

A Systematic Approach to Project Related Concepts of Scrum

Necmettin OZKAN¹
Civan KUCUK²

Abstract

Scrum models have clustered within the boundaries of product development and the limited world of development teams. Nonetheless, any Scrum adoption cannot be isolated from the rest of organization's processes and structures. A typical example of this integration is with project related world. This study firstly focuses on the need for clarifying Scrum concepts to make ground sufficient in order to integrate Scrum world with project related notions. Secondly, it elaborates on project concept and project manager role in the world of Scrum, with additional and special interest on scalable solutions. The study also deals with the notions around the project (program and portfolio) for a full integrity in this context. In order to unite these two spaces, alongside the descriptions provided in Scrum Guide, the study is extended by using COBIT framework and PMBOK Guide through identification, description and evaluation of general roles and structures related to and in relation with the notion of project, which is not included in Scrum but potentially interacted with Scrum structures.

Keywords: *project, program, portfolio, Scrum, COBIT.*

JEL classification: Z00

Introduction

A change to agile methodologies entails major alterations to several aspects of an organization including project cycle, development model of SDLC (Software Development Life Cycle), organization structure, work procedures, culture, communication channels, roles of people and management styles (Schwaber, 2004 and Nerur et al. 2005). Despite such a broad and in-depth change, Scrum models have clustered within the boundaries of product development and the limited world of development teams. Nonetheless, any Scrum adoption cannot be isolated from the rest of organization's processes and structures. A proper agile transformation should go beyond the boundaries of software development and address Scrum and its ecosystem

¹ **Necmettin OZKAN**, Turkiye Finans Participation Bank,
E-mail: Necmettin.Ozkan@turkiyefinans.com.tr

² **Civan KUCUK**, Turkiye Finans Participation Bank,
E-mail: Civan.Kucuk@turkiyefinans.com.tr

from a holistic and wider perspective for a successful integration. In addition, Scrum which describes itself as a framework has left many practice details blank (Schwaber, 2004), like other framework models, thus it provides organizations flexibility. This advantage aside, high level descriptions and these blanks pose some risks especially when integrating Scrum with its outer world over its fuzzy-defined artifacts and roles.

A typical example of this integration is with project related world. Scrum Guide (Schwaber and Sutherland, 2013) does not unequivocally give a place for project and its related notions and mainly takes shape around the product concepts. In other words, descriptions clustered around project notion and interacted with Scrum world, such as project, project manager, program and portfolio have somehow left external as the origin of Scrum (Scrum Guide) does not provide any place for them. Moreover, despite many resources about Scrum project management, it is seen in them that the definition and boundaries of project notion disappear behind the product focus. As a result, some uncertain subjects such as reconfiguration of project concept according to Scrum, examining the place of project manager role in Scrum models, identifying the difference and similarities between a potential project manager role and Scrum Master and product owner roles, providing integration of portfolio and program management processes with Scrum project management processes and arranging the new position of committees that are already in place to select projects to develop still dwells in the world of Scrum. Scrum's distant standing from the plan oriented attitudes of project management keeps Scrum within the boundaries of sprint and release planning only. The most we see in this manner is in the 2011 version of Scrum Guide (Schwaber and Sutherland, 2011), in which there is a dedicated place for release planning but this part has been removed in the subsequent versions. Eventually, questions still survive for the expected relationships between sprint, release and project from project management point of view, Scrum's approach for managing the iron triangle and how to handle the units of customer requirements clustered in a project scope and scattered across the Scrum artifacts during the development from a holistic and a wider perspective.

Meanwhile, the needs for scaling agile practices caused by the charm of being agile increases and naturally creates an effect to shape Scrum and Scrum related processes of many medium and large-sized organizations. As a consequence of this case, Scrum authorities and practitioners have led to some solutions, listed in "agilescaling.org", which apparently support scalable Scrum applications. These scaling solutions shape around the fundamental Scrum concepts and result in enlarging genes of non-scalable core Scrum, and additionally pose possible improvements on program and portfolio management over a well-established project management structure. Consequently, from the window of project related concepts, it is still one of the hot topics of agile software development that is originally designed for relatively small teams (Yvette, 2015) to apply to scalable project sizes (Laanti et al. 2015). In the case of occurrence of corporate or organization size distributed area, subcontracting, developing large and complex systems, inter-team coordination, there are still some limitations with respect to the nature of Scrum (Akif and Majeed, 2012 and Laanti et al. 2015).

This study focuses on these issues: firstly, the need for clarifying Scrum concepts to make ground sufficient and possible in order to integrate Scrum world with project related notions. Secondly, it elaborates on project concept and project manager role in the world of Scrum at fundamental level for the purpose of describing Scrum and its ecosystem with a holistic perspective, with additional and special interest on scalable solutions. The study also deals with the notions around the project (program and portfolio) for a full integrity in this context. Thus, the work integrates these two parts in a conceptual dimension and at fundamental level. It is a fact that it is hard to come up with a complete or fully developed solutions answering all these issues as Scrum authorities still search for the same. The purpose here is to move one step ahead to open new perspectives for these issues.

In order to unite these two spaces, descriptions provided in Scrum Guide have been clarified and examined. Then, the study is extended by using COBIT (Control Objectives for Information and Related Technology) Framework and PMBOK Guide (A Guide to the Project Management Body of Knowledge) through identification, description and evaluation of general roles and structures related to and in relation with the notion of project, which is not included in Scrum but potentially interacted with Scrum structures.

1. Related Works

There are already other studies performed on Scrum notions and Scrum project management which mainly cover the notions in this study. Common feature of this kind of studies is to evaluate Scrum within its world and boundaries. Going beyond, in order to reach a set of related works that define program and portfolio in Scrum over the concept of project, a search with the keyword of “scrum project program portfolio” was conducted throughout of libraries that are accessible by the authors between the dates of 25/08/-06/09/2016. The list and the number of the results mentioning the terms are: CogPrints (0), Compendex (3), DBLP (0), Citebase Search (0), Directory of Open Access Journals (0), Inspec (0), ACM (2), IEEE Xplore (7), ScienceDirect (122), Springer (245), Arnetminer (2), Microsoft Academic Search (3), arXiv (0), and Odysci Academic Search (0). 383 results were investigated one by one over their titles or abstracts (when needed) to see possible matches. The result is that there is no similar work which study these terms in a conceptual dimension and at fundamental level by evaluating under a holistic frame.

Besides, additional investigation has been conducted for scalable models of Scrum that are listed in Agile Scaling (2015), including Scrum-of-Scrums, Large Scale Scrum (LeSS), Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD), Spotify Model, Drive Strategy Deliver More (DSDM), Recipes for Agile Governance (RAGE), Nexus and Scrum at Scale, whether any of them uses project as a mean and bridge to program and portfolio as used with the meaning in this study. Their scaling solutions similarly shape around the fundamental Scrum concepts and result in enlarging the non-scalable core Scrum model; adding more PO's, meetings, PBI's etc. falling far away from the intended solution this study aims to achieve.

On the other hand, there are studies which compare Scrum and classic project management and describe the transition from classic project management to agile project management. A typical example of these studies is the one performed by Sutherland and Ahmad (2011). The output of this study is again far from scalable Scrum structure and does not consider Scrum's environmental components and does not include program and portfolio management notions. Another study by Sliger and Broderic (2008), with a similar style but in a contrariwise way, shows the similarity between classic project management and agile project management by building the bridge among Scrum practices and PMBOK knowledge areas. This study, on the other hand, does not evaluate the notions in the basic level like in this study and is contented with establishing a static relation between PMBOK and agile notions. With this feature, the related study can be regarded as a new interpretation of PMBOK with an agile language.

This study, differently from others, has integrated these two parts in a conceptual dimension and at fundamental level by evaluating under a holistic frame. The other important difference of this study is the concern for the support to scalable and sustainable Scrum models with an overall and holistic perspective by establishing the project notion at the center.

2. Identification of Related Concepts

For Scrum notions, Scrum's common and pure form which is free from any customization and add-ins is aimed, thus Scrum Guide (Schwaber and Sutherland, 2013) is referred to as the primary source in order to identify and describe the Scrum notions. Additionally, the outputs from identification and description phases are enriched and supported with a literature review. At the stage of identification of Scrum notions, roles and artifacts in Scrum Guide are included. Scrum meetings are excluded as this study mainly is about static connections between the nations and they are mainly from dynamic dimensions of nations and do not add value in the context of this study.

Secondly, within the scope of this study, a complete and general list for project related notions is aimed to be achieved. The term 'general' indicates independency from industry and the size of the organizations. Completeness addresses to the purpose of achieving all related notions contributing the scope of this study. It is believed that a list in this scope can be achieved by the help of COBIT that is a universal de-facto standard in information technology. Under normal circumstances, COBIT includes all processes that have to be in IT structures, and provides details at a certain level for the processes. In this context, COBIT can be used to point out the information system processes, which Scrum contacts directly with and alters at some degree, and provides details of the processes. Among the product family of COBIT, COBIT framework document is preferred for this work because it is the basic product and the one that serves properly for the context of this study. Although the latest version of COBIT is version 5.0, version 4.1 which serves for the same purpose in the context of the study is preferred because of the authors' familiarity.

In this context, a scanning has been performed in the related COBIT process that is PO10 (Project Management) to identify project related notions that are typically roles and artifacts. Thus, aligned with the scope of this study, COBIT has gained a complementary position for Scrum, via PO10 process. Eventually, notions identified for both areas are listed below:

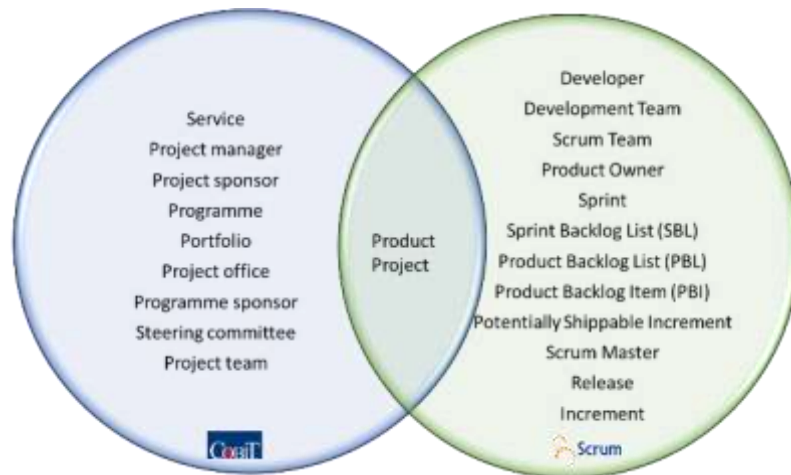


Figure 1 List of concepts

Source: Own study

It is seen that one of the common notions of these lists is product. Although, there is not a direct statement of project in Scrum Guide, yet it is explicitly mentioned in Scrum definition of Scrum Alliance, and in the headline of one of Ken Schwaber's fundamental works (Schwaber, 2004), as many other Scrum books. Thus, in order to ensure the integrity of the picture, project is regarded as an extension of the core Scrum in regard to its importance to Scrum's outer world. Finally, project and product are two key notions that unite these two worlds. For the terms obtained from COBIT, due to their primary importance, the fundamental concepts consisted of project, project manager, program and portfolio were discussed in detail. Secondary ones for the context of the study as project team, all types of sponsors, service, project office, and steering committee were just mentioned in the corresponding places.

In what follows, all these notions have been described and explained through the study and details are provided for those which are needed to be deliberated over.

3. Definition and Discussion of Concepts

3.1 Scrum Concepts

Scrum aims to deliver products via iterative and incremental model allowing early and continuous delivery of valuable software (Schwaber and Sutherland, 2013). Incremental design means organic growth of the system being developed (Nerur et al. 2005). In iterative models, rather than visiting each step only once, all steps are iterated

through until the system is deemed complete. In Scrum, projects are divided into brief and snappy work cadences which are called as sprints. Sprints are typically fixed in duration spanning from one week to four weeks (Sutherland, 2010). Developers and customers (or their surrogates) are actively involved in the development process and dynamically prioritize features in the sprint planning meeting (the beginning of each iteration) (Schwaber and Sutherland, 2013). During the sprint, intense 15 minute daily meetings allow participants constantly adjust to the realities of high-change projects (Schwaber and Sutherland, 2013). Each sprint ends with a working system as a deliverable at the end (Schwaber and Sutherland, 2013). Stakeholders including developers and end users go through repeated cycles of thought-action-reflection at the end of each iteration that foster an environment of learning and adaptation (Schwaber and Sutherland, 2013). These sprint review meetings provide an opportunity facilitating feedback and reflection on what worked and what did not (Schwaber and Sutherland, 2013). In this meeting, transition decision of the potentially shippable product is made. Transition decision pursues the balance between product function, cost, quality requirements, and time commitment (Schwaber, 2004). By the transition, the delivery is embodied to product and thus the system's incremental growth is ensured. Sprint retrospective occurs after the sprint review and prior to the next sprint planning to inspect the development team itself and create a plan for improvements to be enacted during the next sprints (Sutherland, 2010).

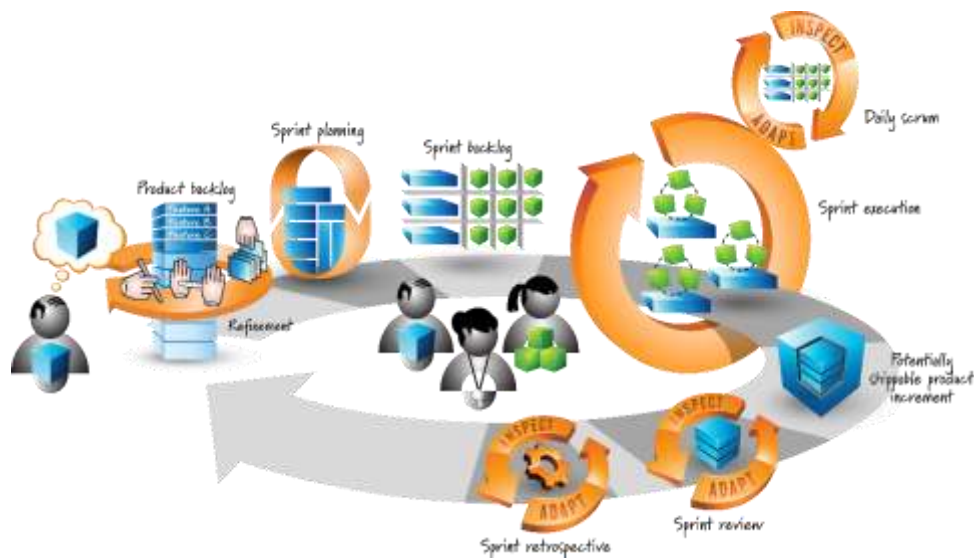


Figure 2 Scrum Activities and Artifacts

Source: Rubin, K. (2012)

There are three primary defined roles in Scrum (Schwaber and Sutherland, 2013). Product owner, one of these, is responsible to ensure the product s/he owns to reach the maximum value (Schwaber and Sutherland, 2013). Product owner is one

person (Schwaber and Sutherland, 2013) and has the final say for the product s/he owns. Product owner can project a committee's requirements to Product Backlog List (PBL), but whoever wants to change the precedence of the items' in PBL shall ask product owner (Schwaber and Sutherland, 2013). This may call for product owners to be a natural member of such committees. His/her position is in between customer and the development team and very close to project sponsors. His/her viewpoint is at again customer's and end user's perspective. S/he is generally selected from customer units and sometimes product owner and the customer can become same person (Sutherland, 2010). S/he ensures the determination of value, size, and risk of development demands and lists the work items via these parameters. S/he records all these demands related to the product in the Product Backlog List (BPL). S/he conveys PBL to the team at the sprint planning meetings held in the beginning of each sprint. S/he supports generally one team, sometimes more than one team (Eckstein, 2010). Mostly, more than one Scrum teams work on the same product (Eckstein, 2010). In this case, only one Product Backlog List is used to describe the works to be performed for the product (Schwaber and Sutherland, 2013). One Product Backlog List can include components regarding more than one product (Kniberg and Skarin, 2010). Development team can work on more than one product within same sprint (Deemer et al. 2010). The result makes it essential to proceed with different virtual combinations between product owners and PBL in the set of product- PBL – product owner (Paasivaara, 2012).

Development team (team) described as the second role in Scrum is responsible for the development of the product potentially ready to use at the end of each sprint (Schwaber and Sutherland, 2013). It is composed of developers amounting from 3 to 9 (Schwaber and Sutherland, 2013). The team is homogeneous to its outer world. Each person in the team is a developer. Scrum designs the development teams as self-organizing teams which have all necessary competence (independent from outer world) (Schwaber and Sutherland, 2013). The team does not accept designation from outside, decides by itself all the works to be performed during sprint and endeavors to make all the best decisions by itself related to its works (Schwaber and Sutherland, 2013). Sprint is the team's area of freedom. The team is the only owner of Sprint Backlog List (SBL) which describes all the work to be performed during a sprint; the list cannot be changed without the team's will. Product Backlog Items selected during a sprint, also, shall not change during the sprint (Schwaber and Sutherland, 2013). Thus, Scrum creates the time slice and the arguments the team can manage with its own will.

With this flow, firstly the customer ensures precedence for his demand in PBL by arguing the "reason" why it shall be developed. Product owner is the determiner of the question "what". S/he decides on the final decisions about PBL, which works shall be prioritized, and the order of the works. The team selects the works which can be completed during a sprint by starting from the top of PBL according to priorities. They reflect details of the selected items at task level on SBL, thus they defines a question "how".

The third of Scrum's defined roles is Scrum Master. Scrum Master is a servant leader for Scrum team (product owner, team, and Scrum Master) (Schwaber and Sutherland, 2013). S/he is responsible for guaranteeing a smooth developing process by

eliminating the obstacles the team may face (Sutherland, 2010). S/he helps the team, product owner, and the organization to understand and apply Scrum properly (Schwaber, 2004). Within this scope, s/he serves as a leader, as a guide, and as a coach for all partners (Schwaber, 2004). By this way s/he protects the team from anti-Scrum approaches (Sutherland, 2010). S/he gets a balancing and protective position for the team against possible aggressive and fast gain desires on the product of customers and product owner. Again for the same reason, product owner and Scrum Master shall not be the same person (Sutherland, 2010).

PBL is a list in which everything needed for a product is listed (Schwaber and Sutherland, 2013). In Scrum model, as long as a product exists, it lives within PBL; new items enter, some are deleted or changed and priorities are set again. Possible work item types in the list are product features required to be developed, product improvement items, bugs, and non-functional requirements. PBI's can be expressed by any method Scrum team prefers. User story is a method to record user needs, which is commonly used by agile methods; but it is not a compulsory practice (Deemer et al. 2010). This method does not provide direct information about the size of the requirement; it is rather a format of describing the need. Epic, differently, is an elaboration argument and corresponds to customer need at the high level which is not elaborated yet (Rubin, 2013). The need at the epic level is elaborated and broken to lower segments by elaborating the needs over time. If the need conveyed by the customer describes directly a behavior of an application, then it is called as feature; and they are mostly in epic size. The theme is used for abstractly grouping related PBIs; and it is not a detailing but a uniting method.

PBL includes description, order, (size) forecast, risk, and value of the items in the list (Schwaber and Sutherland, 2013). Relative abstract sizes (story scores, poker cards, T-shirt size, function score, etc.) or engineering efforts such as person-hour or person- day can be used for the size forecasts. Scrum which stays in the framework level does not impose a special technique and method for this practice (Sutherland, 2010). But it encourages the use of relative sizes (Sutherland, 2010, and Deemer et al. 2010). The team uses these relative size parameters of the works when selecting the works according to their capacity. By this way, any possibility of pursuance, control, and inquisition through tangible engineering efforts by out of the team is destroyed. This encouraged technique, on the other hand, blocks the alternative of proceeding over real effort information for the project as a whole. Size information in sprint tasks, differently from the sizing in PBL, is tangible real effort information. By this way, size information in this form enables the team to adapt to the real world within the sprint.

3.2 Project and Related Concepts

Project: Project is a notion that survives in agile methods. According to PMBOK (Project Management Institute, 2008), project is a temporary effort to fulfil a unique product, service or result. Temporary nature of the projects expresses a definite start and an end (Project Management Institute, 2008). This end is met by finalizing the project when it is understood that the purpose of the project is achieved or that it is

clear that the purpose cannot be achieved or when the need for the project disappears (Project Management Institute, 2008).

According to Scrum, each sprint is like a project (Schwaber and Sutherland, 2013). However, this differs from project definition of the customer who describes it from a holistic and a wider (and scalable) perspective. This Scrum definition, also, includes some application difficulties from its nature: each customer need is a whole and it is, sometimes, not dividable in to a sprint length (let's say 30 days). For instance, imagine dividing the number of 75 by 30 or 400 by 30. Same difficulty prevails for request sizes smaller than a sprint size. As a result, the project, from the customer's perspective free from the applied methodology, is more suitable for the aforementioned general definition provided by PMBOK. Similarly in Scrum world, a project in this sense is finalized when all the work items related to the project is completed or the cost of the next iteration (sprint) is more than the value of the iteration.

By the help of project management, managing of the scope, quality, duration, budget, source and risk limits of the project at an intended balance is ensured (Project Management Institute, 2008). Agile methods, differently from classical methods, provide new approaches for managing this balance as shown in the Figure-3. In this sense, it is obvious that PMBOK which has been used with traditional methods for a long time and Scrum agree on the project definition at the basic level and the main difference is about project management approach.

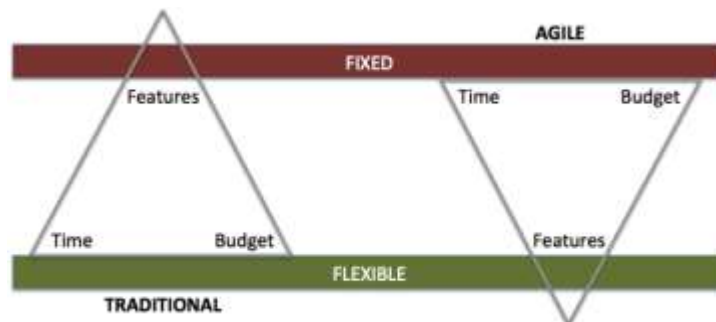


Figure 3 The iron triangle in traditional and agile software development

Source: Morse, L. (2012)

In traditional project management the idea is to fix the scope of the project and focus on controlling the project cost and schedule by controlling changes to scope (Cobb, 2015). A project is deemed successful if the original requirements are met within the budgeted cost and schedule (Cobb, 2015). In agile project management, differently, fixes time and budget of the project during a sprint and allows changes of the requirements during this time of period to reach the goal of the sprint.

Scrum's nature that depends the team structure, meetings and face to face communication on physical platforms manifests itself as a factor inhibiting flexibility of virtuality and accompanying agility. The project may experience a uniting manner as a virtual layer over this physical dependence. In classical structures, while separation of developer functions creates horizontal silos, in Scrum's modular and granular structure

vertical corridors that descent from customer to the team will cause formation of vertical silos and one way or another silos at the end. The project, again, play a uniting role by crossing this vertical silos in the horizontal. Although building Scrum teams in a self-sufficient structure is an advantage, it causes the teams become estranged to outer world and diverged from central designs, structures and formations in time. The project, also, may play a role to minimize this negative effect. And again, because of all these reasons, creating teams specific to project (project teams) provides advantages due to Scrum's recommendation on not changing the team members and not giving multi-tasking to them during a project.

Project Manager: Scrum Guide describes only three primary roles (Sutherland, 2010). Project manager role is not one of them and is not mentioned in the Scrum Guide. However, this does not mean that either project manager role is excluded or it does not exist in Scrum. Because, Scrum usage is not limited to these described three roles, there can be additional roles. This, still, leads up to two different opinions. While one opinion argues that there is no need for project manager role in Scrum and this role's responsibilities are apportioned among development team, product owner, and Scrum Master (Deemer, and Benefield, 2010, Pichler, 2010 and Sutherland, 2010) another opinion argues that project manager role can be in Scrum and it must be when the circumstances call for it (Eckstein, 2010, Sliger and Broderick, 2008).

The argument that project manager shall not be in Scrum, can be explained with three reasons. Scrum is against traditional software development processes and, in addition to this, traditional and deterministic project management approaches (Schwaber, 2004). This requires excluding traditional project management approach at some extent. What is worthy of notice here is the need to separate the classical project management based upon command-control approach and the project management in absolute meaning.

Another reason is the distanced approach of Scrum authorities against using the expression "manager" and the structure it implies. On the other hand, management activities are performed in every situation and even within Scrum's own structure although not carried out by a manager. A case in point is the described activities for product owner to manage the product. Another one is project management itself. Besides, the leveled structure among product owners which is offered by (Sutherland, 2010) and recommended for multiple teams is another case in point. Managers indispensably keep their existence in the organization's all areas beyond Scrum world. Within this context, distinction of manager and "management" functions in such extent does not prevail in practice.

Another reason of the idea that there is no project manager in Scrum is that agile development is originally designed for relatively smaller teams as mentioned by one of the creators of The Agile Manifesto, Dave Thomas (Yvette, 2015), in which there may be no need for a project manager role. On the other hand, due to the benefits agile approaches provide, some companies may need scalable agile structures. The reflex by Scrum authorities as an answer for this requirement has been occurred as applying a multiplying method by its core notions resulting in more teams, more PBLs, more PO's, and etc. A typical example of this is "Scrum of Scrums" structure offered

by Schwaber (2004), and Nexus model (Scrum.org, 2015) introduced years after Scrum's launch. The year of 2015 is a proof which reveals Scrum creators' retarded response to scalable project needs. In the core of all these solutions, achievement of scalable project, portfolio or program management over and centered on project notion, as mentioned in this study, does not seem possible. After all, with these proposed solutions there are currently various limitations and challenges related to scalability bearing on Scrum's nature for the teams scattered in different locations, works that includes broad participant and/or multiple teams, development of large systems, situations consisting an agreement, and the projects that require outsourcing (Akif and Majeed, 2012, Dingsøy and Moe, 2014 and Sutherland, 2010). As a matter of course, a diversity of views occurs between these two edges, one for the Scrum's origin that may not demand a project manager role for small teams and the other for the scalable versions that may possibly include project manager role. It is possible to say that project management responsibilities in relatively small projects can be shared among Scrum's defined roles in a structure which is basically designed for small projects; but a separate project manager role might be beneficial even necessary in practice when it comes to larger projects (Sliger and Broderic, 2008).

When only one team works on a project, what Scrum offers is almost ideal. More than one Scrum team, on the other hand, might work on a project and this makes a unifying role at a higher level essential. Additionally, project management expects adequate team capacities available or team structures (Project Team) built specific to project to be able to supply the resources necessary for the project. This shall be managed from a level beyond and inter the teams. If there is not such a perspective provided at high level, holistic view of products and services may be lost, re-works may occur, and challenges may occur in communication, combination of separately developed pieces, management of dependency between works, and alteration management. Scrum's solutions for this problem, that do not include a project manager role, aim to manage the effect, communication, and coordination with extra teams consisting of representatives of the teams. That creates a separate layer or layers; but with each layer added, on the other hand, it ends up by creating a structure that gradually gets harder to be managed (Rubin, 2013). Additionally, being of product owner product value oriented and his/her position nearer to customer edge, Scrum Master's perspective narrowed to team level and Scrum processes, and such fragmentation of a particular project in terms of Scrum arguments lead to a conclusion that, it is hard to manage, at least for these kind of projects, to the fullest extent by solely Scrum Master and/or product owner roles. Moreover, product owner gives service for more than one customers and establishes connection with the team which consisting of more than one person. Product owner is one person. Thus, s/he represents one of the most fragile points of Scrum structure due to his/her "n-1-n" relationship that arises in this context. S/he manages the "value" which is hard to measure and manage. Product owner role is the toughest role among Scrum roles (Sverrisdottir et al. 2014); and when product owner's duties are increased by assigning him/her project management responsibilities along with existing ones, his/her role becomes from tough to tougher (Eckstein, 2010).

As indicated by the study of Sliger and Broderic (2008) establishing project management structures that keep up with agile culture and even support this culture and the team is possible. Project manager which works in conformity with agile culture:

- Functions as a bridge for Scrum world to open to the world where Scrum does not exist and for the rest of organization's classical structures (Rubin, 2013). Thus, it opens doors for communication between Scrum teams and other organs of the companies who somewhat behaves and have to behave plan-oriented.
- Facilitates and smoothes interim transition phases in the organizations that have long-established traditional work culture.
- Plays a unifying role freely from methodology when Scrum is not applied in each team.
- When third party partners want to keep their own methodology, plays a unifying role above these two parts, Scrum and not Scrum, freely from both (Rubin, 2013).

This role can be undertaken either by Scrum's described roles or by a different person. First option poses a risk of intermingle of the roles in theory (and correspondingly in practice). Hence, it is more appropriate to assign a different person for project manager role.

Program and Portfolio: Program is described as a group of related projects which are managed in coordination in order to ensure benefit and control that are not provided when managed separately (Project Management Institute, 2008). Portfolio is a collection composed of projects, programs, and other works which are gathered in order to achieve strategic work objectives for the purpose of smoothing efficient management (Project Management Institute, 2008). It is important to indicate that program comprises projects while portfolio comprises project, program (multiple projects) or works not included in the project. Program's description is limited to projects. However, portfolio more comprehensively includes all work requirements in any scale. So, its necessity in Scrum world is more than program.

Another reason what makes portfolio management necessary in Scrum: existence of plans that take form according to work requirements and teams' capacity in schedule-oriented classical methodologies creates an intended buffer layer between business environment and development teams. This plan layer smoothes the pressure of unsteady and aggressive dynamism of requirements at a certain level and plays a buffering zone between business and information systems unites. This function of capacity management and planning in agile structures is narrowed to sprint level. Nevertheless, agile structures that are aligned with dynamic work requirements and objectives, and provide proper prioritization methods, capacity management processes and project selection methods are still needed. This can be possible with a systematic portfolio management. For the requirements out of project, PBL and SBL formations with their virtual and flexible structures can be used to enable more movement space for the concrete, static, and limited formations of Scrum such as product and team. However, current designs of Scrum for this issue are on the basis of elaboration

methods of requirements using epic, feature, etc. definitions and not scalable on the basis of projects. On the other side, as is seen, portfolio and program management are basically described and positioned via project notion. In accordance with this purpose, before proceeding with portfolio and program management, the project notion and project manager roles shall be addressed extensively in order to ensure a scalable model for program and portfolio management. However, in Scrum world, as seen, arguments over definition and management of project and the role and responsibilities of project manager still continue. Additionally, agile portfolio and program management are new subjects and they are not mature enough to be a basis for theoretic arguments due to limited examples in practice (Laanti et al. 2015). For these reasons, any solution for scaled portfolio and program management over project notion is not proposed within this study, and the information for these subjects is kept at this level.

4. Discussion

It is a fact that Scrum focuses on product and its continuous delivery. The reflection of this approach on Scrum Guide is the fact of not mentioning the project term in it and giving more space for product related artifacts and roles. Going beyond Scrum Guide, it is seen that a clear definition of project and project manager gets blurred in Scrum. As a clue, project planning cannot find a proper place in Scrum and the horizon in this manner does not go beyond sprint and release planning borders. Similarly, project manager role is engaged to Scrum Master role by the prominent Scrum authorities, even though Scrum Master role is for the pure Scrum processes, and not project oriented. And, design of program and portfolio management similarly stays at the level of requirement management, at management of epics, features and theme. These facts are just among the evidences asserting Scrum is designed for small teams focused on small and continuous flowing of requests for products. As a result, scalable versions aside, Scrum well fits for small bunch of development requests, ideally for “one team - one project” case.

However, more than one Scrum team might work on a single project and this makes Scrum’s “simple to understand, difficult to master” structure simple to understand, more difficult and complicated to master. In such cases, a project may probably include more than one teams (meaning more than one PO’s and Scrum Masters), products, and locations and possibly scattered on more than one PBL’s and SBL’s. The severity of complication gets dipper when it is extended to the program and portfolio management scales.

A possible way to proper solutions should provide a redefinition of project in Scrum from the point of customers and the iron triangle attributes, should evaluate existence of project manager and possible placements of such role with its differences from product owner and Scrum Master, and should provide scalable approaches for program and portfolio notions. This study comes with the idea that the definition of project in Scrum is aligned with the traditional definition of project and suitable to consider and retain the view and boundaries of project definition from customer point

of view. The main differences reveal in the mentality and methods in the management of projects. The study also highlights the contributions spring from the abstraction and ability of connecting of a project. And, the study implies that it is also possible to reach a scalable project management blended with traditional structures by this way. And the work highlights the necessity of well-established project management for the possibilities of scalable program and portfolio management.

Meanwhile, the needs for scaling agile practices increases and naturally creates an effect to shape Scrum and Scrum related processes. Organizations gravitated by the charm of being agile should consider it is still one of the hot topics of agile software development that is originally designed for relatively small teams to apply to scalable sizes. For this reason, some Scrum structures have inevitably led to solutions in the aspects of teams (such as Scrum of Scrums) or product-sub-products (and their relevant PO's) that force the physical limitation of Scrum' core notions. Even for these apparently scalable models, points to take into consideration for scalable project management is that for Scrum structures, weight of holding those physical facts of Scrum will force the flexibility that arises from agility. For the possible solution of the same issue, as an alternative, it is possible to create solutions via the abstraction of project that is the approach presented in this study.

References

1. Agile Scaling, 2015. *Ask: Agile Scaling Knowledge - The Matrix*. Retrieved July 26, 2016, from: <http://www.agilescaling.org/ask-matrix.html>.
2. Akif, R. and Majeed, H., 2012. "Issues and Challenges in Scrum Implementation". *International Journal of Scientific & Engineering Research*, 3 (8)8, 1-4.
3. Cobb, C., G., 2015. *The Project Manager's Guide to Mastering Agile*. Hoboken, New Jersey.
4. COBIT 4.1., 2007. Rolling Meadows, Ill.: IT Governance Institute.
5. Deemer, P., Benefield, G., Larman, C., and Vodde, B., 2010. *The Scrum Primer*.
6. Dingsøyr, T. and Moe, N. B., 2014. "Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda", in T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel and K. Petersen eds., *Agile Methods: Large-Scale Development, Refactoring, Testing, and Estimation*, (vol. 199). Springer Verlag.
7. Eckstein, J., 2010. "Roles and Responsibilities in Feature Teams". In D. Šmite, N. B. Moe, P.J. Ågerfalk, eds., *Agility Across Time and Space: Implementing Agile Methods in Global Software Projects*, (pp. 289-299). Springer, Heidelberg.
8. Kniberg, H. and Skarin, H., 2010. *Kanban and Scrum - making the most of both*.

9. Laanti, M., Sirkiä, R., and Kangas, M., 2015. "Agile Portfolio Management at Finnish Broadcasting Company Yle". *Paper presented at the Scientific Workshop Proceedings of the XP2015*.
10. Morse, L., 2012. *Paradigm Shifts of Agile*. SolutionsIQ. [Online] Available at: <http://www.solutionsiq.com/3-paradigm-shifts-of-agile/>. [Accessed 06 September 2016].
11. Nerur, S., Mahapatra, R. and Mangalaraj, G., 2005. "Challenges of migrating to agile methodologies". *Communications of the ACM*, 48(5), pp. 72-78.
12. Paasivaara, M., Heikkila, V. T., and Lassenius, C., 2012. "Experiences in scaling the product owner role in large-scale globally distributed scrum". *Paper presented at the Seventh International Conference on Global Software Engineering (ICGSE)*.
13. Pichler, R., 2010. *Agile product management with Scrum*. Upper Saddle River, NJ: Addison-Wesley.
14. Project Management Institute, 2008. *PMBOK Guide*. New Town Square, Pennsylvania.
15. Rubin, K., 2012. *Essential Scrum*. Upper Saddle River, NJ: Addison-Wesley.
16. Schwaber, K., 2004. *Agile Project Management with Scrum*. Redmond, Microsoft Press.
17. Sutherland, J. and Ahmad, N., 2011. "How a Traditional Project Manager Transforms to Scrum: PMBOK vs. Scrum". *Presented paper at Agile 2011*.
18. Schwaber, K. and Sutherland, J., 2011. *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. [Online] Available at: <https://www.mitchlacey.com/resources/official-scrum-guide-current-and-past-versions>. [Accessed June 24 2016].
19. Schwaber, K. and Sutherland, J., 2013. *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. [Online] Available at: <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>. [Accessed 18 June 2016]
20. Scrumalliance.org. 2016. *Learn About Scrum*. [Online] Available at: <https://www.scrumalliance.org/why-scrum>. [Accessed 05 July 2016].
21. Scrum.org. 2015. *Nexus Framework*. [Online] Available at: <https://www.scrum.org/Resources/The-Nexus-Guide>. [Accessed 25 July 2016].
22. Sliger, M. and Broderick, S., 2008. *The software project manager's bridge to agility*. Upper Saddle River, NJ: Addison-Wesley.
23. Sutherland, J., 2010. *Scrum handbook*. [Online] Available at: <http://www.scruminc.com/wp-content/uploads/2014/07/The-Scrum-Handbook.pdf>. [Accessed 20 July 2016]
24. Sverrisdottir, H., Ingason, H. and Jonasson, H., 2014. "The Role of the Product Owner in Scrum-comparison between Theory and Practices". *Procedia - Social and Behavioral Sciences*, 119, pp.257-267.
25. Yvette, F., 2015. *Is the Agile Manifesto dead? Not by a long shot*. TechBeacon. [Online] Available at: <http://techbeacon.com/agile-manifesto-dead-not-long-shot>. [Accessed 25 July 2016]