

# Facilitating Change in Decentralized Bottom-Up Agile Transformation

Antti Jaakkonen

2022 Laurea

Laurea University of Applied Sciences

# Facilitating Change in Decentralized Bottom-Up Agile Transformation

Antti Jaakkonen Leading Transformational Change Thesis 6, 2022

# **Laurea University of Applied Sciences**

**Abstract** 

Leading Transformational Change Master of Business Administration

Antti Jaakkonen

#### Facilitating Change in Decentralized Bottom-Up Agile Transformation

Year 2022 Number of pages 87

The purpose of this master's thesis was to create a holistic understanding of the causes for perceived slowness of development organisation and to create a compelling plan to improve it in one of the Finnish telecommunications operators.

The project's initial goal was to investigate one development area and understand why the lead time of development initiatives was slow. The plan later evolved to cover most of TC1's development areas as the performance of a single area was not a product of only that one area. Several interdependencies were revealed between the development areas, structure and management model of the line organisation and how the business lines were organized.

As TC1 aspired to be an agile organisation in scale, this work bases the theoretical framework on studies, literature, and research from lean, agile, DevOps, organisational change, leadership, and management domains. The hypothesis was that improving development organisation performance requires participation from people, process, and technology domains throughout the organisation and not just from the specific development area.

TC1's speciality was a strong bottom-up agile movement with different beliefs about what agile is and what it means for the company in every sub organisation, making the ongoing agile transformation bottom-up and decentralized. The chosen research & development method was used to bring people around the organisation together in structured and well-facilitated workshops.

The workshops built on top of previous research and, when applicable, referenced the findings ranging from managing enterprise-wide work in process issues down to technical practices and end-to-end testing. The Workshops were structured to cover creating the mission, vision and the overall goal for the change initiative down to roles & responsibilities, key values and principles and assessment of how development organisations were organised.

While the change plan created in the workshops had almost unanimous support from the workshopping participants, many of whom were top management representatives. The management team ultimately did not accept the plan.

Some success still resulted from implementing parts of the plan in a few company areas. Implementing parts of the plan in parts of the TC1's organisation resulted in almost complete eradication of the original pain point that triggered the work behind this thesis.

The used method looks functional in many similar cases and organisations with bottom-up leadership traits. The workshops yielded many universally usable concepts that are not necessarily new. Still, the in-depth look into what problems they aim to countermeasure can be valuable to many in a similar situation to TC1. The work also made it painfully clear how the lack of a shared and commonly understood vision and reasoning behind agile organisational aspirations can impede or slow down much-needed changes. For that reason, the standing recommendation for future initiatives is to establish a shared understanding of what agile means for the whole company before anything else.

Keywords: lean, agile, organisational change, facilitating change

# Contents

1	Introdu	uction6
	1.1	Motivation behind this work6
	1.2	Project overview
	1.3	Approach7
	1.4	Company background8
		1.4.1 Line organisation8
		1.4.2 Virtual organisation9
		1.4.3 Agile release train at TC1
		1.4.4 Team types
		1.4.5 Program Increment cadence / Planning window
		1.4.6 Main metrics
		1.4.7 Shared services at TC1
2	Backgr	ound
	2.1	Lean
	2.2	Agile
		2.2.1 Scaled Agile Framework
		2.2.2 Business agility
		2.2.3 Organisational agility
		2.2.4 Team and technical agility
		2.2.5 Lean agile centre of excellence (LACE)
	2.3	DevOps
	2.4	Portfolio Management
	2.5	Project to Product thinking
	2.6	Leadership vs management
	2.7	Organisational Change
3	Projec	t methods
	3.1	Background
	3.2	Problem to be solved
	3.3	Workshop design
	3.4	Target outcomes
4	Facilit	ating the workshops and creating the change proposal
	4.1	Redefining the problem
	4.2	Workshops in action
		4.2.1 Workshop #1 - Defining the goal statement for organization change 41
		4.2.2 Workshop #2 - Defining the KPIs for the organization change
		4.2.3 Workshop #3 - Defining the Actors and Impacts

		4.2.4 Workshop #4 - Defining the concrete changes	<del>1</del> 7
		4.2.5 First change of plans	50
		4.2.6 Workshop #5 - Shared principles and practices 5	50
		4.2.7 Workshop #6 - Roles and responsibilities	53
		4.2.8 Workshop #7 - The Organising for success	56
		4.2.9 Workshop #8 - Supporting structures	58
	4.3	Defining the next steps of Railyard 2.0	52
		4.3.1 Problem statement	53
		4.3.2 Motivation for changes and hypothesis of outcomes $\dots $	4ز
		4.3.3 Goals and key metrics	55
		4.3.4 Backlog content management model	55
		4.3.5 Organising TC1's agile release trains	57
		4.3.6 Initiatives to improve collaboration	59
		4.3.7 Communities of Practice model to share learnings and good practices $\dots$ 7	70
		4.3.8 New Roles: Head of Engineering	71
		4.3.9 Task Force Teams	72
		4.3.10Frequently Asked Questions	73
		4.3.11Proposed next steps	73
		4.3.12Reviewing the change proposal with the workshop group	75
5	Results	·	75
		5.1.1 Results on the single agile release train level	76
		5.1.2 Results on the organisational level	77
6	Conclu	sion	77
7	Figures	· 8	35
8	Tables	8	37

# 1 Introduction

Organisational change is hard, and aspiration towards agility often touches every part of the organisational fabric and its incumbent power structures. At the same time, the long-lived divide between business and technology is getting shallower or, as the CEO of Microsoft stated in 2019, "Every Business is Now a Software Company" (Holmes, 2019). This emerging theme of enterprises being complex sociotechnical systems (Beattie, Hepburn and O'Connor, 2021) and facilitating change in such systems require working simultaneously with organisational structures and attached processes and the underlying technology, architecture, and how we expose our capabilities and data to others.

This work will look at one company in the telecommunication industry, the effects of bottom-up agile transformation, and a method we developed for continuous development of the virtual agile organisation structure and ways of working. We will analyse the start-state using research papers and literature covering agile, organisational change and leadership domains. We will then investigate how we designed our change apparatus and how the principles and practices connect to lean and agile philosophy. Lastly, we will evaluate the effects of our change approach and look into future developments and the transferability of innovation.

#### 1.1 Motivation behind this work

Twenty years have passed since the Agile Manifesto was written and signed by prominent software engineering enthusiasts in Utah (Beck et al., 2001). The DevOps movement (Edwards, 2012) and the introduction of frameworks to scale agile development like LeSS (Vodde and Larman, 2014) and Scaled Agile Framework ("SAFe") (Leffingwell, 2020) are also over ten years old. In many ways, agile and DevOps are beginning to be household terms but often implemented in name only or only on a specific part of the organisation, like software development. This is especially true in action-oriented and busy organisations as they lack time for learning and sense-making.

At the same time, industry reports like *Accelerate State Of DevOps Report* signal a multithousand fold difference between low and elite performing companies when looking at lead times (Smith et al., 2021). On top of that, many case studies say that one can improve productivity by 35% by adopting a particular framework like Scaled Agile Framework (Scaled Agile Inc., 2020).

Top and senior leadership or management is often quite sharp and can sense that something is not right if the company's performance does not match what is being said by reports and

industry analysts. Sooner or later question is asked in most faux-agile companies: "How can we deliver faster?"

This work exists to answer that question in the context of one company. At the same time, countless implementation guides and management fables exist about subjects like how to implement Scrum, SAFe or any other framework (Kim, Behr and Spafford, 2014; Gene, 2019). But there are no holistic playbooks to help answer the question "How can we deliver faster?" and how to facilitate the change or transformation to deliver faster or, as we later find, learn more quickly.

#### 1.2 Project overview

#### In this work we:

- 1. Look into how the project was initiated at TC1
- 2. Review the existing TC1 generated material around the problem domain
- 3. Define the method for problem definition and solution generation
- 4. Walkthrough of the project execution step by step
- 5. Assess the proposed changes to TC1's agile transformation
- 6. Investigate the outcomes and results

#### 1.3 Approach

This work uses a project approach in finding main causes for perceived delivery slowness and running series of workshops to discover required changes to our virtual organisation structure, roles, processes, and overall way of working and align a cross-functional group of people to mandate, lead, drive and execute the required changes in various parts of the organisation.

Key Performance Indicators and underlying data on delivery performance will be used to understand the problem from a numbers point of view. Workshops and debriefing sessions will be used to create a shared understanding of the problem to be solved and devise countermeasures to the chosen set of problems.

The smaller set of changes or experiments to drive the change will be introduced to all areas in ten-week planning cycles called Program Increments (PIs). The author worked as a Lean-Agile Coach supporting the whole TC1s organisation and Release Train Engineer in one of the delivery areas, responsible for introducing and coaching the implementation of chosen changes in his area. Tactical and operational changes are then inspected through this lens.

#### 1.4 Company background

This section looks at how our target organisation is structured from both line organisation and virtual organisation points of view and explores how the scaled agile framework was set up at the start of this work mid 2020.

As mentioned this work is set in the context of one telecommunications company in Finland, later mentioned as "TC1". TC1 has lifted the agile way of working as one of its strategic core competencies. Most development work is organised following Scaled Agile Framework (SAFe) (Scaled Agile Inc., 2021g) model. First team level agile experiments at TC1 were conducted in the early 2010s. Around 2017 company adopted SAFe as the primary way to scale the work of agile teams and as a mechanism to adapt from project-based work to Product (continuous development) style work.

TC1 has two layers of organisation: the line organisation, which is the official organisation that the company management model recognises. Everyone at TC1 has a line organisation job description, and often some level of work also originates from this line organisation department or team. The virtual organisation then pulls people all around the company to work together in a cross-functional manner.

# 1.4.1 Line organisation

The line organisation of TC1 is split into four major organisations; two of the organisations are business organisations, one serving Business-to-Business and the other Business-to-Consumer, and the remaining two are split between Information and Communications technology (IT) and Technology. The difference between IT and Technology units is somewhat vague in some areas, but Technology mainly works with networking; fibre, coaxial, wireless, and radio networks. In comparison, IT works with information systems and development tooling related matters.

The way TC1 is organised means that same line organisation teams and people, especially from IT and Technology organisations, often develop, maintain, and run systems that serve both business organisations. IT and Technology are shared capabilities for the two highly independent business lines. Business-to-Consumer organisation are also organised into two business lines.

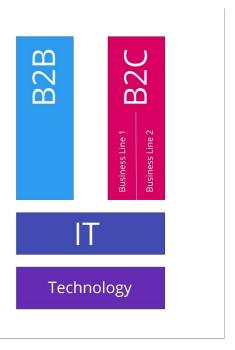


Figure 1: TC1 high-level organisation

Further complications are created because every organisation has its own top #1 priorities without holistic portfolio management of enterprise initiatives. People in different line organisations have incentives tied performance of their organisation, making priority negotiations hard, and sometimes people defer from having such discussions at the management level instead of delegating the burden to people doing the work.

#### 1.4.2 Virtual organisation

TC1's virtual organisation model borrows many ideas from John P. Kotters (2012a) work first published in 2012 Harvard Business Review article called Accelerate - "How the most innovative companies capitalise on today's rapid-fire strategic challenges - and still make their numbers". While Kotter's model leans on pulling volunteers from the incumbent hierarchy inside a company, TC1's approach has been a mix of assigning people and inviting enthusiastic people. Kotter's approach also uses the virtual organisation approach to institutionalise the changes made to the organisation and its ways of working into a culture. One of the big incentives for the virtual organisation approach is to avoid lengthy and anxiety-inducing co-operation negotiation processes when changing where and whom with people work. (Kotter, 2012a)

TC1's virtual organisation is currently tasked with organising the work around systems and service development. There are eight virtual organisations called Agile Release Trains in SAFe vocabulary (Scaled Agile Inc., 2021a). More about TC1s Agile Release Trains in the next section.

TC1s goals for these virtual organisations are to change the traditional project paradigm where people are brought near the work i.e., project and people allocations to create stable, cross-functional teams and teams of teams that work together, learn together, and where work is brought these teams (Kersten, 2018).

These virtual organisations pull people from all over the line organisation to these long-lasting agile release trains and teams inside the trains. SAFe framework instructs one to create virtual organisations that can, as independently as possible, develop a solution and release value to customers (Scaled Agile Inc., 2021o).

Further TC1 currently sees virtual organisations as way to create agility in the organisation as changing the line organisation often affects people's work agreements, and that requires initiating co-operation negotiations which are not only slow but can instil fear, uncertainty, freeze the status quo and attract the attention of various unions. Whereas changing the virtual organisation does not require changing any of the mentioned matters, one can change that organisation as often as needed.

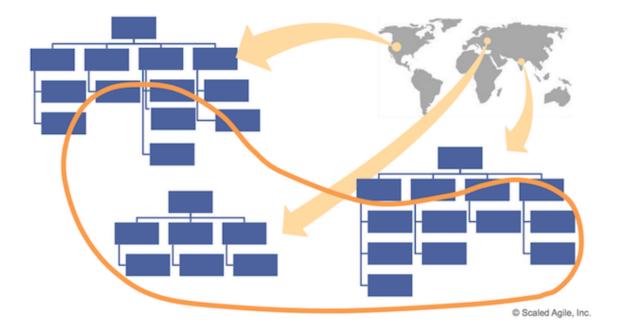


Figure 2: How SAFe virtual organisation pull people from line organisation

TC1s virtual organisations before the work was conducted were almost 1:1 mapped with the layer architecture of the information systems.

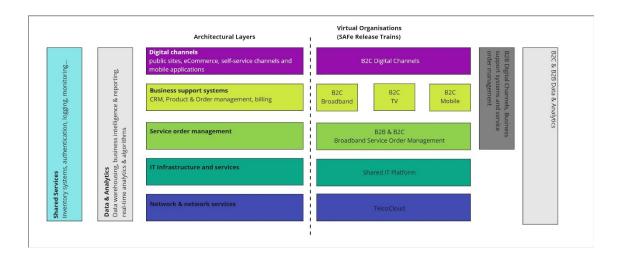


Figure 3: TC1 virtual organisation topology

# 1.4.3 Agile release train at TC1

Agile release train (ART) is a SAFe term for a team of agile teams, and ART is the primary scaling method in Scaled Agile Framework. ART has one backlog called "Program Backlog" for all the work larger than a single team or a single two-to-four-week timebox.

ART has three speciality roles working with the agile teams. The roles are Product Management, Systems Engineering/Architecture and Release Train engineer. These special roles are similar to what is found on the team level agile teams (Product Owner, Development Team and a Scrum Master); even the role responsibilities are the same with the difference that the domain of responsibility covers the whole ART domain (Scaled Agile Inc., 2021g).

While Program and Team backlogs form a type of a parent-child hierarchy, it does not mean all the work in the ART comes top-down for the teams. Product thinking in general and SAFe point out that great ideas originate everywhere and not always from the top or from the designated roles, like product management (Kersten, 2018; Beattie, Hepburn and O'Connor, 2021; Scaled Agile Inc., 2021b). SAFe calls this bi-directional process "Innovation Riptide" (Scaled Agile Inc., 2021e).

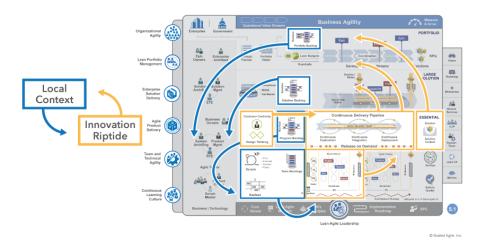


Figure 4: SAFe Innovation riptide

The different backlogs in SAFe hold items of varying abstraction levels; SAFe also gives prescribed guidance on backlog item sizes. The main idea behind this is to create a system that makes the work visible and presented in the right level of detail to the right stakeholders. It also tries to create a system where ARTs and teams are better able to prioritise and order items as they are roughly similar in detail granularity and work size-wise.

The framework (SAFe) gives guidance that ART should contain roughly 50 - 125 people, forming around five to twelve agile teams. Further, one team member should be dedicated to a single agile team belonging to a single ART (Scaled Agile Inc., 2021i). ART should pull all the required persons from the organisation to release value independently. For example, define a problem to be solved, define, design, build, test, deploy and go to market with a software product (Scaled Agile Inc., 2021a).

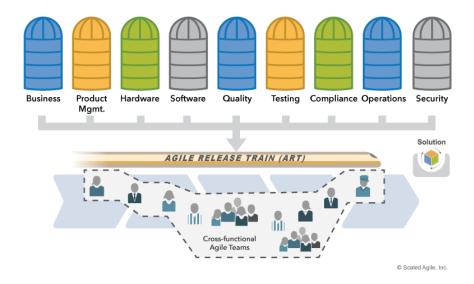


Figure 5: Agile release train pulling people from different departments

Three of eight TC1s ARTs fit this guidance, the rest having either members shuffling between multiple ARTs, traditional projects, or line organisation work. In later chapters, we will investigate the implications this type of resourcing created for the organisation.

Teams belonging to various ARTs exhibit various degrees of autonomy; while some teams can almost always decide on how the given problem or initiative is best solved, some teams are handed fixed scope assignments with fixed "how". Teams, or "Agile Teams", as SAFe puts it, should also possess all the required skills and capabilities to define, build, test & deploy a piece of the solution ART is responsible for (Scaled Agile Inc., 2021b).

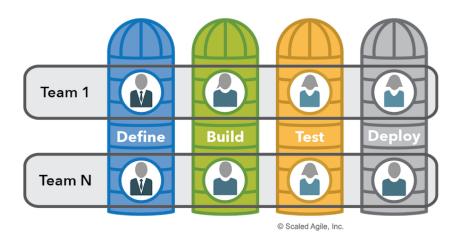


Figure 6: Agile team composition

While above is the guidance, at TC1, many speciality roles like Service & UX Design were central teams or speciality teams at ART level. Teams got support either via ticketing or short-live embedding. In all the cases, more teams depended on these special services than people providing said services, so instead of teams being autonomous units with the capability to develop part of the solution independently, there was an additional layer of dependencies to be navigated.

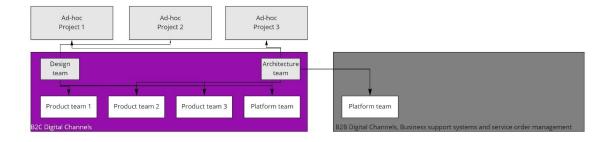


Figure 7: Dependencies created by centralized teams

# 1.4.4 Team types

TC1 has a wide variety of teams ranging from working groups or group people forming oneperson teams to fully-fledged, long-lasting product teams<sup>1</sup> with clear missions, boundaries, and goals. The major team types are as follows:

- Line organisation team
- The fully dedicated virtual organisation team
- Virtual organisation working group
- team, as defined in PowerPoint slide

Line organisation teams are primarily organised around capability, for example, business management, product management, agile coaching, databases, and monitoring.

Fully dedicated virtual organisation teams pull people from line organisations with the goal of forming a product team. The work is then performed inside this team, and the line organisation is responsible for people development, administrative items like vacations, and places to share learnings.

A virtual organisation working group on paper is a team in a virtual organisation with the distinction that the group comes together to plan and synchronise work. Still, actual work happens in line organisation teams.

Team, as defined in PowerPoint slides, is the most complex entity to manage and understand. In practice, most often, this team is a mixture of previous teams. Work management is complicated as often it gets managed in many ways based on where the people actually work, in line organisation, virtual teams or virtual working groups. Often it is unclear who is responsible for facilitating the progress and when something might be ready for others to act on.

<sup>&</sup>lt;sup>1</sup> TC1 does not have an official definition for product team, going forward Product team means the team has all the required capabilities and skills to take full end-to-end (idea to production) ownership of the area they develop on.

#### 1.4.5 Program Increment cadence / Planning window

Planning window or Program Increment (PI) cadence as its, better known at TC1, is a tenweek period split into five two-week timeboxes called sprint<sup>2</sup>. SAFe's intention with PI cadence is to synchronize multiple teams working with the same solution to the same timeboxes to encourage early and frequent integration between teams. Integrating work from numerous teams early and frequently speeds up the feedback cycle and reduces the risk of creating parts of the solution that do not work when put together with other parts (Scaled Agile Inc., 2021m).

Before mid-2019, TC1 had eight ARTs, all running in different cadences, PI's start & end dates and sprint timeboxes differed. This created significant planning problems between ARTs; some ARTs did not know could they expect other ART to complete or even start to work on shared objectives if the other ART was planned a week after the first one. Checking and demonstrating cross ART initiatives was deemed problematic because some ARTs were done, and others could still work on their parts of the initiative. This was directly against the core principles of the chosen framework (Scaled Agile Inc., 2021n) and the late discovery of defects and problems created a lot of surprises and unforeseen events for business stakeholders.

TC1 then decided mid-2019 to shift all ARTs to the same cadence, and from there until at least the end to this project, all the ARTs share the same start and end dates, even for the program increment planning. This means roughly 650 people always plan their 10-week planning window together over two days. Currently, the feedback is that it has become easier to understand what can and cannot be achieved during the PI window.

#### 1.4.6 Main metrics

Metrics at TC1 are separated into two categories, line organisation metrics and virtual organisation metrics. Line organisation tracks the commercial performance incl. Costs, customer satisfaction and average revenue per customer. TC1's metrics measuring its virtual organisation performance revolve around four key metrics:

- Change lead time
- Program Predictability
- Number of deployments
- Amount of late-stage software defects

<sup>&</sup>lt;sup>2</sup> Sprint is the original term for two-to-four-week times originating from Scrum framework, Scaled agile framework calls it an iteration (Scaled Agile Inc., 2021j) ,while TC1 follows SAFe this term was kept intentionally in its more familiar format.

It is important to note here that while all the metrics above only focus on the system's capability to create value, the metrics that measure what type of a value was released are attached to the work items of each ART and the team in ARTs is tasked with. More often than not, these metrics are tied to line organisation KPIs.

Change lead time measurement was created to measure the development organisation's performance. It measured the time from when need, desire, requirement or initiative hit the hands of a development team to the time the initiative was deployed into production. This measurement had a couple of problems:

- Measurement was taken from the issue management or work tracking system, which
  meant that the measure was only as accurate as people using the system
- It did not measure the fuzzy frontend or sometimes long-tail from deployment to release

Change Lead Time measure showed a median score of 17 days for the whole TC1. Sprint length at TC1 was two weeks, and the score means that most teams could not keep the majority of the work inside sprints confines.

Program Predictability is a straight copy from Scaled Agile Frameworks. Program Predictability measure calculates the percentual difference between planned business value and actual business value of an Agile Release Train's committed and uncommitted PI objectives (Scaled Agile Inc., 2022a).

Program Predictability score was 78% at mid-2020, and the official goal is to have that score between 80% to 100% at TC1 same guidance was used.

Number of Deployments measured the total number of production deployments made inside a year from all TC1s systems. The measurement had a couple of problems:

- A couple of systems and agile release trains had reached the capability to deploy multiple times per day and were improving rapidly.
- Around three tracked systems formed the over 80% of the measurement
- Not all of the TC1s systems were tracked. Improvement could be made just by connecting more systems tracking system
- Measurement was not taken in the same way between systems, and some took it from the work tracking system and other from the actual delivery pipeline, the Latter being more trustworthy.

The number of deployments was around 3200+ when interpolated to cover whole year mid-2020. Late Bug Count measured the number of bugs found in user acceptance testing environments or the production. Measurement had similar problems then the number of deployments.

Late Bug Count did not have proper measure mid-2020 as it was just added to the measurement model in August 2020.

#### 1.4.7 Shared services at TC1

A small but critical set of teams and services are called shared services at TC1. Usually, this means that many teams use the service, and there is a heavy functional specialisation with people developing and supporting that service. The service's work is often handled in separate backlogs, pulling work from many teams and internal initiatives. This leads to a single team having to mediate multiple number one priorities into a sensible list. This action almost always creates issues for the teams relying on the service and its development.

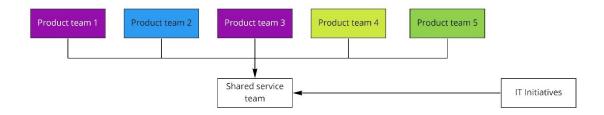


Figure 8: Shared service team as the pace setter

Primary way to negate this problem looks a lot like a virtual organisation working group -team from the Teams chapter. Most ARTs have one person acting as a liaison between the ART and the shared service team. While the communication and understanding of priorities for shared service team has drastically improved by doing this, it is not without problems.

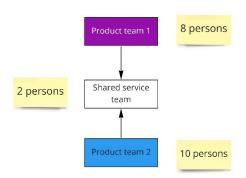


Figure 9: How shared service teams can form a bottleneck for value delivery

TC1's shared service teams staffing levels currently cannot meet the demand created by teams in different ARTs. Understaffing establishes a system where the few shared services teams can set performance, especially velocity for the whole system.

# 2 Background

We will then investigate main components of the framework and what existing research and literature says about the principles, practices and expected outcomes detailing each component.

#### 2.1 Lean

Throughout the project we kept hearing from various people statements like: "let's make this lean" or "we need to lean this out", but what making something lean truly means, what is lean?

(Modig and Åhlström, 2016) said it very well in their book that it is not any one thing that exists or a thing you can do and be successful. (Rother, 2010; Liker and Convis, 2011; Humble, Molesky and O'Reilly, 2015; Modig and Åhlström, 2016) tell the same story, lean is first and foremost a set of humancentric values, ingrained beliefs akin to philosophy one follows in everything one does.

As mentioned by (Modig and Åhlström, 2016) lean does not have an end-state, instead one chooses to believe that there is always a better way to do it. A way that respects the customer, respects the employees and is good for the business and surrounding community. Something "Lean" first and foremost tries to make sure everything we do is value for the customer and while pursuing that, it is made sure we create a system where it is good for people to work in, that it stimulates the innate need of people to be good at what they do and is safe for all parties. DANA's operating model one pager from (Liker and Convis, 2011) book Toyota way to lean leadership is good example how lean look in real world.

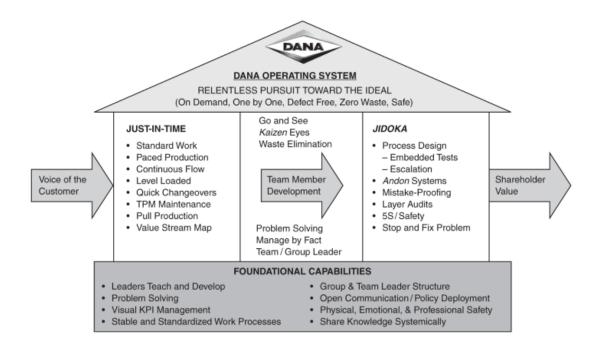


Figure 10 DANA operating model (Liker and Convis, 2011)

In closing, lean is not only focusing on lead time and reducing waste from the process, it is about creating organizational capability to define which are the problems that affect company or its area reaching the state where it needs to be in order to satisfy customer, employees, company and community and being able to come up with own practices propel the company where it needs to be, instead of copy-pasting best practices from others (Humble, Molesky and O'Reilly, 2015).

# 2.2 Agile

It has been twenty-one years since the inception of agile manifesto by group of leading software developers, the manifesto consists of four simple statements and twelve principles stating that avoiding toil, working software, embracing change, keeping things simple and daily cross-functional collaboration are more valuable than heavy-weight processes and bureaucracy when software development is in question (Beck et al., 2001.)

In those twenty-one years the term agile has become an umbrella term that can contain everything from pure rebranding of old behaviours, structures and management ways to how HR organisations should run (McMackin and Heffernan, 2021) and beyond. This is to say that agile in 2022 is so much more than software development, the trend that is not entirely seen as positive. More and more companies start to have software at their heart (Holmes, 2019), while part of the companies keep improving the technical agile practices while extending out to different non-software areas of the company, there are still misguided enterprises (Smith

et al., 2021) that have bought in to the hype and have missed the foundational work. This has created something which is called an agile success theatre or faux agile (Denning, 2020).

Next, we are going to investigate the different flavours of agile that are in the wild today and especially at TC1.

#### 2.2.1 Scaled Agile Framework

The scaled agile framework (SAFe) is a commercial agile scaling framework that introduces itself as: "SAFe for Lean Enterprises is an online knowledge base of proven, integrated principles, practices, and guidance that brings the power of Lean, Agile, and DevOps to the people building the world's most important systems." (Knaster and Leffingwell, 2020).

SAFe also has extensive website with multitude of articles that explain different principles and practices that together forms SAFe (Scaled Agile Inc., 2022b). The articles then link into industry studies and literature as sources for the articles. Articles itself quite often bend the source material in one way or another to fit the framework and its constructs, this has created some criticism over the SAFe prescribed practices (Jeffries, Ron, 2014; Putta, Paasivaara and Lassenius, 2018; Geraghty, Tom, 2020).

SAFe does pose many benefits, specially for enterprises and companies in-process of transforming away from waterfall projects as there are a lot of material freely available at the SAFe website, something which homegrown models often lack.

Difficulties for adopting SAFe often come from time it takes to adopt the mindset and understanding why the introduced concepts work and how they might be applied in the context where SAFe is used. The model builds on top of tens if not hundreds of books, industry knowledge from multiple areas and the teachings should permeate all levels of the organisation implementing SAFe. This is often hard or even impossible in organisations geared towards resource utilisation optimisation rather than learning. Strong management mandate and role-modelling of the new behaviours, especially taking time for learning is required for implementation success (Putta, Paasivaara and Lassenius, 2018; Hesselberg, 2019; Knaster and Leffingwell, 2020; Smith et al., 2021)

#### 2.2.2 Business agility

Business agility (sometimes also known as enterprise agility) is a term describing ambitions for the enterprise to create engaged workforce where everyone from top to bottom focuses on customer value, continuously learning and improving how the company operates with practices grounded in empiricism, geared to rapidly respond to a change while acting in sustainable manner (Humble, Molesky and O'Reilly, 2015; Burlton, Ross, Ronald G. and Zachman, 2017; Hesselberg, 2019; Denning, 2020; Knaster and Leffingwell, 2020).

With latest 5.X version of Scaled Agile Framework, the business agility is the goal for the framework (Knaster and Leffingwell, 2020).

# 2.2.3 Organisational agility

Organisational agility is a term describing any organisations ability and pace to change its organisational groups, forms, and alignment to support the current customer and business needs without loss of cohesion or created chaos (Aghina et al., 2022; Scaled Agile Inc., 2022b).

SAFe has organisational agility as one of the key capabilities of a Lean-Agile Enterprise (Knaster and Leffingwell, 2020), SAFe augments the organisational agility with practices like value stream mapping (Martin and Osterling, 2014) and visualizing work via information radiators like Kanban boards (DeGrandis and DeMaria, 2017).

# 2.2.4 Team and technical agility

Team and technical agility is a SAFe key capability of Lean-Agile Enterprises, team and technical agility contains both practices how to organise as a team, to manage initiatives and technical practices for implementing said initiatives (Knaster and Leffingwell, 2020; Scaled Agile Inc., 2021b; 2021q).

Team and technical agility capability emphasises the need for having cross-functional teams that contain all the skills and capabilities to turn idea into working solution. There is also strong emphasis on need to have an available team coach to help the team improve daily practices, equally significant emphasis exists on technical skills like test-driven development, having extendable architecture and strong engineering practices (Knaster and Leffingwell, 2020; Scaled Agile Inc., 2021q).

# 2.2.5 Lean agile centre of excellence (LACE)

Lean agile centre of excellence is a SAFe construct defining a team whose responsibility is to drive the SAFe adoption and ensure its continuing improvement (Knaster and Leffingwell, 2020; Scaled Agile Inc., 2021h).

LACE has many similarities with other enterprise agile adoption methods like Agile Working Groups (Hesselberg, 2019) similarities take form of guiding the LACE to be staffed with small number of dedicated people responsible of educating, coaching and facilitating the change.

LACE team composition should follow cross-functional nature, with ideas grounded in Kotter's work around guiding coalitions (Kotter, 2012b). SAFe guidelines inform that LACE team should have C-level executive working as a Product Manager for the team as method to ensure top

management level participation and to make sure the implementation stays on management agenda with direct line of communication and guidance (Knaster and Leffingwell, 2020; Scaled Agile Inc., 2021h).

#### 2.3 DevOps

The DevOps is an abbreviation of two words Development and Operations, The DevOps term itself describes certain type of a culture tied to software development. DevOps has its roots in time where development people and operations people were kept separated perhaps misguidedly due to many regulations requiring segregation of duties. Perhaps most famous cases where this segregating stems from Sarbanes-Oxley section 404 (public law 107-204) and combination with ITIL principles where segregation of duties a cornerstone (Humble, Molesky and O'Reilly, 2015). While segregation of duties mandates that end-to-end transactions like moving business critical software to production must required at least two people or a system where the act is detected and controlled by at least one other person, it is not said that the persons could not be in the same team. Industry in the other hand acknowledged it like that there needs to be separate teams for development and operations. This lead into many dysfunctions like lack of communication by rules or teams just working from different rooms, buildings or countries (Edwards, 2012; Humble, Molesky and O'Reilly, 2015).

DevOps rose as an ideal to countermeasure this by straightening the facts, telling stories of ordinary people doing extraordinary deeds when collaborating happens between developers and operations people. DevOps term itself was coined by Patrick Debois by arranging a "DevOps Days" event in Ghent, Belgium being inspired by what he saw 2009 O'Reilly Conference. Patrick later decided to shorten the event name into DevOps and that name stuck for the whole philosophy (Edwards, 2012).

DevOps itself has no central standards body but is an ideal that develops de-centrally, this mean that there are many interpretations what DevOps means and some of them have transformed to mean something totally opposite of what was originally envisioned.

Amazon Web Services (Amazon Web Services, 2022) define DevOps as: "DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market."

Whereas Wikipedia's (Wikipedia, 2022) crowdsourced definition is: "DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high

software quality. DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology."

Already there is a drift between the two definitions, is DevOps a practice in-itself or is it a collection of practices? Some of the industry's thought leaders, such as Gene Kim, Jez Humble, Patrick Debois among some have tried to create more thorough explanations how the DevOps movement stands on the shoulders of giants. The giants being from Toyota to Eliahu Goldratt and beyond. The general notation by their work is that DevOps is way how you think about things and then practices like Continuous Delivery is what you do (Humble and Farley, 2010; Kim, Behr and Spafford, 2014; Kim et al., 2016; Kim, 2019.)

# 2.4 Portfolio Management

Portfolio management being an ambiguous term for which everyone has their own interpretation in their own context, to create clarity, in this work when use term portfolio management in direct connection to TC1 we frame development portfolio to be an entity which contains all the major development initiatives<sup>3</sup> from business and technology.

While there are multiple ways to structure enterprise or development portfolio, both (Humble, Molesky and O'Reilly, 2015) and (Scaled Agile Inc., 2021k) suggest that portfolio should contain work from different horizons (what is required by current business, growth areas in near future and exploration of future growth possibilities).

(Vähäniitty, 2006) argues that management of portfolio is utmost importance to any company wishing to move into chose strategic direction with deliberation. Management of portfolio according to (Vähäniitty, 2006) is successful when four key outcomes are achieved:

- Projects or product development work originating from the portfolio creates best possible financial value for the company
- Work is balanced between different types of efforts liken to horizons mentioned above
- Work originating from portfolio can be connected to company's strategy and aligns with the direction set by strategy
- Process connected to initiating new work from portfolio is constantly assessing how much organization can complete without getting roadblocked by having too much work in progress

<sup>&</sup>lt;sup>3</sup> Major initiative in TC1's case means something that requires work from at least two ARTs and / or takes longer than 10 weeks to complete or requires change into economic frame of the current portfolio i.e., more funding.

(Vähäniitty, 2006; Scaled Agile Inc., 2021m) both advocate for shared rhythm or cadence for events connected portfolio, starting from portfolio review meetings where it is assessed if company has capacity to start something new, review ongoing work and institutionalize learnings from outcomes created completing the portfolio originating work, all the way down to planning timeboxes on team level.

#### 2.5 Project to Product thinking

The project to product thinking connects into trend of businesses becoming more and more software driven but software is never ready now more than ever. What worked<sup>4</sup> in the past, seems to not work anymore<sup>5</sup> and "project to product" thinking tries to emphasize what needs to be changed.

In project setting the budget is often set for milestones and the scope is then pre-defined for those milestones, projects are often also fixed in duration. Success criteria is often matched with the defined budget, scope, and duration of the work. Risk management methods often are front-loaded with heavy planning, specification and decision making. People are also pulled from all over the organisation with percentage allocations, with many people often working in multiple projects at the same time. Visibility into progress is often delivered via roadmaps and status reports (Humble, Molesky and O'Reilly, 2015; Forsgren, Humble and Kim, 2018; Kersten, 2018).

In product setting the funding is often allocated for long lasting teams or in larger settings to value streams (or teams of teams). Time horizon lives with the product's life cycle but usually being multiple years. Success is measured using actual business outcome metrics and the journey is usually split into multiple self-reinforcing learning loops. Risk management is spread throughout the journey leveraging hypothesis driven development where aim is to collect data that allows the team to either validate or invalidate the hypotheses. Learnings are used to either continue, pivot, or scrap the product or initiative if product is large. Visibility is through movement of business metrics (Humble, Molesky and O'Reilly, 2015; Forsgren, Humble and Kim, 2018; Kersten, 2018).

It is said that when people work together for sustained period, own their own decisions that affect the direction of the product, team becomes higher performing than teams formed and

<sup>&</sup>lt;sup>4</sup> It can be argued if software projects ever truly worked using project management driven methods as The Standish Group reports success of IT projects as traditionally measures (on time, on budget with agreed scope) hovering between 36% and 41% between years 2011 and 2015 (The Standish Group International Inc., 2015).

<sup>&</sup>lt;sup>5</sup> Standish group also reports that success of projects measured using modern methods (valuable, on goal & satisfactory) are even lower hovering between 27% and 31% between years 2011 and 2015 (The Standish Group International Inc., 2015).

reformed for projects where tasks are handed from above (Forsgren, Humble and Kim, 2018). Psychological safety is also important for people to be able to take risks, learn from failures and admit errors (Edmondson, 1999; 2019).

The standish group has also found evidence that initiatives where "Infinite Flow" type of development way is used the success percentage is 71% compared to more traditional like projects where it can be between 26% to 44% depending on the approach (building everything from scratch to leveraging commercial of the shelf software) (Johnson and Mulder, 2021).

In scaled agile framework agile release trains, development teams and value streams try to capture similar behaviours and patterns that can be found from product thinking model (Knaster and Leffingwell, 2020; Scaled Agile Inc., 2021a; 2021b).

# 2.6 Leadership vs management

Difference between leadership and management in some languages like Finnish is non-existent but the difference in managerial behaviours can be like day and night. John P. Kotter (Kotter, 2012b, p.28) argues that successful transformations are 70% to 90% leadership and only 10% to 30% percent management.

Kotter says that leadership differs from management so that leadership is about showing the way by developing a compelling vision and set of strategies that produce the change to achieve the set vision. Whereas management is about planning, budgeting, organisational design, controls, and problem solving (Kotter, 2012b.)

Many organisations can be at odds when introducing a more agile model of working, where teams are expected to take large portion of traditional management tasks to be handled inside the team (Adkins, 2010; Syväjärvi and Vakkala, 2012; Humble, Molesky and O'Reilly, 2015; Hesselberg, 2019, p.143). Some managers might feel uneasy or anxious of such changes in the operating model and if concerns go unaddressed or unnoticed natural course of action can be trying to solidify the status quo and keep the existing power structures regardless if there is any real benefit of doing so (Kotter, 2012b; Tökkäri, 2012; 2019).

# 2.7 Organisational Change

In its essence organization change is about changing people's behaviours as much as it is changing the visible parts of an organization, structures, processes, tools and outputs (Kotter, 2012b; Syväjärvi and Vakkala, 2012; Gibbons, 2019). Throughout the literature the importance of compelling vision that targets both the heart & mind is mentioned as one of the central pieces of any change and transformation initiatives (Rother, 2010; Liker and Convis, 2011; Junger, 2012; Kotter, 2012b; Pichler, 2020), but creating a tangible and urgent

reason to change backed by cross-functional group of people providing guidance throughout the initiative is even more important (Kotter, 2012b; 2012a).

Kotter continues that next step would then be communicating the vision followed by empowering of the actors to do even the risky and non-traditional things to get closer to the vision. Kotter also mentions about the need to generate short term wins to create early valuable outcomes that enforce the credibility and benefit of the change to allow more sweeping changes down the line. Kotter ends his model with stage for solidifying and codifying the changes into culture (Kotter, 2012b.)

While the above sounds straightforward and easy, the reality in many organisations can be totally opposite, Tökkäri (2012; 2019) states that there exists a chasm between the experiential and planned organisation, that perception of organisation is created by people working inside and outside the organisation through stories and narratives, coloured by current and past experiences of people. Organisational change from this perspective requires a lot of co-working, dialogue and interaction between people on all levels in order to create a shared perception of the organisation and the wanted direction (Tökkäri, 2012; 2019).

Haslam, Reicher and Platow (2012) show that there exists inner and outer groups between teams or groups of people where leader or change agents needs first to be deemed beneficial for the group in order to gain any influence over the group. Belonging into inner group is seen as a must as otherwise the only option to create change is to use the law-given powers of authority which has quickly diminishing effects and high-chance of moving back to the old ways as soon as the leader or change agent is gone or challenges are faced.

# 3 Project methods

In this chapter we detail the used background information and methods used in the organisational change project called "Railyard 2.0" which forms the basis of this paper. In section 3.1 we look into background how this project came to be, what previous work and data was used to guide the development of railyard 2.0 targets. Section 3.2 provides more information about the problem to be solved and contributing factors of the problem. Section 3.3. is about explaining rationale and ethos of workshop design principles used in railyard 2.0 project. Lastly section 3.4. details the target outcomes set for railyard 2.0.

#### 3.1 Background

As mentioned in section 2.1.6, while success metrics looked favourable there had been a growing dissent on senior management level against couple virtual organisations or agile release trains (ARTs) as they are called at TC1. On top of this many other parties struggled with couple of repeating problems: unclear or lack of ownership, uneven technical practices, and misaligned goals where initiative was required from multiple ARTs. While the former manifested mainly in steering groups and 1:1 talks, the latter created initiative called "better tracks for our trains" workshop series. The series was co-led by author, together with one other agile coach from our shared home team.

"The Better tracks for our trains" -workshops were held inviting cross-functional group of people together, representing product management, tech management, quality management, agile coaches, and heads of agile development. The group identified four key themes on how to improve both commercial (customer outcomes) and non-commercial (software development) performance. Themes identified:

- Lack of development portfolio management cross agile release trains; proposed countermeasure: create systematic way to introduce, analyse and decide which large initiatives get greenlighted, which must wait, and which get no-go decisions
- Lack of shared definition of done; proposed countermeasure: define principled standard of what good looks like to ensure quality outputs
- Confusing & error prone development to production flow when changes must be done
  on multiple levels of enterprise architecture, proposed countermeasure: define and
  rollout principles of development to production flow and ensure implementation help
  reaches all of the teams
- No cross-ART guidance or oversight on end-to-end testing, proposed countermeasure: organize quality organization cross ARTs, improvements to test-data management, define initial good practices for process.

Workshop group unanimously agreed that the most impactful contributing factor for delays and poor quality was the lack development portfolio, groups visualisation of stages and actions of proposed flow is inspired by what SAFe suggests (Scaled Agile Inc., 2021k).

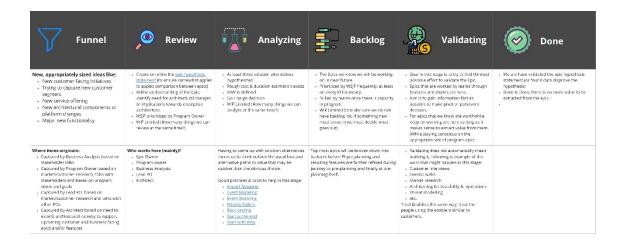


Figure 11: Initial idea of development portfolio flow

Another major outcome of the workshop was the formulation of principles and initial good practices of how to change development to production flow to be more resilient, guarantee better throughput and decoupling cadences between different systems.

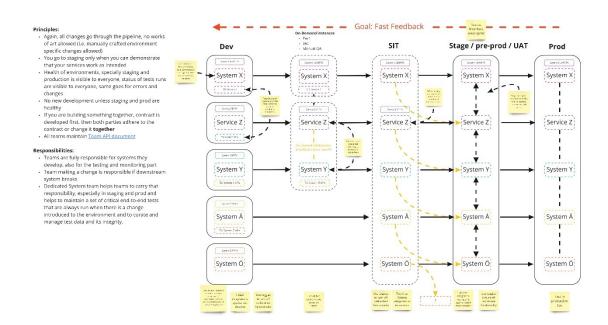


Figure 12: Initial development to production flow principles

Unfortunately, the product of this workshop while garnering universal nodding in senior management circles, did little else than creating useful knowledge base of articles, drawings, and ideas on how to improve the situation. Later it was revealed that this workshop series for

instrumental in planting new ideas to heads of some influential people. Instead, the business side senior management frustration and power games culminated to state where in one meeting late 2020 it was asked that: "do we need to re-organize or can we explain why "Online" -development is so slow?"

That question became the mandate behind "Railyard 2.0" -initiative and the basis of this work. From the previous work that was done little earlier we already knew that re-organizing would not solve the problem or at most local optimization could be done but that would hurt the other side of the organization. As author and author's team was responsible of the whole organization's agile transformation and improvement of the whole, taking the most obvious answer was out of the question. Decision was made to use this as the golden opportunity to make more systemic changes with aim to making the whole TC1 organization benefit.

#### 3.2 Problem to be solved

In its core the problem was quite straight forward, the pace of development is too slow and therefore the project is about assessing why that is, develop countermeasures, test countermeasures, systematise the ones that work and iterate in rapid fashion for as long as the problem exists.

All that would be somewhat easier if there would be any top-down ownership, guidance, and responsibility of the chosen ways of working, i.e., the scaled agile model. TC1 chose to go with purely bottom-up way with heavy de-centralization and only very loose coupling presented as most Release Train Engineers (RTEs) belonging to same Lean Agile Centre of Excellence (LACE) team. This meant that no chance could be done, unless we manage to invite, influence, and convince the people in either formal or informal leadership positions to buy-in to the proposed change.

In the other hand we also needed to make sure we (as company) don't succumb into stressed decision making model (Old, 2015), where senior management would just pick the initial idea of re-organizing to get out of the psychologically loading situation.

Lastly it was identified that we needed to create a shared understanding among the participants of the factors contributing to overall slowness of the development system, whether it be only perceived slowness or true underperformance. The organization, its people, systems we develop and the actual initiatives after all form a complex-adaptive system or sociotechnical system where it is not that straightforward to model impact of change, or to even explain why the system work the way it works.

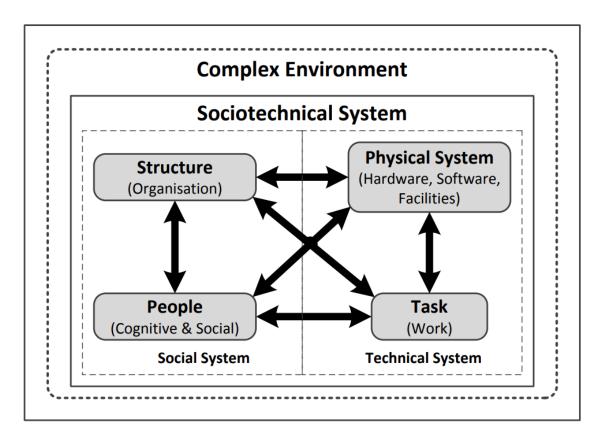


Figure 13: Interaction in Sociotechnical systems (Oosthuizen and Pretorius, 2016)

# 3.3 Workshop design

The project was structured to consist of 4 workshops between mid October and end of November 2020. After couple of changes and surprises the work consisted of 9 workshops each tackling different parts of the current virtual organization and its inner workings. Planned workshops and the agendas of each of them:

- Workshop 1: Target setting, where it shows if we are successful with the organizational development?
- Workshop 2: Target setting, designing key metrics
- Workshop 3: Who is impacted of the change and what kind of behavioural change is required to from each party for us to reach the goal
- Workshop 4: Designing experiments to create wanted behavioural changes
- Workshop 5: Needed changes to supporting structures via shared principles and practices
- Workshop 6: Clarifying roles & responsibilities of key actors
- Workshop 7: Organizing agile release trains
- Workshop 8: Organizing supporting structures
- Workshop 9: Creation of action plan

Every workshop was designed to last for two hours with pre-reading materials and asynchronous work between workshops. Every workshop was also followed by 30-to-60-minute catch-up session for the ones that could participate and for the ones we wanted to stay informed but did not expect active participation. It was planned to have two workshops per week and finish the "Railyard 2.0" project by first week of October 2020.

Due to everyone working 100% from home, it was decided to use virtual whiteboard product to create persistent digital working area that allowed everyone to see the same things that everyone else saw and to be able to simultaneously work together with the materials.

Invitations were sent to all VP & director level people working or benefitting from the work of agile release trains (ARTs) around the organization. This decision was made because people who can authorize changes to where people work or how people work were the ones needed to agree on proposed changes. The same people also needed to hear from each other what might be the outcomes or impacts if we would do change to this or that direction and then agree types of trade-offs, they are willing to make for benefit of the whole system. While the main reason was that we needed the people in authority roles to OK the plan as there wasn't any central party that or even wanted to dictate this type of a change. Secondly it was designed this way to put the ownership and responsibility of the change to hands of the right people.

Invitations were also sent to every key subject matter expert representing wide variety of disciplines from architecture to development and product management, also key participants were all the Release Train engineer role holders as they are the ones that hear and see what is happening on the team level and in daily work of multiple roles.

In total workshops had 30 participants, including author and the after sessions had even more.

Workshops were designed to follow three stage design (open, explore and close)<sup>6</sup> where open -stage is used to set the stage, to introduce the theme to be worked on, clarify motivations and agendas, and get the people started with an ice-breaker exercise of two. Explore -stage functioned to extract insights, observations, desires, frustrations, reveal conflicts or misalignment, and provide first-hand information from the participants regarding the workshop agenda. Ending with the close -stage which worked to group similar ideas and synthesise the ideas into one or more workable items that whole group can stand behind.

<sup>&</sup>lt;sup>6</sup> Diverge, emerge, converge -stages as core part of game design as made famous by Game storming book by (Gray, Brown and Macanufo, 2010).

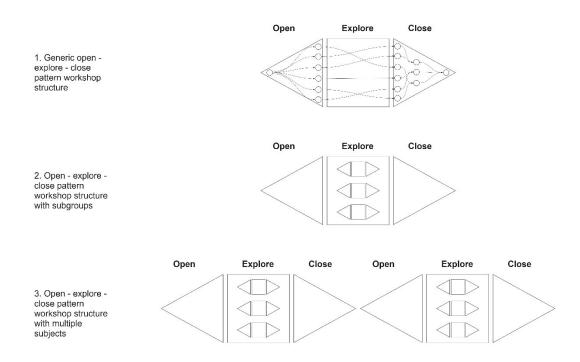


Figure 14: Different Open-Explore-Close combinations

Workshops #1-4 were designed to leverage subgroup breakout -pattern due to number of people invited to the workshops and workshops #4-9 were designed to be leverage pattern #3 as described in figure 14.

Author acted as key facilitator and directly responsible of creating all the workshop materials and researching the supporting materials, articles, books & videos that opened each of the workshop topics. Workshops itself had 2-3 co-facilitators whenever breakout rooms were used to split the group to work on some topic as smaller subgroups.

# 3.4 Target outcomes

Key deliverable for the group and especially agile coaches<sup>7</sup> from TC1s management team perspective was a proposal of changes required to a) how to organize online domain development to speed it up or if not re-organizing then plan how b) speed up development in online domain.

<sup>7</sup> Author worked as enterprise lean-agile coach, a small team consisting of 8-10 coaches of various backgrounds. Team is called LACE -team at TC1. LACE comes from Lean-Agile Center of Excellence (Scaled Agile Inc., 2021h). All the coaches also worked in release train engineer role in one of the TC1s agile release trains. Line organization responsibility of the agile coaches' home team was Agile Development (IT), hence the responsibility to create the key

deliverables.

Secondly target for the workshops was to align on what were the things that required changes, what type of a changes and then create and schedule experiments to tackle each area.

# 4 Facilitating the workshops and creating the change proposal

This section details the work regarding facilitating organizational change in bottom-up driven agile transformation at TC1. The work is detailed in chronological order, presenting all the insights, challenges, surprises, and problems materialized by the people trying to agree common principles and shared ways of working where decision making power is heavily decentralized.

## 4.1 Redefining the problem

Immediately following the initiative 1 launched by TC1's management team mentioned at the end of section 3.1 LACE -team along with author started put together material to assess if the reorganization ask was feasible and viable.

Targets for feasibility as set by LACE were:

- People would not actively work against the change and key people would not leave the company
- Current agreements with partners and vendors won't actively impede the change, i.e., agreements allow responsibility to be shared and multiple vendors work in the same area and there are no output driven bonuses or sanctions
- Development methods support collaboration or up-skilling in short-term is possible
- Key business owners support the plan

Viability targets were set by LACE as:

- Less dependencies than currently between teams
- Overall system throughput and capability improves
- Number of meetings would be reduced from current number
- Reduction of Time to market possible

LACE -team set out to scope reorganization possibilities looking through these lenses. Quickly after drafting initial set of requirements what every agile release train after any proposed

change would need to possess and drawing two of the most popular<sup>8</sup> options for re-organizing the TC1's agile release trains it became evident that re-organizing the virtual organization would not solve the problem but make it worse.

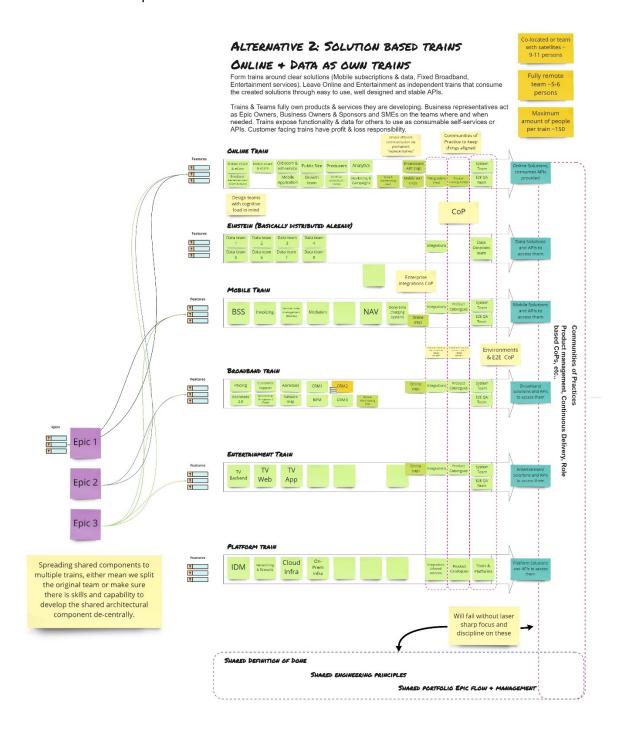


Figure 15: First alternative for re-organising TC1 agile release trains

<sup>8</sup> Popular as ideas proposed by members of the management team on frequent basis and ideas entertained in the organization from time to time.

It was found out that alternative number one, which would meant dissolving the current online development ART into pre-existing more business line aligned ARTs would have created a need to create additional groups to make sure underlying architecture, user experience, technology choices and development methods would stay intact. On top of that key message from the interviews of key architects and lead developers of online ART was that the current architecture did not really support developing and releasing pieces of functionality separate from others. The Online ART was also currently tasked with large eCommerce platform upgrade constraining available resources to support business lines, the proposed reorganization would create so much additional overhead costs that there would be even less available capacity to support business lines that before.

Pros	Cons
Less dependencies between trains	De-centralized development of single architectural component requires cross team collaboration
Ability to deliver complex end to end solutions	Too early decentralizing digital business development, do we have the right skills, good enough ways of working, capabilities to support de-centralized development of platforms that need to have intact UX?
Requires collaboration between different business and technological areas	Requires collaboration between different business and technological areas
Capacity and capability of the system is more visible and easier to understand	Can hide the fact that we still need a lot of effort on CI, CD & release on demand capabilities + engineering and quality practices associated
Looks good on paper	Care must be taken to avoid turning into a feature factory
Creates huge need to truly decouple architecture, teams and processes via services and events	Enabler work / architectural refactoring -> what backlogs to use to drive it?

Solution specific responsibilities and key- personnel clarified	Dependencies are spread just in another direction, unless architecture and processes support independent deployments and releases
	Creates huge need to truly decouple architecture, teams and processes via services and events
	Are there enough synergies between B2B & B2C?

Figure 16: Evaluated pros & cons of alternative one

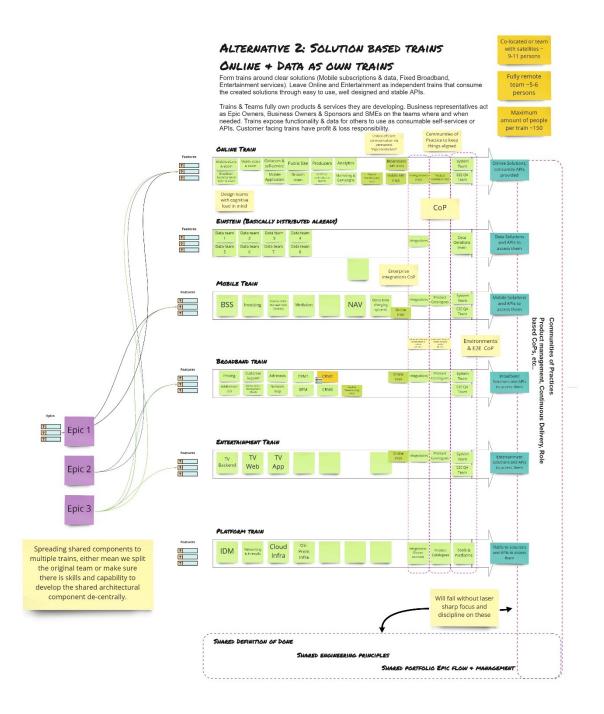


Figure 17: Second alternative for organising TC1 agile release trains

Alternative two instead focused on lack of portfolio management, shortcomings of ART organizational structure, development methods and not acted upon shared principles. Backed by the work done with "The Better tracks for our trains" -workshops there were already good understanding among some more technologically savvy people of what the real problems were. Together with LACE -team we took this opportunity to present it to wider audience and whole TC1 management team as serious opportunity to create true change benefitting the whole company and not just single business line as mentioned in section 3.1. Our thinking at the time supported the idea of improving a) organizational initiative intake, known as (agile)

portfolio management, b) improvement of development methods of all ARTs, c) improvement of our own quality management<sup>9</sup> practices and d) cross-agile release train collaboration practices. Our hypothesis for alternative two was that if we at TC1 were successful in limiting the number of large initiatives we start, able break them down together and put effort in Application Programming Interface (API) design the system would be more capable of getting things done, which would directly influence the capabilities of all business lines.

Pros	Cons
Keeps digital business development in one team of teams, makes it easier to move it to wanted direction holistically which in turn makes it easier to first build the right skills and capabilities before de-centralizing it	Even though it might be good for the systems and customer to say no or not right now to some initiatives, is it politically allowed and safe?
Product & Solution ownership is clearer	Requires more communication effort to align plans cross trains
Less prone to feature factory anti-pattern	Less matrix structures, more cross train dependencies, more pressure to design & develop good APIs
Solution specific responsibilities and key- personnel clarified	Looks bit more complex on paper
Makes it clear that we need to align & come up with better ways to create shared understanding	Cross train work requires us to find better team to team ways of working
Forces us to make choices, using Cost of Delay and systems thinking mindset (we will never have enough capacity to do everything we want)	Might be harder to make sweeping changes to way of working and engineering practices> Shared DoD, Test First development, API Design requirements,

\_

<sup>&</sup>lt;sup>9</sup> Quality management practices often look better on the outside than inside, often the production is quite defect free but the road to that is often too long and there are lot of hassle during development time. At TC1 at one moment we estimated that about 50% of development time was spent dealing with surprises and inconsistencies between environments of integrated system, all which are actions that should have never happened in the first place as mentioned by (Humble and Farley, 2010, p.85).

Pros	Cons
	unlock data and functionality via events &
	interfaces.
Assembles the people and systems required to develop broadband solutions together	
Integrates Stock & Warehousing and Service order management tighter to Mobile train and set the direction to think integration APIs first.	
Less matrix structures	
Cross train work requires us to find better team to team ways of working	

Table 1: Evaluated pros & cons of alternative two

The team presented the finding to TC1 management team the following week together with questions and broader findings based on interviews and previous data. As TC1 been leveraging Scaled Agile Framework since 2017, one could easily believe that the fundamentals and core requirements of the framework would have permeated the organization by 2020. Team chose to ask from management team that do we act according to the core values of the chosen framework and if not, it might not be a surprise that TC1 has such problems as long time to market, development is slow, be it perceived or not.

# What these mean for us?

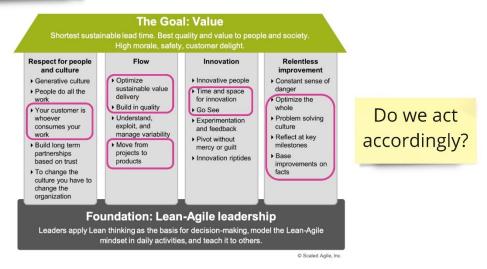


Figure 18: Questioned core values of TC1 agile scaling framework from presentation material

Picture seen in figure 18 sparked quite profound discussion among management team peers, based on that discussion it became clear for the management team and for the LACE team that there is much to be gained from trying to create common understanding around these principles. At the same time, it opened the door to go to more controversial topics, talking about problems created by this lack shared of understanding.

LACE team then went on to suggested that the problem might not primarily lie in how agile release trains are organized but how systematically TC1 had implemented the chosen framework<sup>10</sup>.

<sup>10</sup> While implementing a framework is never the primary goal and seldom one can fix own problems by doing something someone has done somewhere before, there is some truth to first learning the rules before starting to bend them. TC1 as all telecommunications companies are (and so are most of the traditional companies) transforming to software companies, willing or not, one needs to learn the how to develop software, but even more so

companies, willing or not, one needs to learn the how to develop software, but even more so digital product development (Humble, Molesky and O'Reilly, 2015; Forsgren, Humble and Kim, 2018; Kersten, 2018).



Figure 19: Known problems and suspected causes of slow development

The team presented the findings from re-organisation study and after that it was universally acknowledged that further work needs to be done to create a solution that tackles the real problem, instead of rushing in into reorganising agile release trains which would only add to the problem instead fixing it. Every member of the TC1 management team then went on to appoint either themselves or members from their own organisation to take part in the "Railyard 2.0" workshop series as described in section 3.2. Author was appointed to take lead on preparing and facilitating the workshops. Management team also decided that the work, the proposal how to proceed, must be completed by 5th of October 2020.

#### 4.2 Workshops in action

In this section we investigate how workshop series described in section 3.2. played out, we look into methods used and outcomes created.

#### 4.2.1 Workshop #1 - Defining the goal statement for organization change

First workshops focused on making sure there is alignment among people fulfilling key roles of the agile release trains and the line organization sponsoring the chosen way of working. The main facilitation tool for these workshops was impact mapping made famous by Gojko Adzic. Taking into account how output driven everything had been before these workshops we followed Gojko's advice of focusing on the goal first instead of thinking about "shopping list" of things to do (Adzic and Poppendieck, 2012).

Impact mapping starts by identifying the goal with qualitative and quantitative qualities. Next steps are to identify all the actors required to create the wanted outcome, Adzic proposes to find actors from both on-stage and off-stage roles plus identifying the parties that can either hinder or prevent fulfilment of the outcome. Third step is to identify the behavioural changes the impacts one needs to create in the actors for the outcome become possible. The last stage is to envision the tangible outputs, things to do to create the impacts, Adzic tells that quite often the order in many enterprises is actually opposite, the order which is efficient in creating solutions to problems that either do not exists or are wrongly understood (Adzic and Poppendieck, 2012.)

First and all following workshops started with thumbs-up voting for simple rules:

- Eliminate distractions, turn off notifications from mail & phones, kids are OK though
- Respectful disagreement is encouraged, it is not acceptable to be disagreeable
- Seek wisdom of ten versus the knowledge of one
- To Assume is to make an ass out of you and me
- Talk and document
- Be bold
- Respect timeboxes

All participants worked from home and in that setting the first rule was very important, it is uncomfortable enough to put yourself out and to be vulnerable if one has to think about the actions of other family members, especially the kids.

We started the workshop with asking expectations from the whole group twenty people, everyone except SVPs participated these workshops.

# **Expectations**

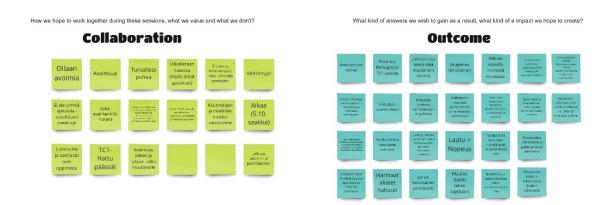


Figure 20: Expectations towards collaboration and workshop outcomes

Figure 21 shows well that even though it was agreed to create all the material in English to prevent from having to translate it when communicating with development teams and other companies in the group, it is not that easy when everyone speaks natively Finnish. While many of the expectations can be grouped into couple of key themes, the results shows that the impact mapping route was at that time good choice.

The workshop group split into two breakouts to start thinking about the overall goal with following guidance:

- Why are we working in lean-agile manner to develop our technology driven business?
- How should the system perform for us to get outstanding customer and business facing outcomes?
- What does excellence look like?
- Keep our strategy in mind
- State in one to three sentences what we (as TC1) aim to achieve by agile TC1
  organization and associated principles and practices

The outcome of the first workshop was commonly agreed goal statement:

"Sustainable and productive way to discover and develop solutions that support our chosen strategy, at the right time.

Throughput / lead time / capability to decide what goes through the system (just enough, right stuff, with right quality).

And to continually eliminate impediments from path to success"

The second workshop then focused on aligning on what the key metrics should be, how do we know we have succeeded with our goal.

#### 4.2.2 Workshop #2 - Defining the KPIs for the organization change

The second workshop concentrated on defining four key performance indicators (KPIs) for the planned organization change, in other words the way how we would measure railyard 2.0 success and communicate our ambitions. Model for the KPI setting was borrowed from Adzic's model introduced in the book Impact Mapping - making a big impact with software products and projects (Adzic and Poppendieck, 2012).

#### Milestone I:

more players, no negative impact on retention, 100% increase in IT costs allowed if necessary

	MORE PLAYERS IN 6 MONTHS	OPERATIONAL COSTS	PLAYER RETENTION
SCALE	#MONTHLY ACTIVE PLAYERS	HOSTING COSTS + OPS SALARIES	% PLAYERS COMING BACK ONE WEEK AFTER SIGN-UP
METER	GAME DATABASE	FINANCIAL ACCOUNTS	GAME DATABASE
BENCHMARK	350 000	\$ 50,000	32,00%
CONSTRAINT	800 000	\$100,000	32,00%
TARGET	1000 000	\$ 50,000	32,00%

Figure 21: Key metric definition example from Impact Mapping book (Adzic and Poppendieck, 2012, p. 43)

KPI table had name of the KPIs on top row for each column, scale row then indicated what we would measure, meter indicated where that number is fetched, benchmark stated the current situation, constraint signalled the minimum acceptable value and target stated where we wanted to go. For this specification we felt we did not want to change anything from the book reference (Adzic and Poppendieck, 2012, p.40).

Workshop team again followed the breakout - sync - breakout - synthesize pattern as illustrated in figure 13. to come up with metrics that would signal that TC1's agile transformation had reached its next goal as defined in Workshop #1.

Workshop leveraged existing material from "better tracks for TC1 trains" -workshop material, existing KPIs and problems statements from earlier work emphasizing the slow time to market.

Workshop methods mainly followed brainstorming, affinity mapping, dot voting (Gray, Brown and Macanufo, 2010) with multiple rounds to come up with the final set of KPIs that would be accepted the group, i.e., reaching consent. Consent means nobody would actively work against reaching the KPIs while individually not necessarily agreeing on every detail.

Most agile frameworks try to make sure that there is stability between product, technology, and process dimensions of the modern product development game. Some frameworks even remember to include the business into the mix. Looking from this perspective it was if not a surprise but pleasant realization that the KPIs decided covered all bases.

	Data informed decision making, we only build validated assumptions	Three-fold increase in end-to-end integrated features built in a PI during 2021.	Validation of outcomes as integral part of the work	Employee and leadership satisfaction index
Scale What is measured?	How many go/no-go decisions we base on qualitative and quantitative research data from the discovery/exploration phase?	Amount of completed B2B/B2C Features in a same PI. (Integrated and same Epic implementing Features)	% of finished Epics & Features containing a link to dashboard or report tracking commercial and non-commercial metrics defined in the Epic Business Outcomes, Feature benefit hypothesis and/or NFR requirements.	Median score from team and leadership health checks per PI
Meter Where we measure from?	JIRA  Portfolio & Program Kanbans	JIRA  Portfolio & Program Kanbans	JIRA	Team & Leadership Health-check survey
Baseline Benchmark	N/A	2020-level	N/A	2020-level
Success criteria Constaint	50%	1,5 X 2020-level	50%	3,5 / 5 median score
Goal <i>Target</i>	100%	3 X 2020 -level	100%	4,5 / 5 median score

Figure 22: TC1 Railyard 2.0 key performance indicators

First KPI: "Data informed decision making, we only build validated assumptions" was chosen because we knew there was too much work in process (WIP) in the system and best way to make sure we'd follow our NorthStar of "Sustainable and productive way to discover and develop solutions that support our chosen strategy, at the right time." is to make sure we test our assumptions and hypotheses early as possible and most of the ideas would need to be rejected even before anyone get developing anything.

Second KPI: "Three-fold increase in end-to-end integrated features built in a PI during 2021." focused on making sure that we change how the system behaves, move in this metric would mean we have been successful in creating better flow and faster time to market or at least faster lead time from idea to production.

Third KPI: "Validation of outcomes as integral part of the work" got chosen because the group felt that there is no responsibility from business side to validate results or communicate if results had been validated. Group felt that making sure we validate the outcomes is a key action to create working learning loop.

Fourth KPI: "Employee and leadership satisfaction index" this metric got chosen as a health metric to measure if any changes to TC1's virtual organization created systemic changes which served the people working in it and the people benefitting from the system. The

reference system for the team and leadership health-check was adopted from work performed by Jimmy Janlén over years at Spotify (Janlén, 2019).

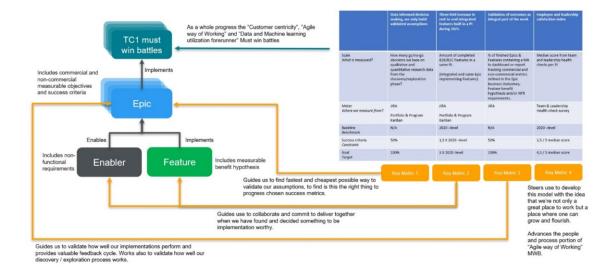


Figure 23: Relation of Railyard 2.0 key metrics and customer / business facing initiatives

De-briefing session after the workshop generated multiple good questions but one of them generated Railyard 2.0 key metrics to business metrics relationship mapping diagram, it was asked "why doesn't railyard 2.0 key metrics contain any business (money/value) metrics?". Reasoning behind this choice was that business or value outcome metrics change with initiatives and system of work is persistent, therefore conscious choice was to put metrics towards ensuring that initiatives have always measurable benefit hypotheses which get validated instead of putting any specific business or value outcome measures as part of the key metrics that measure outcomes of railyard 2.0 work.

# 4.2.3 Workshop #3 - Defining the Actors and Impacts

The focus area for this workshop was to identify all the internal and external on-stage or offstage actors that can help or obstruct TC1 from reaching the goal set in the first workshop.

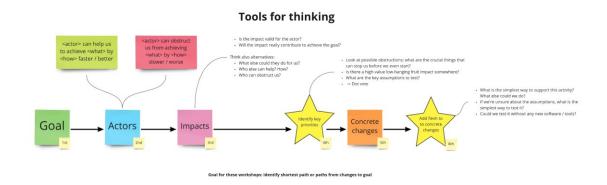


Figure 24: Thinking tools used at third workshop

In this workshop we split the group in to two and tasked first group to think about the onstage actors and second group to think about the off-stage actors needed. On-stage actor was defined as people or roles directly working in TC1's Agile Release Trains (ARTs) and off-stage actors being executive sponsors, line organization supervisors, initiative owners and the like.

After first breakout session groups were asked to review the other group's work and add actors or add comments based on findings.

Second task for the groups was then to identify what type of an impacts for each of the actor for TC1 to get towards the overall railyard 2.0 goal. Impact is defined here as a shift in behaviour written in action-oriented form for example: "Business Owner <Actor> Trust PM/PO to do their job <Impact>". Groups were again asked to review other groups work and again add more impacts or add comments based on what they felt was valuable or otherwise important. In the end we had 35 different actors and over 60 impacts identified, this represented the divergent phase of the workshop. The aim was not to limit possibilities too early and think only the usual suspects and obvious answers.

As last task individuals were asked to think about and markdown which impacts would they "fund". Funding was done by allocating every participant one large and three smaller stack of coins and saying that funding meant that if they would fund something, that meant that they were willing to invest own time and efforts in creating that impact.

As a result, we ended with ten actors and 22 impacts and out of those 22 impacts twelve were identified as primary impacts and rest secondary. Rest of the actors and impacts were recorded but together as a group agreed that no actions would be planned for them.

At the end of this third workshop, we realized that we need to break the original plan and add further workshop to identify the concrete changes and initiatives that would create the planned impacts. This meant that agenda for fourth workshop was changed on the fly and additional workshops were added to the list.

# 4.2.4 Workshop #4 - Defining the concrete changes

This fourth workshop focused on finding the <u>things</u> that would create the wanted behavioural changes or impacts in the chosen actors. Workshop was structured again in the same way than previous workshops, break into smaller groups to work identify the things and share findings before switching the side of actors to be worked on.

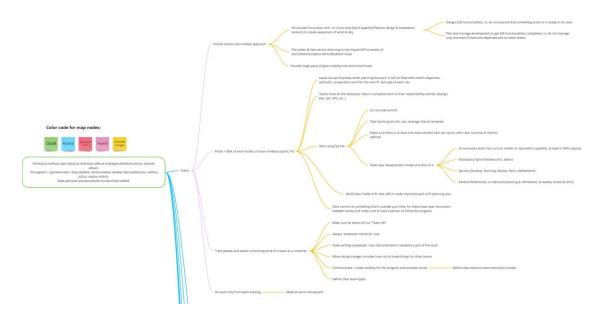


Figure 25: Impact map with development teams in focus

As an outcome there were surprisingly many concrete changes identified on the "teams" actor side and almost all of them were identified without emphasis on the "Online development".



Figure 26: Impact map with business owners in focus

Likewise, there was many fundamental findings on business owner<sup>11</sup> side that would be needed to unlock possibility to do changes with other roles like the product managers and product owners.

<sup>&</sup>lt;sup>11</sup> Business owner term maps into the business line managers or their business development managers. Business owners is here the root level term for the people requesting something to be done or the people having vested interested in the development items.



Figure 27: Impact map with product owners in focus

Key learning for this workshop became the working groups understanding that the problem is something bigger than just re-organizing the singular agile release train. While the realisation created so called "a-ha" moment in many participants, it also created the feeling of anxiety. The task for the TC1 was starting so look a lot bigger than what was upfront envisioned. Two such realisation were gaps in architecture work and role staffing and other one was efficient portfolio management on if not company level but at least business segment level. Especially the architecture which is owned by Tech/IT domain where architects are shared between multiple agile release trains and side projects realized the fact that if TC1 is to be more successful architecture work should use the same scheduling and prioritisation methods as any other development work and greater success would be tied into ability to create a companywide way to limit work in progress. Portfolio management, as story told by company folk lore is a tool that is "not needed" but by the work shopping group identified as one of the key mechanisms for TC1 to create flow instead of resource utilisation maximisation and limit the number of concurrent initiatives for shortening the lead times.



Figure 28: Identified gaps in architecture and portfolio work

The original goal was to formulate all the findings into problem statement, with goal and key metrics defined and coupled with prioritized list of experiments that would be presented to TC1's management team as the way to improve delivery times, quality of outcomes and eventually create system where digital development would also benefit from faster development cycles.

#### 4.2.5 First change of plans

Just as the fourth workshop was finished there was an announcement that one of the TC1's management team members had resigned and as the leaver was the primary sponsor for the workshop work there was going to be some changes. Same time it was also announced that there will be changes in the organisation that is dealing with private customer facing development and that this work would need to be re-structured to support that change.

LACE organisation saw this as an opportunity to influence future organisation design and that we could seed the change with good ideas supporting the overall TC1 agile transformation journey. At this moment LACE organisation and author proposed that group would continue with the workshop series to plan how to create the tangible changes proposed in earlier workshops.

The working group having enjoyed<sup>12</sup> the workshops agreed with the proposal to keep workshopping and going deeper with the subjects. Next, we are going to look what happened between November and 7th of December 2020.

#### 4.2.6 Workshop #5 - Shared principles and practices

Just as Jez Humble et al. believed that every enterprise is different and practices need to be tailored to the context of the business but principles guiding the practices can be universal (Humble, Molesky and O'Reilly, 2015), LACE & author believed that TC1 should not try to dictate universal practices or processes but settle on few ironclad shared principles. The reasoning behind this choice was that there were eight agile release trains operating in very different contexts from purely digital product development to dealing with network interfaces and network building planning. What would work in online development probably would not work with building antennas.

As a result, this workshop yielded selection of new shared principles the group was willing to put into action all over the company. On top of the principles couple of overarching practices

<sup>&</sup>lt;sup>12</sup> Primarily the enjoyment came from the fact that for the first time in many years the people from different functions around the company came together and had a deep and insightful discussions about the shared problem and possible solutions.

were agreed on agile release train (ART) level, practices were limited on improving the transparency and communication between the ARTs.

Workshops again followed the familiar pattern of breaking out, grouping and dot voting the most impactful principles and patterns the group would be willing to put into action.

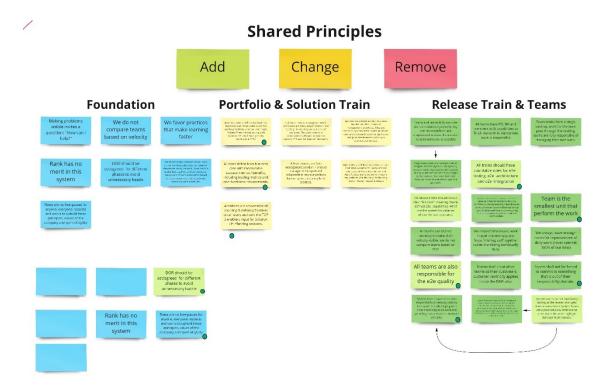


Figure 29: Railyard 2.0 shared principles and dot votes

Following principles made the cut, with area of effect marked first:

- Foundation: DOR should be set/agreed for different phases to avoid unnecessary hassle
- Foundation: There are no free-passes for anyone, everyone respects and works to uphold these principles, values of the company and spirit of agility
- Portfolio: Business owners define clear business objectives with measurable outcomes and help facilitate Solution Train / Agile Release Train mission setting with Solution / Product Management, Architects & RTEs
- Portfolio: All epics define lean business case with measurable success metrics / benefits, including leading metrics and non-functional requirements.
- Portfolio: Architects are responsible of sourcing & defining Enablers on all levels and own the TOP 3 enablers input for Solution / PI Planning sessions.
- Release train: All trains should have countable roles for e2e -testing, e2e architecture and e2e -integration
- Release Train: Team is the smallest unit that perform the work

- Release Train: All teams are also responsible for the e2e quality
- Release Train: Teams shall not be forced to commit to something that is out of their responsibility domain.
- Release Train: System Teams have cross trains responsibility of creating visibility into systems stack, helping with cross train integration work, and providing it as a service to teams in all trains.

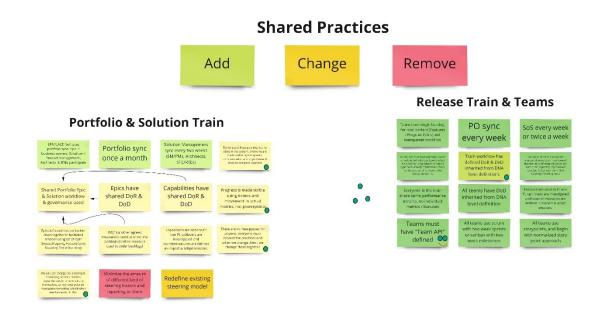


Figure 30: Railyard 2.0 Shared Practices and dot votes

Following practices made the cut, with area of effect marked first:

- Portfolio: Blaming and finger pointing has no place in this system, problems are made visible by transparent communication and improvements shall be designed together
- Portfolio: Progress is made visible using demos and movement in actual metrics, not PowerPoints
- Portfolio: There are no free-passes for anyone, everyone must respect the practices and when we change them, we change them together
- Portfolio: No one can change the ordering & scheduling of work that has
  dependencies to other trains by themselves, on-demand solution management
  meeting is held when need arises to do this.
- Release Train: Train workflow has defined DoR & DoD inherited from DNA level definitions
- Release Train: Teams must have "Team API" defined

While there were many other good choices in both principles and practices area, only the ones where group could reach consent by voting and following discussion were chosen. This is

one of the strong and weak points of decentralized and bottom-up agile transformation or any such type of a change. Only changes that group can currently understand to be beneficial can succeed going forward but, the group might not always know what would be good for it. Though trying to coerce the change one could face even more resistance and get absolutely nothing done or even lesser results as Gibbons said "Resistance is engagement, just not the kind of engagement we wanted" (Gibbons, 2019, p. 299-316). Trust and inclusion seemed to work wonders here whereas previously the people forming the workshop group had not exactly seen the things in similar light at all. Here it was able to reach consent somewhat easily when previously such topics had been escalated to shifting blame and finger pointing.

Second half of the workshop focused on touching team interaction models and team types as singled out by the impact mapping exercise from previous workshops. Team Topologies (Skelton and Pais, 2019) as made popular by the book from Matthew Skelton and Manuel Pais try to create vocabulary for making distinction between team types and their interaction models. Team Topologies identify four different team types: stream aligned, enabling, complicated subsystems and platform team while describing three different interaction modes: Collaboration, X as a Service and facilitating (Skelton and Pais, 2019.)

While this section of the workshop did not yield any concrete outputs the discussion of teams and their interaction models was much valued by the group. Specially the difference between good and bad collaboration was insightful for the group.

Restrict any ongoing collaboration between teams to explicit valuable activity.

Collaboration is expensive. Unnecessary collaboration is particularly expensive, especially as it can mask or hide deficiencies in underlying platforms or capabilities. Any ongoing collaboration activity must, therefore, be justified as valuable discovery, valuable capability building, or valuable deficiency-filling activity.

Figure 31: Team Topologies tip about valuable collaboration (Skelton and Pais, 2019)

The group found it insightful that need to have a lot of meetings which are often costly and slow might stem from the fact that teams at TC1 might have a lot of deficiencies in the technology stack, in its usage and lot more.

### 4.2.7 Workshop #6 - Roles and responsibilities

In this workshop our intention was to clarify and make sure there exists a shared understanding of roles and their responsibilities in TC1's virtual organisation. This workshop was designed due to the findings from the impact mapping workshops. Many of the findings on how different roles should act and what type of a responsibilities they should own mapped

closely to what TC1's chosen agile scaling framework SAFe describes about the roles. We wanted to avoid reinventing the wheel and use as much as possible the material the already exists and focus (role) customization efforts to part where there was actual value to be gained, i.e., adjusting to local context and connection points of the hierarchical organisation.

Following key roles were chosen for topic of this workshop:

- Business Owners
- Product Management
- Solution and System architect
- Development team
- System Team
- Release Train engineer

Roles were chosen as any shift or change in responsibilities and the authority that must follow with responsibility meant that something is delegated away from the line organisation. These changes would create shifts in the established power structures and therefore present challenges if everyone is not onboard with the changes. If workshop group representing the decision-making authority in the line organisation is onboard with the changes, there is better chance for the changes to materialise in action later.

Business owner role, the group agreed that we should follow the SAFe recommendations at least for now and while one key question was asked: "Can BO actually have technical competence of the solution?".

SAFe article on business owner roles says: "Business Owners are a small group of stakeholders who have the primary business and technical responsibility for governance, compliance, and return on investment (ROI) for a Solution developed by an Agile Release Train (ART). They are key stakeholders on the ART who must evaluate fitness for use and actively participate in certain ART events." (Scaled Agile Inc., 2021c.)

Agile release trains at TC1 mainly had one business owner, the business director of the unit the ART was responsible for creating solutions for. This follows the behaviour of IT & Technology units serving the business, which meant the relationship is not equal and that showed here too. The answer to the original question given by the group was "yes, if we staff the role differently". Though no consensus was reached on this subject.

For the product management role, it was recognised that such role and function does not exist at TC1 currently and not in the spirit of what SAFe recommend. The group agreed that such role should exist, and the outcome of the workshops is to recommend putting more

effort into it and creating a playbook what the role should do instead of what it was doing currently.

Architecture role, major agreements were found in areas that every agile release train must have dedicated architect and architect must take more active role in shaping up the enablers<sup>13</sup>. Consensus existed around the fact that there are not enough architects at TC1 to sufficiently support the development flow, instead architects currently form a type of a roadblock slowing down the flow. As TC1 is heavily leveraging partners and vendors in development duties and key decisions and new initiatives must be OK'ed by TC1 architects, there is always more than the current staffing levels can handle.

Product owner, Scrum Masters and Team members also known as the development team had most entries record in the workshop, but amount of action points recorded was the lowest. Everybody agreed that things they are doing are mostly the right ones.

System team<sup>14</sup>, the group agreed that system teams should be responsible of spreading the knowledge and capability to define, design and implement interfaces between systems that eased the system integration work instead of making it harder. System teams should also carry the responsibility of making sure environments get created in a harmonious way and the end-to-end information flows from system to system are documented. System teams should be responsible for a) designing sensible defaults and b) providing them as a service to other teams in the ART. Lastly the need to establish more system teams was acknowledged, the one serving the Online Development ART just was not enough.

<sup>&</sup>lt;sup>13</sup> Enabler term at TC1 follows closely the SAFe terminology: "An Enabler supports the activities needed to extend the Architectural Runway to provide future business functionality" (Scaled Agile Inc., 2021f). Enablers represent system architecture, new components, system non-functional requirement needs, like scalability and performance.

<sup>&</sup>lt;sup>14</sup> System team is a SAFe construct responsible of: "assists in building and supporting the Agile development environment, typically including development and maintenance of the toolchain that supports the Continuous Delivery Pipeline" (Scaled Agile Inc., 2021p). System team equivalents are sometimes called DevOps teams or Developer Experience teams.

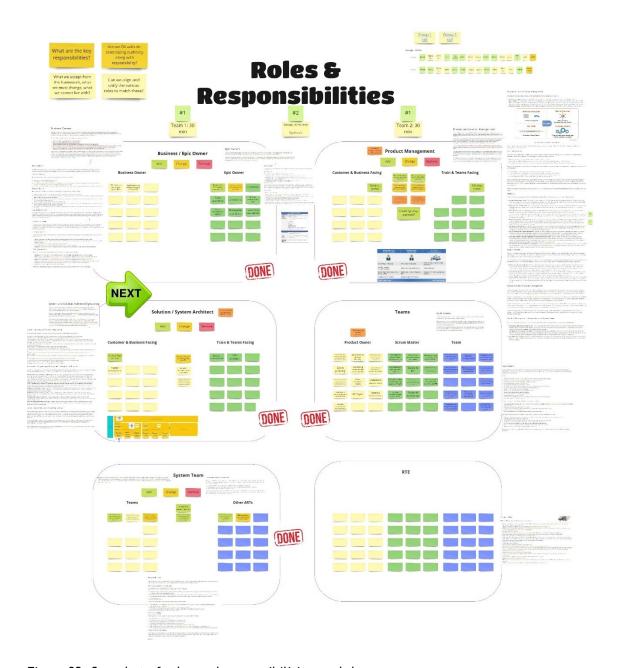


Figure 32: Snapshot of roles and responsibilities workshop

# 4.2.8 Workshop #7 - The Organising for success

We (LACE team & author) acknowledged the fact that omitting re-organisation contemplation was not possible, but all previous workshops signalled that the way how virtual organisation was organised was least of the problems.

To help the group envision the new box, the one where shared principles and practices were acted out, where all the roles did what we had planned and where there were structures in place (system teams) to make sure interfaces supported the development instead of causing problems to it existed. We started the workshop with that exact story, to think the solutions not from today's problems but from the angle of what has already been found and agreed.

We started with revisiting the previous reference re-organisation scenarios developed by the author and LACE -team members based on feedback from the management team. The group worked through the pros & cons of both scenarios and good discussions ensued from the models.

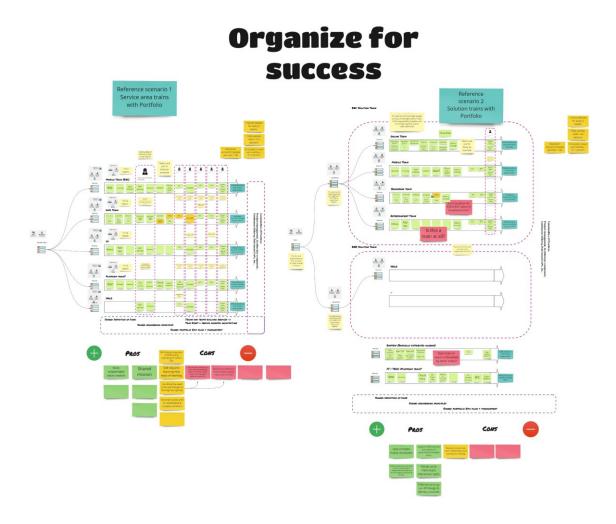


Figure 33: Evaluating different release train organization scenarios

Workshop group was then split into two to think about where the major pain points between teams, roles, system, or system components were. This exercise was designed to find if major similarities existed, signalling that there might be a real problem. First group identified 6 problem areas and the other identified 26 problem areas. Problems were found from three categories, between teams on the trains and teams not on the trains, between system x and system y and between different backlogs (portfolio to program). While the findings were great, it did not help the group to create any better organisation models. Every way we investigated organising the ARTs there were always too many complicated areas left out and pooling everyone 600+ people into one agile release train just was not feasible. There was no value to be gained in having that many people together due to a) group being so large that the shared base would be too small, too much information and talk about topics that are not

even remotely interesting creating lack of focus and demotivation. b) only certain teams interacted together (interface builders) but that group was not static but followed the team compositions. Author suggested that next workshop should be about the ways how we can reduce the pain in interfaces and how it could be possible to improve team to team game. Group agreed with the suggestion and the next workshop was agreed.

## 4.2.9 Workshop #8 - Supporting structures

This workshop hovered around existing findings from the "Better tracks to our trains" - workshops, what feature teams are and are not, and team types and interaction models from team topologies book. This workshop also had pre-reading material from the book Team Topologies - organizing business and technology teams for fast flow by Matthew Skelton and Manuel Pais.

One of the validated hypotheses because development is problematic and times very slow at TC1 is due unclarities in building interfaces or application programming interfaces (APIs), the workshop group went through the core concepts of team topologies and mapped current systems into the model.

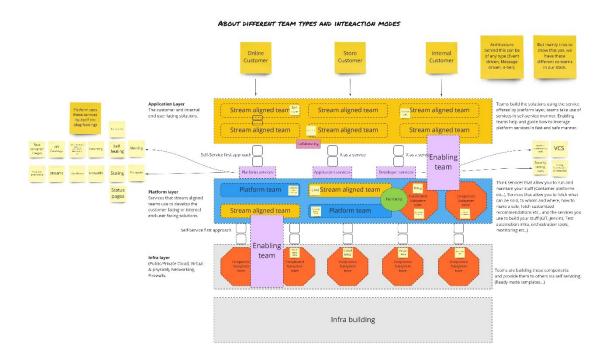


Figure 34: TC1 system mapped into team topologies informed model

One of the most informing statements from this session was "Actually, we would not need trains at all if we could learn how to do this". This statement was made in context of leveraging the team of teams structure (ART) as communication and collaboration tool for places where integrations exist between systems. It was also universally acknowledged that

this type of a setup would take months if not years to build and would not carry TC1 over the next "hump" by itself, there would need to be some transitional practices to go along.

Next session focused on going through feature teams, the long-talked concept how teams should be formed spanning multiple architecture layers at TC1. The group was introduced the feature team as mentioned in multiple discussions from group itself to senior technology management.

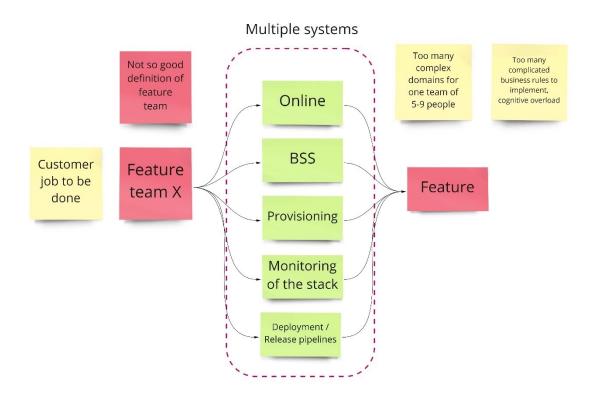


Figure 35: TC1 original feature team idea

After initial discussion it was quite fast deemed not to be feasible idea from multiple perspectives a) current agreements don't support such responsibility sharing, b) too many complex domains inside one team creating cognitive overload in the team, c) could turn just into many teams of one or two persons as no real possibility to collaborate due to different programming languages, repositories, people from different partners etc. While it was acknowledged that this direction is something that we should have as an ideal, it is not feasible right now.

Next the group went through the feature team composition if principles, practices, roles would be as previously agreed.

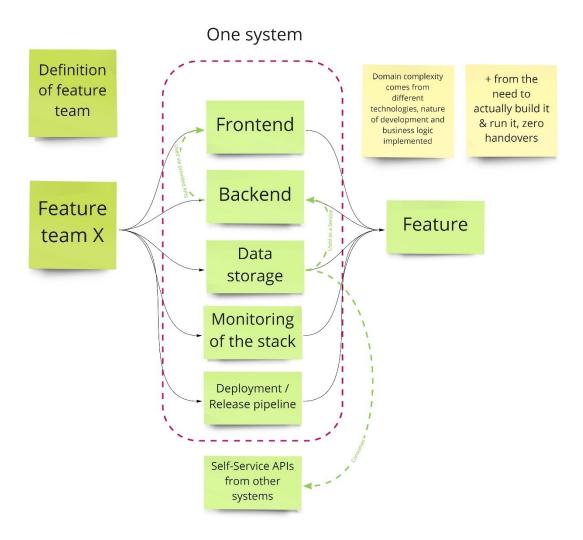


Figure 36: TC1 railyard 2.0 feature team idea

Here the team would fully own their own domain and leverage well defined APIs from other teams to leverage what others have built. Next question was how to come up with well defined APIs if currently none exist.

The group collaborated to visualize how TC1 would want the teams to collaborate when building something spanning multiple architecture layers. It was found that instead of teaming together it would be possible to start from shared API contract<sup>15</sup> and agree regular cadence for integrating the work from both teams.

<sup>&</sup>lt;sup>15</sup> This is more well known as Contract Driven Development or API-first development where purpose is the first design the API specification and derive tests from it before starting any implementation work. API-first specifications expect the specification to be understandable for both machines and humans where the former allow easy mocking of the API to test throughout the design phase improving the specification quality before implementation (Dudjak and Martinović, 2020.)

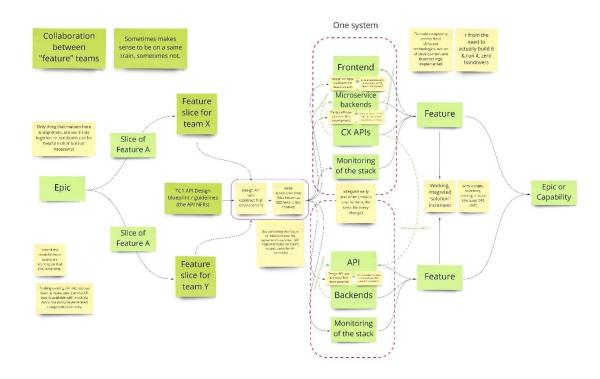


Figure 37: Feature team collaboration when building APIs

This session created further validation in workshop groups technologically savvy persons that "System Team" direction must be explored further.

In closing the group agreed that there is not that much sense to propose changing the virtual organisation, except to pool people from separate tech teams into one train on IT platform level and stop running one administrative train. Reasoning for the former was that there are some 5-6 teams that serve all the teams at TC1, and it is difficult to truly organize backlogs in the form that support value creation in priority order, and it would be difficult to start building self-service APIs as independent teams as there would not be anyone to guard the backlog. Latter for reason that the one team that made the Broadband ART was already actually in another train and the Broadband train existed only to provide reporting structure for one of the business lines. It was deemed to be as easy to invite couple of business owners to different meeting and stop running the train as it only created additional overhead and confusion.

On the group was introduced the concept of shared definition of done, definition of done is used to make explicit what needs to be done to call something done. Based on workshop discussions it had become evident that in some teams testing and documenting is not required, or something can be called done without testing it with the parties that use the API or interface. Group agreed that shared definition of done must be one of the initiatives for TC1.

Lastly author suggested that there should be "best before date" on all the changes, best before date meant that if the workshop group had not reconvened earlier to make changes to whatever is the outcome of the workshops the next iteration of workshops would arrange by that date. This mechanism would be used to keep the TC1's virtual organisation model and its ways of working fit for the current problems of the company.

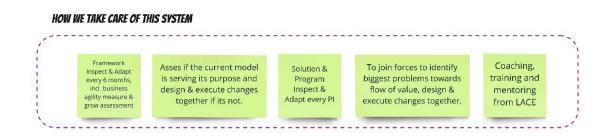


Figure 38: Agreed mechanism to keep the virtual organisation fit for purpose

At the end of the workshop group had gone through the whole TC1's virtual organisation and discussed all the different elements of it from organisation to portfolio and down to how interfaces are built. All the resulting discussions, comments and decisions were recorded into one virtual whiteboard accessible for everyone at the company.

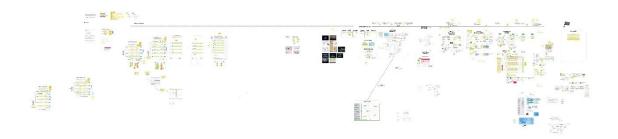


Figure 39: Power of virtual whiteboards

After this workshop the group decided that author and LACE team should distil all the discussions into presentable format, create first version of the action plan with intention that next workshop would be about going through the presentation material before presenting in to TC1 management team. The group felt that TC1 enterprise agile coaches should be responsible for doing this work.

# 4.3 Defining the next steps of Railyard 2.0

Author took main responsibility for refining the workshop outcomes into presentable concepts and models. This work took form of a MIRO board (virtual whiteboard) where more extensive

diagrams were created for whole company to access and a PowerPoint presentation for the management team.

The next steps included finalising:

- Refined Problem statement
- Motivation for changes and expected outcomes
- Overall goal for Railyard 2.0 & key metrics
- Backlog content management model
- Organisation of TC1 Railyard 2.0 Agile release trains and reasoning behind the organisation
- Initiatives to improve the collaboration between product management and agile release trains
- Community of Practice -model
- Head of Engineering -role
- Task Force -teams
- Summary of planned changes
- Frequently asked questions

Next sections focus on detailing the changes we distilled from the workshops and previous materials.

#### 4.3.1 Problem statement

As problem statement LACE team and author crafted following three statement to explain the root cause behind the need for change, taken from the presentation by author and LACE (Jaakkonen Antti, 2020):

"Shared concerns are worked on silos for too long. We collaborate with all the required parties too late in the process, leading to hasty point-solutions that eventually work but are brittle. Every such change increases technical debt. Much of our slowness is designed and architected by this flaw in our practices."

"There are significant differences between roles in our agile release trains, the responsibilities and how the roles are fulfilled. Leading to a state where we systematically forgot to do certain things like early validation of our assumptions, architecture, the building of shared understanding, cost/benefit analysis. Certain people have too many things on the table, or it is unclear what the role's area of responsibility is. Is it about what and why, how, or how fast? Trying to carry too many duties single-handedly leads to mediocre solutions and bad decision making as one person can only have expertise in so many domains."

"The principles and practices used to implement technical agility are unstructured, unevenly distributed, and service provider/consumer responsibilities concerning interfaces, environments, and visibility are unclear. Thus, integrations, end-to-end work is cumbersome, and a disproportionate amount of time is spent solving problems during development."

#### 4.3.2 Motivation for changes and hypothesis of outcomes

We wanted to crystallise our motivation for proposing the planned changes into couple of paraphrases targeting the topics either raised when Railyard 2.0 initiative was launched, or items represent major change from what previously existed. We came up with following phrases(Jaakkonen Antti, 2020):

Our number one objective for proposed changes is to help teams work better together on areas where collaboration creates value or where significant new functionality is created for the first time.

For Digital Solutions ART, the objective is to enhance horizontal integration, bring people together closely related to the same products, and remove communication and collaboration barriers or siloing of information and knowledge. Aim is to make sure we know as a whole what are "the problems to be solved" and can employ the right approach for any given situation, not to treat them only, for example, from a development perspective or only from a producer/analytics perspective.

Thinking behind Platform ARTs is to productize the shared services and clarify the interaction models between teams building the services and teams consuming the services. We believe scalability and the ability to succeed with today's and tomorrow's problems require a shift from the need to collaborate in-person to access services to provide templated self-services powered by automation, RPA, and APIs. Platform ARTs also act as containers for new teams to learn agile ways of working.

We wanted also to provide answer how the problem we defined would cease to exist by implementing the changes envisioned, our hypothesis became (Jaakkonen Antti, 2020):

We believe that by implementing these changes, we can attain more substantial commitment towards initiatives we need to build together (Portfolio creates shared mission, shared ways of working build trust, and help build a shared understanding of what is required).

Clarifying the key roles and making sure there is a balance between responsibility and authority, we can reduce the decision-making lag, make work more fulfilling and allow people on all levels to spend more time on relevant matters.

Vertical structures like Technical Agility CoP enable teams to acquire capabilities we require our teams to possess and to implement practices we need them to implement in order to guarantee frictionless integration and end-to-end quality. We believe we can drastically reduce the amount of time we spend fighting against environment and quality issues during development time and increasing the capacity for value-creating effort.

#### 4.3.3 Goals and key metrics

Overall goal and key metrics did not change from what was agreed in the workshop and shown in figure 21. but were amended with proposal to take the as part of TC1 official company metrics for year 2021/2022.

# 4.3.4 Backlog content management model

For the content management problem and creating countermeasures for having too much work in process (WIP), it was proposed to implement lightest possible lean portfolio approach where initiatives requiring work from multiple agile release trains would be holistically analysed before any work is released into the system. Limiting WIP is currently best and cheapest known way to improve lead times (Martin, 2012; Humble, Molesky and O'Reilly, 2015; Modig and Åhlström, 2016; Forsgren, Humble and Kim, 2018).

Previously the collaboration between agile release trains (ART) stood heavily on individual people having good relations with people in other trains backed with ad-hoc processes. Based on workshop learnings this created a system where multi-ART initiatives were split into independent sub-initiatives without good practices to connect those sub-initiatives. This led to situation where both ARTs could signal success, but end-to-end functionality was either untested or worst, broken. TC1 did not either have a good system to prioritise key initiatives in a way that would guarantee end-to-end success.

LACE team then designed a countermeasure to create a connecting layer between different agile release trains for making sure shared initiatives are done collaboratively and initiative workflow would be visualised to make sure shared initiatives won't get cancelled by local initiatives. This countermeasure was directly aimed at the most pressing overall feedback that "development is slow".

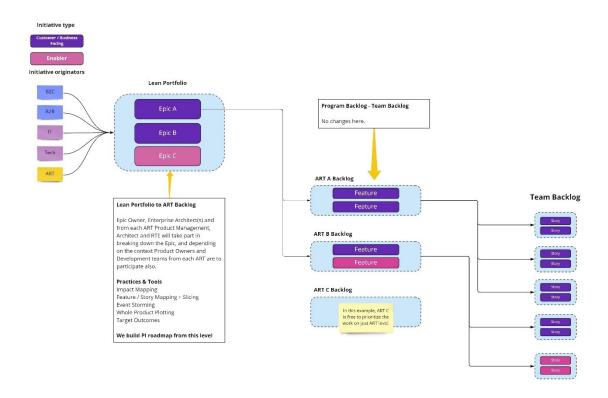


Figure 40: Proposed Simplified Lean Portfolio Model for TC1 Agile Release Trains

General guidance for when something should be considered on portfolio level was also designed and here the guidance follows closely SAFe's reference (Scaled Agile Inc., 2021k).

Which backlog	Who owns the business content?	Who owns enablers?	Duration	Scope
Epic	Epic Owner	Enteprise Architect	Longer than one PI	More than one Agile Release train needed fo discovery & delivery
→ Feature	Product Management	Train-Architect	Shorter than or equal to one PI	More than one team inside one Agile Release Train is required for discovery & delivery
Story	Product Owner	Development team	Shorter than or equal to one sprint	One Team is required for discovery & delivery

Figure 41: Guidance on backlog levels

Additionally short statement to support establishing portfolio level was proposed (Jaakkonen Antti, 2020):

"In content management, we follow the principle of decentralized decision-making (Framework Principle # 9), we strive to always make decisions as close as possible to implementation but consider decisions that have infrequent strategic or long-term

implications (e.g., product strategy direction or when significant economies of scale are present) where it is worthwhile to decide in a binding way and spend the time they need to prepare and plan. These decisions belong to Lean Portfolio Level; all other decisions belong to the Agile Release Train or Team levels.

As mentioned, it is important to note that each level is empowered to decide on initiatives affecting the local context and propose topics for portfolio level. All agile release trains must have (if one does not already exist) a clear mission, objectives, and key indicators.

The role of portfolio and Agile Release Trains levels is to provide direction, and guidance, create alignment and make the strategic or tactical choices clear for everyone. Doing so enables everyone to go in the direction we intend with our strategy and product vision(s)."

## 4.3.5 Organising TC1's agile release trains

Based on workshop materials and earlier findings the workshop group and LACE team along with the author chose not to propose major changes to existing Agile Release Train (ART) organisation. Following guidance was applied when designing the organisation proposal:

- One person should belong to only one team if possible
- One team should belong only to one agile release train
- Teams should expose their data & capabilities as well designed & defined APIs if providing services to many teams
- Organisation should minimize the number of competing number one initiatives for all levels
- Proposed model should make capacity decision explicit

Following these principles one ART was to be disbanded and two new to be created. The disbanded ART did not contain any development teams but existed mainly as communication method towards one B2C organisation. For this ART it was proposed to just move the work from to be disbanded ART to "Core Broadband Services" ART as the team responsible for doing the work was already there. Other change proposed change was to bundle various independent shared services teams providing services to many other teams into two different ARTs. Motivation behind this change was to create one backlog for all the different initiatives to provide more meaningful way to prioritise the needs to better support overall strategic goals.

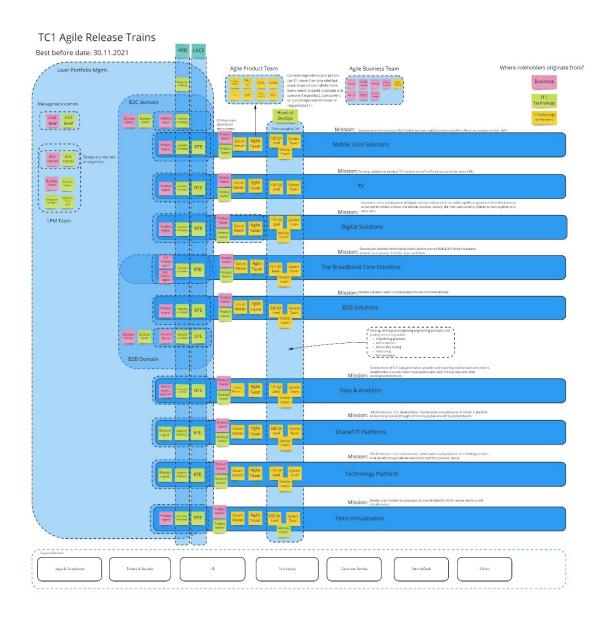


Figure 42: Proposed New Agile Release Train organisation for TC1

### In the picture:

- Principles of De-centralized decision making (Scaled Agile Inc., 2021l) is used to push content and go/no-go decisions as close to team level as possible.
- Pink, Green and amber boxed visualise required roles on each train, one person can fulfil one or more roles, depending on the context and capabilities
- Vertical structures are communities or practice / virtual team type constructs whose responsibility is to keep practices and ways of working unified in the area
- Technical Agility CoP would start with fewer team members than drawn, but the responsibility area is all trains.
- Every level contains guiding agile team responsible of Product, Technology and Way of Working for given area

Larger changes were deferred to next proposed evaluation point marked with Best Before date as the workshop results signalled that many ARTs and Teams had to improve technical practices before further de-centralization would have made sense. This approach would provide much needed stability for teams to be able to focus on their capabilities, practices, and upskilling.

#### 4.3.6 Initiatives to improve collaboration

Many agile transformations and change initiatives fail because it is convenient to leave out parts that feel hard or if there is a perceived "time is not right" -feeling. Together with LACE team we analysed the workshop findings and what was happening at TC1 and quite quickly came to conclusion that collaboration problems stem from the fact that many basic actions of SAFe framework were left out. Unfortunately for TC1, the ones that were left our were mostly about the cross-organisational collaboration and systems & nudges that encourage splitting large initiatives to smaller ones and then focusing on getting those done with production quality.

Our proposed changes to improve collaboration then focused "back-to-basics" items such as: systematise and harmonise core-forums, how they act and make sure they focus on:

- Content management, initiative selection and prioritisation encouraging finishing instead of starting
- Development time collaboration and ensuring there is an intent to create working software, through end-to-end demos and using shared cadence as integration points between agile release trains and teams
- Establishing regular shared forums for solving problems (Portfolio Sync, ART sync, Architecture Sync)
- Regular retrospective on all level, from portfolio to teams, to ensure that whole system develops instead of just local optimization on team level

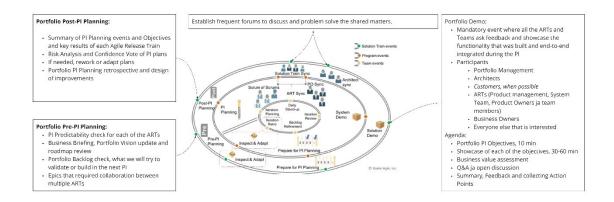


Figure 43: Using the forums that already exist by the framework

#### 4.3.7 Communities of Practice model to share learnings and good practices

Communities of Practice model was chosen as one of the key countermeasures to make sure the core problem of not having the good engineering practices evenly distributed inside TC1 would disappear. We knew that TC1 had a many good teams and capable individual but currently there was no way to connect the people. Although nothing prevented from people discussing with each other or even spend time to help other to solve their problems. Even though it was never said aloud, we assessed that the problem was combination of agreements with vendors, having too much work in process and the fact that TC1 was large enough company to create either physical or social distance between people that would hinder spontaneous collaboration. To support Communities of Practice at TC1 we proposed following leveraging SAFe's guidance (Scaled Agile Inc., 2021d) coupled with TC1 context specific artifacts (Jaakkonen Antti, 2020):

Vertical Community of Practice type structures can look like matrix-structure at glance but that is not the case. Belonging to a CoP does not mean having multiple managers and backlogs, work still flows from single backlog and the community is about taking care of shared concern(s) together.

#### Definition of Community of Practice

"Communities of Practice (CoPs) are organized groups of people who have a common interest in a specific technical or business domain. They collaborate regularly to share information, improve their skills, and actively work on advancing the general knowledge of the domain."

In our model CoPs can be formed around for instance Technical Agility, Microservice Architecture and technology or Test Automation. CoPs do not directly decide or take responsibility what is done but how things from their responsibility area are done. This is used to align for instance how and when microservices need turned to generic ones to serve multiple teams from different contexts.

### Types of Communities

We acknowledge there are going to be two types of CoPs to be formed, ones based on voluntarism and ones that require mandatory participation based on role person has. Instances of the voluntary type are LACE, LAL, DevOps Lean Coffee, while instances of the mandatory type are Technical Agility CoP, Microservices CoP, ARB, RTE CoP. CoPs requiring mandatory participation must be led by TC1 personnel.

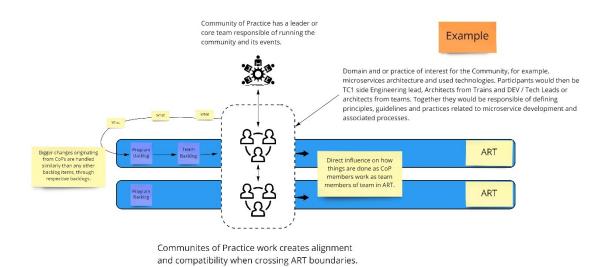


Figure 44: Proposed TC1 Communities of Practice Model

#### 4.3.8 New Roles: Head of Engineering

One of the realisations from the workshops were that no one at TC1 owns the Software Engineering practice, at least no one with capacity to align principles, some key practices and to create vision and standard of what good looks like. This meant that there were as many approaches to software development as there were agile release trains. While many principles and practices that ARTs employed were good for their own context, the resulting combination did not really guarantee good systems outcomes.

For this reason, we proposed that TC1 should take drivers wheel on leading software engineering capability, the way should start with creating a new role or position for CTO type of person that would take the responsibility to form the missing artifacts. As TC1 is striving towards empowerment and agile ways of working, this should not be done in an "ivory tower" but crowdsourced. This crowdsourcing would need some in a role that can provide facilitation for such process and ask critical questions where needed.

For the model we chose to propose something that followed the Communities of Practice - style but relied less on voluntary participation.

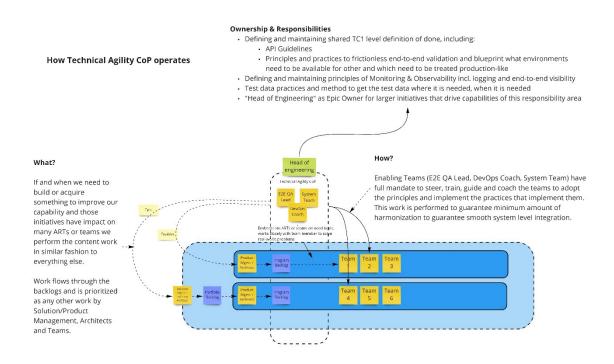


Figure 45: Head of Engineering role proposal with operating model

#### 4.3.9 Task Force Teams

As the workshops highlighted that many areas where teams from two or more different Agile Release Trains (ART) were to collaborate, and sometimes even inside an ART, problematic friction existed. The friction is created by trying to resolve need of face-to-face communication with issue handling system tickets, chat messages and un-prepared ad-hoc meetings, not to mention different priorities and high workloads.

We also found out that most often the problems deal with system interfaces or APIs of the systems and the problem of defining one or understanding how defined interface works. To eliminate this problem, we designed an approach that combines outcomes of Head of Engineer work and existing definitions like: API Guidelines. Our proposal is to assign people from both-sides of an API to one temporary team with 100% focus on resolving the issue and making sure that the outcome (API) is understandable by both parties, its documented and it functions as designed. We called this approach Task-Force Teams, which would be the transitional construct until practices at TC1 would evolve into state where such teams would no longer needed.

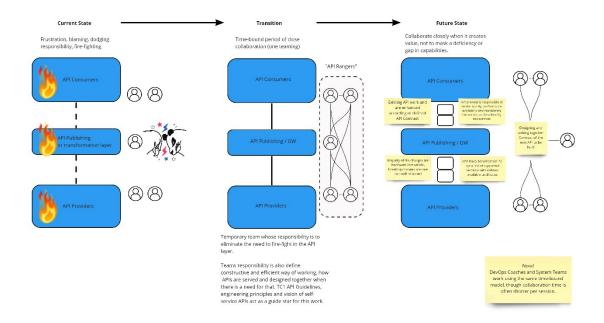


Figure 46: Proposed Task Force team -model

### 4.3.10 Frequently Asked Questions

As the proposed changes encompassed much larger area than just re-organising one agile release train such proposal would be a surprise for the people not having taken part in the workshops. As most of the senior leaders chose not to take part in the workshop, we played a game of Devil's Advocate and tried to anticipate the questions this approach might create. The change proposal consisted of "Frequently Asked Questions" segment with twenty most probable questions with in-depth answers and links to workshop material and the accompanying 67. page presentation deck. Questions and answers ranged from "Why we're not proposing to reorganise the online development / digital solutions release train" to "Is this end of it all model" and covering questions targeting at planned changes for roles.

## 4.3.11 Proposed next steps

As the virtual organisation at TC1 lives in shared ten-week cadence we proposed to align implementing that proposed plan with the same cadence. With such vast set of changes the plan was proposed to be split it into four PIs during year 2021. Each of the PIs would contain roughly equal amount of planning, initiating, and sustaining activities tied to different areas of proposed changes. The last PI of 2021 was planned to contain another iteration of Railyard workshops to create a process to frequently inspect and adapt TC1s agile organisation, its structure and way of working.

PI 2021.1	PI 2021.2	PI 2021.3	PI 2021.4
<ul> <li>Launch         Technical         Agility CoP</li> <li>Release v1 of         shared DoD &amp;         API Guidelines</li> <li>Design B2C         Portfolio flow,         roles, and         responsibilities</li> <li>Design and         plan Platform         train launches</li> <li>Train Product         Management         role holders</li> <li>Agree B2C         Digital         Solutions train         changes</li> <li>Plan how to         implement         shared         principles</li> <li>Inspect &amp;         Adapt</li> </ul>	<ul> <li>Launch B2C         Portfolio         flow and         shared         events</li> <li>Launch at         least one         platform         train and         define its         interfaces to         both         portfolios</li> <li>Launch         additional         CoPs like         Microservices         and APIs</li> <li>Put team         and ART         level shared         practices to         work</li> <li>Inspect &amp;         Adapt</li> </ul>	<ul> <li>Train LPM role holders</li> <li>Design LPM function</li> <li>Launch second Platform ART</li> <li>Enact remaining ART changes</li> <li>Inspect &amp; Adapt</li> </ul>	Arrange     workshop     to asses     situation     and to     create next     changes,     set new     best before     date

Table 2: Planned schedule for proposed changes

#### 4.3.12 Reviewing the change proposal with the workshop group

Before presenting the work to TC1's management team, the final step was to go through the change proposal together with the whole 20+ strong workshop group. This session yielded some small improvement ideas to the presentation material. Still, the consensus among the group was that the material represented the discussions and managed to capture the key pain points while proposing solutions that the group could stand behind.

#### 5 Results

The original aim of this project was to investigate what could be done to TC1's Digital Services development agile release train to speed up the development. Project goals were later altered to investigate the systemic causes for slow end to end development and to generate a list of required changes to change the situation.

Before presenting the change proposal to TC1's management team release train engineers (RTEs) running trains developing solutions for B2C organisations went through the material with respective business leaders and product development responsible. The aim was to explain the changes and capture initial feedback & questions that would be used to fine-tune the message and material used for TC1's management team presentation.

When the change proposal was presented to TC1's management team, one of the senior leaders raised an issue about disbanding the administrative train that provided solutions solely for his organisation. At the same time, the change was deemed minor and non-issue by the members of the same organisation who got delegated the workshop participation responsibility and despite the pre-meeting briefing sessions. The management team decided that the change proposals would need to be presented to every relevant steering board at the company before any actions could be taken at scale.

After the meeting, it became clear that the tour of steering boards would slip into the beginning of the year 2021, and as a result, the momentum for change diminished. The cooperation process had ran its course and the outcomes were communicated. A new department was formed, pulling all product management personnel from different B2C organisations into one organisation. As the change proposals relied heavily on the participation of product management personnel, this change meant that their focus was now on establishing the new department and its ways of working, plus recovering from the quite painful process.

As it became evident that TC1 level actions would not happen any time soon, the LACE lead decided the work would not be brought into any of the steering groups, and the LACE team would move forward with the changes that were inside LACE team's mandate.

Next, we will look at what type of changes were possible and how they affected the overall problem statement.

## 5.1.1 Results on the single agile release train level

As the author worked for TC1's Online Development Agile Release Train (ART), this section covers changes done over that ART. As ARTs at TC1 have much freedom in dictating their own processes and practices, many changes were possible if they followed the SAFe and TC1's agile principles.

The first thing to change was to change the name of the ART from Online Development to Digital Solutions, and this was done to clear the table and stop using the name that had a reputation of being slow to deliver.

The second thing was to build a product management function. While the ART had five product owners, none of them had enough time to collaborate with business stakeholders often enough or to have a proactive discovery process. In other words, we picked the goals and KPIs from the railyard 2.0 workshops and the actions targeting product management and applied them on one ART level.

Third change was to put more focus on the technical practices with the driving idea of making it visible what could be possible in a wider TC1 context too.

The following results were observed in December 2021:

- None of the business stakeholders complained that development was slow, although
  the change came from a raised awareness of what it takes to build quality digital
  products through communication and actions performed by two dedicated product
  managers, one for each B2C business line
- The digital solutions domain rose to the top 2 on deployment frequency metric, from 70 production deployments in 2020 to 360+ production deployments in 2021
- Change lead time observed moderate cuts ranging from -30% to plus 10%, though this
  metric did not have a good baseline at Digital Solutions ART due to not paying
  attention to JIRA that much during 2020
- Subjective assessment based on feedback from other RTEs that communication with neighbouring agile release train had improved, similar feedback was collected from

Digital Solutions ART team members. The most impacting change here was the creation of an initiative based shared messaging channel.

## 5.1.2 Results on the organisational level

While no large-scale collaboration efforts were done based on the work, it acted as a catalyst to kickstart change throughout the TC1's organisation. One of the RTEs with LACE team support managed to form one new agile release train around some of the shared services teams. Architects started to map their people and events on top of the release trains and concluded that they needed to recruit more architects. Technical practices gained more traction, and IT started to pursue the idea of establishing System Teams in other trains following the success of the Digital Solutions train's System Team. An initiative to investigate recruiting more developers in-house was also kicked off.

As the most notable change the work managed to re-kindle, if not spark, the interest of the top management in participating more in various events of the virtual organisation. During year 2021 one or more management team members gave a briefing at the start of the PI planning sessions for all the virtual organisation members.

Railyard 2.0 also became a systematic process, and towards the end of 2021, more than six workshops were held targeting changes with a much tighter scope than the railyard 2.0 sessions. These sessions attracted even more people than railyard 2.0, something which, based on the feedback, would never happen unless railyard 2.0 would not have paved the way.

In the end, the best result was getting people across the organisation together and talking about the shared concerns, which previously had only happened in organisational silos.

#### 6 Conclusion

This work highlights the issues of bottom-up change management approaches and the effects of departmental specialisation in creating communication barriers. It also highlights the need for a compelling vision for the change but even more so for case of going agile. If there is no shared understanding, in this case, of what agile and the chosen scaling framework mean for everyone, many processes become exceptionally brittle. Change in the management team and senior leaders delegating the responsibility down led to a situation where planned changes were not accepted, which could have jeopardised the outcome. TC1's strength and weakness are the almost unlimited freedom to do what one believes to be correct. It worked, but very different results could be observed in other companies repeating what was done at TC1.

It needs to be noted that it is hard to attribute any of the changes to the movement<sup>16</sup> of the main metrics introduced in section 1.4.6. The subjective evaluation says that some positive change could be seen in the metrics, but the results are inaccurate since more systems were connected to the metrics tracking system during 2021.

Having visibility into the whole also seemed to be the most important matter in reducing the finger-pointing and complaints of slowness. As the new B2C organisation is responsible for product development decisions, they must now look at the whole together and make trade-off decisions where previously it was possible to point a finger from business-line organisation to other. The product development organisation leadership took part in every workshop and meeting connected to the railyard 2.0 work, but how much the workshops influenced the cooperation procedures, and department composition remains unknown.

While one cannot and should not try to copy-paste organisational culture and structure, the way how the workshops were constructed and the means used can be seen as universal in nature. Cross-functional teams or groups and facilitation is shown to create better results than homogenous groups in the siloed setting. Especially if the aim is to create a core team responsible for spreading the change to every level of the company (Kotter, 2012b; Humble, Molesky and O'Reilly, 2015; Kersten, 2018; Gibbons, 2019.)

Inspiration could also be found from the suggested changes and workshop insights. While none of the recommended practices or changes are unique or new, companies looking to start moving away from the traditional way of managing teams and projects can benefit from understanding the problems and suggested improvements presented in this work.

Limiting amount of work in process (WIP) is universal and currently cheapest known practice making time for improvement, reducing burnout and churn while improving the lead times and customer experience when combined with adequate product discovery or equivalent practices (Vähäniitty, 2006; Liker and Convis, 2011; Skeie, 2014; Humble, Molesky and O'Reilly, 2015; Modig and Åhlström, 2016; DeGrandis and DeMaria, 2017; Kersten, 2018.)

This work was limited by a lack of shared and commonly understood vision of agile transformation and a lack of accurate metrics due to infrastructure support or metrics capturing the business value. Further study should be done to understand better whether a

<sup>&</sup>lt;sup>16</sup> Change Lead Time metric saw movement from 17 days to 15 days but at the same time there was quite heavy tuning of metrics were calculated in JIRA. Number of deployments went up as expected but outside digital solutions train mainly due automating the metric collection for more of the systems. Predictability improved couple of percentages but it is hard to say what happened, based on general feedback the reason could be that ARTs did not take unrealistic amount of work in to the process. Late Bug Count rose as the metrics was collected from more systems.

shared understanding of agile shared company-wide coupled with trustworthy metrics changes the outcomes of change initiatives in bottom-up and decentralized agile transformations.

#### References

#### Printed

Adkins, L., 2010. Coaching agile teams: a companion for ScrumMasters, agile coaches, and project managers in transition. Upper Saddle River, NJ: Addison-Wesley.

Adzic, G. and Poppendieck, T., 2012. *Impact mapping: making a big impact with software products and projects*. Woking: Provoking Thoughts Limited.

Beattie, T., Hepburn, M. and O'Connor, N., Spring, Donal, 2021. *DevOps culture and practice with OpenShift: deliver continuous business value through people, processes, and technology.* first edition ed. Birmingham: Packt Publishing Ltd.

DeGrandis, D. and DeMaria, T., 2017. Making work visible: exposing time theft to optimize work & flow. First edition ed. Portland, OR: IT Revolution Press.

Denning, S., 2020. The quest for genuine business agility. *Strategy & Leadership*, [online] 48(1), pp.21-28. https://doi.org/10.1108/SL-11-2019-0166.

Dudjak, M. and Martinović, G., 2020. An API-first methodology for designing a microservice-based Backend as a Service platform. *Information Technology And Control*, [online] 49(2), pp.206-223. https://doi.org/10.5755/j01.itc.49.2.23757.

Edmondson, A., 1999. Psychological Safety and Learning Behavior in Work Teams. *Administrative Science Quarterly*, [online] 44(2), pp.350-383. https://doi.org/10.2307/2666999.

Edmondson, A.C., 2019. The fearless organization: creating psychological safety in the workplace for learning, innovation, and growth. Hoboken, New Jersey: John Wiley & Sons, Inc.

Forsgren, N., Humble, J. and Kim, G., 2018. Accelerate: the science behind DevOps: building and scaling high performing technology organizations. First edition ed. Portland, Oregon: IT Revolution.

Gene, K., 2019. The unicorn project: a novel about digital disruption, developers, and overthrowing the ancient powerful order. First Edition ed. Portland, OR: IT Revolution.

Gibbons, P., 2019. The science of organizational change: how leaders set strategy, change behavior, and create an agile culture. Fort Collins, CO: Phronesis Media.

Gray, D., Brown, S. and Macanufo, J., 2010. *Gamestorming: a playbook for Innovators, Rulebreakers, and Changemakers*. Beijing, Cambridge u.a: O'reilly.

Haslam, S.A., Reicher, S.D. and Platow, M.J., 2012. *Uusi Johtamisen Psykologia*. first edition ed. Helsinki: Gaudeamus Oy.

Hesselberg, J., 2019. Unlocking agility: an insider's guide to agile enterprise transformation. The Addison-Wesley signature series. Boston [MA]: Addison-Wesley.

Humble, J. and Farley, D., 2010. Continuous delivery: reliable software releases through build, test, and deployment automation. Upper Saddle River, NJ: Addison-Wesley.

Humble, J., Molesky, J. and O'Reilly, B., 2015. *Lean enterprise*. First edition ed. The lean series. Beijing: O'Reilly.

Johnson, J. and Mulder, H., 2021. *Endless modernization*. The Standish Group International Inc.

Junger, M., 2012. Kommenttipuheenvuoro: Johtaminen monimutkaisessa maailmassa. In: J. Perttula and A. Syväjärvi, eds. *Johtamisen Psykologia*, first edition. Juva: PS-kustannus.

Kersten, M., 2018. Project to product: how to survive and thrive in the age of digital disruption with the flow framework. First Edition ed. Portland, OR: IT Revolution Press.

Kim, G., Behr, K. and Spafford, G., 2014. The Phoenix project: a novel about it, DevOps, and helping your business win; [revised with new resource guide]. Portland, Or: IT Revolution Press.

Kim, G., Debois, P., Willis, J., Humble, J. and Allspaw, J., 2016. The DevOps handbook: how to create world-class agility, reliability, & security in technology organizations. First edition ed. Portland, OR: IT Revolution Press, LLC.

Knaster, R. and Leffingwell, D., 2020. Safe 5.0 distilled: achieving business agility with the scaled agile framework. Third edition ed. Boston: Addison-Wesley.

Kotter, J.P., 2012a. Accelerate - How the most Innovative companies capitalize on today's rapid-fire strategic challenges and still make their numbers. *Harvard Business Review*, 5, Nov., p.13.

Kotter, J.P., 2012b. Leading change. Boston, Mass: Harvard Business Review Press.

Liker, J.K. and Convis, G.L., 2011. The Toyota way to lean leadership: achieving and sustaining excellence through leadership development. New York: McGraw-Hill.

Martin, K., 2012. The outstanding organization: generate business results by eliminating chaos and building the foundation for everyday excellence. New York: McGraw-Hill.

Martin, K. and Osterling, M., 2014. Value stream mapping: how to visualize work and align leadership for organizational transformation. New York: McGraw-Hill.

McMackin, J. and Heffernan, M., 2021. Agile for HR: Fine in practice, but will it work in theory? *Human Resource Management Review*, [online] 31(4), p.100791. https://doi.org/10.1016/j.hrmr.2020.100791.

Modig, N. and Åhlström, P., 2016. *This is lean: resolving the efficiency paradox*. First edition ed. Stockholm: Rheologica Publishing.

Old, J., 2015. Reinventing management thinking: using science to liberate the human spirit. Wimborne: Team Business Development Ltd.

Oosthuizen, R. and Pretorius, L., 2016. ASSESSING THE IMPACT OF NEW TECHNOLOGY ON COMPLEX SOCIOTECHNICAL SYSTEMS. *South African Journal of Industrial Engineering*, [online] 27(2). https://doi.org/10.7166/27-2-1144.

Pichler, R., 2020. How to lead in product management: practices to align stakeholders, guide development teams, and create value together. Wendover, Buckinghamshire: Pichler Consulting.

Putta, A., Paasivaara, M. and Lassenius, C., 2018. Adopting scaled agile framework (SAFe): a multivocal literature review. In: *Proceedings of the 19th International Conference on Agile* 

Software Development: Companion. [online] XP '18 Companion: 19th International Conference on Agile Software Development. Porto Portugal: ACM.pp.1-4. https://doi.org/10.1145/3234152.3234164.

Rother, M., 2010. Toyota kata: managing people for improvement, adaptiveness, and superior results. New York: McGraw Hill.

Skeie, T., 2014. Does Limit on Work-In-Progress (WIP) in Software Development Matter? Master's Thesis. University of Oslo.

Skelton, M. and Pais, M., 2019. *Team topologies: organizing business and technology teams for fast flow.* First edition ed. Portland, OR: IT Revolution.

Syväjärvi, A. and Vakkala, H., 2012. Psykologinen johtamisorientaatio - positiivisuuden merkitys ihmisten johtamisessa. In: A. Syväjärvi and V. Pietiläinen, eds. *Johtamisen Psykologia*, first edition. Juva: PS-kustannus.

The Standish Group International Inc., 2015. CHAOS Report 2015.

Tökkäri, V., 2012. Kokemuksellinen organisaatio. In: J. Perttula and A. Syväjärvi, eds. *Johtamisen Psykologia*, first edition. Juva: PS-kustannus.

Tökkäri, V., 2019. Johtajuus kokemuksellisessa organisaatiossa. In: J. Perttula and A. Syväjärvi, eds. *Johtamisen psykologia*. Jyväskylä: PS-kustannus.

Vähäniitty, J., 2006. Pipeline Management. In: C. Lassenius, ed. *PACING SOFTWARE PRODUCT DEVELOPMENT: A Framework and Practical Implementation Guidelines*, second edition. Espoo: HELSINKI UNIVERSITY OF TECHNOLOGY SOFTWARE BUSINESS AND ENGINEERING INSTITUTE.pp.37-48.

### Electronic

Aghina, W., De Smet, A., Murarka, M. and Collins, L., 2022. *The keys to organizational agility* | *McKinsey*. [online] Available at: <a href="https://www.mckinsey.com/business-functions/people-and-organizational-performance/our-insights/the-keys-to-organizational-agility">https://www.mckinsey.com/business-functions/people-and-organizational-performance/our-insights/the-keys-to-organizational-agility</a> [Accessed 7 May 2022].

Amazon Web Services, 2022. *DevOps - Amazon Web Services (AWS)*. [online] Amazon Web Services, Inc. Available at: <a href="https://aws.amazon.com/devops/">https://aws.amazon.com/devops/</a> [Accessed 7 May 2022].

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D., 2001. *Manifesto for Agile Software Development*. [online] Manifesto for Agile Software Development. Available at: <a href="https://agilemanifesto.org/">https://agilemanifesto.org/</a> [Accessed 18 October 2021].

Burlton, R.T., Ross, Ronald G. and Zachman, J.A., 2017. *The Business Agility Manifesto*. [online] Available at: <a href="https://busagilitymanifesto.org/">https://busagilitymanifesto.org/</a> [Accessed 7 May 2022].

Edwards, D., 2012. The History Of DevOps. *IT Revolution*. Available at: <a href="https://itrevolution.com/the-history-of-devops/">https://itrevolution.com/the-history-of-devops/</a> [Accessed 18 October 2021].

Geraghty, Tom, 2020. A Short Critique of SAFe - The Scaled Agile Framework | Tom Geraghty. *DevOps, Digital Transformation, and Global Health*. Available at: <a href="https://tomgeraghty.co.uk/index.php/a-short-critique-of-safe/">https://tomgeraghty.co.uk/index.php/a-short-critique-of-safe/</a> [Accessed 7 May 2022].

Holmes, M., 2019. Microsoft CEO: "Every Company is Now a Software Company" - Via Satellite -. [online] Via Satellite. Available at:

<a href="https://www.satellitetoday.com/innovation/2019/02/26/microsoft-ceo-every-company-is-now-a-software-company/">https://www.satellitetoday.com/innovation/2019/02/26/microsoft-ceo-every-company-is-now-a-software-company/</a> [Accessed 18 October 2021].

Janlén, J., 2019. Health checks for Teams and Leadership. [online] Crisp's Blog. Available at: <a href="https://blog.crisp.se/2019/03/11/jimmyjanlen/health-checks-for-teams-and-leadership">https://blog.crisp.se/2019/03/11/jimmyjanlen/health-checks-for-teams-and-leadership</a> [Accessed 26 April 2022].

Jeffries, Ron, 2014. *Issues with SAFe*. [online] Available at: <a href="https://ronjeffries.com/xprog/articles/issues-with-safe/">https://ronjeffries.com/xprog/articles/issues-with-safe/</a> [Accessed 7 May 2022].

Leffingwell, D., 2020. About. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/about/">https://www.scaledagileframework.com/about/</a>> [Accessed 18 October 2021].

Scaled Agile Inc., 2020. *Customer Stories: Putting SAFe to Work*. [online] Scaled Agile. Available at: <a href="https://scaledagile.com/insights-customer-stories/">https://scaledagile.com/insights-customer-stories/</a> [Accessed 18 October 2021].

Scaled Agile Inc., 2021a. Agile Release Train. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/agile-release-train/">https://www.scaledagileframework.com/agile-release-train/</a> [Accessed 18 October 2021].

Scaled Agile Inc., 2021b. Agile Teams. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/agile-teams/">https://www.scaledagileframework.com/agile-teams/</a> [Accessed 6 March 2022].

Scaled Agile Inc., 2021c. Business Owners. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/business-owners/">https://www.scaledagileframework.com/business-owners/</a> [Accessed 1 May 2022].

Scaled Agile Inc., 2021d. *Communities of Practice - Scaled Agile Framework*. [online] Available at: <a href="https://www.scaledagileframework.com/communities-of-practice/">https://www.scaledagileframework.com/communities-of-practice/</a> [Accessed 15 May 2022].

Scaled Agile Inc., 2021e. Continuous Learning Culture. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/continuous-learning-culture/">https://www.scaledagileframework.com/continuous-learning-culture/</a> [Accessed 19 October 2021].

Scaled Agile Inc., 2021f. Enablers. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/enablers/">https://www.scaledagileframework.com/enablers/</a> [Accessed 1 May 2022].

Scaled Agile Inc., 2021g. Essential SAFe. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/essential-safe/">https://www.scaledagileframework.com/essential-safe/</a>> [Accessed 19 October 2021].

Scaled Agile Inc., 2021h. Implementation - Create a Lean-Agile Center of Excellence. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/lace/">https://www.scaledagileframework.com/lace/</a> [Accessed 17 April 2022].

Scaled Agile Inc., 2021i. Implementation - Prepare for ART Launch. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/prepare-for-art-launch/">https://www.scaledagileframework.com/prepare-for-art-launch/</a> [Accessed 19 October 2021].

Scaled Agile Inc., 2021j. Iterations. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/iterations/">https://www.scaledagileframework.com/iterations/</a> [Accessed 20 March 2022].

Scaled Agile Inc., 2021k. Lean Portfolio Management. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/lean-portfolio-management/">https://www.scaledagileframework.com/lean-portfolio-management/</a> [Accessed 18 October 2021].

Scaled Agile Inc., 2021l. Principle #9 - Decentralize Decision-Making. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/decentralize-decision-making/">https://www.scaledagileframework.com/decentralize-decision-making/</a> [Accessed 14 May 2022].

Scaled Agile Inc., 2021m. Program Increment. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/program-increment/">https://www.scaledagileframework.com/program-increment/</a> [Accessed 20 March 2022].

Scaled Agile Inc., 2021n. SAFe Lean-Agile Principles. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/safe-lean-agile-principles/">https://www.scaledagileframework.com/safe-lean-agile-principles/</a> [Accessed 18 October 2021].

Scaled Agile Inc., 2021o. Solution. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/solution/">https://www.scaledagileframework.com/solution/</a> [Accessed 19 October 2021].

Scaled Agile Inc., 2021p. System Team. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/system-team/">https://www.scaledagileframework.com/system-team/</a> [Accessed 1 May 2022].

Scaled Agile Inc., 2021q. Team and Technical Agility. *Scaled Agile Framework*. Available at: <a href="https://www.scaledagileframework.com/team-and-technical-agility/">https://www.scaledagileframework.com/team-and-technical-agility/</a> [Accessed 7 May 2022].

Scaled Agile Inc., 2022a. Metrics. Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/metrics/">https://www.scaledagileframework.com/metrics/</a> [Accessed 22 May 2022].

Scaled Agile Inc., 2022b. SAFe 5.0 Framework. [online] Scaled Agile Framework. Available at: <a href="https://www.scaledagileframework.com/">https://www.scaledagileframework.com/</a> [Accessed 7 May 2022].

Smith, D., Villalba, D., Irvine, M., Stanke, D. and Harvey, N., 2021. 2021 Accelerate State of DevOps. Accelerate State of DevOps. [online] DORA Research. Available at: <a href="https://cloud.google.com/blog/products/devops-sre/announcing-dora-2021-accelerate-state-of-devops-report">https://cloud.google.com/blog/products/devops-sre/announcing-dora-2021-accelerate-state-of-devops-report</a> [Accessed 18 October 2021].

Vodde, B. and Larman, C., 2014. *About LeSS*. [online] Large Scale Scrum (LeSS). Available at: <a href="https://less.works/resources/about">https://less.works/resources/about</a> [Accessed 18 October 2021].

Wikipedia, 2022. DevOps. In: Wikipedia. [online] Available at: <a href="https://en.wikipedia.org/w/index.php?title=DevOps&oldid=1084496188">https://en.wikipedia.org/w/index.php?title=DevOps&oldid=1084496188</a> [Accessed 7 May 2022].

#### Unpublished

Jaakkonen Antti, 2020. Railyard 2.0 - Business & Technical agility at TC1 - Constantly organizing and aligning around flow of business and technology change.

# 7 Figures

Figure 1: TC1 high-level organisation	9
Figure 2: How SAFe virtual organisation pull people from line organisation	10
Figure 3: TC1 virtual organisation topology	11
Figure 4: SAFe Innovation riptide	12
Figure 5: Agile release train pulling people from different departments	12
Figure 6: Agile team composition	13
Figure 7: Dependencies created by centralized teams	13
Figure 8: Shared service team as the pace setter	17
Figure 9: How shared service teams can form a bottleneck for value delivery	17
Figure 10 DANA operating model (Liker and Convis, 2011)	19
Figure 11: Initial idea of development portfolio flow	28
Figure 12: Initial development to production flow principles	28
Figure 13: Interaction in Sociotechnical systems (Oosthuizen and Pretorius, 2016)	30
Figure 14: Different Open-Explore-Close combinations	32
Figure 15: First alternative for re-organising TC1 agile release trains	34
Figure 16: Evaluated pros & cons of alternative one	
Figure 17: Second alternative for organising TC1 agile release trains	37
Figure 19: Questioned core values of TC1 agile scaling framework from presentation ma	terial
	40
Figure 20: Known problems and suspected causes of slow development	41
Figure 21: Expectations towards collaboration and workshop outcomes	42
Figure 22: Key metric definition example from Impact Mapping book (Adzic and Poppen	
2012, p. 43)	44
Figure 23: TC1 Railyard 2.0 key performance indicators	
Figure 24: Relation of Railyard 2.0 key metrics and customer / business facing initiative	es 46
Figure 25: Thinking tools used at third workshop	46
Figure 26: Impact map with development teams in focus	48
Figure 27: Impact map with business owners in focus	48
Figure 28: Impact map with product owners in focus	49
Figure 29: Identified gaps in architecture and portfolio work	49
Figure 30: Railyard 2.0 shared principles and dot votes	51
Figure 31: Railyard 2.0 Shared Practices and dot votes	52
Figure 32: Team Topologies tip about valuable collaboration (Skelton and Pais, 2019)	53
Figure 33: Snapshot of roles and responsibilities workshop	56
Figure 34: Evaluating different release train organization scenarios	57
Figure 35: TC1 system mapped into team topologies informed model	58
Figure 36: TC1 original feature team idea	59

Figure 37: TC1 railyard 2.0 feature team idea	60
Figure 38: Feature team collaboration when building APIs	61
Figure 39: Agreed mechanism to keep the virtual organisation fit for purpose	62
Figure 40: Power of virtual whiteboards	62
Figure 41: Proposed Simplified Lean Portfolio Model for TC1 Agile Release Trains	66
Figure 42: Guidance on backlog levels	66
Figure 43: Proposed New Agile Release Train organisation for TC1	68
Figure 44: Using the forums that already exist by the framework	69
Figure 45: Proposed TC1 Communities of Practice Model	71
Figure 46: Head of Engineering role proposal with operating model	72
Figure 47: Proposed Task Force team -model	73

# 8 Tables

Table 1: Evaluated pros & cons of alternative two	39
Table 2: Planned schedule for proposed changes	74