Research Paper

BMIJ

ISSN: 2148-2586

BUSINESS & MANAGEMENT STUDIES:
AN INTERNATIONAL JOURNAL
Vol.:7 Issue:4 Year:2019, pp. 1776-1805

Citation: Köse, B.Ö. (2019), Recent Agile Requirement Engineering Practices In It Projects: A Case Analysis, BMIJ, (2019), 7(4): 1776-1805 doi: http://dx.doi.org/10.15295/bmij.v7i4.1214

RECENT AGILE REQUIREMENT ENGINEERING PRACTICES IN IT PROJECTS: A CASE ANALYSIS

Büşra Özdenizci KÖSE[1]

Received Date (Başvuru Tarihi): 11/08/2019
Accepted Date (Kabul Tarihi): 11/09/2019
Published Date (Yayın Tarihi): 25/09/2019

ABSTRACT

Today, the implementation of high quality and efficient Requirement Engineering (RE) practices in agile software development projects, is gaining great importance. Practitioners and researchers seeks for lighter RE practices that can handle the issues of abstract, unclear and changing requirements, and at the same time that can satisfy the Agile Manifesto philosophy. This study examines importance of RE practices in agile software development projects, and explores which aspects of the RE practices are perceived as most critical and how such aspects are adapted in practice today through two different agile software development projects of a case organization. This study aims to contribute agile RE literature by providing an interpretive analysis on perception of agile RE practices from different perspectives (agile team members, product owners, some top executives). Within this context, this study draws lessons from case studies and presents beneficial agile RE guidelines for practitioners and researchers.

Keywords: *Requirements Engineering, Business Analysis, Agile Methods, Agile RE, Scrum*

JEL Codes: *M15, O22, O32*

BT PROJELERİNDE GÜNCEL ÇEVİK GEREKSİNİM MÜHENDİSLİĞİ UYGULAMALARI: BİR VAKA İNCELEMESİ

ÖZ

Günümüzde, çevik yazılım geliştirme projelerinde, yüksek kaliteli ve verimli Gereksinim Mühendisliği (Requirement Engineering, RE) uygulamalarının gerçekleştirilmesi büyük önem kazanmaktadır. Uygulayıcılar ve araştırmacılar, soyut, belirsiz ve değişen gereksinimlerle ilgili sorunları ele alabilecek ve aynı zamanda Çevik Manifesto felsefesini sağlayabilecek daha hafif RE uygulamaları aramaktadır. Bu çalışma, çevik yazılım geliştirme projelerinde RE uygulamalarının önemini, RE uygulamalarının hangi yönlerinin en kritik olarak algılandığını ve bu durumların bir vaka organizasyonunun iki farklı çevik yazılım geliştirme projesi ile uygulamada nasıl gerçekleştirildiğini araştırmaktadır. Bu çalışma, farklı bakış açılarından (takım ekipleri, ürün sahipleri, bazı aşamalarda üst düzey yöneticiler) çevik RE uygulamalarının algısına ilişkin yorumlayıcı bir analiz sunarak çevik RE literatürüne katkı sağlamayı amaçlamaktadır. Bu kapsamda, bu çalışma vaka çalışmalarından ders çıkarmaktadır; uygulayıcılar ve araştırmacılar için yararlı çevik RE yol haritaları sunmaktadır.

Anahtar Kelimeler: *Gereksinim Mühendisliği, İş Analizi, Çevik Metotlar, Çevik RE, Scrum*

JEL Kodları: *M15, O22, O32*

[1] Asst. Prof., Gebze Technical University, busraozdenizci@gtu.edu.tr https://orcid.org/0000-0002-8414-5252

1. INTRODUCTION

In recent years, the number of available Information Systems Development Methodologies (ISDMs) has been significantly increased with the aim of reducing time to market beyond cost savings, developing high quality system, accessing large multi-skilled workforces and achieving project success (Yeo & Hahn, 2014; Vallon et. al, 2018; Joslin, & Müller, 2016: 364). Software or IS development process is a sophisticated activity in general. Traditional system development methodologies are evolving into more lean and efficient, but also more agile and dynamic software development methodologies.

The CHAOS reports released by Standish Group provides useful insights about current status of IS development over the world (Standish Group CHAOS Report, 2015). Standish Group classifies projects into success (i.e., which are delivered on time, on budget and will all features; meets all the requirements of the project), challenged (i.e., which was eventually delivered but either over budget, not on time or not fully completed) and failure (i.e., nothing was delivered) across organizations workforces (Clancy, 2014). According to CHAOS Report workforces (Standish Group CHAOS Report, 2015: 2), approximately 29% of the projects were completed successfully on time and within budget with all the promised functionality; and approximately 52% of the projects were over cost, over time and/or lacking promised functionality; and the rest of the projects are abandoned or cancelled which means failed projects. Standish Group also identified that the executive management support, user involvement during software development lifecycle as well as clear statement of customer and business "requirements" have critical impact on the project success (Standish Group CHAOS Report, 2015: 11; Wojewoda, 2015).

PMI (Project Management Institute) defines requirement as "*the condition or capability that is required to be present in a product, a service or a result to satisfy a contract or other formally imposed specification*" (PMI, 2008: 445). In traditional software development methodologies, the initial step is to implement a detailed set of requirements elicitation and documentation as the milestone, and then system design and development steps can be followed. However, over time advancing technology and industries and accordingly changing customer and business needs made impossible to define, analyze, document and validate all requirements clearly at the beginning of all software development projects, which is also known as Requirements Engineering (RE) (De Lucia & Qusef, 2010: 214; Paetsch, et al. 2003). In the beginning of 1990s, some practitioners and researchers recognized that traditional software development methodologies including heavyweight RE and documentation driven RE activities

were somewhat frustrating and not easy to handle for all project types (De Lucia & Qusef, 2010:214; Javanmard & Alian: 2015: 1386). By the way practitioners and researchers started to seek for more efficient and lightweight techniques (Serrador & Pinto, 2015:1041).

Another key finding of Standish Group is that the methodology used during project execution is a significant issue (Standish Group CHAOS Report, 2015; Wojewoda, 2015). Over the past decade, the project size was considered as a critical predictor of project success (Awad, 2005). But today, rather than the project size, agile projects (i.e., software development projects that are realized by agile engineering practices and agile managerial methods) are statistically twice more likely to succeed than projects which are realized by waterfall methodology as a traditional software development methodology (Standish Group CHAOS Report, 2015; Wojewoda, 2015; Mersino 2018).

Currently, as a reaction to traditional software development methodologies, agile approaches have become popular that aims to expedite the software development lifecycle and to ensure that output satisfies user requirements. Rigid structure of traditional approaches were transformed into more flexible and agile structure to adapt nature of software project development process to today's dynamic environment and continuously changing customer and business requirements; since agile software development focuses on less initial planning and supports an evolutionary process that includes a number of iterative development approaches over time for project implementation (Serrador & Pinto, 2015:1041; Dybå, & Dingsøyr, 2008; De Lucia & Qusef, 2010: 213-214; Flora, & Chande, 2014: 3626). Agile Manifesto emphasizes "*valuing individuals and interaction over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to changes over following a plan*" (Beck et al., 2001; Schön et al. 2017: 80). Agile methods allow changing business and user requirements, to deliver working software frequently and to achieve close collaboration of all stakeholders and project team.

However, on the other side, integrating traditional RE processes with agile software development processes generally does not work well; traditional business analysis techniques implementation on agile methodologies can seem overwhelming (Käpyaho & Kauppinen, 2015: 335-336; Parker, 2013). Today, practitioners and researchers still seeks for lighter requirement practices that can handle the issues of abstract or unclear requirements and specifications, changing requirements, and at the same time that can satisfy the Agile Manifesto philosophy; fast delivery of software, high user involvement and executive support etc. during

software project implementation. More collaboration and less engineering practices are accepted as the key characteristics of agile methodologies.

Due to increasing reliance on agile methodologies, agile analysis and RE concept is emerged. Unlike traditional software analysis, Ambler (2018) states that "*Agile analysis is highly evolutionary and collaborative process where developers and project stakeholders actively work together on a just-in-time (JIT) basis to understand the domain, to identify what needs to be built, to estimate that functionality, to prioritize the functionality, and in the process optionally producing artifacts that are just barely good enough.*"

Currently, enabling more agile business analysis and agile RE practices is an interesting research field. Practitioners and researchers are working on how RE approaches and techniques can be considered within agile software methodologies; accordingly, there are also some studies which provides useful information about implementation of agile RE practices, lighter requirement practices and its challenging issues (Ochodek, & Kopczyńska, 2018; Wagner et al., 2018; Kasauli et al., 2017; Schön et al., 2017; Musa et al., 2017; Käpyaho & Kauppinen, 2015; Cao et al., 2008).

This study aims to examine the importance of RE practices in agile software development projects and to explore which aspects of the RE practices are perceived as most critical and how such aspects are adapted in practice through a case organization. To do this, two different agile software development projects with different outcomes (completed on time vs. completed with delay) are observed in-depth and studied which are conducted by a large-scale IT company in Turkey for a year. This study fills the gap in agile RE literature by providing a comprehensive analysis and perception of agile RE practices from different perspectives (agile team members, product owners, some top executives) through real world projects. The paper draws lessons from case studies and provides with a set of agile RE guidelines for practitioners and researchers rather than performing hypotheses testing.

The structure of the paper is as follows: Section 2 sheds light on the benefits and limitations of popular agile software development methodologies, the importance of RE in software development and presents relevant research on agile RE practices. Section 3 explains the research design and methodology including brief information about the case organization and profile of investigated projects. In Section 4, significant findings with interview notes and key challenges from different perspectives are presented. Finally, Section 5 provides

suggestions and guidelines for agile RE practices, discusses the limitations of research, and concludes with future research directions.

2. RESEARCH BACKGROUND

The term *agile* originated within the software development field; but now with its success, agile concept is being used by non-software related projects as well. Today, common examples of agile approaches include Scrum, Extreme Programming (XP), Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), Adaptive Software Development and Crystal (Javanmard & Alian, 2015; Phil, 2015). All approaches have different advantages and disadvantages as reviewed in Table 1 in accordance with the studies of Awad (2005), Phil (2015), Flora et al. (2014) and Javanmard et al. (2015).

Collabnet Versione (2019) indicates that Scrum as an iterative and incremental agile software development framework for managing product development is widely adopted agile method. In the Scrum methodology, project is performed in a series of iterations, namely sprints, which generally last from 2 to 4 weeks; at the end of each sprint, a high quality working software needs to be produced instead of detailed RE and business analysis activities (Sutherland, & Schwaber, 2016:8; Stellman & Greene, 2014:43; Rubin, 2012:13-28).

Another new trend which is highlighted by Collabnet Versione (2019) is hybrid methodologies; perhaps the vision behind is that following only one approach is not a perfect solution. According to Collabnet Versione (2019), integration of Scrum and XP methodologies, called Scrum/XP Hybrid methodologies (~64%) continue to be the most common agile methodologies used by organizations today. Recently, combination of two popular -Scrum and XP- approaches is preferred by practitioners to realize maximum success and benefits in projects. In case of Extreme (XP) Programming, everyone involves in all phases of project work; there is no need for specialization of role. Challenging XP approach practices are "pair programming" which means two developers works while sitting on one computer for building the code (Stellman & Greene, 2014:178); and "test-driven development" in which developers first create unit tests of a requirement, then the software is improved to pass that pass (Stellman & Greene, 2014:35).

According to PMI (2015), RE is "*the process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed".* In accordance with Parker (2013), the main goal of RE is to perform engineering driven solutions and to deliver mainly product features, rather than of business side benefits. RE

activities generally focus on functional and nonfunctional requirements instead of business, stakeholder, and transition requirements.

RE practices generally include elicitation, analysis, documentation, validation and management of requirements activities (De Lucia, & Qusef, 2010: 214). RE processes in waterfall like traditional software development methodologies focuses on gathering all requirements and then preparing a specification document before going to design phase (Javanmard et al., 2015: 1388). However, agile methodologies focus on changing requirements even late in the development lifecycle (Flora et al., 2014: 3627); hence handling changing requirements in every phase of agile software development is a critical issue which highlights the necessity of agile RE practices.

In agile methodologies, RE is performed through continual and iterative exploration of the business need; agile RE takes an iterative discovery approach. IIBA and Agile Alliance (2015) presents useful specifications and conceptual flow of business analysis activities in Scrum framework. Requirements are elicited and refined through an iterative process of planning, defining acceptance criteria, prioritizing, developing, and reviewing the results.

Table 1: Overview of Common Agile Methodologies

Agile Methodology	Key Practices	Advantages	Challenges
Scrum	Iterative increments; 2 to 4 weeks iterations as sprints; product backlog; spring backlog; sprint planning; the daily scrum (or stand-up); sprint reviews; sprint retrospectives	High level communication; high involvement of user as Product Owner; self-organizing teams and feedback	Weak documentation; can easily get off track; changing requirements
Extreme Programming (XP)	Iterative increments; 1 to 6 weeks iterations; user stories; pair programming; test driven development; refactoring	Active end user involvement; frequent feedback opportunities; strong technical practice	Weak documentation; unclear needs of clients; lack of disciplines; small teams that are suitable only for smaller projects
Dynamic Systems Development Method (DSDM)	Iterative; detailed documentation; prototyping; feasibility and business study	Strong control on project lifecycle; user involvement through frequent releases; requirement priority approach	Complex and time consuming documentation; expects continuous user involvement
Feature Driven Development (FDD)	Iterative; 2 days to 2 weeks iterations; suitable for complex projects; many members and multiple teams working in parallel; UML diagrams and modeling with detailed documentation	Method simplicity; Easy to understand because of documentation; user involvement through frequent reports	Less communication within and out of team; individual code ownership; complex approach for small projects
Adaptive Software Development (ASD)	Incremental; 4 to 8 weeks iterations; basic documentation; learning cycle	User involvement through frequent releases	Weak documentation; small teams and suitable only for smaller projects
Crystal	Incremental, informal and face-to-face team communication	High risk and highly important component given first; efficient coordination and communication of bigger teams; user involvement through frequent releases	Planning and development is not depended on requirements

Moreover, the studies of Paetsch et al. (2003) and Schön et al. (2017) review the agile RE activities in five phases which also corroborates with the specification of IIBA and Agile Alliance (2015): discovery and elicitation of new requirements (through techniques such as interviews, use case, observation, focus groups, brainstorming, prototyping etc.); refinement and analysis of new ideas and requirements; prioritization of requirements through requirement value measurement; checking and review of requirements; and then documentation.

Fancott et al. (2012) puts forward that agile RE mainly rely on conversations with business and implicit knowledge of the stakeholder. Analysts needs to constantly ensure that the features demanded by the customers align with the business goals, and benefits from

frequent feedbacks from customers. Results of successive iterations help analysts to refine requirements, mitigate risk early in the project and deliver right solution on time within budget.

As stated by the studies of Käpyaho et al. (2015) and Paetsch et al. (2003), the guidelines for RE provided by agile methods are ambiguous; since the concepts related with agile RE are still being developed. Some common characteristics of agile RE are strong use of face-to-face communication, iterative requirements practices and design, continuous requirements prioritization, prototyping or other modeling activities to make sense of requirements, test driven development and acceptance testing to ensure the quality and right direction (De Lucia, & Qusef, 2010; Käpyaho et al., 2015). Another study conducted by Cao and Ramesh (2008) in software development organizations on their agile RE practices found that face-to-face communication, prototyping and reviews and tests are common agile RE practices. However, it is seen that all of these practices bring some inherent challenges.

The literature studies of Cao and Ramesh (2008), Bjarnason et al. (2011), Paetsch et al. (2003) show that projects that are realized by agile methodologies have RE challenges; such as managing with very little documentation, motivation issues of team for constant RE work, not understanding the importance of writing tests first, not understanding the big picture, neglecting quality requirements, unrealistic expectations of customers due to early UI prototypes, neglect of non-functional requirements (NFRs), unavailability of customer during acceptance test writing. The study of Wagner et al. (2018: 11-12) provided a well-defined problem list that are commonly occurring in the context of agile projects and examined how criticality of those problems is judged by practitioners. According to research findings, incomplete and/or hidden requirements and communication flaws between project team and customer were the top of the list of criticality.

3. RESEARCH DESIGN

We conducted an interpretive research to better understand the case context as well as the dynamics of RE in projects realized by Scrum framework. Data was collected qualitatively from the case organization by documentation review, observations and semi-structured interviews for a year.

The case organization is a small-medium enterprise (SME) providing software development outsourcing services for various industries. The company was established in 2006 by two co-founders (i.e., top executives) with aim of providing location-based software and data services for enhancing marketing operations and optimizing sales activities. Within the first four years of its establishment, the company have followed traditional software

development techniques. With technological advancements and increasing demand for company services, existing software development techniques have transformed into more agile approaches in order to develop high quality products with customer involvement and to reduce time-to-market.

Currently, the company makes use of a Scrum agile framework as the most predominant approach in use today. Accordingly, in research context, two different Scrum software development projects with their agile RE practices are examined for period of one year. Profiles of case projects that aim to develop two distinct software systems for external clients serving in different industries are presented in Table 2.

In order to assure validity of qualitative research, this study benefits from two forms of data triangulation; source triangulation and methodological triangulation. In terms of source triangulation, multiple data sources are used for collecting the same data; for instance, different interviewees are asked about what RE challenges the project group have encountered. In terms of methodological triangulation, different types of data collection methods; documents, observations and interviews are used to acquire conclusions from qualitative data.

Table 2: Case Projects Profile

Attribute	**Project A**	**Project B**
Industry of Client	Commercial vehicles and transport	Building products
Scope of the Contract	Software outsourcing agreement of a hybrid mobile and web based Dashboard application development	
Context of Software	Route planning analysis and optimization; sales visits analysis and monitoring services	Sales execution tracking; merchandising performance of sales teams tracking and score card auditing services
Project Complexity	Medium-High (including client's ERP system integration and maintenance)	Medium (including client's CRM system integration)
Project Duration	16 weeks; 4 sprints	12 weeks; 3 sprints
Project Outcome	Completed successfully within scope and on time	Completed within scope with 3 weeks of delay
Financial Scale & Resource Ownership	Medium financial scale for both perspectives; jointly owned and controlled resources	

Preliminary data was collected by observations and documents to achieve a deep understanding of the phenomenon and case context. Attending meetings -kickoff, sprint

planning, daily stand-ups, sprint retrospective meetings- were critical part to cooperate with project members and to acquire knowledge by asking questions about RE work and discussing. Results from preliminary data analysis guided further data collection.

Primary data was obtained from semi-structured interviews for improvisation and exploration of RE issues. Semi-structured interviews do not limit the scope of the answers as much as structured interviews. Two rounds of one-to-one interviews were conducted with agile team members and Product Owners of both projects. Table 3 presents information about profile of the interviewees.

In the first round of one-to-one interviews, practitioners were interviewed about RE practices in use for agile software project development; the second round of one-to-one interviews focus on the perceived challenging RE issues and critical agile terms that affect project success. In both agile projects in research, same agile team -two Software Developers, Business Analyst, Test Specialist- fulfill the role of developing and delivering the software in question.

Another critical role in agile project development, Product Owner is responsible for representing the needs and desires of the stakeholders, defining the product backlog, providing direction of the product and perform backlog prioritization depending on customer value, and representing the work of the agile team members to the stakeholders. In case of Project A, Product Owner is on side of vendor; whereas in Project B Product Owner is preferred from client side; which have different pros and cons. Agile RE practices investigation from different Product Owner perspectives help to reveal critical insights and to provide an exploratory research approach.

Table 3: Interviewees Profile

Role		**Software Development Experience**	**Experience in Organization**	**Experience in Agile Team**	**Experience in RE**
Agile Team Members	Software Developer (1)	13 years	9 years	3 years	Requirements elicitation and experience in UML modeling
	Software Developer (2)	9 years	5 years	3 years	Some experience in writing requirements
	Business Analyst	10 years	4 years	4 years	Worked as requirements engineer and system designer for many years
	Test Specialist	5 years	2 years	2 years	Have knowledge on requirements for writing test plans and scenarios
Product Owner of Project A (on side of vendor)		15 years	15 years	3 years	Requirements elicitation and prioritization
Product Owner of Project B (on side of client)		8 years	11 years	4 years	Requirements elicitation and prioritization

In order to assure validity of qualitative research, this study benefits from two forms of data triangulation; source triangulation and methodological triangulation in accordance with the study of Runeson et al. (2012). In terms of source triangulation, multiple data sources are used for collecting the same data; for instance, different interviewees are asked about what RE challenges the project group have encountered. In terms of methodological triangulation, different types of data collection methods; documents, observations and interviews are used to acquire conclusions from qualitative data.

In the light of this information, the research in this study focus on descriptive analysis of the RE practices and key challenges of RE in agile project development. To this end, three main research questions are formulated to steer the design of our study and these research topics are examined in both case projects: *What are the perceived benefits of agile adoption? What are the agile RE practices in use? What are the key challenges of Agile RE on Project Success?*

4. FINDINGS

4.1. Agile Adoption Benefits Perception

Collabnet Versione (2019) underlies that the top three benefits of agile adoption are ability to manage changing priorities, project visibility and business/IT alignment. Accordingly, in the first step of research, benefits of agile adoption are examined in the case organization from three perspectives: Agile Team Members, Product Owners and also Top Executives of the case organization.

In accordance with survey results illustrated in Figure 1, this preliminary analysis is critical in order to understand perception and motivation of participants for agile approach and how they see agile practices importance for the organization and for project development.

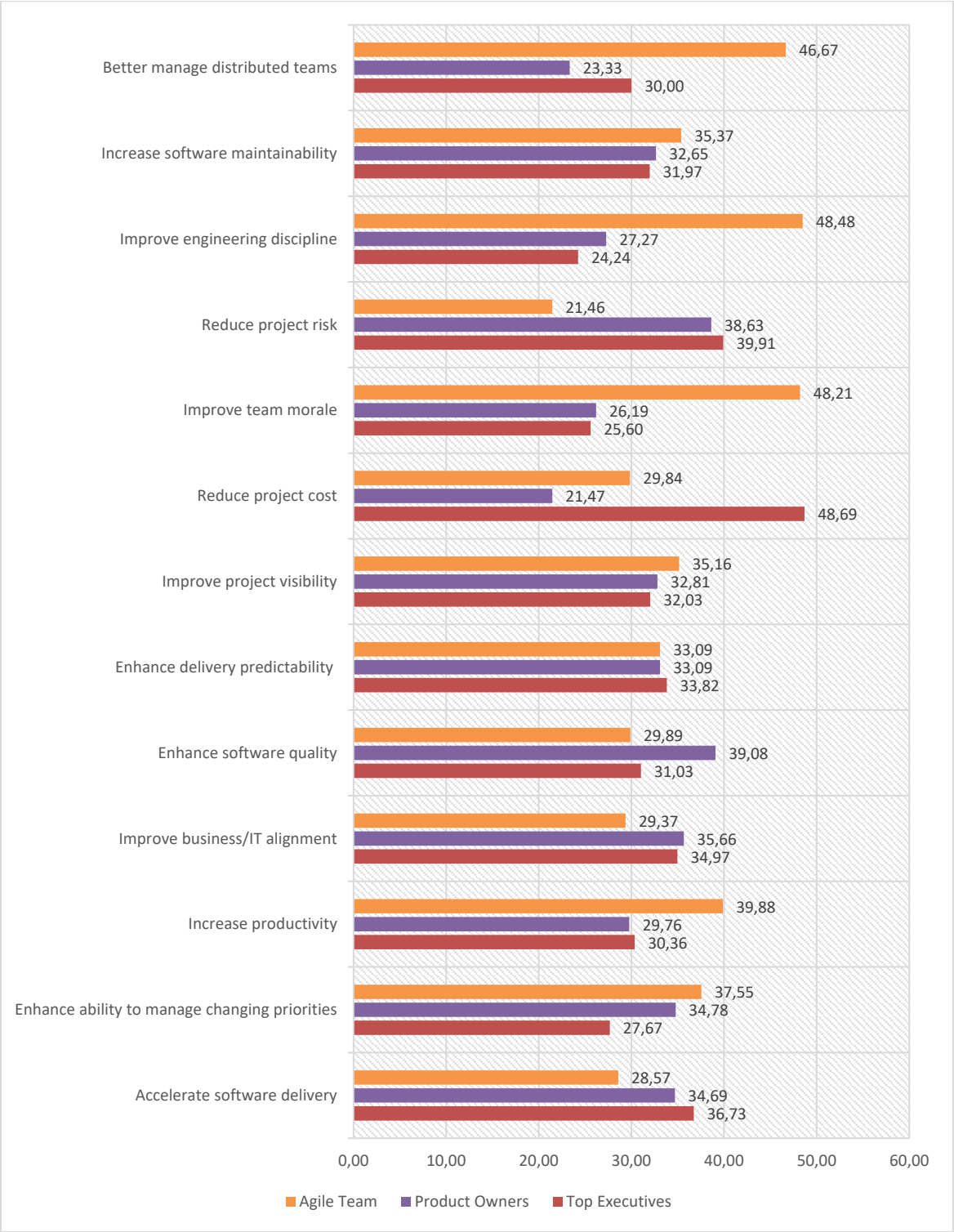

Figure 1: Perception of Agile Adoption Benefits

Two Top Executives of the organization highlighted that three main reasons behind adopting agile practices is to accelarate software delivery as shortening development cycles (36.73%), reduce project cost (48.69%) and reduce project risk (39.91%). Similar to perception of Top Executives, Product Owners put forward the benefits of software delivery acceleration

(34.69%) and project risk reduction (38.63%). Besides, they emphasize the importance of changing priorities management (34.78%) and enhances project outcome quality as software quality (39.08%). Furthermore, as another key finding, Top Executives and Product Owners put forward that agile approaches can yield effective business and IT alignment (~35%) within organization, since involvement of stakeholders from business, IT and client-side increase in agile project developement.

On the otherside, Agile Team Members mainly emphasize that agile practices facilitates better management of distributed teams (46.67%) and team morale improvement (48.21%). Collaboration of team members is described as an incentive of agile project development. In addition, they perceive that agile practices improve their productivity (39.88%) and their work in manner of engineering discipline (48.48%) and help to manage changing priorities better (37.55%). They put forward agile transition as a need for handling change in requirements; therefore, RE activities deserve the greatest care within the organization.

4.2. Agile RE Roadmap

As stated by De Lucia and Qusef (2010: 214), *"37% of the problems occurred in the development of challenging systems are related to the requirements phases"* and accordingly *"the problems inserted in the system during RE phase are the most expensive to remove".* Agile RE processes are not centralized in one phase of software development; they are spread throughout project development.

By using preliminary data obtained from observations –sprint planning, daily stand-ups, sprint retrospective meetings- and documentation studies and also data from first round of semi-structured interviews, general software development and agile RE roadmap of case organization is conceptualized and modeled as in Figure 3. According to the observations –sprint planning, daily stand-ups, sprint retrospective meetings- and documentation studies, "user stories" are the central mechanism for defining requirements in an agile manner. User stories are briefly defined and organized in product backlog by customers of projects (i.e., Project A and Project B). Since product backlog needs to provide expected business value; each user story is detailed -by agile team and product owner- and accompanied by a list of acceptance criteria with discovery sessions and interviews.

Business Analyst confirmed that *"Actually agile practices do not rely on heavy documentation; however product backlog including well-written and well-constructed user stories is necessary for our team to facilititate an efficient requirements discovery. As we*

interact, collaborate with Product Owner regularly in face-to-face meetings, product backlog is efficiently revised and constantly prioritized".

Two main categories of agile RE practices are observed and presented hereunder by using the interviewee notes in agile projects: business level and software level.

4.2.1. Business Level RE

As shown in Figure 2, after definition of product backlog including user stories, business level RE initiates. In business level, agile teams work with stakeholders and Product Owner to find out the enterprise need in detail, system and its services, system's functional, performance, security and other NFRs.

Business process models and process flow diagrams are developed with the collaboration of Product Owner and Agile Team. Business process models are prepared in formal (e.g., BPMN, Business Process Modeling Notation) and in free form textual format in order to document and perform elicitation of functional requirements with the help of face-to-face communication techniques; interviews and brainstorming.

Business Analyst of team stated that "*Involvement, decidability and availability of Product Owners in development of business process models is essential to quickly produce the first touchable conceptual model of the system, it's like a non-working but a prototype of the system for software developers."*

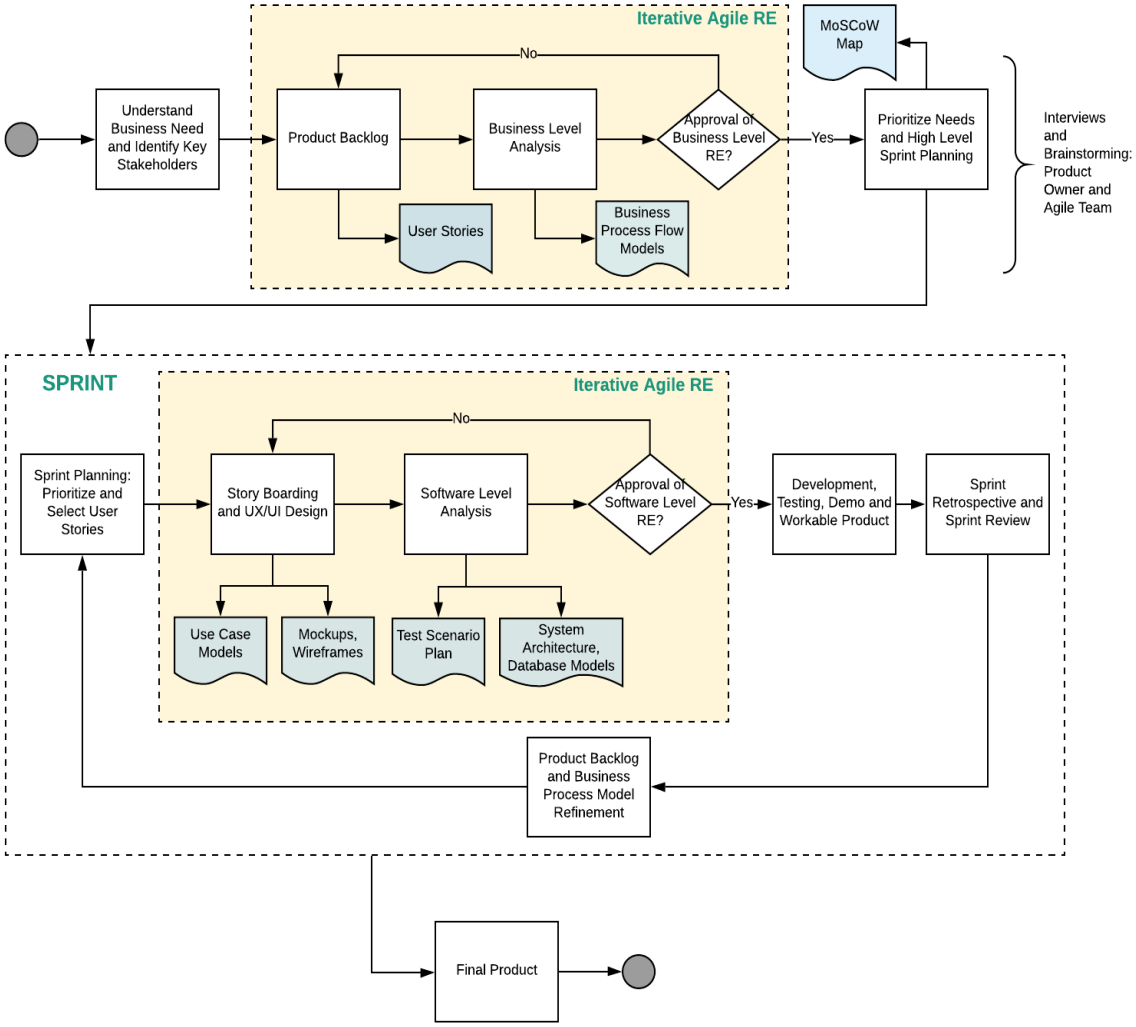

Figure 2: Agile RE Roadmap and Software Development Model

Product Owner of Project A confirmed that *"Documentation of business process model at beginning of software development process is a useful activity to understand and see the big picture as business need, to discover critical user stories and to perform requirements prioritization easily."* On the other side, Product Owner of Project B stated that *"Actually developing the right business process model is somewhat hard and time-consuming due to changing business needs and customer requirements; the refinement of the model is necessary many times to handle priorities faster".*

After approval of product backlog, user stories and business process model, requirement prioritization is performed by using MoSCoW technique that allows to analyze priorities within scope of Must Have, Should Have, Could Have and Won't Have features or requirements. With

results of MoSCoW analysis, agile team estimates how many sprints will be required and then defines tasks will be performed for each sprint.

Software Developer (1) mentioned that *"MoSCoW technique is so beneficial for prioritizing all must-have/nice-to-have functional/non-functional requirements. So that, all parties reach a common understanding on importance of delivering a user story or another item. However, when Product Owners are on client side, they mainly focus on operational, functional issues of the system. We try to perform many irregular meetings at the beginning to direct their focus on non-functional parts of the system. For example, we generally ask questions on availability of the system, authorization and authentication mechanisms, hierarchy of users and user management, prevention mechanisms."*

Furthermore, Software Developer (2) confirmed that *"In some cases, Product Owners on client side may not have enough information about such technical issues; hence detailing non-functional parts and creating acceptance criteria becomes challenging and time-consuming task".*

4.2.2. Software Level RE

At the beginning of each sprint, in sprint planning, Agile Team reviews and selects prioritized items of product backlog -user stories- and identifies necessary tasks to realize each user story within the sprint period. With the preparation of sprint backlog, software level RE analysis is initiated by Agile Team (i.e., Software Developers, Business Analyst and Test Specialist) in an iterative manner.

The findings from observations and interviews expose that use case analysis and User Interface (UI) studies as mock-ups are indispensable on software level RE of case organization which allows to perform user story decomposition and gather list of all use cases.

UML based use case diagrams are developed and documented to identify actors involved in the interaction and to describe the interaction itself. Use case diagram and UI studies Mock-ups and use case diagrams complements each other; they are performed and refined iteratively. Case organization is mainly using Proto.io (https://proto.io) to conduct mockup and prototyping studies; it utilizes a drag and drop UI, provides interactive and high fidelity screens and does not require coding.

Business Analyst highlights importance of UI study as follows: *"Mock-ups show relationships between user stories and use cases that the system user must be able to accomplish on system. It is a good way of representing project outcome instead of traditional wireframing. Even wireframes represent a product's structure but they are not clickable. Today mock-up kits*

allow to display how the product is going to look like; we can also add interactions and clicks on screens that helps software developers to get all features visually in agile manner".

Software Developer (2) also confirmed that *"Mock-ups enable us to define front-end and back-end coding requirements rather strongly. The more details in visualization will provide the more real which will increase efficiency of software development".*

Product Owner of Project A stated that *"Mock-ups like a prototype help us to feel the final product, although mockup screens have some constraints since they have no back-end code, just provides front end design. We can partially experience interactions and use cases in the interface."* Product Owner of Project B stated that *"Mock-ups facilitate next prioritization process, next sprint planning and tasks definition; furthermore, help us to foresee any possible changes and requirements in the early sprints".*

We also observed how Product Owners and Agile Team deal with changing requirements. It is seen that product backlog seems to be the common way to work with changing requirements in agile projects. They update the product backlog and mockups to handle changing requirements, and perform a brief impact analysis for accurate understanding of the implications of a proposed change. Impact analysis is conducted between requirements and mockup design; not on the code itself.

Test Specialist highlighted that *"When business units see mockup design, they generally want to add new requirements, change existing requirements and make current requirements more detailed both in the product backlog. More efficient and robust impact analysis techniques are necessary to monitor changes on testing and code as well; to track requirements from their origin to the deliverables; and to track changes between requirements, test cases and code".*

Business Analyst stated that *"Unfortunately, we are lagging behind the schedule in dealing with test cases, since Product Owners and Software Developers mainly focuses on refining product backlog, mockup designs and business process models".*

In addition to use case models and mockups, case organization focuses on development of test scenario plans and test cases, system architecture and database analysis in free form textual format on spreadsheets; no specific standard or modeling language is used in these studies.

During the observations of both agile projects (i.e. Project A and Project B) it is obviously seen that Product Owners and Software Developers mainly prefer to focus on mock-ups, UI studies for approval of software level RE and development in sprints instead of reviewing and refining other documentations.

Communication techniques and knowledge sharing practices is other important factor to build ties between project team members and stakeholders. According to the observations of both agile projects, case organization prefers to use the following four vital tools which are expressed as "*agile development and RE facilitator*" by Agile Team:

- Trello (https://trello.com) is used for planning sprints, creating lists and cards on boards, tracking their status, prioritizing tasks and allowing to work more collaboratively.
- Slack (https://slack.com) is mainly preferred for team collaboration and messaging; allows teams to create and join a workspace; provides online messaging and document sharing platform and more.
- JIRA (https://www.atlassian.com/software/jira) is used for planning, tracking and managing software development projects.
- Skype (http://skype.com) is also partially used for communication with Product Owner and other representatives of customer; provides irregular meetings via online group video chats as an alternative to face-to-face meetings.

Up to this point, the conducted research tries to comprehend and present existing RE practices and agile software development cycle, and expose the agile RE roadmap of an IT organization (i.e., case organization) which has over 10 periods of experience in software development domain and serving for various industries.

4.3. Challenges of Agile RE on Project Success

The final part of our research focuses tries to figure out and reveal challenges of Agile RE practices in the projects (i.e., Project A and Project B) from different perspectives and understand impact of those challenges on project success by conducting semi-structured, one-to-one interviews with participants. Agile Team members and Product Owners evaluated each other and highlighted some known -from the study of Wagner et al. (2018)- and quite impressive challenges regarding agile RE practices.

The following questions are examined in the second round of semi-structured interviews: *What are the key challenges in agile RE? What are critical issues regarding Software Developers/ Business Analyst/Test Specialist within context of agile RE that may have negative impact on project success? What are the critical issues regarding Product Owners within context of agile RE in terms of Jeff Sutherland's PO requirements?* The interview

research results are summarized and presented respectively in the Figure 3, Figure 4 and Figure 5.

Figure 3 represents key challenges in agile RE regarding two Software Developers of Agile Team. According to the citations provided by Product Owners, Business Analyst and Test Specialist; weak documentation of coding cycle (~83%), insufficient support for business domain (~78%) and more focus on technical perspective, weak test driven development (~76%) and weak modular programming capabilities (~72%) of Software Developers are mainly highlighted critical issues which may have impact on agile project success. Another key finding is Software Developers' insufficient support for analysis and modeling activities (~64%); which is also confirmed in the first round of interview by Business Analyst; mock-ups help *"to get all features visually in agile manner"*.

Figure 4 represents key challenges in agile RE regarding Business Analyst of Agile Team. According to the citations provided by Product Owners, Software Developers and Test Specialist; missing traceability and weak impact analysis (~83%), insufficient support for test driven activities (~79%) and unclear/unmeasurable NFR analysis (~75%) are top challenging issues of Business Analyst. Another highlighted issue regarding Business Analyst within context of agile RE is poor document management capabilities (~68%); more focus and time is given for mock-up studies in order to visualize the functionalities of new system.

Figure 5 represents key challenges in agile RE regarding Test Specialist of Agile Team. Poor test scenarios documentation (~85%), missing traceability and weak impact analysis (~81%), weak usage of test automation tools (~76%) and unclear test scenarios development (~75%) of Test Specialist that allow for various interpretations are the most citations provided by Product Owners, Software Developers and Business Analyst.

The involvement of PO from different parties -client side vs. vendor side- in selected case projects is another conspicuous factor of investigation. Within research scope, the perception and challenges regarding Product Owners is also examined in terms of Jeff Sutherland's requirements. Five-point Likert scale analysis is used for measuring the attitude; it allows survey participant to express how much she agrees or disagrees with a particular requirement.

According to Sutherland (2013), an adequate Product Owner needs to meet at least following requirements: *Knowledgeability* (i.e., needs to have most of or all the knowledge about the product/project), *Availability* (i.e., needs to be available to the team to do the day-to-day work), *Decidability* (i.e., needs mandate to make decisions) and *Accountability* (i.e., is responsible for creating value).

According to Likert scale results regarding both Product Owners, as shown in Figure 6, vendor side Product Owner's availability for agile development and accountability for creating value is found as stronger than client-side Product Owner. Closeness of Product Owner to Agile Team, accessibility of Product Owner for agile work and knowledgeability on agile RE are considered as critical factors by Agile Team members for completing project within scope, budget and on time.

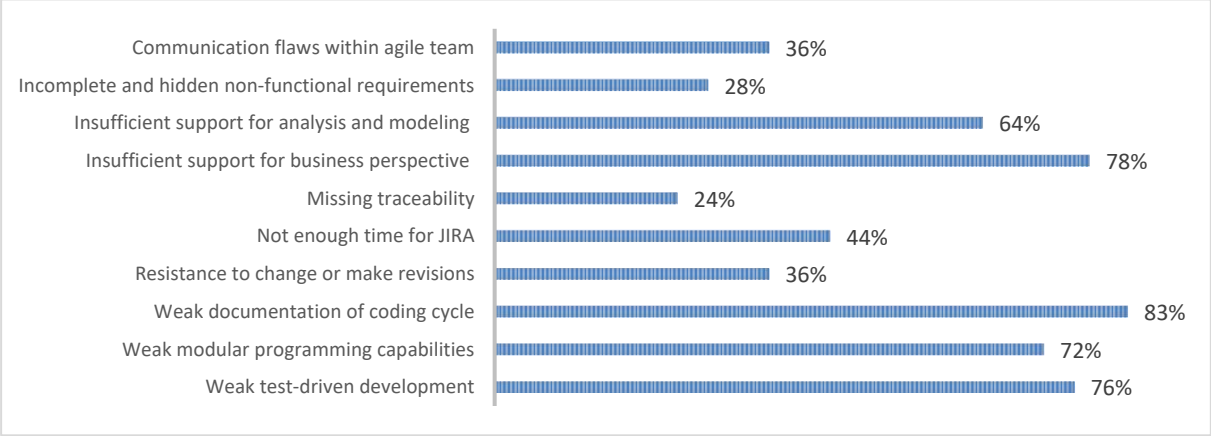

Figure 3: Key Challenges in Agile RE Regarding Software Developers

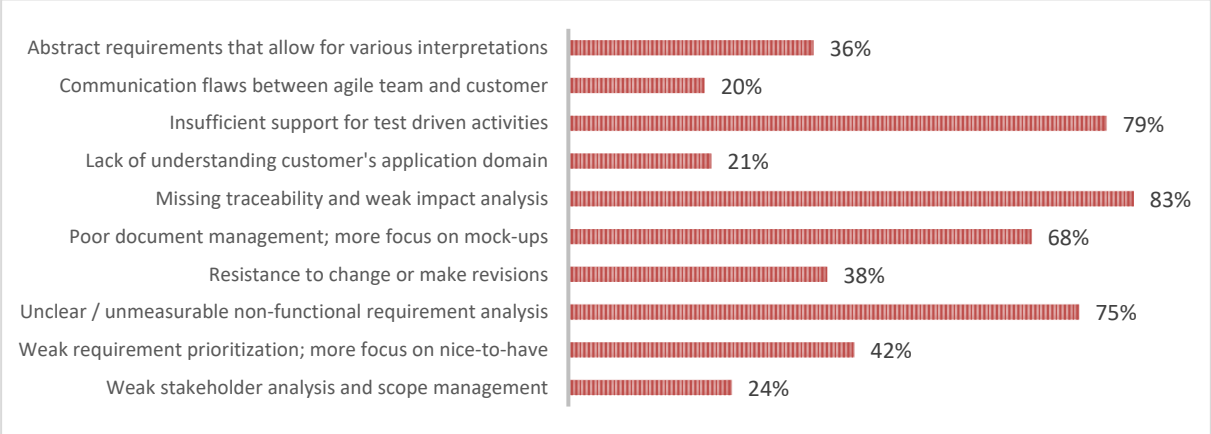

Figure 4: Key Challenges in Agile RE Regarding Business Analyst

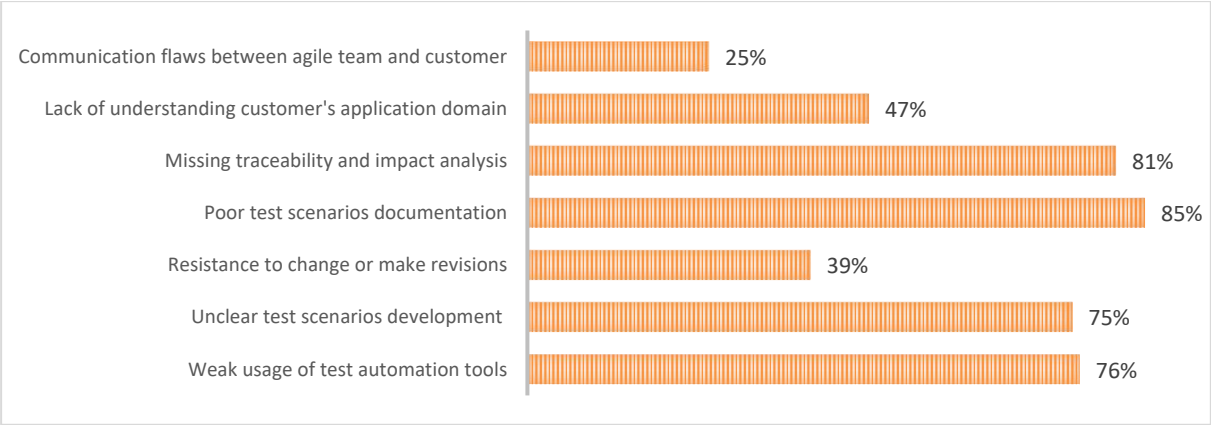

Figure 5: Key Challenges in Agile RE Regarding Test Specialist

On the other side, decision making capabilities and decidability of client-side Product Owner is more advanced and robust in activities such as business process modeling, requirements prioritization, and change management. Vendor side Product Owner's decision-making capabilities and authority mechanisms are not as agile as client-side Product Owner.

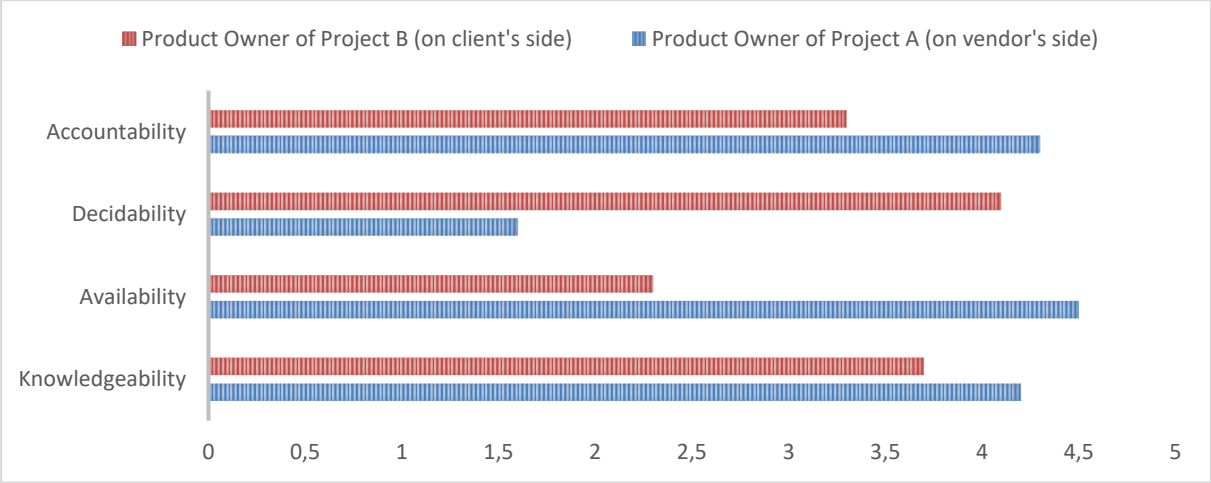

Figure 6: Assessment of Product Owners

5. ANALYSIS AND DISCUSSION

Wide adoption of agile RE practices provide various benefits such as better understanding of customer needs, better productivity, software delivery acceleration, improved engineering discipline. Within research context, the case study revealed mostly same issues as previous relevant literature on agile RE practices and their benefits; furthermore, distinct challenges associated with agile RE practices from different viewpoints are addressed. Key agile RE artifacts in use are as follows:

- User Story for describing a feature of the solution by Product Owner and then detailed with written text and acceptance criteria by Agile Team,

- MoSCoW for prioritization of user stories and sprint planning by Product Owner and Agile Team,
- Use Case Diagram for visualization of the solution which shows the interaction between user and system in UML context by Agile Team,
- Use Case Scenarios for textual representation of the solution and system design description by Agile Team,
- Kanban Board for visualization of requirement progress through development workflow by Agile Team,
- Mock-ups for visualization of the system for describing UI specifications by Agile Team and Product Owner.

Agile Team study participants emphasized intensive communication between the Agile Team and Product Owner that have positive impact on project outcome. As stated by Stellman et al. (2014: 96), *"Like all good agile teams, Scrum teams rely heavily on face-to-face communication to understand exactly what they're building".* Instead of following a clearly defined, formal knowledge sharing and documentation procedure in case projects, more dynamic and adaptive approach is followed in order to describe system solution accurately. Some platforms include various but non-essential features for agile projects; these tools may become overwhelming and time-consuming as stressed by study participants. More user-friendly platforms for online messaging and document sharing and Kanban board techniques are particularly preferred to manage sprints and workflows. Today, selection and use of appropriate communication and documentation platforms that involves efficient issue feature is indispensable within the scope of agile project management; which may also support challenge of requirements traceability.

In addition to mentioned key RE activities, role of Product Owner is another important factor on the project success. Stellman et al. (2014: 96) emphasizes that *"Product Owner helps the team understand the software value and has a very active day-to-day role in the project development"*. It is obviously observed from case project that the notable advantage of a Product Owner on vendor side is the availability and accountability over Product Owner on client side. A Product Owner in general needs to collaborate and engage in most of the project development activities -sprint planning, prioritization, refinement of backlogs etc.- with Agile Team in order to maximize product value. Product Owner on vendor side feels more responsible for delivering right business solution with right features on time and within estimated cost;

briefly delivering higher value. On the other side, closeness to customer stakeholders, business knowledge and client's culture cannot be underestimated. Although Product Owner on client side as customer's official representative with high decidability enables to provide right answers and make right decisions regarding software development; unavailability and unaccountability of a Product Owner may result in delays, deceleration of software delivery, lower quality and lower value delivery which is observed in the second case project (i.e. Project B).

In the light of studied case data, this study proposes and underlies three approaches in order to improve the performance and quality of recent Agile RE processes: Agile User Experience (Agile UX), Agile Business Analysis (Agile BA) and Hybrid Software Development.

5.1. Agile UX

The research demonstrates that the current trend is to integrate UX design with agile approaches; namely Agile UX. Agile approaches strive to develop small sets of working software quickly through high communication and collaboration instead of heavy documentation. In the general agile concept, agile team is responsible for developing UI without having an understanding of the user needs and experience.

UX is more than UI; in UX concept, close cooperation of UX designer with product team and stakeholders is required to discover and define user requirements through user evaluations. At this point, Product Owner has a critical obligation to define and refine requirements with the help of in-depth UX research regarding the system to be developed.

The case projects revealed that UI and mock-up studies expedites agile RE activities of project participants; while having some negative effects on other RE activities such as on UML modeling and analysis, NFRs elicitation and information level requirements elicitation and analysis, data dictionary documentation and so on. To mitigate these negative effects, establishing more well-structured and planned sprints is needful.

Accordingly, Kieffer et al. (2017: 578) reviews beneficial guidelines for facilitating more efficient Agile UX Sprints: iterative and incremental UX and agile activities, continuous user involvement, efficient time allocation for up-front activities to elicit user and functional requirements, realizing rapid formative usability to fulfill UX goals while delivering working software frequently and finally documentation of up-front analysis, design and usability findings. Moreover, the role of UX Designers in agile projects needs to be empowered in agile projects. Their clear understanding of bigger picture, product vision and requirements and their ability to detail of user stories concretely and articulate acceptance criteria will consolidate

Product Owner's position. UX Designers' presence and active participation in agile ceremonies will streamline the prioritization of requirements and refinement of backlogs, and also validation of UX studies.

5.2. Agile BA

Integrating traditional RE practices to agile methods is a complex and critical process which was first seen during the observations and later confirmed in interviews. RE activities primarily focus on developing products or software; do not involve in activities such as improvements of business processes, developing business cases, delivering business benefits. RE activities generally address functional and non-functional technical requirements.

Within this perspective, Business Analysis term in general is much broader concept than RE. PMI (2015) defines Business Analysis (BA) as *"the application of knowledge, skills, tools and techniques to determine problems and identify business needs; identify and recommend viable solutions for meeting those needs; elicit, document and manage stakeholder requirements in order to meet business and project objectives; and finally facilitate the successful implementation of the product, service or end result of the program or project"*. Business Analysis concept focuses on delivering solutions that improve business outcomes and addresses people and process issues not only software, technology.

Transition from RE to BA concept, namely the term Agile BA is another critical approach that should be followed to enhance agile Scrum requirements management studies. The role of an Agile Business Analyst is extremely essential for organizing the business needs and creating roadmap for the project team; focusing on applying an agile approach within BA framework.

Integrating BA approach in an agile environment, Agile BA provide valuable guidance on creating effective user stories and story maps and performing effective information level analysis regarding the business solution. As stated by study participants, poor focus on analysis of NFRs (i.e., performance, reliability, scalability, security, availability, usability etc.) is another critical issue in Agile RE activities which needs to be strengthen with appropriate procedures. They need to be visualized with acceptance criteria in backlogs and tracked by Product Owners and Agile Team members.

Agile BA approach consolidate traceability of all kinds of requirements (i.e., business, stakeholder, functional, non-functional and transition) changes, validation and verification of requirements and stakeholder impact assessment capability of the case organization; at this

point, PMI (2015) and IIBA and Agile Alliance (2015) provide a great portfolio of specifications and standardization of BA techniques.

5.3. Hybrid Software Development: Lean + Agile

In addition to all the foregoing, another guideline is to combine lean and agile approaches in software development. As mentioned in Section 2, the Collabnet Versione (2019) highlights the Scrum/XP Hybrid methodologies (~64%) as the most common agile methodologies used by organizations today. Hybrid agile methodologies allows to take advantage of different approaches and particularly to enhance agile RE capabilities, to achieve better and sustainable productivity in agile RE activities.

Within the research context, it is seen that just applying main Scrum practices alone is not enough; specifically, test-driven development and pair programming aspects of XP programming have a great potential to enhance the sprint performance of Scrum approach in use and to promote higher product value. At this point, engagement of lean and Kanban practices in sprints can heal the workflow management between Agile Team members with the philosophy of continuous improvement and elimination of waste (e.g., non-value adding practices, delays, handovers) that may occur in software development. Actually, this can be achieved with the help of the experts who have knowledge on lean practices in the software domain.

6. CONCLUSION

There is an increasing pace on the information technologies industry; making a short time-to-market is vital in order to have a competitive advantage. As it is seen from the industry implementations, wide adoption of agile approaches by software development lifecycles allow software projects to be conducted with the lower costs, higher quality, better productivity and better customer satisfaction. Software development companies actually needs to compare the benefits and costs of agile practices in their project development environment.

The key point for delivering projects successfully -on time, within scope and within budget- is to produce high quality and clearly defined requirements that meet customer expectations. Besides, the report of PMI also emphasizes that inaccurate requirements gathering is the primary cause of project failure by 37% of the organizations; poor requirements management practices and changing organization priorities are leading causes of project failure.

Today, the practical application of Agile RE -as a dynamic, iterative and adaptive process- and having mature RE practices is gaining great importance. This research tries to understand the recent issues and dynamics of agile RE in projects realized by Scrum framework

by realizing an interpretive study with a case analysis of two agile projects. Our concluding analysis revealed that intensive communication between the Agile Team and Product Owner, and availability and accountability of a Product Owner have notable benefits for the project success. In order to improve the performance and quality of Agile RE processes, three recent approaches are highlighted within research context with their potential contributions for software development; Agile UX, Agile BA and Hybrid Software Development including lean approaches and integrating different side of agile methodologies.

The results of this research are based on two agile projects of one organization; it may therefore create a limitation and a threat to external validity. Instead of aiming to generalize findings or to test hypothesis, this study aims to present recent challenging issues on working practices, draws lessons from case studies for practitioners and researchers, and puts forward with a set of agile RE guidelines for software development organizations that have similar agile RE roadmap.

REFERENCES

Ambler, S. W. (2018). Agile Analysis. Agile Modeling. http://www.agilemodeling.com/essays/agileAnalysis.htm#AgileAnalysis (date accessed: 09.08.2019).

Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. University of Western Australia, 30.9.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Kern, J. (2001). Manifesto for agile software development.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. & Kern, J. (2001). The agile manifesto.

Bjarnason, E., Wnuk, K., & Regnell, B. (2011, July). A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In Proceedings of the 1st Workshop on Agile Requirements Engineering (p. 3). ACM.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. IEEE software, 25(1), 60-67.

Clancy, T. (2014). The Standish Group chaos report. Project Smart.

Collabnet Versione (2019). StateofAgile Report: 13th Annual State Of Agile Report. https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508.

De Lucia, A., & Qusef, A. (2010). Requirements engineering in agile software development. Journal of emerging technologies in web intelligence, 2(3), 212-220.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and software technology, 50(9-10), 833-859.

Fancott, T., Kamthan, P., & Shahmir, N. (2012, December). Towards next generation requirements engineering. In 2012 International Conference on Social Informatics (pp. 328-331). IEEE.

Flora, H. K., & Chande, S. V. (2014). A systematic study on agile software development methodologies and practices. International Journal of Computer Science and Information Technologies, 5(3), 3626-3637.

IIBA & Agile Alliance. (2015). A Guide to the Business Analysis Body of Knowledge (Babok Guide). International Institute of Business Analysis.

Javanmard, M., & Alian, M. (2015). Comparison between Agile and Traditional software development methodologies. Cumhuriyet Üniversitesi Fen-Edebiyat Fakültesi Fen Bilimleri Dergisi, 36(3), 1386-1394.

Joslin, R., & Müller, R. (2016). The impact of project methodologies on project success in different project environments. International Journal of Managing Projects in Business, 9(2), 364-388.

Käpyaho, M., & Kauppinen, M. (2015, August). Agile requirements engineering with prototyping: A case study. In 2015 IEEE 23rd International requirements engineering conference (RE) (pp. 334-343). IEEE.

Kasauli, R., Liebel, G., Knauss, E., Gopakumar, S., & Kanagwa, B. (2017, September). Requirements engineering challenges in large-scale agile system development. In 2017 IEEE 25th International Requirements Engineering Conference (RE) (pp. 352-361). IEEE.

Kieffer, S., Ghouti, A., & Macq, B. (2017). The agile UX development lifecycle: Combining formative usability and agile methods.

Mersino, A, (2018). Agile Projects are More Successful than Traditional Projects. Vitality Chicago. https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/ (date accessed: 09.08.2019).

Musa, F., & Tariq, M. A. (2017). Agile Methodology: Hybrid Approach Scrum and XP. International Journal of Scientific & Engineering Research, 8(4).

Ochodek, M., & Kopczyńska, S. (2018). Perceived importance of agile requirements engineering practices–A survey. Journal of Systems and Software, 143, 29-43

Paetsch, F., Eberlein, A., & Maurer, F. (2003, June). Requirements engineering and agile software development. In WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. (pp. 308-313). IEEE.

Pantiuchina, J., Mondini, M., Khanna, D., Wang, X., & Abrahamsson, P. (2017, May). Are software startups applying agile practices? The state of the practice from a large survey. In International Conference on Agile Software Development (pp. 167-183). Springer, Cham.

Parker, J. (2013). Requirements Engineering vs. Business Analysis. Enfocus Solutions. http://enfocussolutions.com/requirements-engineering-vs-business-analysis/

Phil, M. (2015). Comparative analysis of different agile methodologies. International Journal of Computer Science and Information Technology Research Vol3, (1).

PMI, Project Management Institute (2008). A guide to the project management body of knowledge (PMBOK® guide)—Fourth edition. Newtown Square, PA: Author.

PMI, Project Management Institute (2015). Business Analysis for Practitioners: A Practice Guide.

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

Runeson, P., Host, M., Rainer, A., & Regnell, B. (2012). Case study research in software engineering: Guidelines and examples. John Wiley & Sons.

Schön, E. M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. Computer Standards & Interfaces, 49, 79-91.

Schön, E. M., Winter, D., Escalona, M. J., & Thomaschewski, J. (2017, May). Key challenges in agile requirements engineering. In International Conference on Agile Software Development (pp. 37-51). Springer, Cham.

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. International Journal of Project Management, 33(5), 1040-1051.

Standish Group CHAOS Report (2015). https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf (date accessed: 09.08.2019).

Stellman, A., & Greene, J. (2014). Learning agile: Understanding scrum, XP, lean, and kanban. " O'Reilly Media, Inc.".

Sutherland J., (2013). Requirements for Product Owner: Common Pitfalls. Scruminc. https://www.scruminc.com/requirements-for-product-owner-common/

Sutherland, J., & Schwaber, K. (2016). The scrum guide. The definitive guide to scrum: The rules of the game. Scrum. org, 17.

Vallon, R., da Silva Estacio, B. J., Prikladnicki, R., & Grechenig, T. (2018). Systematic literature review on agile practices in global software development. Information and Software Technology, 96, 161-180.

Wagner, S., Méndez-Fernández, D., Kalinowski, M., & Felderer, M. (2018). Agile requirements engineering in practice: Status quo and critical problems. CLEI Electronic Journal, 21(1), 15.

Wojewoda, S. (2015). Standish Group 2015 chaos report.

Yeo, Y. N., & Hahn, J. (2014). The role of project modularity in information systems development.