JOURNAL OF SCIENCE
SAKARYA UNIVERSITY

Sakarya University Journal of Science

ISSN 1301-4048 | e-ISSN 2147-835X | Period Bimonthly | Founded: 1997 | Publisher Sakarya University | http://www.saujs.sakarya.edu.tr/en/

Title: Agile Methods in Game Programming based on Scrum

Authors: Şahin MERCAN, Yaşar BECERİKLİ
Recieved: 2019-12-12 20:42:54

Accepted: 2020-07-02 10:35:58

Article Type: Research Article
Volume: 24
Issue: 5
Month: October
Year: 2020
Pages: 882-891

How to cite
Şahin MERCAN, Yaşar BECERİKLİ; (2020), Agile Methods in Game Programming based on Scrum. Sakarya University Journal of Science, 24(5), 882-891, DOI: https://doi.org/10.16984/saufenbilder.658752
Access link
http://www.saujs.sakarya.edu.tr/en/pub/issue/56422/658752

Sakarya University Journal of Science 24(5), 882-891, 2020

Agile Methods in Game Programming based on Scrum

Şahin MERCAN*[1], Yaşar BECERİKLİ[2]

Abstract

Game programming requirements, which is one of the problems of game programming in recent times, is constantly changing. Therefore, the process we are conducting while programming the game does not keep up with this change leads to the failure of the product we develop or the result of higher costs. Therefore, the use of agile programming, which brings a more dynamic and modern solution to changing environment conditions, is increasing day by day and the project teams are transitioning to these processes. In this thesis, firstly historical development, content, rules and methods of Agile methods are presented with literature researches. In the next section, Grumpy Ball is applied by using Scrum method which is agile programming method. Since the agile methods provide dynamic solutions, an end date cannot be determined Therefore, the aim of this article is to determine the exact date of a game application using agile method. For this reason, our project is divided into specific parts and the last periods given to these parts were compared according to certain criteria. In the light of the results, determinations were made.

Keywords: agile methods, game programming process, game project management, scrum

1. INTRODUCTION

It is foreseen that there will be changes that may occur in the progressing stages of the software processes. For this reason, traditional methods are based on a comprehensive study at the very beginning of the software processes to determine all the requirements that may occur and to prevent any changes that may occur in the future stages. However, despite the rapidly changing, evolving environment and market conditions, project requirements are changing more and more quickly.

Therefore, it is almost impossible to determine all the needs at the beginning of the project.

*Corresponding Author:sahinmrcn@gmail.com
[1]Kocaeli University, Graduate School of Natural and Applied Sciences, Kocaeli, Turkey.
ORCID: https://orcid.org/0000-0002-8646-6834
[2]Kocaeli University, Department of Computer Engineering, Kocaeli, Turkey, ybecerikli@kocaeli.edu.tr,
ORCID: https://orcid.org/0000-0002-2951-7287

Therefore, it is expected that the software development process will have the features that will meet these changes in less time and with less cost. Therefore, agile software development methods have emerged as an alternative to traditional software development methods [2, 7]. If agile methods cannot be applied to the whole project, it can only be agile where necessary or

- Interpersonal and interpersonal relationships in place of process and development tools [3].
- Detailed documentation instead of running software [3].
- Contract negotiations instead of customer cooperation [3].
- Instead of following the plan to keep up with changes [3].

The above common opinion is included in the Agile Alliance Development Agenda and supported by the Agile Alliance and is considered the key to effective software development [7, 3].

1.1. Agile Software Development Features

- Focus and perspective: Agile methods have a project perspective. The focus of the agile methods is the project and project team [12].
- Management: Agile methods are also traditionally made in the form of coaching (command control) and not management. In order for the team to advance, the coach assists the team in eliminating the barriers [12].
- Planning: Agile methods can be planned on more than one level. Project planning at the top level and sprint planning at the detail level are carried out. This brings about flexibility and re-planning [12].
- Learning: Agile methods also occur during learning, project or sprint and activities [12].
- Evaluation: Agile methods are also made through evaluation, results and product (customer satisfaction, project returns) [12].

agile approaches can be tailored to the project and the structure of the development team [11, 10, 6].

In 2001, after the creation of the Agile Software Development Manifesto, agile software principles were introduced. These principles are given below [5].

- Personal development: Agile methods are team and individual oriented (human is above the process). The team is created by selecting people who are good at their work [12].
- Project lifecycle activities: Agile Methods support simultaneous development, test sprints and peer reviews as needed. What is important is that the product is not working as early as possible. With the assumption that the cost of delay and error will be low, incremental delivery method is adopted [12].
- Estimation: In order to accelerate development activities, the development phase is divided into conditions, design and solutions evolve; thus, an attempt is made to achieve a predictable development speed. The aim here is to avoid the need to narrow down the scope of sprint by examining the time and calendar limit at the end of the sprint at the end of the sprint [12].

1.2. Agile Software Development Methods

Although different agile methods share many features, there are important differences between them. Some are committed to working in project management and collaboration (Adaptive Software Development, Scrum, and Dynamic System Development), while others have focused on Extreme Programming software development work [7].

The most important agile methods are Adaptable Software Development, Scrum, Extreme Programming, Crystal Methods, Feature-Guided Development, Open Source Development and Dynamic System Development [7]. We used the Scrum method in this article, we just examined the Scrum method.

In 1986, Hirotaka Takeuchi and Ikujiro Nonaka defined a new approach in product development where all process phases overlap and the team could work together at different stages. The basis of this new approach is based on case studies of automotive, computer, copier, printer manufacturers, and basically aimed to bring speed and flexibility to production. They gave this approach the name of holistic and rugby, the whole process being carried out by a team that was cross-tasked and working for a single purpose [19].

In the early 1990s, Ken Schwaber applied this approach to Easel in its own company and named it Scrum [15]. In 1995, Sutherland and Schwaber jointly presented the first public presentation of the Scrum methodology in a study in Texas [16]. Scrum is a methodology that has been applied extensively in recent years and produces successful results [2].

The Scrum approach was developed to manage the system development process. Scrum is an empirical approach that implements the ideas of industrial process control theory for the system development process, resulting in an approach that redefines ideas of flexibility, harmony and efficiency [1]. The software deals with how to manage rather than how to develop it [12]. This method facilitates management in small-scale projects, directs teams to small pieces in large-scale projects and requires constant reviews [12]. Through these reviews, teamwork is clearly seen, which increases collaboration and communication within the team [12]. It can be applied at the beginning of the project or in the middle of production or when the production is problematic. It is used not only in software development but also in the production of financial and medical products [4].

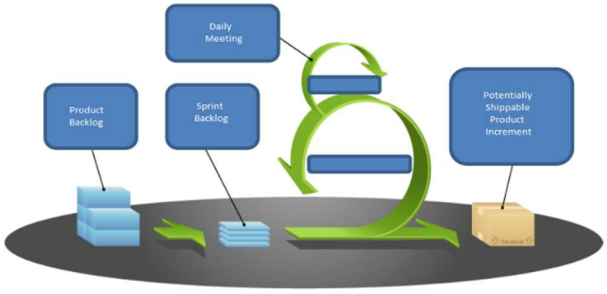

Figure 1 Scrum method

In the Scrum method, planning, job descriptions and documentation spend less time focusing on how the project team can develop a flexible system that is compatible with changing conditions [18]. This method includes sprint planning, sprint backlog identification and Scrum meetings [18]. The loop is restarted by testing the intermediate product customer (job / product owner) as a result of the flag delivery (sprint), and the cycle is restarted and all process steps are reapplied. Figure 1. A second sprint is not started before a sprint is complete (9).

2. SCRUM METHODOLOGY

Scrum initial product requirements will be created and used in architecture, technical details, contracts and so on. It is a project management methodology that starts with the preparation phase, the software is developed with iterative sprints, the product is presented to the customer with intermediate distributions and the product is presented to the customer with the tests and documentation after the final sprint.

In Scrum, organizations perform iterative and incremental development between start and end stages as in Figure 2.

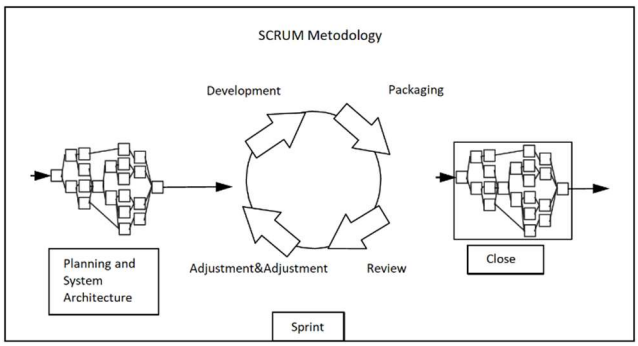

Figure 2 Scrum stages

The start and end stages (planning and closure) consist of defined processes. All process steps and input outputs are defined. With some iterations, the flow at the planning stage is linear [13].

Any changes in the intermediate product after each sprint are re-planned according to the severity of the next conditions. By doing this, it provides the desired product to be produced by providing the desired requirements instead of the

requirements with less value in the product increments [17].

The detailed diagram of Scrum is as in Figure 3. below;

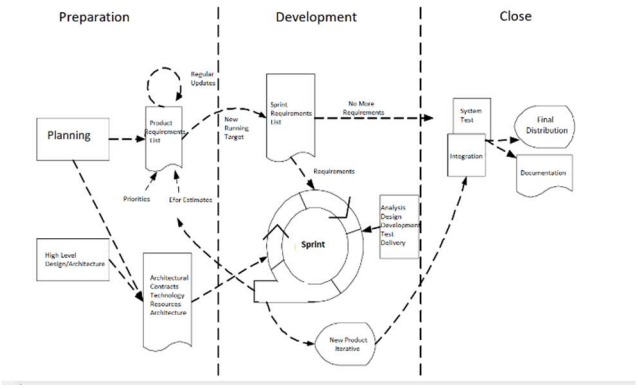

Figure 3 Overview of process in scrum

The development phase consists of iterative development steps as shown in Figure 4. Management deals with time, competition, quality, functionality, completed steps and shutdown phase. This approach is also known as simultaneous engineering [13].

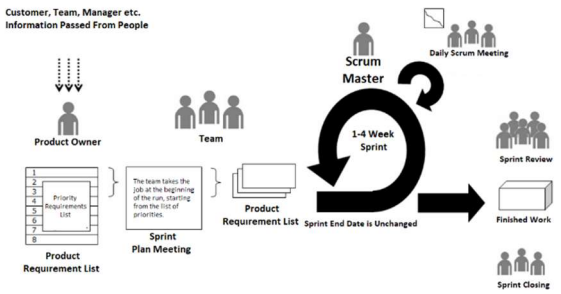

Figure 4 Scrum methodology development phase [17]

In each project there is a time required to understand whether the project is good or not. This period should not be too long. If it is too long, the risks are greater. The project's predictability should be checked at least once a month, except that the project may be out of control or unpredictable [14]. The day starts with a sprint planning meeting and a daily Scrum meeting is held every day.

3. APPLICATION

In this section, game programming is done using Scrum method.

The game basically consists of a ball and rings. Sky and clouds are as background. The speed of the clouds determines the speed of the game. The game progresses faster. Because the game is in the sky, the ball tends to fall down constantly. When you click on the screen, the ball bounces. The rings move from left to right. There are 3 rings in red, green and blue. If the color of the ball is the same as the color of the ring, the user will try to pass the ball through that ring.

The user is given 3 lives at the beginning of the game. There are stars in the right ring every 10 rings. If the user receives this star, one more can will be added.

In addition, our game consists of 4 main sections. In the first part the color of the ball is changed randomly from the colors of fixed rings. In the second part the color of the ball is fixed, the color of the rings is changing randomly. In the third part, the color of both the ball and the rings changes randomly. In the last part, the rings move in addition to the third part.

The screen shows how many rings the user has left.

This is the scenario of our game.

One of the advantages of the agile method has been the flexibility, as it progresses continuously. However, this flexibility constantly changes the existing properties and even adds new features. Therefore, it is not possible to determine an end date of the application. In this application, we are working to determine the end date by using agile method. For this purpose, we will try to make an inference by separating the time estimates at the planning stage and the time elapsed at the end time according to the items. Below we explain the stages of our application. And then we analyzed the results according to the data.

3.1. Pre-Production

The main features of the game were determined at this stage. These are composed of rings and a wholesale in our game. There will be 3 rings of different colors each time. If the ball is the same color as the ball will be tried to pass through the ring. The game will end at a constant speed and a total of 100 rings. The basic principles of our game are determined in this way.

Our game is coded in c # using Unity platform. Desktop, Android and IOS platforms can be played. There is no prototype production. The construction of the pole was started. In order to better test the success of the agile method, the changes in the place where it is necessary in the game have been realized and changed after being realized.

In the later stages of the game, the structure of the game changed a bit as the game was made. The changes were as follows.

What is considered in the changes made here is that the game is more verifiable. For this reason, 3 user rights are given to the user in the game. However, this game has not been easy to finish. Therefore, one user is given a life in every 10 rings. In addition, the game speed is increased as the game progresses to save the game from being monotonized. In order to be able to play the game for a longer period of time, the game was partitioned.

A total of 4 sections. The color of the ball in these sections by changing the color of the ring has become increasingly difficult.

The above changes were related to the game scenario. As a design, the rings in the game were redesigned in 3D.

3.2. Game design documentation

This is a very important issue. Because if something is omitted or neglected in the design, it can lead to the loss of some features in our game. And this leads to a decrease in the "entertainment" factor, which is the most important feature of the game.

3.3. Product requirement list

This is the section where a list of objects will be prepared according to the game scenario. According to the scenario of our game objects are planned as follows.

Sky and cloud are thought as background elements. These are the main factors that determine the speed of the game while the user is playing.

Secondly, it is considered as game information items, game status, life, playing conditions and main menu.

Finally, the game is considered as the main elements of the ball, ring and star.

3.4. Preparatory Stage

Calculations related to the duration of the game will be made at this stage. The priorities of the improvements to be made during this planning are determined. Then the necessary times for these improvements were determined. These levels were determined as high, medium and low 3 levels. In addition, the works that affect each other have been done consecutively.

3.5. Sprint requirement list

The Sprint Requirement List is the physical presentation of the requirements that the current team is committed to doing, such as a task schedule [8].

It serves to the team and other people to show the work they are planning, the sprint and the situation [8].

This is the part of the development time of the items that we have determined in the product requirement list. In this section, you can create a running requirement list to determine the scenario of our game and determine the duration of the game.

The game is expected to be completed after a period of 55 hours in total when the running requirement list is created and the appropriate periods are given. After this list is determined, the items in the list will be started according to the priorities.

This period can be updated in times when some problems are encountered in the evaluation of the daily Scrum.

3.6. Development

At this stage, the game is being developed. These periods are updated every day according to the rate of completion of the items removed from the running requirement list.

The project was initially set up for the game. To do this, the screen rates are first started. Screen ratios have been adjusted so that the screen can be played both mobile and desktop.

Then the sky and cloud designs in the background of the game were made and the clouds were moved according to the speed of the game.

After these works, ball and ring designs were made. Then the ball jump motion was made. This move is the most important point of our game. Because the bouncing motion of the ball must be natural, that is, as in real life. If this is not the case, we will not be able to include the user in the game and the game's fluency will be gone. After the movement of the ball was completed, the movement of the ring was made. The rings were moved from left to right according to this game speed. Now, according to the scenario of the game, the ball passes through the ring of the correct color, that is, its own color.

Tests after this stage revealed that the game was very monotonous and it was very difficult to finish the game. Therefore, some scenario changes have been made in the game.

Firstly there are 3 can fixed in the game. In these conditions, it is understood that it is very difficult for the user to switch to 100 people. For this reason, a star is added in every 10 rings. And if the user passes through this ring, he will gain another life.

Then the game is monotonous and the game is too short to be partitioned. In these sections, the color of the ball changes first as it passes through each ring. In the following sections, the colors of the rings are variable and the rings are moving to make the game more difficult. Thus, the game has become increasingly difficult.

The rings were originally planned in two dimensions. However, it was difficult to understand that the ball passed through the ring fully. Therefore, the rings were then redesigned in 3D. And so our game has become more fun.

Lastly, when the user burns or when he finishes the game, the game menus are displayed and the game status indicators are displayed so that he can see his status in the game.

It is planned to be completed in 55 hours in total. However, due to the flexibility of Agile software, several features have been added to make the game more fun and playable at the end of the 5th and 12th days. And then the sprint was re-planned according to the severity of the added features. The time graph of the application is shown in Figure 5 below.

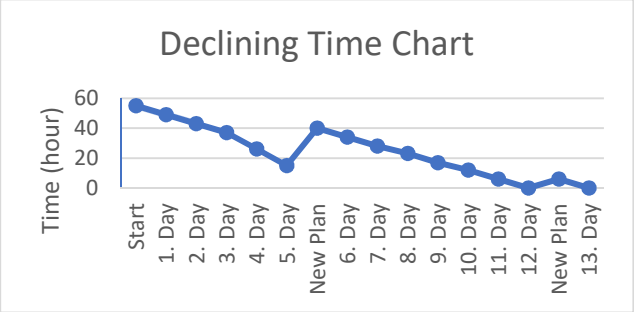

Figure 5 Time schedule of game planning

4. CONCLUSION

A screenshot of the resulting game is shown after the development is completed.

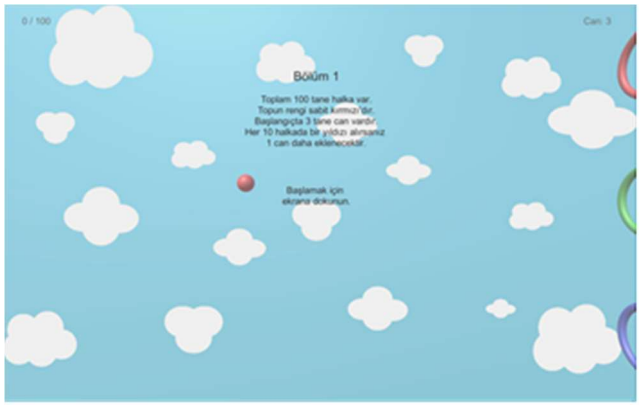

Figure 6 Game screenshot

The game starts when the user clicks on the screen shown in Figure 6. In the upper left corner there is the total number of rings and the number of rings passed. The upper right corner shows the number of lives remaining.

Each time the screen is clicked, the ball bounces. The rings move from right to left. The ball must pass through the ring of its own color. If it passes through the wrong ring, the number of lives remaining decreases by 1. After passing 100 rings, it moves to the next section.

The project is planned to be completed after a period of 55 hours in total. However, as the product was developed later, as the product was examined, some changes were made to make the game better, fun and playable and ended at the end of 86 hours. In our project, due to the flexibility principle of the agile method, a deviation of about 31 hours occurred at the end time. This deviation value is quite a while based on the total time.

Let us examine the causes of this deviation:

When we generalize the game made here, we can say that it consists of 4 main parts. These sections consist of:

- Design of game visuals.
- The menus in the game and the game information.
- Visuals and scenarios created by the more trivial side characters of the game on the game screen.
- Images and scenarios created by the main characters of the game on the game screen.

4.1. *Design of Game Visuals*

In the game planning phase, a total of 14 hours was calculated for this section. As the game progresses, new features are added in the game, but these added features are generally based on the development of existing scenarios. Star design was needed since the game was added to the 10 rings. In addition, the 3-dimensional redesign of the rings was needed. They have caused a total of 6 hours. In other words, 14 hours of the 20-hour period were determined during the game planning phase. That is, a time deviation of 43% was experienced.

4.2. Game Menus and Information

A total of 16 hours has been calculated for this section when planning the game. As new features are added to the game, it is necessary to display these features on the screen. And so, we had to prepare new menus and texts. In this game, the user had to start all over again. This was becoming very frustrating. Reducing the entertainment aspect of the game. Therefore, the game has been added to the feature. A total of 2 hours has been detected in excess of the time since the visualization of this image and the addition of the can, the need for updates such as cancellation when the user is lit. In total, 16 hours of the 18-hour period were determined during the game planning phase. That is, a time deviation of 13% was experienced.

4.3. Making Game Side Characters

A time period of 10 hours was calculated for the game planning. As the game develops, a star image has been added to the ring of the correct color in every 10 rings to improve the game. When the player passes through this ring, the star disappears, and another player is added to the player. In addition, the time needed for this section in the 3-dimensional construction of rings. A total of 3 hours was spent to make these properties. In total, 10 hours of the 13-hour period

were determined during the game planning phase. That is, a time deviation of 30% was experienced.

4.4. Making Game Main Characters

In this game, a period of 11 hours was calculated for this section. As the game evolves, there are some problems. These games have been very monotonous. The rings are stationary. The color of the ball is fixed and it is thought to be a game that consists of only one part, making the game quite tedious. For this reason, the game has been converted into 4 sections in total. In these sections, the changes of the color of the ball randomly changed, such as the movement of the rings to make the game more monotonous and more fun are planned. A total of 20 hours of additional time was needed to make these features. As a result, a total of 11 hours of the 31-hour period were planned at the start of the game. That is, a time deviation of 182% was experienced.

4.5. Results and Suggestions

The graph of time deviation according to the above-mentioned starting and subsequently changing conditions is shown in Figure 7 below.

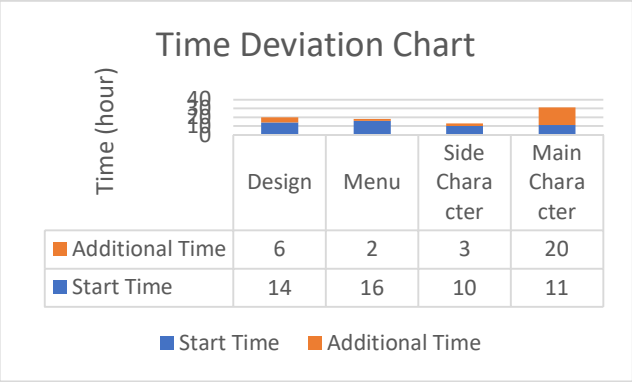

Figure 7 Time deviation graph

As an alternative, a 4-hour period is planned for project initial settings. The reason we don't take this time is due to the speed of the code developers in the background. Therefore, since there will not be any deviation in this period, this part is only added to the end of the project planning in the time calculation.

When the above 4 main sections are examined, this ratio is quite high in the sections that directly affect the game's scenario, fiction and playing situations while the game design, game menus design and game side characters section change in a very limited ratio. As a result, an end date for the project can be calculated if time is added to the planned times at the beginning of the game. However, due to the flexibility principle of agile methods, a definite date cannot be predicted. Just predictable. As these examples are reproduced, a more accurate result will be increased.

The most important part that will negatively affect the removal of this period is the correct identification of the 4 main sections mentioned above. In particular, the game's main characters and the less significant side and back characters must be separated from each other by the total time determination of the section. People who make this distinction must understand the scenario of the game very well.

In a matter that adversely affects this proposition:

Sometimes the side characters of the game are so popular that the effect on the game can be increased and become the main characters of the game. This may adversely affect the time estimate. For this reason, it is important to pay attention to this aspect when deciding the game side characters.

5. FUTURE WORK

In order for a better end date determination to be carried out in further studies, the 4 main sections we use in this article can be further elaborated. The more detailed these sections, the more accurate deviation can be determined.

While we were making the game, we went directly to the development stage without prototyping. Therefore, as the game develops, there have been more locations changed. In particular, changes in the design may have been more because prototyping was not performed. Therefore, if such an application is prototyped without directly developing, the results will be examined.

In addition, the game can be compared with other agile methods or traditional methods and the results can be compared.

Funding

Financial support wasn't received for the production of this paper.

The Declaration of Conflict of Interest/ Common Interest

No conflict of interest or common interest has been declared by the authors.

Authors' Contribution

Ş.M: literature, project construction, writing
Y.B: project construction, writing

The Declaration of Ethics Committee Approval

The authors declare that this document does not require an ethics committee approval or any special permission.

The Declaration of Research and Publication Ethics

The authors of the paper declare that they comply with the scientific, ethical and quotation rules of SAUJS in all processes of the paper and that they do not make any falsification on the data collected. In addition, they declares that Sakarya University Journal of Science and its editorial board have no responsibility for any ethical violations that may be encountered, and that this study has not been evaluated in any academic publication environment other than Sakarya University Journal of Science.

REFERENCES

[1] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, 2017. Agile software development methods: Review and analysis. arXiv preprint arXiv:1709.08439.

[2] V. Baytam, O. Kalıpsız, Scrum Software Development Model Management Tool ScrumMApp. Fifth National Software Engineering Symposium, September (in Turkey), 2011.

[3] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, and J. Kern, Manifesto for agile software development, 2001.

[4] B. Boehm, Get ready for agile methods, with care. Computer, 1, pp. 64-69, 2002.

[5] İ. Duru, Design and implementation of mobile applications in agile methods (in Turkey), 2014.

[6] J. Grenning, Launching extreme programming at a process-intensive company. IEEE Software, vol. 18, no. 6, pp. 27-33, 2001.

[7] J. Highsmith and C. Alistair, What is agile software development?. crosstalk, vol. 15, no. 10, pp. 4-10, 2002.

[8] P. Hundermark, Do better scrum. ScrumSense: http://www. scrumsense. com/wp-content/uploads/2009/12/DoBetterScrum-v2. Pdf, 2009.

[9] R. E. Landaeta, V. S. Viscardi, and A. Tolk, 2011. Strategic management of scrum projects: An organizational learning perspective. In First International Technology Management Conference, pp. 651-656, IEEE.

[10] M. Lippert, P. Becker-Pecbau, H. Breitling, J. Koch, A. Kornstadt, S. Roock, H. Zullighoven, Developing complex projects using XP with extensions. Computer, vol. 36, no. 6, pp. 67-73, 2003.

[11] D. J. Reifer, F. Maurer, and H. Erdogmus, Scaling agile methods. IEEE software, vol. 20, no. 4, pp. 12-14, 2003.

[12] E. Şahin, İ. Keskin Kaynak, and H. Koç, Implementation of SCRUM Agile Software Development Method in Software Development Works in an Organization with CMMI-DEV

Level-3 Certification. In UYMS (in Turkey), 2013.

[13] K. Schwaber, Scrum development process. In Business object design and implementation, pp. 117-134, Springer, London, 1997.

[14] K. Schwaber and J. Sutherland, Scrum guide: developed and sustained. Scrum. org., 2009.

[15] J. Sutherland, Agile development: Lessons learned from the first scrum. Cutter Agile Project Management Advisory Service: Executive Update, vol. 5, no. 20, pp. 1-4, 2004.

[16] J. V. Sutherland and K. Schwaber, The SCRUM methodology. In Business object design and implementation: OOPSLA workshop, 1995.

[17] J. Sutherland and K. Schwaber, The Scrum Papers: Nuts, Bolts and Origins of an Agile Process (2007), 2010.

[18] S. Süloğlu, When the method is agile (in Turkey), 2005.

[19] H. Takeuchi and I. Nonaka, The new new product development game. Harvard business review, vol. 64, no. 1, pp.137-146, 1986.