

**THE REPUBLIC OF TURKEY  
BAHÇEŞEHİR UNIVERSITY**

**AGILE (SCRUM) IMPLEMENTATIONS FOR  
BANKING APPLICATIONS IN TURKEY, IS SCRUM  
USEFUL OR NOT?**

**M.S. Thesis**

**MURAT ŞAHİN**

**ISTANBUL, 2016**



**THE REPUBLIC OF TURKEY  
BAHÇEŞEHİR UNIVERSITY**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
COMPUTER ENGINEERING**

**AGILE (SCRUM) IMPLEMENTATIONS FOR BANKING  
APPLICATIONS IN TURKEY, IS SCRUM USEFUL OR NOT?**

**M.S. Thesis**

**MURAT ŞAHİN**

**Supervisor: Asst. Prof. Dr. SELÇUK BAKTIR**

**İSTANBUL, 2016**

**THE REPUBLIC OF TURKEY  
BAHÇEŞEHİR UNIVERSITY**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
COMPUTER ENGINEERING**

Name of the thesis: Agile (Scrum) Implementations for Banking Applications in Turkey, is Scrum useful or not?

Name/Last Name of the Student: Murat Şahin

Date of the Defense of Thesis: 07/01/2016

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Dr. Ph.D. Nafiz ARICA  
Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Tarkan AYDIN  
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Sciences.

Examining Committee Members

Signature

Thesis Supervisor  
Asst. Prof. Dr. Selçuk BAKTIR

-----

Member  
Asst. Prof. Dr. Tarkan AYDIN

-----

Member  
Prof. Dr. Meltem ÖZTURAN

-----

## ACKNOWLEDGEMENT

I would like to express my deepest appreciation to my thesis supervisor, Asst. Prof. Dr. Selçuk BAKTIR, who continually supported my thesis from start to the end, gave motivation when I was not motivated, gave opportunity to work in the Agile development area -which has a growing popularity in Turkey- was patient, calm and constructive when we did not have good results in the implementations of Scrum and was helpful in many other things when we were working on this thesis. Without his guidance and persistent help, this thesis would not have been possible.

I would also like to thank my colleagues for supporting me with their ideas about Scrum in the pools. Without their help, we couldn't have decided whether Scrum implementations are useful or not.

I would also like to thank my committee members, who are patient and helpful and add valuable contributions to my work.

I would also thank Pelin Turgut Yılmaz for her useful contributions in editing my thesis.

More importantly, I would like to thank my family. I thank my parents, especially my mother, my father and my brothers for being supportive and faithful to finish my thesis. Without their continuous support, none of this would have been possible. Thanks again.

Istanbul, 2016

Murat ŞAHİN

## ABSTRACT

### AGILE (SCRUM) IMPLEMENTATIONS FOR BANKING APPLICATIONS IN TURKEY, IS SCRUM USEFUL OR NOT?

Şahin, Murat

Computer Engineering

Thesis Supervisor: Asst. Prof. Dr. Selçuk BAKTIR

January 2016, 92 pages

The purpose of this work is to study whether The Agile (Scrum) implementations of banking applications in Turkey is useful or not. We have been using the traditional software development methodologies (SDMs) for many years but they have been proven to be too cumbersome to meet the rapidly changing requirements and to have short product life cycles. Finally, a new approach, which is being very popular in recent years, has arrived this area to meet these expectations, it is called Scrum. Turkey is also affected by this development methodology and many companies want to implement this approach in order to achieve more productivity, more efficient and effective software products. Many software professionals believe that it is very useful for software development and it has many advantages, such that it provides faster software development and improves the ability to manage the changing requirements of customers. On the other hand, others are suspicious about the success of Scrum in the development processes of applications; even they call it an evil. To study on usefulness of Scrum, we have many surveys, pools, and interviews with the people who are/were in the Scrum development teams. The results show that Scrum is useful in many ways but it has some problems that should not be negligible and should be fixed to have better software products.

**Keywords:** Scrum In Turkey, Agile, Is Scrum Useful, Software Development Methodologies, Scrum In Banking Industry

## ÖZET

### TÜRKİYE'DEKİ BANKACILIK UYGULAMALARINDA ÇEVİK (SCRUM) YAZILIM GELİŞTİRME METODOLOJİSİ KULLANIŞLI MI?

Şahin, Murat

Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Selçuk BAKTIR

Ocak 2016, 92 sayfa

Bu çalışmanın amacı Çevik yazılım geliştirme metodolojisinin Türkiye'deki bankacılık uygulamalarında başarılı olup olmadığı anlamaya çalışmaktır. Uzun yıllardır geleneksel yazılım geliştirme metodolojisi ile yazılım geliştiriyoruz ve geliştirmeye devam ediyoruz. Geleneksel yöntemlerin değişen hızlı ihtiyaçlara cevap vermekte çok geç kaldığı kanıtlandı. Sonunda bu ihtiyaçlara cevap vermek için yazılım geliştirmede yeni bir yaklaşım çıktı ve bu yaklaşım oldukça popüler bir hale geldi. Bu yaklaşıma Scrum diyoruz. Türkiye de bu yazılım geliştirme metodolojisinden etkilendi ve birçok şirket daha üretken, daha etkin ve daha verimli yazılım ürünlerine sahip olmak için bu metodolojiyi uygulamak istiyor. Birçok yazılım profesyoneli yazılım geliştirmede Scrum'ın yararlı olduğuna ve diğer dillere göre birçok avantajı olduğuna inanıyor. Bu avantajlardan bahsedecek olursak, Scrum daha hızlı bir yazılım geliştirme ortamı sağlıyor ve müşterilerin değişen ihtiyaçlarını yönetmede yeteneğini artırıyor. Diğer taraftan, bazı profesyoneller ise Scrum'ın yazılım geliştirmedeki başarısı konusunda oldukça şüpheli durumdadır, hatta Scrum'ın kötü ve zararlı olduğunu düşünüyorlar. Scrum'ın yararlı olup olmadığını anlamak için, birçok araştırma ve anket çalışması ve Scrum takımlarında bulunan ve bulunmaya devam eden kişilerle röportajlar yaptık. Sonuçlar gösteriyor ki Scrum yararlı bir yazılım geliştirme metodolojisi fakat Scrum'ın ihmal edilemeyecek ve çözülmesi gereken problemleri bulunmaktadır.

**Anahtar Kelimeler:** Türkiye'de Scrum, Agile, Scrum Yararlı Mı, Yazılım Geliştirme Metodolojileri, Bankacılık Sektöründe Scrum

## TABLE OF CONTENTS

<b>LIST OF TABLES .....</b>	<b>VIII</b>
<b>LIST OF FIGURES .....</b>	<b>IX</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>XI</b>
<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>2. LITERATURE REVIEW .....</b>	<b>4</b>
<b>3. SOFTWARE DEVELOPMENT PROCESS.....</b>	<b>12</b>
<b>3.1 CORE ACTIVITIES .....</b>	<b>12</b>
<b>3.2 METHODOLOGIES .....</b>	<b>17</b>
<b>3.2.1 Waterfall.....</b>	<b>18</b>
<b>3.2.2 Agile Development.....</b>	<b>19</b>
<b>3.2.3 Waterfall Vs. Agile Development .....</b>	<b>19</b>
<b>4. SCRUM.....</b>	<b>20</b>
<b>4.1 HISTORY.....</b>	<b>20</b>
<b>4.2 AGILE MANIFESTO .....</b>	<b>21</b>
<b>4.3 SCRUM THEORY .....</b>	<b>22</b>
<b>4.4 ROLES .....</b>	<b>23</b>
<b>4.5 EVENTS .....</b>	<b>26</b>
<b>4.5.1 Sprint .....</b>	<b>27</b>
<b>4.5.2 Sprint Planning.....</b>	<b>28</b>
<b>4.5.3 Daily Scrum.....</b>	<b>29</b>
<b>4.5.4 Sprint Review.....</b>	<b>29</b>
<b>4.5.5 Sprint Retrospective.....</b>	<b>30</b>
<b>4.6 EXTENSIONS .....</b>	<b>32</b>
<b>4.6.1 Backlog Refinement (Backlog Grooming).....</b>	<b>32</b>
<b>4.6.2 Scrum of Scrums (SoS).....</b>	<b>32</b>
<b>4.7 ARTIFACTS .....</b>	<b>33</b>
<b>4.7.1 Product Backlog.....</b>	<b>34</b>
<b>4.7.2 Sprint Backlog.....</b>	<b>35</b>
<b>4.7.3 Definition Of "Done" .....</b>	<b>36</b>
<b>4.7.4 Increment .....</b>	<b>38</b>
<b>5. SURVEYS &amp; INTERVIEWS.....</b>	<b>39</b>
<b>5.1. SURVEY.....</b>	<b>39</b>
<b>5.2 ANALYZING DATA .....</b>	<b>44</b>



<b>5.3 INTERVIEWS</b> .....	<b>80</b>
<b>5.3.1 Interview With PO From IT:</b> .....	<b>81</b>
<b>5.3.2 Interview With PO From Business Unit</b> .....	<b>82</b>
<b>5.3.3 Interview With A Development Team Member</b> .....	<b>84</b>
<b>6. CONCLUSION &amp; SUGGESTIONS</b> .....	<b>86</b>
<b>7. REFERENCES</b> .....	<b>90</b>



## LIST OF TABLES

Table 3.1 : Waterfall versus Agile Approach .....	19
Table 5.1 : The number of participants who respond our survey. ....	39
Table 5.2 : The number of participants who took the Scrum training and who did not. ....	40
Table 5.3 : All responses to the questions.....	42



## LIST OF FIGURES

Figure 1.1 : Waterfall Methodology .....	2
Figure 1.2 : Waterfall vs. Agile Success/Failed/Challenged Rate .....	3
Figure 3.1 : Software Development Cycle.....	12
Figure 4.1 : Scrum Roles .....	23
Figure 4.2 : Product Owner responsibilities .....	24
Figure 4.3 : Scrum Development Process.....	27
Figure 4.4 : Starfish technique .....	31
Figure 4.5 : Scrum of Scrum of Scrums example.....	33
Figure 4.6 : Product Backlog example from a bank in Turkey.....	34
Figure 4.7 : Sprint Burn down Chart from a bank in Turkey .....	36
Figure 4.8 : Scrum Board Example in A Bank in Turkey .....	38
Figure 5.1 : The graph shows the participants' Scrum roles in percentages .....	44
Figure 5.2 : The graph shows the participants who take Scrum Training in percentage .....	45
Figure 5.3 : This graph shows the all answer distribution of Q3.....	46
Figure 5.4 : Responses to Q6 .....	46
Figure 5.5 : Responses to Q4 .....	48
Figure 5.6 : Responses to Q9 .....	48
Figure 5.7 : Responses to Q5 .....	49
Figure 5.8 : Responses to Q30 .....	50
Figure 5.9 : Responses to Q7 .....	51
Figure 5.10 : Responses to Q13 .....	51
Figure 5.11 : Responses to Q8 .....	52
Figure 5.12 : Responses to Q34.....	53
Figure 5.13 : Responses to Q10.....	54
Figure 5.14 : Responses to Q14.....	55
Figure 5.15 : Responses to Q11 .....	56
Figure 5.16 : Responses to Q16.....	57
Figure 5.17 : Responses to Q12.....	58
Figure 5.18 : Responses to Q21 .....	59
Figure 5.19 : Responses to Q15 .....	60
Figure 5.20 : Responses to Q26.....	61
Figure 5.21 : Responses to Q17 .....	62
Figure 5.22 : Responses to Q32.....	63
Figure 5.23 : Responses to Q18.....	64
Figure 5.24 : Responses to Q23 .....	65
Figure 5.25 : Responses to Q19 .....	66
Figure 5.26 : Responses to Q22.....	67

Figure 5.27 : Responses to Q20 .....	68
Figure 5.28 : Responses to Q25 .....	69
Figure 5.29 : Responses to Q24 .....	70
Figure 5.30 : Responses to Q29 .....	71
Figure 5.31 : Responses to Q27 .....	72
Figure 5.32 : Responses to Q31 .....	73
Figure 5.33 : Responses to Q28 .....	74
Figure 5.34 : Responses to Q33 .....	75



## LIST OF ABBREVIATIONS

COBIT: Control Objectives for Information and Related Technology

DoD : Definition of Done

DSDM: Dynamic systems development method

IQ : Interview Question

IT : Information Technology

ITIL : Information Technology Infrastructure Library

PBI : Product Backlog Item

PBIs : Product Backlog Items

PBL : Product Backlog

PMP : Project Management Professional

PO : Product Owner

PSM1 : The Professional Scrum Master 1

Q : Question

SDLC : Software Development Life Cycle

SDMs : Software Development Methodologies

SM : Scrum Master

SoS : Scrum of Scrums

## 1. INTRODUCTION

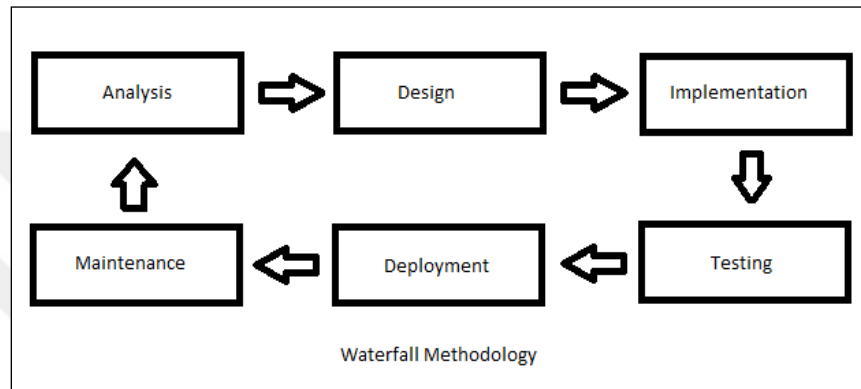
We are living in a world in which technology has high importance in many areas such as mobile phones, televisions, buses, web sites, robots, social media etc. These technological devices can be mainly categorized by two significant elements that can be named as "Hardware" and "Software". They are so connected to each other that if one does not exist, the other will not be so efficient. They require each other; neither can be fully used without the other. Using a musical analogy, hardware is like a musical instrument and software is like the notes played on that instrument (Langton, 2015).

We mentioned that many hardware devices such as mobile phones, watches, refrigerators, many household appliances and many other daily devices which we use, are functional, thanks to software. We can see that software makes our life easier. If we think deeply about how software is developed, we can easily see that software can only be constructed by group of professionals to completely operate some purpose. Of course, software development has many stages to be completed. There are some methodologies that divide software development into stages and phases that contain planned activities to be planned and managed better (Software development process, Wikipedia). Common methodologies include Waterfall, prototyping, iterative, incremental, spiral development, rapid application development, extreme programming, and Agile methodology.

Waterfall has been the commonly used methodology for many years. It divides software development process into specific stages. These are Requirement Analysis, Design, Implementation, Testing, Deployment and Maintenance (Figure 1.1). They can be named as Software Development Lifecycle (SDLC). These stages are sequential and prerequisites of the previous stages. If one stage is not completed, the following stage cannot be started. Many software projects have been developed with this software development methodology. It is used by most of the software development companies as it is easy to implement and adopt for any company. However, there is a debate between software experts whether Waterfall is successful or not. They say that it is seen as a non-iterative approach, it is a sequence of events in which each stages generally finishes before the next one begins. If

any stage fails, the others will also fail. If a change in the first stages is needed, going back to the later stages can be very costly, difficult and time-consuming. For these reasons, Waterfall development methodology is seen non-effective and non-efficient in software development.

**Figure 1.1 : Waterfall Methodology**

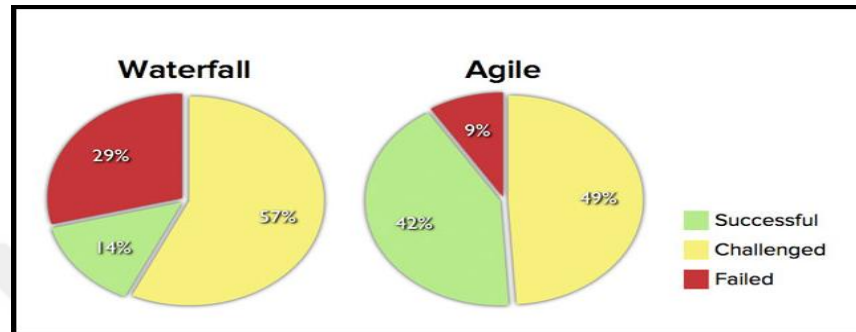


Studies have shown that in about 29 percent of software projects that failed (29 percent failed 57 percent challenged, meaning that under deliver promised value, or outright fail, and with good reason), the usage of the Waterfall methodology was one of the key factors of failure (Gartner 2012). The reason is the project timeline is planned at the start of the project and the product is completed at the end of the project. Hence, if any stage is delayed, the project is delayed. As a result, many new methodologies have emerged that can be counted as modified Waterfall approaches.

Among these methodologies, the most recent one, Agile methodology has been gaining increasingly more reputation and has already been adopted by many organizations all around the world. It is a new software development process, based on iterative development where requirements and solutions evolve via collaboration among self-organized cross-functional teams. If any change is required, it can adapt easily and it will be less costly and less time consuming. According to Gartner in 2012, to make projects successful, providing feedback to customer is essential. In order to do that, we can adopt Agile development methodologies. It is believed that is can solve the problem of failures of

software projects. Agile succeeds 3 times more often than Waterfall with a failure rate of 9 percent and with a success rate of 42 percent. (Figure 1.2)

**Figure 1.2 : Waterfall vs. Agile Success/Failed/Challenged Rate**



*Source : The CHAOS Manifesto, The Standish Group, 2012*

The Agile methodology has some variations like Dynamic systems development method (DSDM), Kanban and Scrum. The most widely used variation of the Agile methodology is Scrum. Scrum is an empirical approach which accepts the idea that the problem cannot be fully understood or defined, and focuses on maximizing the abilities to adapt to changing requirements and delivering product faster. It is a flexible and holistic approach where the development team acts as a unit to reach a common goal. It is a transparent approach; the progress can be observable by the customers and the other actors. It is an iterative development which enables the team to be self-organized by encouraging the team members to communicate collaboratively and closely.

Because of these advantages, it is gradually getting more and more attention by many organizations especially by the banking industry in Turkey. In the competitive environment in Turkey, banks want to have faster product development to be the industry leader. Therefore, they want to adopt the Scrum software development methodology to achieve these goals. However, there are many contradictions among the software professionals regarding whether Scrum is useful or not. Some say that Scrum is evil and is hard to implement, while the others believe it is the best solution in software development industry. In this research, we try to understand the usability of the Scrum variation of Agile development methodology.



## 2. LITERATURE REVIEW

Some software professionals believe that Scrum is very useful and with the help of Scrum, we can meet the rapidly changing business requirements. On the other hand, other professionals believe that Scrum is not the answer to solve the problems of software development. There are many articles and case studies which support Scrum is the best methodology and others that do not support the idea.

Firstly, we want to start with the supporters of Scrum. The first and the most important example can be the Yahoo's rollout. Yahoo is a large enterprise with a \$32 billion market cap and has one of the largest Agile implementations in the world. According to the results of a questionnaire on more than 150 Yahoo! Scrum Teams by Benefield, 74 percent of respondents said Scrum improved thirty-day productivity. 80 percent of respondents said Scrum helped them clarify the team goals. 89 percent of respondents said Scrum helped their collaboration and cooperation within their teams. 64 percent of respondents said Scrum improved the business value and 54 percent of respondents said Scrum improved overall quality. (Benefield 2007)

Second biggest case study is the Intel Corporation's case study. They conducted some case experiments by using Scrum in teams with approximately 50 people at Intel. They reached some major results in the aspects of cycle time, performance to Schedule Morale and Transparency. The first that they reached is that Scrum provides 66 percent reduction in cycle time (Danube 2008) which means that Scrum makes a project finish in a shorter period of time. The second one is performance schedule which shows that they have established and maintained capacity based planning and a two-week cadence for more than a year and eliminated schedule slips and missed commitments. The third aspect is that they have improved communication and job satisfaction, so that even the lowest morale team has become the best performing team, indicating an example for the notion that job satisfaction and higher morale lead higher and sustainable pace in organization. Regarding all these, Intel has changed its organization from a command-and-control and plan-based organization into an inspecting, adapting and self-organizing company with an empirical

planning base. All in all, in spite of some problems adopting it in the beginning, Scrum has been working very well at Intel.

The third case is from Turkey. It is a Turkish bank named Kuveyt Turk. According to the research (Kır,2014) Kır has conducted with a questionnaire, some interviews and pilot Scrum teams; Scrum framework and the other traditional project management approaches have almost the same problems but Scrum can solve some of the problems. For example, according to the interview, Kuveyt Turk had some planning problems when Waterfall was implemented and sometimes some of the projects were suspended because of uncertainty. By adopting Scrum, the product evaluation could be seen step by step and thus if the Product Owner (PO) believed that the increments were worth to be released on the market, the shippable products which were done at the end of the sprints could take place on the market. On the other hand, in the pilot application in Kuveyt Turk, there was a problem of finding required staff to make operational facilities. To solve this problem, Kuveyt Turk needed to have the Project Management Office identify common goals, to maintain required staff to do, to build relationship among the Scrum teams. Moreover, an analysis of an item on her questionnaire show that 85 percent of responses agreed to use Scrum methodology and expressed that overall Scrum was better. On the other hand, we know that management support is very significant in Scrum and Scrum requires complete support from all the stakeholders. However, in Kuveyt Turk, management support and all stakeholder support were not there to be seen. Also, Human Resources department in Kuveyt Türk had no roles related to the Scrum. In summary, Kuveyt Turk may use Scrum as a software development methodology but it needs some organizational changes such as defining Scrum Roles in Human Resources Department for IT. On the other hand, Scrum can solve the problem of the projects which have high uncertainties.

The fourth case is as well from Turkey, from the article in "Optimist" journal (Bolat, 2014). He tells about Scrum and its status in Turkey. Software professionals who know Scrum are very new but we have Agile Turkey, Scrum Turkey and Agile Talks to get updated about Scrum and gradually we have had some Scrum jobs in Turkey. Scrum Turkey took Jeff Sutherland, one of the founders of Scrum to Turkey and had interview with him. According

to Sutherland, even problematic teams have improved about 35 percent in efficiency in projects. He has some clues and suggestions for Scrum in Turkey. "Turkey needs to compete in global economy. I am sure that how much effective Scrum is used as much as Turkey will get stronger in this competition. I have witnessed that Turkey has extraordinary enthusiasm towards Scrum. Adopting Scrum in chaotic working environments is easier than traditional project methods. For this reason, there are many new initiatives who make use of Scrum are successful. You should think your country as a new initiative."

The next case is from an article, also from Turkey, published on Scrum.Org in "Whitepapers" section (Akdağ, Kombak, and Yılmaz, 2014). They tell about the story of the IT Transformation of "Türkiye Finans", a bank with about 300 branches around Turkey and abroad, serving more than 3 million people with approximately 4500 employees. Türkiye Finans focuses on increasing its market share to maximum and has decided to increase its software delivery and services into the market. To do that, it has adopted Scrum as a software development methodology. Before Scrum, there were Waterfall methodologies used for generating products but there were limitations in collaboration, transparency, and communication; efficiency issues and problems in sudden changes in the requirements. Within a period of 3 months, after training more than 200 employees, initiating functional 45 Scrum teams as well as Scrum roles and PO, and getting PSM 1 certificates for all of them; Scrum teams have started to work together collaboratively as teams in which members can help each other learn and improve and take individual responsibilities willingly. Also, the projects have started to be completed before due dates, the communication among all the stakeholders and IT has improved, employee satisfaction and motivation have increased greatly. On the other hand, there have been many problems and obstacles. One of them is the reports requested by the institutions such as BDDK, COBIT, TBB, BKM with close due dates. It is hard to prepare these reports without impairing agility. The second one is the internal audit mechanism requiring documents which are heavy burden for Scrum Teams. The last one is the abundance of problem records that need to be solved urgently. These lead to having difficulties in creating value for most of the Scrum Teams. For these reasons, Agile Studio has been founded in Türkiye

Finans to coach and help the adoption of Scrum in this banking organization. It ensures transparency and that teams are cross-functional and self-organizing.

The next article is from one of the founders of Scrum, Jeff Sutherland. After introducing Scrum into five different companies of different sizes with different technologies -Easel Corporation, VMARK, IDX Systems, Individual Inc., and Patient Keeper Inc.- he worked in as a Vice President (VP) of engineering and a Chief Technology Officer (CTO), he confidently says that Scrum works in any business environment (Sutherland, 2001). In all cases, he says that Scrum radically improves communication and delivery of the programs. Initial Scrum was adopted by Easel Corporation in 1993 and in 1995 Easel Company was acquired by WMARK Corporation and it still continues to work with Scrum to this day and also in the same year, IDX Systems, which is the largest US healthcare software company and 600 developers worked in a dozens of products started to use Scrum. In 2000, PatientKeeper, a mobile/wireless healthcare platform company, chose Scrum as a software development methodology. According to Peter Degrace and Leslie Hulet Stahl in 1990, requirements are not fully understood before project begins. The customers realize what they want before seeing the initial version of the software. After conducting Scrum in Easel, they solved the problem of changing requirements and also gained some features such as team-orientation, improved performance of individuals by meeting everyday to see the progress of the projects, by helping each other via a pair programming and by explaining the most suitable method for software implementation. When VMARK had the Easel Corporation, the original Scrum Team also had worked on the same products. Within a few months, VMARK succeeded in developing new two internet products and repositioned internet applications. In 1996, Individual Inc. introduced the reports and statuses of the company on the web in order to know the status in real time, all the time by the success of the Scrum meetings and the team which were self-organized around a plan to meet the priorities of the company. During the summer of 1996, dozens of Scrum Teams were in operation in IDX which had many products and many teams and hundreds of developers who were working to produce healthcare products in United States. As a result, the IDX had four to five times the industry average in deliverable functionality. In early

2000, PatientKeeper, a software company whose aim is to capture and process financial and clinical data with mobile devices, proved that Scrum worked equally well across technology implementations.

According to another research (Norman and Rising, 2000), implementation of Scrum in small teams is useful and appropriate for AG Communication Systems, which has software development teams ranging in size from two to several hundred professionals. After conducting an experiment on three teams, they have found: with Scrum,

- i. The product becomes more manageable by shipping small products
- ii. Even requirements are not stable, the projects can progress
- iii. Everything is visible to everyone
- iv. Team communication improves
- v. Customers see the increments at the end of the sprints
- vi. Customers can obtain feedbacks about the functionality of the product
- vii. A culture is created where everyone expects projects to be successful.

Now, it is time to have a closer look at the opponents of Scrum framework.

According to the book *"Agile Processes in Software Engineering and Extreme Programming"* (Moe and Dingsøyr, 2008), they have conducted some experiments in an organization with software teams by comparing Scrum and "Big Five" framework for team effectiveness. They conclude that Scrum has several mechanisms and support factors in the framework but these mechanisms were not easy to implement in practice. For example, the aspect of leadership is not appropriately addressed in Scrum and although Scrum focuses on self-organizing and coaching skills, it does not clearly express how this should be implemented. Moreover, they have found that the project suffered from lack of long-term planning, poor handling of problems and inadequate mutual trust. These problems have

also led to lack of self-organization. Thus, they have made a plan on using "Big Five" framework rather than Scrum.

According to the article (Sonmez 2010), Scrum will die. He likes Scrum and thinks that Scrum is a good thing in a way but he also thinks that it is not a good thing as well. He thinks that Agile training is to gain money and he questions the two day training required to become a Scrum Master. Furthermore, he thinks that Scrum addresses many problems of development teams but it also tends to trigger some new problems. He categorizes the problems into six categories. These are Planning, Velocity, Commitment, Tasking, Product Ownership, and Scrum Meetings. To start with planning, he thinks that it is a total waste of time because Scrum teams devote one day for two-week sprint for planning just for gathering the metrics. Velocity is a solid concept in which estimation is always wrong and so the velocity has no meaning for unstable sprint length and unstable team. When we come to commitment, he thinks that commitments are very significant and also critical but the team cannot write more codes when they accept the commitment. The tasks are always copy-and-paste task which are usually alike but when you are developing PBI from PBL, it has some priorities and different ways to implement a solution for this PBI. However, everybody in Scrum Teams copy and paste the same set of tasks such as analyzing, testing, developing, etc. He says that Product Ownership is very crucial in Scrum and it works in theory but it does not in reality because one person manages whole product's life and for this reason, the teams cannot be self-managing because of doing what the PO wants. Furthermore, he thinks that Scrum meetings are too short and that everyone repeats the same stuff there.

There is an article named "*Is Scrum Evil*" (Ardant, 2008). Although he thinks that Scrum provides medium-term benefits in the beginning, he later understands why so many people seem to dislike Scrum. He has found the following as the reasons to call Scrum an Evil:

- i. When Scrum fails, everybody believes that Agile is no good.
- ii. Process is very important rather than people.

iii. It leads to some pointless stand-ups.

iv. Finally, he thinks that it is iterative Waterfall.

On the other hand, he has found the following as reasons to tell Scrum is good:

i. Scrum provides good structure for Scrum team members and POs.

ii. Scrum generates self-organizing teams.

iii. Scrum makes people more responsible to their commitments.

iv. Scrum detects problems earlier.

v. Scrum focuses on delivery.

Moreover, some of the people believe that Scrum is hard to implement. Although everyone likes Scrum because it has some benefits on productivity, delivery and creating an enjoyable working environment, Scrum is hard (Moe and Dingsøy, 2008). It is because everybody has to understand Scrum at its deepest and theoretical level, and also software development is very complex, hence everybody should pay attention constantly and find best solutions to the software problems. As a result, Scrum is not easy; it requires your attention and intelligence constantly.

Furthermore, a scientist at CERN has a pilot project with Scrum to analyze teams with Scrum (Neubert, 2014). He has reached the conclusions that Scrum is appreciated and seen as useful but it is actually hard to master and needs practice because all habits cannot be lost easily. He has analyzed Scrum for 13 weeks in two parts. One is on how well Scrum framework is implemented. And the second one is on what impacts adopting Scrum create on projects. For the first part, they have used one-week sprints and could see the problems clearly by the help of daily meetings and sprint retrospective. However, they have had some problems such that if you are inexperienced, task breakdown is not easy, learning is difficult to plan, they never understand when things are finished, and finally pressure to get something done at the end of the sprint. Finally, he has some advices for the people who

want to use Scrum as a software development methodology: to define clear tasks is very important and to use extensive and stable definition of "done".

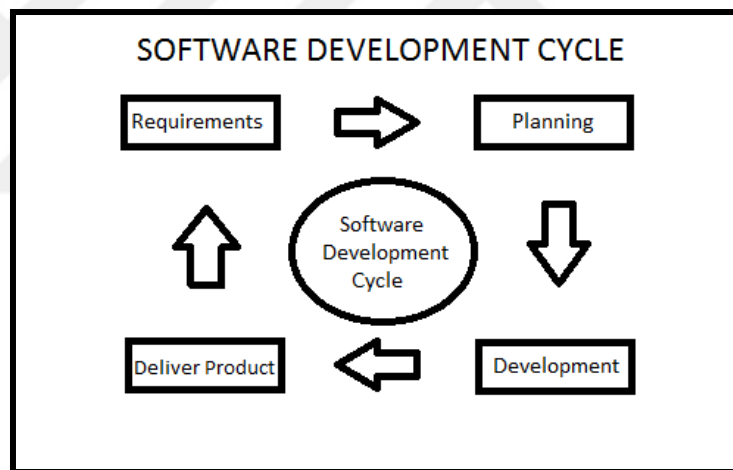




### 3. SOFTWARE DEVELOPMENT PROCESS

Software Development is a process of creating a software product, applications and frameworks. It is ideally structured and planned process which includes research, development, reuse, maintenance and testing. These activities form a process that we can call it as a Software Development Life Cycle (SDLC). It was emerged in 1960s in order to build information systems and it is considered as the oldest development methodology. SDLC is a deliberate, structured and methodological way of delivering the product in which the activities are rigidly sequential and they are so connected to each other, the previous one is finished, the next stage starts (Figure 3.1).

**Figure 3.1 : Software Development Cycle**



*Source :* (<http://www.tara-systems.de/software-development-process.html>, Accessed 24 March 2015)

#### 3.1 CORE ACTIVITIES

Software development is a process of getting a shippable product for any companies. While making a product in the end of the software projects, there are systematic and planned activities in which software is built. These activities are so essential because we can understand the requirements of the software product, decide how to effectively design the product, test whether the requirements are fulfilled, when product is ready to deployment

and making measurements and precautions in order to keep product alive. These activities can vary but we can mainly categorize the activities such as requirements, planning, development, and delivering product.

### **Requirements :**

Before we start to create a software product, we should understand clearly why we need this product and also we should realize the desired requirements and analyze the needs and agree the needs of this product for further implementation. In requirements part, we start with the initiation of the project and then analyze the requirements and risk factors that can prevent this product from being completed. After the requirements are taken, they should be documented, testable, measurable, and traceable in order to understand the needs of the stakeholders and validate them. For these reasons, requirements activity is very critical to the success of the projects because if we do not understand the needs clearly, there can be a no usable software product, or software project can be delayed, more resources can be needed to develop the software which means that the project fails because to finish the project, more time is needed and budget should be increased. Requirements stage can be a long and tiring process in which identifying all the stakeholders, taking into account all their needs. To do that, meetings are settled, use cases are drawn, interviews and review sessions can be done.

There are strengths and weaknesses of requirements activity. Strengths are that it

- i. can be used as a checklist to develop the software.
- ii. can be used as a contract between stakeholders and developers.

On the other hand, the weaknesses are :

- i. The requirements document can be more than hundreds of pages and it cannot be read by the stakeholders carefully. So, it cannot be used as a contract between the sides.

ii. Documenting the requirements may not cover the completeness of requirements. It must be controlled by someone whether this document shows all the requirements.

iii. It is almost impossible to uncover all the functional requirements before the development and testing process. It may lead additional requests and some controversies between them.

There are many types of requirements. Customer requirements, architectural requirements, structural requirements, behavioral requirements, functional requirements, non-functional requirements, performance requirements, design requirements, and derived requirements. They should be taken into account in order to meet all the requirements of the software product. For example, there may be some system restrictions in order to complete one requirement. Therefore, it should be reported to the stakeholders and agreement should be settled.

### **Planning :**

Planning in software development is the application of knowledge, skills, tools and techniques to project activities to meet project requirements. (Sommerille,2011). It may vary from project to project but, for most projects, planning has significant goals:

1. Deliver the software on time and on budget
2. Deliver software which meets customers' expectations.
3. Maintain a happy and well-functioning development team.

Objective of software project planning is to provide a baseline to make reasonable resources, cost and schedule. After gathering the requirements, there should be a plan for the requirements and prioritize them and then scheduling them to be ready for development. Thus, we have the Scope of the development, which is fully understandable by the project team and by the stakeholders, it includes the coverage and limitations of the

software product, and the amount of sufficient resources such as human resource, hardware resource and software resource and their required time to finish the project.

Regardless of the approach used, it can be agile or waterfall methodology, there is still a need for project planning in order to plan how resources will be allocated (Sommerille,2011). However, there are some difficulties in agile planning. The major difficulty is that it is reliant on customer involvement and availability. For example, priorities can be changed for the project team and for the customers. Moreover, when Agile teams are large and geographically distributed or when team member changes frequently, it is impossible for everyone to be involved in the collaborative planning that is essential for Agile project planning.

Project plan process has essential sub activities such as identifying constraints, identifying risks and defining milestones and deliverables. Also, if any risk emerges, initiation of risk mitigation activity and also re-planning the project activities emerge.

### **Development :**

Once we specify the requirements of this project and plan them, we can start the development phase. The development phase contains several parts such as designing, implementation, testing.

First, designing is the activity following requirements specification before implementation. It contains all the activities such as conceptualizing, prototyping, implementing, framing and modifying complex systems. It involves problem solving and planning a software solution in both algorithm designs, architecture design. user experience design. When designing software, one should give importance to all design aspects.

There are many principles of software design. These are :

1. The design process should not suffer from tunnel vision. They should be some alternative approaches and alternative solutions.

2. The design should not reinvent the wheel. Time is short and resources are limited. For these reasons, for the design, used design patterns in previous projects can be used effectively.

3. The design should be uniform and integrated in which it looks like one person designed the entire solution.

4. The design should be structured to adapt change. If there is a need for change, the design can accept this change easily.

5. The design should be reviewed by the experts to detect the semantic errors.

Second, Once the design stage has been completed, the software programmers can begin to write code and design the user interfaces in implementation stage. They are responsible for the design principles and the requirements meet customers' expectations. In this stage, data structures are formed, validation routine are initiated, queries are written, and inputs are printed and showed on the user interfaces.

There are some fundamentals in implementation. These are :

1. It should be less complex and should be simple and readable.

2. It can adapt change.

3. There should be an impact analysis if there exists a software system.

Finally, testing is the process of checking and controlling whether requirements are done, meets performance criteria, deciding product is usable, can be run in the intended environments. Tests are infinite even for simple software, but practically tests are used for finding software bugs and after fixing the bugs, controlling the situation of bugs and the new bugs that come from the solution of the pervious bugs.

Testing process may vary from methodology to other one. For example, in waterfall methodology, it can be performed by the group of professionals named testers after development is done. On the other hand, in test driven development, tests are written by the

start of the project and to fail software initially. Finally, this test process achieves continuous integration of development.

After we complete these activities we have a ready product to deliver the product.

### **Delivering the product :**

After finishing development, the software is ready for deployment and so we have a product in production environment. After software is delivered and installed, maintenance, support and personal training for using the product are the important activities in this phase.

There are deployment activities in which software can be delivered securely. These are Release, Install and activate, deactivate, adapt, update, uninstall, retire. After the development completed, a release is prepared by the programmers in order to install and activate. If there is a problem in the release, it can be deactivated, updated or updated by another release. If there is no need for this release, it can be retired, it is the end of the software product.

There are deployment roles in which delivering the product is guaranteed. In pre-production environments, there are roles such as application developers, build and release managers, deployment coordinators. On the other hand, in production environments, system administrators, database administrators, release coordinators, and operations project managers.

## **3.2 METHODOLOGIES**

Several software development methodologies have been used since the beginning of the software development, mainly in two categories: Traditional methodologies and Agile methodologies.

Traditional methodologies such as Waterfall that have distinct phases are sometimes known as SDLC. On the other hand, Agile approaches which define a process of iteration and

activities such as designing, implementation, deployment can run all the activities simultaneously.

Common methodologies consist of Waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and Agile methodology. In this study, we will cover the Waterfall and Agile methodologies and compare them.

### **3.2.1 Waterfall**

In 1976, Bell and Thayer named this model as Waterfall although its formal description was in 1970 by Winston W. Royce. Waterfall development methodology is a sequence of events; some of the professionals call them stages, with specific orders. Only after one stage finishes, the other one can start. According to Royce, these are the stages of waterfall: requirements, analysis, design, coding, testing and operations such as installation, support and maintenance.

The waterfall model provides a structured approach, easy to understand and adopted and implemented by any software organization. It also provides easily identifiable milestones in the software development process. For this reason, waterfall model is used as a beginning example of development methodology in many books and courses (Douglas, 2009).

Waterfall model is a traditional and non-flexible engineering approach that discourages revising any prior phase. It has been widely blamed for failures (over time and over budget) in large-scale software projects. (Dischave, 2012). It is because clients may not know exactly what their requirements are before they see working software. It leads to redevelopment, retesting (Parnas, David L.,1986). Moreover, designers may not see the difficulties on the development stage while designing the project. This can cause revising the requirements and lead to some problems. (McConnel, Steve, 2004)

### 3.2.2 Agile Development

Agile development is an alternative and very different way to traditional software development. It is an iterative, team-based, self-organizational approach for software development. This method emphasizes rapid delivery and shippable increments in time-boxed periods. Moreover, Agile development provides opportunities to revise and refine changing customer needs by taking continuous feedbacks.

Agile development has many characteristics which are eccentric to many people. These are:

- a. Requirements can change while timescale is fixed. For example, customers can see the product iteratively and want some changes to have a better product.
- b. Develop small, incremental shippable products and iterate
- c. Testing is integrated throughout project lifecycle
- d. The team members must be empowered to make their own decisions.

There are many variations of Agile processes such as Dynamic systems development method (DSDM), Kanban, and Scrum. Scrum is the most widely used variation. We will see its details in the following chapters.

### 3.2.3 Waterfall Vs. Agile Development

Some advantages and disadvantages are seen when we compare these two methodologies. Table 3.1 shows the comparison between these two approaches.

**Table 3.1 : Waterfall versus Agile Approach**

Factor	Status	Agile	Waterfall
Project size and complexity	Smaller, less complex	X	
	Larger or more complex		X
Customer Availability	Available frequently throughout of project	X	
	Customers are not available		X
Customer Tolerance for scope and cost changes	Flexible budget and schedule are possible	X	
	Budget and schedule are fixed or hard to modify		X
Time to market	Partial products can be shippable	X	
	Full project must be delivered to the market		X



## **4. SCRUM**

Scrum is an iterative and incremental Agile software development methodology and it enables the rapid response to the rapid changes in the requirements of customers in the development process. This approach is defined as "empirical approach". It accepts that the problem cannot be fully understood or defined; it focuses on maximizing the team's ability to deliver quickly and respond to emerging requirements.

Scrum is lightweight process, which means the overhead of the process is kept as small as possible, to maximize the amount of productive time available for getting useful work done, simple to understand and difficult to master (Ardant, 2008).

### **4.1 HISTORY**

Jeff Sutherland and Ken Schwaber first announced Scrum process in 1995 in Oopsla conference in Austin, in USA and published the paper "SCRUM Software Development Process". They found the term "Scrum" from the paper "The New New Product Development Game" by the authors named, Takeuchi and Nonaka. The research showed that outstanding performance in the development of new, complex products is achieved when teams, small and self-organizing units of people, are fed with objectives, not with tasks. The best teams are those that are given direction within which they have room to devise their own tactics on how to best head towards their joint objective.

In February 2001, Jeff and Ken were amongst the 17 software development leaders creating the Manifesto for Agile Software Development.

In 2002, Ken Schwaber founded Scrum Alliance with Mike Cohn and Esther Derby in order to have highly successful Certified Scrum Masters. In 2006, Jeff Sutherland created his own company, Scrum Inc. while continuing to offer and teach Certified Scrum courses. Ken left Scrum Alliance in the fall of 2009 and founded Scrum.oeg to further improve the quality and effectiveness of Scrum.

With the first publication of the Scrum Guide in 2010, and its incremental updates in 2011 and 2013, Jeff and Ken established the globally recognized body of knowledge of Scrum.

Ever since its first publication in 1995 up to now, Scrum has been adopted by much software development companies around the world and it is recognized as the most applied framework for Agile software development.

Today, Scrum is not only used for software development but also used in manufacturing, marketing, operations, and education fields.

## **4.2 AGILE MANIFESTO**

On February 11-13 2001, seventeen people, the representatives of Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes, agreed on some principles about the term "Agile". There are twelve principles of Agile Software (Agile Manifesto):

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

### 4.3 SCRUM THEORY

Scrum is an empirical process which asserts that knowledge comes from experience and making decisions based on what is known (Schwaber and Sutherland, 2013). Three significant factors support every implementation of Scrum: transparency, inspection and adaptation.

**Transparency:** Significant aspects of software development process must be visible to all the stakeholders for the outcome. Transparency requires a common language for the process in order to be understood by all participants and which stage it is at.

**Inspection:** Scrum users must frequently inspect Scrum artifacts and progress in order to detect undesirable changes and to realize whether Scrum Goal is attained.

**Adaptation:** If an inspector detects that one or more aspects of a process is deviated; the process must be adjusted to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation, as described in *Events* section: Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective.

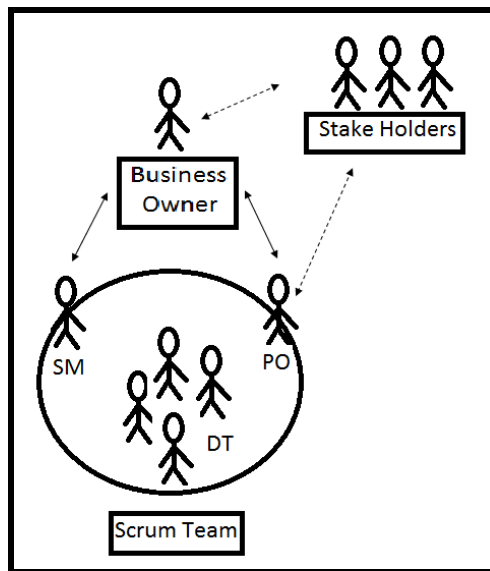
#### 4.4 ROLES

There are three core roles in Scrum Team: Product Owner (PO), Scrum Master (SM) and Development Team (DT) (Figure 4.1).

Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work. Cross-functional teams have all competencies required to succeed in their work without depending on others which are not part of the team.

Scrum teams deliver products iteratively and incrementally. These incremental deliveries must be the useful versions of working product.

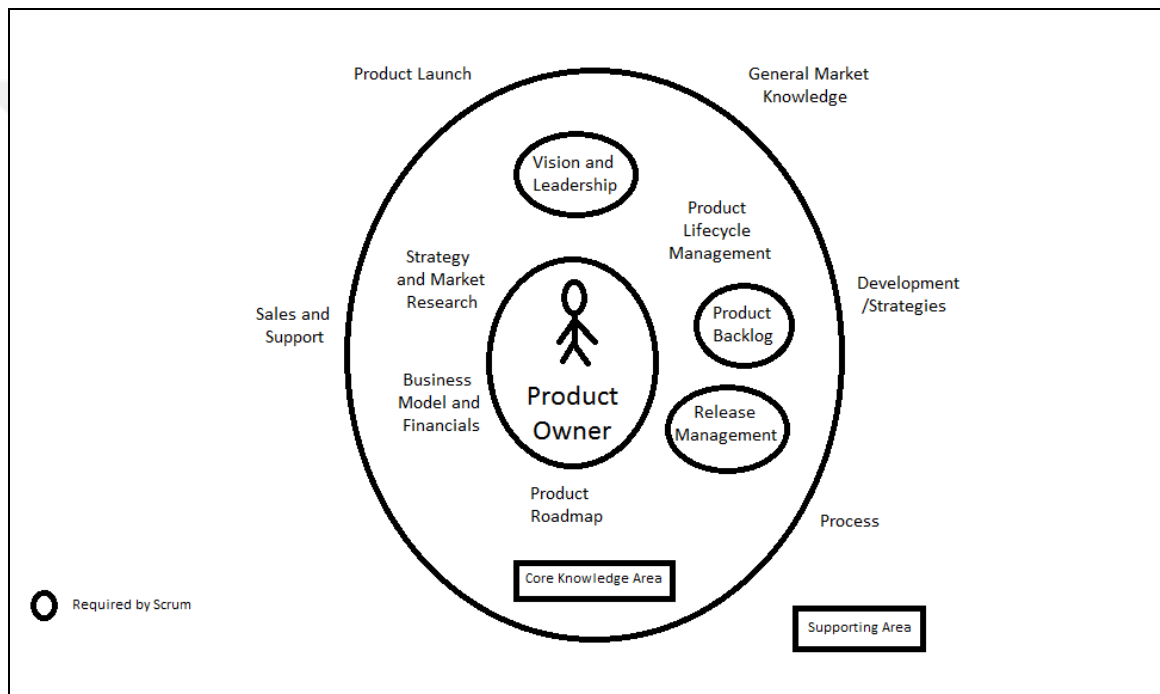
**Figure 4.1 : Scrum Roles**



## The Product Owner (PO)

The PO is responsible for maximizing the value of the product and work of the Development Team. The PO is one person, not a committee. The PO represents the stakeholders, and is the voice of customer. Figure 4.2 shows all the responsibilities of PO.

**Figure 4.2 : Product Owner responsibilities**



Source : (Pichler, 2014, <http://www.romanpichler.com/blog/one-page-product-owner/> accessed 24 March 2015)

The PO is also responsible for managing the Product Backlog (PBL) which contains prioritized features list, which is prioritized by the PO, containing short descriptions of all functionality desired in the product. The PO also has the responsibility of clearly expressing Product Backlog Items (PBIs), ordering the PBIs, ensuring The PBL is visible and transparent and understandable for the Development Team.

After the Development Team's releases of increments, he or she announces them to the stakeholders. PO may do the PBL management or have the Development Team do it. However, the PO remains as the main responsible.

In order to be successful in the organization, the entire organization must be respectful to his or her decisions. No one is allowed to tell the Development Team what to do, and Development Team is not allowed to act on their own.

### **Development Team**

They are a group of professionals who have the responsibility of delivering a releasable increment of "Done" product at the end of each sprint. If it is decided, the increment can be released. Development team is self-organizing and cross-functional. They can create product increments on their own without any help from the others from different teams.

Development team members have no titles like analyst or tester other than Developer. They are all developers. Team members can have some specialized skills and proficiency areas but accountability always belongs to the Development Team as a whole.

The team can consist of 3 to 9 members. The team must be small enough to remain Agile and large enough to complete significant work. Smaller teams can have difficulty in gaining product increment while larger teams can have the difficulty in coordination. The PO and Scrum Master(SM) are not included the size of the team unless they are doing some work from the Sprint Backlog.

### **Scrum Master**

The Scrum master in the Scrum Team has the responsibility of ensuring that Scrum is understood and being implemented well by giving high importance of Scrum theory, best practices and rules.

The SM is not a project manager or not a traditional team lead in the Scrum Team, he or she is a servant leader or team facilitator and helps increase the value of product that development team increment.

The SM has also different responsibilities to the PO of the Scrum Team, to the Development Team and to the Organization.

The SM's responsibilities to the PO are to:

- i. Find techniques for effective PBL Management,
- ii. Help the Scrum Team understand the need of the PBIs,
- iii. Understand and practice agility,
- iv. Organize Scrum Event that are required and requested,
- v. Ensure that product planning is in an empirical environment.

The SM's responsibilities to the Development Team are to:

- i. Try to increase the team's productivity, functionality and motivation,
- ii. Coach the Development Team to be self-organized and to be cross-functional,
- iii. Get rid of the problems and especially impediments, which is anything that holds the team back, and problems that Development Team has for continuity of product progressing.

The SM's responsibilities to the Organization are to:

- i. Lead the guides the organization in its Scrum Adoption,
- ii. Help workers and stakeholders understand Scrum and adopt Scrum in an empirical way,
- iii. Collaborate with other SMs to increase the effectiveness of Scrum in the organization.

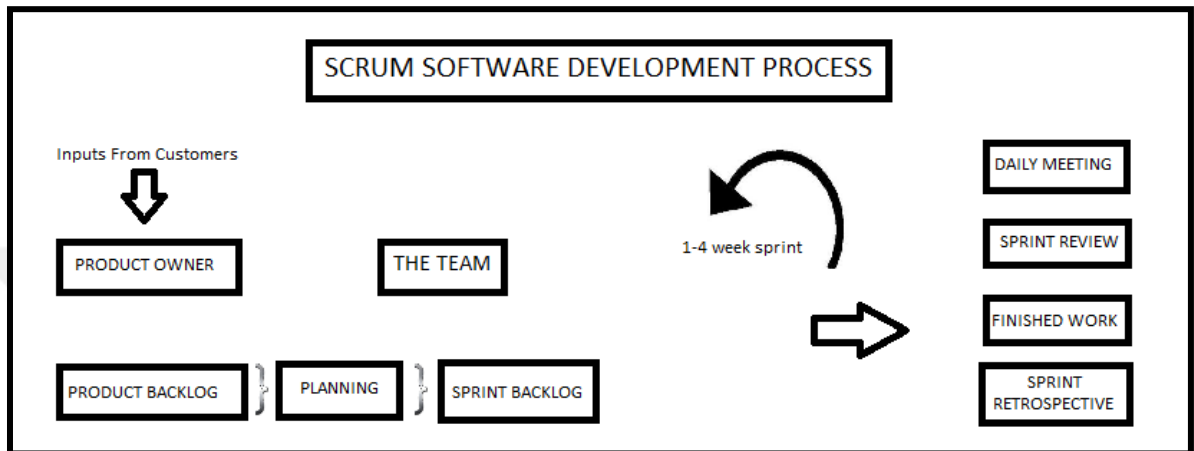
There are six attributes that make a Scrum Master good. SM should be responsible, modest, collaborative, committed, influential, and knowledgeable (Cohn, 2007).

#### **4.5 EVENTS**

All Scrum Events are time-boxed events which mean that every event has a maximum duration in order to create regularity and to minimize the need for undefined meetings. For example, Daily Scrum is limited by 15 minutes (Schwaber and Sutherland, 1991, 2013).

Once the sprint begins, there is no way to shorten or lengthen its duration. The other events may end if their purpose is achieved. Scrum development process is shown on Figure 4.3.

**Figure 4.3 : Scrum Development Process**



*Source :* (Katham,2014, <https://www.scrumalliance.org/community/articles/2014/december/pulse-check-product-owner-disappearance-syndrome> accessed 24 March 2015)

Each event in Scrum is an opportunity and adapts and inspects in order to increase the productivity. If any event is failed, it results reduced transparency and lost opportunity for inspection.

### 4.5.1 Sprint

A Sprint is the main event that contains and consists of the other events Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective during the Sprint. It is a time-boxed event for each sprint which takes one month or most common two weeks. It should be limited to one month because time zone is widened, complexity may rise and risk may increase.

The time for a sprint should be chosen before sprint starts. After a sprint starts, it cannot be changed.



Each sprint has a Sprint Goal and it must not be endangered. At the end of the Sprint, usable and potentially shippable product increment is created.

Sprints can be cancelable by only the PO although he or she may be influenced by stakeholders, the Development Team, or Scrum Master. A Sprint would be cancelled if the Sprint Goal becomes out of date, not usable. Sprint cancellations are very rare and often traumatic to the Scrum Team since everyone has to regroup in another Sprint Planning (Schwaber and Sutherland, 2013).

#### **4.5.2 Sprint Planning**

All Sprints are planned before each sprint is started in the Sprint Planning. This plan is formed by the Scrum Team in order to create Sprint Backlog and Sprint Goal.

Sprint Planning is also time-boxed event, limited eight hours for one-month sprint. Scrum Master ensures that the event takes place and attendants understand its purpose and he or she keeps it within the time-box (Schwaber and Sutherland, 2013).

Sprint Backlog consists of PBIs chosen by the development team to complete and success the Sprint Goal, it is a clear statement of work which will be done during the Sprint.

Sprint Planning should answer the questions of "What can be delivered from the upcoming sprint?" and "How much work needed to deliver the increment?"

In order to choose what can be delivered, according to the performance and the capacity of the Development Team, the items from PBL which is the input of Sprint Planning is selected.

When items are selected from PBL, the development team gives a word to complete the item as a "done".

### **4.5.3 Daily Scrum**

The Daily Scrum is a 15 minute time-boxed activity for the Development Team to plan for the next 24 hours during a sprint. It takes place at the same time and the same place each day to reduce complexity. During the event, every Development Team member explains:

- i. What did I do yesterday?
- ii. What will I do?
- iii. Do I have impediments that reduce Development Team Productivity?

Any impediment (block, risk, or issue) emerges during the Daily Scrum should be noted by the Scrum Master and shown on the Scrum Board of the team.

The Scrum Master ensures that The Development Team has the meeting and limits the meeting's duration to 15 minutes.

### **4.5.4 Sprint Review**

For one month Sprint, it is a four hour time-boxed meeting for shorter sprints, the event is usually shorter.

At the end of a Sprint, the Scrum Team and stakeholders can see what was done in the Sprint. Increment is presented by the PO and he or she explains what were "done" what was "not done" and the PO gets feedback and improvement ideas. The Development Team answers the questions of the increment.

At the end of the Sprint Review, the Product Backlog is revised by The PO and participants collaborate what to do next. It facilitates next Sprint Planning's items to emerge superficially.

#### **4.5.5 Sprint Retrospective**

Sprint Retrospective is a meeting that occurs at the end of the sprint. It is also 3 hour time-boxed event for one month sprint. For shorter sprints, it is usually shorter. The Scrum Master facilitates the event and participates as a peer team member for accountability of Scrum process. (Schwaber and Sutherland, 2013)

Sprint Retrospective provides a formal opportunity for inspection and adaptation for Development Team to improve further and to be more productive.

The Scrum Master also organizes the meeting and ensures that the purpose of the Sprint Retrospective is understood.

The purpose of this event is to:

- i. reflect on the last sprint
- ii. identify and agree on continuous process improvement actions.

There are some guidelines for sprint retrospectives in order to achieve the purpose of this meeting. Every team member expresses his or her ideas about everything in the last Sprint within these guidelines:

- i. The Wheel/Starfish
- ii. Sailboat
- iii. Mad Sad Glad
- iv. The Cool Wall
- v. Lean Coffee
- vi. Questions Retrospective
- vii. 4Ls

viii. Satisfaction Histograms

ix. Circles

x. Quartering

xi. Managing Retrospective Actions

Among these, the Starfish and the Mad Sad Glad techniques are the most commonly used ones. Starfish model divides the Scrum board into five categories as (Figure 4.4):

**Keep Doing :** It is a good starting point for the team members to focus on typically all the good things that they liked during the Sprint.

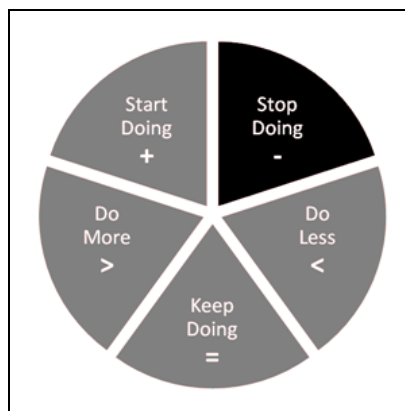
**Less Of :** Every team member has the idea that it should not be used or done in order to be more Agile.

**More Of :** If this happens more, it enables more productivity.

**Stop Doing :** Team members decide to give up doing this because it is not useful or it is an impediment for the team success.

**Start Doing :** If this happens, a more efficient, Agile and effective development environment can be achieved.

**Figure 4.4 : Starfish technique**



Source : (Lithespeed.com,2011)

The latter one is the Mad Sad Glad model. Here, the Scrum board is divided into 3 areas as:

**Mad** – frustrations, things that have annoyed the team and/or have wasted a lot of time

**Sad** – disappointments, things that have not worked out as well as were hoped

**Glad** – pleasures, things that have made the team happy

## **4.6 EXTENSIONS**

Although Extensions are not core practices in Scrum, they are software development best practices which are useful for an organization to have a more efficient and effective product development. They were suspended by Scrum.org in 2012 because Scrum.org believes that they are not needed in Scrum but without them the results cannot be desirable.

### **4.6.1 Backlog Refinement (Backlog Grooming)**

Backlog refinement (Backlog grooming) is the meeting which is the process of reviewing and checking the ongoing process of PBIs such that they are appropriately prioritized and are clear and executable for the Development Team.

Backlog refinement is not a core Scrum practice but has been adopted as a way of managing the quality of backlog items entering a sprint, with a recommended duration of up to 10 percent of the Sprint capacity (Cho 2009).

### **4.6.2 Scrum of Scrums (SoS)**

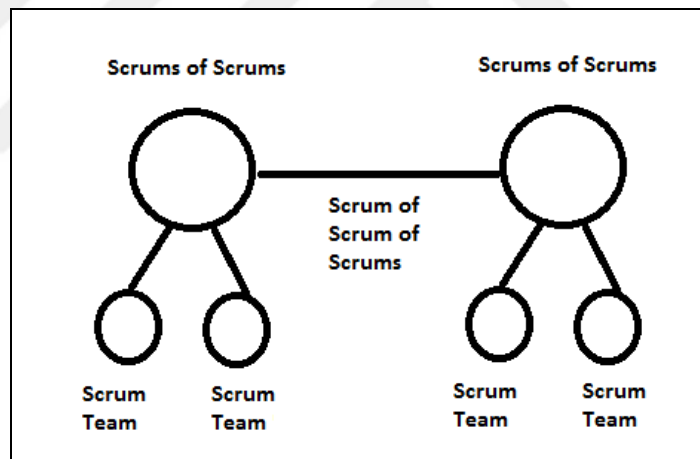
A Scrum of Scrums Meeting is a technique to scale Scrum up for multiple teams working on the same product (Figure 4.5). In this meeting, the Scrum teams discuss their interdependencies to focus on how to deliver the project collaboratively on generally in coordination and integration projects. There is one representative, usually the Scrum

Master, from each Scrum Team, sometimes if required; other members of Scrum Teams can participate in this meeting.

In this SoS meeting, like in a Daily Scrum, representatives of Scrum Teams talk about the issues such as:

- i. What did our team do?
- ii. What will our team do?
- iii. Are there any impediments that slow down my team?
- iv. Are there any impediments that slow down another team?

**Figure 4.5 : Scrum of Scrum of Scrums example**



## 4.7 ARTIFACTS

Artifacts in Scrum have the aim to maximize the transparency of information for everyone to easily understand and provide opportunities for inspection and adaptation. Scrum defines four artifacts: Product Backlog, Sprint Backlog, Increment, and the Definition of “Done”.

### 4.7.1 Product Backlog

The Product Backlog is an ordered and prioritized list of requirements, features, bug fixes, non-functional requirements, and enhancements that might be needed in the product to deliver a shippable product successfully. The PO is responsible for Product Backlog in organizing its content, availability, and ordering.

A Product Backlog is never complete. As long as a product exists, its Product Backlog also exists. A Product Backlog consists of PBIs and they should have the attributes of a clear description, order, estimate, value, size (estimation) and acceptance criteria to make clear what needs to be done. (Figure 4.6 shows a PBL example from Turkey).

The Development team has the responsibility of the estimation of the PBIs. The PO may influence them by helping it to understand but final estimation decision is made by Development Team. The estimation called sizing or the size of the PBI is often stated in story points using a rounded Fibonacci Sequence. Usually, a reference PBI is chosen for the next sizing of the other PBIs.

**Figure 4.6 : Product Backlog example from a bank in Turkey**

PBI ID	Status	SIZE	ITEM	Priority	Sprint No	Output Type	Entry Sprint	ACCEPTANCE CRITERIA
34721	Done	1	Ofsaa projesine destek	9.01	9	NF-Increment	9	İhtiyaç halinde Ofsaa ekibine destek verilmelidir.

34722	In Progress	1	Bir müşterinin bütün bankacılık işlerini tüm şubelerimizden yapabilmesinin sağlanması	9.02	9	NF-Increment	9	Müşterilerimiz tüm işlerini herhangi bir şubeden yapabilmeli.
-------	-------------	---	---	------	---	--------------	---	---

#### 4.7.2 Sprint Backlog

Sprint Backlog is a selected list of PBIs for the next sprint and also is a plan for delivering sprint increment and maintaining the Sprint Goal.

Sprint Backlog items should be selected in terms of the capacity of the Development Team. If the team cannot handle some of the Sprint Backlog Items in a Sprint, it should not be added to the Sprint Backlog because it decreases the velocity of the development team.

When the items selected by the development team, the tasks to increment these items should be decided and be given size and hours by the Development Team and be kept on the Scrum Board. Tasks on the sprint backlog are never assigned; they are set up by the team members according to its priority. It enables self-organization of the development team.

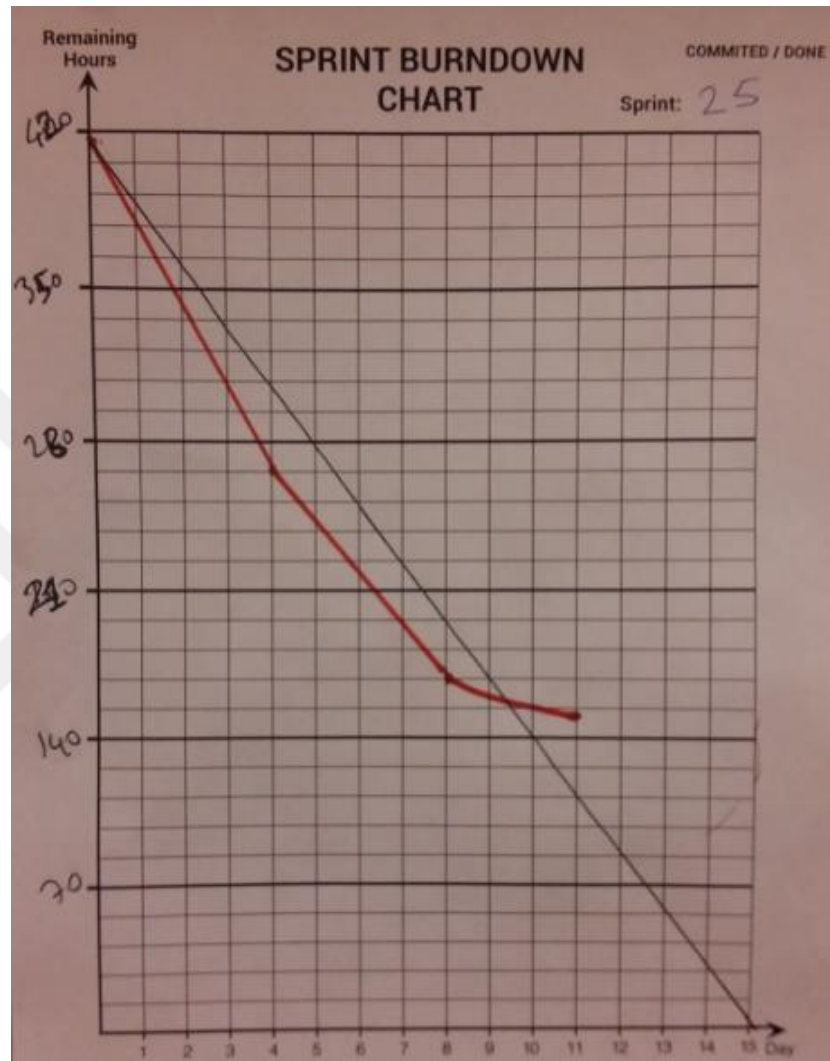
As new work is required, only Development Team can add it to the Sprint Backlog and only Development Team can change Sprint Backlog during the Sprint.

The Sprint Backlog is transparent and highly visible to everyone. It is a real time picture of what will be done by the Development Team.

In order to understand how Sprint goes, there is a Sprint Burn-Down Chart which is a publicly displayed chart that shows the remaining work in the Sprint Backlog (Figure 4.7). The Scrum Master draws ideal burn-down chart in the start of the sprint and day by day, he or she updates the remaining work in this chart. Anyone can understand how the Sprint goes and whether there are any extra works that team commits.



**Figure 4.7 : Sprint Burn down Chart from a bank in Turkey**



### 4.7.3 Definition Of "Done"

When a PBI or an increment is described as "Done", everyone understands what "Done" means. Although it can be different in different Scrum Teams, members must have a shared understanding of "Done" in their own team.

General Done Criteria could include any of the following (Satpathy, 2013):

- i. Completion of requirement analysis
- ii. Completion of all documentation
- iii. Reviewed by other team members
- iv. Completed development
- v. Completed unit testing
- vi. Completion of quality assurance tests
- vii. Completion of user acceptance tests
- viii. All issues are fixed
- ix. Successful demonstration to stakeholders and/or business representatives
- x. Successful Deployment of the product.

Before the first Sprint begins, the Definition of Done (DoD) must be determined by the Scrum Team. When DoD is determined, the team can ensure that a task is finished and the result meets the DoD. A clear DoD is critical because it helps standardization and helps for a better quality of the product.

DoD can be kept in the Scrum Board in order to remind the team the standards of “Done” criteria. Figure 4.8 shows a Scrum Board example from Turkey having a DoD in it.

**Figure 4.8 : Scrum Board Example in A Bank in Turkey**



#### **4.7.4 Increment**

The Increment is the sum of all the Product Backlog items completed during a Sprint. At the end of a Sprint, the new Increment must be “Done,” which means it must be in useable condition and meet the Scrum Team’s definition of “Done.” It must be in useable condition (shippable product) regardless of whether the Product Owner decides to actually release it (Schwaber and Sutherland, 2013).

## 5. SURVEYS & INTERVIEWS

In order to understand Scrum's usability and effectiveness, we prepared survey and some interviews with Scrum professionals. The survey consists of 36 questions, 34 of them are likert scale questions and 2 of them are open questions. Moreover, the interview consists of five open questions.

### 5.1. SURVEY

To study whether Scrum is a good framework in software development, there are 36 questions in which 16 multiple-choice questions to measure Scrum's efficiency and effectiveness, and plus 16 alike and the same multiple-choice questions to realize that the participants respond to questions appropriately and 1 multiple-choice question to understand the occupation group of participants, 1 multiple-choice question to learn that whether the participants take the Scrum Training and finally 2 open questions to get the comments of participants about the advantages and disadvantages of Scrum.

There are 229 participants who have fully responded our survey and 4 participants have not started and not completed the survey. We have applied this survey in 5 banks and banking companies (Türkiye Finans Participation Bank, Kuveyt Türk Participation Bank, Ziraat Technology, ING Bank and Akbank) in Turkey, who are/were adopting or trying to adopt Scrum framework as a software development methodology. The first question (see questions section) is to realize the role of the respondent and thus we can categorize the answers to the questions in terms of these roles (Table 5.1).

**Table 5.1 : The number of participants who respond our survey.**

Participant Group	# Participants
Scrum Master	24
Product Owner	20
Development Team	106
Business Unit	73
Other	2
Total	225

The second question is to learn whether the respondent has taken Scrum training or not (Table 5.2).

**Table 5.2 : The number of participants who took the Scrum training and who did not.**

Yes	159
No	67
Total	226

**Questions:**

1. What is my work title?
2. I took Scrum Training
3. Projects take longer time with Scrum.
4. Scrum is better in responding the rapid changes in software requirements
5. Scrum is compatible with COBIT, ITIL processes in banking applications
6. Scrum helps reduce the amount of time of projects
7. Teams are less motivated and have less performance with Scrum.
8. Overall (advantages, shortcomings) Waterfall is much better than Scrum
9. The other software development methodologies are better in responding the rapid changes in software requirements than Scrum.
10. Scrum enables transparency. I can see what has been done, what will be done, what the problems are in the project.
11. Scrum teams are self-motivated and self-organizing teams that learn anything on their own without any help from outside the team.
12. Sprint backlog is not up-to-date and it is not appropriate for its purpose.

13. Teams are more motivated and have more performance with Scrum.
14. Scrum does not enable transparency. I cannot see what is done, what will be done, what the problems are in the project.
15. Management supports Scrum in IT projects.
16. Scrum teams are not self-motivated and not self-organizing teams that are not capable of learning anything on their own without any help from outside of the team.
17. Continuing to work with Scrum is always useful.
18. The teams do not know what purpose their team works for.
19. Scrum increases assistance, cooperation and the collaboration of teams.
20. Scrum makes projects more qualified and more appropriate for their goal.
21. Sprint backlog is up-to-date and it is appropriate for its purpose.
22. Scrum reduces assistance, cooperation and collaboration of teams.
23. Scrum teams know what to do and realize well what goal the team has.
24. Product backlog is well-prepared, up-to-date and it is appropriate for its purpose.
25. Scrum decreases the projects' quality and also projects do not meet the requirements with Scrum.
26. Management does not support Scrum in IT projects.
27. Scrum teams do know Scrum very well and they give full support for anything and do all kinds of requirements.
28. Business units do know Scrum very well and they give full support for anything and do all kinds of requirements.

29. Product backlog is not well-prepared, not up-to-date and it is not appropriate for its purpose.
30. Scrum is not compatible with COBIT, ITIL processes in banking applications
31. Scrum teams do not know Scrum very well and they are not capable and not eager to do all kinds of requirements.
32. Continuing to work with Scrum is always useful.
33. Business units do not know Scrum very well and they are not capable and not eager to do all kinds of requirements.
34. Overall (advantages, shortcomings) Scrum is much better than Waterfall
35. What are the advantages of Scrum?
36. What are the disadvantages of Scrum?

**Responses to the questions:**

(0-Skipped 1-Strongly disagree 2-Disagree 3-Undecided 4-Agree 5-Strongly agree)  
 (Numbers in the table cells show the respondent number of each question)

**Table 5.3 : All responses to the questions**

Question	# respondents					
	0	1	2	3	4	5
3. Projects take longer time with Scrum.	2	23	115	66	20	3
4. Scrum is better in responding the rapid changes in software requirements	4	6	13	53	121	32
5. Scrum is compatible with COBIT, ITIL processes in banking applications	5	26	48	83	61	6
6. Scrum helps reduce the amount of time of projects	4	4	28	68	104	21
7. Teams are less motivated and have less performance with Scrum.	4	18	124	46	34	3
8. Overall (advantages, shortcomings) Waterfall is much better than Scrum	6	23	96	57	42	5
9. The other software development methodologies are better in responding the rapid changes in software requirements than Scrum.	6	12	112	54	41	4
10. Scrum enables transparency. I can see what is done, what will be done, what the problems are in the project.	6	3	7	32	142	39

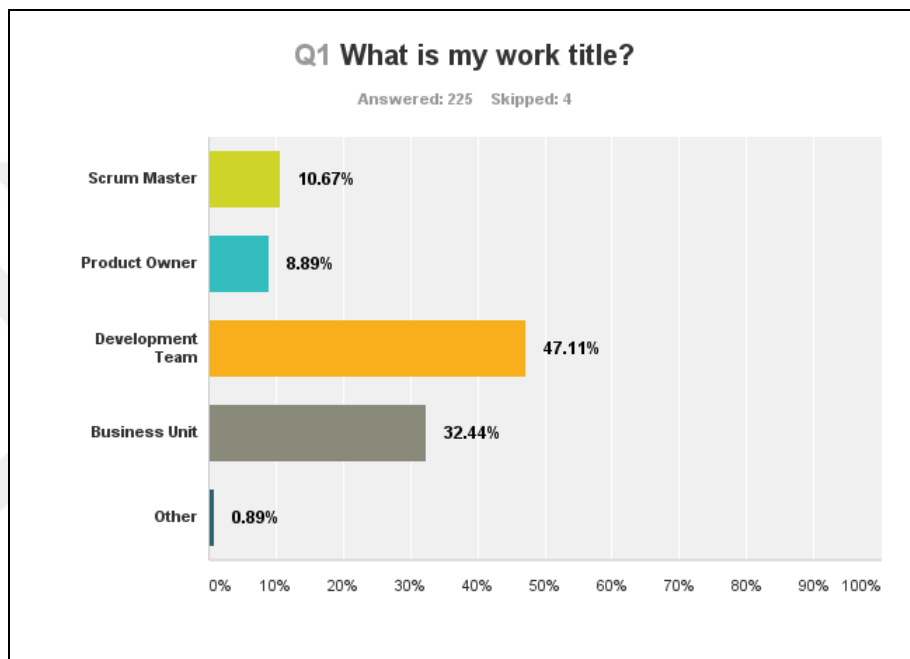
11. Scrum teams are self-motivated and self-organizing teams that learn anything on their own without any help from outside the team.	9	10	73	72	63	2
12. Sprint backlog is not up-to-date and it is not appropriate for its purpose.	7	8	98	72	37	7
13. Teams are more motivated and have more performance with Scrum.	7	5	22	74	103	18
14. Scrum does not enable transparency. I cannot see what is done, what will be done, what the problems are in the project.	6	36	120	44	20	3
15. Management supports Scrum in IT projects.	6	3	18	61	121	20
16. Scrum teams are not self-motivated and not self-organizing teams that are not capable of learning anything on their own without any help from outside the team	8	7	73	70	63	8
17. Continuing to work with Scrum is always more harmful.	7	30	120	50	19	3
18. The teams do not know what purpose their team work for.	7	30	121	42	26	3
19. Scrum increases assistance, cooperation and collaboration of teams.	8	4	15	43	136	23
20. Scrum makes projects more qualified and more appropriate for their goal.	6	3	29	65	111	15
21. Sprint backlog is up-to-date and it is appropriate for its purpose.	7	3	30	63	115	11
22. Scrum reduces the assistance, contribution and collaboration of teams.	9	32	122	46	18	2
23. Scrum teams know what to do and realize well what goal the team has.	9	0	22	51	128	19
24. Product backlog is well-prepared, up-to-date and it is appropriate for its purpose.	10	7	40	70	94	8
25. Scrum decreases the projects' quality and also projects do not meet the requirements with Scrum.	12	21	115	54	24	3
26. Management does not support Scrum in IT projects.	10	19	92	62	40	6
27. Scrum teams do know Scrum very well and they give full support for anything and do all kinds of requirements.	9	6	44	92	75	3
28. Business units do know Scrum very well and they give full support for anything and do all kinds of requirements.	8	20	79	80	35	7
29. Product backlog is not well-prepared, not up-to-date and it is not appropriate for its purpose.	11	6	90	70	46	6
30. Scrum is not compatible with COBIT, ITIL processes in banking applications	11	5	45	81	64	23
31. Scrum teams do not know Scrum very well and they are not capable and not eager to do all kinds of requirements.	9	10	105	62	39	4
32. Continuing to work with Scrum is always useful.	9	3	26	76	94	21
33. Business units do not know Scrum very well and they are not capable and not eager to do all kinds of requirements.	10	3	40	59	100	17
34. Overall (advantages, shortcomings) Scrum is much better than Waterfall	9	7	18	63	106	26



## 5.2 ANALYZING DATA

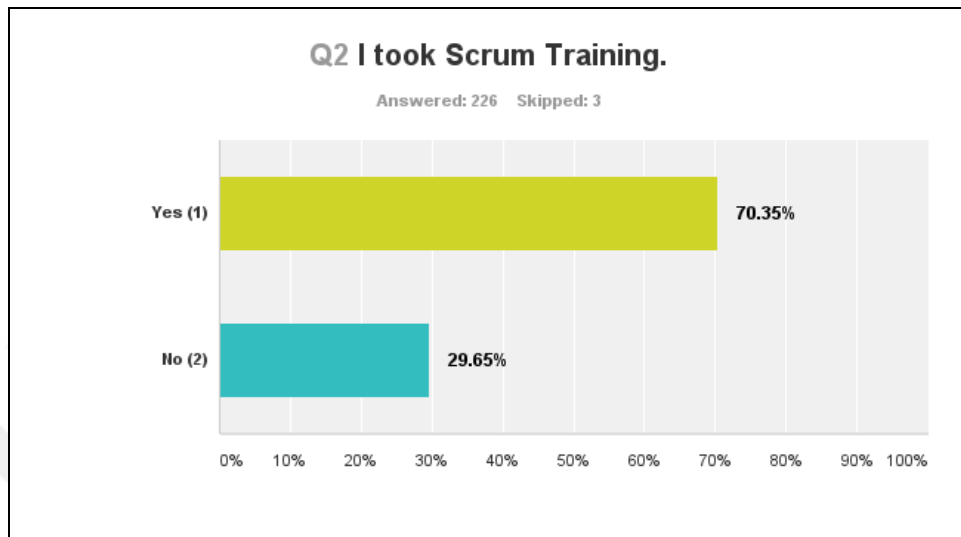
There are 225 participants who have answered the first question and their percentages Scrum Roles are given:

**Figure 5.1 : The graph shows the participants' Scrum roles in percentages**



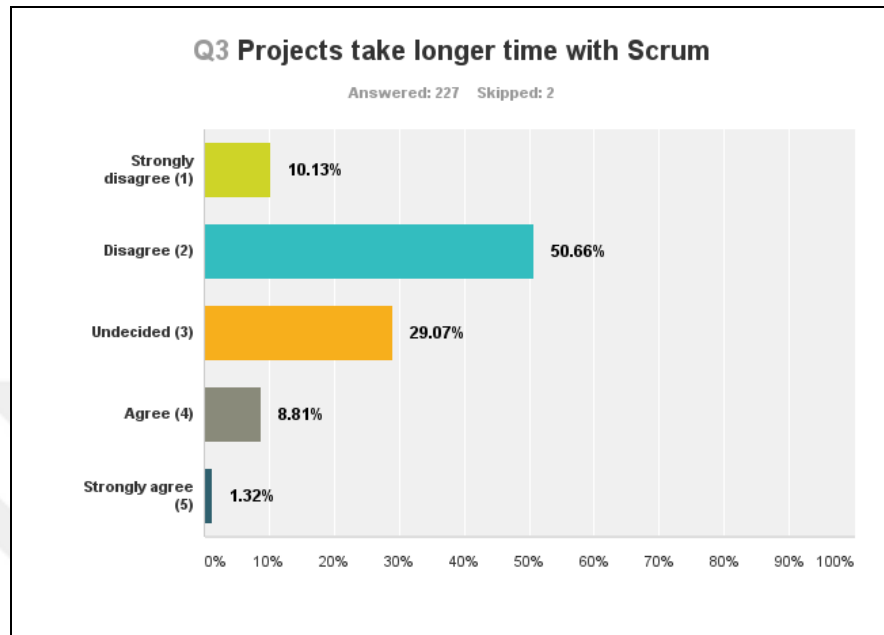
Most of the participants are from Development Teams with 47.11 percent and the second one is from Business Unit with 32.44 percent, the third group is Scrum Masters with 10.67 percent, the fourth one is Product Owners with 8.89 percent and the Others as last one, is the smallest group with 0.89 percent.(Figure 5.1) The distribution of the survey group is very good because we can discover all answers according to all Scrum Roles and the data is consistent such that we have many respondents from all the Scrum Roles.

**Figure 5.2 : The graph shows the participants who take Scrum Training in percentage**

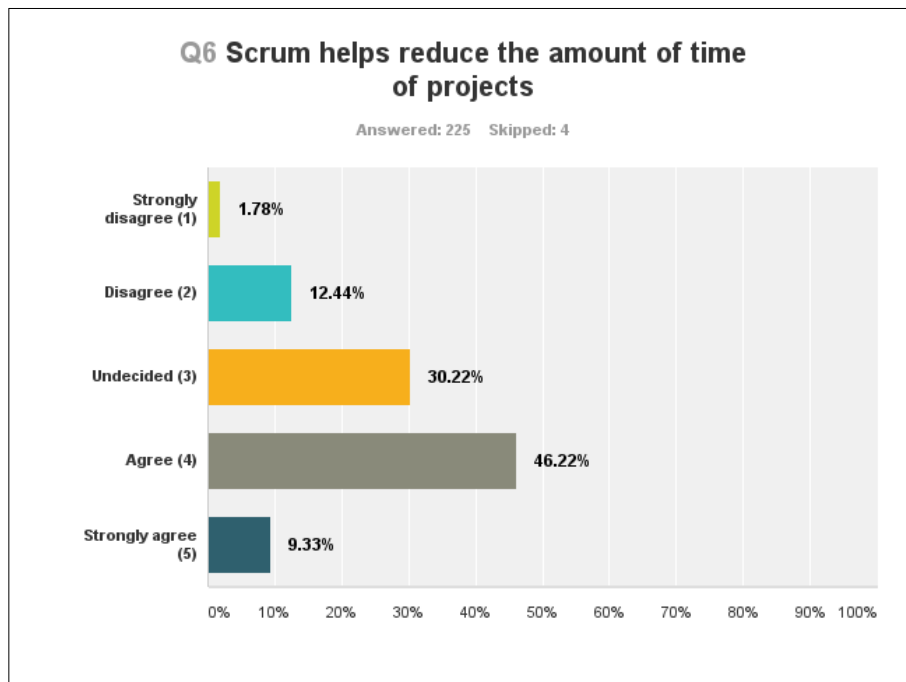


It is an advantage for the study to have that most of the participants, about 70 percent, has had the Scrum Training and hence we assume that most of them are knowledgeable about Scrum and its framework. Moreover, when we look at the "Yes" answers according to the roles, Product Owner percentage is 85 percent, Scrum Master Percentage is 87.50 percent, development team percentage is 96.23 and business unit percentage is 23.61. (Figure 5.2) However, Business units should be more aware of Scrum and should take Scrum training.

**Figure 5.3 : This graph shows the all answer distribution of Q3**



**Figure 5.4 : Responses to Q6**



### **Analyzing the responses to Q3 and Q6**

50.66 percent of respondents disagree with the idea and 29.07 percent of them are undecided. Moreover, 8.81 percent of them agree with and 1.32 percent of them strongly agree with the idea of the question (Figure 5.3). We can say that most of the respondents do not believe "projects take longer time with Scrum". The average is 2.41 out of 5.

On the other hand, about 65 percent of the Product Owners who respond the Q3 disagree, 5 percent strongly disagree, 15 percent undecided, 15 percent agree and surprisingly none of the POs strongly agree with the idea of this question. Moreover, when we look at the answers of Scrum Masters to Q3, 29.17 percent strongly disagree, 45.83 percent disagree, 16.67 percent undecided, 4.17 percent agree and 4.17 percent strongly agree. Also, when it comes to the answers of Business Unit members to Q3, 1.45 percent of them strongly disagree, 11.59 percent of them disagree, 42.03 percent are undecided, 42.03 percent of them agree and 2.90 percent of them strongly agree.

Moreover, 46.22 percent of participants agree that Scrum helps reduce amount of time in projects, 9.33 strongly agree, and 30 percent of them are undecided (Figure 5.4). When we look at the answers of Product Owners, 10.53 percent disagree, 15.79 percent are undecided, 68.42 percent agree and 5.26 percent strongly agree that Scrum decrease the amount of time to finish projects. When it comes to Scrum Masters' responses, 37.50 percent are undecided, 37.50 percent agree and 25 percent strong agree. Business unit answers differ. 2.74 percent strongly agree, 13.70 percent disagree, 36.99 percent are undecided, 39.73 percent agree and 6.85 percent strongly agree with the idea of Scrum's ability to reduce the project time. We can say that most of the participants except some of the business unit members believe with this idea.

Figure 5.5 : Responses to Q4

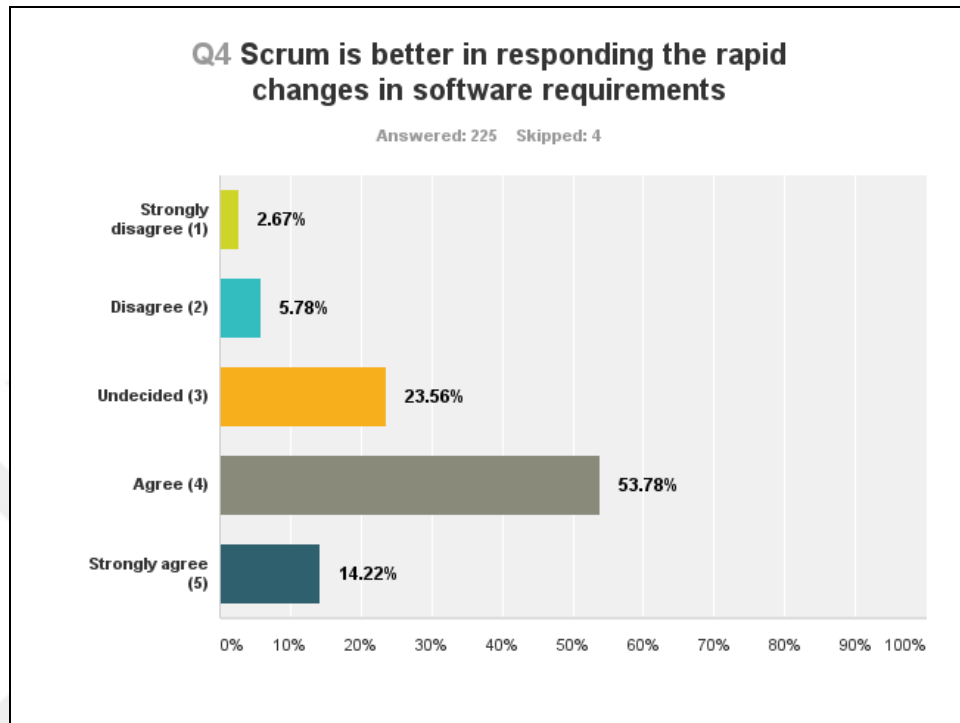
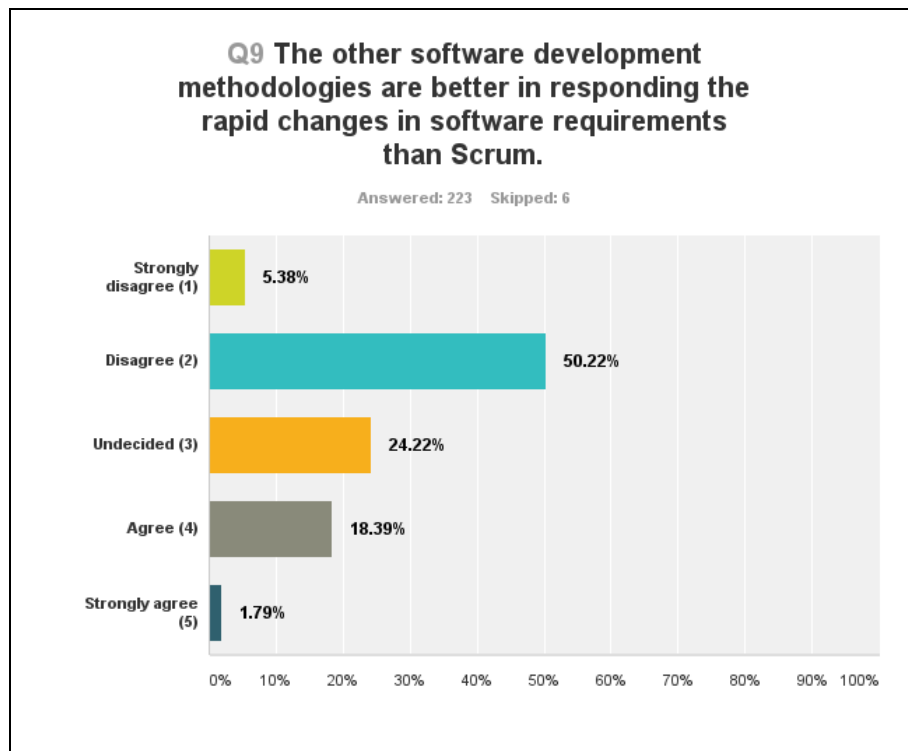


Figure 5.6 : Responses to Q9



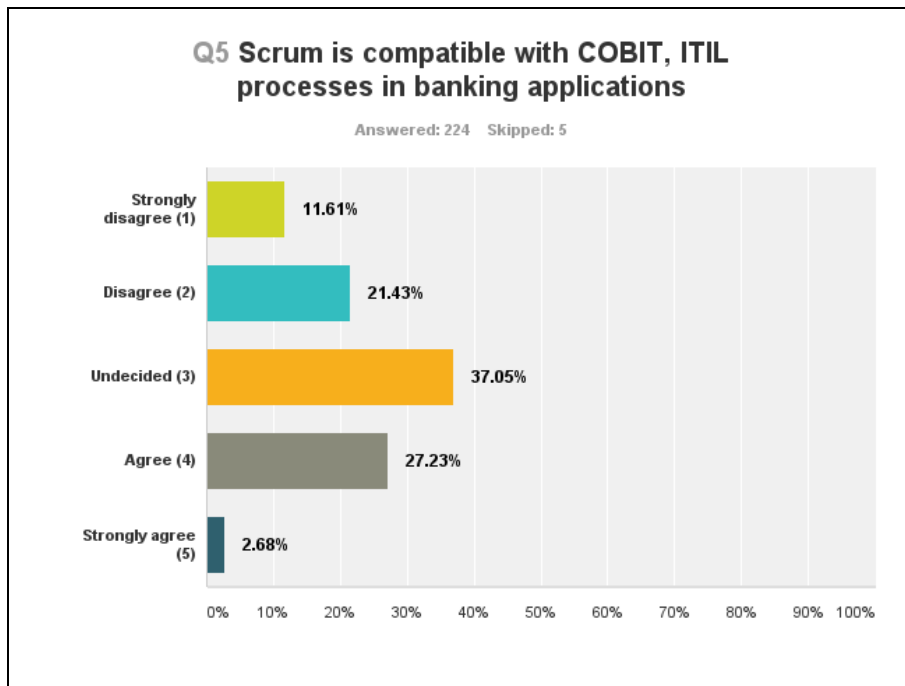
### Analyzing the responses to Q4 and Q9

In the beginning of the survey, we have said that some of the questions are alike or the same to make sure that all participants respond the questions fairly and properly. Now, we can easily see that the responses are so close.

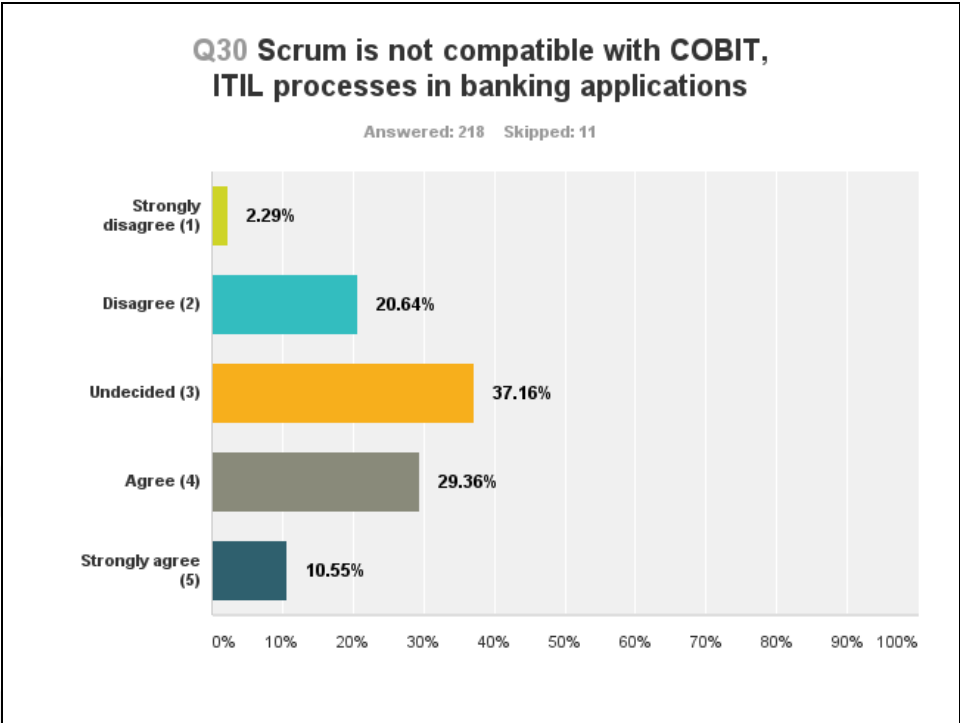
53.78 percent of the respondents agree that Scrum is better in responding sudden changes. Also, about 23 percent of them are undecided whether Scrum is better or the others are better. (Figure 5.5) On the other hand, 50.22 percent of them disagree that the other Software development methodologies are better than Scrum. (Figure 5.6)

Furthermore, 55 percent of Product Owners agree and 15 percent of them strongly agree with this idea of Q4. Also, 58.33 percent of Scrum Masters agree and 37.50 of them strongly agree with this idea of Q4. We can say that Scrum Masters strongly believe in Scrum's power in responding the rapid changes in software requirements.

**Figure 5.7 : Responses to Q5**



**Figure 5.8 : Responses to Q30**



**Analyzing the responses to Q5 and Q30**

It is clear from the two graphs above that most of the respondents are undecided about Scrum's compatibility in Control Objectives for Information and Related Technology (COBIT), Information Technology Infrastructure Library (ITIL) processes. The percentage of undecided is about 37 percent. 27.23 percent of the participants agree that Scrum is compatible with these processes and 10.55 percent of them choose to strongly agree as an answer to the Q5 (Figure 5.7). On the other hand, 50 percent of Product Owners believe that Scrum is compatible with COBIT and ITIL processes in banking applications, 30 percent of Product Owners are undecided. All in all, we can say that most of the respondents do not exactly know the competency of Scrum on these processes.

Figure 5.9 : Responses to Q7

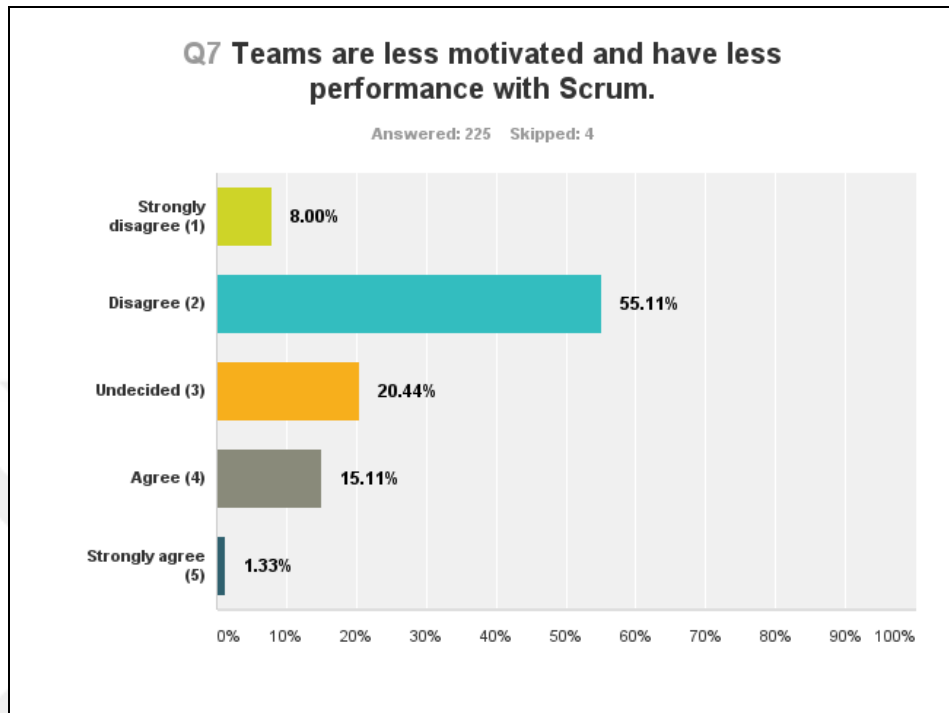
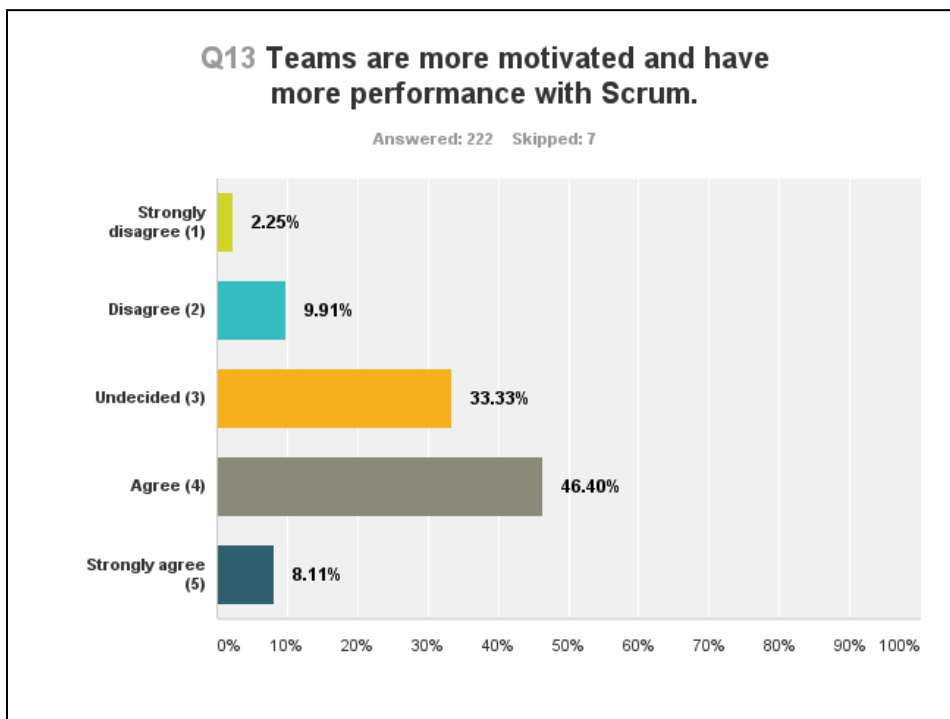


Figure 5.10 : Responses to Q13



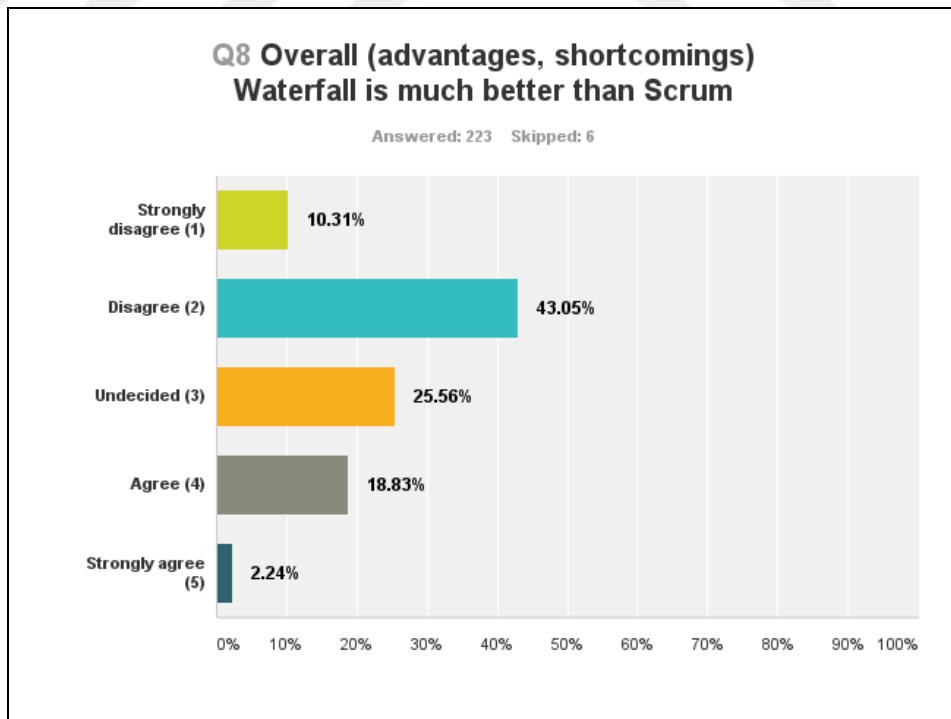


### Analyzing the responses to Q7 and Q13

According to the Q7, 55.11 percent of participants disagree with the idea that Teams are less motivated and have less performance with Scrum, 8 percent strongly disagree, 20.44 are undecided, 15.11 percent agree and 1.33 percent strongly agree with this idea (Figure 5.9). We can say that most of the participants do not believe that motivation and performance are not related to Scrum.

When we consider the answers to Q7 with Q13 together, Teams are more motivated and have more performance with Scrum, 46.40 percent agree and 8.11 percent strongly agree, 33.33 percent are undecided, 9.91 percent disagree and 2.25 percent strongly disagree with the question. To sum up, we can say that 54.51 percent of participants believe that Scrum enables motivation and high performance. However, there are a 33.33 percent of participants who are undecided about this.

**Figure 5.11 : Responses to Q8**

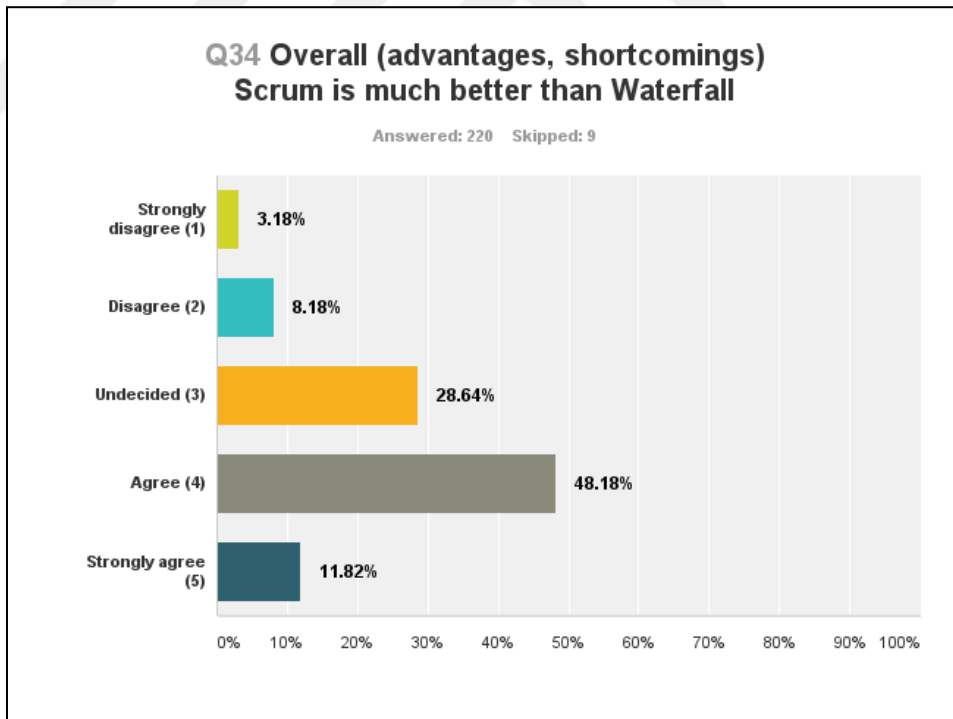


### Analyzing the responses to Q8

When we investigate the results of Q8, we can say that most of the participants do not believe that Waterfall is much better than Scrum, 43.05 percent of them disagree and 10.31 percent strongly disagree. On the other hand, 25.56 percent of them are undecided. 18.83 percent of them agree that Waterfall is better and 2.25 percent of them strongly agree (Figure 5.11).

On the other hand, 41.67 percent of Scrum Masters disagree Waterfall is better than Scrum and 29.17 of them strongly disagree with this question. As a result, the respondents disagree the idea that Overall, in terms of advantages and disadvantages, Waterfall is better than Scrum.

**Figure 5.12 : Responses to Q34**



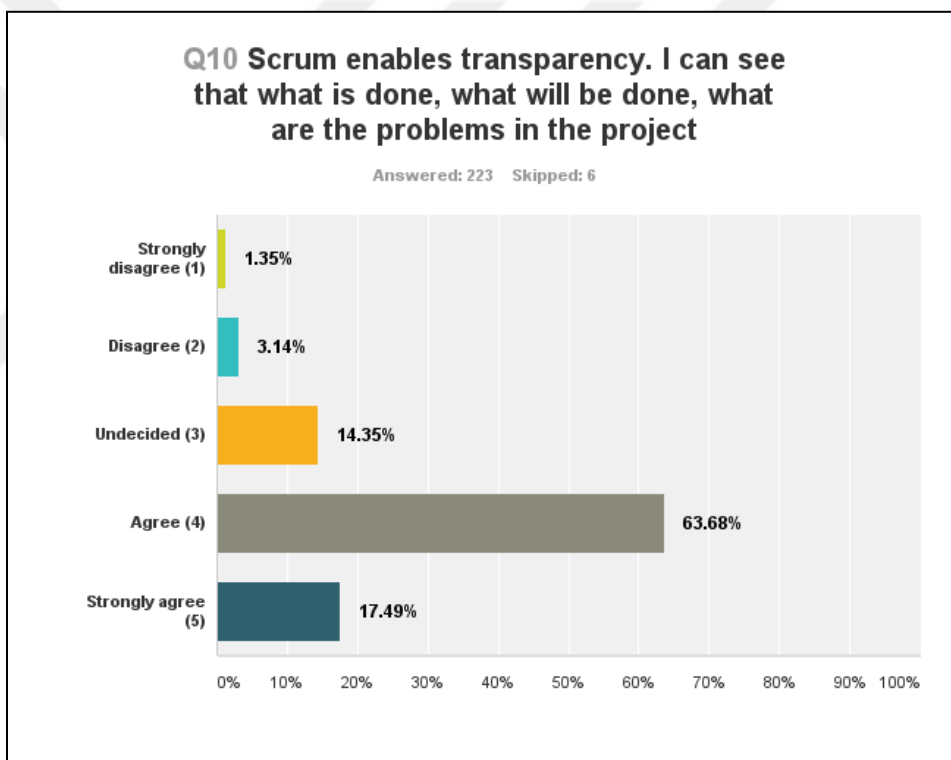
### Analyzing the responses to Q34

When we analyze the results of Q34, most of the participants believe that Scrum is much better than Waterfall. 48.18 percent agree, 11.82 percent strongly agree with this question.

60 percent of them believe that Scrum is much better. Also, we have 28.64 percent of undecided participants who have no idea whether Scrum is much better or not (Figure 5.12).

On the other hand, 41.67 of Scrum Masters agree and 41.67 of them strongly agree with the idea of Overall Scrum is better than Waterfall. Also, 52.63 of Product Owners agree and 15.79 of them strongly agree with this idea.

**Figure 5.13 : Responses to Q10**

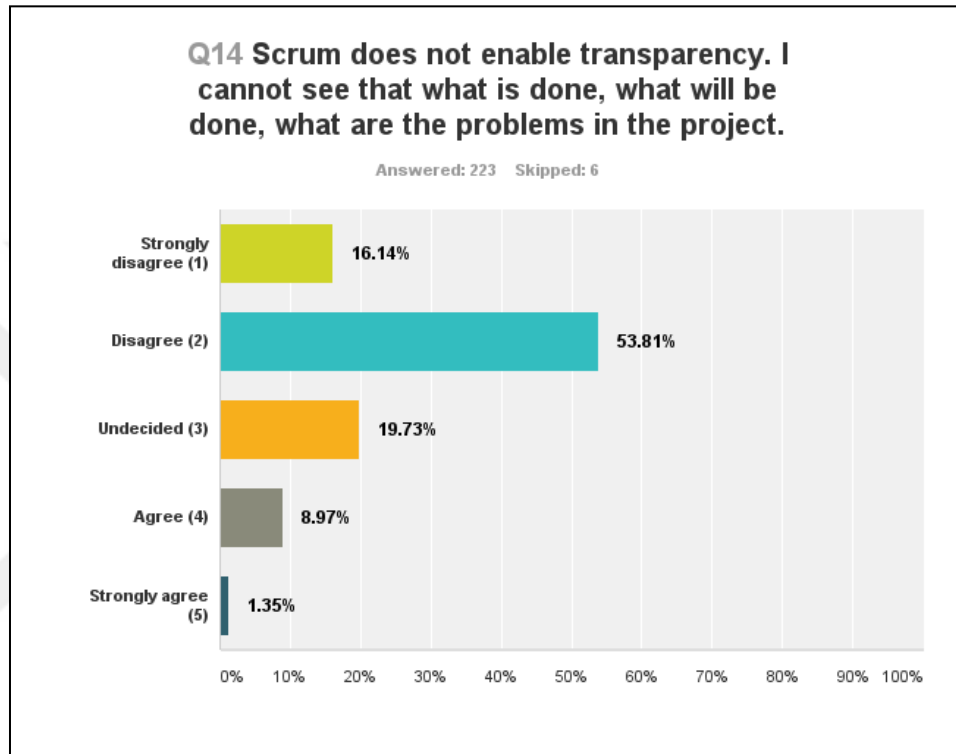


### **Analyzing the responses to Q10**

According to responses to the question 10, most of the respondents believe that Scrum enables transparency by which they can see what is done, what will be done what the problems are that they will encounter. 63.68 percent of them agree and 17.49 percent strongly agree with this question. Totally, more than 80 percent of them believe that Scrum

enables transparency in projects. There are 14.35 percent undecided respondents (Figure 5.13).

**Figure 5.14 : Responses to Q14**



### Analyzing the responses to Q14

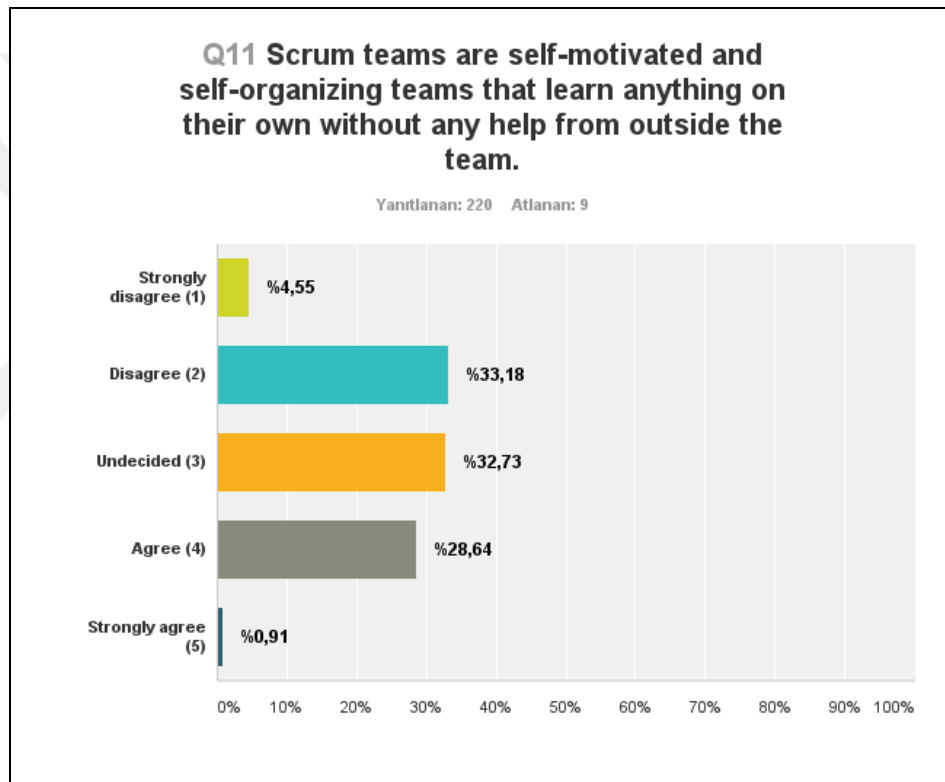
On the other hand, when we ask the question reversely, most of the people do not believe that Scrum does not enable transparency. About 70 percent of them disagree with this idea. There are undecided ones, too (Figure 5.14).

When we look at the responses of Product Owners, 10 percent of them strongly disagree, 55 percent of them disagree, 25 percent of them are undecided and 10 percent of them agree with this idea. Moreover, according to Scrum Master Responses, 37.50 percent of them strongly disagree, 37.50 of them disagree, 16.67 percent of them are undecided and 8.33 of them agree with this idea that Scrum does not enable transparency. Moreover, 5.71 of Business Unit members strongly disagree, 48.57 of them disagree, 28.57 are undecided,

14.29 of them agree and 2.86 of them strongly agree that Scrum does not enable transparency in which the progress can be observable by all the stakeholders.

In summary, we can easily say that Scrum enables transparency and most of the people believe that they can see what stage the project is, what will be done in the project, and they can see the impediments in projects.

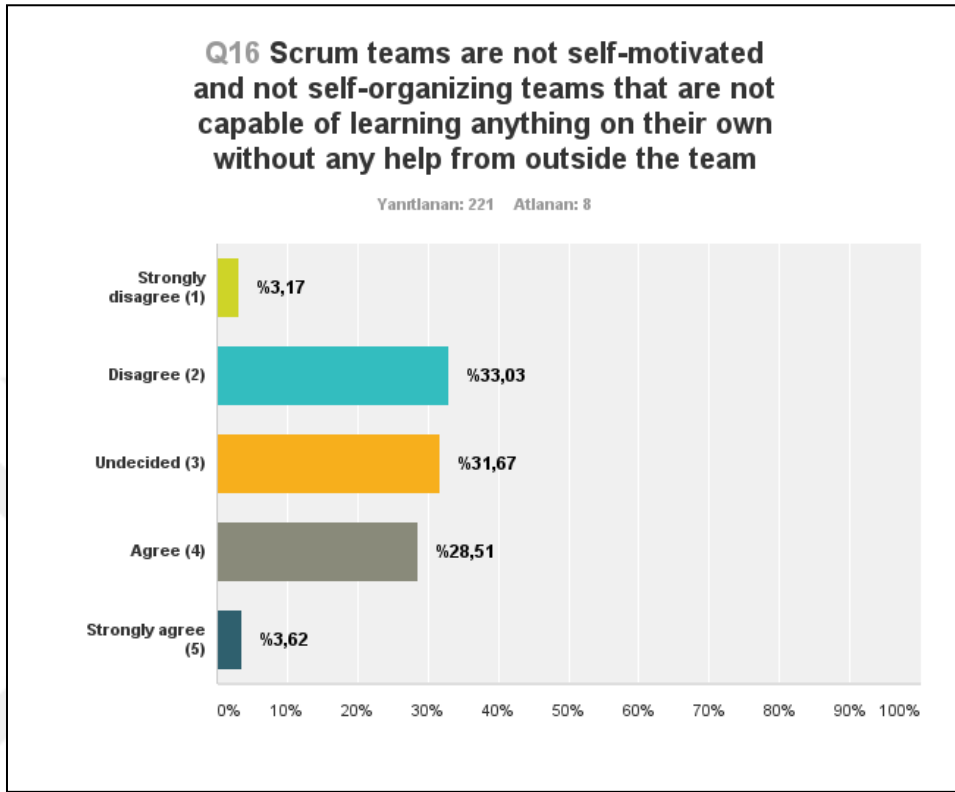
**Figure 5.15 : Responses to Q11**



### **Analyzing the responses to Q11**

According to the results of Q11, most of the respondents do not believe that Scrum Teams are self-motivated teams that learn anything without any help from outside the team (it is 33.18 percent disagree and 4.55 percent strongly disagree with this idea). On the other hand, 28.64 percent of them believe this idea and 0.91 strongly believe. Moreover, we have 32.74 percent of respondents who are undecided about this idea (Figure 5.15).

**Figure 5.16 : Responses to Q16**

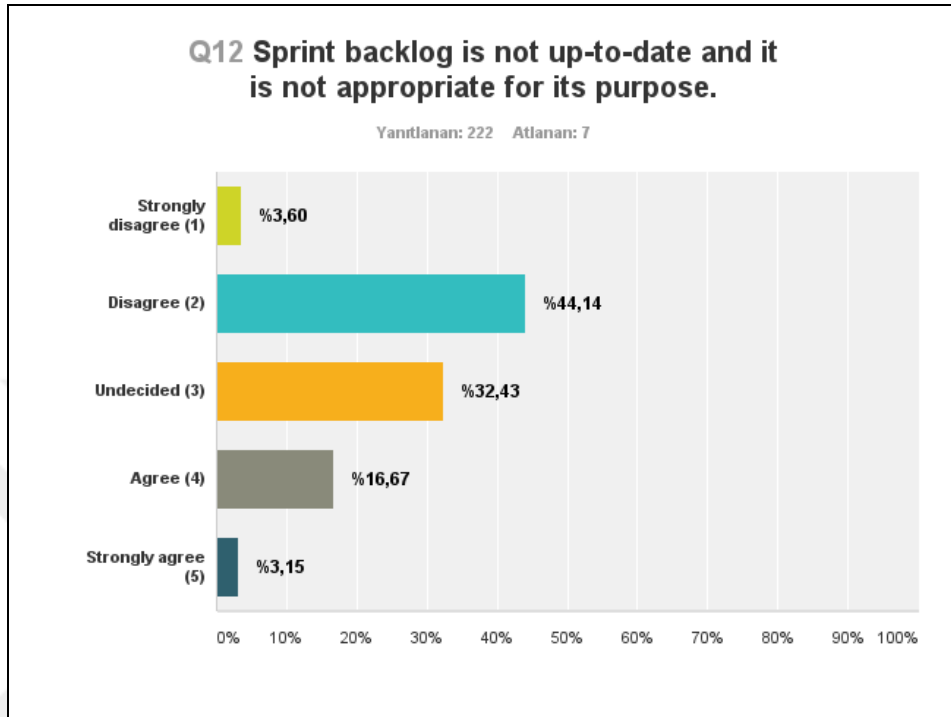


### **Analyzing the responses to Q16**

When we look at the responses of Q16, most of the respondents disagree with the statement that Scrum Teams are not self-motivated and also we have 31.67 of undecided respondents (Figure 5.16). Furthermore, 40 percent of Product Owners are undecided. However, 50 percent of Scrum Masters disagree and 4.17 of them strongly disagree with this idea. On the other hand, according to the responses of Business Unit members, 48.53 percent of them are undecided and 16.18 percent of them disagree and 27.94 percent of them agree with this idea.

When we look at the results of both Q11 and Q16, we cannot decide whether Scrum Teams are self- motivated or not.

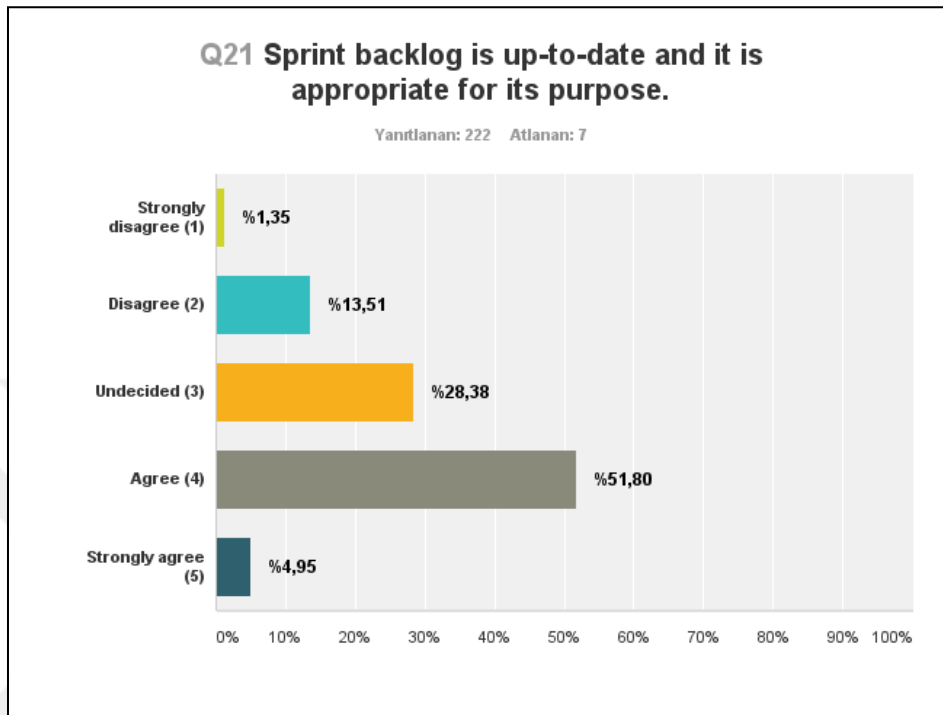
**Figure 5.17 : Responses to Q12**



**Analyzing the responses to Q12**

According to Q12, most of the respondents do not believe that Sprint backlog is not up-to-date and it is not well-prepared for its purpose (Figure 5.17). Moreover, 32.43 percent of them undecided, 16.67 percent of them agree and 3.15 percent of them strongly agree with this idea. On the other hand, 50 percent of Scrum Masters disagree and 12.50 percent of them strongly disagree, 16.67 percent of them are undecided, 20.83 percent of them agree that Sprint backlog is not up-to-date and not suitable for its purpose. Also, 5 percent of Product Owners strongly disagree, 35 percent of them disagree, 30 percent of them undecided and 30 percent of them agree with this idea. According to the Business Unit responses, 1.43 percent strongly disagree, 31.43 percent disagree, 55.71 percent are undecided, 10 percent agree and 1.43 percent strongly agree that Sprint Backlog's up-to-date status and its real purpose.

**Figure 5.18 : Responses to Q21**



### **Analyzing the responses to Q21**

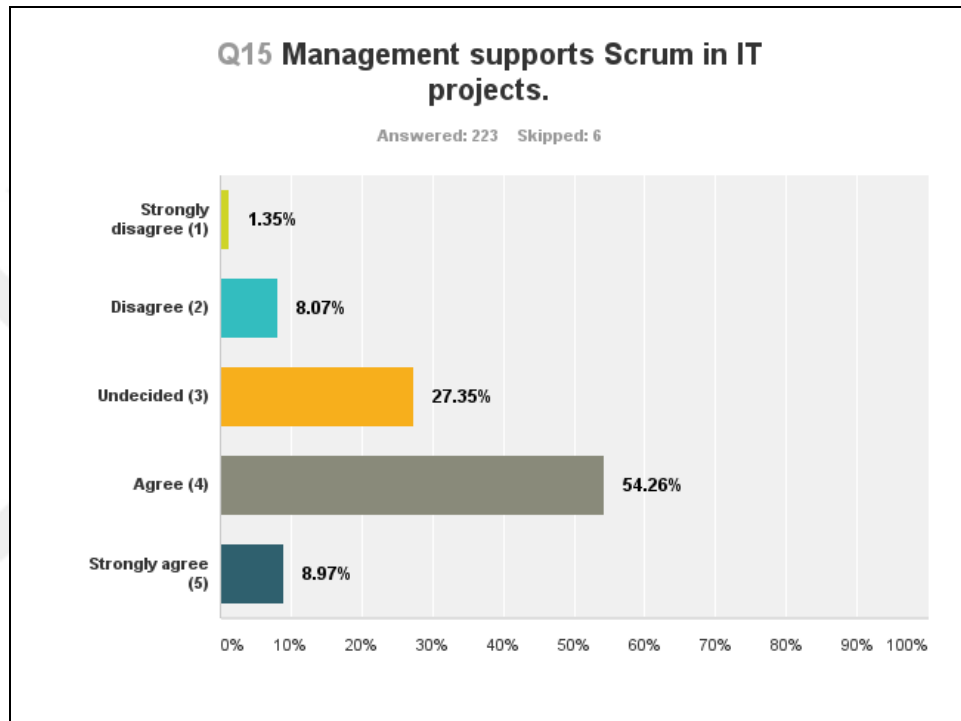
When we analyze the Q21, we can easily see that most of the respondents believe that Sprint Backlog is well-prepared and up-to-date. Still, we have 28.38 percent of people who are undecided (Figure 5.18). Besides, 35 percent of Product Owners agree, 30 percent of them undecided. Furthermore, 58 percent of Scrum Masters agree and 20.83 percent of them strongly agree that Sprint Backlog up-to-date.

When we look at the responses of each Scrum roles, 5 percent of Product Owners strongly disagree, 25 percent of them disagree, 30 percent of them are undecided, 35 of them agree and 5 percent of them strongly agree. Moreover, according to the responses of Scrum Masters, 8.33 percent disagree, 12.50 percent undecided, 58.33 percent agree and 20.83 percent strongly agree. On the other hand, responses rates of business unit members, 5.80 percent disagree, 50.72 percent are undecided, 40.58 percent agree and 2.90 percent strongly agree that Sprint Backlog is up-to-date and its purpose is appropriate.



To summarize, we might not say that the Sprint Backlog is very sufficient completely. In my opinion, I agree to a certain level that Sprint Backlogs have some problems; they are neither up-to-date nor well-prepared.

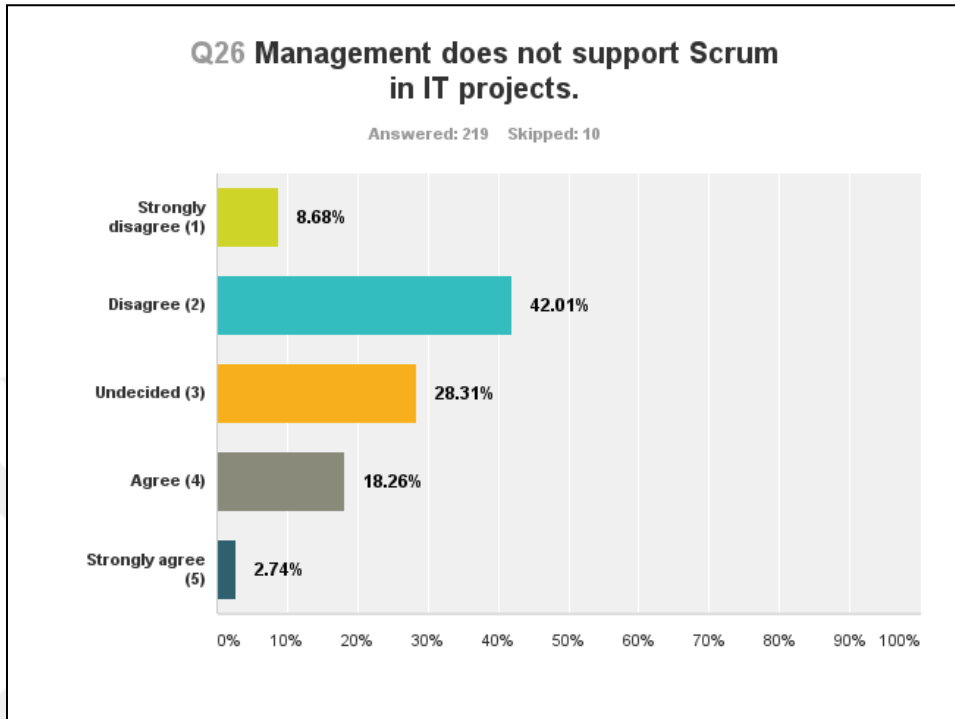
**Figure 5.19 : Responses to Q15**



### Analyzing the responses to Q15

When we look at the responses of the Q15, we can see that most of the participants believe that Management supports Scrum. 54.26 percent agree, 8.97 strongly agree (Figure 5.19). Moreover, 60 percent of Product Owners agree and 15 percent of them strongly agree that management supports Scrum. Also, 58.33 percent of Scrum Masters agree and 8.33 percent of them strongly agree with this idea. Management support is very important while adopting Scrum because Scrum is hard to implement and for this reason, management should be patient and supportive to have better results with Scrum.

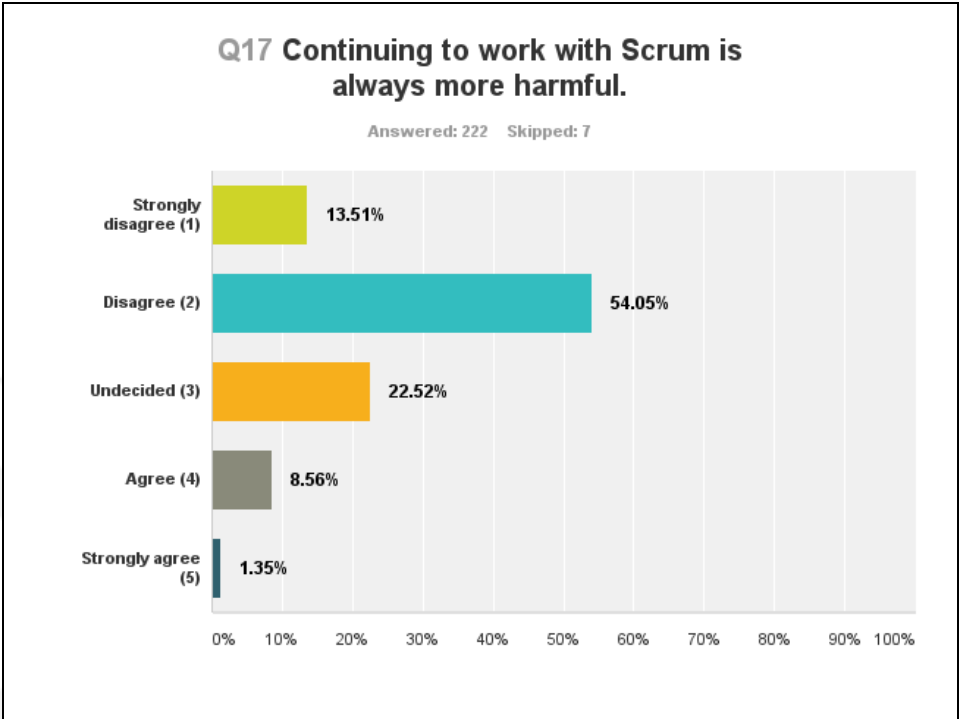
**Figure 5.20 : Responses to Q26**



**Analyzing the responses to Q26**

On the other hand, about 42 percent of participants disagree that Management does not support Scrum in IT projects (Figure 5.20). Similarly, 45.83 percent of Scrum Masters disagree with this idea. Also, 52.83 percent of Product Owners disagree with this idea. Furthermore, 50 percent of business unit members are undecided and 26.47 percent of them disagree about this. Hence, depending on the results of Q15 and Q26, we can say that Management have support Scrum but they should be more supportive in the organizations.

**Figure 5.21 : Responses to Q17**

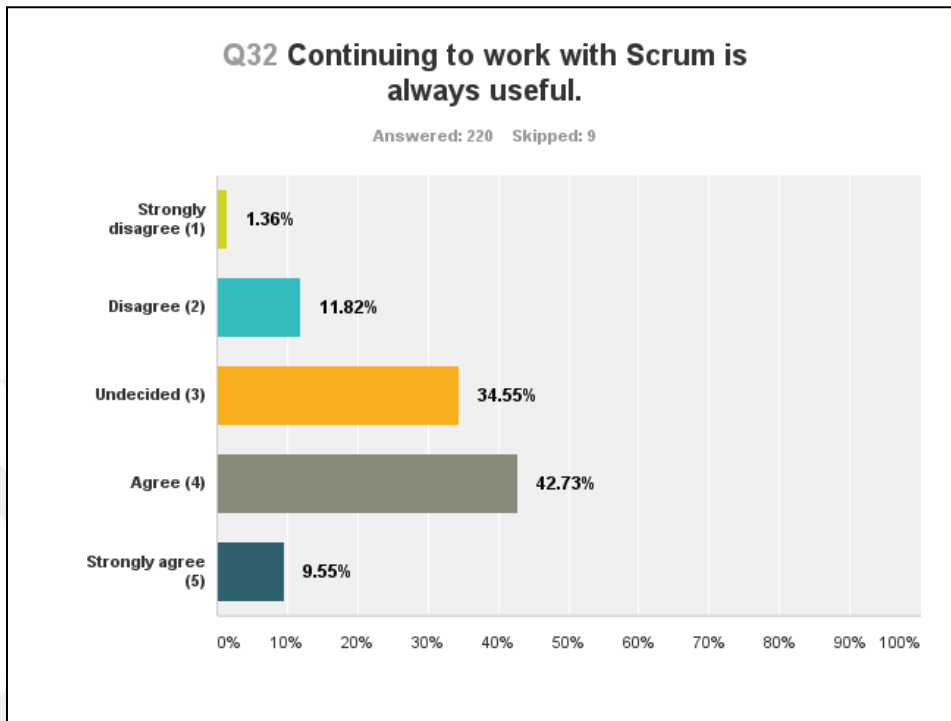


**Analyzing the responses to Q17**

According to the responses of Q17, most of the attendants do not believe that continuing with Scrum is always more harmful. 54.05 percent of them disagree, 13.51 percent of them strongly disagree, and totally 69.56 percent disagree with this idea. However, fewer amounts of them agree and strongly agree with this idea (Figure 5.21).

On the other hand, 65 percent of Product Owners disagree with this idea. Similarly, more than 66 percent of Scrum Masters disagree. Surprisingly, 40 percent of Business Units are undecided and 34.29 percent of them disagree. For these reasons, we can say that most of the participants do not believe that Continuing Scrum framework is harmful but business units should be more aware of Scrum and its usefulness.

**Figure 5.22 : Responses to Q32**

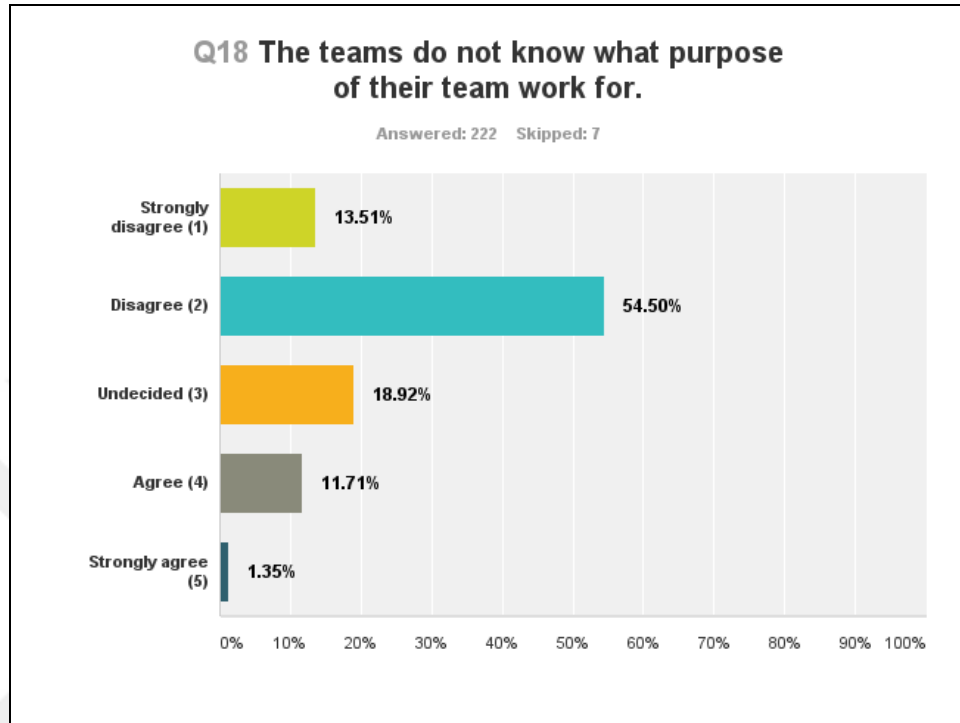


### **Analyzing the responses to Q32**

On the other hand, most of the people believe that Continuing with Scrum is always useful. About 52.28 percent of them agree with this question (Figure 5.22). When we look at the answers of each role carefully, 42.11 percent of Product Owners agree and 36.44 percent of them are undecided. Moreover, 45.83 percent of Scrum Masters agree and 25 percent of them strongly agree with this idea. Differently, 42.03 percent of Business unit members are undecided, 42.03 of them agree with this idea and 2.03 of them strongly agree with this.

In summary, we can say that more than half of the attendants believe that Continuing with Scrum is useful but some participants have some doubts that continuing with Scrum will be very useful.

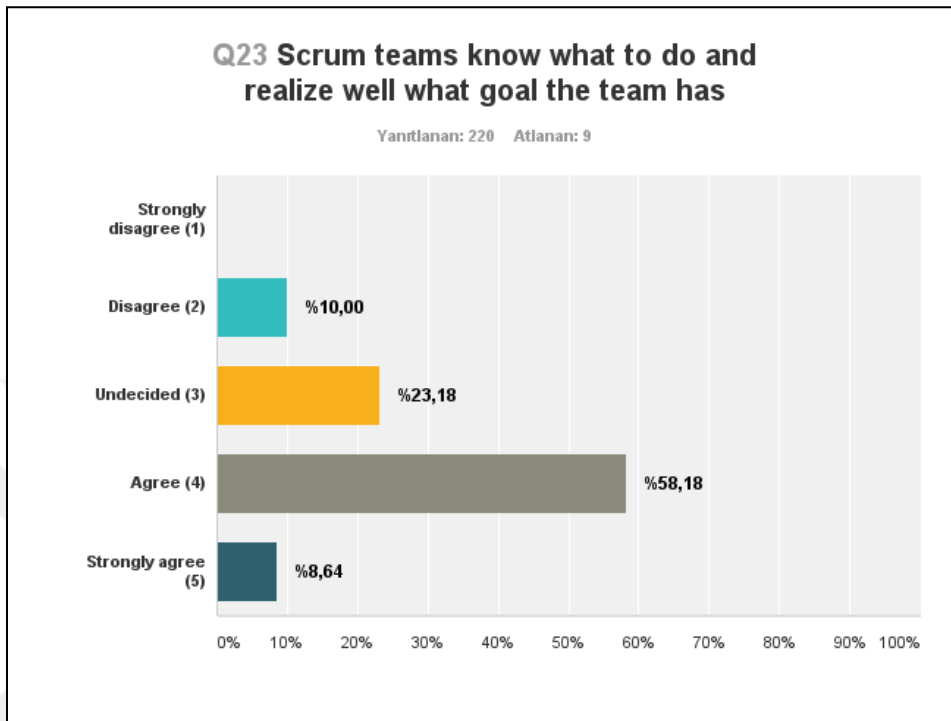
**Figure 5.23 : Responses to Q18**



### **Analyzing the responses to Q18**

According to the answers of the Q18, 54.50 percent of participants disagree and 13.51 percent of them strongly disagree that Scrum teams do not know what purpose their team works for (Figure 5.23). Similarly, 55 percent of Product Masters disagree and 15 percent of them strongly disagree with this idea of Q18. On the other hand, Scrum Masters have different responses. 30.43 percent of them strongly disagree and 34.78 percent of them disagree. Also, Business unit members have dissimilar aspect to the Q18. 11.43 of them strongly disagree, 44.29 percent of them disagree, 32.86 of them are undecided, 8.57 of them agree and 2.86 of them strongly agree the teams' consciousness about their Scrum purpose.

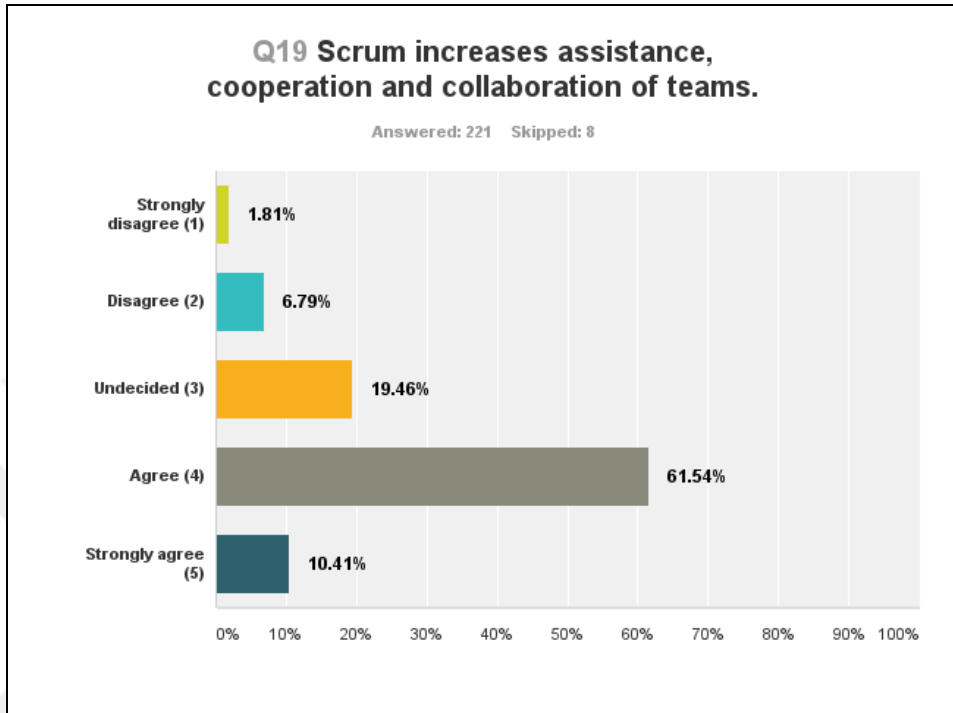
**Figure 5.24 : Responses to Q23**



**Analyzing the responses to Q23**

Besides, 64.72 percent of the participants do believe that Scrum teams know what to do and realize well and what goal the team has (Figure 5.24). According to responses of Product Owners, 73.68 percent of them believe the idea. Also, 58.33 percent of Scrum Masters believe and 61.66 percent of Business Unit members agree the idea that Scrum Teams know their goal and they can accomplish it if they persevere to succeed.

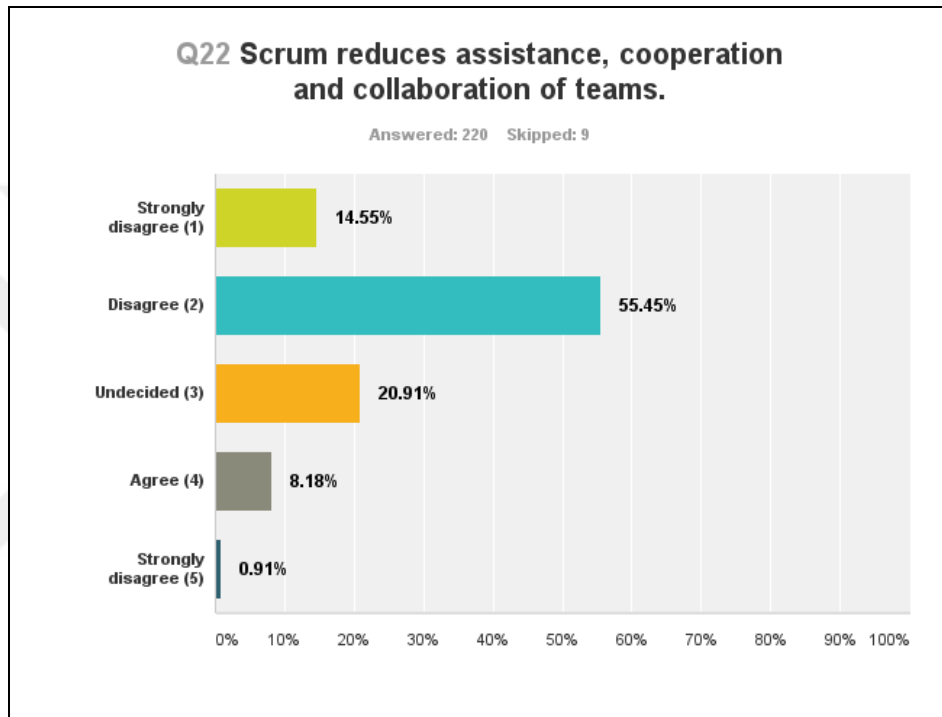
**Figure 5.25 : Responses to Q19**



**Analyzing the responses to Q19**

In the lights of the responses of Q19, most of the participants believe that Scrum increases the level of assistance, cooperation and collaboration in teams (Figure 5.25). It is very significant. Thus, a team can be more motivational, can have higher performance and can solve their problems with these good characteristics. When we look at the responses of each role, 70 percent of Product Owners agree and 10 percent of them strongly agree. According to responses of Scrum Masters, 45.83 percent agree and 25 percent strongly agree. Also, 62.32 percent of Business Unit responses agree and 4.35 percent strongly agree that Scrum's power of increasing cooperation, collaboration and motivation.

**Figure 5.26 : Responses to Q22**

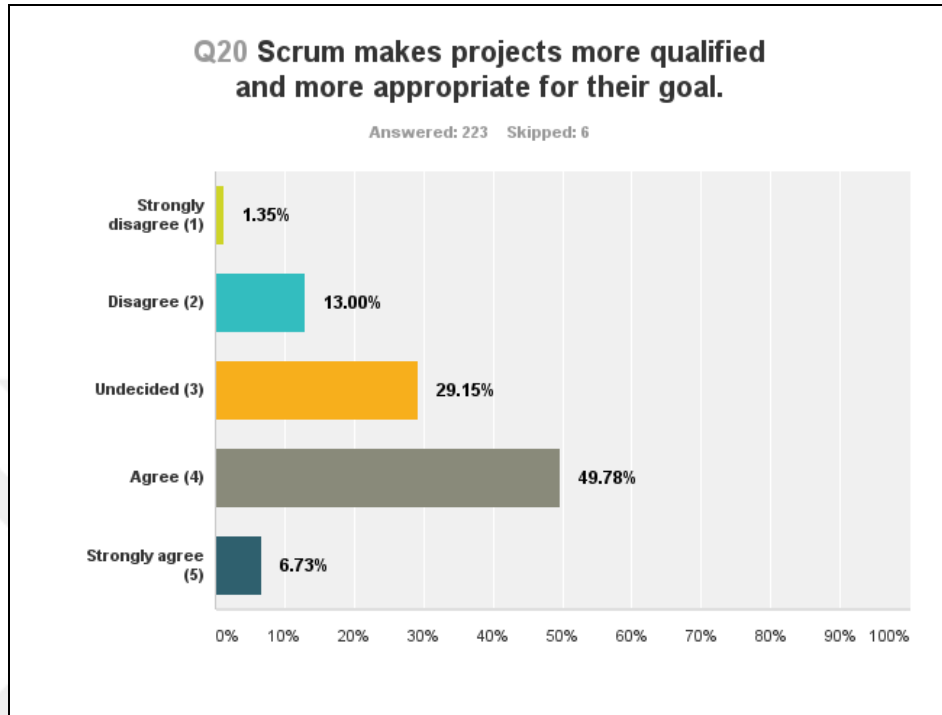


### **Analyzing the responses to Q22**

Furthermore, most of the respondents do not believe that Scrum reduces assistance, collaboration and cooperation in development teams (Figure 5.26). For example, 60 percent of Product Owners disagree, 75 percent of Scrum Masters disagree and 50 percent of Business Unit responses disagree with this idea of Q22. All in all, we can say that with the implementation of Scrum, the members of the teams are assisted, cooperated and collaborated among themselves.



**Figure 5.27 : Responses to Q20**

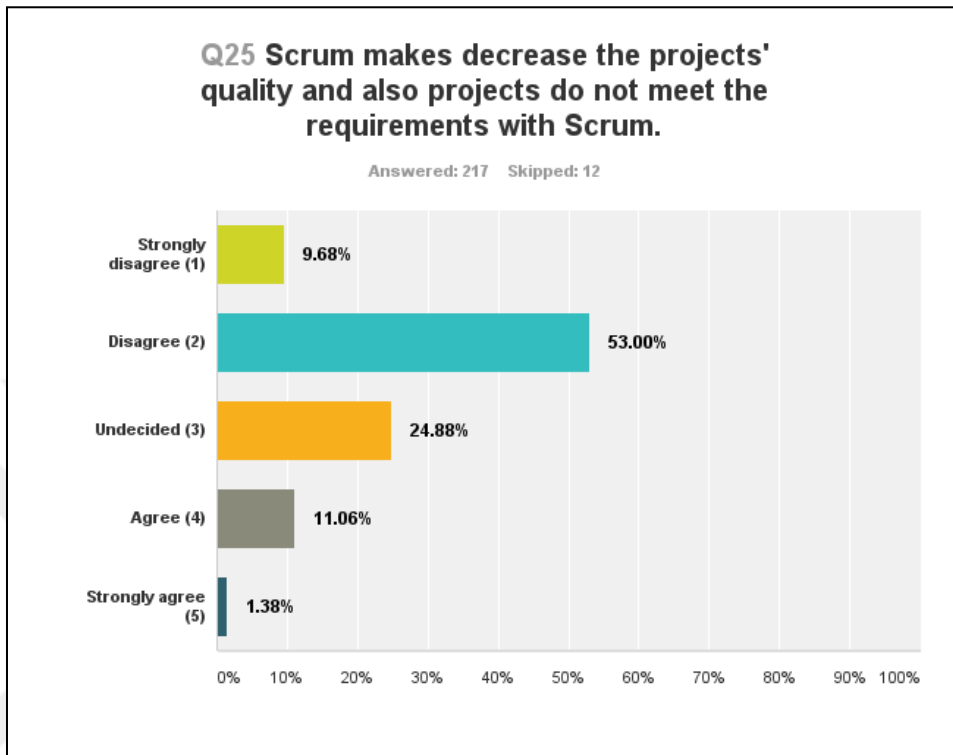


**Analyzing the responses to Q20**

In the lights of the answers of the Q20, most of the participants with 49.78 percent, believe that Scrum adds projects more quality and the purpose of projects is more understandable with Scrum. On the other hand, there are 29.15 percent undecided attendants who do believe or do not believe this. Also, 13 percent of them disagree and 1.35 percent of them strongly disagree with this idea (Figure 5.27).

According to the answers of Product Owners, 60 percent of them agree, 5 percent of them strongly agree, 10 percent of them are undecided, 25 percent of them disagree. Moreover, 50 percent of Scrum Masters agree, 20.83 percent of them strongly agree, 20.83 of them are undecided, 4.17 of them disagree and 4.17 of them strongly disagree. On the other hand, 44.29 percent of Business unit members agree, 4.29 of them strongly agree, 35.71 of them are undecided 15.71 percent of them disagree with that Scrum makes projects more qualified.

**Figure 5.28 : Responses to Q25**

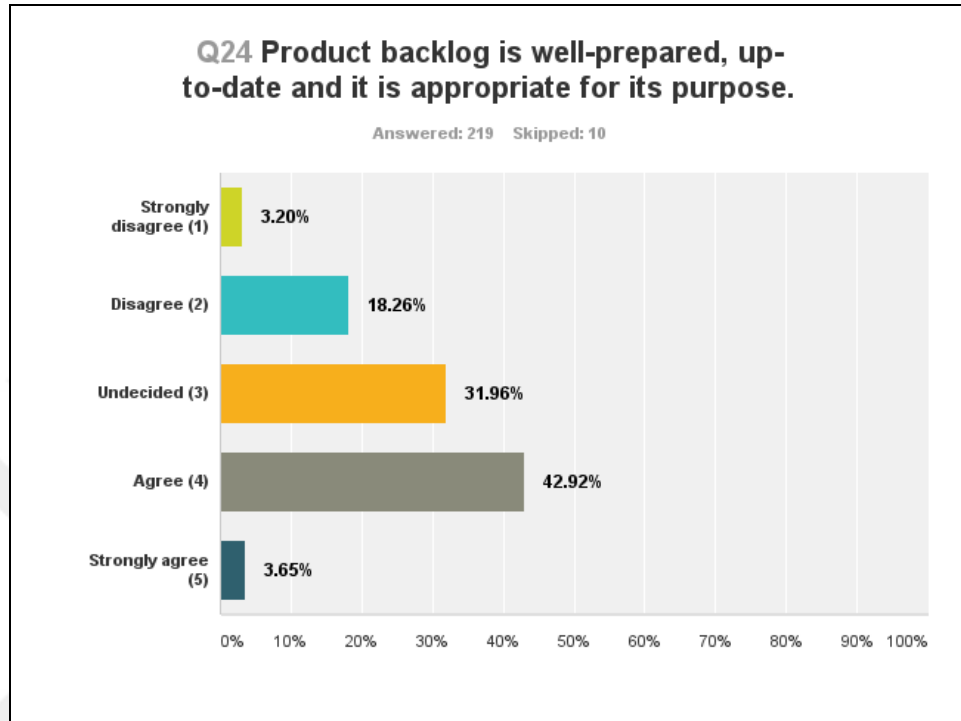


**Analyzing the responses to Q25**

In another point of view, most of the participants with 53.00 percent do not believe that Scrum takes from the quality of the projects and also the projects do not meet the requirements with Scrum. There are still undecided people (approximately 25 percent) who do not believe or do believe this idea. However, there are 11.06 percent who agree that Scrum reduces projects' quality and also projects do not meet the requirements with Scrum framework (Figure 5.28).

In summary, we can say that Scrum increase software project quality, and with Scrum, requirements of projects meet the expectations of customers.

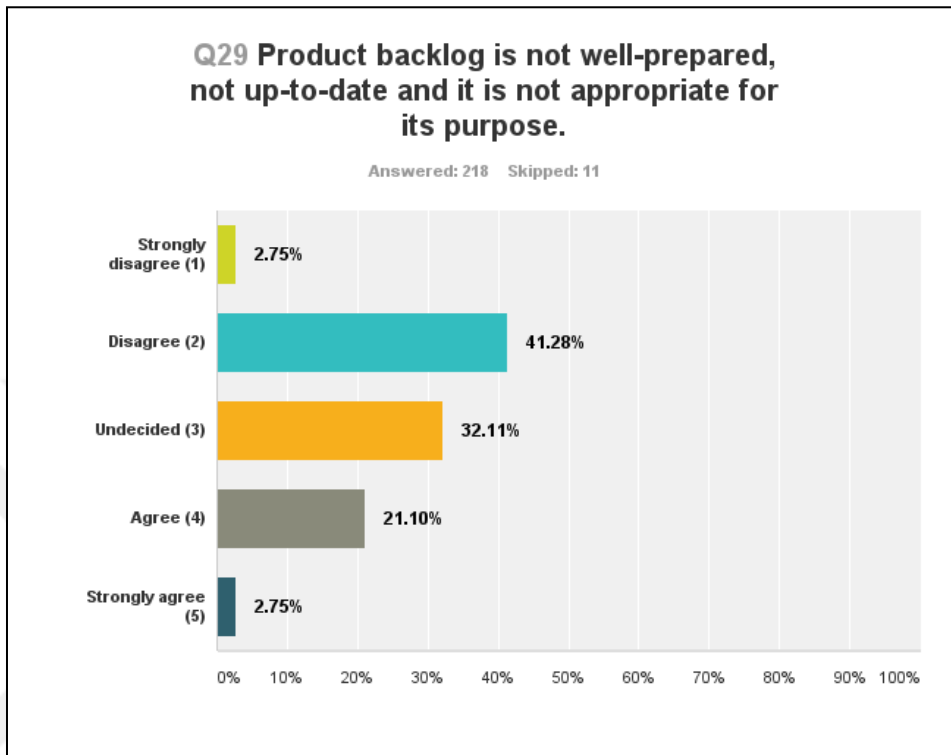
**Figure 5.29 : Responses to Q24**



### **Analyzing the responses to Q24**

According to the results of Q24, 42.92 percent of respondents agree and 3.65 percent of them strongly agree with the idea that Product Backlog is well-prepared, up-to-date and it is appropriate for its purpose. There are 31.96 percent people who are undecided and 18.26 percent who disagree and 3.20 percent that strongly disagree with this idea (Figure 5.29). Similarly, 47.37 percent of Product Owners agree, 5.26 percent of them strongly agree, 21.05 percent of them are undecided, 26.32 percent of them disagree. According to the responses of Scrum Masters, 37.50 percent agree, 4.17 percent strongly agree, 16.67 percent are undecided, 37.50 percent disagree and 4.17 percent strongly disagree. Also, 39.13 percent of Business Unit members agree, 4.35 strongly agree, 50.72 percent of them are undecided, 4.35 of them disagree, 1.45 of them strongly disagree that product backlog is prepared well. In summary, we can say that most of the people have doubts about product backlogs' quality and its aim's suitability for Scrum Team.

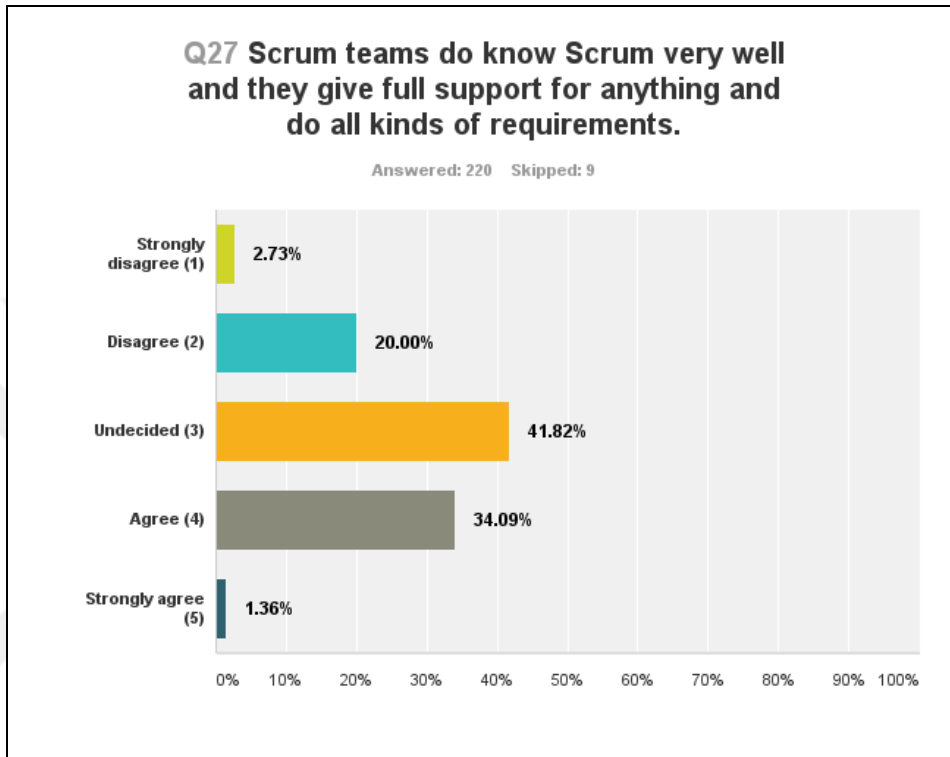
**Figure 5.30 : Responses to Q29**



### **Analyzing the responses to Q29**

According to the answers of Q29, 41.28 percent of the participants do not agree with that the product backlog is not well-prepared and not up-to-date. There are 32.11 percent of the participants who are undecided. Also, 21.10 percent of them believe and 2.75 percent of them strongly believe that the product backlog is not well-prepared and not updated regularly (Figure 5.30). On the other hand, 50 percent of Product Owners disagree, 5.56 percent of them strongly disagree, 22.22 percent of them are undecided, 22.22 percent of them agree. 37.50 percent of Scrum Masters disagree, 4.17 of them strongly disagree, 12.50 percent of them are undecided, 37.50 percent of them agree, 8.33 percent of them strongly agree. Furthermore, 48.53 percent of Business Units are undecided, 35.29 percent of them disagree, 1.47 percent of them strongly disagree, 11.76 percent of them agree, 2.94 percent of them strongly agree with this idea of Q29. To summarize, we can say that most of the participants suspicious of the product backlogs' quality and that it is updated regularly.

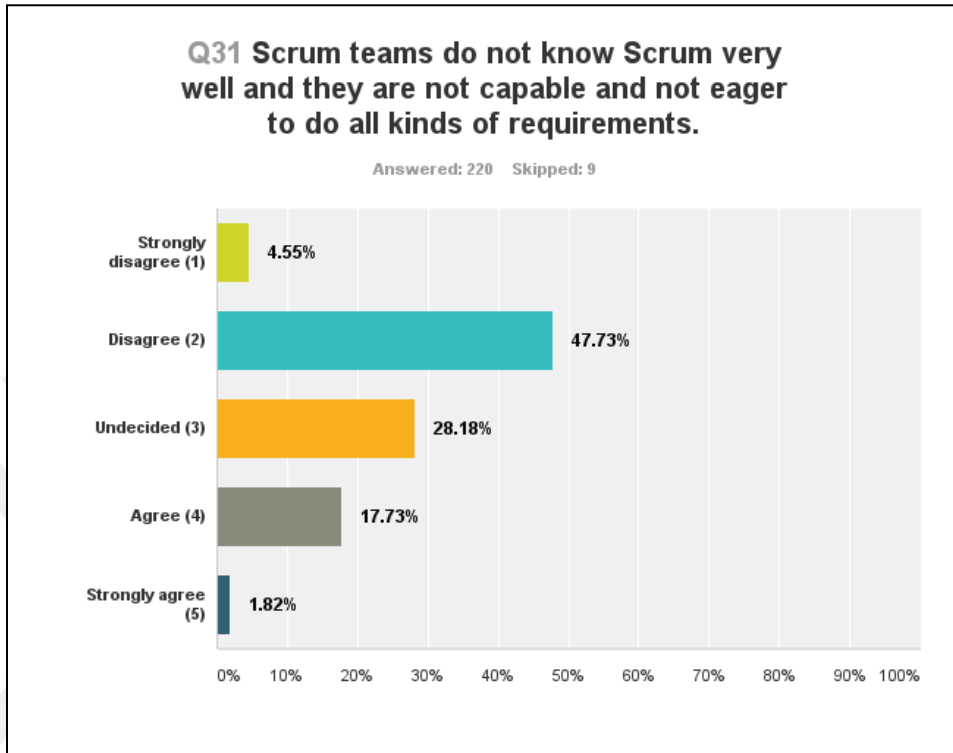
**Figure 5.31 : Responses to Q27**



### **Analyzing the responses to Q27**

Most of the respondents who have answered the Q27 are undecided with 41.82 percent about whether Scrum teams do know Scrum well and do anything for the projects' success. 34.09 percent of them agree and 1.36 percent of them strongly agree with this idea. On the other hand, 20.00 percent of them disagree and 2.73 percent of them strongly disagree with this idea (Figure 5.31). Besides, 42.11 percent of Product Owners are undecided, 36.84 percent of them agree, 15.79 percent of them disagree, 5.26 percent of them strongly disagree. Further, 37.50 percent of Scrum Masters disagree, 33.33 percent of them are undecided, 25 percent of them agree and 4.17 of them strongly agree. Additionally, 47.83 percent of Business Units are undecided, 39.13 percent of them agree, 10.14 percent disagree and 2.90 percent of them strongly agree. All in all, we can say that, most participants suspect whether Scrum Teams give full effort to the projects or not.

**Figure 5.32 : Responses to Q31**



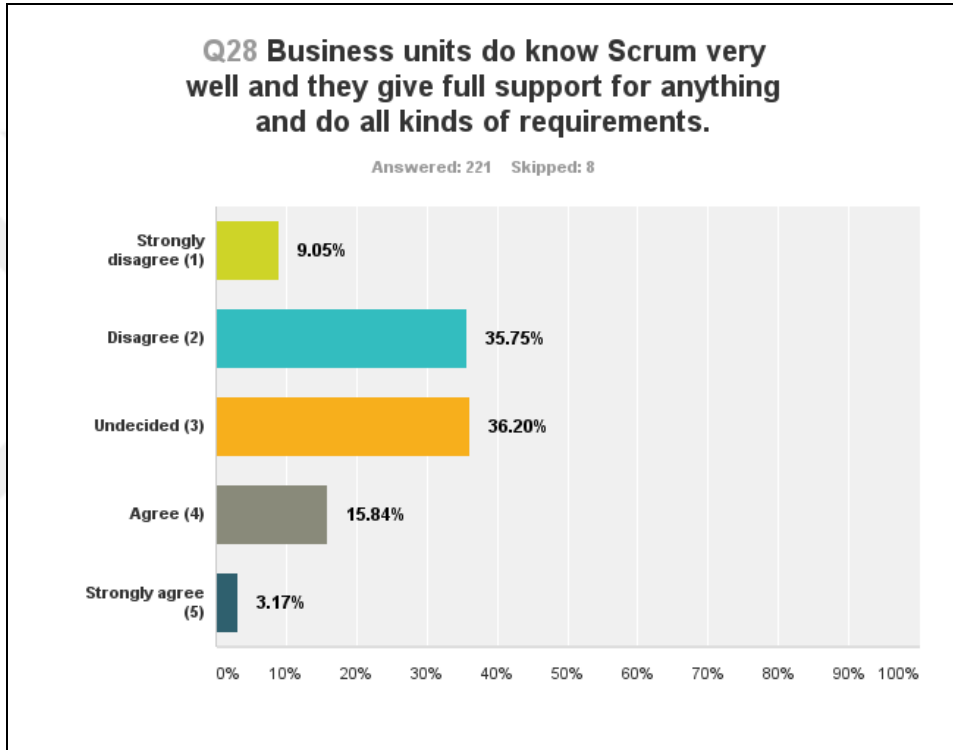
### **Analyzing the responses to Q31**

On the other hand, we ask the participants a reverse question in which Scrum teams do not know Scrum well and they are not capable of and not eager to do all kinds of requirements. 47.73 percent of them disagree and 4.55 percent of them strongly disagree with this idea. Furthermore, 28.18 percent of them are undecided and 17.73 percent of them believe and 1.82 percent strongly believes this idea (Figure 5.32).

When we evaluate the results of Business Units, 38.57 percent of them disagree, 4.29 percent of them strongly disagree, 37.14 percent of them are undecided, 18.57 percent of them agree, 1.43 percent of them strongly agree. According to Scrum Master Responses, 37.50 percent of them disagree, 8.33 percent of them strongly disagree, 16.67 percent of them are undecided, 33.33 percent of them agree and 4.17 of them strongly agree. On the other hand, when it comes to the results of Product Owners, 42.11 percent of them are undecided, 36.84 percent of them disagree, 15.79 percent of them agree and 5.26 percent of

them strongly agree that Scrum teams don't know Scrum very well and they are not capable and not eager to do all kinds of requirements. In summary, most of the respondents believe that Scrum Teams know Scrum but they have suspects about the Scrum Team's knowledge about Scrum and their effort for projects' success.

**Figure 5.33 : Responses to Q28**



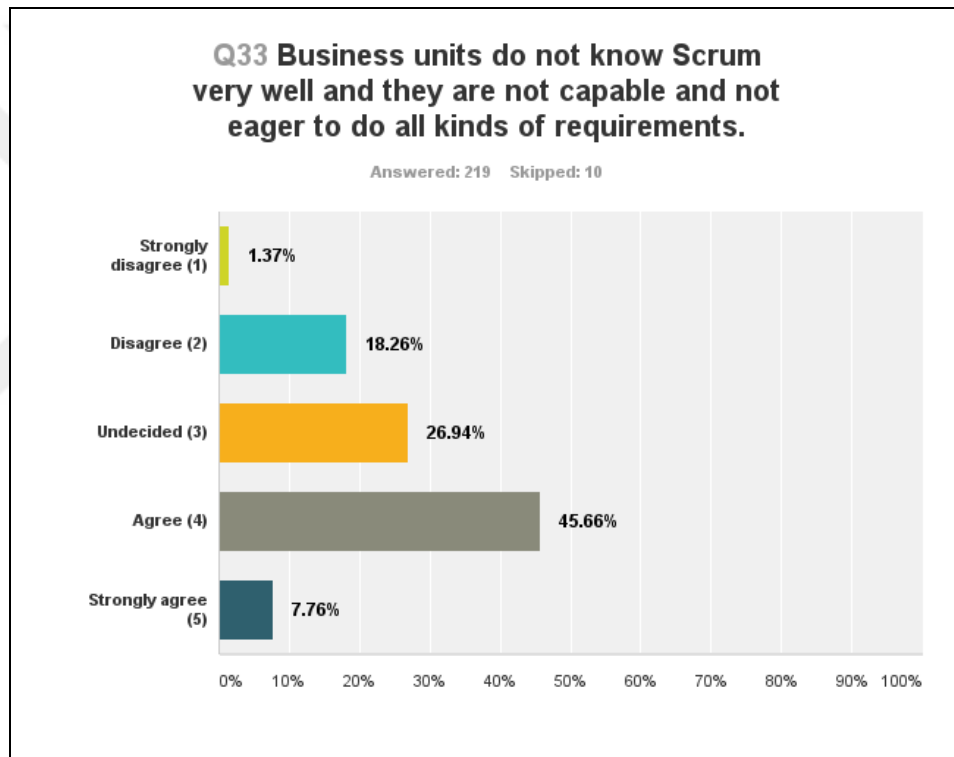
### Analyzing the responses to Q28

According to the responses of the Q28, there are 36.20 percent who are undecided and 35.75 percent disagree and 9.05 percent strongly disagree with the statement that business units do know Scrum well and they give full support for the projects to be successful. On the other hand, 15.84 percent of them agree and 3.17 percent of them strongly agree with this idea (Figure 5.33).

According to the responses of Product Owners, 47.37 percent of them disagree, 15.79 percent of them strongly disagree, 21.05 percent of them are undecided, 15.79 percent of

them agree. Responses of Scrum Masters show that 41.67 percent of them disagree, 8.33 percent of them strongly disagree, 29.17 percent of them are undecided, 12.50 percent of them agree and 8.33 percent of them strongly agree. On the other hand, 47 percent of business units are undecided, 22.86 of them agree, 7.14 of them strongly agree, 17.14 percent of them disagree and 5.71 percent of them strongly disagree that Business Units know Scrum well and give full support to all.

**Figure 5.34 : Responses to Q33**



### Analyzing the responses to Q33

According to the answers of the Q33, most of the people with 45.66 percent believe and 7.76 percent strongly believe that business units do not know Scrum well and give support to the Scrum. Moreover, 26.94 percent of the participants are undecided. There are 18.26 percent of them disagreeing and 1.37 percent of them strongly disagreeing with this idea (Figure 5.34). All in all, we can say that most of the respondents believe that business units



do not know Scrum well and also they are not capable of and not eager to do all kinds of the requirements which projects need.

## **Open Questions**

### **Q35: What are the advantages of Scrum?**

There are 97 respondents out of 225 answered to the question. These are the combination and the summary of the results:

- a. Scrum enables transparency in which we can see who is working with the tasks, and see how much work any member does in Scrum team. Moreover, it is easy to follow up the status of the tasks of the projects such that Daily Scrums are very beneficial because it increases self-motivation of the team members. We can see the big picture in projects and decide whether the project will finish on time and we will have the opportunity to take precautions for unfinished projects such that extra resource can be added to finish on time.
- b. Scrum enables easy adaptation and more flexibility to the sudden changes in requirements.
- c. Impediments in the projects can be solved more easily and rapidly with the help of Scrum Masters.
- d. If the product backlog is well-prepared, the work done is more efficient and effective.
- e. Scrum is faster; it decreases completion time of the projects and increases efficiency.
- f. It is very advantageous to have shippable products at the end of the sprints.
- g. The team members are more productive and effective with Scrum.
- h. Task distribution and planning are more effective.
- i. Documentation is reduced with Scrum. We give more importance on increments rather than documentation.
- k. The team members have single focus on the goal of the project.

l. Scrum enables partially deliverable products in which the business units can see whether the partial product meet the requirements and see the errors and thus the Scrum team can fix them before the project reaches the production environment.

m. The Scrum team members are gradually improving themselves in both technical area and business area. Furthermore, their motivation, interaction, collaboration, cooperation and self-organization increase with Scrum because it decreases bureaucracy and the time of management of projects.

n. We can give priorities to the high valued projects and contribute to profitability of our organization.

o. Scrum increases customer satisfaction because business units can see how project is progressing, how much work remains, and how much work is done. Before Scrum, we could not follow up the projects' status clearly. There was uncertainty in the projects. Finally, Scrum has eliminated this uncertainty.

p. Scrum reduces time pressure of the projects.

q. It is more open to the innovation.

### **Q36: What are the disadvantages of Scrum?**

There are 87 people who have responded to this question. These are combination and summary of the results:

i. Scrum is not compatible with COBIT, ITIL processes in banking applications although Scrum is adopted and is implemented by the rules of the Agile Manifesto.

ii. There are some open issues that Scrum does not cover. Thus, most of the people behave differently and reach different points when they need to decide to solve the open issues.

iii. Continuous requests and new demands from business units.

- iv. Scrum can result to misuse in planning, task management, and duration because of self-organization of Scrum teams. Moreover, Scrum Teams can get less commitment since they think that the PBIs will not be done on their sprints. Thus, the teams can work more slowly.
- v. Scrum is hard to implement and adopt in the organizations which resists the change.
- vi. Some teams should not adopt Scrum because with Scrum, there are some priorities after the sprint begins and any other task cannot be taken into account in the sprint although it is so urgent.
- vii. There is less analysis and less documentation in Scrum which can cause some errors and problems in the product which might not meet the requirements. Documentation and analysis is very important in banking applications and the Scrum teams give less importance to them.
- viii. Impact analysis is not sufficient and it may cause serious problems in production environment.
- ix. There is a risk in determining the requirements on time. Thus, the products cannot be efficient and effective. So, these products can be revised again.
- x. Because there is no analysis document in the start of the project, the scope may not be determined by the team and it causes the problem of not seeing the whole project as a one in the large scale projects.
- xi. The Scrum teams can change while the projects are being completed. It causes the problem of determining and measuring the performance of the team members.
- xii. Because of the abundance of meetings, there may be loss of motivation in the Scrum team members. It may result in Scrumbut which are the results of modifying Scrum. For example, "(We use Scrum, but) (Retrospectives are a waste of time,) (so we don't do them.)". (Scrumbut, Scrum.org)
- xiii. Business units have difficulties in adaptation to Scrum.

xiv. General project management does not exist with Scrum.

xv. If nonflexible deadlines exist, sprints are not sprints, they are kind of mini Waterfalls.

xvi. In Scrum, there are no tester and analyst position; every team member can do the same task in projects. However, testers and analysts cannot write programmes and thus developers in the Scrum Team can work at development, analysis, and testing stages of the projects. So, the developers can be overloaded and it can cause a decrease on their performance because of this overload.

xvii. Because of the Scrum Events, there are requirements of meeting rooms, smart phones and laptops to make these meetings more productive.

xviii. Business units do not decide on the product because they think that Scrum Teams can handle the sudden changes on the requirements. In every Sprint reviews, they change their demands which cause many reworks that can result in loss of motivation in Scrum Teams. Finally, there is a product which does not meet the all the requirements. For this reason, the product backlog should be well-prepared before the project starts.

xix. Scrum is hard to implement in large-scale projects and in large teams.

xx. Scrum causes delay in fixing urgent bugs and maintenance issues. These problems may not be fixed after the Sprint starts because the PBIs are determined in the start of the sprint. There should be some buffer time for fixing these maintenance issues.

xxi. Determining the inadequate PO and SM can result in less performance of the Scrum team in the projects. Product backlog should be well-prepared and up-to-date because it has huge impact on the projects. For this reason, POs should be well-trained and well-chosen to solve this performance issues.

xxii. If there is a requirement that a team member should work in different Scrum Teams, he should work in both teams. It is also a Scrumbut. Management should prevent this from happening.

xxiii. If PO is from Business Units, it adds extra responsibilities to this person. He or she must do his or her responsibilities on time with PO responsibilities.

xxiv. Scrum is also hard for the new recruits; they need some adaptation period and need to take the Scrum Training.

xxv. Project planning and management are hard with Scrum. Projects are completed incrementally and thus the budget and the finishing time cannot be determined in the start of the project.

xxvi. Getting rid of the management role can lead to team members not being evaluated well enough. Moreover, it results in an imbalance among the members where some members work hard and some others do not. Motivation of the team can decrease.

xxvii. Because there is no requirement analysis in the start of the project, there may be reworked data modelling process and rewritten programming codes.

xxviii. The outputs of the sprints have many errors. This should be fixed.

xxix. There are some problems in working with Scrum of Scrum Teams because their priorities differ from each other.

xxx. There is an analysis in the start of the sprint and there are tests at the end of the sprint. Development phase should be in the middle of the sprint. For this reason, developers can be more stressful because they need to finish the tasks on time to be tested before sprint finishes.

### **5.3 INTERVIEWS**

To understand how Scrum changes our working habits and what the advantages of Scrum are, we ask five questions to different people such that one person is the PO from Information Technology (IT), one person is also PO but in Business Units, one person is a member of a Development Team. The interview questions (IQs) are:

**IQ1:** What changes does Scrum contribute into your business life and your job description?

**IQ2:** What are the contributions of Scrum which are related to the duration, the budget and the scope of your projects?

**IQ3:** What are the positive results of Scrum to the products of projects and their functionality?

**IQ4:** What are the contributions of Scrum with regard to your relations with the stakeholders?

**IQ5:** Is there anything that you want to add?

Our purpose to choose the people from different aspects is to take different comments, perspectives and ideas of the people from different departments. Some of them can be not satisfied with Scrum although some of them can be happy with working with Scrum.

### **5.3.1 Interview With PO From IT:**

He has so much experience in Banking, System Analysis and Project Management. He has the certificates of Project Management Professional (PMP), ITIL and The Professional Scrum Master 1 (PSM I). Now he is a Product Owner in a bank which has adopted Scrum as a software development methodology in Turkey.

#### **His response to IQ1**

I have new responsibilities about product rather than my administrative functions in my job description. My new responsibilities are to contribute new features to my responsible products, while adding those features selecting most profitable works, and to follow the future vision and direction of the products in my responsibility.

#### **His response to IQ2**

Scrum has meaningful contributions about duration of projects such that we can deliver the products incrementally and it enables transparency between stakeholders and IT and also

makes risk management easier. So that, it leads to shorter time in projects. Another factor can be that Scrum reduces the rework costs. Thus, we can witness positive reflections on the duration of projects.

Moreover, it has positive influences on the budgets of projects, too. We can see the inaccurate implementations and incorrect works not at the end of the project but at the end of the sprints and thus, we can fix these errors easily and can reduce the rework costs and personal costs.

Furthermore, business units can choose more profitable works/products for the Scrum Teams thanks to that they can reorganize the product backlog and prioritize the PBIs. Thus, it has positive earnings to our organization.

### **His response to IQ3**

Because the prioritized items are done smoothly, necessary work is really done, lower priority jobs are done after the more prioritized ones. So, it leads to maximum profit from our development work.

### **His response to IQ4**

By the help of collaborative working with the stakeholders and business units, Scrum Teams can realize more about their precisions, urgencies and problems. We learn to speak the same language and understand each other better. As a result, IT and business units have strong collaboration and cooperation between each other.

### **His response to IQ5**

I have no comments for now.

## **5.3.2 Interview With PO From Business Unit**

She has about 9 years' experience on banking, retail credits and payment systems. She has the certificate of PSM 1. She has been working as a product owner for 6 months in a bank which has adopted Scrum as a software development framework in Turkey.

### **Her response to IQ1**

By using traditional software development methodologies, it took very long time to analyze and design processes in order to realize and to predict all the requirements of the project. In these methodologies, business unit and IT has less communication in this process. For this reason, the products of the development in such methods do not meet the requirements of the customer and also after that we could realize the required changes in the process of development and could not change the requirements and not enlarge the scope of the project. The market is not stable, customer requirements can differ in any time and new requirements can emerge. For these reasons, there is a need for flexible project model. With Scrum, we as a Scrum Team, have the purpose of determining the significant requirements, prioritize and develop them. Our first aim is to maximize customer satisfaction and customer expectations. In regular events Scrum provides, we review our profitability to increase it and can do the required improvements.

### **Her response to IQ2**

Before Scrum, we had many difficulties in the concept of product development between IT and Business Units such as managing the process in the different services, trying to unite all these services in a shared goal and having hard times in getting the desired product on the due dates.

Scrum has efficient contributions to these problems and it adds positive meaning to the team work. With this new model, every member has the same goal, they are all aware of not only their responsibilities but also their teammates'. They make decisions collaboratively on the shared goal and start to deliver the increments on time.

Moreover, after adopting Scrum, we have rapid shippable products and increments in our projects. Thus, we can show these increments to our customers and continue to other requests and can check them or can make them more effective.



### **Her response to IQ3**

With Scrum, we have not only successful projects but also by handling the changing requirements of customers faster we reduce the risks in our projects. Thus, we can present more qualified products to our customers.

### **Her response to IQ4**

With Scrum, we communicate with stakeholders better, IT and business units can understand each other and have single focus on the products.

### **Her response to IQ5:**

I don't have anything to add. Thanks.

### **5.3.3 Interview With A Development Team Member**

She has approximately 2 years of experience in business analysis. Now, she has the certificate of PSM 1. She currently works as a member of a Scrum Team.

### **Her response to IQ1**

After we adopted Scrum as a software development framework, my job description has changed as a member of development team. I used to work as an analyst but now I am working as both an analyst and a tester on our planned PBIs.

### **Her response to IQ2**

Rather than the stages of analysis, development and testing; incrementally delivering our products reduces the duration of the projects and handling the sudden changes on the business requirements is beneficial to make projects more up-to-date.

### **Her response to IQ3**

With the help of the Scrum framework, I think that getting feedbacks at the end of the sprints and sprint reviews brings more qualified products.

**Her response to IQ4**

It is required in Scrum to be in communication with all the stakeholders. In sprint reviews, the incremental developments are presented for their evaluations. Getting feedbacks from them are advantageous for better communication and collaboration.

**Her response to IQ5**

Scrum creates the consciousness and perception of being a team which leads to have reduced project time and increased motivation of all members in a Scrum Team. Moreover, Scrum has just a developer role in which every member in a development team has similar features. Hence, every member can improve himself. If it does not exist, development, analysis and testing will be done like in Waterfall methodologies and nobody can improve himself.

## 6. CONCLUSION & SUGGESTIONS

We are living in a very competitive business world and creating new values. To achieve this goal, we have been using traditional software development strategies to create new products. However, traditional software development strategies have been suffering from the problems such as they cannot respond to the sudden changes in customer requirements, the projects are failing because of overtime and over budget. For these reasons, recently, a new framework has come to this technology world: Scrum. Scrum claims that it can solve the problems of late response to the sudden changes and by using Scrum, projects will be successful and IT professionals will be happier. Some people believe that Scrum is useful and can be adopted by any organization in any scale and some of them do not. Our aim is to find out the usefulness of Scrum in banking applications in Turkey because banks are very competitive in the market and need to be the pioneers of their products.

To start with the supporters of Scrum, they believe that Scrum can solve the problems of transparency, team communication, sudden changes in requirements, and project success. For example, the Yahoo Company had some experiments in more than 150 Scrum Teams and results show that Scrum improved thirty day productivity, it helps collaboration and cooperation, and it improves business value and overall quality (Benfield, 2007). Intel also had similar results. By using Scrum, job satisfaction and higher morale can be achieved; Scrum helps projects finish in a shorter period of time (Danube, 2008). Furthermore, according to the case studies in 5 different companies, even requirements are not stable, the projects can progress, everything is visible to everyone (transparency of increments), and the teams are self-organizing to finish their tasks. In a pilot application in a bank in Turkey, 85 percent of participants believe that overall Scrum is better (Kır, 2014). According to Sutherland, even in problematic teams, Scrum makes an improvement with about 35 percent more efficiency.

On the other hand, there are many people who believe that Scrum is hard to implement, Scrum is evil and Scrum will die. According to Ardant (2008), Scrum is evil because it is iterative Waterfall, it values process over people, and when it fails, all the people involved

think that all of Agile is no good. Furthermore, because software development is a complex problem, Scrum is hard to implement. It needs our attention and intelligence to solve these complex software problems (Moe and Dingsøy, 2008).

When we think about the situation of Scrum in Turkey, results are promising but not best. When we consider one of the pilot applications in Turkey, Overall Scrum is better (Kır, 2014). Also, many organizations like Scrum Turkey and Agile Turkey try to adapt the Scrum in Turkey and want to use it in software development. Many new Scrum jobs are gradually emerging.

On the other hand, we have a questionnaire and some interviews in order to learn whether Scrum is useful or not. The questionnaire has been implemented in the 5 companies who are using / have used Scrum as a software development methodology. There are 225 people who respond our questionnaire. The results show that:

- i. 70 percent of the respondents took Scrum training.
- ii. 68 percent of them believe that Scrum is better in responding sudden changes in customer requirements.
- iii. Most of the respondents, it is about 70 percent, do not believe that Scrum is compatible with COBIT, ITIL processes in banking applications.
- iv. 55.55 percent of them believe that Scrum helps reduce the amount of time for the software projects.
- v. Approximately 80 percent of them believe that Scrum enables transparency.  
Everybody can see what is done, what will be done and what the impediments are.
- vi. Most of the people do not believe that Scrum teams are self-motivated and self-organized.
- vii. About 63 percent of them believe that management supports Scrum.

- viii. 54.51 percent of them agree that Teams are more motivated and have more performance with Scrum.
- ix. 71.95 percent of them believe that Scrum increases assistance, cooperation and collaboration of teams.
- x. Approximately 57 percent of the participants agree that Scrum makes projects more qualified and more appropriate for their goal.
- xi. Most of the people, 54 percent of the participants, are undecided that Scrum teams do know Scrum very well and give support for anything.
- xii. About 52 percent of them believe that continuing with Scrum is always useful but some participants have some doubts that continuing with Scrum will be always useful.

Moreover, according to the interviews,

- xiii. Scrum can tolerate sudden changes in requirements

xiv. Scrum makes transparent working environment in which everybody can see the progress of projects.

All in all, Scrum seems to be useful in banking applications in Turkey. However, it has some problems, too such as Sprint Backlog and Product Backlog are not prepared appropriately and they are not up-to-date, Scrum knowledge of Scrum Teams and business units are not well enough, Scrum is not appropriate for COBIT, ITIL processes. When these problems can be solved, Scrum will be more useful in Turkey. Also, there are limitations for finding a participant who works with Scrum in banking industry in Turkey. There are just 5 banks who are trying to adopt Scrum as a software development methodology. Scrum is still very new in Turkey. In the future, adopting Scrum percentage will also increase.

Moreover, I have some recommendations in order to have better software development environment with Scrum. These are:

- i. Change is hard, but people should change their working habits according to Scrum and there will be no resistant to change in Scrum environment.
- ii. Product backlog management is very important. Product backlog should be well-prepared for further implementations and POs must be very careful while preparing PBL and adding new items into the PBL.
- iii. If there are deadlines for software projects, there will be no Scrum, it will be mini waterfalls. Also, deadline pressure can lead to reduce motivation of the Scrum teams. For these reasons, there should not be any deadlines for Scrum projects.
- iv. Some Scrum Events like Retrospective and Planning are accepted as time consuming and boring meetings. For this reason, Scrum events should be made enjoyable by the participants of Scrum team in order to increase team motivation.
- v. Scrum enables self-organizing teams but it can cause some problems such that the team can work less and take less product backlog items into the sprint. For these reasons, there should be a control mechanism in order to inspect and detect these behaviors and take precautions to these situations.
- vi. Auditing standards of banking industry in Turkey are COBIT and ITIL standards. However, Scrum is defined by the Agile Manifesto. For this reason, there should be some authority to revise the COBIT and ITIL standards according to Agile Manifesto. As a result, there will not be any conflict between these standards and be better working environment.

## 7. REFERENCES

### *Books*

Davis A. M., 1995. *201 Principles of Software Development*. New York: McGraw-Hill

DeGrace, P. and L.H.Stahl, 1990. *Wicked Problems, Righteous Solutions*. Prentice Hall: Yourdon Press

McConnell S. 2004. *Code Complete: A practical Handbook of Software Construction*, 2nd edition: Microsoft Press.

Moe N. M., Dingsøy T., 2008, Agile Processes in Software Engineering and Extreme Programming. *Scrum and Team Effectiveness: Theory and Practice* Volume 9: Limerick, Ireland: Springer, pp. 11-20

Schwaber K., Sutherland J. 2013. *Scrum Guide, Definitive Guide to Scrum: The Rules of the Game*: Scrum.org

ScrumStudy, 2013. *A guide to the Scrum Body of Knowledge*. 2013 Edition. :VMedu, Inc.

Sommerille I., 2010. *Software Engineering*. 9th Edition. United States: Pearson.

### *Other Sources*

Akdag A, Kombak T., Yılmaz T. (2014) "Türkiye Finans Goes for Higher Benefits from Change" Scrum.org Whitepapers.

Ardant E.L., 2008, *Scrum is evil*.<http://ericlefevre.net/wordpress/2008/10/07/scrum-is-evil/> [accessed 5 April 2015]

Beck K., Beedle M. Bennekum A.V., Cockburn A., Cunningham W., Fowler M., Grenning J., Highsmith J., Hunt A., 2001. *The principles of Agile Manifesto* <http://agilemanifesto.org/principles.html> [accessed 24 March 2015]

Benefield G., 2008, *Rolling out Agile in a Large Enterprise*, <http://www.controlchaos.com/storage/scrum-articles/YahooAgileRollout.pdf> [accessed 5 April 2015]

Bolat S., 2014, *Yüksek risk ve belirsizlik altında SCRUM*. Yönetim Dergisi, pp. 64-69

Cho, L (2009). *Adopting an Agile Culture a User Experience Team's Journey*. Agile Conference: 416. "Scrum of Scrums". Agile Alliance. <http://guide.agilealliance.org/guide/scrumofscrums.html>

Elwer P., (2008), *Agile Project Development at Intel: A Scrum Odyssey* <http://scrumtrainingseries.com/Intel-case-study.pdf> [accessed 5 April 2015]

Hughey D. 2009. *Comparing Traditional Systems Analysis and Design with Agile Methodologies*. University of Missouri – St. Louis.

Kır G. 2014, *Scrum Adoption in KuveytTürk IT* masters thesis.

Lotz, M. 2013, *Waterfall versus Agile Development Methodology* [online], <http://www.seguetech.com/blog/2013/07/05/waterfall-vs-agile-right-development-methodology> [accessed 24 March 2015]



Parnas, David L.; Clements, Paul C. 1986. *A rational design process: How and why to fake it*. *Software Engineering, IEEE Transactions: pp. 251–257.*

Patrick. 2006. *Starfish retrospective*.

<https://www.thekua.com/rant/2006/03/the-retrospective-starfish/> [accessed 24 March 2015]

Rising L., Janoff N.S., 2000. *The Scrum Software Development Process for Small Teams*.

<http://web.lindarising.info/uploads/IEEEScrum.pdf>

Scrumbut. 2015. <https://www.scrum.org/ScrumBut> [accessed 5 April 2015]

Software. 2015. <http://en.wikipedia.org/wiki/Software#History> [accessed 24 March 2015]

Software development. 2015.

[http://en.wikipedia.org/wiki/Scrum\\_%28software\\_development%29](http://en.wikipedia.org/wiki/Scrum_%28software_development%29) [accessed 24 March 2015]

Software development process.2015.

[http://en.wikipedia.org/wiki/Software\\_development\\_process](http://en.wikipedia.org/wiki/Software_development_process) [accessed 24 March 2015]

Sonmez J., 2010. *Scrum will die*

<http://simpleprogrammer.com/2010/02/23/scrum-will-die/> [accessed 5 April 2015]

Sutherland J., 2001,"Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies" Volume 14, No.12, Cutter IT Journal.

Waterfall Model. 2015. [http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model) [accessed 24 March 2015]

The history of Scrum. 2015.<http://www.scrumguides.org/history.html> [accessed 24 March 2015]