UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

COMPUTER SCIENCE

Leo Nyholm

SAFe MODEL AND USING IT IN DIFFERENT CIRCUMSTANCES

Master's Thesis in
Computer Science

VAASA 2018

UNIVERSITY OF VAASA
Faculty of technology
Author: Leo Nyholm
Topic of the Master's Thesis: SAFe model and using it in different circumstances
Instructor: Timo Mantere
Degree: Master of Science in Economics and Business Administration
Major: Computer Science
Year of Entering the University: 2011
Year of Completing the Master's Thesis: 2018

ABSTRACT:

As SAFe model is still young method on the industry, this research studies how Scaled Agile Framework (also known as SAFe model) differs from traditional waterfall model and what are the best circumstances to use SAFe model.

The objective of this research is to give a comprehensive overview about the history, the methods and the ideology of SAFe model based on literature and compare it to the practice with interviews in one case project. Through the literature and the interview results that are compared to waterfall model, this research is trying to find the best circumstances for applying SAFe model.

The qualitative research methods were chosen as the research methods for this paper as it aims to create a large-scale understanding of the research objective. The research material is collected from the earlier research about the subject and from the interviews.

As a result, this research found pros and cons from the use of SAFe model. Moreover, the research results give a good perspective how agile method such as SAFe differs from waterfall model and how the differences can be seen in practice while working. Based on the results, the use of SAFe model depends on the current project circumstances.

KEYWORDS: SAFe, Scaled agile framework, agile, software development, waterfall

Table of contents

FIGURES

1 Introduction

This research will be introducing agile developing methods and more precisely Scaled Agile Framework (SAFe model). Purpose of the research is to familiarize reader into agile project work in IT industry, its pros and cons and how it is done in SAFe model. As an example model, SAFe model, is one of the most recent developing models used in IT industry. After theory part SAFe will be compared to more traditional waterfall model and how people that are working with SAFe experience that. This being said, the goal of the research is to introduce Scaled Agile project management, compare it to waterfall model, research how people working in SAFe projects feel about it and what is there still to develop in SAFe model's methods.

Agile could be called a hot topic and trend in IT industry at the moment. Even though there are relatively quite small amount of academic researches made concerning this subject during the past few years. In that manner, there is not a lot of fresh researches out there where to refer. From my point of view, that just makes this project interesting and challenging. I have been working in a project that is following SAFe model for a little less than a year now so it will be interesting to compare own experiences and what literature is saying about SAFe.

This research will be made with following structure. First part will be shortly presenting agile development methods. After that will be basic walkthrough to what SAFe is all about and how it is used at the moment in IT industry. Empirical part of this paper will be made as an interview research where we come to the actual research question of this paper. The research question of this paper will be *how is SAFe model working on circumstances where employees are located in different places with having different culture backgrounds and how does it effect on working that other party is working with waterfall model.*

In the end of the research interview answers will be analyzed and conclusions part will be comparing these interview answers to the theory that has been reviewed in the beginning of the paper. This subject has been chosen for two reasons: first, I am

interested in different ways to manage and organize IT projects. Another reason is that I am at the moment of doing this research working in such project, where there exists this kind of circumstances, that I have at hand in the research question. For that reason, there will be also somewhat of my own thinking based on the experience that I have been able to collect during my working.

In the scope of the research will be basic concept of agile methods shortly. Then more precisely SAFe model, it's pros and cons and how it should be used by the book and in which circumstances. Interview scope will be in using SAFe model in practice in one specific project and how people in different roles of the project experience the SAFe. In the scope will not be other agile methods following models except the SAFe.

As a result of this research I expect to find out that SAFe model is very practical and agile in the exact meaning of the word while it is used properly in proper circumstances. On the other hand, I except that different people in different roles of the project might experience SAFe in different ways and one should always use deep consideration before aligning project based on SAFe model by asking themselves does the upcoming project have the optimistic circumstances and baseline to practice SAFe model.

2 Basics of agile methods

In this chapter I will present what have already been researched about this topic at hand. With this approach, it is easier both for reader and for researcher to dive into the world of agile methods and especially, after this chapter, into the world of *Scaled Agile Framework* (SAFe) model.

So, what are we getting while we are bringing agile methods on the table. First of all, the priority in agile is to satisfy customer by delivering product partially in continuous phase and end of all early, with quality. Sounds legit, does't it. To that goal agile methods are driving throughout continuous refinements of the requirements if needed, working with constant pace all the time, taking into attention technical excellence and good design continuously, keeping things simple as possible and probably the most importantly, always aiming to become more effective by tuning and adjusting teams' behavior accordingly. For following these principles, you need certain type of individuals whom will form the functional group, an agile team. There must be solid, daily working alignment with business people and developers throughout the project. And despite were they business people or developers, they must be highly motivated and trusted to get the job done. Team which can have face-to-face conversations has the best readiness to work together. That way they can share the information the most efficiently and effectively. The most important feature of the agile team and where it all comes down to be able to work in agile project is to be self-organized. To measure progress of this sort of project work is, as simple as it sounds, working software. In agile it is easier to track if project either is or is not fulfilling this, while partial delivering should happen frequently between time periods of a couple of weeks to a couple of months. (Beck, Beedle, van Bennekum, Cockburn, Cunningham, Fowler, Grenning, Highsmith, Hunt, Jeffries, Kern, Marick, Martin, Mellor, Schwaber & Sutherland: 2001, Leffingwell & Renertsen: 2010)

Once company or any group that is working with agile methods has, like usually the case is, faced and overcame the challenges of adapting agile methods the biggest issue is to maintain the focus in key factors and sustain agile way of working. Big factor for

sustaining the agile is to drive for innovation while working. In traditional manner, innovation can be divided into six phases: Initiation, adoption, adaptation, acceptance, routinization and infusion. If one is willing to focus on sustaining the innovation, three last phases of prior mentioned should be considered as a high priority. These make three make sure that the new innovations flow into practice and becomes a part of daily routines. To be innovative, one must control the basics first. That can be achieved with building a strong knowledge in theory of agile methods and adopting the acceptance factors of those. (Mali & Meghann: 2017.)

3 SAFe model

SAFe is a system and software development model which is aiming to organize the whole enterprise to be agile. Model provides guidelines for aligning the work at enterprise portfolio, value stream, program and team level. These all levels, working at these different levels and roles of individuals are presented more precisely in this chapter. All this should give enterprises readiness to increase their productivity 20 – 50 %, quality over 50 % and 30 -75 % faster delivery time to market. And with all this, increase significantly employees' engagement and job satisfaction. But first, lets take a look into short history of SAFe in the next subchapter. (Scaled Agile, Inc.: 2016.)

3.1 History of SAFe

The first version of SAFe, 1.0 (picture 1) was released at 2011 on Scaled agile framework community's website. It was developed by Dean Leffingwell, entrepreneur, executive, author and consulting methodologist. SAFe 1.0 was led from the scratch version made by Leffingwell that can be seen in the following picture (presented in Figure 1).

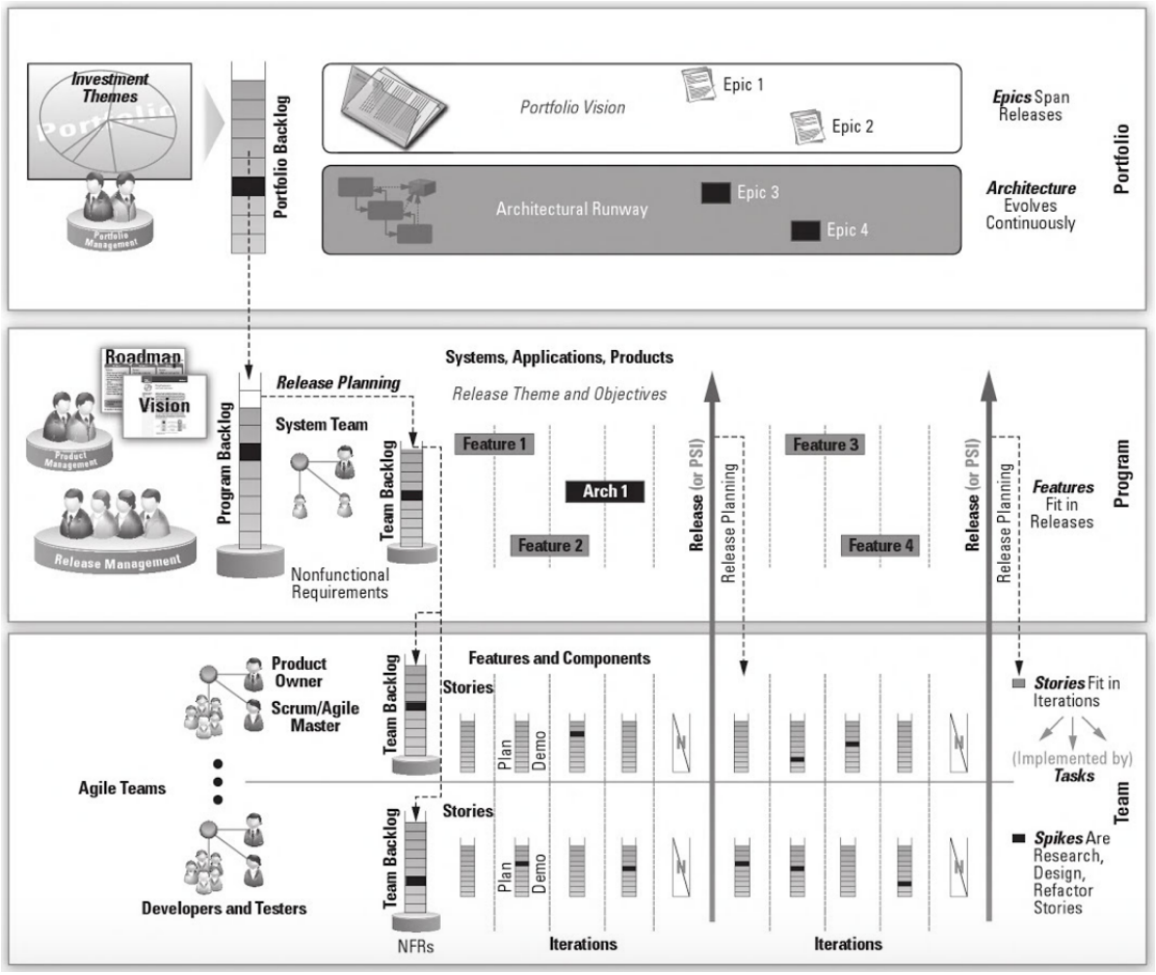


Figure 1: First draft of SAFe model. (Leffingwell: 2011.)

This picture and the base of the SAFe was created in Leffingwell's (2011) book: *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise.* Motivation to write this book came clear in the foreword where it was stated by Don Reinertsen (2011) that 80-85 % projects fail because of the incorrect requirements. This book was introducing new approach for defining requirements away from waterfall's so called "Iron Triangle". With this book Leffingwell wanted to turn focus also on non-functional requirements (NFRs) and balance in software development needs of technical decision makers, end users, system operators, and financial decision makers. And also, notice the key issue, that usually needs of these earlier mentioned actors' will change during development process so requirements would need to follow those changes. These kind changes of needs can be result of different things. In most

cases customer where all the requirements should come from, does not know exactly what they want in the early phase of the development. And even though they know, they can not describe it well enough. Or if they know how to describe the need, they would not describe the real need but rather the proposed solution. On the other hand, customer's competitors' moves affect also to customer's needs. To answer this call, to have better requirements, Leffingwell wanted to take best practices from three different approaches to define software requirements and unified these as one framework. These three were old school management practices, agile methods, and lean product development. That is how the journey of SAFe got it's beginning. (Leffingwell: 2011.)

3.2 SAFe values

SAFe has four core values which need to realize in daily working to make this framework work, as it should be. In SAFe community, people are talking about "House of Lean" while referring to these values which are: Respect for people and culture, Flow, Innovation and Relentless improvement. These values are aiming to achieve best possible quality and value to people and society, high morale, safety and customer delight. I think Sam Walton, a famous American businessman, has said it all when he stated: "*There is only one boss. The customer. And he can fire everybody in the company*". (Scaled Agile, Inc.: 2016.)

So where do these values come from and how to live by them. First, people do all the work and they are the ones defining the enterprise culture. So, *respect for people and culture*. Customer(s) is/are always also people. Do not overload them or make them wait. Do not force them to do wasteful work or impose them with wishful thinking – just be honest. This comes with long-term partnership which is based on equal both sided trust. And the culture, will change last, not first. If you want to change it, you need to change the organization. (Scaled Agile, Inc.: 2016.)

What about flow? It can be understood in so many ways. But in SAFe the flow comes with continuousness of value creation. This happens by avoiding different starts and

stops during project and delays for those, building only quality, really understand the variability of the project, take it as a possibility rather than threat, and manage it accordingly. Frequent integration between teams and different components and informing decision-making via fast feedback comes pretty much hand in hand and are important part of keeping the flow on-going. (Scaled Agile, Inc.: 2016.)

One does not simply build quality software without innovation. So no wonder it is one of the values of the SAFe. And the innovation should come from producers, customers are only validating what have been done based on their need. To achieve that, you need to think outside of the box, experience things outside of the office and get creative. And when you figure something does not work and figure out something better, do not be afraid to amend to it. Pivoting without mercy and guilt will make all the innovation count. (Scaled Agile, Inc.: 2016.)

Some wise man once said that improvement stops at satisfaction. It might be little exaggerated but still, relentless improvement is key factor for quality regarding SAFe values. There should be constant sense of danger so that teams are performing as their best capacity and capability. The whole project should be optimized, facts considered carefully but followed by quick actions. And if there are some impediments occurring, find the root causes of those and solve them by using Lean tools. Relentless improvement is easier to achieve and measure by having key milestones frequent enough. That way project can identify and address to that kind of shortcomings in their work. (Scaled Agile, Inc.: 2016.)

Following these values and putting everybody on the agile mindset is obviously starting from leadership. Change should be led by managers and they should know the way to encourage life-long learning cycle. Leaders need to be inspired to develop people also by themselves. Managers need to clarify the mission and inspire to aim for that, and align project to it. Leaders should also encourage towards centralized decision-making. That decreases delays and motivates people with added responsibility. All in all, leadership should be based on developing, motivating and inspiring people and minimizing constraints along the way towards the mission. (Scaled Agile, Inc.: 2016.)

3.3 SAFe principles

These following principles will build core of SAFe. We could also speak about guidelines of SAFe. These principles are actions that one should focus on while building software with SAFe model.

- *Take an economic view*

With this we do not mean that one should always calculate every decision's value in money or something like that. But every member of the SAFe train (as we speak while referring to project based on SAFe) from developers to project management should switch on their business sense and think all the decision also from economic point of view, not only technologic. Especially workers on team level whom in most cases does not see the big picture and not from economic point of view. There are five main factors that should be considered to make decision-making economic. Four first of these factors, which are related to each other are cycle time, costs, value, and development expense of each decision. The fifth element in taking economic view in decision making is noticing risks of each decision and action followed that decision.

- *Apply systems thinking*

This second principle is emphasizing that everybody in the train should consider all tasks as a part of the Value Stream. In other words, everybody should have good understanding on how the value is created from customer's request to building it. Every process between those and why and how those processes are done.

- *Assume variability; preserve options*

Third principle refers mostly to adjustable requirements. In most cases, it is not possible to know everything at the start of the work so requirements need to be adjustable to new findings during building the software. In this principle also the mindset is again in key position. Like Allen ward stated: "*Aggressively evaluate alternatives. Converge specification and solution set.*"

- *Build incrementally with fast, integrated learning cycles*

As we can see in the picture 2, already in early phases of SAFe, the work is divided in increments. That is so for having short iterations for fast learning cycles. In all of these iterations the work is done in plan-do-check-adjust cycle. That is to deliver and learn all the time on all levels (portfolio, program and team).

- *Base milestones on objective evaluation of working systems*

As we speak of milestones in this case we could refer for example on specific day that our requirements are complete or on this day the design is complete. But in those cases, referring to SAFe, we would most probably force decisions in too early phase, so being make wrong decisions, and then be forced to come back to those in later phase. As an alternative for that SAFe encourages to assume that you can build the software right at the first time with optimum solution if you have objective milestones which enables learning cycles and continuous adjustments during building.

- *Visualize and limit WIP, reduce batch sizes, and manage queue lengths*

With principle number six we come to kind of 101 of the SAFe. Work should be aligned so that queue lengths stay feasible with fast processing time. That is because with SAFe all comes down to delivering fast and delivering quality. With long queue times that is not possible. We can have our dose of mathematics in this paper in this case. To make principle six reality one should understand Little's Law which is: *Average wait time = average queue length / average processing time.* So in English, what faster is the processing time, the effect of that can be seen as decreased waiting time. And to that we come with short queue lengths. That happens effectively by having batches small enough on the processing at the same time. Right size of batch can be defined by estimating team's capacity and workload of every batch that will go on implementation for the team. But to this how to estimate the work we come back more precisely later on the paper.

- *Apply cadence, synchronize with cross-domain planning*

SAFe leans on cadence-based planning which is done together with the whole train. With cadence-based planning SAFe is referring in this case to planning routine which is

predictive and made for certain length intervals every time. That way this kind of flow-based system is able to work with its basic rhythm in two phases - research and then, develop. That is done to limit variability of the work to a single interval.

- *Unlock the intrinsic motivation of knowledge workers*

Unlocking the hiding motivation comes a lot from example of leaders. In SAFe also the goal is with the whole framework to make working culture and circumstances to be such that it motivates and gives tools for workers to perform at their best level.

- *Decentralize decision-making*

The last principle was brought up on the values in the previous chapter already. This is also the one way to make flow continuous and interrupted. With this principle decision-making is made possible also, when needed, on lower levels and so unnecessary delays can be avoided. (Humble, Jez and David Farley: 2010, Scaled Agile, Inc.: 2016.)

3.4 Structure of SAFe

In this chapter I will walkthrough on SAFe structure by presenting different levels of it and what kind of working methods and roles those levels consist. There are also different working iterations and specific phases inside those iterations where in the work is divided which will be introduced in this chapter. At first we could take a look into picture 3 on the next page, which shows the latest version of SAFe, SAFe 4.0.

3.5 Portfolio level

Portfolio level is the highest level of SAFe. On portfolio level are made decision concerning the big picture of the work to be done. It is the next step from enterprise level to bring common enterprise strategic themes to practice. Like it can be seen from the picture 3, decision makers on portfolio level are program portfolio management,

epic owners and enterprise architect. Portfolio level decisions are based on strategic themes which come from enterprise level and those work as a bridge to create business objectives for each portfolio that are aligned with enterprise business strategy. Each portfolio consists either one or more value streams which are coordinated by portfolio. While speaking about value stream in SAFe, I am referring to building a solution that is delivering value for the enterprise. I will dig deeper to defining value stream in next subchapter. Then there are also epics on portfolio level. Those can be two of a kind, business epics or enabler epics. Both are smaller elements of bigger entity, portfolio, helping it to reach its goals. Business epics are directly delivering business value to the portfolio. Then again enabler epics are the ones that are supporting future business epics, doing preliminary work for those to enable straightforward implementing for business epics. Enabler are mostly created from architectural point of view. Both of these earlier mentioned epics are mapped into a Portfolio Kanban board where are defined that in which phase of progress each epic is. For example, is it in analysis, implementation or done. (Leffingwell, Dean Foreword by Renertsen, Don: 2010, Scaled Agile, Inc.: 2016.)

Responsibility regarding decisions and management on portfolio level lays on three different actors. As mentioned earlier those actors are Program portfolio management (PPM), Epic owners and Enterprise architects. PPM is providing activities and governance for the portfolio. In more specific level that means decisions on investments, returns and what gets built and what is not in the scope. It is also equally important to be able to draw the line on what is not relevant so that train keeps on its trail to the right direction. As a PPM should be person who understands the enterprise strategy well and how to handle investment funding, program management and governance. PPM needs to also be able to decide how to combine that business knowledge and strategic goals to the right technology methods and tools. PPM gets assisted in most cases from Program management office (PMO) to share the load with program execution guidance and governance. Naturally when in every value stream the goal is to create value with the end product, the budgeting is a major issue. That being said, budgeting is one of the key responsibilities of PPM (with assistance of PMO). In SAFe, that is done with lean-agile approach; *Beyond project cost accounting.* The goal in that approach is to provide fast, empowered decision-making based on trust control. Budget should be composed

separately for each value stream and unify those to a whole portfolio budget. Based on that budget value stream managers have pretty free hands to deliver solution the way that it is profitable in economic and business wise. With this process value streams should be able to drive their actions and investments to match enterprise's business priorities. Epic, both business and enablers, should follow those business priorities as well. Portfolio level epics are on responsibility of epic owners and enterprise architects. Epic owners are managing epics and that managing is visible through the highest-level backlog in the framework, the portfolio backlog. To the portfolio backlog end up approved epics by PPM where epics are prioritized and then wait for the implementation accordingly. (Scaled Agile, Inc.: 2016.)

Then again, what comes to Enabler epics, thosef are on responsibility of enterprise architects. In many cases enterprise architects act as epic owner of enabler epics or at least give their recommendations for those. Enterprise architects are providing guidance across value streams and programs in strategic technical issues to make portfolio deliver as it best. (Scaled Agile, Inc.: 2016.)

Value streams are building the base for each SAFe train. Another name to value stream could be flow of value. Those need to be recognized to have capability to understand, organize and at the end, deliver value with the provided solution. Value stream is providing with its work the flow of continuously delivering value to the customer. It consists series of steps that Enterprise is taking to provide that continuous flow. "*A value stream is a long-lived series of steps used to deliver value, from concept or customer order to delivery of a tangible result for the Customer"* (Scaled Agile, Inc.: 2016). Value streams are basically divided in two categories. Those ones that deliver value directly, and those which support other value streams. (Leffingwell & Renertsen: 2010.)

Identifying and organizing value streams is not that simple as one could think. Nevertheless, it is always needed in SAFe as one of the most critical skills of the lean-agile enterprise for many reasons. Value streams provide benefits such as faster learning, shorter time to market, higher quality and productivity and at the end solutions that

serve the intended purpose better. First step is always for PPM to understand the value streams. After that SAFe agile release trains (ARTs) can be organized based on those. Value stream can only match it's goal, delivering value to customer, if it can provide beneficial new solution or capability. That is achieved only if enterprise, portfolio and all the other levels in SAFe train really understand the flow of value, what it delivers and how it delivers it. (Scaled Agile, Inc: 2016.)

3.6 Program level

Next level from portfolio level in SAFe is program level. That is where the development work is organized to teams and other resources. Program level includes ARTs which are teams of agile teams that are implementing the solution or capability itself, in other words delivering continuous flow of incremental releases of value. To describe program level in a nut shell; It is *long-lived, self-organized and flow-based entity fulfilling the SAFe portfolio.* (Scaled Agile, Inc.: 2016.)

3.6.1 Agile release train (ART)

ART is on program level functioning unit which is delivering those earlier mentioned steps of the value stream. It is also possible that one ART delivers the whole value stream. To ensure to keep the flow continuousness, work is divided in program increments (PIs) which have length of 8 to 12 weeks. One ART consists from 5 to even 12 teams which are responsible of delivering tested, functioning system-level software. As mentioned earlier, on program level, ART's responsibility is to fulfill the SAFe portfolios vision. That happens in practice by discovering, defining and developing features and enablers which portfolio has planned and then visualized in portfolio Kanban board. To manage this work on program level, business epics and enabler epics are divided in smaller pieces, to business features and enabler features which are then again managed in a program Kanban board. Features are terms to describe solutions to customers. Features are first analyzed and made then fit to PI time boundaries. Optimal scope of features is such, that it provides new functionality and so being, delivers value

in while it is implemented during one PI. In program Kanban board features are prioritized and they have been enriched with acceptance criterions so, that it supports implementation and testing. (Humble, Jez and David Farley: 2010, Scaled Agile, Inc.: 2016.)

3.6.2 Key roles on the program level

Key element of ART is a nature of teams being based on self-management and self-organized working tightly together. Even though, teams need guidance in keeping the common mission in mind and operating based on same technical architecture and also, providing solutions that are providing mutual user experience for the end user. For those issues, Release train engineer (RTE), System architect/engineer and Product management are providing guidance. These three actors are top responsible of ART fulfilling portfolio's vision. In practice, they are ensuring that teams are aligned towards the right direction and tackling impediments that are blocking teams in implementation. To be more precise, responsibilities of RTE, Product management and System architect/engineers are as follows: RTE is taking care of program execution. He or she is servant leader of the ART and is optimizing the flow of value through the program. That happens for example managing work through program Kanban, helping teams to learn with inspect & adapt workshops and facilitating PI planning sessions which are to be held inside ART, to plan the work to be done for every PI. Product management is managing the content. He or she works between customer and ART to ensure that all are aligned with same understanding of customer's needs and features are defined based on those needs. Program backlog is under program management's responsibility. Then there is the system architect/engineer who is responsible of used technologies in solution. System thinking is highly appreciated in this role, which can be also assembled of more than one person. The most important task for them is to define the overall architecture for the solution. They are also supporting in defining non-functional requirements and interfaces and how those interfaces are integrated. System architect/engineers also define the key elements and subsystems of the system such as used databases. (Leffingwell & Renertsen: 2010, Scaled Agile, Inc.: 2016.)

3.6.3 Other roles on program level

There are several other roles to consider on program level besides these which were brought up in previous chapter. For every ART, there need to be persons to have the responsibility of steering the train to the right direction by participating in planning, helping to block impediments, speaking with the mouth of the development, the business and the customer. For that there are group of business owners in each SAFe train, which will consist of 3-5 persons. They are in key role to help train to deliver value. Maybe the biggest responsibility that business owners has, is assigning business value to PI objectives and after planning, approving the PI plan. But they also participate the work after planning mostly with developing and supporting methods that teams use and act as mentor to those self-organized and managed agile teams.

One of the key figures of SAFe train is to keep delivering value with continuous flow. That is ensured with proper development and deployment operations, which are taken care in SAFe by DevOps team. DevOps is maintaining the readiness of the train to deploy and lead it to production more frequently through the value stream. Key element for getting to that goal is to keep delivery batches comparatively small. This is something to handle by DevOps. But DevOps is also so much more than just a set of people working on deployment operations or their practices in that work. DevOps is also manifested as attitude, mind-set which is driving people in the train to work together with proper methods and tools, give and receive feedback to develop own and other's skills, and also to make work flow so that it brings the best possible solutions to the customer and gives economically good outcomes. Usually DevOps team consists of system and/or database administrators, operational engineers with network and storage engineers. DevOps is a part of ART so it needs actively participate to ART's events and communicate and work with agile teams as well as with system and solution architects/engineers and business owners. (Kim, Gene, et al: 2013)

System team is created to integrate different parts of the solution into one whole entity on daily or even hourly basis. There might be several teams developing on several different parts of the end product. So, that ART can deliver value through the work flow,

those different parts need to be integrated together and to the system and it need to be demonstrated for ART itself and for the customer. System team is also responsible of solution's end-to-end testing.

For assisting in planning, managing and governing releases of the solution with given authority and with that, also responsibility of helping to guide the Value stream to achieve business goals, there is Release Management. Release management has responsibilities both inside and outside of the ART. It actively communicates between ART, external stakeholders and customer. Internally the work of release management is mostly ensuring that release will be as it is required in business goals and how ART will work according to those goals. It also works as a coordinator and communication channel between program level to portfolio level and mostly to Program portfolio management. Externally then, release management communicates to customer and other stakeholders to marketing the release to them and providing the last authorization for it. Release management can consist of different kind of people from different areas. There can be engineers, business owners, solution and product managers, sales and marketing representatives, teams' development managers, personnel from internal IT, production or deployment, solution-level personnel how are ensuring quality, performance and user experience, and lastly, there can be architects to visible and organize architectural integrity. (Scaled Agile, Inc.: 2016.)

Even though shared services is necessary role for ART or Value stream to success, it is not full time job usually. These are helping, part time resources which are used when needed. These can be almost any kind of support that ART could possibly need during its journey. Good example of that kind could be quality assurance, when ART wants to improve specification quality, they hire specialist(s) to guide analysts for writing specifications. It can also be something that usually every train will need but only temporarily at some point. For example, while taking solution to the production, end-users should be trained, so ART needs to hire end-user trainer to do that.

User interface (UI) is always in SAFe implemented by agile teams. In order that teams are driving towards same goals in that manner they need to have straightforward

guidelines for desired user experience (UX). UX team is responsible of taking care of design guidelines, prototypes of the system, wireframes, style sheets and that kind of visual and architectural guidance for the teams. In practice UX team is doing that with firm co-operation with stakeholders to understand business goals that are achieved with human and computer interaction in train's end solution and providing guidelines for teams and across the program based on that. They are also validating UX through the value stream with UX testing and supporting engineers and the system team while they are doing the UX testing. UX team is leading workshops regarding UI. Usually those workshops handles either some guidance or teaching teams some new features in UI or teams are getting aligned with new features with UX team how they should take it to implementation. In the end, basically all UI related work is taken through UX team. (Scaled Agile, Inc.: 2016.)

3.6.4 Vision

Vision is something that comes to program from upper level, from portfolio level. Then again ART vision might differ somewhat a lot from the portfolio's vision, since portfolio level is looking things from wider perspective. Vision is defining the future developed solution based on customer's and other stakeholder's needs. Envisioned solution is to be such, that it has those features and capabilities, that answer to those earlier mentioned needs. Vision is also answering questions such as what are we doing and why are we doing it. It gives a purpose for the solution and base for workers to get themselves inspired from it. Vision as well as it should explain the work and give people motivation to do it, it also gives strategic perspective. For example, it could give state of mind that in each decision one does regarding ART's work, him/her should think it from the customer satisfaction point of view. How does this decision effect on end user's satisfaction while he/she uses the system? Vision on program level is more precise that it is on portfolio level. It can consist even feature specific information. One part of the program vision is the roadmap described on the next chapter. (Scaled Agile, Inc.: 2016.)

3.6.5 Roadmap

Roadmap is a tool for SAFe program to schedule its near future on a timeline. Usually it does not describe nearly the whole project especially when there is a very big project at hand. Roadmap consists of about six months' future deliverables of Value stream and ART. It gives visibility with high confidence of the on-going PI and with little less confidence forecast of the next PI and maybe even one after that. Solution and product management is responsible of composing roadmap. It should always be updated as well, while circumstances change in the train. Roadmap is a good tool for keeping the whole train updated what are the hot topics of current PI and what is coming in the near future. (Scaled Agile, Inc.: 2016.)

3.6.6 Milestones

Roadmap is structured based on milestones, since every element on roadmap is a milestone. There can be two kinds of milestones. Either learning milestones defined by the teams or date milestones usually driven by events, which are not tied to teams' doings. So that being the case, we could state, that milestones are for emphasizing development progress in timewise as well as risks involved in development. Usually those learning milestones are referred also as PI milestones, which are occurring on cadence so that ART has milestones for each PI. Date milestones are always dependent on someone else than teams' work and not so easily to predict so those are also known as fixed-date milestones. Modern softwares are very complex, demand lot of co-operation with third parties and the implementation of systems rely many times on external resources such as other projects or companies. Milestones dependent on that kind of factors are fixed-date milestones. As mentioned earlier milestones are there for the whole train to follow-up the on-going and up-coming work. Another important motivation for milestones is money. Money in that sense, that behind every delivered solution by ART there should be a business benefit which again should bring straightly economic value for the customer. For making all this happen, customer and sometimes other third parties are investing to the project a lot of money and naturally they want some security during the ART's journey that project is really making things happen and

delivering value continuously as it should be in SAFe. So milestones are good way to prove investors that they are making progress, and investors will not be just blindly waiting for the end solution which can take even several years in big projects. (Leffingwell & Renertsen: 2010, Scaled Agile, Inc.: 2016.)

3.6.7 Release

As mentioned in earlier chapters the basic idea of SAFe is to deliver continuously value which is possible only if train can provide often small parts of the solution. At those times train is releasing value to the customer. Releasing on frequent cadence is necessary especially in big and complex systems which has many elements in it. Basic structure of building the final release in SAFe consists of four layers which are team increment, system increment, solution increment and finally releasable solution. This layering is visualized in picture 4 below.

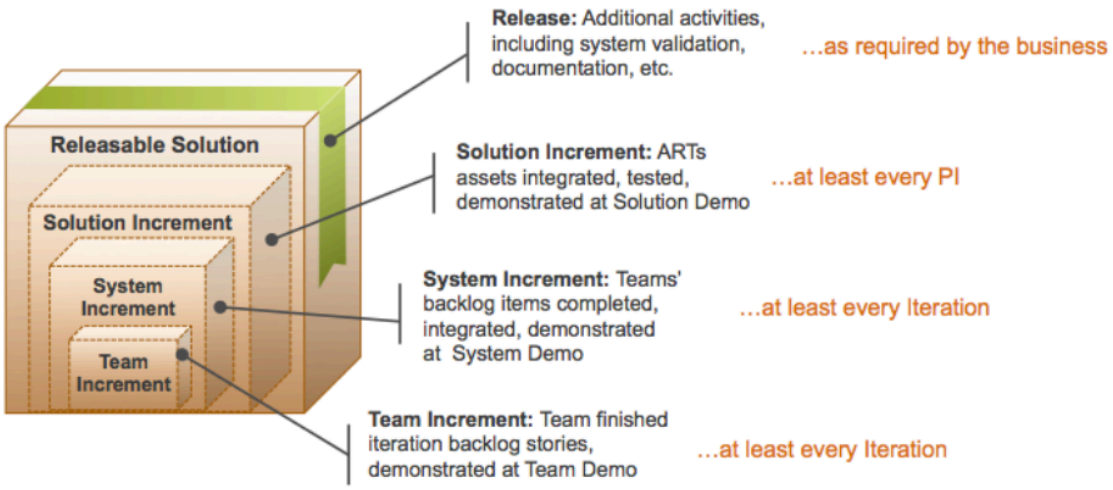


Figure 2: Release structure of SAFe model. (Scaled Agile, Inc.: 2016.)

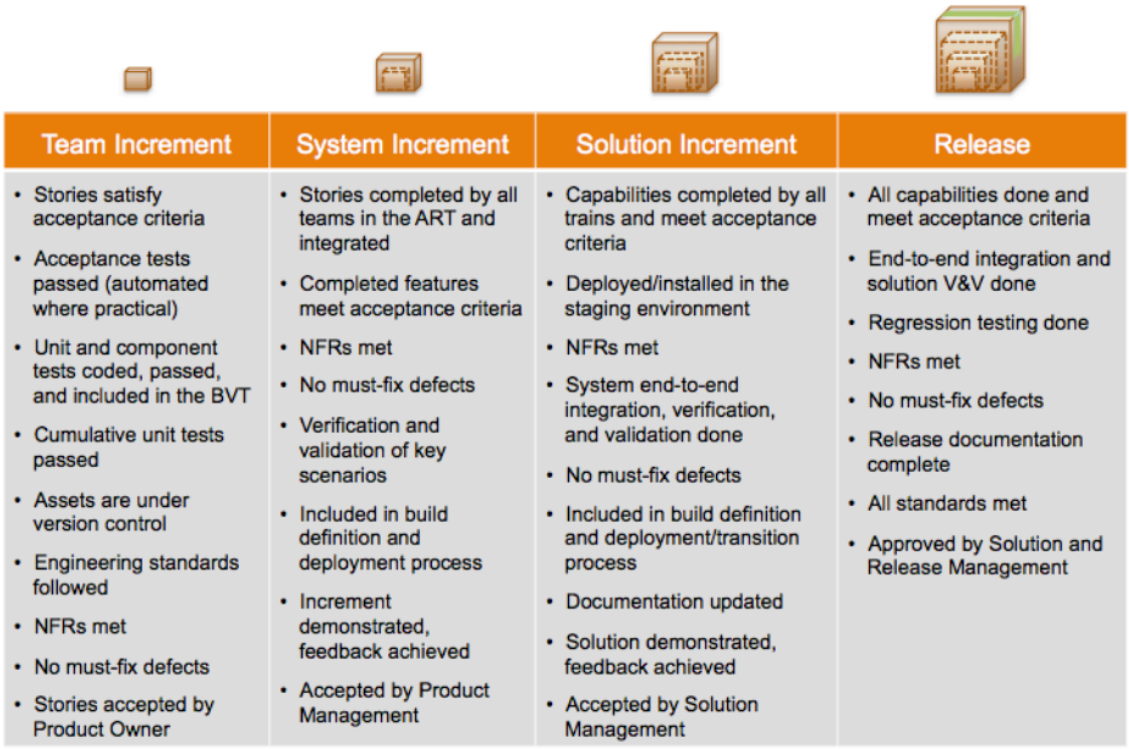
In team increments teams are developing stories based on their team backlog. Stories are developed to meet their requirements and then demoed in *team demo* when ready. On system increment phase features from all teams are added to the system by one ready story at the time and that way those are integrated to the system on continuous

flow. New features of the system are demonstrated in *system demo*. This happens every two weeks. Then again while going forward in development the next step is solution increment where system increments are combined and there should be working, integrated and fully tested system or at least a part of that system. This entity is demoed in *solution demo* and the frequency of solution increment is at least and usually exactly one PI at the end of the PI. After enough of these first three steps have been taken, ART should reach its final destination, releasable solution. At that point solution development has proceeded with each little part on time and all the solution increment combined to that point, where solution consists every story, feature, capability and non-functional requirement fulfilled. In this final phase of building release, there might be still some additional activities to be done such as solution verification and validation, documentation, and some other supporting activities also usually occurs at this point, as we are talking about complex systems. This final release process can also include some transition actions, for example to integrating solution to another ART's solution. Or on the other hand, released solution can be released as it has been build but the delivery itself is so complex that it needs to be handled by another party or same enterprise is taking care of the delivery as its own project, because of the massive efforts that it might take. (Humble, Jez and David Farley: 2010, Scaled Agile, Inc.: 2016.)

3.6.8 Definition of done

All, partial and final releases of SAFe train are based on certain requirements. There are also equal scaled definitions of done which can be applied to all releases to define, that release meets its expectations.

These definitions of done (DoD) have been defined separately for all release levels in SAFe as one can see in picture 5. In practice these DoDs should not be taken that literately as they have been given, but rather to take those of DoDs that are able to be applied to story, feature or epic at hand. One thing also to consider here is that for each increment, there is DoD that implementations meets acceptance criteria. So in acceptance criteria will always be included case specific criteria for each implemented issue. (Scaled Agile, Inc.: 2016.)



Team Increment	System Increment	Solution Increment	Release
• Stories satisfy acceptance criteria • Acceptance tests passed (automated where practical) • Unit and component tests coded, passed, and included in the BVT • Cumulative unit tests passed • Assets are under version control • Engineering standards followed • NFRs met • No must-fix defects • Stories accepted by Product Owner	• Stories completed by all teams in the ART and integrated • Completed features meet acceptance criteria • NFRs met • No must-fix defects • Verification and validation of key scenarios • Included in build definition and deployment process • Increment demonstrated, feedback achieved • Accepted by Product Management	• Capabilities completed by all trains and meet acceptance criteria • Deployed/installed in the staging environment • NFRs met • System end-to-end integration, verification, and validation done • No must-fix defects • Included in build definition and deployment/transition process • Documentation updated • Solution demonstrated, feedback achieved • Accepted by Solution Management	• All capabilities done and meet acceptance criteria • End-to-end integration and solution V&V done • Regression testing done • NFRs met • No must-fix defects • Release documentation complete • All standards met • Approved by Solution and Release Management

Figure 3: Definition of Done by SAFe model. (Scaled Agile, Inc.:2016.)

3.6.9 Demos

Function for demos is to demonstrate the developed tasks for relevant audience. As mentioned in release chapter there are certain demos for each increment release. For team increment demonstration SAFe provides team demo. Team demo is held inside the agile team and the idea is that developers and testers demonstrate the stories implemented to relevant business analysts (BA) and product owners (PO). After demo PO checks if the story meets all DoDs and if so, as a final DoD, accepts the story to be done. Team demo occurs at the end of every iteration, which in this case is usually a sprint of two weeks. (Scaled Agile, Inc.: 2016.)

As increment goes towards the release the next demo is system demo for demonstrating ART's full system for POs, executive sponsors, other teams, development management and for the customers. This is good way for teams to get feedback if they are going to

the right direction in development or how they should improve in development process. As well as team demo, system demo occurs at the end of every iteration, which in this case is usually the sprint of two weeks. Even though, it should be noticed, that system demo is not replacing team demo, but rather it combines the work of all the teams in the ART and demonstrates that to the audience as a full system.

Demo with the widest scope of SAFe partial releases is solution demo. Solution demo is a major learning point for ART based on the given feedback from stakeholders. In this event, the idea is to demonstrate all the done development efforts combined as a working feature of solution which is delivering major value for customer. Solution demo is held at the end of every PI for all the members of ART as well as all the high level stakeholders related to ART. This is one of the most important events during ART's journey since it emphasizes well the progress achieved and defines the overall situation of the ART. (Scaled Agile, Inc.: 2016.)

3.6.10 Inspect & Adapt

Referring to SAFe principle #4 "*Build incrementally with fast, integrated learning cycles*", one of the key elements in working with SAFe is to develop working habits continuously. For that, inspect & adapt (I&A) workshops are critical events to see where are the possible places for improvement. This event is held after every PI and there should be all program level stakeholders participating. The I&A is divided in three parts as follows: first, there is PI system demo. Here is important to notice that this is not the same than system demo held after every sprint. This might be little more formal situation than basic system demo with more high-level business persons in the audience. But anyway, the idea is quite much the same, to demonstrate the full integrated system that is implemented so far. After PI system demo comes quantitative measurement where relevant persons, usually business owners, customers, teams and maybe other stakeholders first of all, look what PI objectives they have been achieved during the PI and what kind of business value those PI objectives have been creating. Business value counted from all planned PI objectives compared to business value from achieved PI objectives gives for ART an achievement percentage which should be between 80 % -

100 % so that train could be described to be a reliable train. After measurement comes the final part of I&A workshop, retrospective and problem-solving. That is settled so that stakeholders can give feedback to themselves and for the whole train, positive and negative. From occurred problems that train might have, they then pick those ones that should be tackled and relevant persons to do that. For bigger problems there can be settled a problem-solving workshop for later time. That is separated event facilitated by the RTE to get to the root cause of the problem and to solve it for good. Goal for this whole I&A workshop is to answer question "what we could do better during future PIs"? As a result, I&A should provide improvement stories to be added to the backlog of future PI planning. This is perfect opportunity for ART to improve in every PI. (Scaled Agile, Inc.: 2016.)

3.7 Team level

Team level is probably the most practical level on the ART where it comes to building the solution. This is where all the "magic" happens and requirements, plans and analysis is implemented as features, capabilities, a system, a solution and at the end of all, as a released solution for the customer. Even though team level is described in SAFe as an own level, it is still basically part of the program level. Actually, it is forming kind of the core of the program level. Function of the team level is to provide framework for teams to organize themselves, what are the roles for each person and what kind of processes they are doing. After features are groomed for the team and added to their team backlog, teams are responsible of defining, building and testing stories based on the features in their backlog. That should be done in iteration cadence defined in SAFe and with co-operation with other teams in the train. That way the system integration is made as easy as possible. (Leffingwell & Renertsen: 2010, Scaled Agile, Inc.: 2016.)

Basic goal for the team level is to build a high quality end product for the customer, which comes piece by piece while team is creating value after every iteration. To achieve that, teams should always emphasize in their practices the *built-in quality* core value of SAFe mentioned more detailed earlier in the chapter 3.2. Agile teams should

also be organized around that goal of creating value continuously. That being said, agile teams should be organized around features and components in to-be solution and that way maximize the velocity of the work. In best case scenario team should be working in firm professional relationship with each other. Also collocation is critical for agile team to work effectively. Teams are five to nine persons strong and roles are organized so, that team is able to define, build and test stories independently. For those tasks, an agile team consists usually of business analysts, developers and testers. Teams are supported from the program level usually as much as needed by RTE, product management, system architects/engineers, system team, shared services and DevOps. Besides them, there are two leading and supporting roles in agile team which are scrum master and product owner. They are both fully team members. Scrum master is often described as a servant leader of the agile team. He or she is one of the team members and person in this role can be also switched for example in change of every PI. Scrum master can also at the same time be shared across 2 to 3 teams. Scrum master's responsibility is to ensure and help team to self-organize and self-manage without issues and to achieve its goals. This is done with using and emphasizing SAFe principles and values and bringing those to practice. Scrum master also facilitates team events such as daily stand ups and represent team in events with wider participant list. He or she communicates actively with other teams' scrum masters to align teams accordingly. Scrum master is in key role to identify together with the team impediments that occurs in implementation and eliminating those or bringing those up on a higher level if needed. (Kim, Gene, et al: 2013, Scaled Agile, Inc.: 2016.)

Product owner (PO) is the one person which has the highest responsibility on teams' working. He or she acts as a customer to answer for developers queries about implementation. In that case, PO need to have solid interaction with customer all the time as he or she acts kind of a translator between team and the customer. PO works in the similar role between product management and team and also participates to plan releases with management. In the implementation itself, PO's key responsibility is to define and accept stories for the team. That comes with owning and managing the team backlog. POs need to make the final decisions on what kind of implementation and how much can team take. Obviously, he or she uses team's opinions and estimates for those

decisions. During the team demo team is kind of selling the stories to PO. Every team has only one PO but one PO can be shared also between two teams. (Scaled Agile, Inc.: 2016.)

As in all levels in SAFe, on team level also the work is organized, managed and prioritized in backlog. Filling of the team backlog falls from the program backlog. That filling is features which are already identified, prioritized, estimated and maintained on program level. On team level backlog features are splitted in to stories which can be then further taken in to implementation. PO leads the creation, prioritizing and management of the team backlog. Like on feature and epic level, also in stories there are in addition to normal user stories also enabler stories to build infrastructure and architectures around stories to make them possible. User stories are written in the basic user story structure, so that it emphasizes the user role (who or what is doing something), activity that he, she or it does (what user can do with the system) and the business value (what value that activity creates for the user). For example, of such user story: "*as a researcher, I can limit the scope of the search with century, so that I can get more time-relevant search results*". After stories have been created, team will estimate the workload that each story takes, so that PO can divide and prioritize the work according to estimates and team's calculated capacity. After team has been with the lead of PO and scrum master planned the iteration, they fully commit to that plan. It is crucial to plan it together and have such end result as a plan, that the whole team can commit to that with 100 %. (Leffingwell & Renertsen: 2010, Scaled Agile, Inc.: 2016.)

Teams are aligned with the program level while looking it time vice. Teams are working on same iteration cadence as well as on same PI boundary. All teams are working as well aligned in the same flow of work. (Scaled Agile, Inc.: 2016.)

3.8 Optimizing SAFe

How could one optimize all this that have been brought up on this paper so far? What are the circumstances that literature sets as best case scenario to use SAFe model in

practice. While an enterprise is developing its IT services and willing to do it with agile methods it should have needed expertise with enough diversity. While speaking about diversity here, I do not refer to one's gender but professionality and industry knowledge. There should be enough of technical as well as business knowledge involved while developing something based on SAFe. Even though there would be technical solution as a goal, it is trying to create business value and it should be built from the economic point of view as well as from technical. This was mentioned already in the principle #1 of SAFe, "take an economic view". While taking advantage of SAFe, agile methods should be adopted on the whole enterprise scale. Dean Leffingwell himself, the creator of the SAFe, has said that SAFe can be adopted in small enterprises that employs around 50-100 persons likewise in bigger enterprises that are employing thousands of people. It is just all about alignment, state of mind and strategy. It is not enterprise size that matters in question of should or should not one use SAFe. The whole enterprise should be organized so that it is supporting agile software development. For working with SAFe, you need also a lot enthusiastic, highly motivated people to work. That is important especially because of the independence nature of the work in SAFe and all the time with realistic manner maximizing the working capacity. SAFe itself states for example that collocation is the key element and critical point for agile team to work effectively. So at least the agile teams should be collocated (Leffingwell: 2010, Scaled Agile, Inc.: 2016.)

3.9 Benefits

SAFe processes are attaining solution alignment between teams. Firstly, while project is initiated, the clear vision is created in feature vise and in general level. Then those visions are discovered on high level and after elaborating high level plans to features, project will create release plan, roadmap and PI objectives. Also key element is to recognize and set clear milestones that need to be accomplished during the development. Evolving that plan happens increment by increment. It can be modified during the journey. Teams are all the time aligned with effective communication and shared learning cycles. This of course demands, that communication between teams and among

the whole train is solid and well organized. At this point it is natural to mention quicker delivery timelines in SAFe. While all teams in the train are aligned with same synchronized iteration cadence, it is easier to group iterations into common program increments and that way integrate different developed components to form one working system, release solution. (Leffingwell: 2010, Scaled Agile, Inc.: 2016.)

SAFe provides also good tools for tracking the status of on-going work, and all teams are aligned with that issue. This comes with consistent DoDs and batch sizes for work to be done. Batch sizes stays equivalent across the teams while stories are estimated with story points based on the workload, that those demand and which is divided equivalently across the teams. That way all can keep up the working phase, as it should be. That way train can minimize delays, keep on track what has been done already and what is still left to be done. While teams are aligned workload and time vise, also the trust between teams is in many cases better. As Pitkänen (2015) mentioned in his study, delivery amount of teams was increasing and the level of trust between teams improved after moving to use SAFe model. Also changes to be innovative and individual's improvement of themselves and as a team increased as well. (Scaled Agile, Inc.: 2016.)

On organizational level, SAFe should give stability to the planning process by using standardized planning methods of SAFe. Besides planning, the whole organization has better changes to continuously improve while issues are raised actively with SAFe methods and noted impediments are escalated after that. Earlier in chapter 3.6.10 mentioned creation of improvement stories and adding those to program backlog emphasize well noted impediments. Hence, impediments will not be forgotten. Issues regarding those will be actually solved and action points taken to block occurred issues accordingly. Many times, faced problems might be noticed but then easily forgotten, because of the lack of needed action points and active escalation. (Pitkänen: 2015, Scaled Agile, Inc.: 2016.)

In Dikert's, Paasivaara's and Lassenius' (2016) article *challenges and success factors for large-scale agile transformations* study is mentioned many challenges in transformation to agile methods. Even though lots of these challenges occur also in the

case of SAFe, lot of those challenges are applying to other agile methods as well. SAFe again, provides solutions to those challenges with its framework. First this kind of challenge that enterprises have been dealing with and can be blocked with SAFe, is autonomous model of teams. In many other agile methods following cases teams are struggling with prioritizing their own goals compared to broader goals of the organization. SAFe provides good tools to align among the whole organization with train wide planning events and other information sharing events. That way teams are well aware of the broader goals and their work on stories is based on those goals, so that every story and feature made, provides value towards the common goal and customer needs. Same applies in problems in achieving technical consistency with agile methods. In SAFe system architects/engineers are guiding the whole train to follow same kind of architecture base in requirements, as well as in development. Features are written with the same agreed structure and then splitted to stories according to SAFe. This should keep coding style aligned among the teams and ensure equal quality in deliveries. (Scaled Agile, Inc.: 2016.)

Another benefit of SAFe which is blocking challenges occurred in other agile methods is regarding requirements and managing those. In SAFe developers has analysts supporting them, while the build is on-going. Developers can check from them in case of uncertainties and together communicate it upper level, if requirements need adjustments. Also for creating stories based on features and estimating those has been guided well with SAFe structure which has been causing challenges in other agile models. Same thing goes with planning the work to be done on long term versus short term. Even though exact plans are made only for every PI in SAFe, roadmap provides further vision for the future work also. On a higher level the portfolio has even further sight of the entity and milestones are set accordingly to the roadmap. That way the gap between long and short term planning is minimized with SAFe. Defining non-functional requirements (NFR's) is one of the key elements in SAFe and those are applied in to testing within the acceptance criteria and testing is made inside the teams. That way there will be no gap between development and NFR's testing since non-functional testing is included in normal testing actions. That, and also lack of automated testing is one challenge that agile teams have been facing in many cases other than SAFe projects.

As well as SAFe is focusing on NFRs and that those are noticed in testing and in approving stories, it is also always encouraging towards automated testing. Like it has been said in SAFe team level training material (2016): "*Test first: Automate now!"* Automated tests are there in the same sprint with building a feature. That kind of approach is ensuring that building velocity is not bottlenecked, quality comes first and scaling is made possible. (Dikert, Paasivaara and Lassenius: 2016, Scaled Agile, Inc.: 2016.)

While many agile methods are such that many times, against like it should, changes only development teams agile but other organization is still working with their old habits, SAFe is different. The primary idea of SAFe is to scale the agile mind set among the whole enterprise, so that all are aligned and running the business, as well as software development with agile methods. For that reason, also challenges in adjusting to incremental delivery pace and product launch activities should not occur in SAFe train. All work among the enterprise is aligned with iterative development and delivery time-cadence such as marketing, running campaigns and other business processes. That way there should not be any gap between these two sectors. (Dikert, Paasivaara and Lassenius: 2016, Scaled Agile, Inc.: 2016.)

All in all, as SAFe's clear benefits could summarized following things:

1. Increased productivity
2. High quality releases
3. Faster time to market (releasing faster)
4. Defect reduction
5. Increased happiness and motivation of employees
6. All ends up to high customer satisfaction

3.10 Challenges

Surprisingly many software development projects are still doing their work based on waterfall model. As a SAFe train, it is nearly impossible or at least very hard to totally

avoid third parties or other co-operation stakeholders that are in waterfall model. Especially when developed, future solution is large and complex. Integrating SAFe train's work to waterfall work will most probably raise issues. Most of these issues come along with requirements and flow of the work. As it has been come quite clear during this study paper, in SAFe, requirements can evolve during the journey towards the solution. What does not change, is the working rhythm. SAFe works with steady flow and releases part of the solution in every PI with same cadence-time. Compared to waterfall it works totally vice versa. Waterfall first does the requirements, lock those and does the building phase based on requirements and after that, tests the whole solution. Obviously, this kind of differentiation in work phases will most probably create some issues. Same kind of situation might occur from differentiation of releasing timing. While SAFe is releasing frequently parts of the solution, waterfall is aiming to only one final release after everything is done and then look what has been created and how it meets with expectations of the customer. (Scaled Agile, Inc.: 2016.)

Any kind of significant transformation will most probably face some issues. Same rule is valid also in enterprise transferring towards SAFe. Change resistance will be always there, if it can not be justified well enough and the transformation would not seem to be that easy. Many people have doubts concerning new roles and tasks that would occur to them alongside moving to SAFe and that those new things would not meet with their expertise. Others have mentioned that losing the freedom in working would be an issue since in SAFe, teams should work collocated in the same team area. One often faced challenge in transforming to SAFe have been skepticism about new working methods. People just do not trust that agile development would bring any benefit, but rather vice versa, it would cause unnecessary mingle to existing way of working. If there is a lot of skepticism among employees, management might think that it is not worth of all the effort to transfer to SAFe. Management might think, that enterprise would waste too much time and effort for ensuring to employees SAFe's benefits, which they would not recognize. Skepticism often fountains from thoughts, that agile methods would not work with complex systems and everything in agile needs to be done literally by-the-book. Thoughts, that every seminar mentioned in the framework must be held and working with self-organized and self-management teams is the same, as working with no

governance and completely without plan. There has occurred also other management related skepticism in the transformation to agile methods. Some employees think that while decision to turn into agile comes from the top to down meaning from management level to employees it does not meet agile methods. So if agile methods go wrong already at his point, how could it work during normal working on daily basis. On the other hand, management can be also resisting change or simply not just understand it correctly. This have happened in such cases where managers have not been involved in practice level to the agile transformation and so being the agile mindset and methods have not spread beyond agile teams. In most of the cases the middle management is the most critical point in transformation, since normal schema is that decision of transformation comes from the top management. Usually team level adopts it pretty well, but middle management do not understand the reasoning behind the transformation or do not involve themselves enough to transformation and hence, lack the know-how of agile methods since it changes their old roles. (Dikert, Paasivaara and Lassenius: 2016, Scaled Agile, Inc.: 2016.)

One way to go wrong with SAFe is the lack on required training. Hence, the value of training should never be questioned, neither by management or any other employees. In SAFe, there are different levels of training depending on what is one's role in the train. The right level of training should be applied accordingly. As a reference to studies, lack of training causes eventually lower use of agile methods and lower motivation of employee's. Lack of right kind of professional training is as well critical point to take attention while going into SAFe. Even though that should not be a problem in SAFe, if just enterprise is willing to invest to right kind of training. After employees has had the right kind of training, applying agile methods should go smoothly. Still sometimes it is just hard to let go of old working habits. While people are still adopting the new ways of working, management should not expect too big amount of delivery at once. Adopting new ways of working takes always some time and both, management of train and customer should notice that, while setting expectations for development in the beginning after transformation to SAFe. Transformation to SAFe requires also physical rearrangements from enterprise. Teams should have placed so that they are working in mutual team area. That way arranging daily stand ups and other SAFe events is easy

and team can actively communicate during work, which is one of the key elements in SAFe. Even after environment is organized according to SAFe and required training has been applied, there might still occur misunderstandings in taking agile methods to the practice among the employees since it usually differs quite a lot from old working methods. Maybe the biggest issues in that might just simply be digesting the agile manifesto, which is the spine of the SAFe. If teams are just blindly doing SAFe methods because it has been decided so, without understanding why, what profit it should create and without having agile state of mind it most probably will not work. Only overloading teams and leading to frustration and lack of motivation to use SAFe and most of all do the work. (Dikert, Paasivaara and Lassenius: 2016, Scaled Agile, Inc.: 2016.)

While some new ways of working are applied to enterprise, it should be customized properly to serve exactly certain circumstances. Same goes with applying SAFe. Even though there are straight forward framework, while taking it into practice, all employees and especially management should rather think how can we apply SAFe methods so that it serves ours and customer's needs the best. Going to much by-the-book is not the idea of SAFe and it is not serving anyone. In the end, SAFe is just giving guidelines and the right, agile state of mind. Also just skipping framework's methods too much is dangerous and might lead to confusion in decision making and controlling the quality in implementation. So there should be found as we say, the "golden middle path" in customizing SAFe. This might take a while at start. Only so simple thing than using the new vocabulary and agile terms helps to set the new way of agile mindset. Patience is gold in transformation to SAFe. Expecting too much too early will in most cases lead to too big expectations at once and while performance might decrease at the beginning of the new way of working, teams might little by little revert to old methods. (Dikert, Paasivaara and Lassenius: 2016, Scaled Agile, Inc.: 2016.)

4 Research method

As a research method for this paper will be used qualitative interview. This interview will be made in company “X” and more precisely inside a project “Y” which is acting in such circumstances where there is a mix of SAFe model and waterfall model. Company and project details will not be mentioned in this research because of business secret issues. With this interview research, I assume to get answers to research questions; how is SAFe model working on circumstances where employees are located in different places with having different culture backgrounds and how does it effect on working that other party is working with waterfall model. Interviewees were selected so that there would be perspective from few different working roles. Interviews were executed as face to face conversations.

4.1 Interview questions

1. What is your working role in the project?
2. How long have you been working in that role, and has that role been changing during the time in the project?
3. Do you have other experiences concerning SAFe than this project?
4. What kind of benefits do you see that SAFe is providing to this kind of project?
5. What are the challenges or problems in using SAFe in this project? Can you give a practical example(s) of these?
6. How well do you see that people across the project have been digested the SAFe methods and state of mind?
7. How much do you communicate with people without face to face interaction?
8. Does it affect somehow to the communication? How do you feel it in general?
9. Do you feel that SAFe methods can be well executed in multi-located working environment?
10. How do you feel working with people from different cultures? Do you see it as a benefit or challenge?

11. Do cultural differences across the project members affect somehow to practicing SAFe model?
12. Do you work with other parties that are working with different model than SAFe?
13. What kind of affects does it have that co-operative parties are working based on different models?
14. How do you feel about SAFe in general?
15. What are from your point view the best circumstances to use SAFe?
16. Does there come anything else worth of mentioning about this subject? Free word.

4.2 Interview results

This chapter will describe results from the interviews as in written format. Specific answers of the interviewees can be found from the appendices at the end of the research paper.

4.2.1 Benefits and challenges of SAFe model

In the interviews there came up clearly two topics that can be counted as benefits of doing SAFe. These are short release frequency and enterprise level of scaling agile methods. And also to control and to get good visibility to big projects and many different projects inside the enterprise. This is what differs SAFe from other agile methods.

What comes to challenges of project to work in SAFe model it is clear that while SAFe is usually used in big projects it is hard to get the whole project to be aligned regarding SAFe methods and how to get the most out of those. If one has been working a long time with more traditional waterfall model for example it is hard for those to move into agile methodology and change the state of mind to something completely new and different. You have be well aware along the whole project what it requires and what is expected from every individual in different role to do for successful SAFe delivery. One

challenge in SAFe that also came out in the interviews was the illusion of agility of the work fixes everything and project can just work on different things in the scope all mixed up without wondering about deadlines or integrations of different parts of the product.

4.2.2 Digesting SAFe methods in the project work

According to the interviews, it is clear that SAFe methods are digested the best on a team level than on the higher levels of project. It was also mentioned that it is more likely that SAFe methods become part of daily routines if the ideology grows from the lower, team level, to the highest level. That way it will fly further since those people that are actually doing the agile development in SAFe project have come up with the idea of this approach. Based on the interviews enterprise level is able to recognize the benefits of SAFe model pretty well. Even though it might not be that visible in practice what it is on team level since the basics of all agile work happens on the team level. This might cause some problems also if management is not well aware of the work that happens on team level and what kind of practicalities they are using. The most important thing on higher level is to digest the right agile state of mind in leading the SAFe train to the right direction. As stated earlier in this paper, SAFe is working on its full potential only if the agility can be scaled across the whole train or even better, across the whole enterprise. Always it does not go like that. If there occur some problems in digesting SAFe methodology it usually evolves from the middle management. This has been stated in the literature as well as in the interviews. Interviews raised reasons for this such as lack of trust inside the project between different working levels and also habits to stick to the old ways of working. To make SAFe model work especially if people are used to work in waterfall, they need to be flexible and open minded for new approach and agile practices need to be digested on every level. Another issue that raised in the interviews was lack of understanding of the needs from requirement point of view to keep the continuous delivery pipeline ongoing. This happened mostly on customer's side which was quite big enterprise in the case so one might expect, what bigger the enterprise is that harder it is to digest SAFe methods.

Especially, if it is a new approach and requires a lot of adapting and learning new ways of working.

4.2.3 Affections of locational and cultural differences to SAFe project

Interviewees were all working in situation where communication without face to face interaction was daily routine. There raised up that it depends a little of the working role and expertise level what kind of affection does lack of face to face interaction have to the communication. Interviewees working as a member of development team or as a business analyst felt that face to face interaction would always be the best way to communicate compared to skype calls or instant messaging not to mention e-mails. Especially among developers it would be better to be located at the same place. It would ease up the communication while one could read co-workers' body language and make for SAFe sessions such as daily stand ups and especially PI plannings much more effective. The main idea of the PI planning is that the whole project gathers to one place to have a joint session. Obviously, that can not work as good as it could if people are divided to multiple locations. Also, the daily working is not that transparent among the development team which is having people working at multiple locations since there might occur some unawareness about what is going on and how problems and such are worked out. On the other hand, it was mentioned in the interviews that sometimes communication and for example having a meeting can be more effective through skype when there is less room for small talk and people usually go more straight forward to the topic itself. All in all, we can say that based on the interviews, at least for the teams inside the SAFe train it would be important to have same locations to make the most of working in SAFe model. (Bass: 2016.)

4.2.4 SAFe model synchronizing with waterfall model

SAFe model is basically opposite delivery method compared to traditional waterfall model. That was very clearly emphasized in the interviews as well. Interviewees were working in a project where there is in addition to their customer also a third party included, which is working in straight waterfall model. That difference in delivery

method is causing multiple challenges for both sides but at least according to interviews more challenges for SAFe side.

Biggest challenge mentioned in the interviews was expectedly releasing cycle. While the one using SAFe methodology would aim to release very often, even every two weeks, third party working in waterfall model is releasing with much longer and at the beginning of the project decided frequency. Synchronization between these two is obviously challenging. In this project at hand, this challenge has been tried to tackle with implementing with agile, short frequency but by building the product piece by piece to warehouses and releasing it same time with the counterpart working in waterfall world once they are ready for that simultaneous release. There have been occurring two kind of issues with this approach. Issues are escalating while requirements are changing, which happens quite often in SAFe delivery and on the other hand, while defects are found in testing. Firstly, when requirements are changing in SAFe train it takes time from third party to change their requirements which should basically frozen already. This then again causes delays to SAFe delivery which requires updated specifications from third party. Another challenge, as mentioned, comes along while defects are found in testing. Once from frequently implemented and tested components from SAFe train has been found defects in testing which requires changes from third party it takes again a lot of time to do the re-factoring work on their end. So, scheduling is causing challenges basically in all phases of the work while trying to find balance between SAFe and waterfall models. Regarding testing there was raised in the interviews another challenge that very likely would occur in later phase of the project. When SAFe train is building with big volume to warehouses and releasing it simultaneously with third party for example once in every half a year it will require large joint testing which most probably has its own issues and there might occur remarkably big number of defects in that testing. And when these defects are revealed this late, one big advantage of SAFe model, short feedback cycle, is washing away.

4.2.5 Best circumstances for practicing SAFe

We know already that term SAFe comes from scaled agile framework which is meaning that SAFe methodology and agile way of software development should be scaled on the enterprise level according to SAFe model. Scaling agile methods across the whole enterprise needs commitment from everybody and not only from the team level which might be usual case. People making the business decisions should be committed to agile way of working, have open minded attitude for new ways of working and be able to make needed decisions with no dependency to other parties. Scaling agile methods perfectly on enterprise level would mean that if there are several projects on-going inside the enterprise, those should all work with SAFe. That was raised also in the interviews as the best circumstances to make SAFe work properly. According to interviews SAFe would be good option when in enterprise level project there is clear business vision what needs to be done, but on the other hand it was also mentioned that SAFe provides advantage if requirements are not so clear at the beginning of the project or those might change even a lot during the project. All the interviewees were stating that key thing is that all in the project are aligned with SAFe methods and have digested those well. All in all, following key elements were raised to make SAFe work as its best: agile and open state of mind, clear business vision from the highest level of enterprise and people at every level digesting SAFe methods very well.

5 Conclusions

This chapter is about to make conclusions regarding prior foundings of this research. It will compare the literature review to the interview results and elaborate how literature's SAFe guidelines, benefits and challenges are shown in practice in the case project.

5.1 SAFe values

Like mentioned in the chapter 3.2 there is four core values in SAFe model. These are *Respect for people and culture, Flow, Innovation and Relentless improvement.* If we think about how well interviewees have adopted these values or what kind of picture one gets from interview results regarding how well their projects have adopted SAFe core values, I would say pretty well. Mostly interviewees saw cultural differences as a richness and as long as people were doing their job it did not matter where they were from or what was their cultural background. So there clearly can be sensed high *respect for people and culture*. What comes to *flow*, the case is little different. In perfect SAFe project, train is proceeding with continuous flow and releasing end product piece by piece frequently to production. In this specific case that we have in the interview, project is having big dependency to third party which is delivering in waterfall model which is affecting their releasing as well. Case project has been forced to synchronize their releasing time schedule with the third party and so being, they have not been able to get full potential out of SAFe model's frequent delivery cycle. That being the case, releasing have not been going exactly by the book and clearly it has caused some negative affection for agile delivery in this case. Based on the interviews, it is quite hard to define the level of *innovation* in the case project. Nevertheless, while it is one of the major corner stones of SAFe model and none of the interviewees did not mention lack of innovation or such while asking about challenges with working in SAFe model, one can conclude that lack of innovation is not occurring in this project. Presence of *relentless improvement*, the fourth core value of SAFe model, can be from my point of view identified from the case project according to the interviews. Interviewees all mentioned that their requirements are changing often and project is adjusting working

based on those continuously. Also, they have faced some troubles on customers side to adapt SAFe methodologies but improvement on that side has been happening and the scaling on enterprise level has been going forward. But clearly relentless improvement is one area where this project has still room to grow.

5.2 Structure of SAFe model project

In chapters 3.4-3.7 in this paper there has been introduced different levels of SAFe model, what are the different roles and actions done on each level and what are expected from those. According to interviews this factor is at least from delivery side well adopted that which are the different roles and what is expected from people in different roles. Even though, interviewees also brought up that some have problems on doing exactly their part and especially on customer side's middle management is having troubles on providing what is expected from them. According to interviews I would say it is also result of customer's middle management questioning SAFe methodologies and so being meeting their expectations might be difficult. Another issue that case project is facing compared to SAFe model guidelines for structure is releasing on program level. This was already brought up in previous chapter while thinking about core values of SAFe. Program level's biggest responsibility is to release in frequent cycles to production and simply this project has not been doing so. It is major difference to what founders of SAFe model have been thinking and it is not surprise that this factor is causing challenges for this project. Positive thing is that people in the project recognize well different roles and levels of the agile train and knows what they should deliver and how. Next step for them would be to adjust their working accordingly.

5.3 Optimizing SAFe

For SAFe model to work with full potential it needs certain circumstances which were brought up in chapter 3.8. From my point of view interviewees were aligned with literature on this but they mentioned also some additional points to consider based on

their own experience. Main message from literature and from interviews was clear; alignment of the whole enterprise to SAFe methods and adapting methods to every day working is the key. In addition to that literature and interviews both highlighted one key thing, releasing with short frequency. Difference with literature and case project was that both stated that frequent releasing would be the right way but currently case project was not doing it due to dependencies to the third party. All in all, interviewees had a clear picture how to optimize SAFe model and it was aligned with SAFe guidelines. Only problem was that because of existing circumstances, it has been challenging to follow SAFe guidelines strictly in the case project. And when one practices SAFe without it key functionalities, it is obvious that the model is not working with its full potential.

5.4 Benefits of SAFe

Based on the literature and interviews, it can be stated that when SAFe is used well its biggest benefit is quick, consistent and quality technical solution delivery. This is possible only if SAFe methods are executed as they should be. It is very hard to get full potential out of this model if it is twisted too much. Key benefits of SAFe model which leads to quick, consistent and quality delivery are visualized in the picture on the next page.

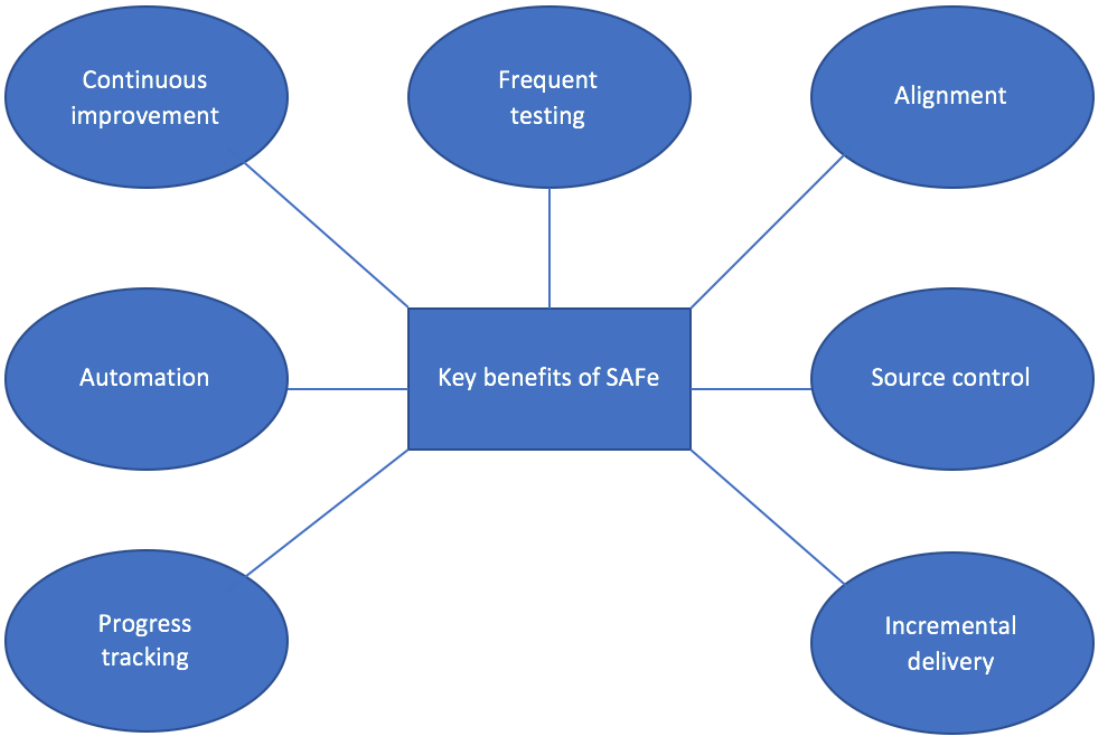


Figure 4: Key benefits of SAFe model.

Continuous improvement

SAFe is aiming to develop individuals and common processes all the time whenever there is a change to it. Obviously, no matter which development model you are using, it should aim to that. What makes SAFe different, it provides practical tools for the people as individuals and for the SAFe train and even for the enterprise to improve during the delivery process. Incremental delivery combined with different improvement practices such as *Inspect & Adapt* (chapter 3.6.10) makes that possible. In the interviews the one from interviewees who had been working the longest time in the project brought up that working has evolved a lot from the beginning of the project. This was mostly because of that people have been digesting SAFe methods and learned how to use those better during the time. Also natural change was that amount of people has increased a lot. So there has been some good improvement in the case project and it has grown to be able to use SAFe its practices.

Frequent testing

Testing in SAFe model is done frequently in every increment and most importantly, early enough. This testing approach ensures quality end product while defects are noticed and hence, fixed frequently and early. That way end product gets tested piece by piece before implementation is fully done. This usually has major effect on finishing project in time and really makes a difference in quality. Even though the case project were not releasing to production frequently as it should be done in SAFe, they were still testing frequently. Anyway, due to the dependency to the third party, project could not do joint testing with the third party in the same frequency which was a big concern raised in the interviews. This being the case, it can be concluded that the case project has been failing to get advantage of this key benefit of SAFe.

Alignment

When SAFe is executed correctly all different scrum teams working in the project are aligned and working towards common goal with synchronized frequency and work load. In addition to that alignment should go through the whole SAFe train and further, the whole enterprise if possible. From the interviews it can be learned that there was some misalignment among the case project. At least on the team level it was caused partly from cultural and locational differences among the teams. On the higher level again, for example on system team level it was brought up that location of team members does not effect at all. On the other hand, as mentioned in the chapter 3.6.3, system team should actually be the one who is making teams aligned with each other in the best possible way. Apparently it has not succeeded perfectly in the case project.

Source control

As mentioned in the chapter 3.6.3 DevOps team is making sure that all resources of the project can work with their full potential. DevOps team does that by providing their expertise and with checking that methods and tools used are correct and those are correctly used.

Incremental delivery

Delivery in SAFe model happens in increments with small batch sizes. With small batch sizes the continuous delivery can happen with short lead time. That ensures the continuous flow in incremental delivery.

Progress tracking

Status tracking of the work progress is ensured in SAFe with consistent DoDs (refer to chapter 3.6.8). It can be easily tracked when certain feature or story is ready when it fulfills the DoD defined for it.

Automation

Automation in SAFe occurs mostly in testing. When testing is automated, the amount of human errors is minimized and the velocity of testing maximized which makes testing multiple times more effective compared to manual testing.

5.5 Challenges of SAFe

As per the literature and interviews there should not occur that much challenges in practicing SAFe model if it is done properly. Nevertheless, as soon as SAFe is not followed as it is supposed to or for one reason or another, things start to get complicated. Most of the times this is consequence of people not digesting SAFe methodologies well enough. That is usually due to the fact, that people are too much sticking to their old habits and familiar ways of working. As mentioned in the chapter 3.10, digesting SAFe methods is the hardest for the middle management. Same thing was raised also in the interviews.

Another issue that makes practicing SAFe challenging is if it is modified too much. One should of course always use common sense and adjust SAFe the way that it suits best

for the project and enterprise at hand. Still, if SAFe model is adjusted too much so that it starts to miss some of its corner stones, framework will lose its value. As it was also seen in the case project of the interviews. They were basically forced by dependency circumstances to adjust their release frequency and environments which is, most definitely one of the corner stones of SAFe model. That was already now starting to cause some challenges for the case project and interviewees were not so optimistic about future phases to come, especially about future testing activities.

Hence, it can be stated that as two major challenges of SAFe model are digesting of SAFe methodology and alignment with other parties which are not working with the same model. Even though, both of these challenges can be handled. Digesting with well aligned and trained project with patient transformation period and perfect amount of common sense in framework adjustment. Alignment with other parties working in different project model should be done with extra careful planning (chapters 3.6.4-6) including well visualized roadmap with clear milestones along the progress.

5.6 Further research possibilities

In the future when people on IT industry has gained more experience regarding SAFe model, reasearches can go deeper in the conclusions that how the model work in different circumstances and how it can be evolved. For time being SAFe is still relatively young method on the industry compared to for example waterfall model or to some of the other agile models. It is also interesting how SAFe model will evolve based on the experiences of the users and further research results. This would be one interesting point of view to do SAFe model research in the future. That how SAFe model has been changing from the beginning of the model's life cycle.

Bibliography

Ball, Andrew, Ajay Nair & Mirco Hering (2015). *Key Accenture Learnings on Scaled and Distributed Agile Delivery*. Scaled Agile Framework. Available: http://www.scaledagileframework.com/accenture-case-study/.

Bass, Julian M. (2016). *Artefacts and Agile Method Tailoring in Large-Scale Offshore Software Development Programmes.* Information and Software Technology. Volyme 75. Pages 1-16. Available: https://www-sciencedirect-com.proxy.uwasa.fi/science/article/pii/S0950584916300350.

Beck, K., M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber & D. T. Sutherland (2001). *Principles behind the Agile Manifesto.* Manifesto for Agile Software Development. Available: http://agilemanifesto.org/iso/en/principles.html.

Buchan, J., Holvitie, J., Hyrynsalmi, S., Leppänen, V., Licorish, S. A., MacDonell, S. G., Mendes, T. S. & Spínola, R. O. (2017). *Technical Debt and Agile Software Development Practises and Processes: An Industry Practitioner Survey.* Information and Software Technology. Volyme 96. Pages 141-160. Available: https://www-sciencedirect-com.proxy.uwasa.fi/science/article/pii/S0950584917305098.

Cao, D., Djordjevic, M., Nikolic, V. & Stankovic, D. (2013). *A Survey Study of Critical Success Factors in Agile Software Projects in Former Yugoslavia IT Companies.* Journal of Systems and Software. Volume 86. Issue 6. Pages 1663-1678. Available: https://www-sciencedirect-com.proxy.uwasa.fi/science/article/pii/S0164121213000496

Cawley, O., Conboy, K. & Wang, X. (2012). *"Leagile" Software Development: An Experience Report Analysis of the Application of Lean Approaches in Agile.* Journal of Systems and Software. Volume 85. Issue 6. Pages 1287-1299. Available: https://www-sciencedirect-com.proxy.uwasa.fi/science/article/pii/S0164121212000404

Dikert, Kim, Maria Paasivaara & Casper Lassenius (2016). *Challenges and success factors for large-scale agile transformation: A systematic literature review.* Journal of Systems and Software. Available: http://ac.els-cdn.com/S0164121216300826/1-s2.0-S0164121216300826-main.pdf?_tid=4e6f7594-d994-11e6-873c-00000aacb362&acdnat=1484314347_4cb4400576e62077159f23d17b00029e.

Humble, Jez and David Farley (2010). *Continuous Delivery: Reliable Software Releases through Build, Test and Deployment Automation*. Addison-Wesley. ISBN-13 978-0321601919.

Kim, Gene, et al (2013). *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. IT Revolution Press. ISBN: 9781942788294.

Leffingwell, Dean et al. (2014). *Applying the Scaled Agile Framework in an Outsourcing Context: Infogain Case Study.* Scaled Agile, Inc. Available: http://www.scaledagileframework.com/infogain-case-study/.

Leffingwell, Dean Foreword by Renertsen, Don (2010). Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. Agile Software Development Series by Alistair Cockburn and Jim Highsmith. Pearson Education, Inc. Boston. ISBN-13: 978-0-321-63584-6.

Lilja, Valtteri (2017), Java developer. Company X. Interview, Helsinki 19.4.2017.

Mali, Senapathi & Meghann L. Drury-Grogan (2017). *Refining a Model for Sustained Usage of Agile Methodologies.* The Journal of Systems and Software. Volume 132. Pages 298-316. Available: https://www-sciencedirect-com.proxy.uwasa.fi/science/article/pii/S0164121217301498.

Pitkänen, Antti (2015). *Agile Transformation – A case study.* Aalto University. School of Science. Degree Program in Computer Science and Engineering. Master's Thesis.

Scaled Agile, Inc. (2017). *Scaled Agile Framework.* Available: http://www.scaledagileframework.com.

Viljanen, Jukka (2017), Product owner. Company X. Interview, Helsinki 28.3.2017.

Åkerman, Ilkka (2017), System test manager. Company X. Interview, Helsinki 26.4.2017.

Attachments

ATTACHMENT 1.

The first interview 28th March 2017, transcribed 30th March 2017

What is your working role in the project?
Product owner. Consisting taking care of the team backlog, features stories and grooming features into stories, and following the work keeps it schedules and implementation is up to date with requirements.

How long have you been working in that role, and has that role been changing during the time in the project?
Two and a half months in this role. Yes, I started as a business analyst working on features and definitions.

Do you have other experiences concerning SAFe than this project?
Not in SAFe model. So, SAFe was new to me but agile methods in general were not.

Did you have troubles to assimilate SAFe methods or the framework in general?
Not so much. At the beginning, SAFe methods were of course new to me.

What kind of benefits do you see that SAFe is providing to this kind of project?
Continuous releasing and implementation work. You can see every 2 weeks the situation of the product. In our case, we get the UI screens up and running, end customer can easily see where we are going. They usually do not see the code what is going behind the screens. So, it is easy to see if there is any defects or something to clarify further. Alongside good visibility to features and stories turning into practice prioritizing has its benefits in SAFe. These are though same in other agile methods.

What are the challenges or problems in using SAFe in this project? Can you give a practical example(s) of these?

Main problem is that everybody that are working in the project does not follow SAFe model. We have other external party in this project which does not work in SAFe model which is causing some problems. When other party is working on waterfall model it causes long waiting times in case of changes to requirements. While SAFe model tries to implement continuously this has caused lot of scheduling problems, especially regarding testing. When we find some errors in testing, it takes lot of time that other party is ready with their re-factoring work. Also, if someone in the team is not following SAFe precisely or does not participate actively to SAFe ceremonies it causes troubles. In team going through all the SAFe phases might be forgotten which will cause some extra work.

How well do you see that people across the project have been digested the SAFe methods and state of mind?

New people has usually bit of learning curve if they are not familiar with SAFe already which is understandable. Those kinds of issues can be mitigated quite well. On team level, it is easier that on higher level. Program level is not that focused or aware what the teams are doing in practice. On the enterprise level, you do not even care what teams are doing since responsibilities are on more higher level in the big picture. There needs to be the trust that everybody is doing their role on the level that SAFe requires. You should always know what is your role and what is required from you and doing only that.

How much do you communicate with people without face to face interaction?

We have offshore delivery center where our scrum teams are located. Also even in onshore everybody in the project are not located in the same place while some are at customer premises and other at our own office. Skype and e-mails are used regularly. Lot of meetings through Skype. Though meetings are done always face to face whenever it is possible.

Does it affect somehow to the communication? How do you feel it in general?

Face to face is of course most effective way but Skype and especially Skype calls are quite effective as well. Then there is instant messaging which is better than e-mails but it is still written text so you are not able to read others body language etc. so it is not as effective as face to face which is always recommended and the best way.

Do you feel that SAFe methods can be well executed in multi-located working environment?
Yes and no. As long as the communication is fluent and open and everybody does their work it is not a problem. Though if everybody would be continuously face to face it would be better.

How do you feel working with people from different cultures? Do you see it as a benefit or challenge?
It is good thing but you need to take cultural differences into account and understand how they affect to work. It can be beneficial but it might also require some additional steps or attention but that is just the way that people are.

Do cultural differences across the project members affect somehow to practicing SAFe model?
Similarly, than to any kind of work with people from different kind of cultures. It is more beneficial than it is not. Safe is focusing on continuous flow rather than waiting possible issues or concerns regarding for example cultural differences. So the work in SAFe is most probably going forward even though there would be some issues because of cultural differences which wouldn't be the case necessarily in the waterfall model.

Do you work with other parties that are working with different model than SAFe? (*Already answered before*)

What kind of affects does it have that co-operative parties are working based on different models? (*Already answered before*)

How do you feel about SAFe in general?

It is a good method, but can be bad if everybody is not committed to SAFe methods and principles. But as long as everybody are working as agreed with SAFe principles and guidelines then it can be good. Cultural differences can be an issue but as long as those are taken into account it is not a problem.

What are from your point view the best circumstances to use SAFe?
Large projects with not so clear requirement or possibly changing requirements. Multilocated projects are okay if there are proper tools in use like Skype but if these are not available multilocated projects will suffer a bit for SAFe. All in all, SAFe needs proper tools available that it works.

Does there come something else to your mind worth of mentioning about this subject that has not come up during this interview?
Everybody needs to be aligned and work based on principles of SAFe. Otherwise it will not work. As in all project work methods.

Second interview 19th April 2017, transcribed 25th April 2017

What is your working role in the project?
Java developer.

How long have you been working in that role, and has that role been changing during the time in the project?
10 months. Maybe little bit more quality work but mainly tasks has been the same. Responsibilities has increased.

Do you have other experiences concerning SAFe than this project?
Not before this project.

What kind of benefits do you see that SAFe is providing to this kind of project?

Safe benefits are mostly on the enterprise level to keep on track multiple projects. On a team level scrum methods etc. can be applied to many different project models. From the developer's perspective, Safe does not provide anything spectacular compared to other agile models. To me safe's benefit is the agility of the whole enterprise.

What are the challenges or problems in using SAFe in this project? Can you give a practical example(s) of these?
Safe does not bring any problems as itself but in any project, you should know well how to use it and have the clear direction where to go. If you do not have that visibility it does not matter what project model do you have, you are lost. I have not myself found anything negative about Safe. Maybe it brings agile illusion for some people that project can just work on many different tasks and nothing needs to be ready ever which is the wrong approach. Agile should make things easy to adjust but still project should have visibility where, how and when they want to be.

How well do you see that people across the project have been digested the SAFe methods and state of mind?
Among the people that I am in contact usually in daily working I would say no there is not differences on digesting SAFe methods or resistance against agile state of mind. Some people might be using the methods wrong but that is totally different thing.

How much do you communicate with people without face to face interaction?
Really often. Almost 75 percent of the communication is something else than face to face. I would like it to be little less.

Does it affect somehow to the communication? How do you feel it in general?
Sometimes it makes things harder. Some information might be missed since people are not speaking with their native language so some language barrio might occur.

Do you feel that SAFe methods can be well executed in multi-located working environment?

On a developer level, it would be better to have the whole team in same location to have face to face communication. My all other developer team members are located offshore so some miscommunications occur every now and then. I am not always aware what is happening there or they do not know what is happening here. Perfect example would be daily stand up. That would be definitely better to have face to face than via skype.

How do you feel working with people from different cultures? Do you see it as a benefit or challenge?
It is affecting to work in some cases. My offshore colleagues' work culture is very hierarchical. It affects to the agile way of doing things since their managers might be pushing people a lot to do something that they think is right but in agile every developer should have their specified tasks from the backlog on to do list so managers should not mess that with their own priorities. In the end, I would say that having different cultures in the project is little bit both benefit and challenge.

Do cultural differences across the project members affect somehow to practicing SAFe model?
There is no affect that would be only related to SAFe.

Do you work with other parties that are working with different model than SAFe?
Our project does but me myself I am not in contact with third parties so can not really answer to that question.

What kind of affects does it have that co-operative parties are working based on different models? (*Already answered before*)

How do you feel about SAFe in general?
SAFe is a working model definitely. It keeps the enterprise level in the loop when you have multiple projects going on enterprise level can keep tracking them all. It is little different on the developer level since we are in our own bubble so we are not that concerned about the project management. All in all, SAFe works if you know how to use

it. And really actually use it, not just some SAFe-ish way of working. Hybrid models do not work from my point of view.

What are from your point view the best circumstances to use SAFe?
Any enterprise level project where you have clear business need and something to develop to fulfill that need. And if you need to put up new project perfect place to use SAFe is at enterprise where there are already SAFe trains existing. Also, perfect project for using SAFe is when it is all the way in enterprise's own hands and decision are not depending on anyone else and everybody working in the project are from the same company and same party.

Does there come anything else worth of mentioning about this subject? Free word.
Always when using SAFe, you should think in agile way. Really have the knowledge how agile methods works and how you will get things done with those methods. If you do agile in a wrong way you do not get anywhere and things just get more messed up compared to old waterfall model. While implementing SAFe, the whole enterprise needs to adapt the agile way of thinking and that probably the hardest thing in SAFe. As a developer, it is easy for me but for someone who has been working with waterfall model for his or her whole life it is much harder. In current technology world, you need to adapt quickly, otherwise you will nott make it.

Third interview 26th April 2017, transcribed 27th April 2017

What is your working role in the project?
Onsite test manager. If you look on the SAFe model on paper, there is no test manager there. Test manager is more like coming from the old world where we had separate team for testing and separate people to manage testing team and their activities, functional and non-functional testing. In the SAFe model, I would place myself to the system team together with people that are responsible of the environments and tools that are used in the project. In safe test manager is not directly managing the testing but rather to be a support person for the team needs for the definition, coding and specially

to do testing and make it to be as easy as possible. In safe you must consider all the other activities as well so I ca not focus blindly to testing only. In perfect SAFe world, we would have continuous integrated pipeline where you could not say that now I am doing coding and now testing since these should be streamlined to one unified process.

How long have you been working in that role, and has that role been changing during the time in the project?

One and a half years now during the duration of our SAFe train almost from the very beginning. Yes, it has been changing. We have learned more how to work in SAFe. And there was not that many people in the project while we were ramping it up. We had only one team and now when we have lot more teams so we have also more need for common rules and practices how to work. In the beginning, I was more involved with the hands-on testing activities and reporting but now my tasks are on a higher level so that teams are aligned.

Do you have other experiences concerning SAFe than this project?

No this is my first SAFe project. But I have experience on other agile development and have otherwise also long background in IT and software development so I have been involved also in many waterfall projects and in several agile projects later on. On the development team level, SAFe brings nothing new for teams compared to other agile methods. New things come up on a portfolio, program and enterprise level where conflicts might easily occur.

What kind of benefits do you see that SAFe is providing to this kind of project?

SAFe is scaling agile projects on an enterprise level and that way SAFe's goal is to minimize those conflicts mentioned prior. Everybody can run project with one team but magic of SAFe is to make several teams to work aligned with each other. To make large project to work on agile way and without SAFe methodology it would be hard to handle all the projects and all the teams that enterprise is doing. That's what SAFe is build for.

What are the challenges or problems in using SAFe in this project? Can you give a practical example(s) of these?

Challenging to get the organization that has been working long time as a waterfall project to understand the needs to successfully deliver SAFe project. The biggest challenge is that organization level understands how they should work in order to get the best out of the SAFe model. In our project, it has been hard to get the whole organization to work in a way what supports SAFe delivery. But on the other hand, the whole SAFe methodology is made to tackle that challenge. So, I am still believing that SAFe has potential to support a large-scale delivery but it requires the whole organization to understand what is expected from them.

How well do you see that people across the project have been digested the SAFe methods and state of mind?

There are differences depending on in what role people are working. If the new ideology idea comes fro the higher level of the project it's usually hard that people on the lower level of the project digest that well. On the other if something new comes from the working level to the highest level it's more likely that it will fly further on the long run in the daily working habits. Our customer's top management see the benefits of SAFe way of working and the team level also but somewhere between these two levels there is some kind of, not resistance, but more like habits to stick to the old way. They are not having so much readiness to be flexible and open minded for changes needed from that level. It has potential but it has it's challenges for the company that has been doing waterfall model long time to change their way of thinking to SAFe model.

From technical perspective, there has not been total understanding for the needs to deliver with agile methods from requirement's point of view from customer's side so that the agile building pipeline would work as it should be working. This is something that needs time to change inside company of this size that our customer is.

How much do you communicate with people without face to face interaction?

A lot since nowadays bigger part of the project is located offshore. Third party is also on the other location and the customer as well so there are several physical locations for this project. Skype meetings are very regular.

Does it affect somehow to the communication? How do you feel it in general?
Not only bad thing. It has its pros and cons. Many times, it can be even more effective to have communication through skype compared to have face to face meeting. It might be easier to stay on the topic rather than have small talk and other via skype.

Do you feel that SAFe methods can be well executed in multi-located working environment?
Development team would be more effective if they would be on the same location. Same PI planning we are now having onshore in one big meeting room and offshore in another one so that would be better if the whole project would be at the same location. I am myself working inside the system team and for us it is easier to work in different location than it is for development teams or business analysts. So, people working on same level of expertise should be in same locations but communication between these different teams can be well done through skype etc.

How do you feel working with people from different cultures? Do you see it as a benefit or challenge?
Challenges regarding cultural differences are there regardless of the delivery method. On the other hand, SAFe provides well documented structure how we are suppost to communicate and what is expected for definition of done and definition of ready. So that is good thing in SAFe that we have clear rules what is expected and it helps to align regardless of the culture. One can not understand it differently if you have clearly for example five things that needs to be done before definition of ready or done can be accomplished. If there would not be this clear structure there would always be discussion on whether things are ready or not. So, SAFe is providing benefit for multicultural project since it provides the needed meetings and tells what kind of output and input is expected. For some it might feel that SAFe is too strict and it is limiting too much but for me it is good thing. For other cultures than Scandinavian people these strict rules might feel tighter but it is exactly the benefit of SAFe so all are then aligned regardless of the culture.

Do cultural differences across the project members affect somehow to practicing SAFe model? (*Already answered before*)

Do you work with other parties that are working with different model than SAFe?
Yes, and they are not using SAFe. They are delivering in releases from the waterfall delivery method. Intgrating our work with them is one of the challenges we have faced. On the other hand, we have not taken the full benefit from the SAFe in the delivery way. We have not been delivering incrementially to the production or even to the pilot what would have been helped to tackle these challenges. We have building stuff to warehouses and we would be then releasing at the same time with our counterpart and that will need big joint testing activities and possibly bery likely there will be a lot of issues or bugs and defects in that testing. Golden thing with agile is to shorten feedback cycle for example to find bugs and correct those. But when we are building lot of stuff and releasing only once in half year or once in a year then bugs will be revealed only very late which is against of agile methods and SAFe model. So, to sync Safe model to waterfall is challenging because of the release frequency differences. But it is not waterfall's fault that we are not releasing and should not say that we can not release if other parties are not ready. We should release often in SAFe regardless from other parties to have the feedback cycle kept short because that is the of agile.

What kind of affects does it have that co-operative parties are working based on different models? (*Already answered before*)

How do you feel about SAFe in general?
I am feeling optimistic about SAFe. It has lot of good things. It is hard to get agile delivery on going in large enterprises but I would see that inside of new aeras it would work also inside this customer. It is easing to give new business direction for the customer and gives readiness for getting agile mindset for our customer.

What are from your point view the best circumstances to use SAFe?
Lot of open mindness at least. Upper level support is definitely needed from the highest management level. Business level needs to have clear vision what is required from them

to work in SAFe project. Also, lower technical part need to make continuous delivery happen.

Does there come anything else worth of mentioning about this subject? Free word.
SAFe will be around for a while at least in case of big companys. It is not yet ready and model is developing all the time by developers of SAFe. It is worth of trying and SAFe developers are welcoming all improvement ideas and changes are coming among new versions of SAFe. It is not expensive methodology so it is tempting companies to try it