

Selecting the Best Agile Team for Developing a Web Service

Marta Kristin Larusdottir and Marcel Kyas

Reykjavik University, Menntavegur 1, 102 Reykjavik, Iceland
marta@ru.is; marcel@ru.is

Abstract

Selecting a good agile software development team to develop a particular software is a complex issue for public authorities. This selection is often based on the estimated total cost of the project in an official request for proposals. In this paper we describe an alternative approach where three performance factors and the estimated cost were evaluated and weighted to find the best agile team for the project. The performance factors included: team collaboration, user experience focus, user stories delivery and the quality of the code. Teams that fulfilled predefined technical requirements were invited to take part in workshops. We describe the process of evaluating the three performance factors during and after the workshops and the results of the evaluations. The team that focused on one user story during the workshop and emphasised user experience, accessibility and security issues got the highest rating and were selected for the project.

Keywords

User experience · Accessibility · Security · Agile development · Team collaboration

How to cite this book chapter:

Larusdottir, M.K. and Kyas, M. 2020. Selecting the Best Agile Team for Developing a Web Service. In: Loizides, F., Winckler, M., Chatterjee, U., Abdelnour-Nocera, J. and Parmaxi, A. (eds.) *Human Computer Interaction and Emerging Technologies: Adjunct Proceedings from the INTERACT 2019 Workshops*. Pp. 289–302. Cardiff: Cardiff University Press. DOI: <https://doi.org/10.18573/book3.al>. License: CC-BY 4.0.

1 Introduction

When public authorities want to make new software systems to be used by citizens and employees for solving various tasks they often negotiate with software companies for developing the software. The selection of the software company for making the software needs to be free and open for competition according to European Union legislation, so the public authorities must issue a public request for proposal (RTF). Typically the RTF contains two sections: (1) the requirements and needs for the system to be developed, and (2) the selection criteria [12]. Often the selection criteria is based on the cost solely, so the software companies estimate the hours needed to be able to develop the software fulfilling the requirements and needs stated. The company with the lowest prize gets the job [12]. In a case study of four software companies in Denmark developing for public authorities, the software companies focused on what the public authorities are willing to pay for and what they wanted to citizens to be able to do [2]. So the software companies did not include quality factors like user experience (UX) or security issues, in their proposal, if it was not requested in the RFT.

In some cases the selection criteria is based on both the prize and quality factors, so the price could weight 60% and the quality criteria 40% for example [12]. Requirements for quality factors, like user experience (UX) and security, can be included in the requirement section of the RTF defining the level of the UX and security in the developed system. The requirements can also be included in the selection criteria, defining how much weight in the selection process the UX and security factors have [22]. Typically, the usage of particular methods like user testing and the frequency of using those methods would be stated in the selection criteria. Another option would be that the public authority may state performance criterias for the users, for example that the users will be able to accomplish a particular task within a particular time limit [22]. One possibility is to base the selection criteria on the competences of the software team getting the job, but that is not frequently done. The selection criteria should state the wanted knowledge, skills and competences of the team, in that case. Possibly, the criteria could also include the focus on quality aspects that the team should have. In any case, the objective of the process is to find the best team for the job according the predefined criteria and thereby get the best outcome for the money spent.

There are many aspects that affect an project outcome. A study of four similar software teams developing software to fit the same needs, described 1 to 6 variation in the prizes of the outcome [21]. The teams were similar in technical competences. The quality of the outcome was also evaluated and the team with the next lowest price scored best on the three quality aspects in the study, usability, maintainability and reliability. That team had one project manager, one developer and one interaction designer in the team, but the other teams had two developers and one project manager. The best team used intermediate

process models for the development, with analysis and design in the first four weeks, then implementation from week 4 to 10 and testing in the last six weeks of the project [21].

In this paper we describe an approach, where the performance of five software teams was evaluated as a part of the selection criteria for selecting the best agile team for making a web service. The performance factors included: team collaboration, user experience, user stories delivered and quality of code including accessibility and security. The performance factors were evaluated during and after a one day workshop with the team, where the teams were observed and their deliverables reviewed. The performance factors weighted 70% and the cost 30% in the selection criteria for the best agile team.

2 Related Work on the Performance Factors

In this section we briefly describe the related literature on the performance factors evaluated in this study. First we give a brief overview of agile development and team collaboration, we explain the format and usage of user stories and then we briefly describe the concept of user experience and code quality.

2.1 *Agile Development and Team Collaboration*

The agile process Scrum [20] has gained popularity in the software industry in recent years. According to an international survey, Scrum was the most popular process of the agile processes with more than 50% of the IT professionals surveyed were using it [23].

A similar trend is seen in the software industry in Iceland, but the lean process Kanban [17] has also been gaining popularity lately [15].

A characteristic of Scrum is the observation that small, cross-functional teams historically produce the best results. Scrum is based on a rugby metaphor in which the team's contribution is more important than each individual contribution. Scrum teams typically consists of people with three major roles: 1) a Scrum Master that acts as project manager/buffer to the outside world; 2) a Product Owner that represents stakeholders, and 3) a team of developers (less than 10). One of the twelve principles behind the agile manifesto is: "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation" [16]. In agile development the teams should collaborate openly and all the team is responsible for delivering a potentially shippable product after each sprint.

Some of the more important artifacts and ceremonies with-in Scrum is the Sprint, which defines 15–30 days iteration, the Product backlog of requirements described by user stories and managed by the Product Owner and the Daily Scrum meeting, which is the daily meeting for the team and the Scrum Master to plan the work of the day and report what was done the day before [20].

2.2 *User Stories*

In Scrum, the user requirements are usually described by user stories. The most common format for describing a user story is: “As a [user role], I want to [do some task] to [achieve a goal]” [4]. The user stories are used to describe the requirements for the whole system being developed kept in the Product Backlog. During the Sprint planning meeting, the team, the Scrum Master and the Product Owner select the user stories that the team will work on during the next sprint in accordance to how many user stories it is possible to implement during the time of a sprint. The Product Owner describes the priorities of the user stories, so the most important user stories will be selected for the particular sprint according to the Product Owners criterias. During the daily Scrum meeting, the team members report what user stories and tasks they will be working on during the day and what the finished they day before.

2.3 *User Experience*

UX has gained momentum in computer science and is defined in the ISO 9241-210 in the following way [10]: “Person’s perceptions and responses resulting from the use and/ or anticipated use of a product, system or service”. Researchers agree that UX is a complex concept, including aspects like fun, pleasure, beauty and personal growth. An experience is subjective, holistic, situated, dynamic, and worthwhile [8]. A recent survey on what practitioner’s think is included in the term UX shows that respondents agreed that user-related factors, contextual factors and temporal dynamics of UX are all important factors for defining the term UX [14]. The temporal dynamic of UX also reached consensus amongst the respondents.

Many methods have been suggested for active participation of users in the software development process with the aim of developing software with good user experience. Some of the methods for focusing on either the expected UX or the UX after users have used a particular system, including interviews with users, surveys, observations and user testing [19]. IT professionals rated formal user testing as the most useful method for active participation of users in their software development for understanding the UX of the developed system [11].

2.4 *Quality of Code Including Security and Accessibility*

Code quality is generally hard to define objectively. Desirable characteristics include reliability, performance efficiency, security, and maintainability [5]. Metrics to assess code quality usually include volume of code, redundancy, unit size, complexity, unit interface size, and coupling [1, 9]. The process of measuring properties like complexity and the decision on what unit size is acceptable depend on the context and is often subjective.

Accessibility of web application is typically realised by conforming to the WCAG 2.0 recommendation [3]. Following these recommendations allows a web page to be interpreted and processed by accessibility software. For example, by a.o. preferring relative font sizes over absolute ones allows the web page to be rendered in any font size and making it accessible to users with visual impairments. The WCAG is seen as an important part of making web pages accessible [13].

Indeed, for any web application and any mobile application used by the public sector in the European Economic Area must conform to the WCAG [6].

3 The Case – The Financial Support RTF

Reykjavik city has decided to make the digital services easy to use for all the citizens of Reykjavik. The motivation came from two new employees, that wanted to change the web services to being more user centred. One of the first projects for this attempt had the objective to make the application for financial support more usable to citizens, but to focus also on security and reliability of the code. An official request for proposals was made to select “the best” team for taking part in developing a web service in collaboration with IT professionals at Reykjavik city. One of the constraints was that the team had to follow an agile development process similar to Scrum, by using user stories, conducting daily Scrum meetings and focus on the values of agile team work and collaboration.

The teams that submitted a proposal were evaluated according minimal technical requirements and their performance and delivery after a one day workshop. There were five steps in the selection process: a) First the team submitted a proposal, b) The applying teams were evaluated according to the minimum technical requirements, c) the teams fulfilling the technical requirements were evaluated according to performance criteria, d) the hourly prices of each team member were evaluated and e) the final selection of a team was decided. In this section we describe the minimal technical requirements for the teams and the three performance factors evaluated during and after the one day workshops.

3.1 *The Minimal Technical Requirements*

The minimal technical requirements were described in the request for proposals document. The teams had to provide at least 5 team members, whereof at least:

- a) 2 members had to be skilled backend programmers, which had experience in writing code that was tested for security. For confirming these skills, the team members were asked to deliver a list of projects were they had worked on security issues for the system. They also had to list at least 5 software projects that they had been involved in. They had to be experienced in automated testing and have knowledge of .NET programming.

- b) 1 member had to be a user interface programmer. This persons had to have the experience of making apps or web services that fulfilled the accessibility standard, European Norm EN 301 549 V1.1.2 [7] that includes the WCAG 2.0 Level A and Level AA and are scalable for all major smart equipment and computers. This person had to describe his/her involvement in five software development projects.
- c) 1 member had to be interaction designer or a UX specialist. This member had to have taken part in developing at least 5 software systems, (apps or web services), with at least 100 users each. They should describe their experience of user centred design with direct contact with users and what methods they had used to integrate user in the development.
- d) 1 member should had to be an agile coach or a Scrum Master. To fulfill this, the person had to have led at least one team with at least three members with at least 10 two week sprints. This member should describe his experience regarding coaching team members.

3.2 *The Workshop Organisation*

Five teams fulfilled the above minimum technical requirements. Each of them were invited for a one day performance workshop. The workshops took place at an office at the IT department of Reykjavik city in October and November 2018.

The teams got four user stories to as possible tasks to work on during the workshop. The user stories were the following:

1. As a citizen of Reykjavik that has impaired intellectual ability I want to be able to apply for financial assistance via web/mobile so that I can apply in an simple and easy-to-understand manner.
2. As a employee of Reykjavik city with little tech know-how I want to be able to see all applications in a “employee interface” so that I have a good overview of all applications that have been sent.
3. As a Reykjavik city employee which is colorblind I want to be able to send the result of the application process to the applicant so that the applicant can know as soon as possible if the applicant is eligible for financial assistance.
4. As a audit authority for financial assistance I want to be able to see who has viewed applications so that I can perform my audit responsibility.

The workshops were organised by a project manager at Reykjavik city. The schedule was the following:

1. The team got an one hour introduction to the schedule of the day and to the work environment at Reykjavik city, the services and systems, the organisation and work practices. Also the user stories were introduced briefly.

2. The teams were asked to do a daily Scrum meeting for 15 minutes for selecting the tasks for the day and to organise the day for 15 minutes. The experts focusing on team collaboration and UX focus observed this part of the workshops.
3. The teams worked on developing their deliverables during the day.
4. The last 45 minutes of the day, the teams were asked to present to all the involved experts and the organising team, their work practices and their deliverables. The teams could plan these 45 minutes as they preferred. They had been introduced to the performance factors that were being evaluated, so some of the teams deliberately organised the presentation according to these factors.

3.3 The Performance Factors Evaluated During and After the Workshops

The workshops had the goal of evaluating the following three performance factors:

1. The teams collaboration and user experience (UX) focus
2. Their delivery of user stories
3. The quality of the code delivered

An evaluation scheme was conducted for each of the three factors. Four external experts were asked to conduct the evaluation. The team collaboration and UX focus contained four subfactors and in total these gave the maximum of 25 points. These were evaluated by two external experts by observing the teams twice during the one day workshop. The delivery of user stories and the quality of the code delivered were evaluated after the workshop. Two external experts in security issues and performance were asked to review the code delivered. The user stories delivered gave maximum 10 points and the quality of the code 35 points. In total these three performance factors added up to 70 points. The hourly price for the team members could give a maximum of 30 points. Experts at Reykjavik city reviewed the hourly prizes. The agile team could get 100 points in total, if they got the maximum points for all the three performance factors and the hourly prizing. We will describe the process of the data gathering for evaluating the three performance factors resulting from the workshops in the next section.

4 Data Gathering for Evaluating the Performance Factors

In the following we will describe the process of gathering data to be able to evaluate the team collaboration, the user stories delivered and the quality of the code.

4.1 Data Gathering for Evaluating the Team Collaboration and UX Focus

Two experts in team collaboration and UX focus were asked to evaluate this performance factor. Four subfactors were defined:

1. How well did the team perform at the daily meeting (max 4 points) ?
2. How problem solving oriented was the team (max 8 points)?
3. How much did the team emphasize UX (user experience) (max 8 points)?
4. How well did the team present their work at the end of the workshop (max 5 points)?

The two experts observed the teams during an half an hour session in the morning, when the teams had a daily Scrum meeting and when selecting tasks for the day. The experts took notes and evaluated the first subfactor. They tried to keep silent and not ask questions so the five workshops would be as similar as possible.

Forty five minutes were used as the last part of the workshop for presenting the work practices that the team used during the day and the deliverables. The two experts observed the presentation and took notes. The experts only asked, if there were issues, which the experts were about to evaluate, that were not mentioned during the presentation, to have better information on all the performance factors.

There was a short evaluation meeting with all the experts involved and the organising team at Reykjavik city right after each workshop. The goal was to discuss the first impression of the workshop of that day. Each of the experts rated the teams within 48 hours on the four subfactors and noted an argument for each of the ratings. The two experts met shortly after that evaluation and discussed their individual ratings and made a consolidated rating for the team that was sent to the project manager of the workshops. When all the teams had been evaluated the two experts met again to make the final comparison of all the ratings and made the final version of the ratings that was sent to the project manager of the workshops as the final rating from the experts.

4.2 Data Gathering for Evaluating the User Stories Delivered

A second team of two experts was assigned the task of evaluating whether the user stories had been successfully implemented. The second team had to rely on the documentation of the submission to identify the code that was supposed to implement the feature described by the user story and the test cases for that story.

Each agile team submitted their project as a dump of a git repository. Some teams also submitted sketches, mock-ups and photographs of all documentation written down during the workshop day. In addition, some teams kept a test instance of their system running for the two experts to test.

The evaluation criteria were:

1. Did the submitting team make a claim that a user story was implemented? Lacking such a claim the experts would assume that the story was not implemented.
2. Did the submitting team document what functions were used to implement the user story? The experts would look at the code only for names that related to concepts in the user story.
3. Did the submitting team provide test cases to test the user story?

The verdict for each user story was pass or fail. The score was with respect to the maximum achieved by all teams. One team managed to implement 3 stories, which gave the maximum number of 10 points. All other teams scored a fraction of three, according to the number of stories they achieved. A finer distinction than pass and fail was rejected, because the experts could not agree on how that should be done objectively, and they felt that it was not worth the effort.

4.3 Data Gathering for Evaluating the Quality of the Code

As mentioned above, each team submitted their code as a clone of a git repository. This enabled the experts to evaluate the way the teams were documenting their software development process. The properties that the two experts evaluated were:

1. Quality of the documentation in the code
2. Quality of the log messages in version control
3. Quality of web accessibility
4. Error handling in the interface
5. Error handling in code
6. Functionality of the database scripts
7. Correct use of the model-view-controller pattern
8. Error free functionality

Points 3 and 4 were most relevant to the interaction with the user. The experts used the WAVE web accessibility evaluation tool to assess the quality of web accessibility and to check compliance with WCAG 2.0 at levels A and AA [7]. The experts investigated the choice of colors by hand and by using filters to simulate how color vision deficient users would see the web site. Overall, all submissions had some issues with web accessibility, like laying out information in the wrong order, missing alt tags for images, and so forth.

The two experts referred to the way erroneous behaviour is conveyed by the user for evaluating the error handling in the interface. The experts checked whether the error messages were displayed in a meaningful manner, how an encountered error would be addressed, and whether a pointer to assistance was provided.

No formal audit was defined concerning security. The evaluation of secure coding standards was guided by the documents of the Open Web Application Security Project [18]. The two experts audited the submitted projects for possible injection attacks and sufficient logging and monitoring, as well as security configuration. However, ensuring security of the system and verifying that security goals have been met was outside of the scope of the evaluation.

5 Results and Discussion

The results from the evaluations of the performance factors are shown in table 1.

Team A got the highest number of points in total for the three performance factors. This team had an interesting approach. They only focused on one user story, which was user story 1, during the workshop, but all the other teams selected more than one user story to focus on. This is why Team A got the lowest number of points for the user stories delivered.

The user story that Team A selected was the only story that included the citizens of Reykjavik, the other three user stories included employees of Reykjavik city. Team A got the highest number possible for team collaboration and UX focus. This was the only team that contacted a domain expert to understand the needs of the this particular group of citizens. They called a person at the service center to interview her/him to enhance their understanding of the needs of the user group. One of the team members also went to the service center, which was in the same building, and tried out how the application process was during the day of the workshop. The other teams did not contact any people outside the team for gathering information on the users and only imaged how the users would behave.

The team collaboration factors were more similar for the teams, but still there were some differences. For some teams we did not see much communication during the daily Scrum meeting and the organizing meeting, so the team

Table 1: The total points that each team received for the three performance factors evaluated.

Performance factor	Team A	Team B	Team C	Team D	Team E
Team collaboration and UX focus max 25 points	25,0	12,4	9,4	7,6	19,4
Delivery of user stories max 10 points	3,3	6,7	6,7	10,0	6,7
Quality of code max 35 points	22,0	16,2	18,0	22,4	22,4
Total max 70 points	50,3	35,3	34,1	40,0	48,5

members did sit by their computers and work individually. This is against the fundamental rules of agile, where team communication and collaboration is vital [16].

The aggregate score for the quality of the code had much less variation between the teams. Teams A, D, and E received almost the same score on code quality. Each of these teams were very competent. The experts observed some differences in each of the 8 categories among these teams but the differences averaged out.

Team B did not document their code and did not trace decisions to requirements and stories. Exceptional behaviour was not handled, and no tests were provided. Team C did not document parts of their code well, had many non-descriptive messages like “log in stuff” as commit messages to their version control systems, and did not take care of exceptional code paths. One error message displayed to the user was: “An unexpected error happened” and some errors were silently ignored. They aimed to implement three of the four stories, but only managed to finish two of them. Team D worked on a technical level, planning to implement all the user stories with a high standard of quality. At the same time, they chose the simplest stories. Team D and E received the same scores on code quality but aspects of code quality differed, e.g., team E had worse documentation of their process and the code, but handled web accessibility, error handling, and software architecture better than Team D.

To summarize, it was surprising for all the experts how much variation there was in how the teams worked and what they delivered. All the teams included IT professionals with the technical requirements fulfilled. Team A got the job since they got the highest score of the summary of all the performance factors and their prize estimations were in line with the other teams, so they got the highest total score and the job. They were the only team that reached out to understand the users of the service, while focusing on the code quality in parallel.

References

1. Baggem, R., Correia, J. P., Schill, K., and Visser J.: Standardized code quality benchmarking for improving software maintainability. *Software Quality Journal*, 20(2), 287–307 (2012). doi: <https://doi.org/10.1007/s11219-011-9144-9>
2. Billestrup, J., Stage, J., & Larusdottir, M.: A Case Study of Four IT Companies Developing Usable Public Digital Self-Service Solutions. In *The Ninth International Conference on Advances in Computer-Human Interactions*, (2016).
3. Caldwell, B., Cooper, M., Guarino Reid, L., and Vanderheiden, G.: *Web Content Accessibility Guidelines (WCAG) 2.0. W3C*, (2008).
4. Cohn, M.: *User Stories Applied*. O'Reilly Media (2004).

5. Curtis, B., Dickenson, B., and Kinsey, C. CISQ Recommendation Guide (2015) <https://www.it-cisq.org/adm-sla/CISQ-Rec-Guide-Effective-Software-Quality-Metrics-for-ADM-Service-Level-Agreements.pdf> (last accessed June 27, 2019).
6. Directive (EU) 2016/2102 of the European Parliament: Directive (EU) 2016/2102 of the European Parliament and of the Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies (Text with EEA relevance). Homepage: <http://data.europa.eu/eli/dir/2016/2102/oj>, last accessed 27th June 2019.
7. European Telecommunications Standards Institute: Accessibility requirements suitable for public procurement of ICT products and services in Europe, EN 301 549 V1.1.2 (2015). Retrievable: https://www.etsi.org/deliver/etsi_en/301500_301599/301549/01.01.02_60/en_301549v010_102p.pdf
8. Hassenzahl, M. (2013). User experience and experience design. In: Soegaard, Mads and Dam, Rikke Friis (Eds.). The encyclopedia of human-computer interaction, 2nd Ed. Århus, Denmark: The Interaction Design Foundation.
9. Heitlager, I., Kuipers, T., & Visser, J.: A practical model for measuring maintainability. In 6th international conference on the quality of information and communications technology (QUATIC2007), pp. 30–39. IEEE Computer Society. (2007).
10. International organisation for standardisation: ISO 9241-210:2010. Ergonomics of human-system interaction Part 210: Human-centred design process for interactive systems, (2010).
11. Jia, Y., Larusdottir, M. K., & Cajander, Å.: The usage of usability techniques in Scrum projects. In Human-Centered Software Engineering (pp. 331–341). Springer Berlin Heidelberg, (2012).
12. Jokela, T., Laine, J., & Nieminen, M.: Usability in RFP's: The current practice and outline for the future. In International Conference on Human-Computer Interaction (pp. 101–106). Springer, Berlin, Heidelberg, (2013).
13. Kelly, B., Sloan, D., Phipps, L., Petrie, H., & Hamilton, F.: Forcing standardization or accommodating diversity?: A framework for applying the WCAG in the real world. In Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A) (pp. 46–54). ACM., (2005), <https://doi.org/10.1007/s11219-011-9144-9>
14. Lallemand, C., Guillaume G., Vincent, K.: User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. *Computers in Human Behavior* 43: 35–48, (2015).
15. Law, E. L., Lárusdóttir, M. K.: Whose experience do we care about? Analysis of the fitness of Scrum and Kanban to User Experience, *International Journal of Human-Computer Interaction*, Vol. 31 (9), pg. 584–602, (2015).

16. Manifesto for Agile Software Development homepage, <https://agilemanifesto.org/>, last accessed 27th June, 2019.
17. Ohno. T.: The Toyota Production System: Beyond Large-Scale Production. Productivity Press, (1988).
18. OWASP Homepage, <https://www.owasp.org>, last accessed 27th June, 2019.
19. Preece, J., Rogers, Y., Sharp, H.: Interaction design: beyond human-computer interaction, 5th edition, John Wiley and sons, Inc., (2019).
20. Sjøberg, D. I.: The relationship between software process, context and outcome. Proceedings of International Conference on Product-Focused Software Process Improvement (pp. 3–11). Springer, Cham (2016).
21. Schwaber, K.: Scrum development process. In: SIGPLAN Notices, 30(10), (1995)
22. Tarkkanen, K., Harkke, V.: Evaluation for Evaluation: Usability Work during Tendering Process. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15). ACM, New York, NY, USA, 2289–2294, (2015). DOI: <https://doi.org/10.1145/2702613.2732851>
23. Version One (2019): 13th Annual State of Agile survey. Online at: <https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>, (last retrieved 27th of June, 2019)