On the Delicate Balance between RE and Testing

Experiences from a Large Company

Krzysztof Wnuk
Department of Computer Science,
Lund University, Sweden
Krzysztof.Wnuk@cs.lth.se

Linus Ahlberg, Johannes Persson
Department of Computer Science,
Lund University, Sweden
ain09lah@student.lu.se, ain09jpe@student.lu.se

***Abstract*—Requirements Engineering and Testing are considered inseparable and together enable successful projects. One could make a statement that serious negligence of any of the two will bring catastrophic consequences for the project, delays, failures and unhappy customers. Thus, alignment and cooperation between the two are not only recommended but highly required for contemporary complex software and system development. In this paper, we share some experiences from a large company that significantly downplayed requirements engineering activities while shifting the focus to testing and QA activities. Despite that, the company's products are well received by its customers and the company remains profitable and significantly growing. We share our experiences and discuss how this unbalance may actually be a recipe for success and what contextual factors enable using this approach.**

I. Introduction

Requirements Engineering (RE) and Testing are considered inseparable and together enable successful project development and deployment. Both competences should be activated as early as possible and continuously support software development activities until and even beyond the successful project launch. This tandem offers not only the sense of security that the system to be built is meeting the customers' needs and provides sufficient quality, but also minimizes the risks of failures caused by the development organization going off course. Thus, it appears logical that as much effort as possible should be dedicated to supporting the RE and Testing tandem. Proper alignment is particularly important when scaling up software development activities or processes.

The lack of alignment can cause serious issues and challenges. Bjarneson et al. listed defining clear, verifiable and complete requirements, keeping requirements documents updated and defining well matched requirements for the abstraction level. Traceability between requirements and test cases was also listed as one of the alignment challenges [1]. Uusitalo et al. investigated the alignment practices and concluded that early tester participation, traceability policies, consider feature requests from testers, and linking test and requirements people to the produced artifacts are highly valuable [2]. Kukkanen et al. reported from a case study that early definition and assignment of RE and testing roles, augmented by connecting processes and people and integrating processes are among the key alignment activities. The key for the alignment seems to lie in ensuring that 'the right information is communicated to the right persons'[3]. Interestingly, none of the above discuss the results of downplaying any of the two; instead each paper suggest increased efforts in both. This increased effort in both may be the main obstacle in achieving effective scalability of the development processes as companies have limited resources that can be dedicated for the process work. In this paper, we share some experiences from a large company that significantly downplayed requirements engineering activities while shifting the focus to testing and QA activities. Despite that, the company's products are well received by its customers and the company remains profitable and significantly growing.

This paper is structured as follows: Section II provides the company context while Section III discusses our experiences from exploring the company's requirements process and its relation to testing and QA activities. Section IV concludes the paper and provides hypotheses based on the summarized experiences.

II. The Investigated Context

The case company develops and manufactures embedded systems for computer networking and has been growing extensively in recent years, reaching the global market leader position. This fact is important from the analysis perspective since the company downplayed requirements engineering and management and focused on testing and QA as surrogates. The company has 1800 employees, from which a few hundred are involved in embedded software development.

The software is developed towards an open platform, which causes the company to prioritize clean and stable APIs for its users. The company builds its product based on the SPLE approach with high degree of code reused between the *platform* and the variants, configured from the platform. Thus, the platform can satisfy multiple customer needs via multiple product types.

The two main development organizations are the product development and the platform organizations. The product development organization uses the functionality available on the platform when developing products, striving to minimize the amount of software changes done before integrating the software back to the platform. Any changes to the *platform* that should be integrated into the platform are integrated after

978-1-4799-6333-1/14/$31.00 © 2014 IEEE

the product release. The products are released after the features are implemented and extensively tested by the QA organization.

The main focus on this paper is the platform organization that consists of 125-150 people allocated to product management, development and QA teams. The product management teams are responsible for channeling the market needs into requirements. The teams create roadmaps where ideas resulting from the business strategy discussions are brought up and prioritized. From the prioritized items, feature descriptions are created and handed over to the development organization for further refinement and development. Feature descriptions are created for both functionality and architectural enhancements.

The development teams are responsible for developing and maintaining the software platform according to the received high level requirements and performing unit testing. During the feature development, these teams create project specific documentation and detailed requirements but these requirements are not stored or systematically maintained. The QA teams are responsible for testing the developed functionality on the software platform both in terms of new features testing and regression testing. Each development project has at least one assigned QA resources participate in the project's activities and write system level test cases for the output of the developers.

Different teams are responsible for different functional areas and each team mainly conducts projects in their own area. The areas are for the most part architecturally separated in order to limit the number of teams that are affected by a project. One project manager, one technical lead and QA resources are assigned to each project, carrying the responsibility of leading the development work.

The company uses agile methods for software development. However, the projects in platform development follow predefined guidelines for the development process and documentation, but are allowed to do changes to the process if the changes are approved by all project stakeholders. This combination of agility and restrictiveness results in different teams working always slightly differently. Code owners and architects share code responsibility. The architects and the code owners that are tied to one functional team.

The culture at the company leans towards spreading knowledge about requirements verbally rather than through specific requirements documents. The company values an open climate and encourages its employees to be communicative. Informal communication is promoted, e.g. making it culturally accepted to ask questions and have discussions without booking formal meetings. Thus, much knowledge is shared through discussions, even between departments, and helpfulness and team spirit are emphasized.

The company uses several type of documents for recording the project requirements, progress and decision making results. However, the amount of requirements documentation is minimal compared to generally accepted recommendations and guidelines. QA and testing personal gets early involved into the requirements activities and discussions and help to detail them. When the project is over, the project documentation maintenance is discontinued and the company focuses on maintaining test cases and reference documentation containing descriptions for design and functionality.

We have interviewed 16 practitioners from this company and summarized our experience regarding the balance between RE and testing in the section that follows.

III. Experiences

Compared to the state of the literature and practice and current recommendations about requirements engineering practice, the company performs little requirements engineering and documents even less. Interestingly, the interviewees surveyed in this work were happy about the current requirements engineering process and considered it scalable and sufficient. The interviewed QA experts saw little use in looking at the requirements documentation as often the final system diverges from the view documented in the requirements. Several interviewees from QA expressed that they think their test cases can be seen as a representation of the requirements. A senior tester agitated for focusing on test cases instead of requirements documentation. The interviewed developers argued that, in reality, the tests developed by QA constitute the requirements and these developers seemed content with the current approach of finding requirements in the organization. Both QA and developers gain the knowledge about requirements and functionality via extensive communication between each other and consulting the available, often incomplete documentation.

It appears that four aspects contributed to the success of this approach. Firstly, requirements knowledge is acquired through other than requirements documents, tests and colleagues who may have more experience in the matter at hand. Secondly, the feature descriptions are high level and focus on the problem that need to be solved rather than how it should be solved. This gives implementation flexibility. Thirdly, the company uses benchmarking and prototyping to get quick feedback on the implementation suggestions. Fourthly, the culture at the company focuses on openness as well as informal and direct communication, where face-to-face communication is emphasized. In practice, this is seen through an open climate at the company, where asking questions is always allowed.

The main challenging aspect of this approach is how to handle quality requirements. In this case, performance was the main quality aspect that the products were evaluated against and a lack of clearly stated performance requirements created confusions among the interviewees. Due to embedded nature of the developed products, other quality aspects, e.g. usability or supportability are less important. The performance between products varies greatly, e.g. due to different hardware as well as different functionality setups. Several developers noted that there exists some confusion about what the required level of quality is, while others point to that the reason for having certain levels of quality was unclear.

However, a senior tester from QA mentioned that specifying quality requirements is dangerous, since the quality desired by the customers always keeps increasing. Thus, any old specifications that define "high quality" may actually represent low quality in current terms. This creates a risk that people will interpret those kinds of specifications as the absolute truth, without considering that quality needs constant improvement. Thus, measuring the actual performance was suggested as the more optimal way of managing quality instead of writing quality requirements.

IV. Implications and Conclusions

In this paper, we have shed the light on an interesting aspect that we believe is underrepresented in the literature. In the studied context, the requirements engineering activities and documentation are downplayed and more effort is dedicated towards the testing activities and early QA involvement.

Despite a number of identified challenges, the general performance of the organization as a whole seems to be satisfactory and the quality of the released software seems to be good. The developers working in the customer maintenance department highlighted that the customers are, in general, happy with the quality of the products. We believe that these statements, together with the fact that the company is the market leader in its area are sufficient to start the discussion about the balance between RE and testing. Thus, we advocate that more effort should be dedicated into better understanding of this balance rather than advocating continuous increased effort in both.

Based on our investigation, there seems to be several enabling factors that contributed to the success of this approach. Among them, the results suggest that the open culture, focus on informal and direct communication and more effort dedicated into the testing activities are significant.

Analyzing the results from the scalability perspective, there seem to be interesting implications of the process direction followed by the company. Considering the definition of scalability suggested by Laitinen et al. [3] as a "property of increasing the scope of software engineering methods, processes and management according to the problem size" we assume that enabling scalability often requires simplification to achieve greater process efficiency and effectiveness. Our experiences confirm that assumption since the company decided to skip some parts of the documentation as the cost of storing maintaining and updating that documentation in relation to the potential benefits was too high. Similarly, some highly recommended components of the traditional requirement engineering process are also left out and considered too expensive to be properly carried out.

The widely recommended tool support as a way to achieve efficiency and effectiveness is only one part of the solution. Every tool offers significant process automation but requires staff that can continuously update, maintain and follow up on the tool and process changes. Most growing companies undergo significant process and organizational changes that need to be reflected in the process tools and therefore generate additional costs. Therefore, it appears logical that process simplification could be considered as a possible solution when scaling up the development processes. This simplification provides abstraction and thus could help to keep the focus and the main goals of the processes, leaving the freedom of details implementation. As a result, the organization that is challenged with significant growth should be able to better cope with rapidly changing processes and organizations and focus on delivering good products to the customers.

This focus on product delivery is important in market-driven contexts where meeting market expectations in a timely manner dictates the success of the products. Our work suggest that the correlation between the process quality and products quality is not so obvious and should be revisited and further explored. In particular, our results suggest that it is possible to successfully capture the customer values expressed in a multitude of business scenarios and transfer it into needed business flexibility and a business effective architecture. Early test involvement and flexible processes seem to be important enablers of this transition.

In future work, we plan to further explore the balance between RE and testing. In particular, we are interested in exploring which additional context factors may influence the balance between RE and V&V and in which direction this balance will be shifted. Moreover, we plan to further explore the risks of adapting the described approach. In this paper, we focused primarily on performance but we are aware that more risks or challenges could be uncovered and should be explored. Finally, we plan to focus on exploring the role of the RE and V&V balance in supporting software development scalability for both embedded and software intensive companies.

Acknowledgment

The work is supported by the SCALing softwARE: Supporting Industry in Managing Software Scalability project (SCALARE) under the ITEA2 Call 7 program.

References

[1] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, R. Feldt, "Challenges and practices in aligning requirements with verification and validation: a case study of six companies", Emp. Soft. Eng. vol. 58, Jul. 2013, pp. 1-47.

[2] E. J. Uusitalo, M. Komassi, M., Kauppinen, A.M. Davis "Linking requirements and testing in practice". Proc. of 16th IEEE International Requirements Engineering Conference, 8-12 Sept. 2008, Barcelona, Spain, IEEE Computer Society, pp. 265–270.

[3] J. Kukkanen, K. Vakevainen, M. Kauppinen, E. Uusitalo, "Applying a systematic approach to link requirements and testing: A case study" Proc. of Asia-Pacific Software Engineering Conference (APSEC '09), 1-3 Dec. 2009, Penang, Malaysia, IEEE Computer Society, pp. 482–488.

[4] M. Laitinen, M. E. Fayad, and R. P. Ward, "Thinking objectively: The problem with scalability," Commun. ACM, vol. 43, no. 9, Sep. 2000, pp. 105–107.