Preparing Students for Agile Projects and Requirements Engineering Prioritisation Techniques

Sarita Pais
Whitireia Polytechnic
Sarita.pais@whitireia.ac.nz

ABSTRACT

Requirements engineering is part of software engineering and is taught in combination with Project Management and Systems Analysis and Design papers. With the increase in the number of agile projects, traditional requirements engineering techniques cannot be applied. Moreover, there has been a gap between requirements engineering in literature and its practice in industry. The focus of this paper is to analyse the requirements prioritisation techniques available in the literature and their suitability for agile projects. Empirical studies on agile requirements engineering and prioritisation techniques are analysed to prepare students for capstone projects using agile methods.

Requirements prioritisation is necessary when a software system has various requirements from stakeholders. It is challenging when software development is constrained by limited resources like an inadequate budget or time. With agile projects it is important to understand the prioritisation as there may be new requirements in the software development life cycle. Hence, it is important to know which prioritisation technique can be applied to a pool of requirements and the order of implementation of those requirements.

Keywords: requirements engineering, requirements prioritisation technique, business value, risk, teaching requirement engineering, supervising capstone project

1. INTRODUCTION

The CHAOS Manifesto report (2013) from the Standish Group states that small projects of labour value less than $1 million had a better success rate for agile than the traditional waterfall methodology. Of these, 45% of projects follow agile while 14% still follow traditional waterfall methodology. The overall success rate of projects from 2002 is around 40%, while the other 40% are challenged projects in terms of being over budget, over time and having less than the required functionalities satisfied; around 20% of projects had failed because of cancellation prior to completion. There is an improvement over the original report dated 1994, where only 16.2% of projects were successful; 52.7% were challenged; and 31.1% were failed projects. Most software development in industry today follows agile methods like Scrum, extreme programming (XP), feature driven development, Kanban, Crystal and test driven development. The Agile Manifesto (Beck, 2001) states "Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan". Stakeholders and customers play an important role in the development of the software. A high level project scope and requirements document are planned in the initial meeting, which get revised with every iteration.

The main focus in agile methods is to change and evolve software to meet the customer's expectation. Higher priority requirements get implemented earlier to get the most business value, while ensuring a low risk to the project. Customers come up with requirements of business values, which are validated by developers for risk. In traditional waterfall, prioritisation is completed once, before the start of the project by the development team.

The traditional Requirements Engineering (RE) has been challenged (Ramesh, Cao & Baskerville, 2010) with the introduction of agile projects. There has been a gap between RE in literature and its practice in industry (Pais, Talbot & Connor, 2009). There are over 50 Requirements Prioritisation Techniques (RPT) available in literature (Achimugu, Selamat, Ibrahim, & Mahrin, 2014), but it is unclear whether these RPT are suitable for agile projects. The literature was reviewed for agile RE and the suitability of any RPT for agile projects.

Students studying capstone projects have limited or no previous work experience. Text books are not practical and hence class room teaching had to incorporate practical activities in preparing students in RE and RPT.

How suitable are the different requirements prioritisation techniques for agile projects?

How can students be supported with requirements engineering and requirements prioritisation techniques in preparation for capstone projects in an undergraduate Information Technology programme?

2. RESEARCH METHODOLOGY

The literature was searched for suitable studies in RE and RPT over the last five years. The "agile" keyword was added into the search along with the expanded form of RE and RPT. The results revealed that there were not many papers available in agile RE, although there was a large amount of literature for traditional RE.

The purpose of this research study was to identify existing RPT suitable for agile projects. The literature related to industry practices for agile projects were reviewed. This helped in choosing some of the RPT for undergraduate capstone projects.

Qualitative analysis of secondary literature was used as the research methodology.

3. LITERATURE REVIEW

Traditional RE practices, like interviews and focus meetings are the main components in agile projects. The continuous prioritisation of requirements by customers is part of agile methods (Racheva, Daneva, Sikkel, Herrmann & Wieringa,

This quality assured paper appeared at the 6th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2015) and the 28th Annual Conference of the National Advisory Committee on Computing Qualifications, Queenstown, New Zealand, October 6-9, 2015. Michael Verhaart, Amit Sarkar, Rosemarie Tomlinson and Emre Erturk (Eds).

2010). In an empirical study (Cao & Ramesh, 2008), it was found that reprioritisation takes place at each iteration, which is the main difference when compared to traditional RE. Continuous reprioritisation may lead to unstable projects when not practised cautiously, and so there should be guidelines to follow some criteria to reprioritise requirements at each iteration.

There are different agile methodologies. One of these methodologies is XP where the customer is part of the team and writes user stories, which are discussed with the development team and developed into prioritised requirements. Planning game prioritisation technique (Beck, 2000) is utilised here, where customers categorise the requirements into three categories: essential, conditional or optional. Business value is the main criterion used by customers to prioritise requirements while risk is checked by developers. Developers are considered the most influential stakeholders in requirements prioritisation processes (Bakalova, Daneva, Herrmann & Wieringa, 2011). Each iteration takes 2-3 weeks, where user stories are developed as requirements for building a working software. The customer tests the software at each iteration (Racheva, Daneva & Buglione, 2008).

In Scrum, the product owner represents all the customers and writes the user stories, ranking and prioritising them. The list of prioritised requirements thus generated are put in to the product backlog. The top priority requirements from the list will be in the sprint backlog to be implemented in each iteration by the development team. The requirements listed by customers are re-prioritised with each iteration. However, there is no specific prioritisation technique described in this method.

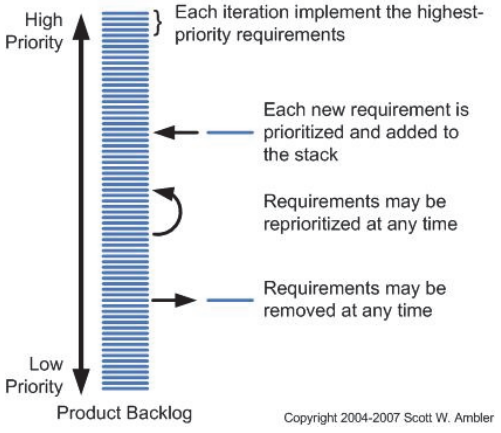

Figure 1: Scrum Product Backlog (Agile Best Practice: Prioritized Requirements)

Documentation is kept to a minimum and the focus is on face to face meetings. The challenge here is that if some members of the team quit, then it becomes difficult to continue with the project smoothly.

In all these agile methods, the requirements are prioritised by customers and checked by software developers. The traditional RE process is not defined. Many studies are undertaken to check if any requirements prioritisation techniques are utilised in industry is discussed in Section 3.2.

3.1 Requirements Prioritisation Techniques in the Literature

Achimugu, Selamat, Ibrahim, & Mahrin (2014) conducted a systematic literature review of the most cited RPT found in the literature. Analytical Hierarchy Process (AHP) was the most cited technique in the literature. AHP is a pair-wise comparison of all requirements. While theoretically it proved to have high accuracy, it however failed with scalability when the number of requirements was large. In a study by Karlsson et al. (2007) a tool based pair-wise technique was less time consuming than the XP planning game. Most RP techniques listed in the literature were time consuming and hence not suitable to run in every iteration for agile projects.

Some techniques which involved stakeholders like customers, were of interest to this study as it could apply to agile projects. Techniques like different types of voting system, top ten prioritisation were considered. Racheva, Daneva, & Buglione (2008) listed RPT recorded in text books, software consulting organisations, white papers and literature. The techniques were from traditional RE and were proposed suitable for agile projects depending on the time consumption, accuracy and ease of use.

Some of the RPT suitable are listed and explained below:

MosCow stands for requirements to be implemented in a project, by their importance in order of 'Must have', 'Should have', 'Could have' and 'Won't have'. Highly desirable requirements are put in the 'Must have' category while the nice to have requirements are in the 'Won't have' category. This is a technique popularly described in Project Management text book (Schwalbe, 2014) and adopted in Project Management paper.

Planning Game as described earlier in the XP process is of interest as customers categorise the requirements as essential, conditional or optional. Once again scalability is an issue when the project size increases together with many functional requirements.

Round the Group Prioritisation (Berteig, 2006) is where requirements are on flash cards and arranged by stakeholders in the order of preference which could be challenging for projects with a large requirements.

Dot Voting (Gottesdiener, 2007) is a technique where stakeholders place sticky dots against a requirement. The highest priority requirement will have the most dots assigned.

Cumulative Voting or $100 Allocation (Achimugu, Selamat, Ibrahim, & Mahrin, 2014) is a method where each stakeholder is given a fictitious $100 to spend on each requirement. The total money allocation for each requirement is divided by the number of stakeholders to get a weighting scale. This sets the prioritisation value. This is also not suitable for projects with a large number of requirements.

Top ten requirements (Achimugu, Selamat, Ibrahim, & Mahrin, 2014) are listed from a pool of requirements after a brainstorming session with stakeholders. The downside to this approach is that no weighting is given to each ranked requirement.

Value oriented prioritisation (Achimugu, Selamat, Ibrahim, & Mahrin, 2014) is linked to business value which is rated by stakeholders. This could be suitable to agile projects where business value is the key to customer satisfaction. However requirements dependencies are ignored and this approach is also not suitable for large projects.

3.2 Agile Requirements Prioritisation Techniques in Industry

Empirical studies on agile industry projects (Cao & Ramesh, 2008) (Ramesh, Cao & Baskerville, 2010) (Racheva, Daneva, Sikkel, Herrmann & Wieringa, 2010) observed two key differences between traditional and agile RE. Agile RE placed

an emphasis on reprioritisation of requirements in each iteration and on getting the best business value as described by customers. Developers would analyse the requirements of business value and identify technical risk, implementation difficulties and costs.

In the study by Cao & Ramesh (2008), 16 organisations were identified to analyse the agile RE practices in their organisations. Requirements prioritisation was described as 'extreme', the highest priority requirements were implemented early on in the project so that customers could appreciate the most business value.

In a study by Svensson, et al. (2011), it was found that nine of the eleven organisations investigated had prioritised requirements on an ad hoc basis. Quality requirements prioritisation is given less importance by decision makers who do not want to invest time and resources. Once the functional requirements were implemented, then the team would look into the quality requirements for improving the quality of the product overall. Very few organisations had assigned numerical assignment value or pair-wise comparisons.

4. DISCUSSION

RPT described with stakeholders' involvement are suitable for agile projects. Each stakeholder is a valued member of the team and the power balance between customers is maintained through the RPT. No one member can dominate the process or the priorities of the requirements. If there are no defined RPT and requirements are decided on an ad hoc basis, then there is the possibility of the most important requirement not being fulfilled in the initial iteration(s). With project time and cost factor constraints, there is danger of project failure.

In some cases, developers with significant experience may dictate the requirements. RPT like cumulative voting system, need to be utilised to ensure that the most important requirements are not put on the back burner.

In most cases, RPT have scalability problems with projects involving a large number of requirements. However the very nature of agile is to break a large project into small manageable parts, thus reducing the requirements to a point where they can be managed in an efficient way.

With customer involvement, prioritisation techniques like voting systems are manual. Hence, requirements traceability and dependency are difficult to implement.

4.1 RE in Capstone Projects

Students studying their Bachelor's and Graduate Diploma are to undertake a 45 credit capstone project in the last term of their study. Papers like Systems Analysis and Design and Project Management help students in completing the capstone project. Most students get a project from local industry. Students are supervised by an academic supervisor who acts as a mediator between project students and the industry client.

As part of the preparation, class room activities involve RE workshops, like writing user stories for a case study. Role play has been used in the past to teach requirements elicitation and validation, however prioritisation was not part of the activities (Zowghi & Paryani, 2003). Bachelor's students were studying with Graduate Diploma students, which changed the dynamics of the class. Some of the Graduate Diploma students had related work experience in IT and were able to contribute well in the group activities. Prioritisation techniques like the multi user voting system, top ten requirements and priorities with weightings were enacted in role plays. In some cases tutor acted as a customer in the role play. Useful online resources (Mountain Goat) were used in designing workshops for students. This gave students exposure to techniques not described in prescribed text books, but available in reference text books, the literature and online resources. The workshops worked well most of the time. Not all techniques could be used in each course because of the time limit. Students remember the technique better after enacting it, rather than reading textbooks. A few students did not take it seriously enough to appreciate the practical activities. Coincidently those students generally were not doing well in their studies. Initially some were shy, however with each week's activities they became more proactive in their group activities. The activities were time consuming and a time limit was assigned to allow other teaching to take place and the class was concluded with an overall review and feedback from the tutor.

Another issue some students faced while studying their capstone project was that their industry client was busy and not available for meetings. Students had to rely on feedback by email correspondence. In one such project where XP methodology was used, students took the initiative and wrote user stories. Clients validated the user stories in a brainstorm session to form concrete requirements. The XP planning game technique of categorising requirements into essential, conditional or optional, gave priorities to fulfil the essential requirements in the first iteration. What worked better for students was the development of the software as a prototype in iterations. The prototype testing and the reviews were checked for functional and non-functional requirements. If required, new requirements were introduced at each iteration. The regular meetings with the academic supervisor and the three reviews during the term helped students to focus on the requirements and implement them in the software development project.

4.2 The Literature and the Industry Gap

There has been a gap between RE in the literature and real world projects. The tools and techniques developed in research have not been embraced well in industry. Some industry practice, like prototyping have led to theory in Information Systems (Fitzgerlad, 2003). Increased collaboration between academics and industry is projected to be useful to bridge the gap between literature and practice (Pais, Talbot & Connor, 2009). Internship papers offered by the faulty can give students real life work experience.

Although there are close to 50 RPs cited in the literature, not all are successfully used in real world projects. RPT did not work in traditional RE and do not have an uptake in agile projects. Some of the RPT involving customers may be suitable for agile projects, however there is little evidence of it in industry agile projects.

Suitable prioritisation in RE should be used to avoid bias among team members. Business value is not defined well in agile RE and solely depends on customers. Domain experience is very important in understanding a project's requirements. Developers also need related domain experience matured through their work to evaluate risk with the elicited requirements. This will not help graduating students with no work experience, and so it is necessary to introduce practical workshops and role play to make the RE processes as real as possible.

Non-functional requirements are not well defined in agile RE (Ramesh, Cao & Baskerville, 2010), as customers do not focus on this. Developers will have to suggest these and explain their importance to clients.

Only a small number of RPT were analysed in this study. A more comprehensive study is required to identify other RPT and analyse them to check their suitability for agile projects.

5. FUTURE WORK

After the literature review study on agile RE, the next intention is to conduct interviews with industry organisations involving agile projects in New Zealand. The RE and RPT processes will be the main area of interest when analysing the business value and risk in these projects.

Capstone projects have limited exposure to the RE processes in real world projects. The internship paper introduced in the curriculum should give an insight into the RE process in agile projects in industry.

6. ACKNOWLEDGMENTS

I acknowledge the knowledge and experience gained through the requirement engineering paper studied at Master's level. This allowed me to apply it to my teaching, supervising capstone projects and motivated me to think about agile projects.

7. REFERENCES

Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. R. (2014). A systematic literature review of software Requirements prioritization research. *Information and Software Technology*, 56(6), 568-585.

Agile Best Practice: Prioritized Requirements. Retrieved from http://agilemodeling.com/essays/prioritizedRequirements.htm

Bakalova, Z., Daneva, M., Herrmann, A., & Wieringa, R. (2011). Agile Requirements prioritization: What happens in practice and what is described in literature. *In Requirements engineering: Foundation for software quality, Springer Berlin Heidelberg*. 181-195.

Beck, K. (2000). eXtreme Programming Explained: Embrace Change, *Addison Wesley*.

Beck, K. (2001). Agile Manifesto. Retrieved from http://www.agilemanifesto.org/

Berteig, M. (2006). Methods of prioritisation. Retrieved from http://www.agileadvice.com/2006/03/20/agilemanagement/methods-of-prioritization/

Cao, L., & Ramesh, B. (2008). Agile Requirements engineering practices: An empirical study. *IEEE Software*, 25(1), 60-67.

CHAOS Manifesto report 2013. Retrieved from http://www.versionone.com/assets/img/files/ChaosManifesto2013.pdf

Fitzgerald, B. (2003). Informing each other: Bridging the gap between researcher and practitioners. *Informing Science*, 6, 13-19.

Gottesdiener, E. (2007). At a Glance: Other PrioritizationMethods, Retrieved from https://www.ebgconsulting.com/Pubs/Articles/At%20a%20Glance-Other%20Prioritization%20Methods-supplement-EBG%20Consulting.pdf

Karlsson, L., Thelin, T., Regnell, B., Berander, P., & Wohlin, C. (2007). Pair-wise comparisons versus planning game partitioning—experiments on Requirements prioritisation techniques. *Empirical Software Engineering*, 12(1), 3-33.

Mountain Goat Software, Retrieved from https://www.mountaingoatsoftware.com/

Pais, S., Talbot, A. & Connor, A. (2009). Bridging the research-practice gap in requirements engineering, *Bulletin of Applied Computing and Information Technology*, 7(1).

Racheva, Z., Daneva, M., & Buglione, L. (2008). Supporting the dynamic reprioritization of Requirements in agile development of software products. *In Software Product Management, 2008. IWSPM'08. Second International Workshop. IEEE.* 49-58.

Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., & Wieringa, R. (2010). Do we know enough about Requirements prioritization in agile projects: insights from a case study. *In Proceedings of 18th Requirements Engineering Conference (RE), IEEE.* (pp. 147-156).

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile Requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 20(5), 449-480.

Schwalbe, K. (2014). Information Technology Project Management. Course Technology : Cengage Learning

Svensson, R. B., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., & Aurum, A. (2011). Prioritization of quality Requirements: State of practice in eleven companies. *In Proceedings of 19th Requirements Engineering Conference (RE), IEEE*. (pp. 69-78).

Zowghi, D., & Paryani, S. (2003). Teaching Requirements engineering through role playing: Lessons learnt. *In Proceedings of 11th Requirements Engineering Conference. IEEE*. (pp. 233-241).