





# The Role of the Customer in an Agile Project: A Multi-case Study

Erno Vanhala<sup>1</sup>✉  and Jussi Kasurinen<sup>2</sup> 

<sup>1</sup> Tampere University, Tampere, Finland  
erno.vanhala@tuni.fi

<sup>2</sup> LUT University, Lappeenranta, Finland  
jussi.kasurinen@lut.fi

**Abstract.** In this multi-case study we report the findings from three software projects conducted with SCRUM agile development framework. Each project took approximately a little less than a year to design, develop and test before the launch to the user groups. All project vendors utilized SCRUM framework customized to suit their processes, and included customer as a participant in the overall process. Due to this fact, this study focuses on the role of the customer in daily life of an agile project. The findings show what is actually required from the customer – especially when the sprint length is only one week and the development process is very time-intensive. Although a one week sprint cycle can lead to improved efficiency it required a full time worker from the customer side and it burdened also the developers. Based on our observations, as the developer teams and customer were located in various places around Europe, smooth communication was a key for success. In all cases the asynchronous communication tools, such as Slack, were highly praised, although also direct communications were used to handle more complex issues. According to our findings, these agile projects did not have significant issues caused by the online communication being the preferred way of communication. All of the cases had difficulties in fitting the agile project to the fixed budget, but good collaboration, partnership and trust alleviated most of these problems.

**Keywords:** Agile software development · SCRUM · Customer relationship · Multi-case study

## 1 Introduction

Software engineering fundamentals are not very swift to change. For example, the nowadays commonly used agile methods such as eXtreme Programming (XP) and Scrum, are already more than twenty years old [15]. Even the agile manifesto itself is turning twenty in two years [2], and it more or less codifies software process expertise, which was already known fifty years ago [15]. Agile software engineering methods have been studied from various perspectives; yet, the role and especially the requirements set for the customer in an agile software project

require deeper understanding since there seem to be only few works where this issue is even mentioned [16,22].

The agile manifesto itself states how interaction and customer collaboration are important parts of software development [2]. How the actual software design, development and testing lifecycle is then carried out depends on the case and the study. The reported issues are different whether the research concentrates on customer side, developer side or both (e.g. [16,20]). To get more in-depth understanding of the customer problems and issues with the communication between the customer and the vendor, in this study it was decided to get interview data from both sides to see how the role of the customer is formed. On the software process aspects, it was interesting to understand what the customers actually do or understand, and how they relate themselves to the rest of the development team. Based on earlier work, we had an understanding that the most common ways of customer participation was on the first stages in definition phases, and on the last stages in acceptance testing [12]. But was this still the case with the agile development practices and if not, how had the adoption of the agile methods in large scale affected this dynamic?

In this work we discuss and analyze this role of the customer with the following research questions:

1. How do the customers consider working in an agile project?
2. What are the appropriate communications mechanisms and how effective are they?
3. What do the participants from software organizations expect from the customers in an agile project?

With these research questions in mind we studied three software projects conducted in 2018–2019 and present the results in this article.

The rest of the article contains first related research in Sect. 2, description of research process in Sect. 3, results in Sect. 4 and discussion of findings in Sect. 5. Section 6 concludes the study.

## 2 Related Research

The agile world has embraced change happening in the software development [13,30]. Yet, especially public organizations have preferred fixed pricing when buying software [7]. This has created an equation, which has been described problematic, but also manageable [1,4]. Agile software development has nevertheless become the new norm [9] in all but the most heavily regulated areas of the industry.

When talking about agile software development one is describing an umbrella term: the agile world consists of many different process models, frameworks and development strategies which may vary to a large degree from each other [2]. In the beginning of this millenia eXtreme programming (XP) was discussed a lot in industry and also in academia, but the shift has then been towards scrum,

albeit the continuous development models lean heavily on the principles first introduced with the XP approaches [10].

In general, XP and the Continuous models are strongly related to the actual software-in-development while SCRUM or Dynamic System Development Model (DSDM) are more project management methods. Other methodologies, such as, Feature Driven Development (FDD) or Kanban can be considered between those two [18].

Agile methods have been studied from various point of views; Google scholar return hundreds of results when searching for agile method in title. In their systematic literature review Dybå and Dingsøy identified 36 articles discussing empirical studies of agile software development [3], but only a handful of those focused on the participation of the customer and collaboration. There are studies discussing agile methods and user centric design [24] and the role of user stories [23]. And it has been discussed how daily communication with the customer and the vendor reduces overruns [17,22]. Martin et al. [20] discuss how important the role of the customer is in XP project. The role of the customer includes not only to provide user stories and acceptance testing, but also communication to external stakeholders and keeping the trust between the vendor and the funder; the customer is the glue keeping the project together. The overall communication, collaboration and coordination is important and it has been even discussed how these elements ensure quality and productivity in an agile project [11,21]. Also Korkala et al. [14] present their findings on how lesser communication with customer reflects on the higher defect rates. They embrace face-to-face communication but also accept online video collaboration when participants are remote.

Sprint length in SCRUM development is usually 2–6 weeks [6,28] and conventionally it has been preferred that the development teams are physically in the same place [13]. However, this has changed with the improved Internet connections and online communication and collaboration tools [26], to the point where it has been reported how distributed teams can be as productive as collocated teams [27].

In a nutshell: agile methods, which are many, have been studied quite a lot, yet the role of the customer has not been in the focus.

### 3 Research Process

This study is a multi-case study and it follows the frameworks and principles presented by Gable [8] and Eisenhardt [5]. We followed seven steps: defining the strategy, reviewing the literature, developing the case study protocol, conducting a pilot case study, conducting a multiple case study, developing a conceptual model and interpreting the findings.

The research questions, presented in section one, determine the overall strategy. Section two illustrates the related literature. The case study was based on two interview rounds where the first one was conducted with the customer representatives and the second one with the vendor representatives. Data was collected through interview rounds where the first author interviewed the customer

and the vendor representatives and the second author validated the interview questions and interview recordings. The case organizations were selected from the pool of professional contacts, which were working with a software project utilizing an agile method. The aim was to interview the project managers and leaders – the persons who worked most for the project – from the customer side and the main architect and/or project manager from the vendor side. Typically one interview lasted for one hour, and included approximately ten semistructured questions, with subquestions, which allowed also open discussions. Key information of the interviewees is presented in Table 1.

**Table 1.** Information on the interviewed persons

Case A	Case B	Case C
Customer product owner (PO) ***	Customer PO **	Customer PO A ***
Customer secondary PO **	Customer secondary PO, head of content production *	Customer PO B **
Vendor chief architect ***	Vendor chief architect ***	Vendor project manager ***

All the projects had a person dedicated to the project and responsible for budgeting, reporting and taking care of running the project from the customer side, this is called product owner (PO). Secondary PO is the person who helps the most and works as a PO when the real PO is not present (e.g. on holidays). The PO of Case C was changed during the project. Asterisks illustrate how extensive project work experience that person had (\* = none, \*\* = some, \*\*\* = extensive). Product owners were not software engineering professional – with the exception of Customer PO A, who had software engineering background and formal training to act as PO.

### 3.1 Description of the Cases

This study discussed three cases. The customer organization built three systems almost simultaneously. Cases A and B were developed by Vendor 1 and Case C was delivered by Vendor 2. Although some of the customer’s people were working on all of the case projects the projects also had their dedicated product owners from the customer side.

With Case A there was a strict deadline when the system needed to be in production and there was no option to miss the date. With Case B it would have been optimal if the system had been up and running with the same date as Case A, but that deadline was not that crucial. With Case C the schedule was more flexible as the first ideas were to finish the Case C before Case A, but it was also acceptable to postpone the release of Case C after the Case A and that was also the final outcome.

### 3.2 Details of the Cases

Table 2 presents the key figures of the cases. The software in Cases A and B were bought from the Vendor 1 and Case C was delivered by Vendor 2. All the software were browser-based aimed to provide tools to share information and materials to both users in the organization and to the public, and to integrate to

a larger framework utilizing for example Drupal and WordPress. There had been some requirements for analysis and specification with consults before the vendors were selected. The GDPR, data security and system validation requirements of all the observed systems were similar, and generally comparable to each other.

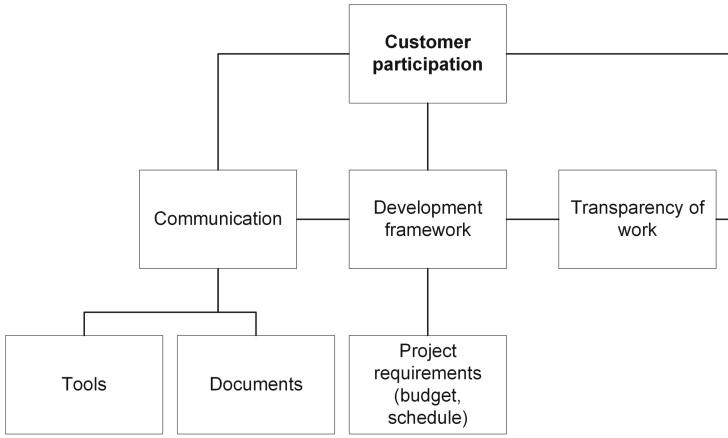
**Table 2.** Key figures of the case projects

	Case A	Case B	Case C
Vendor	V1	V1	V2
Schedule criticalness	Release date cannot be missed, features can be compromised	Both release date and feature richness can be flexible	Feature completeness is more important than the release date
Expected number of authenticated users	Hundreds	Tens of thousands	Hundreds
Software development framework	Scrum	Scrum	Scrum
Sprint length	1 week	1 week	2 weeks
Dailies mandatory for product owner	Yes	Yes	No
Customer testing	Weekly	Weekly	Before the release
Software framework	Drupal	Custom React code + Drupal	WordPress
Vendor team size	ca. 10 persons	ca. 10 persons	ca. 5 persons
Customer team size <sup>a</sup>	ca. 10 persons	ca. 5 persons	ca. 5 persons
Vendor documentation platform	Google Drive	Google Drive	Google Drive
Customer documentation platform	Wiki, O365	Wiki, O365	Wiki, O365
Communication tool	Skype for Business, Zoom, Slack	Skype for Business, Zoom, Slack	Google Meet, Slack
Tool to handle product related daily tasks	Jira	Jira	Jira
Pricing	Time & material	Target price	Fixed price
Budget	Hundreds of thousands of euros	Hundreds of thousands of euros	Tens of thousands euros
Estimated project duration (achieved)	7 months (10)	6 months (8)	7 months (10)

<sup>a</sup>Does not include content production team

## 4 Results

In this section the results are discussed. Overall, we observed seven main aspects which greatly influenced the customer participation and client roles in the software projects. This includes discussion of general frameworks utilized, what tools and documents were used, how communication was carried out, what happened to the budget and scheduling and how transparent the project work was. These relations are illustrated in the Fig. 1.



**Fig. 1.** The most common observed related aspects on the topic “customer participation”.

The customer participation was observed to be associated with communication policies, the development framework of the projects and transparency aspects. These aspects were further defined by the applied tools, documents and the general project requirements, especially budgets and schedules, which affected the way the customer participated in the development work. There might also be further underlying aspects, but within our data and our observed projects, these were the most meaningful influences which affected the roles and types of the participation. In the following subsections, we discuss the different aspects separately, and define how they affect the customer participation and working roles.

### 4.1 Framework

On the general topic of the first research question, the applied process models were investigated to understand how the development work is done in general and how much these approaches demand cooperation and customer participation. The vendor of Cases A and B utilized a scrum framework customized for their organization. The key points of this customization were: estimated 60% workload

for a person from the customer organization and the utilization of sprint length of one week instead of the de facto two week length.

Although the vendor required only 60% of time to be scheduled for the project the reality was different. Almost double work was required from the product owner.

“In fact I spent 120% of my time on the project.” - product owner of Case A.

“My workload was something between 60 and 100%.” - product owner of Case B.

Besides the workload, also the meaningfulness of the one week sprint length was questioned. Although it was also considered good when there were more issues to be decided each week. The product owner was also expected to participate dailies five times a week. In the beginning the customer's role was just to listen, but when everyone got to know each other the customer was also giving feedback.

“The one week sprint length produced a huge load of overhead. It was meetings and planning all the time.” - secondary product owner of Case A.

“I think it was a good balance of planning and developing” - product owner of Case B.

The Vendor 1 argued how the one week sprint length had increased their productivity. They had a two week sprint length, but the move to one week had been considered as a good choice. Though it resulted in increased productivity, they had also noted that it required a lot from both developers and customers. It was considered a good choice if the developer could work in a maintenance project for a while after scrum development project had ended. This is an opposite finding when compared to, for example [17], where one month sprint length was used. Still the Vendor 1 considered one week sprints most suitable.

“In one week period we can really be sure what we need to do.” - chief architect of Vendor 1.

Vendor 2, which developed the Case C, utilized two weeks sprints and did not require customers participation as much as Vendor 1. Dailies were held only internally and no customer participation was required or even offered. Although there was less participation, there was still much to do for the customer: sprint meetings, testing, design decisions, to name a few.

Testing was very different between the two vendors. With Vendor 1 the customer tested the new features each sprint week and had to do quite a lot of work with testing. This was also noted in the interviews:

“It looks like I am working in Vendor 1. Sometimes it feels like I am doing their jobs” - product owner of Case B.

It was criticised how the testing responsibility was on the customer side – although this is in line with the findings in [11]. For example all the integrations needed to be verified by the customer and in many cases it was reported how one field here and another there was missing. It created a burden. Vendor 1 also noted this.

“The customer’s role in testing has been too big. It has to be also noted how pedantic the customer has been. There has been very little bugs in the production.” - chief architect of Vendor 1.

With Vendor 2 in Case C the testing workload was smaller and stressed the project group merely in the end of the release cycle, not on a weekly basis. The philosophies of Vendor 1 and Vendor 2 can be considered quite different, yet they both worked.

## 4.2 Tools and Documents

On more in-depth topic regarding cooperation and work, the communication mechanisms and cooperation-enabling tools were also an area of interest. Fundamentally, all case projects included collaborative work with various documents, such as requirements and customer testing. In all cases the work was done in Google Drive environment. Both customer and vendors were satisfied with these tools. With Case A it was reported that documents were made, but not updated that much during the project. With Cases B and C both customer and vendor reported that all non-temporary documents were kept updated and used during the project.

“All the documents created were really in use, but with meeting memos there were problems when same issues were discussed in various meetings and the results were not consistent” - product owner B of Case C.

“All the documentation was in Google Drive and all the documents were linked in Jira. The developer could always go from Jira ticket to up-to-date information found from Google Drive.” - project manager of Vendor 2 (Case C).

Both vendors used Atlassian Jira as the issue and bug tracking project management tool. Although some project members from the customer side had never used it before and described it as “spooky” when first seen, the utilization of story and bug reporting in Jira was a success.

## 4.3 Communication

Besides tools, the methods and volume of communication between the client and the developer were assessed, since the communications and exchange of information between organizations are considered one of the key values of agile approaches. Both vendors used Slack online discussing tool as the main communication method. The Vendor 1 also used Zoom and Vendor 2 Google Meet. Also Skype for Business was used, especially internally on the customer side. Both vendors also arranged live meetings. Email was disfavoured although still used occasionally. Especially with Vendor 1 there were several face to face designing sessions before the implementation part that led to intensive work as partners from the beginning. Although both Case A and Case B had disagreements between the customer and the vendor no conflicts arose. The product owner of Case B felt it good how the work was intensive between the vendor and the customer:

“They have become like colleagues”, product owner of Case B.



Vendor 1 had developers in several cities in two different countries. Some of them never visited the customer physically, yet they still fit in the project group and were considered as good partners as those who were met face to face. Slack and Zoom were found to be sufficient tools to handle day-to-day communication. This is a method of work that was not emphasized in the early XP visions [13], yet it is common and effective nowadays [26].

Although almost all key customer persons had at least some experience in working in software development project, there were still some problems with the communication, especially in Case B, where the key persons had the least of experience.

“The customer always responded, but sometimes we only got a fragment of what we needed and after that piece by piece. In this sense we build the software from hand to mouth.” - chief architect of Vendor 2.

The product owner of Case B mentions how it was felt straight from the beginning that resources were not enough.

“We should have had more resources internally in this project” - product owner of Case B.

It was not a problem that the vendor would be the bottleneck, but the customer who could not respond in time or get all the necessary information. Also too optimistic schedules from the Vendor’s side contributed to the missing of the deadline. Thus there was a mismatch in communication that resulted in missed deadlines; wrong requirements in correct date and vice versa.

With Case C the communication was not that intensive with the vendor and the customer and there were days when no messages were delivered. Still the Vendor 2 considered it supportive when the project communication was successful and helped when the customer and the vendor did not see eye to eye.

The overall view was that in the beginning of the projects face to face meetings were more common. When people started to know each other and the broad lines were set, and the actual work started, the need for physical meetings decreased but online communication – both textual and with voice – was fortified and it was considered working well.

#### 4.4 Transparency of the Project Work

With one week sprints Vendor 1 was able to communicate the project progression weekly and the customer was using Jira tool from day one, so that the project was all the time under an expressive supervision of the product owners of Case A and Case B. There was also a need for transparency the towards steering groups and the end users, but the lack of time prevented that.

“There simply was not enough time to communicate all the things in the project group not even mentioning the need to communicate with the end users.” - product owner of Case A.

With Case C the customer was not that intensively included in the daily work, but rather in testing features when they were announced. Sometimes some features were presented even if they were not required and the customer was not

sure if they were the ones they needed. Still the overall feeling within the vendor was that everything went smoothly.

“It was nice to see how the customer liked to work in this project” - project manager of Vendor 2 (Case C).

It seemed that the transparency required intensive collaboration that also required resources from the customer side so that it cannot be described only to have good parts. If resources are not a problem on the customer side, a deep communication can improve the transparency.

#### 4.5 Budget and Schedule

Finally, all projects are subject to some restrictions and objectives, usually defined by time, money, resources or quality to assess the success rate of the project. Although the customer and the vendor were in intense communication all the time, all the cases missed their budget and/or schedule in some way. This was especially bad with the Case B as it meant that the content needed to be updated simultaneously in old systems and in the production environment of the new system where the new guidelines of content production were set. Finally the product was released more than a month late and with beta-status. It was already decided in the beginning of the project that the first release was nothing but final, but the lack of features was still overwhelming.

There was also an enormous pressure to get the Case A done in time as the deadline was strict and could not be missed. There had already been delays for weeks in the previous beta and soft launches, which led to reducing features from the release. These features were then implemented after the release and that was also considered a burden as customer’s representatives were eager to move on with other tasks in hand.

With Case C everything else went smoothly but the authentication with the organization wide method was not easy to integrate to WordPress and that lead to missing the final deadline. The project had also a problem with the key developer’s sick leave as there was no replacement available.

“We had quite good resources for this project and could keep the deadlines. Although the injured developer had negative effect to meet the final deadline.” - project manager of Vendor 2 (Case C).

“I think the only problem with the schedule was the sick leave and a little lightweight know-how, that caused delays” - product owner B of Case C.

The Case C utilized a fixed price model and managed to get all done within the budget although they did not manage to do it in the time they had set internally, thus they used their own resources to get everything done.

“Fixed price and agile project – is it even possible? I think it is” - project manager of Vendor 2 (Case C).

With Case B there were negotiations after it was realized that the estimated work amount would exceed and with Case A fixed pricing was not even tried. This underlines how fixed price and agile project are challenging to combine.

## 4.6 Conclusion of the Findings

It was found how in these three cases agile software development required resources from the customer more than they anticipated and the product owners were overwhelmed by the workload the project gave them. The one week sprint length in Cases A and B was considered exhausting; yet the vendor had experienced its power.

From the actual tasks testing was considered the most burden. There was much more to test than the traditional waterfall acceptance testing.

Cloud environments as the backend for document sharing and collaborative editing were highly praised. As were also online communication tools that were used.

All the projects missed their budget or schedule in some way and it was noted how the customer had too little dedicated resources – i.e. manpower – that was a bottle neck in various occasions and also produced the lack of necessary communications.

## 5 Discussion

In the beginning we set three research questions: (1) How do the customers consider working in an agile project? (2) What are the appropriate communications mechanisms and how effective are they? and (3) What do the participants from software organizations expect from the customers in an agile project?

Four key points arose from interview after interview:

- Agile sprints require a lot from the customer; the customer has to provide information on a short notice and live with the schedules and workloads even if they are incompatible with their own organization.
- Communication through modern asynchronous online tools works as well as face to face; direct communications between the client and the developer are not considered overtly intrusive.
- Close collaboration and trust between the partner organizations can alleviate most of the problems; most of the issues are based on the lack or limited amount of communications between the client and the customer organizations.
- Agile project with a fixed budget is still a tricky concept; the amount of revisions and redesigns are difficult to estimate beforehand especially with a new client.

Especially Vendor 1 required a lot from the customer. They had experienced how one week sprint length is efficient and they put a heavy load of testing responsible to the customer. On one hand this burdens the customer, but on the other hand it guarantees that the customer gets what he wants and no unnecessary work is done when the sprint length is kept short. The problem is that the burden might be too much if the customer is not prepared for the workload. Within this study the customer had experience of agile software work,

but the workload was still considered too heavy on many times. If the vendor is working with the customer with no prior knowledge of software projects there could be a significant risk that the customer rejects the working method and the project fails. *We recommend that the agile software vendors communicate the responsibilities beforehand and underline how the customer's participation is crucial for the success of the project.*

XP has traditionally emphasized close physical proximity [13], yet these projects embraced online communication tools. Discussions popping up all the time in Slack – questions getting answered, bugs being fixed – illustrated that the tools we have today are sufficient to diminish the need for continuously physically shared spaces. And when textual chatting was not enough Skype and Zoom brought the customer and the developers to the same virtual room to discuss the issues at hand. In the beginning of the project physical meetings were held, but when approaching the release online communication had replaced the physical meetings almost entirely. *We recommend the customers and the vendors make a point of creating digital work space for all participants, and apply modern online communication tools whether they would be as sufficient as this study describes.*

Although none of these projects were complete successes, there was never real blaming from either side. The customer could always trust that the vendor gets all done even if it would mean being late few months or requiring more work. A deep collaboration and partnership helped all cases to overcome problems that could lead to courthouse. *We recommend to begin a software project with partnership in mind so that the problems are tackled together and not by blaming each other.*

Monetary issues were not the main research theme, but they arose from the interviews. With Case A it was decided that fixed pricing was not to be used, with Case B target pricing was used, but budgeting was an issue and with Case C fixed pricing was used and the customer was satisfied, but it resulted the vendor doing some development without payment. This emphasises how agile software development and fixed pricing is still a concept that needs a careful thinking whether it is suitable for the project or should some other pricing model be used. *We recommend to avoid strict fixed pricing with an agile project method, and to consider for example target price or similar more adjustable pricing.*

To summarize the findings in a nutshell: agile project requires more expertise from the customer and flexibility from the budget than traditional plan-driven projects to succeed, and the current online communications and collaboration tools enable agile development teams to locate physically all over the globe.

In qualitative studies, the validity issues concentrate on the generalizability and bias aspects of the researchers reflecting their own expectations on the data [25]. In this work, the research data was analyzed and documented by a team of researchers, and the qualitative data was collected from a group of experts representing different viewpoints in the software development project. In qualitative studies, the key aspects are integrity, authenticity, credibility and criticality [29]. In our work, we selected organizations and people directly involved in these projects to gain first hand information, and conducted the interviews personally,

getting observations from several roles to establish several viewpoints into the analyzed projects. In this sense, the integrity, authenticity and credibility of our observations should establish a firm chain-of-evidence between our observations, and the activities which have taken place in these development projects. As for generalizability, the qualitative studies in general cannot be generalized into all-encompassing theories such as in mathematics or physics but in transferability [19], with the study results being treated as areas of interest, or enhancement proposals, when transferred outside the original study ecosystem.

## 6 Conclusion

In this article, we present our qualitative multi-case study on the customer expectations and the roles in different types of agile processes. The concept was to study the effect the customer has on the software process, and by observing real-life software development projects, understand the benefits and risks of active/non-active customer participation in the development work. Our research work included three real life software projects, several different viewpoints and experts from both customer and vendor organizations to the study. The results were analyzed and agreed upon by a group of researchers, and the strongest leads based on the qualitative chain of evidence were reported as results.

Based on our results, the customer participation has a significant impact on the quality assurance activities and to the overall success of the agile project. Additionally, we observed that the customer participation does not require physical presence as documented in some agile practices such as XP, but for a meaningful participation it is sufficient that the customer participates for example via shared digital workspace. In fact, we did not find strong indicators of added benefits from the on-site presence by the customer representatives at the development team.

Finally, it seems that we have a number of attributes which affect the customer role and have impact on the overall project outcome. As a future research, it would be interesting to test these attributes for example with larger quantitative surveys to further validate our observations, or assess how software projects succeed, when the client behavior, responsibilities and representative requirements are specified to ease the identified problematic process areas.

## References

1. Banerjee, U., Narasimhan, E., Kanakalata, N.: Experience of executing fixed price off-shored agile project. In: Proceedings of the 4th India Software Engineering Conference on - ISEC 2011, pp. 69–75. ACM Press, Thiruvananthapurama (2011). <https://doi.org/10.1145/1953355.1953364>
2. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: towards explaining agile software development. *J. Syst. Software* **85**(6), 1213–1221 (2012). <https://doi.org/10.1016/j.jss.2012.02.033>

3. Dybå, T., Dingsøy, T.: Empirical studies of agile software development: a systematic review. *Inf. Software Technol.* **50**(9–10), 833–859 (2008). <https://doi.org/10.1016/j.infsof.2008.01.006>
4. Eckfeldt, B., Madden, R., Horowitz, J.: Selling agile: target-cost contracts. In: *Agile Development Conference (ADC 2005)*, pp. 160–166. IEEE Comput. Soc, Denver (2005). <https://doi.org/10.1109/ADC.2005.39>
5. Eisenhardt, K.M.: Building theories from case study research. *Academy of management. Acad. Manage. Rev.* **14**(4), 532 (1989). <http://search.proquest.com/docview/210938650?accountid=27292>, copyright - Copyright Academy of Management Oct 1989; Last updated - 2010-06-08
6. Flora, H., Chande, S.: A systematic study on agile software development methodologies and practices. *Int. J. Comput. Sci. Inf. Technol.* **5**(3), 3626–3637 (2014)
7. Franklin, T.: Adventures in agile contracting: evolving from time and materials to fixed price, fixed scope contracts. In: *Agile 2008 Conference*, pp. 269–273. IEEE, Toronto (2008). <https://doi.org/10.1109/Agile.2008.88>
8. Gable, G.G.: Integrating case study and survey research methods: an example in information systems. *Eur. J. Inf. Syst.* **3**, 112–126 (1994)
9. Hoda, R., Salleh, N., Grundy, J.: The rise and evolution of agile software development. *IEEE Softw.* **35**(5), 58–63 (2018). <https://doi.org/10.1109/MS.2018.290111318>
10. Humble, J., Farley, D.: *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education (2010)
11. Kautz, K.: Customer and user involvement in agile software development. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) *XP 2009. LNBIP*, vol. 31, pp. 168–173. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-01853-4\\_22](https://doi.org/10.1007/978-3-642-01853-4_22)
12. Kettunen, V., Kasurinen, J., Taipale, O., Smolander, K.: A study on agility and testing processes in software organizations. In: *Proceedings of the 19th International Symposium on Software Testing and Analysis - ISSTA 2010*, p. 231. ACM Press, Trento (2010). <https://doi.org/10.1145/1831708.1831737>
13. Kircher, M., Jain, P., Corsaro, A., Levine, D.: Distributed extreme programming. In: *Proceedings of the International Conference on eXtreme Programming and Flexible Processes in Software Engineering*, pp. 66–71. Sardinia, Italy, May 2001
14. Korkala, M., Abrahamsson, P., Kyllonen, P.: A case study on the impact of customer communication on defects in agile software development. In: *AGILE 2006, AGILE 2006*. pp. 76–88. IEEE, Minneapolis (2006). <https://doi.org/10.1109/AGILE.2006.1>
15. Larman, C., Basili, V.R.: Iterative and incremental developments. A brief history. *Computer* **36**(6), 47–56 (2003). <https://doi.org/10.1109/MC.2003.1204375>
16. Lohan, G., Lang, M., Conboy, K.: Having a customer focus in agile software development. In: Pokorny, J., et al. (eds.) *Inf. Syst. Dev.*, pp. 441–453. Springer, New York (2011)
17. Mann, C., Maurer, F.: A case study on the impact of scrum on overtime and customer satisfaction. In: *Agile Development Conference (ADC 2005)*, pp. 70–79. IEEE Comput. Soc, Denver (2005). <https://doi.org/10.1109/ADC.2005.1>
18. Margini, A., Cutrona, G., Fantuzzi, C.: Comparison of different agile methodologies and fit assessment in an industrial context. *Int. J. Adv. Res.* **5**(7), 673–690 (2017). <https://doi.org/10.21474/IJAR01/4768>
19. Marshall, M.N.: Sampling for qualitative research. *Fam. Pract.* **13**(6), 522–526 (1996)

20. Martin, A., Biddle, R., Noble, J.: The XP customer role in practice: three studies. In: Agile Development Conference, pp. 42–54. IEEE, Salt Lake City (2004). <https://doi.org/10.1109/ADEV.2004.23>
21. Mishra, D., Mishra, A.: Effective communication, collaboration, and coordination in eXtreme programming: human-centric perspective in a small organization. *Hum. Factors Ergon. Manuf.* **19**(5), 438–456 (2009). <https://doi.org/10.1002/hfm.20164>
22. Molokken-Ostfold, K., Furulund, K.M.: The relationship between customer collaboration and software project overruns. In: AGILE 2007 (AGILE 2007), pp. 72–83. IEEE, Washington, DC, August 2007. <https://doi.org/10.1109/AGILE.2007.57>
23. O’heocha, C., Conboy, K.: The role of the user story agile practice in innovation. In: Abrahamsson, P., Oza, N. (eds.) LESS 2010. LNBP, vol. 65, pp. 20–30. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-16416-3\\_3](https://doi.org/10.1007/978-3-642-16416-3_3)
24. Raison, C., Schmidt, S.: Keeping User Centred Design (UCD) alive and well in your organisation: taking an agile approach. In: Marcus, A. (ed.) DUXU 2013. LNCS, vol. 8012, pp. 573–582. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39229-0\\_61](https://doi.org/10.1007/978-3-642-39229-0_61)
25. Robson, C.: *Real World Research*, 2nd edn. Blackwell Publishing, Oxford (2002)
26. Stotts, D., Williams, L., Nagappan, N., Baheti, P., Jen, D., Jackson, A.: Virtual teaming: experiments and experiences with distributed pair programming. In: Maurer, F., Wells, D. (eds.) XP/Agile Universe 2003. LNCS, vol. 2753, pp. 129–141. Springer, Heidelberg (2003). [https://doi.org/10.1007/978-3-540-45122-8\\_15](https://doi.org/10.1007/978-3-540-45122-8_15)
27. Sutherland, J., Viktorov, A., Blount, J., Puntikov, N.: Distributed scrum: agile project management with outsourced development teams. In: 2007 40th Annual Hawaii International Conference on System Sciences (HICSS’07), pp. 274a–274a. IEEE, Waikoloa (2007). <https://doi.org/10.1109/HICSS.2007.180>
28. Vlaanderen, K., Jansen, S., Brinkkemper, S., Jaspers, E.: The agile requirements refinery: applying SCRUM principles to software product management. *Inf. Softw. Technol.* **53**(1), 58–70 (2011). <https://doi.org/10.1016/j.infsof.2010.08.004>
29. Whittemore, R., Chase, S.K., Mandle, C.L.: Validity in qualitative research. *Qual. Health Res.* **11**(4), 522–537 (2001). <https://doi.org/10.1177/104973201129119299>
30. Williams, L.: Agile software development methodologies and practices. In: *Advances in Computers*, vol. 80, pp. 1–44. Elsevier (2010). [https://doi.org/10.1016/S0065-2458\(10\)80001-4](https://doi.org/10.1016/S0065-2458(10)80001-4)