

Large-Scale Agile Frameworks : Dealing With Interdependences

Beatriz MUCAMBE^{1,2}, beatrizjorgeseabra@gmail.com
Anabela TERESO¹, anabelat@dps.uminho.pt
João FARIA¹, joao.faria@algoritmi.uminho.pt
Tiago MATEUS², tiago.mateus@pt.bosch.com

¹ Department of Production and Systems / Centre ALGORITMI
University of Minho, Campus de Azurém, 4804-533 Guimarães, Portugal

² Bosch Car Multimedia Portugal, S.A.
Rua Max Grundig, 35 - Lomar, 4705-820
Braga, Portugal

Abstract

The market environment and customer requirements are changing at an increasing pace. It is crucial that organizations remain competitive by embracing new challenges. These changes require different and more agile leadership, mindset and collaboration, including more team empowerment. With the constant increment of projects and programs, organizations have to adopt new methods to respond to the demands of the business environment. Large-scale agile approaches are the way. In this research, a literature review about Scaled Agile Framework (SAFe), Large-Scale Agile (LeSS) and Scrum of Scrums (SoS) was conducted, and an analysis of the collected material was made.

The primary goal of this research was to understand why large-scale agile methods are used and how do they respond to the interdependences between teams.

In this paper, we conclude that SAFe incorporates an enterprise focus similar to LeSS. LeSS is a framework which implicates extinction of roles, processes and artifacts, giving the product a relevant place. SoS is more like a scale-agile process which applied with other large-scale methods can reach its full potential. Having interdependences in mind, several events and processes described in the large-scale agile methods specified in this article are held in order to address these topics.

Keywords: Project, program, interdependences, agile, large-scale

Introduction

We live in a competitive world. The current business situation is active, with an accelerating speed of change, demanding organizations to focus on a global perspective, competitive awareness, and innovation. To remain competitive in the global economy, it is imperative to embrace different challenges with new approaches combining the integration between projects and focusing on long-term benefits (Blichfeldt & Eskerod, 2008; IPMA, 2015). Customer requests are aggressive and demand a new set of practices that allow quick adaptations from the developers and match the modern product development needs (Papadopoulos, 2015). The iterative and incremental agile methods allow matching deliverables and work flexibility to developers providing the right product to the customer (Serrador & Pinto, 2015).

These approaches are based on observed realities during the project and its iterations. Although the team has to create requirements and designs, such as developing artefacts, testing them, fixing any problems and integrating with other products, the work is done by sprints (Layton & Ostermiller, 2017; Serrador & Pinto, 2015).

Adopting agile methods in software enterprises is valuable in terms of fast and high-value deliveries. It also establishes a work environment that promotes creativity and productivity, rapid adaptation to change and value to the customer because of the improved identification of needs and priorities and

faster multiple deliveries of functionalities (Hobbs & Petit, 2017). In the software development world, the complexity underneath is high due to the volatility of the user's requirements, interoperation with other software systems and interaction with people. Nowadays several frameworks exist to be applied in the field having agile values as a foundation. Scrum, Extreme Programming (XP), Scrumban, Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM) or Kanban, with focus on the team level, and Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS), Nexus, Disciplined Agile Delivery (DaD), Recipes for Agile Governance in the Enterprise (RAGE) and Scrum of Scrums (SoS), with focus on scaling to the upper level. In this paper, we analysed and discussed the following frameworks: SAFe, LeSS and Scrum of Scrums. These three were chosen because they seemed to be the most appropriate to be used in our future research and also because of the availability of literature on these topics.

The objective of this study is to provide collected evidence of how large-scale agile methods can be used and how interdependences are handled using these methods.

Research Methodology

In this paper, we present the results of a literature review about large-scale agile methods application in organizations.

The method used in this research is based on a literature review (LR) and analysis. In this LR study, we focus on research and experience reports about empirical findings on the real use of large-scale agile methods. The search was done in Web-of-knowledge, Scopus, IEE explores, Science Direct and SAGE Journals. The search strings used were: "large-scale agile" or "agile methods" or "agile frameworks", with the filters: software design, software engineering, agile development. A total of 41 papers were retrieved, and a selection was made considering the relevance to the following research questions:

- What are the reasons for using large-scale agile methods?
- How do the methods help to deal with the interdependences between teams?

The selection of papers focused on the ones that clearly describe why large-scale agile methods are used and how they work. We excluded documents that focused on agile methods at a smaller level like SCRUM, XP-Extreme Programming, Kanban and Scrumban, and in a large level, DaD, RAGE and Nexus.

Several other documents were also analysed, namely official websites of the frameworks LeSS, SAFe and Scrum Alliance.

Literature Review

In 2001, a group of software experts created the Agile Manifesto, which brings valuable base principles for successful software development:

"Individuals and interactions over processes and tools,
working software over comprehensive documentation,
customer collaboration over contract negotiation,
responding to change over following a plan"
(Beck et al., 2001).

This means that the human role is the focus, the code should be clear and straightforward, without too many paperwork and bureaucracy. There should be a close relationship between customer and developers and all the participants on the process should be well-informed and know about possible changes throughout the project.

Agile approaches are based on observed realities during the project and its iterations. Although the team has to create requirements and designs, such as developing artefacts, testing them, fixing any problems and integrating with other products, the work is done by sprints (Layton & Ostermiller, 2017; Serrador & Pinto, 2015).

Nowadays, many projects have dependencies between each other. Also, projects can have different teams dedicated to achieve one product vision and deliver in a reasonable time horizon (Ostermiller, 2017; PMI, 2017). When these situations occur, it is required multi-team coordination enabling inter-team collaboration, communication and synchronization. Scaling agile arising from these needs faces challenges such as (Heikkilä, Paasivaara, Lasssenius, Damian, & Engblom, 2017; Ostermiller, 2017):

- Project planning: It is difficult to build a vision with many stakeholders and all scrum teams and also to define a product roadmap, and product backlog with collaborative inputs from all the parts involved requires a different approach;
- Release planning: Releases focuses on specific planning of scope and release timing. The coordination of who will work on what and when requires a clear vision of the dependencies, scope gaps and resource allocation to match the needs of the project;
- Breakdown: To decompose the more extensive requirements in the same backlog, teams need to be involved in understanding what is needed and how to do it;
- Sprint planning: This event defines the amount of work a team will execute. At this stage, dependencies between teams became real, and if they are not considered, the development of the product roadmap can be blocked;
- Daily coordination: In order to have an explicit synchronization it is essential for teams to collaborate each day, without losing the focus on the scope execution;
- Sprint reviews: At this event, each team shows what was done and seeks feedback, which in a larger group of teams can be ineffective. Also, product owners show how the product backlog was updated having in mind what the teams accomplished and how they see what was done;
- Sprint retrospective: It is essential for the teams to know how they identify and implement opportunities for improvement provided from that event;
- Integration: To create a product with all features, it is critical to provide an infrastructure for the teams to build a system and ensure that the work is valid;
- Architecture decisions: It is crucial that someone oversees if the architecture and technical standards are being followed, to control dependencies, keep technical debts in check and also provide tools, so the teams work autonomously and are self-organised.

Scaling agile connects to the management of a portfolio and/or program. Organizations feel the need to manage teams at an upper level (the portfolio/program level). Portfolio management enables the alignment between programs and projects, the achievement of business value and handling of possible risks (Blichfeldt & Eskerod, 2008; Sweetman & Conboy, 2018). At a program level, the benefits are realised incrementally and as soon as possible, governance focus in creating a clear capability, managing how projects overlap and what interdependencies may occur. With agile approaches, the synchronization among customer needs and organizational strategy increases, overcoming the restraining effect of the traditional top-down portfolio method (Sweetman & Conboy, 2018). Adaptiveness is required at the portfolio and program levels due to the instability of requirements specification.

The Frameworks

SAFe - Scaled Agile Framework

SAFe – Scaled Agile Framework, is a framework focused on program management, which incorporates Agile practices and Lean principles at the enterprise level, providing better alignment with customer needs. SAFe integrates Scrum, XP-Extreme Programming, Lean and Product Development Flow (Turetken, Stojanov, & Trienekens, 2017).

Focuses in taking an economic view, applying systems thinking, assuming inconsistency, building incrementally with fast, integrated learning cycles, having base milestones on objective evaluation of operational systems, looking beyond and restraining the work in progress, reducing batch sizes and managing queue lengths, applying cadence; synchronizing with cross-domain planning; cracking the intrinsic motivation of knowledge employees and decentralizing decision making (PMI, 2017).

As described in Figure 1, the framework is divided by levels: team, program, large solution and portfolio.

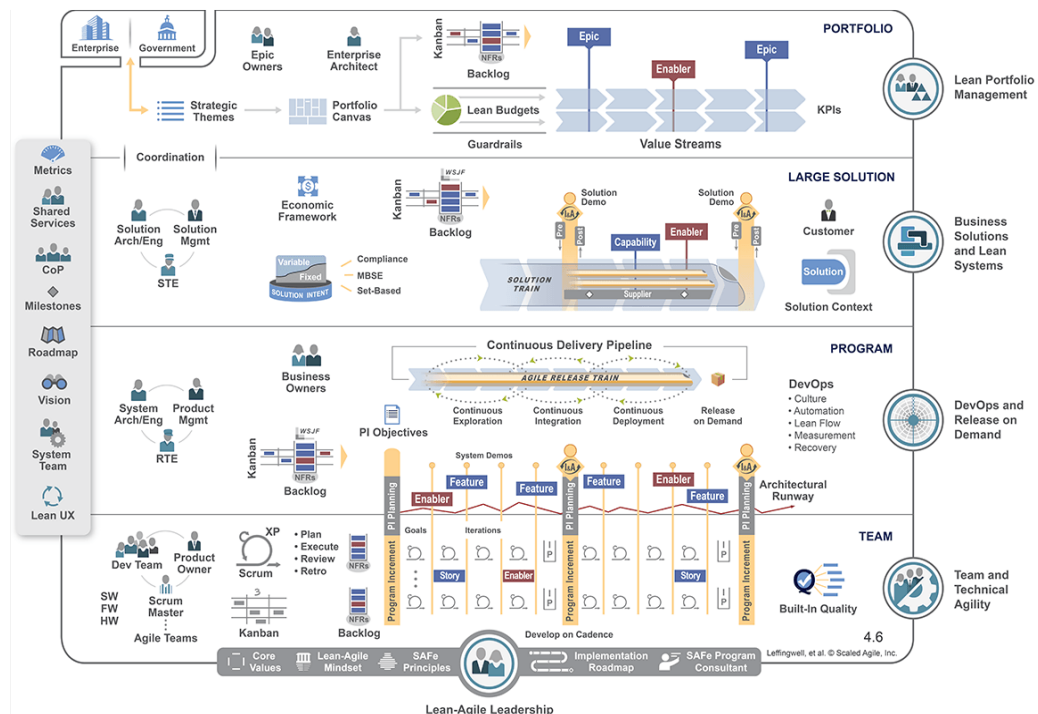


Figure 1 : How SAFe works

(Scaled Agile, 2018)

At the team level, agile teams work using Scrum practices, Kanban and Extreme Programming (XP) (Ebert & Paasivaara, 2017). The concept of user stories borrowed from XP is combined with sprint planning and daily stand-ups. In each iteration, retrospectives and “definition of done” are implemented. The primary objective is to align the work among teams in order to provide better integration. At the program level, development teams, stakeholders and other resources work together to develop a system and optimise the value delivery of requirements (Turetken et al., 2017). Specific roles arise from this framework such as system team, product manager, and release train engineer (RTE) to provide guidance and direction. At this level, also an agile release train is built, which combines several agile teams with a typical business and technology mission. Typically, there are 50 to 125 people that plan, commit, develop and arrange together (Ebert & Paasivaara, 2017). The Large-solution level consists in coordinating agile release trains and stakeholders with the intent of capturing requirements behind the scope and build large-scale solutions which with a single agile release train (ART) would not be able to be developed (Scaled Agile, 2018).

The portfolio level, the highest in the structure, aligns programs with the business strategy and investment funding along value stream boundaries. Value streams are a series of steps used by an organization to set solutions that deliver value to a customer (Scaled Agile, 2018). This level requires a reliable model of governance and management.

LeSS – Large Scale Scrum

LeSS – Large Scale Scrum is a framework that combines Scrum applied in different teams that work together on the same product which can be seen in Figure 2. With a common goal of delivering one common shippable product at the end of a sprint, teams create done items that enclose the final delivery. This approach is about understanding how to apply all the elements and purpose of Scrum in a large-scale context in the simplest way (The LeSS Company B.V., 2018). The core principle is to apply as much as possible Scrum elements which are possible to see in

Table 1.

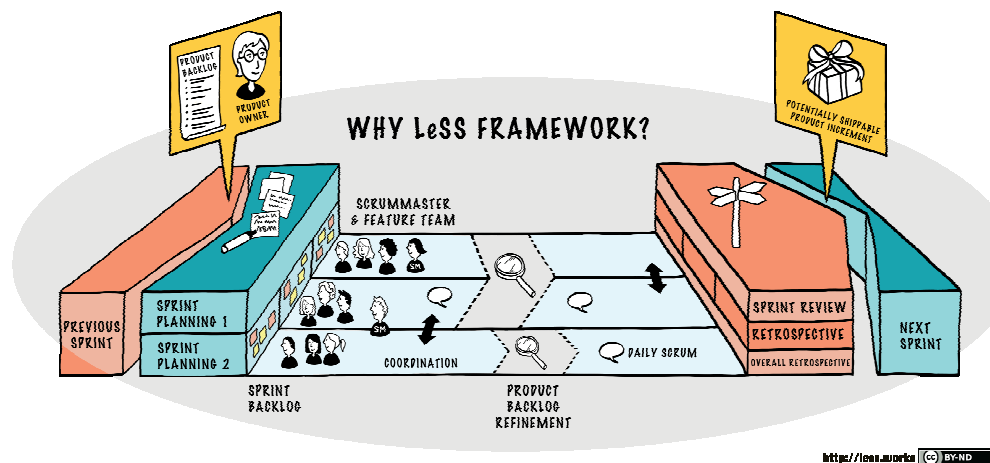


Figure 2 : How LeSS framework works

(Larman & Vodde, 2017)

There is an increase of team’s responsibility, promoting ownership, responsiveness and transparency (Bergman, 2016). There are two types of LeSS: Basic LeSS for two-eight teams and LeSS Huge for more than eight teams. A scrum master can assist 1 to 3 teams. This person ensures guidance to the teams on how to work well with LeSS. Each product owner manages one product backlog which is worked by teams. Features are a significant accumulation of items that describe functionality that is valuable for the customer. These are split into smaller backlog items which can be executed during a sprint.

In this framework, the agile principle “individuals and interactions over processes and tools” is emphasized. During a sprint, the flow of the team (meetings and interactions) is highlighted. In order to have a product backlog with some ready items and a definition of done, an initial product backlog refinement is executed (Larman & Vodde, 2010). In each sprint, an ongoing product backlog refinement is done to adjust items to be ready for future sprints.

Regarding LeSS Huge, it is the application of LeSS but for more than 8 teams. With the customer requests, requirements areas are built. Each team is expert in one requirement area. Although, a requirement has between 4-8 teams. There is one product backlog, in each item, that belongs to a requirement area. This produces an area product backlog managed by an area product owner, which also acts as product owners towards their teams (Larman & Vodde, 2017). The sprint is focused on building a whole integrated product at the end. The synchronization between the product owner and

the area product owner is regular and before the sprint planning, they ensure that teams focus on the most praiseworthy items (Larman & Vodde, 2017). The sprint review allows product-level adjustments.

Table 1 : Comparing LeSS and SCRUM

Similarities of LeSS and Scrum	LeSS techniques added to Scrum
One product backlog	Sprint planning is more formally divided into two parts of what and how
The corresponding Definition of Done for all teams	Organic cross-team coordination
One product owner	Whole cross-team refinement
One sprint	The whole retrospective focused on cross-team developments
One potentially shippable product increment at the end of each sprint	
Complete, cross-functional teams	

(Ostermiller, 2017; PMI, 2017, p. 113)

Scrum of Scrums

Scrum of Scrums is a technique used to scale Scrum up to large groups (more than 12 people), with agile teams of 5-10 people. Each ambassador of different teams joins with others and report results, tasks completed, blockers and highlights on behalf of the teams they represent. In figure 3 the interaction can be seen. This procedure enables the resolution of impediments, the sharing of knowledge and cooperation between teams (Agile Alliance, 2015). The goal of this method is to outside the meeting, relevant individuals volunteer to deal with the operational blockers that were identified by the teams.

Main bases of Scrum of Scrums:

- Meeting once a day after the sub-teams have their Daily Meeting;
- Raise the relevant topics addressed on the Daily Meeting;
- Remove obstacles from the teams;
- If necessary, raise the problems to the next level (enterprise level);
- Work with the enterprise level.

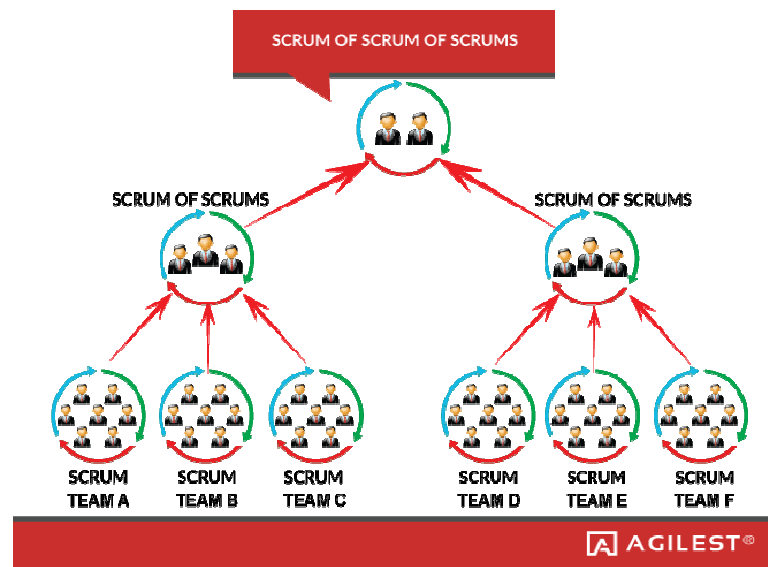


Figure 3 : How Scrum of Scrums works

(AGILEST, 2019)

Ensure that the top-level and down-level of the organization are on the “same page” with the impact of their work as a whole. The frequency of these meetings should be decided by the team, according to the complexity and dimension of the project (Alqurashi & Qureshi, 2014).

For the Scrum of Scrums meeting to be efficient and effective, providing a return of investment to the organization, the representative of each team must have technical expertise regarding the topics and communicate clearly, knowing how to answer all questions, comments or concerns of others.

Findings

The first aim of this study is to investigate how large-scale agile frameworks deal with the interdependences between teams. The findings show that the three approaches studied support differently at two levels: team and enterprise. As a primary finding, LeSS framework is not designed to work with the concept of projects, programs and portfolios.

Team Level

Regarding Scrum of Scrums, the meetings help to facilitate the work and remove blockers.

LeSS allows observers at the daily scrum from other teams to be aware. A representative takes part in the daily scrum and communicates to the team to determine what actions can be made. The product backlog can be divided across teams, and multiple teams can interact with the same system or technology components, giving the chance to collaborate informally on the components area. Promotes multi-team meetings and experts that can coach and mentor teams in their knowledge area. SAFe is built around a teams-of-teams (called Agile Release Train(s)) – ART(s) working together on a single solution, within one value stream. All ART teams have a standard and synchronised cadence (2-3 weeks). The Agile Release Train also has a cadence of typically 8 to 12 weeks - the so-called Program Increment (PI). After this PI, the ART delivers a reasonable increment of value to the customer. This fosters feedback and learning cycles. The heart of SAFe is the so-called PI planning, where, ideally, all teams meet face to face to plan and align the work and objectives for the next PI. At this level, these frameworks rectify the gaps such as communication between teams, increases motivation in teams, and increased flexibility of the teams.

The interdependences between teams are worked in different ways:

- SAFe: several teams work together to build a product. Each team works in sprints and has a scrum master which supports the team with all impediments. At the beginning of each sprint, the teams get together and define the focus and what to do, based on the demand of the program backlog. This is called a Program Increment Planning. All stakeholders meet face-to-face, management sets the mission with minimum possible restraints, requirements and design emerge, important stakeholders decisions are accelerated (Knaster & Leffingwell, 2014). The program backlog has features that in this event are decomposed into stories. In this event, the teams estimate what they will deliver (scope) and highlight their risks and dependences with other teams (Scaled Agile, 2018). Also, Product Owners have sync meetings where they assess how well the ART's is progressing in order to meet the objectives and refine the backlog. At the end of each Program Increment (PI), a system demo (where all features accomplished are integrated) is shown, in an Inspect&Adapt event, where teams reflect, find solutions and take on improvement actions to increase velocity, quality and reliability of the next PI. SAFe holds Scrum of Scrums meeting where the Release Train Engineer (the person responsible for facilitating ART events and processes, also this person establishes a connection with the top-level), Scrum Masters and others show their blockers and highlights, if milestones, program objectives and internal dependencies are being achieved.
- LeSS: One of the LeSS techniques is the adoption of feature teams. These focus on delivering more value and provides organizational-flexibility perspective. Traditionally companies adopt component teams, which develop the maximum number of code lines and are less specialized. The features team contain multiple specializations, enabling the flexibility and customer-centric towards all the feature. This technique enables the resolution of interdependences problems in a product. LeSS contains two sprint plannings. The first focuses on selecting the ready items from those offered by the Product Owner, wrapping up the hefty questions (major questions) and defining the sprint goal. This one is common for all teams while the Sprint Planning Two is done separately. The second is used to create a plan of work to be done for each item (Larman & Vodde, 2017). LeSS works with teams doing different features that are related to the same components. With multi-teams sprint planning two, teams can share design sessions, ask questions to each other, coordinate shared work and find opportunities to work together and learn from each other (Larman & Vodde, 2017). Also, a daily scrum is done by each team, where they clarify what they did and want to do. In this meeting members of other teams can join and observe. As a result if necessary, follow up discussions may occur. Scrum of Scrums can be used in this approach but is not recommended. Can be a sign of unnecessary reliance or coordination problems caused by single-function groups and component teams, or by teams not bright or willing to identify and to share work (The LeSS Company B.V., 2018). Other events like multi-teams meetings of product backlog refinement and design workshops occur. With design workshops, all members of the feature team - people with different skills, modelling related to their upcoming goals or the overall architecture. Also, there is an Inspect and Adapt and Sprint Reviews Bazaar where teams show to the customers and stakeholders and discuss the items developed.
- Scrum of Scrums: This event joins different representants from each team and managers to evaluate how the work is going, if there is an impediment, what was completed and define the next steps. With this, solutions are found in order to solve issues and enables coordination between teams which leads to continuous improvement.

Enterprise Level

Scrum of Scrums enables recognition of the work that is being done, what is needed from the organization to improve, and tools for the workers.

LeSS suggests at the high-level organizational structure, product definition and management of product backlogs when many people are involved. It can implicate rejecting projects, adjusting budget structures and re-aligning the organization around few products. The principle “More with LeSS” is the heart of the approach. Defined as the removal of organizational complexity, this is fewer artifacts, less roles and less processes (Larman & Vodde, 2017).

For SAFe, the portfolio level guides the company in its mission. Focuses on the core strategic decisions that should bring value to the organization. The vision and roadmap for the entire portfolio are defined. At this level, three roles drive portfolio decisions. Lean portfolio management (LPM) is responsible for the alignment of the strategy, investment funding, agile program guidance, and lean governance of the entire portfolio (Scaled Agile, 2018). This person cooperates with many groups across different levels of the organization. Epic owner works with solutions management and product management at the large solution and program levels. Epics are the significant and most long-standing initiatives and drive the business value for the enterprise (Knaster & Leffingwell, 2014). Start as features and are broken down to user stories that can be executed by teams.

At the enterprise level, these methods allow a shorter development lead time of the product, evolving a fast time-to-market and better system alignment.

The critical differences between these methods can be seen in Table 2.

Table 2 : Key differences between SAFe, LeSS and Scrum of Scrums

	How it is applied	Size of the company	Company Level structure	Function	Cost of implementing
SAFe	Sees agile as a whole	Large	Covers the entire enterprise	Retains iteration and agile practices at a team level and adapts another three levels-program, large-solution and portfolio	Little demand for restructure
LeSS	Adapts SCRUM	Midsized (Basic LeSS) to Large (LeSS Huge)	Focus on a product	Adds backlog refinement and retrospective meetings	Demands of rethinking and restructuring of the organization
Scrum of Scrums	Adapts SCRUM	Small	Team Level	After each daily stand-ups of the teams, the SoS meeting occurs	Little demand for restructure

Conclusion and Further Work

Agile methods have become an established reality in software projects and are extending to other areas in organizations (Hobbs & Petit, 2017). The use of these methods requires a significant adaptation at many levels inducing the implementation of mixed approaches.

The goal of this paper is to provide collected evidence of how large-scale agile methods work and how interdependences are handled using these methods. Through a literature review about these topics, it was identified, analysed and compared how these methods work. With this research, it

became clear that SAFe incorporates enterprise focus like LeSS. Although LeSS is a framework for product development, forcing a new organization and displacement of roles. Scrum of Scrums is more likely a scale-agile process which applied with other large-scale methods can reach the full potential. SAFe provides organization for middle management's involvement with agile teams.

As a way to solve interdependences, these agile methods have several events to address these issues, such as plannings, daily meetings, review and retrospectives. In each of these events, depending on the size of the company and complexity, it is possible to deal with interdependences.

With the development of this research, it was clear that LeSS does not work with programmes or projects, focus only in the product, having the customer as the centre. Roles like project manager or program manager are disregard.

Implementing agile is a challenge to the organizations and has to be defined as the culture of the enterprise. In order to agile be effective, the people involved at all levels must be trained and coach about the subject.

Agile enables the focus in continuous improvement, the involvement of the teams, collaboration, build in quality, which the mentioned frameworks enhance. However, our results are based on our perceptions as researchers. In the future, a qualitative and quantitative study about the effects of these methods and adaptations of how they work in organizations would be interesting.

As a limitation, there is a lack of literature about this subject due to the novelty of the topic in the enterprise world. More studies are necessary to understand precisely how large-agile frameworks work in companies.

References

- Agile Alliance. (2015). Scrum of Scrums. Retrieved December 29, 2018, from <https://www.agilealliance.org/glossary/scrum-of-scrums/>
- AGILEST. (2019). Scrum of Scrums - Guide to Agile Scaling Frameworks - Agilest. Retrieved March 18, 2019, from <https://www.agilest.org/scaled-agile/scrum-of-scrums/>
- Alqurashi, S., & Qureshi, M. R. (2014). Scrum of Scrums for Large Size Teams Using Scrum Methodology. *Life Science Journal*, 11(8), 443–449. <https://doi.org/1097-8135>
- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from <http://agilemanifesto.org/>
- Bergman, G. (2016). How to achieve more with LeSS – exclusive interview with Bas Vodde. *Lean Magazine*. Retrieved from <http://leanmagazine.net/scrum/interview-with-bas-vodde/>
- Blichfeldt, B. S., & Eskerod, P. (2008). Project portfolio management - There's more to it than what management enacts. *International Journal of Project Management*, 26(4), 357–365. <https://doi.org/10.1016/j.ijproman.2007.06.004>
- Ebert, C., & Paasivaara, M. (2017). Scaling Agile. *IEE Software*, 34(6), 98–103. <https://doi.org/10.1109/MS.2017.4121226>
- Heikkilä, V. T., Paasivaara, M., Lasssenius, C., Damian, D., & Engblom, C. (2017). Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. *Empirical Software Engineering*, 22(6), 2892–2936. <https://doi.org/10.1007/s10664-016-9491-z>
- Hobbs, B., & Petit, Y. (2017). Agile Methods on Large Projects in Large Organizations. *Project Management Journal*, 48(3), 3–19. <https://doi.org/10.1177/875697281704800301>
- IPMA. (2015). *Individual Competence Baseline for Project, Programme and Portfolio Management*.

Version 4.0. International Project Management Association.

Knaster, R., & Leffingwell, D. (2014). *SAFe® 4.0 Distilled Applying the Scaled Agile Framework® for Lean Software and Systems Engineering*. (Addison-Wes, Ed.). Pearson Education. <https://doi.org/10.1016/j.biortech.2011.05.027>

Larman, C., & Vodde, B. (2010). *Practices for Scaling Lean & Agile Development*. (Addison-Wesley, Ed.) (1st ed.). Boston: Pearson Education.

Larman, C., & Vodde, B. (2017). *Large-Scale Scrum - More with LeSS*. (Addison-Wesley, Ed.) (1st ed., Vol. 84). Indiana: Pearson Education. Retrieved from <https://ir.obihiro.ac.jp/dspace/handle/10322/3933>

Layton, M. C., & Ostermiller, S. J. (2017). Modernizing Project Management. In *Agile Project Management for Dummies* (2nd ed.). John Wiley & Sons, Inc.

Ostermiller, M. C. L. and S. J. (2017). *Agile Project Management For Dummies*. (A. Fandrei, S. Pink, D. Morrow, & A. Sami, Eds.) (2nd ed.). John Wiley & Sons, Inc.

Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. *Procedia - Social and Behavioral Sciences*, 175, 455–463. <https://doi.org/10.1016/j.sbspro.2015.01.1223>

PMI. (2017). *Agile Practice Guide* (1st ed.). Pennsylvania: Project Management Institute.

Scaled Agile. (2018a). Overview of the Scaled Agile Framework ® for Lean Enterprises. Retrieved December 29, 2018, from www.scaledagile.com/resources/safe-whitepaper

Scaled Agile. (2018b). Scaled Agile Framework – SAFe for Lean Enterprises. Retrieved December 29, 2018, from <https://www.scaledagileframework.com/#>

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. <https://doi.org/10.1016/j.ijproman.2015.01.006>

Sweetman, R., & Conboy, K. (2018). Portfolios of Agile Projects : A Complex Adaptive Systems ' Agent Perspective. *Project Management Journal*, 49(6), 18–38. <https://doi.org/10.1177/8756972818802712>

The LeSS Company B.V. (2018). Introduction to LeSS - Large Scale Scrum (LeSS). Retrieved December 29, 2018, from <https://less.works/less/framework/introduction.html>

Turetken, O., Stojanov, I., & Trienekens, J. J. M. (2017). Assessing the adoption level of scaled agile development : a maturity model for Scaled Agile Framework. *Journal of Software: Evolution and Process*, (July 2016), 1–18. <https://doi.org/10.1002/smr>