Requirements Change Management in Agile Software Development

11/07/2023

Presented by: Belen Bonilla

CS 846 – Requirements Engineering

UNIVERSITY OF WATERLOO

Outline

1. Introduction
2. Research Method and Questions
3. Review Approach
4. Results
5. Discussion
6. Conclusion
7. References

Requirement Changes

- Requirement changes refer to additions, modifications, or deletions of requirements at any stage of the software development process (Wiegers & Beatty, 2013).

- According to Carniel and Pegoraro (2018), the most common reasons why requirements change are:
  - ➢ Unidentified needs at the beginning of the project
  - ➢ Changes in the business and customer environment
  - ➢ Correction of errors detected by quality processes
  - ➢ New perspectives from stakeholders

Requirement Changes

Unmanaged or improperly managed requirement changes may have negative consequences on the system's development, such as:

- Cost and schedule overrun
- Endless testing
- Project failure and business loss

Therefore, managing requirement changes in software development is not just important but crucial for the success of the final product (Jayatilleke & Lai, 2018).

Requirement Changes in Traditional Software Development Methods

According to the literature, managing changes represent a significant problem in projects that follow a traditional (plan-driven, linear) software development method (Inayat et al., 2015; Papadopoulos, 2015; Wagner et al., 2018).

In these methods, the acceptance of change is generally low since they affect the project’s schedule and costs (Highsmith & Cockburn, 2001).

Requirement Changes in Agile Software Development Methods

On the other hand, agile methods are known for following a paradigm that advocates acceptance of change (Dingsøyr et al., 2012).

According to the Agile Manifesto and its second principle, agile methods welcome changing requirements even late in the software development process. This seems to be one of the main reasons software development teams move from traditional to agile methods (Paetsch et al., 2003; Cao & Ramesh, 2008; Papadopoulos, 2015).

Requirement Changes in Agile Software Development Methods

It seems that, theoretically, agile software development methods effectively address the problem of requirements change management faced by traditional methods. However, a question arises as if this effectiveness is perceived in real-life software projects and practice corresponds with the theory.

Research Method and Questions

In this research, a literature review was conducted to answer these two questions:

1. How do agile methods manage requirement changes?

2. According to the empirical evidence, are agile methods effective in managing requirement changes?

Review Approach

Databases	IEEE, ACM, ScienceDirect, SpringerLink		
Search criteria	C1	C2	C3
	"change management", "changing requirements", "requirement changes", "changes in requirements", "requirement engineering", "traceability", "impact analysis", and "prioritization"	Agile, Scrum, "Extreme Programming", "Dynamic Systems Development Method", Kanban, Lean, and "Feature-Driven Development"	empirical, "case study", survey, "qualitative study", and experiment
	C1 AND C2 AND C3		
Studies retrieved	373		
Final selection after applying the exclusion criteria	25		

Requirements Change Management

According to the International Requirements Engineering Board (Bühne & Herrmann, 2022) and ISO 29148, the requirements change management process includes or is related to important activities such as:

- Requirements change documentation
- Requirements traceability and change impact analysis
- Requirements prioritization

Results – Research Question 1

Based on these three activities, the first question,

How do agile methods manage requirement changes?

was responded to considering what the methods' guidelines indicate about the activity and how the activity is conducted in real-life projects as reported in the literature.

Requirement Changes Documentation – Agile Guidelines

- New requirements lead to new user stories.
- Changes to requirements lead to a user story being changed or replaced by a new one.
- Versioning of user stories is unnecessary.

(Bühne & Herrmann, 2022)

Requirement Changes Documentation – In Practice

Source	Study	Objective	Findings related to Requirement Changes Documentation
(Alsalemi & Yeoh, 2015)	Survey (89 Scrum practitioners)	Determine how product backlog changes are managed by practitioners in Scrum projects.	Most practitioners record requirement changes as new requirements even when these changes do not correspond with additions. No change history and versioning are maintained. The product backlog is saved immediately as requirements change. No change record is maintained for the product backlog.
(Wagner et al., 2018)	Survey (92 companies from 10 different countries)	Analyze the current state of practice and potential problems in agile requirements engineering.	Most practitioners indicated that they update the product backlog when requirements change. The study does not mention if they record requirement changes as new requirements independently of the type of change. In addition, 16 % of the practitioners only work with change requests, and 15 % even have a requirements specification they regularly change.
(Kasauli et al., 2021)	Multiple case study (seven large-scale system development companies)	Report on industrial requirement engineering challenges related to applying agile development in large-scale systems.	Three of the seven companies reported that their requirements are often outdated; they are not being updated as they change. Although it is not implicitly reported in the study, it may indicate that requirement versioning is not made. The study does not mention if these companies record requirement changes as new requirements independently of the type of change.

Requirements Traceability and Change Impact Analysis

Traceability is important in requirements change management because it assists in tracking changes in software architecture, code base, and test suites that requirement changes can cause. When a requirement change is requested, it is necessary to evaluate its impact on other requirements and the components that implement them (Carniel & Pegoraro, 2018).

Requirements Traceability – Agile Guidelines

1. Traceability among requirements (dependencies) is not mentioned. These tracing links are discussed during planning or inter-iteration meetings (Beck & Andres, 2005) (Bühne & Herrmann, 2022).

2. Traceability of the source code changes to the requirement they implement (and vice versa) is not covered (Asklund et al., 2004).

3. There is implicit traceability of requirements to the corresponding acceptance test cases (Bühne & Herrmann, 2022).

4. Traceability links do not need to be documented because they can be discovered and discussed during face-to-face meetings (based on developers' knowledge of the system) (Rubin, 2012).

Requirements Traceability (RT) – In Practice

Source	Study	What type of traceability is considered?	Who is responsible for RT?	How are traceability links identified?	What tools are used to manage RT?	Other findings related to RT
(Martakis & Daneva, 2013)	Focus group (six practitioners from five different companies)	Traceability among requirements.	No specific role is indicated.	Analysis during iteration planning and continuous communication and collaboration between project members.	JIRA, Pivotal tracker + Smartsheet, a list with linkage in Microsoft Sharepoint.	Participants of this focus group agreed that no agile approach considers requirements traceability explicitly, and there was no commonly used agile practice that agile project organizations meant for systematically approaching traceability in their projects.
(Duc, 2013)	Interview (14 practitioners from 12 different companies)	• Traceability from a test case back to a code element. • Traceability from a test case back to an user story.	No specific role is indicated.	Using a tool (see next column).	Small companies: Excel sheet, product Backlog, and whiteboard. Big companies: Visio Studio and Team foundation server, IBM rationale (ClearQuest, DOOR), Jira.	Participants of this study indicated that performing adequate requirements traceability in Agile projects is challenging because implementing traceability requires companies to spend more resources (time, employees). In addition, implementing traceability at a low level may be problematic in Agile because requirement changes occur frequently.
(Nurdiani et al., 2016)	Survey (52 practitioners)	Traceability among requirements.	Developers, analysts, scrum master, testers, manager.	Using a tool, common sense/logic/intuition, manually checking backlog items, analysis during iteration planning, team feedback.	JIRA, in-house developed tool, database repository, Git, CASE tool, Google docs, StoryMaps, Spreadsheet, Confluence, Wiki, Inflectra Spira Team	Although most participants of this study did not find managing requirements traceability challenging, the authors of this study argued that Agile practices themselves are not enough to manage requirements traceability since most respondents indicated that they use issue tracking and project management software tools to identify and manage requirements traceability, and this contradicts Agile Manifesto which advocates interactions over tools as part of its values.

Requirements Traceability (RT) – In Practice

Source	Study	Objective	Findings related to RT
(Kassab, 2014)	Survey (247 practitioners from 23 countries)	Report on the current practice landscape in requirements elicitation, analysis, management, effort estimation, and tools concerning agile methods compared with the traditional waterfall.	85% of the agile practitioners who participated in this study reported satisfaction with the process followed to trace requirements during their projects.
(Wagner et al., 2018)	Survey (92 companies from 10 different countries)	Analyze the current state of practice and potential problems in agile requirements engineering.	More than half of the respondents answered that they explicitly manage traces between the requirements and the code. A third manages the traces between requirements and design documents explicitly. More than a fifth of the respondents do not explicitly manage traces at all. This study reported "missing traceability" as a critical problem in agile requirement engineering.
(Kasauli et al., 2021)	Multiple case study (seven large-scale system development companies)	Report on industrial requirement engineering challenges related to applying agile development in large-scale systems.	Creating and maintaining requirement traceability is challenging. For example, some user stories need to be broken into more user stories, creating traceability issues as it is hard to understand which high-level user story can be traced to detailed requirements. Some interviewees considered requirements traceability to be documentation, which should not be part of the agile process.

Change Impact Analysis – Agile Guidelines

- Change impact analysis is not explicitly mentioned in Agile methods guidelines. However, Agile methods suggest that requirement changes and their implications can be properly addressed through constant face-to-face communication and feedback (development team members, customers – developers) (Beck & Andres, 2005), (Rubin, 2012).

- Not specific roles in charge are mentioned.

- Agile methods do not suggest specific impact analysis techniques.

Change Impact Analysis – In Practice

Source	Study	Objective	Findings related to Change Impact Analysis
(Wagner et al., 2018)	Survey (92 companies from 10 different countries)	Analyze the current state of practice and potential problems in agile requirements engineering.	Most respondents do impact analysis between requirements. More than a third analyzes the impact of requirement changes on the code. A fifth does not analyze the effect of changes to the requirements. The study also showed that besides looking at requirements and code, the test suites and direct discussions with stakeholders seem important for impact analyses in agile projects.

No other study mentioned anything about change impact analysis.

Requirements Prioritization

Prioritization is essential in requirement change management. When an addition, modification, deletion, or error fix is requested, the list of requirements must be re-prioritized to accommodate the changes (requirements) according to their degree of importance for the business or another prioritization criteria.

Requirements Prioritization (RP) – Agile Guidelines

When does RP take place?	During the initial planning stage and in the planning meeting at the beginning of each iteration
Who participates in RP?	• Customer • The development team takes the advisor role by estimating effort and judging technical risk
Who makes the final decision about RP?	Customer
Which criteria are considered in RP?	Business value
Which RP technique is used?	There is no specific requirement prioritization technique to be used

(Beck & Andres, 2005; Rubin, 2012; Schwaber & Sutherland, 2020)

Requirements Prioritization (RP) – In Practice

Source	Study	When does RP take place?	Who participates in RP?	Who makes the final decision about RP?	Which criteria are considered in RP?	Which RP technique is used?
(Cao & Ramesh, 2008)	Multiple case study (16 organizations)	Planning meeting (at the beginning of each iteration).	Customer and developers.	Customer	Business value	-
(Hoff et al., 2008)	Survey	-	-	-	Impact to the organization, fixes errors, delivery date, and cost-benefit to the organization.	-
(Racheva et al., 2010)	Multiple case study	-	Developers, customers (sometimes).	Developers	Negative value	-
(Daneva et al., 2013)	Embedded case study (3 large projects)	Planning meeting (at the beginning of each iteration).	Product owner, business analyst, tech lead, domain owner, delivery team head, and test scenario team lead.	Product owner (customer)	Requirements dependencies, volatility, risk, effort, and technical debt.	
(Jarzębowicz & Sitko, 2020)	Survey	Planning meetings, during iteration, at the beginning of the software project.	Product owner, developers, analyst, Scrum Master, customer representative, and testers.	Customer representative	Business value for customer, interdependencies, complexity and difficulty, and stability.	Numerical scale, MoSCoW, and Kano
(Naicker & Maharaj, 2020)	Survey	-	Project manager and/or business analyst.	The project manager or team leader or consensus with business analysts.	Opportunity cost, business value, and requirement dependencies.	No specific prioritization technique used
(Borhan et al., 2022)	Survey	Mostly done at planning meetings.	Product owner, and development team.	Product owner (95%), development team (5%).	Business value, cost, complexity.	MoSCoW, cost-value ranking, cumulative voting,and Kano

Discussion – Research Question 1

- According to the studies reviewed, practitioners seem to follow the agile guidelines regarding how to document requirement changes.

- According to Agile guidelines, requirements traceability does not need to be documented, and it can be addressed through practices such as constant face-to-face communication between stakeholders. However, the studies reviewed reported that practitioners consider traceability in agile projects challenging, and it seems that practices proposed by agile guidelines are not enough to manage requirements traceability in their projects; consequently, they use software tools to support this activity.

Discussion – Research Question 1

- Although change impact analysis was only mentioned in one study, it is worth highlighting that, according to this study, direct discussions with stakeholders seem to have been important for impact analyses in those agile projects. This claim corresponds to what is suggested by the agile methods guidelines.

- Most studies reviewed indicate that practitioners follow the suggested agile guidelines regarding requirement prioritization.

Discussion – Research Question 1

Based on the studies reviewed, practitioners seem to follow agile guidelines regarding activities related to requirement change management.

Results - Research Question 2

According to the empirical evidence, are agile methods effective in managing requirement changes?

Results from Empirical Studies

Source	Study	Findings related to requirements change management	Are agile methods effective in managing requirement changes according to this study?
(Sillitti et al., 2005)	Survey (16 project managers from 16 companies)	Based on the study's results, 87% of Agile companies indicated that requirements changes do not represent a problem or challenge in their software projects because their development process is based on constant interaction with the customer, which increases the level of understanding of the changes and allows to manage them adequately.	Yes
(Cao & Ramesh, 2008)	Multiple case study (16 organizations)	According to the study participant organizations, changes are easy to implement and cost less in Agile development due to early and constant feedback and validation of the requirements, which minimize the need for significant changes. However, several organizations also indicated that the most significant issue in managing changes is that the software architecture becomes inadequate as requirements change, and refactoring often does not completely address this problem.	No
(Liu Jun et al., 2010)	Case study	According to the study's results, changes were easy to conduct during the agile development of the evaluated project. Planning was a regular job; thus, fitting changes was relatively simple. Significant changes seldom took place during development activity. Constant communication between clients and developers handily avoids that.	Yes

Results from Empirical Studies

Source	Study	Findings related to requirements change management	Are agile methods effective in managing requirement changes according to this study?
(Maierhofer et al., 2010)	Survey (72 companies)	According to the study's results, Agile practices such as iterative development, constant and direct communication, and continuous customer integration did not demonstrate a mitigating effect on the negative impact of requirements changes in projects' success.	No
(Pikkarainen & Wang, 2011)	Multiple case study (two companies)	According to the study's results, the main issues manifested by the Scrum teams are related to requirement change management. Some of the issues they highlighted were that requirement changes were not analyzed deeply enough because the Agile way of working does not support change request analysis. In addition, they indicated that change management was complex because the increasing amount of requirement changes complicated the product backlog control and the requirements prioritization.	No
(Papatheocharous & Andreou, 2014)	Survey (377 respondents)	The respondents indicated that the ability to manage changing requirements and priorities is the most significant benefit they perceived of implementing agile methods. In addition, the study indagated the major causes for unsuccessful agile projects, and none of the respondents' answers were related to issues with the agile methods processes perse which includes requirement change management.	Yes

Results from Empirical Studies

Source	Study	Findings related to requirements change management	Are agile methods effective in managing requirement changes according to this study?
(Alawairdhi, 2016)	Multiple case study (two projects, one organization)	The study's results showed that the projects achieved high user satisfaction, good quality compliance, and low error rates by using agile methods. In addition, it was found that strong communication and constant feedback were determinants to manage the changes effectively.	Yes
(Wisitpongphan & Khampachua, 2016)	Multiple case study (two projects, one big organization)	The study's results showed that change management was one of the main challenges of applying agile methods in the two public sector software developed. The frequent meeting and reviewing of the development progress generated numerous change requests, which represented a problem in terms of costs because public sector projects usually have a fixed budget.	No
(Schön et al., 2017)	Survey (26 participants)	Experts of this study indicated that their companies have issues with the continuous management of requirements since not all of them are fixed at the beginning, and they change constantly throughout the projects. They also indicated that agile methods do not address these issues properly.	No

Results from Empirical Studies

Source	Study	Findings related to requirements change management	Are agile methods effective in managing requirement changes according to this study?
(Sebega & Mnkandla, 2017)	Survey (25 practitioners)	According to the study's results, 58.3% of the practitioners indicated that requirements changes are a significant issue in agile requirement engineering. In addition, 60.9% indicated have faced issues with requirements prioritization during agile project development.	No
(Wagner et al., 2018)	Survey (92 companies from 10 different countries)	Most practitioners indicated that they update the product backlog and do impact analysis when requirements change. In addition, more than half of them answered that they explicitly manage traces between the requirements and the code. However, practitioners also reported that one of the main and frequent problems they face in agile projects is moving targets (changing goals, business processes, and/or requirements) and their management.	No

Results from Empirical Studies

Source	Study	Findings related to requirements change management	Are agile methods effective in managing requirement changes according to this study?
(Rizkiyah et al., 2020)	Case study	According to the study, constant changes of requirements and re-evaluation, difficulties in managing the history of changing requirements, and difficulties in requirement prioritization because the customer considers all features important were some of the main issues why the analyzed government outsourcing project failed.	No
(Barata et al., 2023)	Survey (46 Brazilian software development professionals)	According to the study's results, 73.9% of respondents informed that requirements specifications and management represent challenges in agile projects and need to be improved. Among the main issues, they highlighted the insufficient change control in requirements.	No

Results from Empirical Studies – Research Question 2

Studies that evidence agile methods are effective in managing requirement changes.	4
Studies that evidence agile methods are not effective in managing requirement changes.	9

Discussion – Research Question 2

- Most studies showed evidence that agile methods are not effective in managing requirement changes. A constant statement in these studies is that requirement changes are one of the most significant issues practitioners face during an agile project, and their management represents a challenge. Therefore, it can be argued that if agile methods were completely effective in managing requirement changes, requirement changes and their management would not be an issue constantly highlighted in real agile projects.

Discussion – Research Question 2

- In the four studies that showed evidence that agile methods are effective in managing requirement changes, practitioners only mentioned that constant communication and feedback between customers and developers and continuous planning helped them manage requirement changes properly.

- It is worth noting that these four studies correspond to two case studies that examined small-sized projects, one survey with 16 agile practitioners from medium/small companies as respondents and another survey with 377 respondents where it was indicated that the second most widely used method by respondents' companies is a customized approach, almost 50% of the companies have developed only a small number of agile projects, and the typical number of developers recorded in agile teams in these companies is small.

Discussion

- Based on the results from research question one, practitioners seem to follow agile guidelines regarding activities related to requirement change management. In addition, results from research question two evidence that agile methods are not effective in managing requirement changes. Therefore, it can be argued that agile methods, **as described in their guidelines**, are not effective in managing requirement changes.

Conclusion

- This research presented a literature review to examine how requirement changes are managed according to agile methods and determine if these methods are effective in managing requirement changes.
- This study's findings suggest that agile methods, as described in their guidelines, are not effective in managing requirement changes.
- Several agile practitioners from different real-life projects and countries manifest that requirement changes represent a significant issue in their projects. Managing requirement changes, including activities such as change impact analysis, requirements traceability, and requirements prioritization, is a constant challenge reported by practitioners in the reviewed studies.
- Although the Agile philosophy advocates acceptance of change, and methods that follow this philosophy are expected to address changing requirements effectively, the practitioners' experiences seem to indicate another reality.

References

- Alawairdhi, M. (2016). Agile development as a change management approach in software projects: Applied case study. *2016 2nd International Conference on Information Management (ICIM)*, 100–104. https://doi.org/10.1109/INFOMAN.2016.7477541

- Alsalemi, A. M., & Yeoh, E.-T. (2015). A survey on product backlog change management and requirement traceability in agile (Scrum). *2015 9th Malaysian Software Engineering Conference (MySEC)*, 189–194. https://doi.org/10.1109/MySEC.2015.7475219

- Asklund, U., Bendix, L., & Ekman, T. (2004). *Software Configuration Management Practices for eXtreme Programming Teams*.

- Barata, J. C., Lisboa, D., Bastos, L. C., & Neto, A. (2023). Agile Requirements Engineering Practices: A Survey in Brazilian Software Development Companies. In C. Rocha, C. Santana Júnior, F. De Sá, & T. Silva Da Silva (Eds.), *Agile Methods* (Vol. 1642, pp. 110–119). Springer International Publishing. https://doi.org/10.1007/978-3-031-25648-6_9

- Beck, K., & Andres, C. (2005). *Extreme programming explained: Embrace change* (2nd ed). Addison-Wesley.

- Borhan, N. H., Zulzalil, H., Hassan, S., Hayati, N., & Ali, M. (2022). REQUIREMENTS PRIORITIZATION IN AGILE PROJECTS: FROM EXPERTS' PERSPECTIVES. *Journal of Theoretical and Applied Information Technology*, *100*(19).

- Bühne, S., & Herrmann, A. (2022). *Handbook Requirements Management*.

- Cao, L., & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, *25*(1), 60–67. https://doi.org/10.1109/MS.2008.1

References

- Carniel, C. A., & Pegoraro, R. A. (2018). Metamodel for Requirements Traceability and Impact Analysis on Agile Methods. In V. A. D. Santos, G. H. L. Pinto, & A. G. Serra Seca Neto (Eds.), *Agile Methods* (Vol. 802, pp. 105–117). Springer International Publishing. https://doi.org/10.1007/978-3-319-73673-0_9
- Daneva, M., Van Der Veen, E., Amrit, C., Ghaisas, S., Sikkel, K., Kumar, R., Ajmeri, N., Ramteerthkar, U., & Wieringa, R. (2013). Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *Journal of Systems and Software*, *86*(5), 1333–1353. https://doi.org/10.1016/j.jss.2012.12.046
- Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.033
- Duc, V. H. (2013). *Traceability in Agile software projects*.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, *34*(9), 120–127. https://doi.org/10.1109/2.947100
- Hoff, G., Fruhling, A., & Ward, K. (2008). *Requirement Prioritization Decision Factors for Agile Development Environments*.
- Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, *51*, 915–929. https://doi.org/10.1016/j.chb.2014.10.046
- Jarzębowicz, A., & Sitko, N. (2020). Agile Requirements Prioritization in Practice: Results of an Industrial Survey. *Procedia Computer Science*, *176*, 3446–3455. https://doi.org/10.1016/j.procs.2020.09.052

References

- Jayatilleke, S., & Lai, R. (2018). A systematic review of requirements change management. *Information and Software Technology*, *93*, 163–185. https://doi.org/10.1016/j.infsof.2017.09.004

- Kasauli, R., Knauss, E., Horkoff, J., Liebel, G., & De Oliveira Neto, F. G. (2021). Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software*, *172*, 110851. https://doi.org/10.1016/j.jss.2020.110851

- Kassab, M. (2014). An Empirical Study on the Requirements Engineering Practices for Agile Software Development. *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, 254–261. https://doi.org/10.1109/SEAA.2014.77

- Liu Jun, Wang Qiuzhen, & Gao Lin. (2010). Application of Agile Requirement Engineering in Modest-Sized Information Systems Development. *2010 Second World Congress on Software Engineering*, 207–210. https://doi.org/10.1109/WCSE.2010.105

- Maierhofer, S., Stelzmann, E., Kohlbacher, M., & Fellner, B. (2010). Requirement Changes and Project Success: The Moderating Effects of Agile Approaches in System Engineering Projects. In A. Riel, R. O'Connor, S. Tichkiewitch, & R. Messnarz (Eds.), *Systems, Software and Services Process Improvement* (Vol. 99, pp. 60–70). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15666-3_6

- Martakis, A., & Daneva, M. (2013). Handling requirements dependencies in agile projects: A focus group with agile software development practitioners. *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, 1–11. https://doi.org/10.1109/RCIS.2013.6577679

- Naicker, N., & Maharaj, M. (2020). Investigating Agile Requirements Engineering Practices in the South African Software Development Market. *Journal of Computing and Information Technology*, *28*(1), 33–58. https://doi.org/10.20532/cit.2020.1004868

References

- Nurdiani, I., Jabangwe, R., & Petersen, K. (2016). Practices and Challenges of Managing Requirements Interdependencies in Agile Software Development: A Survey. *2016 International Conference on Engineering, Technology and Innovation/IEEE Lnternational Technology Management Conference (ICE/ITMC)*, 1–8. https://doi.org/10.1109/ICE/ITMC39735.2016.9025919

- Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, 308–313. https://doi.org/10.1109/ENABL.2003.1231428

- Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. *Procedia - Social and Behavioral Sciences*, *175*, 455–463. https://doi.org/10.1016/j.sbspro.2015.01.1223

- Papatheocharous, E., & Andreou, A. S. (2014). Empirical evidence and state of practice of software agile teams: EMPIRICAL EVIDENCE AND STATE OF PRACTICE OF SOFTWARE AGILE TEAMS. *Journal of Software: Evolution and Process*, *26*(9), 855–866. https://doi.org/10.1002/smr.1664

- Pikkarainen, M., & Wang, X. (2011). An Investigation of Agility Issues in Scrum Teams Using Agility Indicators. In W. W. Song, S. Xu, C. Wan, Y. Zhong, W. Wojtkowski, G. Wojtkowski, & H. Linger (Eds.), *Information Systems Development* (pp. 449–459). Springer New York. https://doi.org/10.1007/978-1-4419-7355-9_38

- Project Management Institute (Ed.). (2017). *The Agile practice guide*. The Project Management Institute.

References

- 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. *2010 18th IEEE International Requirements Engineering Conference*, 147–156. https://doi.org/10.1109/RE.2010.27

- Rizkiyah, K., Nisyak, A. K., & Raharjo, T. (2020). Agile-Based Requirement Challenges of Government Outsourcing Project: A Case Study. *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, 267–273. https://doi.org/10.1109/IC2IE50715.2020.9274659

- Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular agile process*. Addison-Wesley.

- Schön, E.-M., Winter, D., Escalona, M. J., & Thomaschewski, J. (2017). Key Challenges in Agile Requirements Engineering. In H. Baumeister, H. Lichter, & M. Riebisch (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (Vol. 283, pp. 37–51). Springer International Publishing. https://doi.org/10.1007/978-3-319-57633-6_3

- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*. www.scrum.org

- Sebega, Y., & Mnkandla, E. (2017). Exploring Issues in Agile Requirements Engineering in the South African Software Industry. *The Electronic Journal of Information Systems in Developing Countries*, *81*(1), 1–18. https://doi.org/10.1002/j.1681-4835.2017.tb00597.x

References

- Sillitti, A., Ceschi, M., Russo, B., & Succi, G. (2005). Managing Uncertainty in Requirements: A Survey in Documentation-Driven and Agile Companies. *11th IEEE International Software Metrics Symposium (METRICS'05)*, 17–17. https://doi.org/10.1109/METRICS.2005.29
- Wagner, S., Fernández, D. M., Felderer, M., & Kalinowski, M. (n.d.). *Requirements Engineering Practice and Problems in Agile Projects: Results from an International Survey*.
- Wiegers, K., & Beatty, J. (2013). *Software Requirements*. Pearson Education. https://books.google.com.pa/books?id=nbpCAwAAQBAJ
- Wisitpongphan, N., & Khampachua, T. (2016). Agile in public sector: Case study of dairy farm management projects. *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 1–5. https://doi.org/10.1109/JCSSE.2016.7748916

UNIVERSITY OF WATERLOO

Thanks for your attention.

Do you have any questions?