end-users their own voice. In

Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE2010, pages 37–46, Sydney, Australia.

Wang, X., Conboy, K., and Cawley, O. (2012). Leagile software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6):1287–1299.

Yahoo (2016). Statista - Das Statistik-Portal: Prognose zum absatz von tablets, pcs und smartphones weltweit von 2010 bis 2020 (in millionen stück). https://de.statista.com/statistik/daten/studie/256337/umfrage/prognose-zum-weltweiten-absatz-von-tablets-pcs-und-smartphones. Accessed on 2016-11-02.