A Balancing Act

Defining a control-oriented approach to public sector agility

Appendices 1-3
Theoretical Study, Interviews and Case Analysis

Ferre Westermann
Delft University of Technology
Faculty of Technology, Policy & Management
Final Version
4 March 2019

TU Delft

Front page photo credit goes to Milada Vigerova

Table of Content

Appendix 1: Theoretical study

This Appendix elaborates on the academic basis for control mechanisms to be retrieved from existing scholarly work. Firstly, an overview of the different clusters of control mechanisms is presented, including their specific mechanisms. Then, the Appendix illustrates how these mechanisms have been identified from four relevant application areas: agile-oriented methodology, agile project management, agile governance, and management in networks. The chapter ends by summarizing the findings and showcasing how these together provide the selection of control mechanisms per cluster.

1.1 Control clusters and their mechanisms

At the end of the previous chapter ten categories for control mechanisms were already identified. Academic literature on theories and frameworks related to agility follows these clusters and analysis of the available literature provides a list of control mechanisms which can again be categorized to these clusters. Table A1.1 provides an overview of the different control mechanisms categorized to their specific cluster.

Portfolio and budget • Assess whether there is indeed presence of uncertainty (internal and/or external) and whether the organizational set-up serves for agility (common objectives, shared agile culture) • Define a complete business case (stakeholders, goals, budgets, specifications) • Avoid pre-project fixed budgeting (especially when specifications are unknown) • Focus the budget on objectives and value of the project, not on time and motion from traditional EVMS • Set financial constraints on the highest level
Continuous change • Continuous adaptation • Perform continuous estimation, using empirical data, and adaptation • Use Agile EVMS • Measure flow (WIP, SLE (% and days), cycle time, throughput and item age) and keep stable • Visualize the flow process

- Use one portfolio, make abstract top-down and bottom-up estimations (detailed if needed) and continuously adapt it

Collaboration and feedback

- Define a shared language throughout the organization (incl. definition of done, independence, interdependence, self-organization) to create alignment
- Create and communicate a common vision and mission to create alignment
- Use collaborative development with interaction between developers and operational staff
- Have formal events for feedback
- Define roles responsible to provide linkage between teams
- Facilitate bottom-up information provision
- Create an environment of trust (more in leadership section)
- Have both a team room and a space for privacy
- When staff is not in one location, assemble them at the start (2-6 weeks) and reassemble them later on; also use technical communication tools
- Educate and train agility throughout the whole organization

Teams and roles

- Define clear roles and rules which should be adhered to and continuously adapt the team roles
- Have distinguished roles for those responsible for the architecture of the development process and those managing the content (like scrum master and product owner)
- Have facilitator and secretary roles for each meeting
- Keep teams diverse and team membership stable
- Have one person/team in charge of stakeholder interaction and feeding user requirements into the process
- Have different roles for creating the architecture of the stakeholder involvement process and for managing the process
- Ensure the board's membership reflects stakeholders' interests

Events and procedures

- Use formal events for planning, execution, review and retrospective on separate levels for simplicity and efficiency
- Have strict procedures for the formal events with tight planning for efficiency

- Have separate meetings for tactical and strategic issues
- Organize special-topic meetings for important subjects that cannot be discussed in the formal events
- Have a stable sprint time that is not too long to prevent complexity
- Have daily stand-ups to facilitate interaction
- Change sprints when necessary to reach the sprint goal
- Cancel sprints when the goal cannot be reached anymore (to be avoided)

User and stakeholder involvement

- Share data or a system demo for feedback
- Involve stakeholder participation at every level
- Have board membership reflect the stakeholders
- Identify the right customer
- Select required partners on both their power and potential substantial contribution
- Perform continuous actor analysis
- Ensure a fair process and protect parties' core interests
- Create a sense of urgency for participation
- Create (a sense of) progress
- Consider the process as a multi-issue game
- Use framing
- Define negotiated knowledge

Technology and tools

- Have a systems and software architecture in place early on
- Prevent path dependency and use flexible infrastructure
- Use Build Tools to manage the development of increments
- Use Continuous-Integration Tools to integrate and test developed work
- Use continuous and automated testing and deployment
- Track data through logging and monitoring tools and survey data
- Use a Collaboration Framework to integrate all the tools and their data
- Use productivity tools which foster innovation, creativity, and collaboration.
- Use communication technology to facilitate interaction in teams that are not in one location

Agile leadership • A manager should have agile skills (diplomatic, effective communicator and listener, analytical thinker) • A manager should create an environment of trust (for diversity, inclusiveness, learning, mentoring, critical reflection and creativity) • Prevent and eliminate dysfunctional elements of self-organization (absence of trust, fear of conflict, lack of commitment, and absence of focus) • A manager should motivate people • A manager should promote accountability
Agile mindset • Only focus on tensions that arise • Go for workable solutions • Balance between adaptiveness and stability (mature and large organizations) • Be both business and agile driven and care for tensions technology and business
Organizational set-up • Set distinctive paths for development and operations but collaboration • Consider the agile process as different structures per level • Keep the requisite organization in mind when making changes • Adapt control mechanisms to the organizational context

Table A1.1: The ten clusters with their specified control mechanisms

The mentioned control mechanisms are retrieved from academic literature on agile methods or similar frameworks and theories. In total, have been used which can be categorized into four application areas: agile frameworks or methods which are often used to implement agility (agile oriented methods), agile project management, agile governance, and network and process management. Agile-oriented methods provide direct practical relevance given the wide use of such mechanisms to accomplish and create agility. Agile project management uses existing academic literature on how agility should specifically be managed and therefore has a more management perspective rather than process or result oriented. Agile governance methods deliver more coordination related mechanisms. Network and process management set a spotlight on customer and stakeholder interaction which is otherwise left unmentioned since the other application areas generally mention its importance but do not provide specific

recommendations on how this should be achieved. Together, these four areas provide a complementary picture.

The following paragraphs explore these application areas and provide insight into different methodological and theoretical frameworks and their vision on control and appropriate mechanisms. It therefore does not provide a detailed image of each of them, but an extensive summary to grasp the core aspects, goals and mechanisms of a certain method or theory.

1.2 Agile-oriented methodology

The first cluster of frameworks consists of approaches which are often associated with agility or are even created as agile methods. Scrum, perhaps the most popular framework associated with agility is first discussed as its characteristics are recurring elements in some of the other approaches. Then Kanban, which combines agility with lean principles is discussed as another popular method. DevOps, which focuses on the distinction and interaction between development and operational processes – something not always clearly defined – is considered, before holacracy, which provides a look into the increasing trend of self-organization linked to agility, while the paragraph ends with SAFe which can be used to apply agility on different levels of an organization.

1.2.1 Scrum

Scrum is a framework which is often used for agile projects with a focus on the project team level. It contains roles, events and artifacts which can be used for self-organizing Scrum Teams to create a product for a complex problem in a short-cycled an iterative manner. As it is a widely used concept with clear demarcation and characteristics, its definition of required control is valuable for agile projects which almost always entail some sort of scrum element. While there is a vast body of literature on scrum (for clear overview of examples see (Azanha, Argoud, Camargo Junior, & Antoniolli, 2017) its academic foundation generally lies in “The Scrum Guide” by Schwaber and Sunderland (2013) which was used to create this summary.

The Pillars of Scrum Theory

Scrum is empirically focused where the information regarding the current state and planning solely comes from the progress already made. There are therefore three aspects which should not be compromised during a scrum-based project: transparency, inspection and adaptation. *Transparency* entails firstly that that there should be common standards and language of the

process, in which it is important there is a shared definition of "Done". *Inspection* relates to frequent tracking of progress and detecting of undesired variances. *Adaptation* means that when it becomes clear that the process or product is altered when the inspection shows unacceptable deviation.

Roles and rules

Scrum Teams are self-organizing and cross-functional teams which are flexible, creative and productive, that deliver products iteratively and incrementally. They consist of a Product Owner, Development Team and a Scrum Master.

The Product Owner

The Product Owner is responsible and accountable maximizing the value of the product to be delivered and therefore as well for the managing the Product Backlog (see artifacts). His decisions should be transparent within the Product Backlog and should furthermore be respected by the entire organization. Even though the Product Owner may delegate parts of his/her tasks, he/she remains accountable.

The Development Team

The Development Team works on a releasable increment of the "Done" product during the sprint. They are self-organizing and cross-functional and do not have any sub-teams. Even though individual skills and efforts are very important in Scrum, accountability belongs to the Development Team as a whole. No one tells the Development Team what to do, but they use the requirements set by the Product Owner on the Product Backlog. The size of a team can be anything between three and nine members: a fewer amount would decrease interaction and entails lower productivity, while larger teams become too large to coordinate in complex empirical processes.

The Scrum Master

While the Product Owner is responsible for maximizing the value of a to be developed product, the Scrum Master is responsible that Scrum is understood and acted upon in a proper manner. She/he therefore has a more procedure focused role and not a value-driven role. The Scrum Master thus facilitates the Scrum Team in adhering to its rules and helps external parties understand how to interact with the Scrum Team. He/she therefore helps to maximize the impact of interactions. Within a Scrum Team a Scrum Master thus facilitates scrum events,

coaches the team, and removes impediments. Within the organization he/she helps stakeholders understand scrum, cooperates with other Scrum Masters, leads and coaches in the process of organizational scrum adaptation and plans its implementation.

Events

The key of Scrum efficiency is that planning is tight and occurs in obedience to the rules. Each event has a specific purpose and failure to perform these events in a proper manner, leads to reduced transparency and inspection, and thus adaptation.

The Sprint

The Sprint is the core of Scrum as this sets the timeframe in which potentially releasable and usable product increment is developed according to the definition of "Done". Sprints consists of Sprint Planning, Daily Scrums, the development work, Sprint Review and the Sprint Retrospective. Sprints are not supposed to last over one month to enhance inspection and adaptation: a longer period can lead to increased complexity and risk.

Certain rules provide the setting in which the Sprint evolves:

- "No changes are made that would endanger the Sprint Goal;
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned" (p.7)

Sprint Cancellation

Sprints can, be cancelled if the Sprint Goal cannot be met with anymore due to a change in strategy or conditions set by the market or available technology. Sprint Cancellation is very rare, but when they occur they bring a high chance of "traumatizing" the Scrum Team.

Sprint Planning

As can be expected, the Sprint Planning takes place in the starting phase. It is limited to a maximum of eight hours for a one-month sprint and discusses what can be delivered in the upcoming sprint and how it will be achieved. The Product Backlog and latest project increment are used as artifacts, while the future capacity of the Development Team during the respective sprint and its past performance are also considered. Only the Development team can decide what can be achieved during a sprint which is called the Sprint Goal. To define this, it chooses

items from the Product Backlog, which can be further explained by the Product Owner. At the end, the Development Team should be able to illustrate to both the Scrum Master and the Product Owner how it plans to accomplish the stated Sprint Goal. Other parties can be invited to present further expertise and advise related to the Product Backlog items.

Daily Scrum

The Daily Scrum serves to synchronize activities by discussing the work done between the previous and current session and set the plan for the coming day. It is held at the same place and same time every day and lasts only fifteen minutes to prevent complexity. It is the task of the Scrum Master to plan the meeting while the Development Team is in charge of actually conducting it. The Daily Scrum can only be attended by Development Team members.

Sprint Review

During the Sprint Review, which takes place at the end of the Sprint and lasts four hours for a one-month Sprint, the developed increment is inspected and required changes are made to the Product Backlog. It provides a good moment for collaboration between the Scrum Team and stakeholders to reflect on the Sprint.

Sprint Retrospective

Taken as a separate event from the Sprint Review, the Retrospective, which lasts three hours for a Sprint of one month, entails the self-inspection of the Scrum team to identify areas of improvement and act upon them before or during the next sprint in case this has not been done during the process already. In case something went wrong during the Sprint, the Review provides a good opportunity to stand still at such moments to identify them and respond adequately. A core characteristic of this is the adaptation of the definition of “Done”.

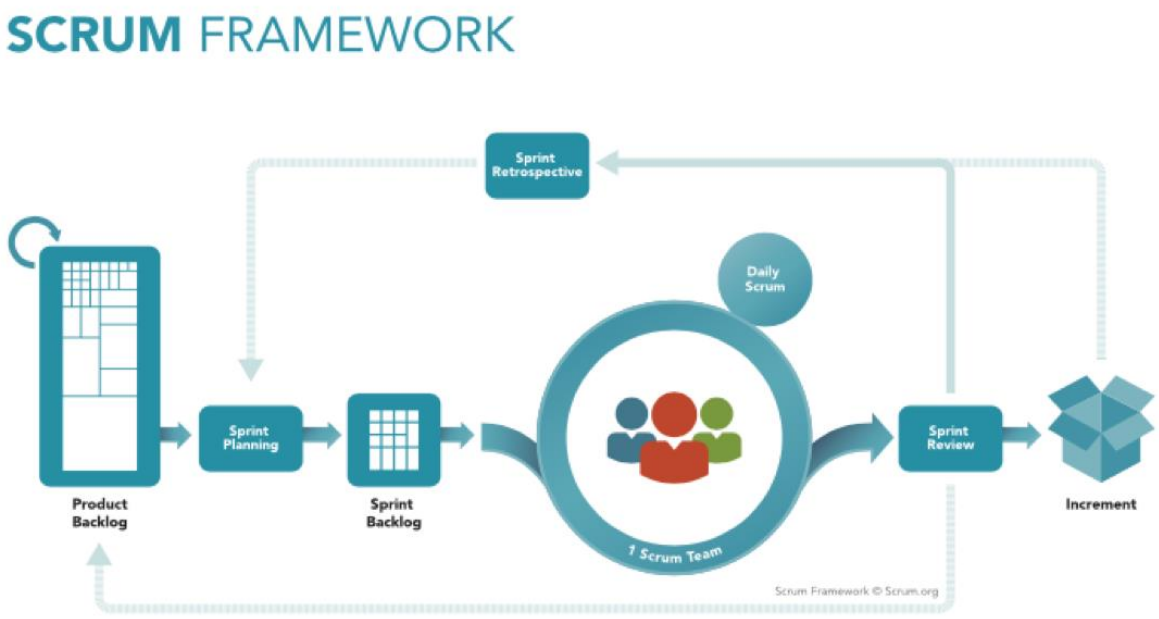


Figure A1.1: Scrum framework Image from: https://www.scrum.org/resources/scrum-framework-poster

Artifacts

Product Backlog

The Product Backlog serves as a list of requirements and serves as the only document to be used for this purpose. Regarding rules, the Product Owner is responsible for not only the content but also the sharing of the Product Backlog. The Product Backlog is ever changing as new requirements will come up along the process. The items on the list have a description, order, estimate and value. While there are multiple teams working on a product, there is only one Product Backlog. The specification of estimates can only be done by the Development Team. In complex problem environments, it is important to link decision-making to empirical findings from the development already made.

Sprint Backlog

Each sprint some items are selected from the Product Backlog which should be met with. Together these form the Sprint Backlog which also includes a plan on how to develop the product implement and reach the Sprint Goal (the specific objective to be reached within that sprint). The Sprint Backlog should be detailed and can be modified during the sprint when more is learned. Every time work is performed or completed, an estimate is made to assess the remaining work.

Definition of "Done"

While not specifically an artifact in itself, the Definition of "Done" is a crucial aspect of achieving successful Scrum as it ensures there us a shared understanding of when work is actually considered done. This contributes to the transparency principle so crucial in scrum. It helps to assess how much work can be done within one sprint to further develop the product increment. Furthermore, there is parallel development during scrum where different teams work on the same product. This makes a shared understanding even more important to ensure a working product.

Control mechanisms

From the given description multiple control related aspects come to the forefront. These are presented in table A1.2 with a short description

Control mechanism	*Description*
Retrieving information from stakeholders	To ensure the developed product meets its requirements, interaction with stakeholders is crucial. It can be considered a form of control to ensure the product takes shape in its desired form.
Formal events and tight planning	The formal events are crucial to ensure the proceedings of scrum coincide with the principle with simplicity and therefore enhance efficiency rather than create complexity. Furthermore, they allow specific moments for estimation of progress and required change. This includes: planning, execution, review and retrospective.
Continuous estimation and adaptation in the Product and Sprint Backlogs	The Product Backlog, of which there is only one, allows for clear planning and estimates how much work still is required. It is crucial this is based on empirical findings retrieved throughout the development process. For each Sprint there is a separate backlog with this information.
Sprints can be changed or even cancelled	Changes can be made throughout the Sprint to achieve the Sprint goal. When it becomes clear this goal cannot be achieved, the whole Sprint can be cancelled. Yet, this is to be avoided as it has a lasting negative impact on Scrum teams.

Definition of "Done"	A shared definition of what "Done" entails paves way for parallel development by Scrum Teams in which the increments can work together rather than conflict. It furthermore allows for a common definition to be used for creating estimates.
Clear and agreed upon rules and roles	Scrum entails clear roles and rules to ensure a transparent and process which allows for high quality of inspection and paves way for targeted adaptation. Especially the distinctions between Product Owner, Scrum Master and Development Team are important.

Table A1.2: Control mechanisms Scrum

1.2.2 Kanban

Kanban, which, just like lean, finds its origins in the Toyota production system, is a framework which optimizes the flow of value in a development process. Junior and Godinho Filho (2010) provide an extensive overview of different variations of Kanban, which is often fine-tuned for a specific development process or market. For this research, the Kanban guide for Scrum Teams (Scrum.org & Vacanti, 2018) is used as it provides a variation of Kanban which is specifically oriented for Scrum. Kanban is an often-used concept in agility since it works well with Scrum and can be used in approaches which scale agility (such as SAFe, discussed in 2.2.3).

Principles of Kanban and its relation to Scrum

Kanban is "a strategy for optimizing the flow of stakeholder value through a process that uses a visual, work-in-progress limited pull system" (3). It aims to optimize this through providing transparency by visualization, and this is where it proves valuable for Scrum, which recognizes the importance of transparency but does not provide means on how to achieve this. Kanban furthermore presents more opportunities to inspect and adapt, which is important when it comes to control in agile approaches.

Main concepts of Kanban for Scrum

Workflow

The emphasis of Kanban on flow requires a solid definition. The framework of Kanban leaves this, however, to the Scrum Team. There are six elements such a definition should include:

Elements of definition of workflow
1. There should be **defined points** to indicate when work has started or finished
2. **Individual units of customer value** that flow through the Scrum system (such as PBIs)
3. **Different workflow states** that PBIs flow through from start to finish with at least one active.
4. Clear **policies on how work should flow** through these states.
5. How **WIP** will be **limited**
6. A pre-determined **Service Level Expectation** that indicates how long work is expected to take.

Table A1.3 Definition of workflow Kanban

The definition can impact or be impacted by artifacts of Scrum. The responsibilities and accountabilities remain the same as defined in the Scrum framework.

Service Level Expectation (SLE)

The SLE is an important part of Kanban and its control as it estimates how long it is expected to finish work by moving it through all the work flows. The SLE entails not only an estimation of days, but also a percentage of the work which will be finished. Important for the SLE is that whenever possible it is operationalized by using empirical evidence from previous work.

Kanban practices

Visualizing the workflow

One of Kanban's key elements is that transparency is enhanced through visualization. Use of the Kanban board, which signifies the work with in a scrum and the state they are in and therefore visualizes the workflow, should stimulate team members, provoke discussion and help in identifying room for improvement.

Figure A1.2 Kanban board Image from: https://agilescrumgroup.nl/wat-is-kanban-methode/

Limiting Work in Progress (WIP)

Limiting the WIP relates to the control of Kanban as too much work increases complexity and makes inspection and adaptation more difficult. It uses a "pull system" in which an item is only exercised when it is the right time to do so, for instance when there is a WIP below a certain stated limit. This very much coincides with Scrum's principles since a Sprint is meant to limit the amount of work being done at a specific moment or period.

Active Management of WIP

Just limiting the WIP is not enough, it also requires active management. This would, for instance, entail responding to blocked items, an even amount of pulling to what is finished, preventing unnecessary aging of work items, and unclogging piled up work. It is the Scrum Team's responsibility that this management is done properly.

Inspect and Adapt Workflow

I&A is already an important element of Scrum, but Kanban requires more specific and thus explicit policies which would be captured in the definition of Workflow.

Measuring Flow

Measuring flow in Kanban is key to make the flow transparent and allow for inspection and adaptation. The following four elements are the minimum which should be measured, Scrum Teams can always add more.

WIP

The amount of work items that have started but are not yet finished.

Cycle Time

The amount of time it takes a work item to go from start to finish.

Work Item Age

The amount of time which elapsed from when a work item started and the state it is at that moment.

Throughput

The amount of work items which are completed per time unit.

Kanban and Scrum events

The Sprint

The events of Scrum Sprints serve well as feedback loops for measuring Kanban flow, while the same can be said vice versa.

Sprint Planning

Kanban's flow metrics can be used when developing the Sprint Backlog.

Daily Scrums

The Daily Scrums take place at the Kanban board while the Daily Scrum is performed as usual. Furthermore, the use of Kanban requires three other things to consider: which work items are blocked and how can these be unblocked, what is the WIA of each item still in progress and which items violated their SLE, and are their factors which might impede the functioning of the Scrum Team that is not represented on the Kanban board?

Sprint Review

The Kanban flow measurements can be used to monitor the progress towards a goal. The specific reviewing of work items' SLEs can also lead to desired changes in the Product Backlog.

Sprint Retrospectives

The inspection and measurements of flow are used to define areas of improvement for the different events and processes. Furthermore, the Retrospectives serve as good moments to

reconsider and possible redefine the definition of Workflow. Last, but not least, a cumulative flow diagram, including WIP, cycle time and throughput can prove valuable insights into the Scum Team's functioning and score on planning.

Control Mechanisms

Kanban entails a couple of clear control mechanisms, which mainly have to do with assessing status, rather than prescribing actions to be taken.

Control mechanism	*Description*
Emphasis on flow and state of work	Flow is the holy grail of Kanban and a focus on it should help in prevent stagnation through blocked processes and allows for more efficient planning.
Actively manage and continuously measure flow	By continuously measuring WIP, SLE (as both number of days and percentage), cycle time, throughput and item age, Kanban provides specific measurements of past records and projections which provide valuable information to control steady flow and keep WIP limited.
Continuous inspection and adaptation	Through continuous inspection, the estimations can be adapted which enhances the workflow.
Visualization for transparency	Kanban visualizes the process which contributes to the understanding of the development processes and therefore the control. Difficulties or troubling factors, such as the clogging of work, come to the forefront more easily and can thus also quicker be solved when the right mechanisms are available.

Table A1.4 Control mechanisms Kanban

1.2.3 DevOps

DevOps is a software development method in which, as its name already signifies, development and operations are integrated through cross functional teams by using automated development, deployment and monitoring (Ebert, Gallardo, Hernantes, & Serrano, 2016; Jullian & Sangeetha, 2016). It aims to improve the cooperation between software developers and IT operators and their respective systems. DevOps embodies a combination of agile and lean concepts, such as continuous integration, and as it is being increasingly applied, entails multiple new and

upcoming technologies (Ebert et al). For the academic framework of this research, the general concept of DevOps is being presented through multiple complementing articles.

Principles

DevOps embodies aspects of three principles: flow, feedback, and continuous learning and experimentation (Kim, Debois, Willis, & Humble, 2016). The principles are interwoven and should not be considered as separate in their functioning, but rather separate in their core characteristics.

Flow: The principle of Flow focuses on visibility and transparency, limiting WIP and reducing complexity. See paragraph 2.2.2.

Feedback: DevOps is oriented at delivering fast feedback which "is critical to achieving quality, reliability, and safety in the technology value stream" (ibid). This is achieved through identifying problems when they occur, collecting and solving problems to build knowledge, push quality closer to the source, and continuously developing with an eye on operations and its non-functional requirements (ibid).

Continuous learning, collaboration and experimentation: This principle focuses on the cultural dimension surrounding DevOps to build on individual knowledge and its transition to team and even organizational knowledge (ibid). This requires enabling an environment of trust and collaboration

DevOps Architecture and tools

Jullian and Sangeetha (2016) distinguish four paths within the DevOps architecture: steering, developing/testing, deploying and operating.

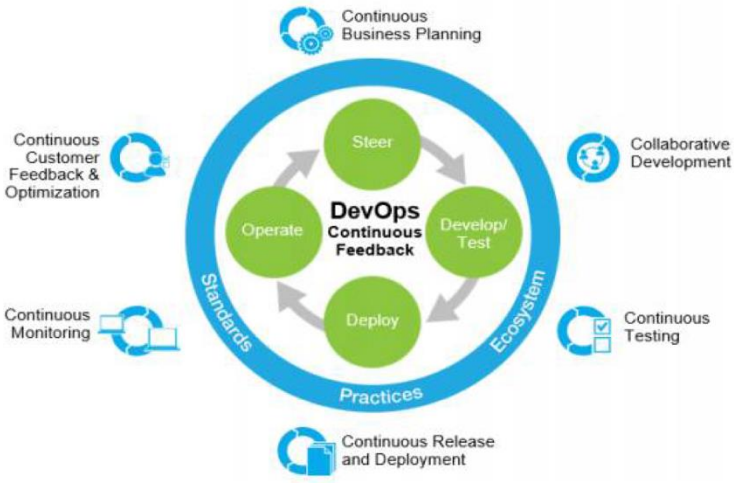


Figure A1.3 DevOps framework

Steering: This path entails the adaptation of goals when this is deemed necessary after customer feedback or findings from internal monitoring. It therefore falls parallel to evaluation moments

in Scrum and Kanban, such as the Sprint Review. This ongoing process is can be considered to be supported by continuous business planning.

Developing/testing: This step entails a twofold approach by combining continuous integration with collaborative development (ibid). Collaborative development relates to the interactions between staff in the development and operations sector through "a common set of practices and a common platform" (ibid, p.12570). Continuous integration refers to the highly frequent/continuous integration of work performed by different software developers or teams, which is preferable even done on a daily basis (ibid). DevOps uses Build Tools and Continuous-Integration Tools for this. While there are many variances of each sort, in essence the former focuses on the development and managing of software, whereas the latter ensure the continuous integration and testing of developed work (Ebert et al., 2016).

Deploying: A key aspect of DevOps is automated deployment, in which developed software can be deployed within any environment at any time after development. DevOps allows for this fast configuration by considering the software through a microservice approach in which the system and applications are broken down to "a more granular, modular level" (ibid, 98). For this to be achieved all systems devices should be connected. Orchestration technologies can assist in this process.

Operating: To fully achieve feedback from the operational processes, DevOps requires continuous monitoring and sharing of data, and continuous customer feedback and optimization (Jullian & Sangeetha, 2016). Furthermore, it is important to ensure that implemented changes do not hamper the stability and overall performance of the system's infrastructure (Ebert et al., 2016). Logging tools can assist in tracing implications of developed increments which are important to know for management. Furthermore, monitoring tools can assist in identifying potential hazards to critical infrastructure or processes before these actually unfold (ibid). DevOps uses empirical data to manage processes, but Forsgren and Kersten (2018) beware of the risks of only relying on systems data. The use of systems data fails to capture behavior situated outside the boundaries of the system, makes it difficult to gain a holistic view, is challenging to keep up to date given newly implemented increments, and generally cannot measure the implications of cultural or perceptual measures. To achieve a complementary approach on retrieving empirical data, DevOps can therefore benefit substantially from the use of survey data (ibid).

Control mechanisms

DevOps, even though strongly related to familiar methods like Scrum and Kanban, comes with its own control mechanisms given the focus on smoothening the transition/collaborations between the development and operational segments of projects and organizations.

Control mechanism	*Description*
Clear iterative cycles with distinctive paths/steps for development and operations	The distinction between different cycles of development and operations allows for clear evaluative and steering moments.
Continuous integration and testing through Build-Tools or Integration Tools	The concepts of continuous integration, along with its specific tools, allows for maintaining control over quality of the product to be developed while changes are implemented through developed increments. The integration allows for testing a more complete image of an increments impact on the system.
Collaborative development and fast feedback	This does not only allow for harmonization between developers and operations staff, it also creates an extra feedback stimulus to ensure the product meets all its requirements and the developed increments support each other.
Continuous and automated deployment	Automated deployment allows for the integration of developed increments to be applied anywhere within the given system. When performed well, the functioning of the product after deployment of developed increments is maintained and therefore control over its functioning can be achieved.
Orchestration technologies	DevOps, similar to Scrum and Kanban, considers the system as a sum of separate parts and functions. To ensure that these complement each other, rather than conflict, orchestration technologies can prove valuable.
Continuous monitoring and feedback through both logging and monitoring tools, and survey data.	To enhance the learning process and enhance efficiency continuous monitoring is required. Whereas logging tools can help improve management by analyzing the implications of implementation of developed increments, monitoring tools can assist in identifying potential major hazards before they unfold. Combining this systems

	data with survey data provides a more complementary holistic view with checks for false or incomplete information.
Continuous Customer Feedback on shared data	By maintaining a continuous process of customer feedback on the data that is shared, the business planning can integrate the feedback of the customer rapidly.
Creating an environment of trust and cooperation	The creation of an environment of trust is required to facilitate the process of transference of individual knowledge to team and even organizational knowledge.

Table A1.5 Control mechanisms DevOps

1.2.4 Self-organization through holacracy

Self-organization as an important characteristic of agility should be included in the research. While Scrum teams are generally defined as self-organizing, there are practices which further explore the concept of self-organization that might bring interesting perspectives on how to exercise control. Holacracy, related to agility, lean and peer-governance systems, is a well-known example on how to consider self-organization of organizations and teams (Robertson, 2007). Finding its roots in sociocracy, requisite organizational theory, and psychology (Robertson, 2006), holacracy dives further into "democratic" autonomous functioning than most other approaches (The Economist, 2014) as it provides a team oriented approach of David Allen's 'Getting Things Done' productivity method (Janse & Bogers, 2013). Focusing on peer-to-peer governance and clear accountability and control through roles, holacracy aims to put the focus on the organization's needs, rather than individual needs and politics. For this short overview, an article by Brian Robertson, the "godfather" of holacracy is used (2007) and is complemented by other literature for further in-depth mechanisms.

Principles

As mentioned in the introduction, holacracy focuses on the organization, which can be traced back to its name. Key in the approach is the term "holon", an autonomous unit which is part of a larger whole, and "cracy" which indicates it is the ruling factor (WhatIs, n.d.). Holacracy is therefore the consideration of an organization as a sum of self-organized teams, called circles, which function in the organization's interest. By using clearly defined roles and tasks, holacracy takes out personal elements, which enhances accountability, autonomy and transparent self-governance.

Main concepts

Dynamic steering: Holacracy, alike to other agile methods, takes dynamic steering as its main approach rather than hard control mechanisms. This is facilitated through three rules (Robertson, 2007):

1) As new information becomes available throughout the process, issues can continuously be reconsidered at any given time when deemed required. It can be that predict & control mechanisms are required at certain stages, but continuous steering sets the tone.

2) The goal is to aim for a workable solution, not the best one. The latter will surface over time when progress is made.

3) Only focus on the tensions that occur in the present.

Requisite organization: An important concept of holacracy is that of the requisite organization. It means that at any point in time, an organization has an objective ideal type of how it should be structured, and this ideal shape wants to emerge. Holacracy envisions that organizations go through a discovery process to reach that structure which allows for better self-organization. The decisions made regarding the circles and structure should evolve from the requisite organization, and thus the organizations' needs rather than personal interests or power structures (ibid).

Shared language: Common and shared language and definitions, reoccurring concepts in agile approaches, also stand at the center of holacracy. This aims to enhance communication within and between circles and take the personal element out of correspondence which aims to contribute to an environment of trust (ibid).

Feedback and double linkage: Holacracy aims at self-organization through feedback, peer governance and double linkage. Communication between circles is enhanced through procedural roles which focus on enhancing this interaction.

Roles and Rules

Circles: Circles are the self-organizing teams in holacracy which have their own aim and "authority to define and assign its own roles and accountabilities" (ibid, p.10). A circle can be defined as doing three things: making its own policy and taking decisions (leading), producing something or taking a certain actions (doing), and collecting feedback from what has been done (measuring). These three tasks together make a self-organizing feedback loop which the circle continuously flows through. Circles use integrative decision making, which entails that no member of a circle still raises an "objection" to a proposed action or decision.

The circles of which an organization consists of can, and given the notion or requisite organization, should change over time. An example of how a hierarchical organization structure would be defined in holacracy's circle approach is visualized in figure A1.4.

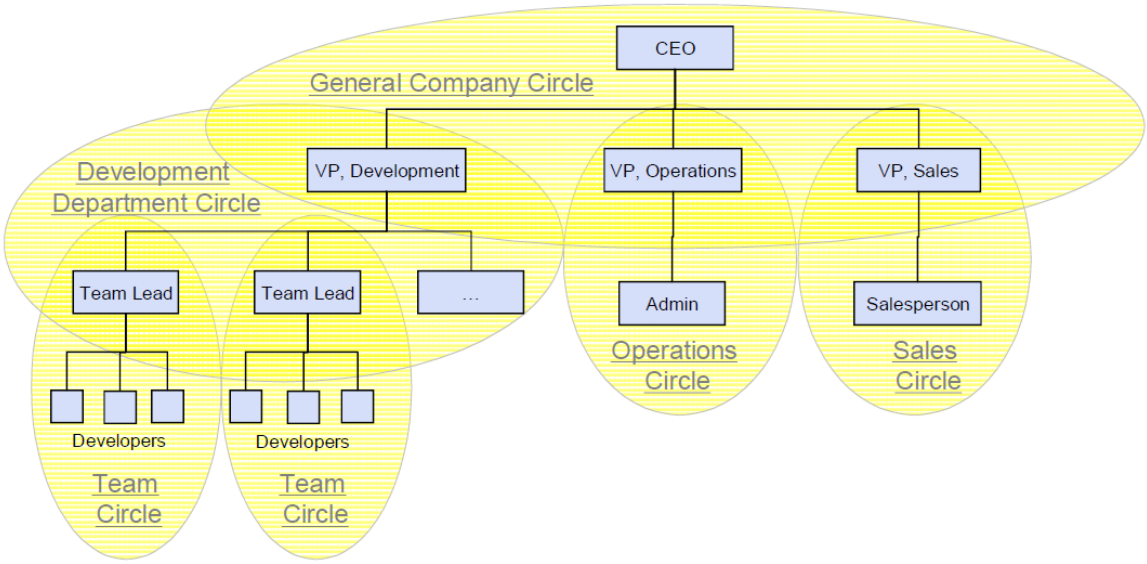
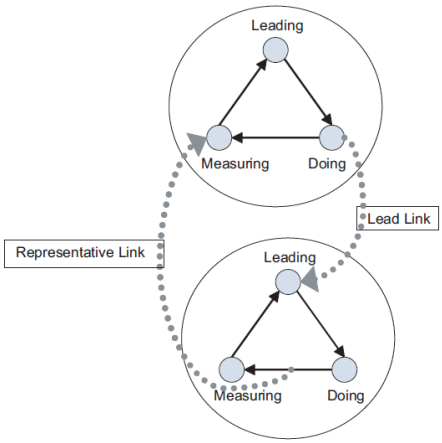


Figure A1.4 Circles in organization Robertson (2007, p.11)



Lead Link: The double-linkage of holacracy, the interactive feedback process between circles, is also capture in roles. The first one is the Lead Link. He/she "is accountable for aligning the sub-circle's results with the super-circles' needs" (p.12).

Representative Link: The counterpart of the Lead Link is the Rep Link, who is accountable for ensuring that the sub-circle's key findings can be and are transmitted to the super-circle's functioning.

Figure A1.5 Rep Link and Lead Link Robertson (2006, p.8)

Secretary: A secretary lists all the decisions made and keeps a list of the roles and responsibilities as distributed within that circle.

Facilitator: The Facilitator runs the circle meetings and ensures that the procedures are followed. The role is therefore rather similar to the Scrum Master in the Scrum approach.

Board: Holacracy also places other demands on boards as usually can be expected. They should hold governance meetings through integrative decision-making and should also be double-linked to the biggest circle which includes the operational aspects of that organization (commonly the "General Circle"). Within this double-linkage the CEO performs as the lead

link, while the rep link should be integratively elected. The board should speak for the organization and act in its spirit and towards its purpose. Holacracy therefore shapes its structure to entail “roles representing the different needs of the broader environment” (ibid p.26).

Events

Governance meetings: Each circle meets regularly to create and adapt its roles and related accountability and control options that come with them. Within these meetings no discussion of operations related issues should take place. For this there are separate operational meetings.

Integrative elections: After it has been defined which roles are required within a circle, integrative elections are held to ensure that roles are distributed to the circles’ individuals in the best manner according to a best-fit principle. While this primarily holds for the functions of rep link, facilitator and secretary, this can also be used to distribute further roles within a circle.

Operational meetings: In contrast to the governance meetings, the operational meetings focus on issues and challenges regarding the operations side of the organization. They are meant for exchanging information and making decisions regarding the integration of roles. There are different operational meetings which are followingly discussed.

Daily Stand-Up Meetings: As the name already implies, these meetings take place on a daily basis and serve for fast integration and coordination on the circles’ current standing and upcoming plans.

Tactical Meetings: In contrast to Daily Stand-Up Meetings, the Tactical Meetings are held about once a week and serve to collect data and integrate any tactical issues a circle might be facing. This leads to a list of the actions to be taken, which is similar to the Product and Sprint Backlog of Scrum.

Strategic Meetings: Taken on an even more infrequent basis, the Strategic Meetings are meant to evaluate and analyze the process made. In contrast to the Daily Stand-Up and Tactical Meetings, the items, usually only one or two, to be discussed during Strategic Meetings should be shared well beforehand to allow for preparation and in-depth discussion.

Special-Topic Meetings: These meetings, meant for only discussing a specific topic, can be initiated more on an ad-hoc basis when an issue pops up or when an issue is too big for a tactical one but cannot be included in the next strategic one.

Control mechanisms

Control mechanism	*Description*
Retrieving information from stakeholders	Within holacracy, the engagement with stakeholders and diverging interests should be reached through the board of the organization. It can be seen as a form of control to ensure the product takes shape in its desired form.
Formal events with set procedures	Holacracy entails events with clear procedures and carry a clear distinction between governance and operational, and tactical and strategic related discussions to enhance control. To ensure efficiency of meetings set procedures are useful, while Special-Topic Meetings can be used to discuss issues left out of regular meetings. There are also daily stand-up meetings
Continuous adaptation, measurement and steering	The Product Backlog allows for clear planning and estimates how much work still is required. It is crucial this is based on empirical findings retrieved throughout the development process.
Clear and agreed upon rules and roles	Even more than Scrum, holacracy stresses the importance of roles. The concept is even based on it. Distinct roles with clear accountability and identified control per role, along with the continuous adaptation of these descriptions add to the overall control over team functioning. The roles furthermore decrease the influence the personal and political implications which might hamper the effectiveness of individuals' and teams' functioning. On a board level, this should even represent the needs of the wider environment. Furthermore, each circle has a secretary and facilitator (similar to Scrum Master)
Feedback and double-linkage	Flow of information is facilitated through holacracy by the importance of feedback and continuous steering. To support inter-circle communication, the double linkage between the lead link and rep link is crucial.
Common language	A shared understanding of definitions language to be used enhances the communication and increases the control over the potential of successful integration of inter- and intra-circle work. Furthermore, it contributes to enabling an environment of trust.

Integrative decision making	Decisions should be made when there are no longer any objections present. This also counts for the election of roles.
Only focus on tensions that arise	To keep things simple, avoid unnecessary complexity and act towards the goal at hand, only tensions that arise should be dealt with in the adaption of role description and control authority.
Go for workable solutions, not the best	To avoid complexity and not get lost in unnecessary actions, the aim to solve issues should be on the base of workable solutions, not the best. This will emerge over time.
Keep the requisite organization into account	While changes are being made, development should be reflected on the requisite organization to avoid personal politics and rigid power structures.

Table A1.6 Control mechanisms holacracy

1.2.5 Scaled Agile Framework (SAFe)

Whereas Scrum focuses on the functioning of teams, scaled agile approaches look at the application of agility throughout different levels of an organization. While there are more types of scaled agility, such as Disciplined Agile Delivery and the Agility Path, this summary uses SAFe (Scaled Agile, 2017) since it has become rather popular as it has proven its added value in practice on multiple occasions (Laanti, 2014). SAFe is a framework to apply agility on a larger level throughout organizations. By differentiating between the project, program, larger solution and portfolio levels, SAFe aims to provide organizations with clear roles, artifacts and events to manage each accordingly by incorporating aspects from Kanban and Scrum.

The Core of SAFe

Whereas Scrum has Transparency, Inspection and Adaptation as its core values, SAFe defines them a bit differently as Alignment, Built in Quality, Transparency, and Program Execution. Alignment means that even when agility is scaled, there is a common thread and strategy throughout the portfolio and team members involved. Built-in Quality focuses on the level of quality, since a lack of it has a higher impact on a higher scale. Transparency, to build a trustful environment, and Program Execution to ensure a lean and responsive process through set procedures. SAFe considers the organization as a network which is self-organizing and self-managing.

SAFe incorporates both aspects from agility and lean, which focuses on maximizing value in the shortest time possible. The four pillars of lean are quite similar to agility as they stand for: respect for people and culture, a continuous development process, innovation and an everlasting search for improvement. Leadership is essential to secure successful self-organization, self-managing and cross-functionality.

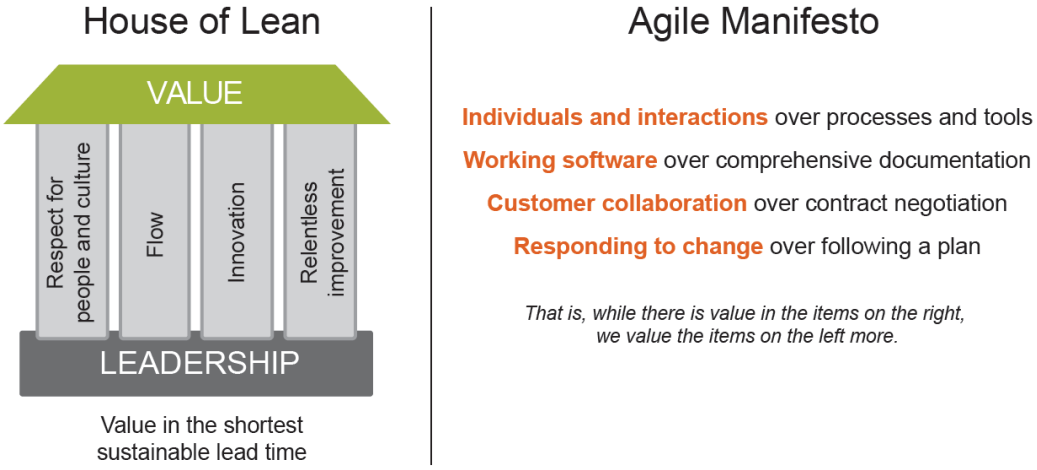


Figure A1.6 Scaled Agile (2017, p. 8)

SAFe holds 9 principles which derive from lean and agility.

SAFe Principles
1. Take an **economic view** with short lead times and routines by defining a strategy for delivery of incremental value, setting up a framework to define risk trade-offs, Cost of Delay (Cod), and operational and development costs, while simultaneously facilitating decentralized decision-making.
2. **Apply systems thinking** to understand the many interfaces and connected components of both the organization and the system to be developed.
3. **Assume variability** by **preserving options** until evaluation using empirical data states that the focus can be narrowed down.
4. **Use short integrated working cycles** to develop **fast** and in an **incremental** manner to get fast customer feedback and mitigate risks.
5. **Frequently evaluate** working systems **objectively** and use this to base **milestones** on.

6. Ensure a continuous **flow** by **managing que lengths**, **visualizing and limiting Work in Progress**, and **reducing batch sizes**.
7. Use **cadence** to create predictability and **synchronize** through **cross-domain planning** to create a wide and common understanding in the light of uncertainty.
8. Facilitate the thriving of the **intrinsic motivation of team members**.
9. **Decentralize** the **decision-making** to reduce delays, increase flow, fasten feedback and easy the way for innovation.

Table A1.7 The nine principles of SAFe

There are four configurations for SAFe: Essential SAFe, Portfolio SAFe, Large Solution SAFe and Full Safe. Essential SAFe focuses on the team and program level and is incorporated in all other configurations while Full Safe also embodies Portfolio and Large Solution SAFe. Portfolio differs from Large Solution by focusing more on aligning portfolio execution to an organization's strategy, while the latter tackles the most complex issues through programs where portfolio alignment is not required but a solution to a specific problem stands central. The Large Solution SAFe is the most common one used for governmental scaled agility and will thus be used within this paragraph. First, however, the Essential SAFe, the core of every SAFe configuration, is discussed as the basis.

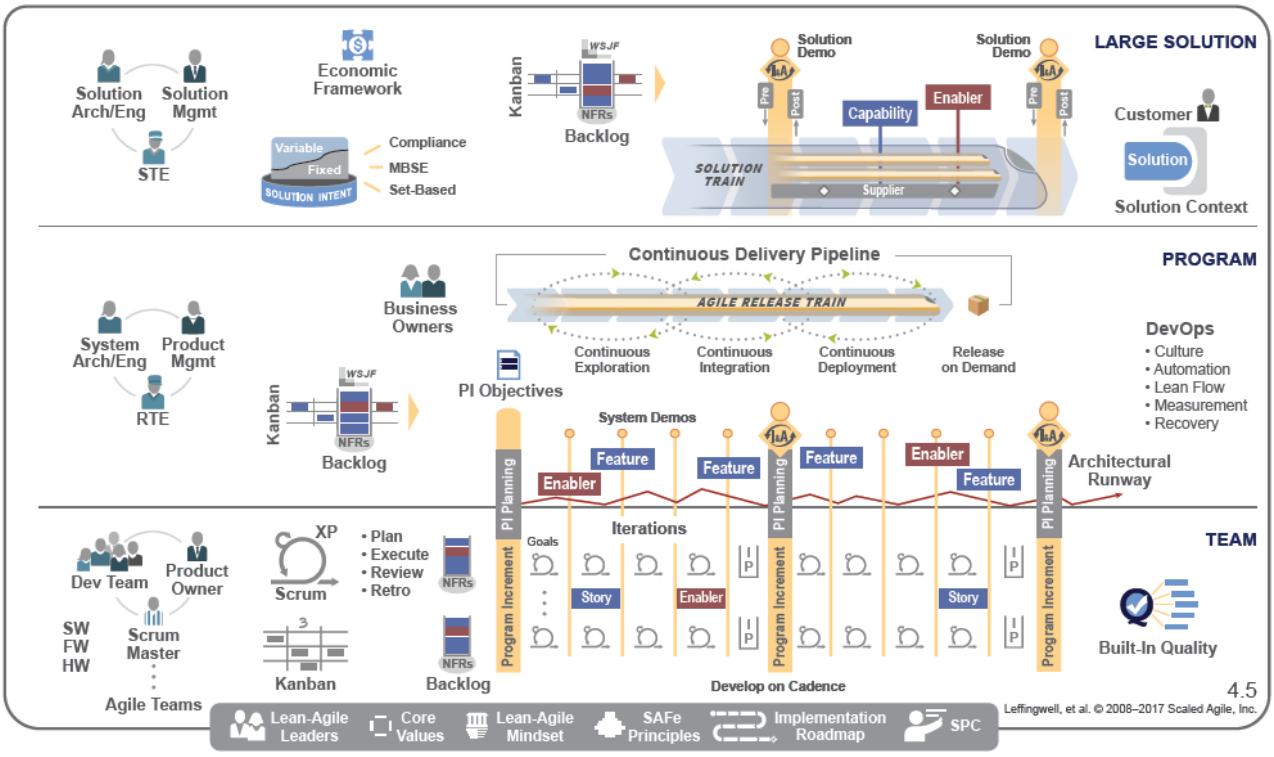


Figure A1.7 Scaled Agile (2017, p. 19)

On the team level, SAFe works as a combination of Scrum and Kanban with no further specific alterations and will therefore not be discussed in further detail. First the implications of using Safe on a program level with Essential SAFe is considered, and then Large Solutions SAFe is explored.

Essential SAFe roles and rules

Agile Release Train (ART)

The Scrum Teams and program levels together with important stakeholders form the ART, which can be considered as an organizational structure. Within the ART, a product increment is developed, while functioning as a Continuous Delivery Pipeline.

Systems Architect/Engineer

Either an individual or a team, the Systems Architect/Engineer designs the overall architecture, including the subsystems, major elements and required interface collaboration, for the system along with its Nonfunctional Requirements (NFR's).

Release Train Engineer (RTE)

The RTE serves as a Scrum Master for the ART and aims to improve its functioning through the Program Kanban and formal events.

Business Owners

A small group of stakeholders should be involved who have the primary responsibility to add technical and business-related aspects to the ART and therefore actively participate in ART events.

Customer

SAFe considers the customer as the decider of value and are therefore an integral part of the process and value-stream.

Essential SAFe events and activities

Product Increment Planning (PI Planning)

Considered the "heartbeat of the ART" (p.15), the PI Planning is face-to-face planning on a cadence basis to align teams to the common mission at hand.

Inspect & Adapt

This event, in which stakeholders and teams convene, serves to demonstrate and evaluate the current state of the solution. The outcomes are used to improve the backlog items with regards to what is needed to increase quality, velocity or reliability of the PI to come.

Innovation and Planning iteration

This event serves to estimate what is still required to meet the PI objectives and takes place every PI, and allows for learning, planning and Inspect & Adapt events.

Essential SAFe artifacts and concepts

System Demo

The System Demo can be used to measure the progress of an ART through empirical evidence retrieved from a working solution. Every two weeks, the total work delivered in an ART for that specific iteration, is shared with stakeholders to retrieve feedback and adapt its course when necessary.

Cadence and synchronization

Cadence is desired in SAFe as it presents a rhythm and therefore provides consistency in routine in the development process. To ensure that the multiple perspectives between teams and system-related aspects can be interpreted and acted upon at the same time synchronization is important. Together this enhances the effectiveness of operating in the uncertainty of product development.

DevOps

DevOps is a framework which serves to link the run and change activities in an organization along with relevant aspects. A more detailed overview of DevOps is presented in 2.2.3.

Lean-Agile Leadership

Leadership stands at the core of SAFe to ensure the agile lean way of thinking is embedded in the actors involved and the rules are adhered to. Furthermore, it should serve to motivate people. It also serves to take responsibility in the implementation process, since an absence of it increases the likelihood of failure.

Large Solution SAFe roles and rules

Solution Train

Similar to the ART, the Solution Train is an organizational structure on the Large Solution level which integrates the teams and stakeholders regarding a common vision, mission and backlog.

Solution Architect/Engineer

Either an individual or small team who defines the technical and architectural vision for the developed solution.

Solution Management

A group which collaborates with customers to define requirements and creates a solution vision to guide work through the Solution Kanban.

Solution Train Engineer (STE)

The STE serves as a type of Scrum Master on the Large Solution level who guides the ART's and suppliers work.

Supplier

Internal or external organizations which develop and deliver certain aspects that are part of the Solution Train.

Large Solution SAFe events and activities

Pre- and Post-PI Planning

Two events which take place before and after the PI Planning to coordinate between individual ARTs and suppliers within one Solution Train.

Inspect & Adapt (I&A)

An important event in which the current state of the solution is demonstrated and evaluated by representatives of multiple ARTs and suppliers. It serves to reflect and identify improvement backlog items.

Large Solution SAFe artifacts and concepts

Economic Framework

A framework which sets the financial constraints for the Solution Train.

Solution Demo

Consisting of the results of all developments made by ARTs and suppliers, this demo is evaluated and shared with customers and other stakeholders.

Control mechanisms

Control mechanism	*Description*
Consider the agile process as different structures per level	While interaction between the project, program and organization (solution) level is included, there is still a clear distinction between the three levels as each has its own objectives, events and roles.
Retrieving information from stakeholders through engagement in different levels and sharing of data (system demo)	To ensure the developed product meets its requirements, interaction with stakeholders is crucial. It can be considered a form of control to ensure the product takes shape in its desired form. In SAFe the engagement with stakeholders takes place on multiple levels. They are part of the organizational structure which integrates a common vision and mission (Solution Train), but they are also included on the project level as business owners to add technical and business-related requirements. A Solution Demo is shared with stakeholders every two weeks to retrieve feedback and create new requirements.
Formal events for planning, inspection and adaptation, and evaluation on multiple levels	The formal events are crucial to ensure the proceedings of the scrum teams on one and between multiple coincide with each other's activities. The set structure follows the principle to work simple and decrease complexity. By having these events take place on multiple levels integration is eased.
Coordination of teams through collaboration and synchronization	Frequent interaction and collaboration within teams and between teams through formal events helps harmonize the product development and secure integration.
Clear and agreed upon rules and roles on both team, program and organizational level	SAFe entails clear roles and rules to ensure a transparent and process which allows for high quality of inspection and paves way for targeted adaptation. Roles between those designing the infrastructure (similar to Scrum Master) and those involved on the content are differentiated, and roles focus on one level at a time. Next to the internal teams, there should be a team in charge for the customer engagement and separate roles should be identified for external suppliers.

Setting financial constraints on the higher level	SAFe uses an economic framework to set constraints to the Solution Train which takes place on the highest level and will trickle down to the specific functioning on a project level.
Short iteration cycles	Short iteration cycles reduce complexity and increase the control on the direction in which development takes place.
Consistent inspection, planning and adaptation	There is continuous tracking of the progress made to adapt the planning and implement desired adaptation.
Alignment through a common solution vision and mission	The shared solution vision and mission provide guidance provides guidance for all parties involved in the development process and increases alignment.
Agile Lean Leadership	Specific leadership skills are required to motivate people in agile projects, and especially those applied on a higher scale, while also assisting in promoting the principle of accountability.

Table A1.8 Control mechanisms SAFe

1.3 Agile project and portfolio management

To provide insight into how management approaches consider agility and required control, the following paragraph discusses three of its aspects: when agility should (not) be choses as a management approach, agile project management, and agile portfolio management. Agile portfolio management is included since agile projects are rarely implemented in an organization or context where there are no other projects. Therefore, attention should also be given to the portfolio perspective and the implications a portfolio has on a project and vice versa.

1.3.1 When to apply agile project management

Agility provides a rather different approach for businesses and organizations to execute their development or operational processes, and as already has become clear, brings its own difficulties which require specific control mechanisms. In general, just the choice in itself to go for an agile approach can be considered a control mechanism in itself, or at least bring insight into the potential success or fail factors at play. Chin (2004) provides insight into when agile project management can be chosen. While reading the paragraph, one needs to remember these are guiding principles, as these conditions do not by themselves guarantee success or failure.

Principles

Chin's approach focuses on the importance of types of environments in which the project takes place and the type and level of uncertainty associated with these environments. By distinguishing two criteria, being the project environment and the organizational environment, the approach aims to shed light into the potential impact of these environments on the agile functioning.

Project environments

The first type of environment is the project environment, which can be divided into three main categories: the operational project environment, the product/process development environment, and the technology development environment. It therefore differs from DevOps, which only differentiates between the operational and development processes.

The Operational Project Environment

Chin characterizes operational projects as those "that are run with a regular frequency, are very similar to each other, and are critical to the day-to-day running of the business" (14) such as service provision. As the processes for these projects are similar, there is a low level of

uncertainty as all processes are rather regular. Therefore, more traditional project management methods are more applicable for these types of projects.

The Technology Development Project Environment

The second type of project environment focuses on creating breakthrough technology platforms. Technology platforms as such are not a single product or application but rather the underlying technology which provides a base for such developments. These projects are unique and require great creativity. This internal uncertainty should be supported by the project management environment. Therefore, these types of projects require agile project management approaches.

The Product/Process Development Project Environment

The third category is placed between the two previously mentioned ones as it includes "less front-end science/engineering expertise and more business acumen" (16). While still requiring creativity from the team, this needs to be balanced with business objectives. Furthermore, there is a lot of external uncertainty. Next to the uncertainty associated with the field of science or the specific technology, there are uncertainties in the business and market domain. Together this brings good opportunities for the use of agile project management in these types of projects.

Organizational stakeholders

While the first criteria focused on the project specifics, the second one focuses on the organizational context by considering the stakeholders involved. Distinctions can be made between projects unfolding in a single organization, multiple organizations, and a single company with multiple organizations.

Single Organization

Technology development projects that are still in their early days generally do not per se have an external customer and the only actor really involved is the business itself. In general, projects performed under a "single organizational umbrella" (18), meaning without further subcontractors or partners, generally have common objectives which provides an easier environment for agility to thrive in.

Multiple Organizations

When a project cuts across multiple organizations it becomes more difficult to create common expectations and vision, for which classic project management approaches might be more suitable. While it is deemed more likely that projects which last a longer time allow for the creation of an agile culture, this would not be the case for shorter projects. Furthermore, money talks. Agility's uncertainty in path trajectory can bring unexpected changes to the parts to be played by each organization, which can be considered undesired or even not possible given contractual obligations. While projects with multiple organizations bring substantial challenges, agile project management can still be applied. Not only a timeframe of multiple years, but also the presence of a "strong, prime contractor that can drive organizational change across subcontractors" (19) can make the use of agility not just possible but even fruitful.

Single Company, Multiple Autonomous Entities

This third category again provides an in-between situation where there is a single organizational umbrella, but the different divisions or silos act as rather autonomous entities. In these situations the success of agility is more dependent on the motivation of a division's leader to introduce and use the concepts.

	Multiple, External Stakeholders	Multiple, Internal Stakeholders	Single Organization
Operational Projects	Classic	Classic	Classic
Product/Process Development Projects	Classic/Agile	Classic/Agile	Agile
Technology/Platform Development Projects	Classic/Agile	Agile	Agile

Figure A1.8 The types of project environments and suitability for classic or agile management approaches

Control mechanisms

Control mechanism	*Description*
Use agile PM when there is uncertainty in the project environment	The use of agility should fit the project environment and is mainly suitable for situations that embody internal and/or external uncertainty. This relates both to scientific uncertainty, but also market and business uncertainties at play.
Use agile PM in projects when the organizational set-up allows for an agile culture	The creation of an agile culture takes time and requires common objectives. These aspects are more easily met in single-organizational projects but also can be achieved in multi-organizational projects. In these cases, projects should last multiple years and include a strong prime contractor which can promote organizational change throughout subcontractors.

Table A1.9 Control mechanisms conditions agile project management

1.3.2 Agile project management

Agile project management provides a new perspective on how managers should position themselves and act in agile projects. It focuses on both social and technical concepts like team dynamics, modern leadership and soft skills, collaboration tools and new ways of measurement. For this research, Crowder and Friess (2015) are used since they provide a combination of these social and technical characteristics of agile project management. It focuses on choosing the right team, facilitating the team, and trusting the team, as performing agility without guidance would most probably entail project failure.

Principles

While the approach generally embraces the agile principles, it puts further emphasis on the need for trust along with control and empowerment. The former is conceptualized as 'Locus of Control' which "refers to the extent to which individuals believe that they can control events that affect them" (3), with a distinction between internal control, the extent they control events themselves, and external, where control is performed by another team member, the manager or even the customer. The 'Locus of Empowerment' is a process which involves a team member's ability and perception of being accepted, belonging, and capable to function both individually and in a team. Together, the Locus of Control and Locus of Empowerment help to create independence and interdependence required to make agile team members succeed in agility.

Skills of an agile manager

Crowder and Friess mention four skills an agile project manager should have to ensure agility is performed efficiently and purposefully. While these skills might not be only attributable for agile project management, but project management in general, these are still included as their importance is even more important when agility is applied given its greater emphasis on change and embracing of uncertainty.

Effective communication: This is required to increase transparency, build trust, and be adaptive to cultural differences. Therefore, they should communicate their values and expectations clearly and professionally.

Diplomatic: Consideration of the implications of the project on the stakeholders and adequate follow-up is also required for trust and transparency.

Effective listening: The manager should be a good listener and not interrupt when a team member is expressing his/her views to gain a complete image.

Analytic thinking: To distinguish small and big issues and understand the problems at hand from a multi-disciplinary approach, the manager should be a strong analytical thinker.

Concepts to be defined by the agile manager

Early in the process there should be a common vision of certain concepts to ensure team members are have the same understanding of how the development process will be performed. It is the job of the agile manager that these concepts are defined early on and adhered to during the process.

Definition and creation of independence: Independence allows for the autonomous functioning of team members and self-organization of and within teams which requires trust. A shared understanding of what independence entails in a project can facilitate this process.

Definition and creation of interdependence: To prevent independence from leading to a lack of team work and to ensure that the team members are committed to a common vision and the stakeholders, interdependence is also required.

Creation of overall, team, and individual goals and objectives: By identifying these goals at an early stage helps to compose teams in an efficient manner and secure motivation and development of team members, while the overall goals are not lost in the process.

Creation of self-organization concepts: Teams do not automatically self-organize efficiently and therefore need guidance such as open communication, giving examples of desired behavior, and being transparent.

Creation of feedback and collaboration timelines and objectives: There should be clear points where customer feedback and reviews are planned to ensure collaboration. This also enhances the potential for trust building.

Creation of stable sprint team membership: Another interesting remark by Crowder and Friess is that agile teams should be kept rather stable in membership since continuous shuffling takes a lot of time to redefine the mentioned concepts and build trust.

Creation of a suitable sprint time which is kept to: While there is no set time for sprints, they should not be too long or short. When they are too long it becomes difficult to test the implications of the developed increment, the evaluation becomes problematic since a long time passed after events of the beginning of the sprint, and stakeholders will be flooded with new information which makes efficient feedback more difficult. Still, typically a sprint lasts 30-45 days and the time set in the beginning should be kept stable throughout the projects.

Creation of the ability for teams to challenge and question the sprints: The monitoring and learning character of agility should also be used to enhance the sprints. The manager should therefore listen to the team and let them challenge the setting. This leads to a better understanding and commitment.

Creation of an environment for mentoring, learning and creativity: The different experience levels present in teams provide good learning and development options for its members. This process should be facilitated through the sprint schedules and a positive and transparent atmosphere.

Prevention or elimination of dysfunctional elements of self-organization

Agile teams generally face five challenging dysfunctions which can arise in the light of self-organization. Crowder and Friess provide recommendations on how managers should cope with these.

Eliminating the absence of trust: Through taking the lead and stepping forward to do whatever is required to get the project moving, managers can enhance the trust team members have in the team and the project.

Overcoming the fear of conflict: Self-organization comes with interpersonal feedback, which especially with more introvert team members, can be toned down or even silenced in the fear of conflict. Managers should be observant of the team members and can sometimes even help team members to overcome this fear. An environment of trust also helps with this phenomenon.

Creating and sustaining commitment: When a team member delivers low quality of work or is not willing to actively participate the manager should try to understand the reasons for this and help the team member overcoming this hurdle. Even though it sounds rather harsh, Crowder and Friess state that if the manager's engagement still does not help, the team member can better be removed from the team.

Keeping an eye on results: While some developers might only keep an eye on their own work, it is important everyone focuses on the results of the team and the overall product to be developed. The overall vision should therefore be promoted by the manager to ensure everyone is working for a common goal.

Design and management of teams

Crowder and Friess also provide specific recommendations for the team culture and setting they work in to enhance efficiency through diversity and allow for independence and interdependence.

Create both a team room and space for privacy: Whereas a team room is necessary to facilitate and stimulate interaction, it is also important that team members can withdraw to a private space when they have the need to.

Foster interaction in long-distance teams: When agility is performed over a larger geographical area, the day-to-day interaction should still be performed, which can be done through video

conference, Skype or even phone calls. It is desired to have the team convene at the same location at the beginning of the project for 2-6 weeks and have them reassembled again later on to enhance the team spirit.

Create teams which consider diversity and inclusiveness as valuable: Diversity is desired to enhance team creativity and ensure all required skills are present. The teams themselves, should also be aware that this diversity is present and desired, and that inclusiveness benefits this process. To a certain extent, Crowder and Friess state this as a self-fulfilling prophecy, where diverse teams create good results and that recognition of these results leads to awareness on the importance of diversity and inclusiveness. The facilitation of openness and trust is also important for this process, and brainstorming can be used to make team members value diversity (Crowder and Friess use the example of inter-generational diversity to illustrate this).

Technical design and tools to be used

Agile project management also comes with specific recommendations with a technical characteristic which should be used, next to the more social team-oriented recommendations.

Have a systems and software architecture in place early on: If the overall systems and software architecture is not in place at an early stage testing becomes problematic and it can even lead to the need for a lot of recoding later on in the project, or even a completely new system.

Use productivity tools that facilitate agility: Certain productivity tools, like Microsoft Excel or Microsoft Project, do not meet the requirements of agility when it comes to fostering collaboration or creativity. The tools chosen should be people-centric and should focus on fostering an environment of innovation, collaboration, efficiency and automation.

Use a Collaboration Framework for the tools: To integrate the findings of all the tools used, a Collaboration Framework with an Electronic Engineering Notebook is useful for efficient oversight since it “automatically captures, logs, correlates, and publishes all aspects of the program/project” (47). The analyses can then be based on the Collaboration Framework, rather than each tool separately.

Replace traditional EVMS with agile EVMS: As there is a need to understand in agility that even though one can complete a project on budget and schedule, the project can still fail because it does not respond to the (changing) customer’s needs. An agile Earned Value Management

System should take the entire development life cycle into account, not just the software assessment, but also the process, sprints, and assessment tools. The core of agile EVMS is that is that complexity and uncertainty are represented in the formulas through entropy which indicates that uncertainty comes with higher volatility. In agility, however, there is not just uncertainty over changing requirements, but even uncertainty over the increase of entropy during the process due to the uncertainty in factors which drive the agile teams' functioning, which can be regarded as double uncertainty. According to Crowder and Freiss there should therefore at least be a focus on uncertainty of team stability and software defect volatility.

The budget should be centered on project values and objectives: Budget should not be centered on time and motion (cost and schedule in the traditional EVMS approach), as in traditional project management approaches, which would use classical EVMS, but on "perceived values and the objectives of the project" (46) as these stand at the core of the project's adaptive development.

Control mechanisms

Control mechanism	*Description*
The manager has agile skills	To ensure agility is facilitated and the staff is motivated, an agile manager should have effective communication skills, be diplomatic, be an effective listener, and think analytically.
Clearly define concepts such as independence, interdependence, and self-organization	Shared understanding of these concepts is required for harmonization and a common vision on what the desired behavior is and aligns congruency in ways of working.
Create feedback and collaboration opportunities and objectives	Not only should the schedule include moments for collaboration and feedback, there should even be objectives on what should be the goal of this interaction to facilitate fruitful interaction.
Use a stable sprint	Stability in the sprint time is needed to decrease complexity and stream line the process. Typically, it is 30-45 days, as it shouldn't be too long. This would add complexity and hamper the evaluative potential.

Keep team membership stable	Even though agility is adaptive, it is important to try to keep the members of team relatively stable to increase trust, familiarity and collaboration.
Create an inclusive environment with space for diversity, learning, mentoring, critical reflection, and creativity.	Diversity in teams is required in agility to enhance creativity and have a wider skillset at hand. Yet, diversity should also be appreciated and used for further learning. Mentoring is therefore also important, for instance with regards to inter-generational diversity. Inclusiveness is overall important to ensure that critical reflection and interaction takes place in an environment of trust.
Prevent and eliminate dysfunctional elements of self-organization	An agile manager should pay attention to and try to prevent and eliminate the absence of trust, fear of conflict, lack of commitment and absence of focus on the overall results. If these are not cared for, the concept of self-organization can provide negative results since feedback and creativity will fall.
Have separate spaces for teamwork and privacy	While face-to-face interaction is crucial for agility, it is also important that team members can withdraw to have some privacy for their thinking process or relaxation.
Foster interaction in long-distance teams	If agile teams are geographically distributed, digital media tools can be used to facilitate interaction. Furthermore, the group can convene at the beginning of the project (2-6 weeks) and reassemble at a later stage to enhance the team spirit and collaboration.
Have a systems and software architecture in place early on	To make sure that testing is done effectively and prevent late recoding or even complete system failure, a systems and software architecture should be in place early on.
Use productivity tools which support agility	Productivity tools which are people-centric and focus on fostering an environment of innovation, collaboration and automation, should be used to support collaboration and creativity in agility.
Use a Collaboration Framework for integrating tools	To integrate all the tools used in agility, a Collaboration Framework can be used with an Electronic Engineering Notebook, to create efficient oversight and measurement into the what is happening at the agile project.

Use agile EVMS instead of classical EVMS.	The Earned Value Management System should be based on the entire development life cycle of agility and its different characteristics (product, sprint, assessment tools) to gain an understanding of how the project is unfolding. Uncertainty is represented in the agile EVMS and there is a specific focus on the stability of teams and software defect volatility.
The budget should be centered on project values and objectives	Instead of having a budget focused on time and motion (schedule and costs), as traditional project management requires, the budget should be based on the perceived objectives and values which stand at the core of the project.

Table A1.10 Control mechanisms agile project management

1.3.3 Agile portfolio management

Agile projects rarely stand alone as an organization often uses multiple ones which are integrated with regards to a system component to be developed. The resources made available to a project depend on the overall resources available and portfolio prioritization, which is prone to change given competition and other changes in the environment (Chin, 2004). A change in allocation of resources can influence the trajectory of a project and therefore agile portfolio management is worth considering in the light of control. This paragraph draws mainly on Chin's chapter on Agile Portfolio Management (2004), which focuses mainly on the strategic dimension, and a Stettina and Hörz (2015) article, which combines the main academic articles on agile portfolio management with empirical findings.

Principles

Agile portfolio management is about aligning project and program performance with business and strategic interests to meet an organization's objectives. Or, in the words of Chin, portfolio management should be considered as "the projects are the business" (p.183). It furthermore tailors to the agile approach as it aims to provide scaling of agile principles.

Focus points

Use one portfolio: By using one portfolio for the entire organization, complexity is reduced and furthermore enhances the transparency regarding the allocation of resources over the projects. Projects are furthermore connected due to technical interfaces, which should all be capture and understood (Chin), and sometimes projects which are not connected content-wise still make use of the same resources. Together this asks for a one-portfolio approach (Stettina & Hörz).

Prioritization according to business: Agility as an adaptive approach plays into the changing customer needs and environment. Prioritization should therefore be done on the basis of strategic interests (Chin). To increase the transparency and facilitate a common understanding strategic backlogs can be used for this (Stettina & Hörz).

Use both top-down and bottom-up input for prioritization: Given the general bottom-up nature of agility and the closeness of projects to feedback from its environment, it is important to use this knowledge present at the lower levels (Chin). This counts both for the external influence and resource allocation.

Place portfolio management in the program manager's hands: In line with the previous finding that decisions on the portfolio should include more bottom-up information, the shift of responsibility from executive management to program management facilitates this process. This will also increase reaction speed as it takes less time, and again reinforces the notion that projects are the business (Chin).

Make estimations by using both top-down and bottom-up estimations: To gain a good understanding of the resources used in different projects, it serves well to make both top-down and bottom-up estimations. These should not be too detailed given the changing nature of agile projects as this would else create 'unnecessary' spending of time. When the two estimations provide a mis-match it is worth going into detail to figure out more exact numbers (Chin).

Perform continuous/frequent portfolio reviews and routinized interaction: In contrast to traditional project management, which might use annually portfolio reviews, agile projects require continuous care for the portfolio given the embracement of change. Requirements and business objectives can change and might require fast re-allocation of resources (Chin; Stettina & Hörz).

Senior management should commit to strategically managed portfolios: The agile approach entails a shift of responsibility where decision-making power is likely taken out of the hands of senior management. To ensure the agile way is successful, higher levels of management should sponsor and motivate the agile adoption, awareness and communication (Stettina & Hörz).

Stable teams are still preferred: Even though the strategic interests can create a redistribution of resources over the different projects, agile teams still prefer stability where membership is kept rather consistent over time (Stettina & Hörz).

Control mechanisms

Control mechanism	*Description*
One portfolio should be used	To ensure project functioning and allocation of resources is in line with strategic interests and project interdependencies are fully understood, one portfolio should be used.
Review the portfolio continuously	To play into the changing environment and facilitate the adaptive nature of agility, the portfolio should be reviewed continuously and not just on an annual basis.
Prioritization according to business with both top-down and bottom-up information	Given the changes that occur it is important to prioritize according to the business objectives of the organization, while both top-down and bottom-up information is used. A strategic backlog can be used to make the process transparent.
Shift portfolio responsibility to mid-level	To increase the reaction speed and support the premise that the projects form the business, the responsibility for the portfolio should be shifted to the program manager. This requires support from higher level staff.
Make top-down and bottom-up abstract estimations	Given the changing nature of agile projects, abstract estimations are sufficient, but they should be performed both bottom-up and top-down. In case there is a mis-match between the two, more detailed estimations can be made.
Try to keep team membership stable	Even though strategic interests might steer for a reallocation of resources, it is important to try to keep the membership of teams stable to secure the teams work well in an environment of trust.

Table A1.11 Control mechanisms agile portfolio management

1.4 Adaptive and agile governance

This paragraph sheds light on the lessons which can be learned from governance approaches. Agile governance and agile project governance are used to provide specifics for agility, but also the paragraph also dives into adaptive governance, which can be considered as a broader term for agile governance which touches upon some of its characteristics, along with polycentric and organic governance, but then in broader terms (Wang, Medaglia, & Zheng, 2018). Digital Government can then again be considered as a broader concept than adaptive governance (Soe & Drechsler, 2018) but it has not been included in this research since its relation to agility is already of a second order, while the most relevant aspects are included through adaptive governance already.

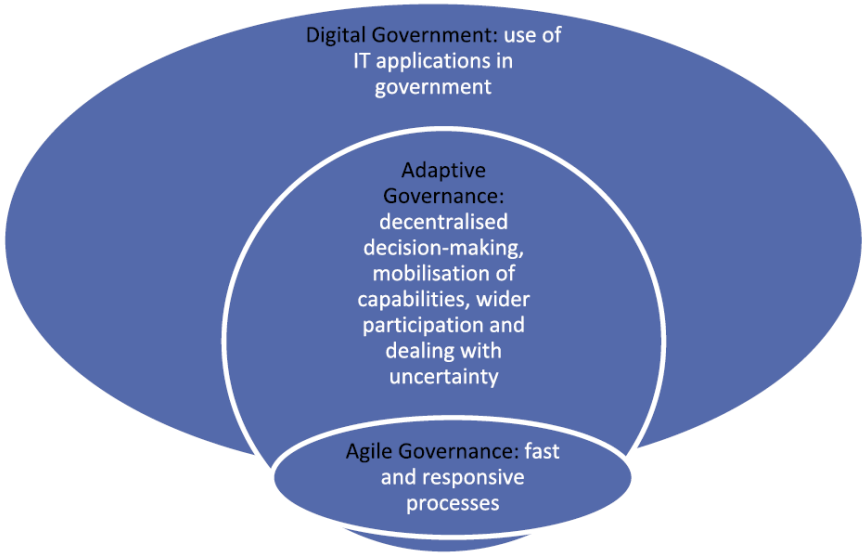


Figure A1.9 Relation of concepts from Soe and Drechsler (p.324)

1.4.1 Adaptive governance

Adaptive governance recognizes the need for organizations to enhance their capacity to adapt to changes while at the same time one should be wary this does not lead to instability. Therefore, organizations should act in a balancing act as ambidextrous organizations: "organizations that adapt conflicting values within them and, accepting and addressing the tensions among those values" (Janssen & Van Der Voort, 2016, p. 4). The tension derives mainly from "a problem of rhythm" (ibid) between the speed of technology and other types of governance such as network governance, which calls for vertical governance which connects these different horizontal layers. Overall, adaptive governance provides an umbrella definition for adaptive methods,

such as agility. For this research, the article by Janssen and Van der Voort (2016) is used as it provides a rather holistic summary of adaptive governance, including its strategies.

Principles

Adaptive governance accepts the presence of uncertainty and therefore embraces learning as its core value for complex societal issues with a lot of stakeholders (Gunderson & Holling, 2002). Within this process it aims to prevent instability which can derive from three sources. Firstly, tension arises from general fast responses regarding technology and other fields which require adaptive change, and stability, accountability and transparency as values of the public sector. Secondly, there is a tension between the different horizontal levels of governance (see figure A1.10), and finally, the mere introduction of technological development and decentralization can bring about changes in existing power structures.

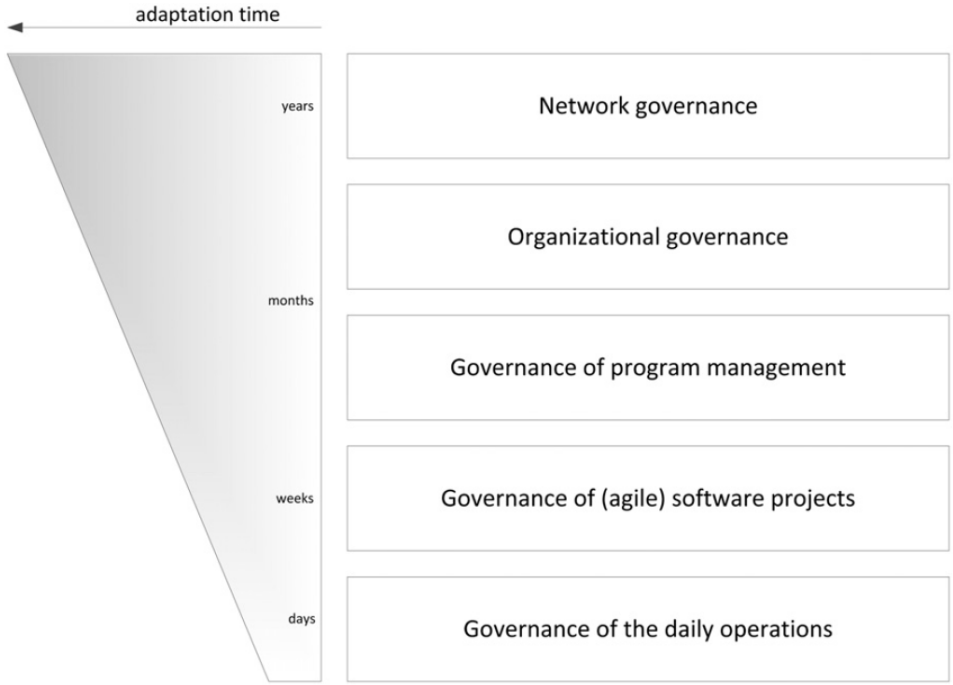


Figure A1.10 Overview of the different levels of governance at play for an organization with their respective timeframes (Janssen & Van der Voort, p.2).

Characteristics

By acknowledging the tensions at play and providing strategies to sooth these tensions and prevent instability, adaptive governance aims to facilitate the use of adaptive methods by matured organizations. Generally speaking, the strategies can be categorized in the following clusters.

Decentralized decision-making: To enhance the adaptiveness of an organization, more decision-making authority is shifted to lower levels where there can even be self-organization to act in the desired speed for adaptiveness.

Mobilizing internal and external capabilities: Adaptive governance states that the unexplored potential of decentralized tacit knowledge should be mobilized throughout the organization. Furthermore, external stakeholder engagement needs to be performed with more care in adaptive environments given the different interests and uncertainty at play.

Continuous adjustment and balancing of values and approaches: Especially the more mature organizations with a history of stability, and bureaucracy, need to act as ambidextrous organizations and balance the use of control and 'free learning' to prevent that the adaptiveness leads to instability. This prevents an organization to fall into either the 'competency trap', when too much emphasis is put on current competences, or the 'failure trap' in which the failure which explorative development sets the general tone in an organization. Continuous adjustment and balancing is therefore required.

Janssen and Van der Voort name multiple strategies which can be used to enhance each of these processes. These can be grouped according to these three overall strategies.

Decentralized decision-making	**Continuous adjustment and balancing of values**
Decompose complex problems into smaller components, such as in agility (Sutherland, van Solingen, & Rustenburg, 2011).	Keep options open which cannot be changed easily later on in the process to keep potential directions open (Dym, Little, Orwin, & Spjut, 2009).
Shorten decision-making times on a lower level to quickly respond to innovations.	Use and produce flexible and adaptive infrastructure (Janssen, Chun, & Gil-Garcia, 2009).
Mobilizing external capabilities	**Mobilizing internal capabilities**
Know the stakeholders' interests, objectives, behaviour and their capabilities to know how these can be used for the project (De Bruijn & Ten Heuvelhof, 2008).	Promote and use self-organization that draws on various types of knowledge present in the organization (Folke, Hahn, Olsson, & Norberg, 2005).

Through the knowledge gained, mobilize the stakeholders through cooperation and selection of partners (ibid).	Ensure that high-level decisions are based on information retrieved at the lower levels which transcends bottom-up.
Make use of both private and public partners to achieve the aforementioned strategy, but be aware to know their interests and objectives (Klievink, Bharosa, & Tan, 2016; Klievink & Janssen, 2014).	Training and education are required to improve the capability of an organization's adaptiveness. This also holds for the higher levels of such organizations.
Make sure to confront stakeholders which delay or keep the cooperation process from progressing.	

Table A1.12 Categorization of control mechanisms adaptive governance

Control mechanisms

Taking these strategies in consideration, the following control mechanisms can be identified.

Control mechanism	*Description*
Decompose complex problems	While already incorporated in the agile approach, the decomposing of large complex issues into smaller increments makes the control over development easier.
Use short decision-making times on a decentralized level	To play into the innovative nature of adaptive projects, decision-making should be fast and take place on the lower levels of an organization.
Mobilize external capabilities	Stakeholders should be engaged in a productive manner. To achieve this, knowledge of these actors' interests, objectives, behavior and capabilities is required, to then mobilize them and if need be even confront them when the process stagnates. Both private and public sector actors should be considered. For more in-depth analysis see 2.4.
Use self-organization	To promote decentralized decision-making and allow for individual knowledge to transcend into the wider spectrum of the organization, self-organization can be applied. For more in-depth analysis see.
Facilitate bottom-up information	To ensure that high-level decision-making incorporates decentralized knowledge, there should be vertical information provision facilitation.

Educate and train	The employees of an organization should be trained to adapt to the new ways of working, even higher management and the board, to ensure that the principles of adaptive governance are understood and performed correctly.
Prevent path-dependency and use flexible infrastructure	Control is lost in adaptive governance when the possibility to be adaptive is ruled out. Therefore, path-dependency should be avoided when possible, along with choices of which alteration can be very costly at a later stage. Therefore, also the infrastructure used should be flexible.
Balance between adaptiveness and stability	For certain, especially larger and mature, organizations, adaptiveness without consideration for its further characteristics can lead to instability.

Table A1.13 Control mechanisms adaptive governance

1.4.2 Agile governance

Agile governance as a concept surfaced in academic literature from 2007 through Qumer (2007), and has since been further been explored theoretically. Luna et al (2014) developed a definition of agile governance to incorporate previous findings and shape the debate towards one shared definition as such examples were still lacking or unthoroughly explained:

"Agile governance is the ability of human societies to ***sense, adapt and respond rapidly*** *and* ***sustainably*** *to changes in its environment, by means of the coordinated combination of* ***agile and lean*** *capabilities with* ***governance capabilities****, in order* ***to deliver value faster, better, and cheaper*** *to their core business." (Luna et al, 2014, emphasis added).*

Whereas the definition does not provide further specifics on what is meant exactly with agility, the direct inclusion of lean and sustainability in the definition is noteworthy. Sustainability is linked to organizational sustainability and provides similarity to the focus on stability in Janssen and Van der Voort. Overall, agile governance is considered a steering method which is constructed to a framework by Luna et al (2015). While the framework rather serves as a set of lenses to be used to consider agile projects and cluster its characteristics, it also provides some control-oriented mechanisms.

Principles

The principles the governance framework abides to relate to those of the Agile Manifesto, as previously discussed, and those of lean, as previously discussed in.

Characteristics

The framework considers the analysis of agile projects to be divided into constructs and laws of interaction, boundaries, and system states. The framework approaches the organizational context, in accordance with Creswell (2017), as a control variable which affects the construct values but does not influence the overall consistency or causality of the system.

Constructs and laws of interaction: Through theoretical exploration, the framework provides six different constructs which can help explain the functioning of agility due to their characteristics, relations and interactions.

Construct	*Influence*
Effects of environmental factors	The influence of the external environment on the organizational context.
Effects of moderator factors	The influence of internal processes on the organizational context which can oppose or restrain the agile functioning of the organizations.
Agile capabilities	The capability to perform the project in accordance with the principles, values and practices of agile and lean approaches.
Governance capabilities	The manner in which the organizational context, and thus the parties involved and aims pursued, is influenced.
Business operations	The activities related to the daily functioning of the business to create value delivery.
Value delivery	The ability to produce results through value delivery which influences the business of the organization and its sustainability (current and future).

Table A1.14 Constructs of agile governance

Boundaries: To delineate the concept of organizational context, the framework mentions "only those units and laws of interaction that relate to the organizational contexts of the Agile Governance in IT teamwork perspective…: team, project, business unit, enterprise, or a multi organizational setting" (12). It therefore considers the constructs environmental factors and value delivery as the external environment, while the other four are considered as the internal environment.

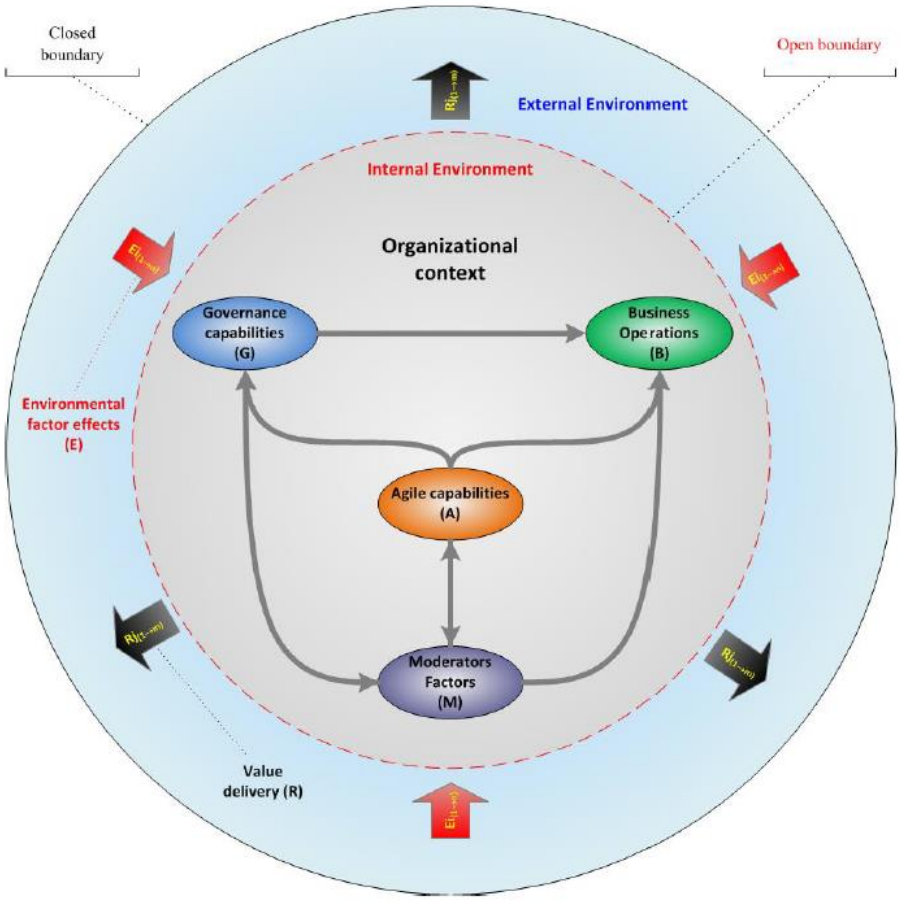


Figure A1.12 The organizational context of agile governance (Luna et al, 2015)

System states: To characterize the agile state of an organization, the framework provides macro-states and system states to categorize cases. As the framework does not provide insight into how these should be governed, but only names them, they are not further included in this research.

Meta-principles: Agile governance already provides six recommendations for improving agile governance in the form of meta-principles.

1. "**Good enough governance**: "The level of governance must always be adapted according to the organizational context".

2. **Business-driven:** "The business must be the reason for every decision and action".

3. **Human focused**: "People must feel valued and incentivized to participate creatively".

4. **Based on quick wins**: "The quick wins have to be celebrated and used to get more impulse and results".

5. **Systematic and Adaptive approach:** "Teams must develop the intrinsic ability to systematically handle change".

6. **Simple design and continuous refinement**: "Teams must deliver fast, and must be always improving."

Control mechanisms

Control mechanism	*Description*
Adapt control mechanisms to organizational context	Control mechanisms should be chosen and adapted to the six constructs of the organizational context to enhance it.
Be both business and agile driven	By combining agility's value delivery with business incentives, the business side of the organization is incorporated and will not block the agile functioning.
Ensure participants feel appreciated, motivated, and encouraged to be creative	A climate should be created and sustained where participating staff and stakeholders feel appreciated, motivated and encouraged not only to influence the efficiency of agility but even to prevent internal and external opposition.
Deliver fast and celebrate	While the process should be continuous with fast delivery, the steps made should also be celebrated to provide extra motivation.
Ensure the adaptiveness to be systemized	Over time the teams should not intrinsically apply the adaptive functioning of agility to be able to work fast and prevent internal blocking of the agile process.

Table A1.15 Control mechanisms agile governance

1.4.3 Agile project governance

Whereas the agile governance of 2.4.2 focuses on the organizational context, agile project governance, as can be deduced from its name, takes a closer look at the project level. The approach generally recognizes the multi-actor implications and the tension derived between traditional conventions and new agile ways of working in the public sector. Lappi and Aaltonen's (2017) article provides the main piece of academic study and is therefore used for this analysis. The article provides a combination of literature research and case study of three agile projects in the Finnish public sector.

Principles

The approach revolves around the principles of agility as discussed in chapter 1.

The six dimensions of agile project governance

Lappi and Aaltonen provide six dimensions with their own characteristics which influence the functioning of agile projects in the public sector: business case, contracting, controlling, steering, decision-making, and capability and competence building.

Business case: The business case of an agile project should consider the "stakeholders, goals, budgets and specifications" (272). This dimension is rather crucial for the unfolding of the agile project, yet there are general issues with regards to public sector projects. Firstly, it is unclear how stakeholders should be involved in a systematic manner and the identification of partners, even the customer, can be tricky. Furthermore, pre-project fixed budgeting provided negative constraints on the agile functioning, especially when stated before the specifications are identified.

Contracting: The contracting dimension involves the "sourcing, procurement process, risks and incentives" (ibid) used in or for the agile project. Difficulties for public sector projects derive from the restrictions in public procurement law. The sourcing models available to use for agility, given the option for more flexibility, generally create a situation in which the risks of the project are not shared, but are mainly carried by the PMO, who in turn mainly uses traditional methods to perform risk management, which then hamper the agile functioning.

Controlling: With regards to controlling, the approach considers "monitoring, reporting, and measuring" (ibid). Use of traditional measurement of result, which generally focus on project management metrics of costs, quality and budget, are not per se unsuitable, but if only these are used there is basically no measurement on the agile functioning of the project. Reviews should be performed systematically, and resource availability, though challenging, should be performed and testing automation also provides a well-suited mechanism for agile projects.

Steering: Steering is defined by Lappi and Aaltonen as "coordination, planning and communication" (ibid). Also in this dimension there are quite some difficulties for public sector agencies. Higher levels of an organization tend to keep using traditional strategic steering and communication which disintegrates the interlevel communication and functioning of agility. This is showcased in the lack of a common language to be used in all different levels of the organization.

Decision-making authority: This dimension, which is also a rather problematic one, focuses on authority, roles and organizing. In practice, the decision-making authority is not always as clear when it comes to external to internal relations. A common trait found by Lappi and Aaltonen was that there were multiple product owners and engagers with stakeholders, which only added to the complexity, rather than making the system simpler. Furthermore, care should be given to the tensions that derive between the technical and business-related aspects of the project and organization.

Capability and competence building: The final dimension of agile project governance focuses on the training and education of staff at all levels of the organization to understand and apply the agile principles.

Control mechanisms

Control mechanism	*Description*
Use a systematic method to identify and approach stakeholders	The stakeholder involvement for agile projects is currently performed randomly in the absence of a systematic process. Such an approach can make it easier to assess efficient mechanisms to do so.
Identify the 'right' customer	By carefully identifying which customers will make use of the product, the user requirements can better be identified.
Define a complete business case	The business case should consider stakeholders, goals, budgets, and specifications to incorporate as much knowledge as is available from the start.
Prevent pre-project fixed budgeting	Budgeting in accordance with agility when more is learned over time is more suitable, rather than having pre-fixed budgets, which tent to be restraining; especially when stated before the specifications.
Share the risks of developing the end product	When allowed by legal constraints, sourcing models which share the risks with external partners are preferred to prevent overexposure of the project manager.
Stay away from traditional risk management	Risk management should be tailored to agile approaches to not disrupt them.

Perform systematic reviews (also on resource availability)	The reviews planned in agile methods that are planned should also really be performed.
Use testing automation	Testing automation proves valuable as a monitoring, measuring and reporting tool which fits the agile approach.
Develop and use a common language suited to agility	A common language which resonates throughout the organization improves the interlevel communication and steering.
Have a single product owner	Limiting the product owner to one person or entity limits complexity.
Care for the tensions between technology and business	The tensions which will arrive in decision-making authority between the business and technical aspects of the system should be cared for so they are executed in a desired manner.
Train and educate on agility's functioning throughout the whole organization	To ensure that the agile functioning is understood, performed well, and is not hampered by staff behavior, education and training is needed.

Table A1.16 Control mechanisms agile project governance

1.5 Stakeholder interaction

As can be seen from the previous paragraphs, there is still specific information missing on how to go about customer and stakeholder interaction, even though its importance has been repeatedly stressed. This paragraph therefore dives into this issue by considering process management and management in networks.

1.5.1 Process management and management in networks

Process Management (De Bruijn, Ten Heuvelhof, & In 't Veld, 2010) provides a management approach which focuses on the process of decision-making in multi-stakeholder environment. It is closely related to decision making in networks (De Bruijn & Ten Heuvelhof, 2008) since these multi-stakeholder arenas have the characteristics of networks where different actors have their own strategic interests and operate on a rather horizontal level. This paragraph provides a combination of the two approaches while it primarily focuses on process management. Even though the approaches embody recommendations for decision-making, they are still used in this research for their valuable insights to reach a fruitful process of stakeholder interaction. Especially since the meetings with stakeholders and the related feedback can be regarded as negotiation processes, mechanisms on how to steer this interaction can provide worthwhile results for agility. Whereas this paragraph provides insight into the main points, both books entail further recommendations tailored to specific situations.

Main principles

Treat networks as networks: Since decision-making and progress in a network takes place in a political arena with different actors, each with their own interests and strategic behavior, it is crucial that these characteristics are acknowledged and to even accommodated for in this approach. This will help prevent the blocking of the process by actors or even the absence of important players which are required for cooperation (De Bruijn & Ten Heuvelhof).

Process over project tools: Given the network characteristics, and therefore horizontal nature of the stakeholder environment, a management approach should generally use facilitating and steering mechanisms rather than command-and-control instruments. By incorporating a process-approach one can also enhance transparency which can help de-politicize the cooperation (De Bruijn, Ten Heuvelhof, & In ‘t Veld).

Main concepts

Network characteristics: As aforementioned, special attention should be paid the network characteristics. Generally, networks entail interdependent actors of wide variety of which some are closed to the process. This also helps explain why unilateral control-oriented approaches hardly find resonance in network situations. The wide variety can make it easier to find partners which are open to the process, but sometimes actors can be potentially important, or even crucial, partners, while they are not open to cooperation from the outset. The process should focus on the network and actor characteristics to ensure desired stakeholder interaction. Attention should be paid to the strategic behavior of such actors, which generally can be flexible and rather unpredictable in dynamic settings (De Bruijn & Ten Heuvelhof).

Four elements of process design: A process should be open, protect the core values of participating stakeholders, provide progress, and deal with substance. Therefore, the process should be considered as fair and entail clear "rules of the game" to enhance trust of the stakeholders in the process. Going into detail can bring major issues to a process, so timing is crucial and linking of issues can be helpful (De Bruijn & Ten Heuvelhof).

Sense of urgency required: Stakeholder involvement cannot be taken for granted and might need some management to secure interaction. Therefore, a sense of urgency should be present which entails two components. Firstly, there should be the perception that there is an issue which needs to be solved and, secondly, the stakeholders should be convinced that cooperating through the respective process can solve this issue (ibid).

Negotiated knowledge: It is sometimes forgotten that stakeholders can have different perceptions of a problem or its definitions. Within the process it is therefore important that there is an agreement on the perspective towards the issue and "correctness of the information" (ibid, p.70). This is rather similar to the use of common language in scrum. The difference is, however, that this extends beyond one organization to the multi-actor stakeholder playing field.

A process can't be measured: Measurement is a rather important aspect of control, as the previous findings of the research illustrated. Yet, the stakeholder involvement process cannot be measured objectively as there is generally no consensus on the criteria and the problems and solutions are rather dynamic. Yet, the stakeholder interaction can potentially benefit from the findings of holacracy, to assess whether stakeholder involvement leads to output, user requirements, which are not objected by other stakeholders.

Roles

Process architect: The process architect designs the process in such a way that that parties with a stake in the system are included and that they have opportunities to achieve their own interests and have their core interests protected.

Process manager: The process manager ensures that participating parties adhere to the rules of the game, that communication is efficient, and all voices are heard, along with solving other issues which might arise to keep the process fruitful and desirable for actors to engage in the process.

Main strategies on actor involvement

Manage relationships with redundancy: Redundancy in the network can be useful as it enhances the available information. Yet, redundancy of relations of an actor participating in the process makes that actor more unpredictable as it has more strategic options.

Managing winners and losers: The process of stakeholder interaction can bring situations where there are not only winners, but also losers. Especially when these parties are important to the process at a later stage, or in public sector projects are even required for legitimacy, the losers should be managed. Examples on how to go about this phenomenon are: stressing that it is a multi-issue process or find compensation on another turf. Within agility, it would be important that this does not end up damaging the value of the releasable product or service.

Naming and framing: In case the process slows down or stakeholders appear to be left out or damaged by the process, naming and framing can create a positive influencing mechanism to get the process back on the right track: that of fruitful interaction and cooperation.

Embrace 'the shadow of the future: Referring to Fearon's (1998) concept of 'the shadow of the future', the process should encourage good cooperation since parties are expected to interact throughout later stages of the project. Furthermore, bad experiences are to be avoided, since this can set the tone for future interaction. Therefore, unilateral actions considered unfair by stakeholders, can be rather damaging for the process, just like a too loose approach which leads to nothing can set a demotivating undertone.

Actor analyses: As no network is alike, the approach taken should focus on the characteristics of the network and its actors. It is therefore important to know what the core interests of the parties are to ensure these will not be damaged. Furthermore, it allows for identifying of potential partners and to recognize the intentions of actors' strategic behavior. Continuous actor scanning is a useful tool for this process.

Using experts for negotiated knowledge: In case that the involved actors do not manage to come to negotiated knowledge, experts can be used to facilitate the discussion and prevent an agreement which is incongruent with the actual state and characteristics of the system. They therefore can help achieve negotiated knowledge and prevent "negotiated nonsense" (p.71).

Main strategies for a sound process

Perform the process as a multi-issue game over time: At certain moments stakeholders can be afraid to commit to decisions, or decisions can be deemed too sensitive or contested by themselves. Issues can therefore be coupled to broaden the discussion. Furthermore, timing becomes an important element, where certain issues might be frozen when deemed to be disrupting, to be unfrozen at a later stage when the momentum or decisions taken in the meantime ease the introduction of such issues.

Create a sense of urgency: When there is a lack of sense of urgency, the agenda can be altered, and framing can be used to stimulate parties to engage in the process.

Ensuring a fair process: As aforementioned, actors need to have trust in the process next to a sense of urgency. The process should therefore be fair, which entails that the rules of the game are kept to after establishment and are preferably changed through the process if need be, and that core interests of engaged parties are respected.

Use windows of opportunity: Throughout the process, windows of opportunity can come up, either through internal or external developments. These moments can be used to put certain issues on the agenda which were beforehand to sensitive or not mature enough content wise. Also when the process slowed down, these moments can be used to increase the perception of gain and provide new stimulus to participation in the process.

Control mechanisms

Control mechanism	*Description*
Ensure a fair process	A fair process should be provided for to keep parties engaged in the process in a positive and productive manner to ensure their trust in the process. This is done by setting and obeying to rules of the game and protecting core interests.
Secure (perspective of) progress	The process should flow and therefore requires progress, so parties are interested to keep being engaged. When general progress is lacking, quick gains can create this image or framing can be a useful tool.

	Furthermore, windows of opportunity should be used to strengthen the progress.
Create and secure a sense of urgency	To ensure that the stakeholders are and remain inclined to participate there should be a sense of urgency that participating in the process is in their interests.
Selection of partners	It is important that the right parties have a seat at the table early in the process. The selection should not only focus on the power of actors but should also relate to substantial elements.
Continuous actor analysis	Throughout the process, the involved parties should be analyzed on their core and strategic interests. This helps to identify strategic behavior of actors or their desired outcome of the process.
Command and control can be used when necessary	While process management provides a counter-narrative to command and control project management, it does not mean that there is no room for such interventions. It even recognizes the need for hard interventions when a facilitating or steering management approach does not lead the process further.
Different roles	The process should be orchestrated and managed by different persons.
Negotiated knowledge	To ensure that the process is oriented towards a common goal, the parties should have a shared perception and interpretation of the problem at hand. Experts can be used to validate this knowledge or assist in its creation.
Make it a multi-issue game	Even when it is one product that is developed, the process can be managed as a multi-issue game with (un)freezing, coupling and efficient timing of issues to facilitate the process.

Table A1.17 Control mechanisms process and network management

Appendix 2 Interviews

This appendix provides insight into the preparation process of the interviews, along with its academic basis.

2.1 Setting up the interviews

There is a multitude of sources available on how to prepare and conduct interviews. For this research Rubin and Rubin's "Qualitative Interviewing: The Art of Hearing Data" (2011) is chosen given its detailed recommendations and vast stance in academics due to its frequently being cited (over 18,000 at the time of this research), to provide guidance and enhance the academic validity of this process.

Choosing the interviewees

As the source of the to be retrieved information, the choice of interviewees is crucial for the quality of the data and thus results of the research as a whole. Rubin and Rubin mention four aspects on which a potential interviewee should be assessed. Firstly, an interviewee should have the actual experience to talk about the subject at hand. Secondly, this person should be knowledgeable, given the premise that just that a person should know given previous positions does not mean he or she actually does. Thirdly, the selection of interviewees should provide "a variety of perspectives" (p.67) to provide a complementary picture and where possible even crosscheck retrieved data. Finally, the selection should already incorporate the tendency to build and test the theory around which the research revolves. For this research this entails for instance that not only team-level staff was interviewed, but also senior management and a deputy director. Furthermore, to provide a complete picture, relevant external parties should also be included in the interviewees, which has been done by including a use

Research thorough and accurately

This entails that relevant follow-up questions should be asked, facts should be checked and that relevant options are explored. In general, the researcher should be open to new findings and track newly discovered paths which might lead to important insights. Accurate research entails both the absence of errors due to mistakes and caring to represent exactly what was said to the extent possible when it comes to interpreting the retrieved information. Accuracy can be enhanced by checking whether the researcher's interpretation of the setting and unfolding of the process that is being discussed.

Assess believability

It is important to receive believable information even though this cannot be guaranteed. Interviewees can have their own motives to keep themselves from speaking the truth, for instance when they were the one accountable for a mistake or if they are afraid criticism would endanger their position in the organization. Redundancy of information can be used to check statements and it can help to already have some important information before asking sensitive questions. Rubin and Rubin, for instance, recommend to even start with low-level staff in certain cases before speaking to those higher up the ladder.

Be transparent and credible

When presenting the retrieved information and the conclusions made it should be clear to the reader which are the words of the interviewees and which of the researcher. Clear transcription and citation enhance this process and adds to the credibility of the research.

2.2 Process of the interview

Rubin and Rubin describe a five-step process which should be followed when conducting interviews. Firstly, the researcher should introduce himself and the research, which is followed up by some simple questions to get the conversation going. When there is already a certain flow, more sensitive questions can be asked which might make the interview a more emotional process for the interviewee. Finally, the interviewer should tone down the emotional level with less-stressful questions before closing off.

This research provides a similar procedure, since it is not known up front whether the interview can bring emotional response from the interviewee. In general, the research reflects the different stages, since, first, simple questions are asked about the project and the interviewees role. Then the evaluation of control mechanisms takes place which can potentially be a more emotional process for the interviewee in case there was project failure. Afterwards, the interview will focus on the control mechanisms which were not used in practice but are mentioned by the literature. This starts to take the interview away from the emotions which might be linked to previous control mechanisms and the unfolding of the project. Furthermore, the research uses the recommendation of Talmy (2010) to give the interviewee the possibility to make comments, which can be used as feedback or provide even more valuable information, or even ask a question at the end of the interview.

2.3 Interview questions

The interview can be categorized into three types of interview questions: main questions, follow-up questions, and probes. In general. the questions should be open and not leading.

Main questions

These questions stand at the core of the interview and can be considered as its "skeleton" (Rubin & Rubin, p.134). They should ensure that each part of the research problem is explored in the interview. Main questions are usually defined prior to the interview taking place to ensure this process. There is no set number on such questions although these are many in the given research given the diverse yet detailed nature of the control mechanisms.

Follow-up questions

These questions are reactions to statements made by the interviewee to further explore and retrieve details. While follow-up questions relate to specific statements by an interviewee, they can to a certain extent be prepared. Yet, the interviewer should be aware and think quick to follow-up appropriately.

Probes

Probes, just like follow-up questions, serve to dive deeper into a narrative and retrieve details. Yet they differ since they do not lead the interview to another relevant topic, but dive into detail on the one that is discussed at that moment. For instance, by asking for examples or clarification on statements made.

For this research, the main questions are oriented on the control clusters to identify which mechanisms were used. Then probe questions are asked to analyze the use of the mechanisms and the ways in which they were used, while follow-up questions are used when a potentially interesting topic occurs.

Interview protocol

As a guiding document during the interviews, the author used an interview protocol as recommended by Creswell (2017) yet extended with the previously mentioned points by Rubin and Rubin to assist in ensuring the interview process goes as desired and required information is attempted to be retrieved by asking the right questions.

1. Heading
2. Instructions for procedures (including introduction and concluding statement)
3. The questions (main questions, probes, areas for follow-up)
4. Evaluation of information (primary/secondary statements, reliability)

2.4 The interview protocol

Interview for research on agility in public sector projects

Interviewer: Ferre Westermann

Interviewee:

Position interviewee:

Date:

Time of interview:

Introduction

Good morning / Good afternoon,

First of all, thank you, ……, for making some time available to meet with me today for this interview on the use of agility in the project….

As mentioned in our previous correspondence, my name is Ferre Westermann and I am conducting this research for the thesis research of my MSc Complex Systems Engineering and Management at TU Delft.

My research is oriented at agile projects in the public sector and …. is part of a case study. The overall aim is to identify mechanisms which can assist in performing control. In this light, a control mechanism is a technical or social instrument or action which is used with the aim to enhance control over a specific component or execution of the project, either related to operations and/or strategy.

Before we start, I would like to ask for your approval to record this interview and I would like to ask if you would prefer to take the interview anonymously.

Before we start exploring the use of agility in the project and the control mechanisms that were used, could you please give me a short description of your role and tasks?

Main questions

1.How is the agile process set-up?

2. Are there aspects or components of ARTs not under control or where is more control required?

3. Which control mechanisms have been used/are being used to enhance control and what is/was their effect?

4. Where does the desire for control come from?

Probes

1. What do you mean specifically with… ?

2. Can you give examples of how the mechanism was used?

3. Can you give examples of how this affected control over the process?

4.What was the reason for this development?

Follow-up questions

1. How did this relate to?

[2. in general make links between the different mechanisms or clusters for relationships]

[3. In general refer the technical unfolding of the project to the social dimension and vice versa]

Concluding statement

Thank you very much. Your answers to my questions will surely be of good use.

Before concluding, I would like to ask if you have comments on the proceeding or content of this interview or perhaps even have a question for me.

Thanks again for your cooperation. Lastly, I would like to ask you if you believe there is a team member or external party who would be able to provide further valuable information for my research.

Evaluation [after interview]
Is the information provided reliable? Which statements were of primary or secondary nature?

Table A2.1 Interview protocol

Appendix 3 Analysis case findings

This appendix includes the analysis of the case findings which lead to the results as presented in the main report. As mentioned in paragraph 3.5 of the main report the research uses thematic coding to identify organizing themes by clustering the identified codes. Each segment starts with the codes belonging to that cluster which is then followed by further analysis. First, the benefits of agility are discussed along with the desire for control and choice for agility before the Appendix presents the analysis of the mechanisms according to cluster to provide an overview of the identified control mechanisms along with their characteristics according to the Revised Levers of Control framework. More information can be found in the other documents related to the thesis report: Appendix 4 includes the full interview transcripts while Appendix 5 contains the codes along with their quotations (both are, however, confidential documents).

3.1 Agility and the organization

Benefits of agility

Closer relation to business	More effective but not efficient
Embraced by employees, except CT	More cooperation
Happier teams	More transparency

Table A3.1 Codes related to benefits of agility

Interviewees indicated the benefits of agility, where increase in transparency, communication, cooperation and faster delivery were mentioned along with that it makes teams happier. This includes a closer relation between business and developers while also challenging the former to think of what it wants. Yet, it was also mentioned that even though agility makes cooperation easier, the road to get there can be a tough one

"Eventually it makes it [cooperation] easier, but you first have to cross the road to get there. You have to talk a lot to each other, communicate, make and invest time in contacts and relations and get all eyes in the same direction. Really put in a lot of attention and energy because it does not work yet how it should". (Program level employee)

Desire for control

Big projects going overbudget and years of delay	False sense of control
Deadline	Increases with short timeframe
Enhancing flow and predictability	Personal

Table A3.2 Codes related to desire for control

When the interviewees were asked about where the desire for control comes from, a clear distinction can be identified between respondents from the team level or from higher management. While team members stated their desire for control is related to facilitating team performance and enhancing flow and predictability, higher management mentioned rather different aspects. Accountability, achieving results, and the presence of a deadline were mentioned to increase management's desire for control, which increases when there is shorter time left. But also the current system of HR planning provides incentives for control, as the budgeting for HR is performed years in advance in the Directorate. Two interviewees stated that the desire for control enfolds rather as a fake appearance of control, than real control, where higher management wants to have the feeling to be in control, even though the methods through which this is enacted might not be the most suitable. An example of this is the focus on metrics (more in Metrics, technology and tools) This has been strengthened over the years by multiple big projects going over budget and suffering heavy delays.

> *"My desire for control? Because I have to be finished on time with the development of the service" (Program level employee)*
>
> *"Because here projects are delivered with millions over budget of years of delay, there is more desire to manage and control, to get a better grip on everything" (Program level employee)*
>
> *"Because they want 'appearance of control'. Because I believe that when you have your process well taken care of, that you do not need many other control mechanisms. The better your process is in order, the less mechanisms you need… Because I see control mechanisms as an inherent part of the process." (Program level employee)*
>
> *"The desire for control… that is purely oriented from my responsibility, and that increases when the pressure increases and time gets shorter, the desire than increases". (Program level employee)*
>
> *"My desire for control comes from the fact that I want to deliver results faster than I currently can. My desire for control thus comes from wanting the preparation to be done correctly, so my team can easily identify what the next step will be and that they do not have to invent a lot of things again, no." (Team level employee)*

Choice for agility

Already implemented before decision higher up
Business cares about results not methodology
Started at the bottom

Table A3.3 Codes related to choice for agility

The interviewees were also asked why agility was chosen for the projects at hand and most of the respondents answered they did not know exactly why. Yet, a recurrent statement was that agility was implemented by the IT section of the organization and started to grow from the bottom-up, which entails that it was not a clear decision made by the business: they simply had to follow up. Two interviewees representing the business mentioned that the business generally does not care too much about what methodology is used as long as it provides results. One interviewee present from the start mentioned that the first time agility was implemented in the project, it was because there was a dire need to develop fast, even though they soon found out they were mistaken as this led to technical and functional debt as there was no guidance for the process.

> *"We were a bit surprised by the IT side of the organization who took the decision [for agility]. That did not really happen with the business. It was… more of an announcement and this is how we will do it. That led to the business not being able to make those steps immediately" (Program level employee)*
>
> *Agility started from the bottom-up, driven by the IT section, without clear approval from the business side of the organization. Agility in the projects at hand was first used for its expected benefits to develop fast but the lack of guidance created technical and functional debt. (Program level employee)*

3.2 Control clusters and mechanisms from case findings

Through the method of thematic coding, the following code categories were identified, which are to a large extent similar to the categories derived from the theoretical exploration of chapter 4.

Agile mindset, skills and leadership	Planning
Continuous change	Portfolio and budget
Cooperation, communication and feedback	Specification and prioritization
Delivering value	Teams, roles and responsibilities
Events and procedures	Uncertainty and learning
Interfaces	User and stakeholder involvement
Metrics, technology and tools	Vision
Organization	Waterfall and agility

Table A3.4 Control clusters from case study

Each segment starts with the identified codes belonging to the cluster. Then a text is provided which gives insight into the findings. Afterwards some relevant quotes are presented before the main summarizing conclusions of the cluster are presented along with the control mechanisms including their Revised Levers of Control classification.

Concept	Definition
Strategic/operational	Does it relate to the operational or strategic aspects?
Performance/boundary	Is the objective of the mechanism to create and enhance performance or is the end goal to comply with regulation or standards?
Social/technical	Is the mechanism based on social aspects (i.e. values and norms) or technical (i.e. rules and procedures)?
Diagnostic/interactive	Is the mechanism used rather continuously or only when there is a specific need to?
Enabling/constraining	Does the mechanism enable creativity and flexibility, or does it increase predictability and reduce options?
Reward/punishment	Does the mechanism provide a reward or punishment?

Table A3.5 Characteristics of the Revised Levers of Control framework

Agile mindset, skills, and leadership

Increasing control	*Losing control*
Agile helps create a feeling of `us`	Absence of agile leadership
Coaching and training agile	Agile does not always fit all team members

Investing in staff and ICT knowledge Just begin	Agile doesn't flow Agile mindset not that present yet Agile places higher demands on staff Business needed switch mindset Culture lack of responsibility Development teams not ready for agile Fear for failure Going for optimal not workable Guidelines over value Lack of discipline Lack of focus Lack of sense of "us" Lack of support for agile from higher up Mindset not on short-cycles Not always positive atmosphere Not enough attention on preparing and training staff People need to trust each other more SAFe little guidelines for taking along higher management SAFe too early Scrum requires specialized skills and knowledge Surprises decrease trust Teams rather pragmatic with metrics Too much according to the book counterproductive Top-down mindset not on facilitating teams Unhappy employees

Table A3.6 Codes related to agile mindset, leadership and skills

An overarching conclusion from the interviewees was that the agile mindset was not present yet even though it was growing. When it comes to the mindset surrounding delivering value, there is fear for failure, tendency to go for optimal solutions rather than workable, and that

guidelines hamper the development of value. Furthermore, it was mentioned by interviewees on the team level that there is a lack of support from higher up for teams and that the **mindset** is not set on short cycles yet. It also seems problematic that there is a culture where lack of responsibility is embedded, and at times more discipline and focus can be imagined. One aspect of the agile mindset was used, even though this took longer than expected in one of the trains, which is to **just begin** and not remain in the preparation of specifications phase.

When it specifically comes to **leadership**, it was mentioned that there is an absence of agile leadership from the top. It did occur that a message from management that they did not want to be too innovative provided a solution to a situation of conflict, created by the lack of such **clear vision**.

Another observation from the interviews was that agility does not always fit all team members, and when there is too much "according to the book", it can work counterproductive and even hamper the agile mindset of cooperation. Furthermore, sometimes there is deviation from SAFe, for instance by teams who can at times be rather pragmatic with metrics

Agility places higher demands on the skills of staff than traditional waterfall, and it was mentioned that the teams perhaps were not **ready** yet for such full agile implementation.

Lastly, multiple statements were identified related to the atmosphere at hand. Not only is there at times a lack of '**sense of us'** but there is also a lack of **trust**, both in the ARTs and in the organization. The **atmosphere** is also not always positive. It was mentioned, however, that agility helps to develop a 'sense of us'.

> *"With a part of the population within the organization that [agile leadership] is missing. But it is very important, management should stand behind it. A part does and a part does not, it does not go smooth" (Team level employee)*

> *"The development like you would ideally want it is with trust in the teams that they come up with the best solution for your problem, that is not there yet completely, that is a cultural thing." (Program level employee)*

> *"You see that [this department] has trust in its own organization, but you also see that in the interfaces there is little trust." (Program level employee)*
> *"(in relation to using wrong metrics)… but it gives us as a team the feeling that we are not trusted. They should have more courage to let the teams work." (Team level employee)*

Agile mindset: Just begin					
Social	Performance	Operational	Interactive	Enabling	-

Agile mindset: Go for workable solutions in short cycles					
Social	Performance	Operational	Interactive	Constraining	-

Agile leadership: Facilitate the teams					
Social	Performance	Strategic	Interactive	Enabling	Reward

Agile leadership: Create a positive environment of trust, motivate, promote accountability					
Social	Performance	Operational	Interactive	Enabling	Reward

Agile skills: Train staff					
Social & Technical	Performance & Boundary	Operational & Strategic	Interactive	Enabling & Constraining	Reward

Continuous change

Increasing control	*Losing control*
Continuous adaptation	A lot of discussion but no decision
Continuous change prioritization	Discussions continue after decision made
	Cannot keep up with technological innovation
	Fear for failure
	Going for optimal not workable

Table A3.7 Codes related to continuous change

Just like is meant to in agility, **continuous change** aids the control over the agile process as mistakes can be rectified. This was showcased in an example where it became clear during an **agile PI event** that there was a mismatch between what the Business Owner desired and what the teams were going to work on. Work was reshuffled and in the end the team members even mentioned this was one of the most helpful moments of the PI. Adaptation is also used in the **prioritization**, since it can be that priorities change given external uncertainties or

developments in the interfaces within the organization. Yet, it is also difficult to perform continuous adaptation since there is a tendency to discuss rather long in the management spheres until a decision is taken, and sometimes when there has been one, the discussion continues while it cannot be changed, for instance with regards to the chosen development technology. There is also a fear for failure which prevents commitment to a certain trajectory which might be changed. This is linked to the phenomenon that there is a tendency to go for optimal solutions, and not **workable** ones. Lastly, the technical improvements in the environment of the projects develop so fast that even continuous change cannot keep up with these. Even though this is not a hard requirement of agility, it is important to note that the slow **pace** in which the trains perform now, this is not possible.

> *"That is something the [organization] also finds very difficult because they rather make things correct. But making everything right in one go means you need to prepare really well, and that preparation entails you might miss things you can only see in the working system" (Program level employee)*

> *"… And then you together look at 'what was the conclusion of today', 'are we good on our way or do we need to step in'. And stepping in fast, that is an important and rather crucial factor." (Team level employee)*

> *"You want to do the refinement sessions with your teams as late as possible because in the project the priorities can change or the world around it can change." (Team level employee)*

Continuous adaptation of tasks and prioritization					
Technical	Performance	Operational	Interactive	Enabling	-

Agile mindset: Go for workable solutions					
Social	Performance	Operational	Interactive	Constraining	-

Agile mindset: Decrease the fear of failure					
Social	Performance	Operational	Interactive	Enabling	-

Agile mindset: Better to commit and change later on than not commit at all					
Social	Performance	Operational	Interactive	Enabling	-

Cooperation, communication and feedback

Increasing control	*Losing control*
Collaboration and feedback Collaboration RTEs Combining reports when possible Cooperation agile and waterfall teams More involvement BO Open communication PM more involved Realization teams in the same room Specification team in same room as realization teams Teams interfaces collaborate better in shared space	Architects different cadence than teams Certain information not transparent in organization Communication not always efficient Different cadence leads to uncertainty Different cadence teams Different languages between ARTs Different languages business and technology Inefficient communication and cooperation business and technology leads to frustration Influence from outside the train though wrong channels creates delays Information at the wrong time Information sharing and searching not used to its potential Lack of continuity in tasks Lack of cooperation Lack of participation events team members Lack of reporting Lack of sense of "us" Lack of trust and appreciation for teams Lack of trust between business and technology Lack of trust between interfaces Little feedback between teams Management in beginning not involved

	Management not understanding implications changing team membership Management switching positions More insight in process needed to gain trust No control over team] Nondedicated teams Not always positive atmosphere People need to trust each other more Tasks requiring specialized knowledge Tasks to specialized to take over Team instability Team members do not always fit Team membership not stable due to changes IT departments Team staff decreasing Teams not involved in retrieving new staff Too much according to the book counterproductive Unhappy employees Workspace that does not facilitate agile

Table A3.8 Codes related to cooperation, communication and feedback

As can be expected, increased cooperation benefits the agile process with examples where the RTEs of the two trains **discuss** which metrics to use or a Product Manager who is more closely involved. Yet, there are issues with regards to cooperation. There is still relatively little **feedback** between teams which is enhanced through **sharing of workspace** and **similar cadence**. Due to the large size of the organization, however, interfaces generally run on different cadence and there is even a team within the national train which runs on a different cadence, leading to a loss of almost two days every sprint. Furthermore, the tasks to be executed are rather specialized and not everyone can take over or help. Finally, there was a lack of **involvement by management** in the beginning of the process, leading to little input for teams at the time, leading to technical debt which needed to be solved at a later stage.

Workspace plays an important part in facilitating cooperation and this is shown in the case as both realization and specification teams cooperate better when office space is shared. Yet, it is important to note that in many other ways, the offices of the organization do not support agile working, as there are few spaces where teams or team members can separate themselves for quick discussions or team meetings, and a lack of workspace on popular days given the flex-work policy. One interviewee mentioned it can take fifteen to thirty minutes to find a suitable room or that short meetings are even held at coffee machines. There is also no room to hang templates, so the teams planning boards are placed on the wall of the room where also the lockers are.

Open communication also contributes to the performance of the process, which requires certain control to be performed efficient. In general, however, certain information is still not open in the organization, as an example was mentioned with a new team working on the implications of Brexit, which works separated and closed off from the rest of the organization, and the cooperation and information sharing and inclusion between different departments can be more transparent.

Certain departments have different **standards for reporting** in which case it helps to combine reports when possible. It was also mentioned that management still uses traditional forms of reporting which do not really suit agility while a lack of reporting on team performance was mentioned by someone representing the business. More on that is discussed in the paragraph on metrics.

Finally, it was mentioned that more **trust** is needed since there is a lack of trust present in the cooperation. Not only is there a lack of trust between the business and technology, but also between interfaces, and certain members of teams feel unappreciated and not trusted by higher management. This leads to a situation where there is also not always a **positive atmosphere** in which cooperation takes place.

> *"It is a big room with three teams which is very cozy and fun but on Monday Tuesday and Thursday it is actually too crowded. And when you then want to discuss something or brainstorm you have to look for a spot which is not really there. Also if people would like to retreat to a quiet place to work in concentration, that is also not possible. It makes it [agile work] difficult." (Team level employee)*

> *"Sometimes you are looking for a spot so long to sit together. And we then often sit at the coffee machine, there is a TV too. Very nice to show it like that. But there are a lot of people*

getting coffee all the time who also want to chat… that is very disrupting." (Team level employee)

"You notice that they try to protect their own island. If you are head of a department, that your area, your tile is clean, and then you notice they do not look for cooperation. 'This should not be done with me because then my processes will go different again while my tile is clean'… Yes you are responsible for your own tile… but if your neighbor's tile is dirty then the process just does not flow. Also there is mention of a culture switch." (Team level employee)

Combining reporting standards for different departments and only report what is necessary					
Technical	Boundary	Strategic	Diagnostic	Constraining	-

Create a shared language					
Social & Technical	Performance	Operational	Interactive	Constraining	-

Open communication and transparency					
Social & Technical	Performance	Strategic & Operational	Interactive	Enabling	Reward

Realization and specification teams working in the same space					
Technical	Performance	Operational	Interactive	Constraining	Reward

Enhance the agile workspace					
Technical	Performance	Operational	Interactive	Enabling	Reward

Ensure teams have similar cadence					
Technical	Performance	Operational	Interactive	Constraining	-

Agile leadership: Create a positive environment of trust, motivate, promote accountability					
Social	Performance	Operational	Interactive	Enabling	Reward

Delivering value

Increasing control	*Losing control*
Just begin	Deadline over value
	Desired functionality not flexible
	Fear for failure
	Functional debt because fast
	Functional debt because not financed in past
	Going for optimal not workable
	Guidelines over value
	Path dependency minimal development
	Path dependency requirements old systems
	Technical debt
	Technical debt because fast

Table A3.9 Codes related to delivering value

The delivering of value in agility is increased by **just starting** the development, although guidance on **requirements** is required to prevent creation of technical and functional debt. Besides this, however, there are two groups of categories which hamper the delivering of value so central in agility. Firstly, there are factors related to the **mindset and culture**. Given the importance of external deadlines given regulations or internal interface dependencies, time sometimes plays a bigger role than value. Furthermore, there is a fear for failure which drives the striving for optimal solution rather than **workable** ones. The numerous **guidelines** also provide boxes which need to be ticked off, which shapes the deliverance of value, along with path dependency on requirements of already existing systems. Secondly, the delivering of value is hampered by the functional and **technical debt** present in the ARTs which can hardly be left as it is since the desired functionality is not flexible. Functional debt is present given fast developments or incomplete ones given lack of funding, which also counts for technical debt, although this also can derive from requirements of interfaces to keep the technology up to date.

> *"[There is a lot of technical debt] because we previously created a situation in which there was little time to develop the services well" (Program level employee)*

> *"You now see a couple of points that we want to have in for the architecture… all project units which have not been realized because in the short time available to produce*

productivity for the business to do its job to execute work from Brussels and now you have an uncontrollable situation that is expensive to adapt again" (Program level employee)

"That is something the [organization] also finds very difficult because they rather make things correct. But making everything right in one go means you need to prepare really well, and that preparation entails you might miss things you can only see in the working system" (Program level employee)

Agile mindset: Just begin					
Social	Performance	Operational	Interactive	Enabling	-

Agile mindset: Go for workable solutions					
Social	Performance	Operational	Interactive	Constraining	-

Agile mindset: Decrease the fear of failure					
Social	Performance	Operational	Interactive	Enabling	Reward

Have guidance from specifications in place from the beginning to steer creation of value					
Technical	Performance	Strategic	Interactive	Constraining	-

Change guidelines to fit agility					
Technical	Boundary	Strategic	Diagnostic	Constraining	-

Measure technical and functional debt and solve when required					
Technical	Performance	Operational	Interactive	Constraining	-

Events and procedures

Increasing control	*Losing control*
Agile elements in place	Business influencing timing and structure events
Agile events	
Agile events prevent mistakes	Business not invited

Agile events prioritizing Agile planning on non-SAFe teams Inclusive PI events Insight into agile events helps when to visit	Different cadence leads to uncertainty Different cadence teams Difficult planning PI due to agendas Diverging from sprint planning In the beginning time is rather short to finish all tasks Lack of content discussion in events Lack of participation events team members Sprint planning difficult with vague specifications Too much according to the book counterproductive

Table A3.10 Codes related to events and procedures

From the interviews it can be seen that agile events add to the control of the agile process. **Inclusive agile events** enhance cooperation and communication while preventing potential mistakes or miscommunication and can even be used for the prioritization. It also appeared that the **agile planning** can work on non-agile teams, such as with the Continuity Team in the National ART. Yet, some issues were identified which hamper the control over the process. Firstly, certain representation of the business and users were not invited to the events in the beginning and only joined at a later moment, while their input in the beginning could have been valuable. The business is increasingly participating and it is now even coming to a point where it starts to influence the planning and agenda of the agile events. While it is in general difficult to plan the events given peoples' busy **agendas**, this new influence does not fit the **facilitating of teams**. **Sprint planning** is furthermore difficult when the **specifications** are vague and the difference in **cadence** between teams, present in the National ART, also creates uncertainty and delays. Teams should also keep themselves to the sprint planning and there is a challenge in **having all team members participate in the events** since not all members are interested while they can provide helpful input or might hear things interesting for their own tasks. Furthermore, there is sometimes a lack of **discussion on content** in the events, and it was also stated that when the events are **too much according the book** it also doesn't help. One Scrum Master stated that a predecessor cut of an important discussion because the reserved time of fifteen minutes was passed, which hampered the discussion and led to frustration in the team.

"Some team members do not like to go these events and when we would push them, they would just sit and code on their laptop. When I attend the event with another team member, we might not know that specific information and would have to make assumptions which would then again have to be checked later on." (Team level employee)

"You also have ICT staff that is not always open to cooperation and would rather code than be in a meeting while these meetings can give them a lot of background information for their work, but to involve them in that and show it that they see the added value, there is a lot of work in that" (Team level employee)

"To go to [office location] and visit the PI's we reached a lot of insight into the process which gave a clearer overview. Clearer to see when to jump in and influence more by describing what the business wants" (Program level employee)
"Also with the Product Management meeting with 15 Product Managers, but this is more an update session than a joint discussion and prioritization." (Program level employee)

Agile events					
Technical	Performance	Operational	Interactive	Constraining	-

All relevant parties should be invited					
Technical	Performance	Operational	Interactive	Enabling	Reward

Planning of agile events is done by the RTE and is primarily team-facilitating					
Technical	Performance	Operational	Interactive	Constraining	-

Ensure teams have similar cadence					
Technical	Performance	Operational	Interactive	Constraining	-

Have a sprint length of 30-45 days					
Technical	Performance	Operational	Interactive	Constraining	-

Ensure relevant team members want to participate in the agile events					
Social	Performance	Operational	Interactive	Constraining	Reward

The strict procedures should leave room for crucial discussions					
Technical	Boundary	Operational	Diagnostic	Enabling	-

Interfaces

Increasing control	*Losing control*
Agile more flexible with interfaces	Conflict interfaces
Agreement between interfaces	Cooperation management interfaces more challenging than team interaction
Agreement interface business case	Creating interfaces when not necessary
Anticipating changes interfaces	Delays because of interfaces
Discussing prioritization with interfaces	Dependencies interfaces
Discussions with interfaces to improve cooperation	Dependencies interfaces leads to delays and frustration
Interfaces can increase speed	Dependencies interface value stream
Teams interfaces collaborate better in shared space	Development interfaces disconnected
Understanding why delays occur with interfaces	How to communicate with interfaces
	Lack of agreement interfaces
	Lack of transparency on decision-making interfaces
	Lack of trust between interfaces
	Non-dedicated teams
	Responsibility on whole value chain while value chain is split
	Technical debt because of interfaces

Table A3.11 Codes related to interfaces

One thing that was very present in the case study, in contrast to the theoretical findings, was the presence and importance of interfaces, either with other system applications, departments, or development teams. The dependencies with interfaces are so present since the value stream in public sector bureaucracies is in general cut up over different departments and systems. These

interfaces can play an important role in the agile process as they allow for more flexibility and can increase speed but from the findings of the interviewees it became clear this process is currently not under control. Development teams can suffer delays due to interfaces, either because they are dependent on them or they have to change their planning to facilitate other projects or updates. This process thus becomes problematic given the lack of agreements and transparency and can lead to delays and technical or functional debt. This can, however, largely be avoided through making **agreements with the interfaces** and **plan interactions** beforehand to smoothen cooperation, by anticipating upcoming changes in development of interfaces, discussing prioritization with interfaces and also understanding why delays occur in the first place. This will not only help the planning, but also adds to the trust between departments, managers or teams involved in interfaces.

"An advantage of agility is that you can be more flexible with interfaces, but this is sometimes misused." (Program level employee)

"[Department B] knows where it is heading and also wants to make a big transition like we are doing with the renewal. There we have the agreement of 'okay fine, that is good plan, but then after January 2021'. These kinds of agreements you do to prevent that in case they do it earlier that it goes wrong and then you have a risk on your project" (Program level employee)

"These parties like to cooperate on that [making agreements and planning] they encounter their own issues too. So meeting together once in a while to see how to make the process go smoother and easier, but it just goes slow, you would like it to go faster" (Team level employee)

Use interfaces for flexibility and fast delivery					
Technical	Performance	Operational	Diagnostic & Interactive	Enabling	-

Cooperation and agreements with interfaces on anticipated changes, prioritization and eliminating delays					
Technical	Performance	Strategic	Interactive	Constraining	-

Metrics, technology and tools

Increasing control	*Losing control*
Agile Tooling Jira and Confluence	Business no insight into performance
Agile Tooling WSJF	Demanding too many and the wrong metrics
Discuss which metrics to measure and share with whom	Difficult to measure performance
Metrics	Generic systems not easily integrated
RTE chooses metrics	Insufficient or incomplete information on performance system
Story points for predictability and flow	Misinterpretation metrics can lead to wrong performance
Tools for automatization guidelines	Misinterpretation or wrong metrics give teams feeling they are not trusted
WSJF for prioritizing	Not that predictable yet
	Pokering points difficult when vague and not always consistent
	Some measures difficult given complexity and uniqueness tasks
	Teams rather pragmatic with metrics
	Technical debt not acknowledged by business
	Technology NS time efficiency
	Too early for sensible data
	Too much reporting RTE
	Top-down mindset not on facilitating teams
	Traditional reporting does not fit agile
	Unstructured reporting
	Waiting for documentation
	Wrong metrics can lead to wrong performance

Table A3.12 Codes related to metrics, technology and tools

Metrics are an important aspect of the agile process as it can be used to increase performance by gaining insight into predictability and sprint planning. However, as becomes present in the interviews, using the wrong metrics can stimulate wrong performance since the teams could

change the point system to look like they are performing better, rather than actually producing the most value. Misinterpretation of the metrics, including comparison between teams can also provide negative effects on team performance as it can demotivate its members. It is therefore important to have fruitful **discussions** between business and technology, management and teams, in which the RTE eventually decides which metrics are used. This can also assist solving the issue of teams being rather pragmatic with metrics and ensure the **right metrics** are used which assist in predictability. Furthermore, it should be noted that at the current stage, the ARTs have a considerable load of technical and functional debt to solve and have also coped with capacity changes, which both significantly hamper the validity of the retrieved points and velocity. **Technical debt** and **implications of change in capacity** should therefore also be measured and given values reflecting function points per hour to be used for planning. The previous presence of vague specification also hampered this process. **Tools** can also be used to enhance control, where Jira and the Confluence Board provide transparent platforms for relevant parties to see, and when authorized adapt, the Program and Product Backlogs. WSJF is used as well to prioritize tasks. It was also mentioned that in the future, automatization tools for guidelines can assist in saving time which can then be spared for delivering value. Opinions over whether technology is efficient or inefficient seemed rather subjective from the interviews, although the difficulty of **integrating generic systems** clearly does not support the agile process. An early presence of **basic architecture** or vision on how this should be created also aids the development process as it provides a main template from which can be built on.

> *"Something we as teams used to be annoyed by, things like function points per hour that are used to compare teams with one another, that is very outdated to use these kind of methods… at the end of a PI event they discuss it and then I think you do not even know what you are talking about. If we as a team would want that we can make that graph go upwards but that depends on what is being documented… we can also change the designs and then the function points go up. Would you want that? Well, we do not. We are not waiting for such a wrong way to show them we are doing right because then we actually are not performing well. These kinds of things and then I think you should not do that because then we as teams will start doing the wrong things" (Team level employee)*

> *"But [he/she] should also realize to still report how the ART is doing. They will report less on deadlines but they should look a lot more like how is my train performing, what can I*

learn, are the functionalities going, is the business value I deliver enough" (Program level employee)

"How management works now 'the team has two new builders so they are at full strength', read 'they can perform on full velocity and deliver functionality'. That sounds nice. Those builders know well how to program Java but they have to get to know the application. Before you get going you are three months further ahead. That is something you need to feedback to management. It is still the old management structure they are maintaining." (Team level employee)

Use the right metrics (team facilitating and planning and flow oriented)					
Technical	Performance	Operational	Interactive	Constraining	-

Incorporate metrics for changes in capacity and technical debt					
Technical	Performance	Operational	Interactive	Constraining	-

Measure flow					
Technical	Performance	Operational	Interactive	Constraining	-

Use agile tooling (Jira and Confluence)					
Technical	Performance	Operational	Interactive	Constraining	-

Use WSJF for prioritization					
Technical	Performance	Operational	Interactive	Constraining	-

Use automatization tools for guidelines					
Technical	Boundary	Operational	Interactive	Constraining	-

Generic systems should be developed to be easily integrated					
Technical	Performance	Operational	Diagnostic	Constraining	-

Have basic architecture (vision) in place early on					
Technical	Performance	Strategic	Diagnostic	Constraining	-

Organization

Increasing control	*Losing control*
-	Accountability drives island structures
	Agile organization reverse pyramid
	Big organization
	Business did not choose agile
	Business not in the lead
	Business not ready for agile
	External staff not knowing implications organizational characteristics
	Financing authority shift to business
	Inflexible organization
	Islands structure
	Multiple reorganizations
	Multiple simultaneous renewals
	Organization
	Organization not ready for agile
	Organizational complexity
	Organizational complexity delays teams
	Organizational complexity requires deviation
	SAFe
	Political games interfaces

Table A3.13 Codes related to the organizational set-up

The organization has been undergoing multiple reorganizations in the past years as one interviewee claimed it to be "constantly moving". Next to the grand schemes of reorganization, there are also constant renewal programs within the different departments, as has also been mentioned in the part on interfaces. It is a rather big and inflexible organization which makes the implementation of agility throughout more difficult. Furthermore, its magnitude drives the department to function as islands with political games, defined as sectarianism in the previous chapters. The organizational dynamics shape the agile process which cannot easily be recognized by externally hired staff. Within this environment, agility was implemented, and from the interviews it came to the forefront that the organization was not **ready**. The choice for agility primarily came from the IT section, which led to the situation where the business was

not ready as it did not possess the **required knowledge and skills** to perform its required tasks. This came to be because a reorganization moved the staff specialized in formalizing the requirements to the IT side, rather than the business side. In the new situation, the business should not only specify the requirements, but is also in charge of the financing. Both tasks are rather new to the business and it is still growing into its task. The organization therefore still had to put things into place while it was starting to use agility and SAFe. Lastly, the organization's **portfolio** embodies specialized tasks while demographics provide a relatively old staff which is not as prone to flexibility, new tasks and new techniques, and the organization can also be considered a reverse pyramid, or as one interviewee called it an "agile water head", with more people in higher management and on the program level then that there are actual team members. Care should be given to having enough **team level staff** internally within the organization. Lastly, it was noted that the number of interfaces increase in public sector agility since the value stream is broken up into functions, such as registration and payment. Over a longer period of time it would be desired if the organization would be shaped in such a way that **departments reflect the value delivery streams** to keep interfaces with other departments as little as possible which would gravely decrease the current complexity and along coming challenge for coordination.

> *"We were one of the first who knew how to use IT for giant processes which made the IT section into a powerhouse… That is a weird interaction that now exists, especially when you see now that the business is put in the driver's seat and has to say what happens at the IT section. That is a difficult culture switch" (Program level employee)*

> *"The process on how to shape it [specification] used to be part of the IM organization and have now gone to the IT. People look at the business but the knowledge has never really been transferred to the business, so we are asking executive units to provide policy who are not organized to do so" (Program level employee)*

> *"Now the organization has a water head with more people in place who have an opinion than that there are people that do it. There are relatively few people and they are even becoming less." (Program level employee)*

> *"This did not use to be a big issue, but because of the reorganization we have split up in four structures. Then something has to go to the other… that does not always go smooth.*

Many captains on many ships, who is keeping an overview, who does what when, there is also machismo" (Program level employee)

"We also have to interact with [department X] which is another branch, so you get a lot more influence around it and they all have their own political agenda, which makes it more difficult to control, because they have their own interests for which they act, that makes it complex". (Team level employee)

Ensure the required knowledge and skills for the agile roles are placed in the right departments before and during implementation					
Technical	Performance	Strategic	Interactive	Constraining	Reward

Prevent agile water head (more management than team members)					
Technical	Performance	Strategic	Interactive	Constraining	-

Ensure a fair process of interaction between departments in which core interests are protected					
Social & Technical	Performance	Strategic & Operational	Interactive	Constraining	Reward

Consider the different projects and activities within the ARTs as a multi-issue game					
Social	Performance	Strategic	Interactive	Constraining	-

Create a common perception shared by stakeholders on the problem and definitions at hand (negotiated knowledge)					
Technical	Performance	Strategic	Interactive	Constraining	-

Organizational change should work towards reflecting value delivery streams					
Technical	Performance	Strategic	Diagnostic	Constraining	-

Planning and delivering on deadlines

Increasing control	*Losing control*
Agile planning on non-SAFe teams	Agile planning difficult

Bring risks to the forefront Planning interfaces beforehand Smaller assignments with clear WSJF	Agile requires time Business Owner wants estimates planning Cannot plan interfaces ahead more than 2 PIs Deadline over value Deadlines Delays because of interfaces Delivery pieces too big for sprints seems like no progress Difficult planning PI due to agendas Difficult to plan long term Mindset not on short cycles Plan-based hampers the now Planning more difficult with big projects Planning more difficult with sickness or inexperienced team members Product Owner more managing on planning than steering on value Small delays more impact with agile than waterfall Time pressure for business value Waterfall helped with planning and risk management

Table A3.14 Codes related to planning

From the interviews it became clear that planning is difficult, especially in the long term, given the short-cycled development in agility which only gives clarity over the next 2 PIs (about six months). However, given the portfolio of the organization to implement legislation at a certain date, there is a strong focus on deadlines and time pressure to deliver business value, even though this becomes more challenging given the big size of projects. Planning furthermore becomes more difficult when there are changes in staff capacity, when staff members are inexperienced or sick, or when there are dependencies in interfaces. The presence of **technical debt** also influences the planning so this should be **measured and included in the planning**. Furthermore, because the development is so short-cycled, small delays can have a lot of impact. To increase control over the planning capability, risks should be brought to the forefront of the

project where **interfaces should be identified and planned** when possible, even though the exact content of the task is not yet known. Taking this in mind, one interviewee mentioned how waterfall used to be better with risk management and planning, regardless whether it delivered better content. Furthermore, planning becomes easier when tasks are narrowed down and have clear WSJF values as agility prescribes but the **sprint length** should also fit to ensure small delays do not bring substantial delays. Agility also indicates that one can and likely should divert from the initial planning. When dealing with deadlines regarding legislation the general mechanisms agility offers should be used: an overview of features which create the **MVP** should be included in the program backlog along with the required function points. In large projects creation of a **road map** can help to indicate which parts should be ready when. When placed against the performance of development teams an estimation can be made on how many sprints are required to reach the MVP. Agility also asks for **planning of volatility** given unexpected changes. The many dependencies with interfaces in public sector bureaucracies only increase the need to not make planning too tight. Furthermore, a finding from the theory should be used when it comes to **survey data** since there is always a chance that data does not represent all the information available to judge whether a project meets a deadline or not while in the beginning of a project or in the event of team changes metrics are not always that trustworthy. Taking these aspects into account one can make an estimation on whether a deadline is met with. When it becomes clear that this will not happen more **team (members) can be added** to the project even though it will take a couple months to get them up to speed.

> *"A year has 4 PIs. If you ask… say something about a PI that is four PIs away, whatever I say I know almost for sure that it will not be true… where we will be… end of PI 1 is clear, and what moments there are with issues, that we are not in synch with other interfaces in planning we all know that. PI 2 we know what we have to do and which interfaces we have but we do not have an image or other teams we can get involved in that, PI 3 becomes unclear." (Program level employee)*

> *"Now you have sprints of two weeks and you want to have things finished at the end of the sprint so the things you deliver are smaller, but they still need to go through the whole process including user acceptation tests so there are many phases it has to go through. You didn't use to have that with waterfall where you had 9 months' time to get the whole chunk through the chain and if you then have to wait a day or two then who cares, you have 9 months." (Team level employee)*

"You also have development trajectories and surroundings, there a lot of changes are put through, so also from the infrastructural group we frequently get invites that something has to happen. The past year that has not been managed well which made a lot unexpected and pushed forward which lead to a pile of it which makes it seem like there is a lot of technical debt to be solved and functionality is pushed back" (Program level employee)

Keep team membership stable					
Technical	Performance	Strategic	Interactive	Constraining	Reward

Calculate implications technical debt, changes in capacity, and general volatility into the planning					
Technical	Performance	Operational	Interactive	Constraining	-

Identify risks and interfaces at the forefront and plan in advance					
Technical	Performance	Strategic	Interactive	Constraining	-

Smaller tasks with clear WSJF value make planning easier and more predictable					
Technical	Performance	Operational	Interactive	Constraining	-

Add team (staff) when it appears deadlines are not met with					
Technical	Performance	Strategic	Diagnostic	Enabling	Reward

Use survey data next to metrics					
Technical	Performance	Operational	Interactive	Constraining	-

Have a sprint length of 30-45 days					
Technical	Performance	Operational	Interactive	Constraining	-

Use a road map and MVP for planning					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Portfolio and budget

Increasing control	*Losing control*
Agreement interface business case	Budgets designated for one year
Awareness need to change portfolio	Business works in projects not in visions
Business case	Continuity old system and agile renewal too
Departments fit financing to their preferences	much for management
Financing on train or ruling	Financing authority shift to business
	Inefficient budgeting
	Portfolio budget per assignment
	Portfolio does not support agile
	Traditional portfolio preventing vision

Table A3.15 Codes related to portfolio and budget

The portfolio and budgeting are currently performed in a traditional manner which does not support agility as it is focused on separate tasks and one-year timeframe. This also prevents a long-term vision from developing since there is no security when it comes to budget and furthermore hampers the stability of teams and therefore the predictability. Focusing the **financing on the train** in which **multi-year budgets** are committed increases control in this aspect, while within the organizations different department shape the financing system to their preference. The development of a **business case** also assists in a new view on financing next to its aid in creating a vision, although it is important that such a business case is shared and agreed upon by important stakeholders or interfaces.

"You introduced a methodology that is opposite to the linear portfolio of the organization' (Program level employee)

"It works on the basis of what they call portfolio finance. So instead of looking what is your vision, what is your roadmap, how many people do we need because you say we have a big team, that is the money we need and this is the project for two years. But now it is like we have money for piece A, then for piece B, that one wants D, we do not have C yet. They try to get money from everything, that is a bit the old management structure." (Team level employee)

Change the portfolio set-up to facilitate agility (finance on train level, multi-year, ensure team stability)					
Technical	Boundary	Strategic	Interactive	Enabling	-

Define a complete business case and road map (stakeholders, goals, budgets and specifications)					
Technical	Performance	Strategic	Interactive	Constraining	-

Avoid pre-project fixed budgeting (especially when specifications are unknown)					
Technical	Boundary	Strategic	Diagnostic	Enabling	-

Specification of requirements and prioritization

Increasing control	*Losing control*
Agile events prioritizing	Business not used to specify requirements
Continuous change prioritization	Complexity specifications
Downsizing and structuring specifications	Delivery pieces too big for sprints seem like no progress
Hierarchical specification	Differences in requirements specifications team
Just begin	Lack of roles for specifications
Specific roles for preparing specifications	Lack of specification requirements
WSF for prioritizing	No decision-making on specification
	Product Owner focusing on the wrong things
	Searching for balance prepare and execute
	Spring planning difficult with vague specifications
	Technical debt
	Vagueness specification

Table A3.16 Codes related to specification of requirements and prioritization

Agility places rather different expectations at the business when it comes to specifying requirements for the to be developed products. Rather than a big design up front, the specification is provided in smaller chunks, especially when working with Scrum principles.

Epics, features and user stories need **clear hierarchical definitions** to guide the specification process and provide workable input for teams. The case illustrated that this specification is so complex that typical SAFe roles such as Product Managers and Product Owners could not do this specification by themselves given the complexity. Therefore, **special specification teams** were created to translate the epics to features and user stories. There is, however, still disagreement on when an epic, feature or user story is specific enough while this debate still continues. To illustrate the importance of clear epics, features and user stories, the introduction of agility in one application shows a suitable example: at the start of the project, specifications were lacking which lead to functional and technical debt the teams are still coping with. The introduction of a specification team assisted in smoothening the specification process, along with downsizing and structuring the requirements. Furthermore, first specification phase should not go on for more than one PI session

"You see that the complexity in the functional side is heavier which means you have to have a lot of preparation before you let your scrum teams work" (Program level employee)
"I think there is something going on. For example, there are Epics but if you really look at it, many of those Epics we would now say those were actually features, so what was the real Epic? Well that is quite a searching process to get the levels right from Epics to Features" (Program level employee)

"[On complexity specification] We want to do it well really fast, but it is a totally different way of working" (Program level employee)

Have clear hierarchical definitions of what epics, features and user stories are and let their specifications flow hierarchically into each other					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Have a separate team for specification if it is too complex or time-consuming for the PM, PO and teams					
Technical	Performance	Strategic & Operational	Interactive	Enabling	-

The first specification preparation phase should not go on for more than one PI, just begin					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

A vision is important for prioritization					
Technical	Performance	Strategic	Interactive	Constraining	-

Teams, roles and responsibilities

Increasing control	*Losing control*
Agile coaches	Agile organization reverse pyramid
Agile Triangle in Train	Agile places higher demands on staff
Agreement on who puts what on the Product Backlog	Cannot just add team members
Clear Product Management and Business Owner	Changing positions
Clear responsibilities and roles	Changing team membership
Increasing capacity	Demographic characteristics organization
Investing in staff and ICT knowledge	Development teams not ready for agile
Involving Subject Matter Experts and users	Disconnection head of Product Management and PM for teams
Keep Scrum teams around 9 members	Double roles
Keeping team membership stable	External staff not knowing implications organizational characteristics
Product Management more involved	Knowledge not captures
Product Management with technical and business knowledge	Lack of capacity
Product Owner with technical background	Lack of capacity creates technical debt
Separate Scrum master in realization teams	Lack of job security and task certainty
Separate Scrum Master in specification team	Lack of regularity in roles
Specific roles for preparing specifications	Lack of specification on role
Specification team	Little staff and time
	Management in beginning not involved
	Management not understanding implications changing team membership
	Management switching positions
	Mismatch old and new roles

	No clear decision-making authority Non-dedicated teams Part-time Product Management Product Management not independent enough given lack of business Product Owner focusing on the wrong things Product Owner more managing on planning than steering on value Product Owner not representing business Realization tams not multidisciplinary Requiring more quality staff Short term hiring not effective Small teams Sometimes inexperienced for roles Stepping out of your role Team instability Team members do not always fit Team membership not stable due to changes IT departments Team staff decreasing Teams not involved in retrieving new staff Tendency not to take decisions higher up Too many Business Owners involved in decision making Too many Product Managers Unclear how to use responsibilities in agile Unclear responsibilities Unhappy employees

Table A3.17 Codes related to teams, roles and responsibilities

The teams in the ARTs are rather small with around 5-6 team members while tasks are rather specialized. Teams membership was rather instable as changes were made which hampered the progress since every time the team needed to re-find its balance. **Stable team membership** is therefore a clear control mechanism which aids the agile process as it makes performance

more predictable which is required for planning. It is a general issue in the organization that team staff is decreasing while its demographic is rather old, which can become an issue the coming years. The water head structure, with more people in management than in development teams is also a worrying aspect to take into account when agility is supposed to "bring power to the teams". Given the troubles in the employment of team members and difficulty on the job market it is furthermore worrying that team members can be unhappy and worry about their task security. It is important that the organization changes its approach towards teams and **invest in capacity and skills**. Especially because the hiring of external staff is not particularly desired for short term given the time needed to train them, their lack of knowledge about the organization, and lack of future knowledge capturing. When possible, **internal staff** should therefore be used and in case only external is possible that **knowledge transferring** is facilitated.

The agile roles are mostly filled in, although the performance would benefit from **full-time** Product Owners, Product Managers and Scrum Masters. Furthermore, there is currently a lack of business affiliation and product ownership in the Product Owners and Product Managers roles since the required knowledge already was not present at the time of implementation of agility. In one of the ARTs, however, the Product Management strongly deviates from what theory prescribes as it consists of about thirteen persons with one relatively inexperienced Product Manager in the lead while it is preferred to have **one or two Product Managers** with vast experience and knowledge of the organization, to be developed product, and knowledge of the technology and business side of the program. This ensures that there is a single point of communication with the stakeholders and users which leads to the bundling of knowledge and that, furthermore, authority over the Program Backlog is concentrated. A recurring statement on roles which was also repeated was that certain roles, such as Product Owners and RTEs, are performed by previous project managers which have a tendency to fall back to their old roles which can hamper the team facilitating mindset and decentralization of agility while creating uncertainty. **Clear description of roles, tasks, accountability and responsibilities** are needed while the Scrum Master and RTE are supposed to ensure people act within their roles.

> *"Stable teams are very important and I see within the train, with [interface X] that they like to change that and very easily say we will completely reassemble all teams. Then you lose everything, you lose a part of your knowledge, your velocity, teams need to get used to each other again and you lost another three months" (Team level staff)*

"Who has responsibility for what? Is the business expected to show up or should we wait for an invite?" (Program level employee)

"We sat together with then people to discuss what the role is of an Epic Owner and a couple hours later we decided we have to look at what his good for supporting the Business Owner and let's say that is his role and that is it. It is not really tangible. Am I only the driver of the car or do I need to replace tires, put gas in it, wash it, what should I do? There is a lot of unclarity and everyone even thinks they know everything." (Program level employee)

"So it has to be better under control who has which responsibilities within the train between [two departments], there is a lot of fuzz on the line." (Program level employee)

Keep team membership stable					
Technical	Performance	Strategic	Interactive	Constraining	Reward

Prepare for the implications of the organization's demographics and job market conditions to reverse the trend of decreasing number of team member employees					
Technical	Performance	Strategic	Diagnostic	Enabling	-

Agile roles should be performed full-time to the extent possible while the persons' placed in their roles should fit the required knowledge, skill and position within the organization for that role.					
Technical	Performance	Strategic & Operational	Interactive	Constraining	Reward

Roles and their tasks and responsibilities should be made clear for the trains internal and external processes					
Technical	Performance	Operational	Interactive	Constraining	-

Invest in agile mindset, skills and capacity

Social & Technical	Performance	Strategic & Operational	Interactive	Enabling & Constraining	Reward

Have one or two Product Mangers per ART					
Technical	Performance	Operational	Interactive	Constraining	-

Ensure the use of internal staff and, if necessary, use external staff for longer periods of time					
Technical	Performance	Strategic	Interactive	Constraining	Reward

Uncertainty and learning

Increasing control	*Losing control*
Bring risks to the front Management does not want to be too innovative	Cannot keep up with technological innovation Complexity innovation and legacy Desired functionality not flexible Fear for failure Going for optimal not workable Guidelines over value Mindset not on short cycles Not clear what level of innovation is accepted when vision is lacking Political volatility Technical complexity Technical complexity legacy systems Uncertainty amount users Uncertainty requirements EU

Table A3.18 Codes related to uncertainty and learning

There is uncertainty present for the ARTs when it comes to political volatility, technical complexity and technological developments, while the International ART also faces uncertainty over EU requirements and the expected number of users. While this indicates agility could be an appropriate method given its short-cycled development and learning character, the

organization fails to learn to full potential as there is a fear for failure and the desired functionality is not flexible given its societal importance and link to implementation of legislation. Management does not want too innovative solutions and goes for optimal rather than workable solutions. Also the many guidelines present hamper the development of value and learning. An example from the interviews showed that it is important to have a **vision** to indicate what level of **innovation is tolerated** to prevent conflict, since two team members ended leaving a team as they could not work as innovative as they would wish.

"They [two team members who left] were too progressive. I also have to partly agree with management. You have to look at the project we are in. The project is not the place for developing a whole new platform to run your software package. They went a bit too much out of the project boundaries, so I can understand management… They widened the scope a bit too much. They could because there was no vision, so yes then you also have the freedom." (Team level employee)

"We will prepare for the new world, the new process will immediately go in the new world but sometimes some aspects have to go into the old world. That makes it so complex and that is why the progress goes quite slow. Also because we have to see how do we want it to be later on in the new world and how should it relate back to the old world? (Team level employee)

A vision is required to indicate the level of innovation which is desired and accepted					
Social	Performance	Strategic	Interactive	Constraining	-

Agile mindset: Go for workable solutions in short cycles					
Social	Performance	Operational	Interactive	Enabling	-

Agile mindset: Decrease the fear of failure					
Social	Performance	Operational	Interactive	Enabling	-

User and stakeholder involvement

Increasing control	*Losing control*

Users involved in PIs	Islands structures Lack of cooperation Lack of guidance business specification teams Lack of input stakeholders Lack of sense of 'us' Parties do not feel involved Too many Business Owners involved in decision making Uncertainty amount users Users at distance

Table A3.19 Codes related to user and stakeholder involvement

From the interviews it becomes clear that the inclusion of users at **inclusive events** provides helpful input and increases cooperation. Yet, there generally is a lack of cooperation which indicates the need to **overcome sectarianism** and parties do not always feel involved or invited. Together this also created the situation there is not really a '**sense of us'** within the organization. This also leads to a lack of input from stakeholders which can be important for development of the system. The users are also at a distance since they are distributed over different locations in the Netherlands, but recent efforts to include them more by being physically present at important PI events helps in receiving valuable feedback.

> *"Within the train you have [two departments] and then you need to agree together. And I notice, [department A] is in the lead but I notice that [department B] is not always taken along. Sometimes they don't have to be, but sometimes it is very useful but then [department B] is left out. And then you go after it and hear that your neighbor is developing a part of the application with which makes you think shouldn't we have been involved? This has happened before… That is something we can be better in. It is also understandable, some people work here for 30 40 years on their island and are then asked to look to their neighbors and involve them in their story." (Team level employee)*

Island structures should be overcome for successful user and stakeholder involvement					
Social	Performance	Strategic	Interactive	Enabling	-

Users and stakeholders should be invited to events, which should be accessible to have them physically present					
Technical	Performance	Operational	Interactive	Enabling	Reward

Framing to increase the 'sense of us'					
Social	Performance	Strategic & Operational	Interactive	Enabling	Reward

Vision

Increasing control	*Losing control*
Agreement interface business case	Business works in projects not in visions
Creating vision roadmap and architecture	Complexity specifications
Management does not want too innovative	Deadline over value
Program plan	Deviation from SAFe brings grey areas and more work
Small business case per Epic	How to create architectural vision
Team makes own architectural template	Keeping track of overall architecture
Team taking the lead	Lack of vision
	Lack of vision how to use agile
	Mismatch deadline projects and architecture visions interfaces
	Not clear what level of innovation is accepted when vision is lacking
	Organizational complexity prevents organizational vision
	Tendency not to take decisions higher up

Table A3.20 Codes related to vision

From the beginning there was a lack of vision on the products that were going to be developed exactly, which was also hampered by the complexity of specification, uncertainty and a tendency not to commit and take decisions high up in management. The current portfolio set-up also does not promote the forming of visions since there is a focus on separate short projects. Together with the presence of deadlines, it is therefore also not always possible to create or implement a vision since projects have to deviate from the desired overall architecture to

achieve the desired functionality on time. The lack of vision increased the complexity of the specification process since there was a lack of strategic guidance. This also led to a situation where teams come up with their own architectural template. It appeared that the process of creating a vision was kept rather separated from the team members, while these have extensive knowledge of the systems. The new developments with program plan, business case, charcoal sketch and road map contribute in creating a vision and sharing this with relevant parties. These relate **to process and architectural visions.** This also relates to the organization at large to define a vision on which **legacy systems** should be transformed or even replaced over time. These visions are required for two reasons: firstly, it can enhance public service provision efficiency and fast adaptiveness over a longer period of time and secondly it is required for guidance on **prioritization**. Agility means adaptiveness but especially in the face of deadlines there should be prioritization when it comes to which services or application need to be ready on time. Given the complexity of **interfaces** it is also likely that a certain update or renewal is dependent on developments in another segment of the organization. If the vision and thus prioritization does not keep such interactions into account the prioritization will fall behind technical demands. The vision should therefore include both **business and technology** developments.

From the interviews it also became clear there are different perceptions on how agility is or should be implemented within the organization. Also taking a statement into account that **roles and responsibilities** are not always clear, which lead to delays and miscommunication, and that sometimes deviation "from the book" is required and sometimes not, it can prove helpful to also create a vision document **to specify how agility** and the used methodologies or frameworks, such as SAFe, are exactly implemented within the two ARTs. Also when implementation of agility throughout the organization might be desired full on implementation of agility might not be. Especially in the public sector certain services need to keep running and performing for the public good. A potential **transition** to agility should therefore also be clarified. While this might not be expected for the whole organization, departments that start with agility can create such a vision to illustrate what the implications of agility are for the interaction with their interfaces.

"With that [creating vision] I also say that I will not do a certain thing and you notice we are very careful with that. Often also with making it more specific, SMART, you have to keep to that then. You can see that we find that very difficult. Especially when you do not know what you want." (Program level employee)

"In the past you could not create vision, then there was money for monitoring, then for financial accountability, well what you saw a lot in projects, and still can, is that then all of a sudden building teams work on financial accountability, but is that then really the most important within the vision? No, but because you do not have a vision it goes from one thing to the other." (Team level employee)

Creation and promotion of a vision is required to create a common goal and aid the specification process					
Social and technical	Performance	Strategic	Interactive	Constraining	-

Balance between adaptiveness and stability					
Social & Technical	Boundary & Performance	Strategic & Operational	Interactive	Constraining	-

Combine the drive for agility with business incentives					
Social & Technical	Boundary & Performance	Strategic	Interactive	Enabling	-

A document elaborating on how agility is implemented can clarify issues related to roles, tasks and responsibilities					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Waterfall and agility

Increasing control	*Losing control*
Agile planning on non-SAFe teams Do not enforce agile on waterfall show benefits Escalation when agreements are not met with Sometimes need linear action	Continuity old system and agile renewal too much for management High decision-making structures do not fit agile Tendency for waterfall

Traditional structure in place given lack of trust in functioning SAFe and new structure Traditional structures aid in finding partners	Traditional decision-making structures Traditional portfolio prevents vision Traditional reporting does not fit agile Waterfall helped with panning and risk management

Table A3.21 Codes related to waterfall and agility

The organization and specifically this department do not fully function according to agility and SAFe. There are traditional structures present in the organization such as other project teams, management layers, the chain tables and even a Prince2 Project Board in one of the ARTs. One interviewee mentioned these are still in place given a lack of trust of the ARTs functioning well enough without further management interventions, which also portrays itself through the **traditional escalation mechanism** in case agreements are not met with. In general, there is still a waterfall mindset at the top and one respondent involved in these traditional structures mentioned most of the participants often do not even know the basics of SAFe and agility. While certain interviewees framed the influence of these traditional structures as negative, one respondent stated that these can also help with clearing up responsibility issues. Next to these traditional decision-making structures, the traditional portfolio management does not support agility and hampers the creation of a vision. As mentioned in the paragraph on planning, waterfall was better at bringing risks to the forefront along with generic **linear planning** which can be used to plan interface interactions more on time. Certain aspects of agility can also be used by waterfall teams as stated by the RTE of the local train, such as the **agile planning events**, although this should not be forced upon such teams. In general, a recurrent element was that interaction with waterfall structures is very present and will likely persist given the rooted bureaucracy within the organization. Besides, a complete immediate turn to agility is even undesired since agility requires time and care to be implemented next to the presence of deadlines and legacy systems. In the meantime control over the agile process and the performance of the department and organization at large requires a **balancing act** between agility and waterfall. This includes: combining the interfaces between agile and non-agile teams into one planning, indicating which teams and departments work agile or will start to given the required functionality the applications are supposed to provide, and the extent to which innovation and dynamic learning by failure is tolerated to ensure harmony between interfaces. Furthermore, the interviews indicated that **separate management** of agile and waterfall teams is required since the shift to agility is quite a demanding one regarding time and mindset.

"Because both Prince2 and the traditional structure have difficulties with SAFe agile not giving answers to how to deliver the realization of legislation. Legislation is linear… We cannot tell our director or the minister they should go to the Second Chamber to tell them my agile train took longer… Legislation is also that you have to implement it as a whole, you do not have a choice." (Program level employee).

"Waterfall was not all that bad, but then I mean the fact that you made a rough planning and had to give a place to your risks" (Team level staff)

"It is a lot and within [the department] we noticed that it [management of both agile and waterfall teams] is too much, if you want to be fast you need the focus and the concentration and time." (Program level employee)

Traditional structures and escalation mechanisms can exist next to agile ones as long as division of responsibilities and tasks is clear					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Change the portfolio set-up to facilitate agility (finance on train level)					
Technical	Performance	Strategic	Interactive	Constraining	-

Stimulate the agile mindset at the top of the organization					
Social	Performance	Strategic	Interactive	Enabling	-

Show non-agile teams the benefits of agile planning					
Technical	Performance	Operational	Diagnostic	Constraining	-

Cooperation and linear agreements with interfaces on anticipated changes, prioritization and eliminating delays					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Keep the management of agile and waterfall teams separate					
Technical	Performance	Strategic	Interactive	Constraining	-

3.4 Overview control mechanisms

This paragraph provides an overview of the mechanisms along with their characteristics according to the Revised Levers of Control Framework. They are categorized to their clusters and whether they are strategic or operational in nature.

Strategic

Agile mindset

Invest in agile mindset, skills and capacity					
Social & Technical	Performance	Strategic & Operational	Interactive	Enabling & Constraining	Reward

Agile leadership: Facilitate the teams					
Social	Performance	Strategic	Interactive	Enabling	Reward

Agile skills: Train staff					
Social & Technical	Performance & Boundary	Strategic & Operational	Interactive	Enabling & Constraining	Reward

Interfaces

Cooperation and agreements with interfaces on anticipated changes, prioritization and eliminating delays					
Technical	Performance	Strategic	Interactive	Constraining	-

Ensure a fair process of interaction between departments in which core interests are protected					
Social & Technical	Performance	Strategic & Operational	Interactive	Constraining	Reward

Metrics, tools and technology

Change guidelines to fit agility

Technical	Boundary	Strategic	Diagnostic	Constraining	-

Organization

Ensure the required knowledge and skills for the agile roles are placed in the right departments before and during implementation					
Technical	Performance	Strategic	Interactive	Constraining	Reward

Prevent agile water head (more management than team members)					
Technical	Performance	Strategic	Interactive	Constraining	-

Organizational change should work towards reflecting value delivery streams					
Technical	Performance	Strategic	Diagnostic	Constraining	-

Prepare for the implications of the organization's demographics and job market conditions to reverse the trend of decreasing number of team member employees					
Technical	Performance	Strategic	Diagnostic	Enabling	-

Planning

Keep team membership stable					
Technical	Performance	Strategic	Interactive	Constraining	Reward

Identify risks and interfaces at the forefront and plan in advance					
Technical	Performance	Strategic	Interactive	Constraining	-

Add team (staff) when it appears deadlines are not met with					
Technical	Performance	Strategic	Diagnostic	Enabling	Reward

Use a road map and MVP for planning					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Portfolio

Change the portfolio set-up to facilitate agility (finance on train level, multi-year, ensure team stability)					
Technical	Boundary	Strategic	Interactive	Enabling	-

Avoid pre-project fixed budgeting (especially when specifications are unknown)					
Technical	Boundary	Strategic	Diagnostic	Enabling	-

Roles

Agile roles should be performed full-time to the extent possible while the persons' placed in their roles should fit the required knowledge, skill and position within the organization for that role.					
Technical	Performance	Strategic & Operational	Interactive	Constraining	Reward

Specification

Have guidance from specifications in place from the beginning to steer creation of value					
Technical	Performance	Strategic	Interactive	Constraining	-

Have clear hierarchical definitions of what epics, features and user stories are and let their specifications flow hierarchically into each other					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Have a separate team for specification if it is too complex or time-consuming for the PM, PO and teams					
Technical	Performance	Strategic & Operational	Interactive	Enabling	-

The first specification preparation phase should not go on for more than one PI, just begin					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Teams

Ensure the use of internal staff and, if necessary, use external staff for longer periods of time					
Technical	Performance	Strategic	Interactive	Constraining	Reward

User and stakeholder involvement

Consider the different projects and activities within the ARTs as a multi-issue game					
Social	Performance	Strategic	Interactive	Constraining	-

Create a common perception shared by stakeholders on the problem and definitions at hand (negotiated knowledge)					
Technical	Performance	Strategic	Interactive	Constraining	-

Island structures should be overcome for successful user and stakeholder involvement					
Social	Performance	Strategic	Interactive	Enabling	-

Framing to increase the 'sense of us'					
Social	Performance	Strategic & Operational	Interactive	Enabling	Reward

Vision

Have basic architecture (vision) in place early on					
Technical	Performance	Strategic	Diagnostic	Constraining	-

Define a complete business case and road map (stakeholders, goals, budgets and specifications)					
Technical	Performance	Strategic	Interactive	Constraining	-

A vision is important for prioritization					
Technical	Performance	Strategic	Interactive	Constraining	-

A vision is required to indicate the level of innovation which is desired and accepted					
Social	Performance	Strategic	Interactive	Constraining	-

Creation and promotion of a vision is required to create a common goal and aid the specification process					
Social and technical	Performance	Strategic	Interactive	Constraining	-

Balance between adaptiveness and stability					
Social & Technical	Boundary & Performance	Strategic & Operational	Interactive	Constraining	-

Combine the drive for agility with business incentives					
Social & Technical	Boundary & Performance	Strategic	Interactive	Enabling	-

A document elaborating on how agility is implemented can clarify issues related to roles, tasks and responsibilities					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Waterfall

Traditional structures and escalation mechanisms can exist next to agile ones as long as division of responsibilities and tasks is clear					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Cooperation and linear agreements with interfaces on anticipated changes, prioritization and eliminating delays					
Technical	Performance	Strategic & Operational	Interactive	Constraining	-

Keep the management of agile and waterfall teams separate					
Technical	Performance	Strategic	Interactive	Constraining	-

Operational

Agile mindset, skills and leadership

Agile mindset: Just begin					
Social	Performance	Operational	Interactive	Enabling	-

Agile mindset: Better to commit and change later on than not commit at all					
Social	Performance	Operational	Interactive	Enabling	-

Agile mindset: Go for workable solutions in short cycles					
Social	Performance	Operational	Interactive	Constraining	-

Agile mindset: Decrease the fear of failure					
Social	Performance	Operational	Interactive	Enabling	-

Agile leadership: Create a positive environment of trust, motivate, promote accountability					
Social	Performance	Operational	Interactive	Enabling	Reward

Continuous change

Continuous adaptation of tasks and prioritization					
Technical	Performance	Operational	Interactive	Enabling	-

Cooperation

Realization and specification teams working in the same space					
Technical	Performance	Operational	Interactive	Constraining	Reward

Enhance the agile workspace					
Technical	Performance	Operational	Interactive	Enabling	Reward

Communication

Create a shared language					
Social & Technical	Performance	Operational	Interactive	Constraining	-

Open communication and transparency					
Social & Technical	Performance	Strategic & Operational	Interactive	Enabling	Reward

Events

Agile events					
Technical	Performance	Operational	Interactive	Constraining	-

All relevant parties should be invited					
Technical	Performance	Operational	Interactive	Enabling	Reward

Users and stakeholders should be invited to events, which should be accessible to have them physically present					
Technical	Performance	Operational	Interactive	Enabling	Reward

Planning of agile events is done by the RTE and is primarily team-facilitating					
Technical	Performance	Operational	Interactive	Constraining	-

Ensure relevant team members want to participate in the agile events					
Social	Performance	Operational	Interactive	Constraining	Reward

Have a sprint length of 30-45 days					
Technical	Performance	Operational	Interactive	Constraining	-

The strict procedures should leave room for crucial discussions					
Technical	Boundary	Operational	Diagnostic	Enabling	-

Interfaces

Use interfaces for flexibility and fast delivery					
Technical	Performance	Operational	Diagnostic & Interactive	Enabling	-

Metrics, technology and tools

Measure technical and functional debt and solve when required					
Technical	Performance	Operational	Interactive	Constraining	-

Use the right metrics (team facilitating and planning and flow oriented)					
Technical	Performance	Operational	Interactive	Constraining	-

Incorporate metrics for changes in capacity and technical debt					
Technical	Performance	Operational	Interactive	Constraining	-

Measure flow					
Technical	Performance	Operational	Interactive	Constraining	-

Use agile tooling (Jira and Confluence)					
Technical	Performance	Operational	Interactive	Constraining	-

Use WSJF for prioritization					
Technical	Performance	Operational	Interactive	Constraining	-

Use automatization tools for guidelines					
Technical	Boundary	Operational	Interactive	Constraining	-

Generic systems should be developed to be easily integrated					
Technical	Performance	Operational	Diagnostic	Constraining	-

Use survey data next to metrics					
Technical	Performance	Operational	Interactive	Constraining	-

Planning

Ensure teams have similar cadence					
Technical	Performance	Operational	Interactive	Constraining	-

Have a sprint length of 30-45 days					
Technical	Performance	Operational	Interactive	Constraining	-

Calculate implications technical debt, changes in capacity, and general volatility into the planning					
Technical	Performance	Operational	Interactive	Constraining	-

Show non-agile teams the benefits of agile planning					
Technical	Performance	Operational	Diagnostic	Constraining	-

Smaller tasks with clear WSJF value make planning easier and more predictable					
Technical	Performance	Operational	Interactive	Constraining	-

Roles

Have one or two Product Mangers per ART					
Technical	Performance	Operational	Interactive	Constraining	-

Specification

Smaller tasks with clear WSJF value make planning easier and more predictable					
Technical	Performance	Operational	Interactive	Constraining	-

3.4 Reflection on mechanisms

Synergy agility and waterfall: Planning interfaces

From the case it could be seen that when agility was implemented, the short-cycled mindset created situations where relatively small aspects, such as changes in capacity or delays from interfaces could have rather big consequences. The planning mindset changed even though the dependencies on interfaces only increased, while these themselves struggled with capacity issues and/or multiple requests for tasks. Therefore, **linear planning** of interface dependencies, which is generally more associated with waterfall development, can be very valuable in agile projects with many **interfaces** to **prevent unexpected delays.**

Challenge in the value chain

The case at hand had further difficulties with controlling the delivery of value as the **client-to-client process was cut up and distributed over multiple organizational departments** and systems, while the responsibility for the value chain was placed with the department in charge of the ART. This increased the complexity of development given the dependencies on interfaces and increase in required input from users and stakeholders.

Agility requires time

When considering control over the agile process in the case study it became clear there is a distinction in the desire for control between team level staff and higher management. While the former mainly stated the desire for control originated from the desire to enhance performance and enhancing flow and predictability, the latter related it to accountability and delivering deadlines. The desire for control of higher management therefore opposes the agile mindset of facilitating teams, enhancing flow and opting for value driven development. It is important to start implementing agility on time if there is a deadline at hand, as can hypothetically facilitate retrieving an agile mindset. Furthermore, choosing to implement agility to deliver fast without the appropriate mechanisms in place is very likely to result in the creation of technical and functional debt which at a later stage would again need to be solved, delaying the future development of the agile process and the product itself.

Mechanisms not typically associated with agility

Certain mechanisms identified were neither related to management concepts or theories discussed in chapter 3 or part of agility itself. The **implications of the organization itself**, whether the required knowledge and skills are based in the right departments, there is no water head structure with more people in management and teams, and that there are mechanisms to care for the island structures and the lack of 'sense of us', was prominent in the case study. The **dependencies and impact of interfaces** on the process, which require agreements and more linear planning, was also something that was not identified in the theoretical study. While literature on scrum mentions the characteristics of the **requirements specification**, there is barely attention for the actual **complexity** of this process: **defining** what epics, stories and features are and then actually **creating** them. Related to this point there is the paradox within agility some sort of **vision on the to be developed product is** needed beforehand, while most is learned during the process, with the former becoming increasingly important when there are deadlines and low tolerance of failure. A **vision on how agility is implemented** in which the

events, roles and responsibilities are clarified. A lack of both types of visions leads to grey areas and uncertainty and increases the likelihood of undesired performance. Lastly, agility in the public sector, given the financing structure, deadlines and presence of complex legacy systems, comes with a strong presence of **technical debt**, which is generally unrecognized in the field even though it has grave impacts on teams' ability to create business value. It should therefore also be measured and calculated into the team performance.

Bibliography

Azanha, A., Argoud, A. R. T. T., Camargo Junior, J. B. d., & Antoniolli, P. D. (2017). Agile project management with scrum: A case study of a brazilian pharmaceutical company it project. *International Journal of Managing Projects in Business, 10*(1), 121-142.

Chin, G. (2004). *Agile project management: how to succeed in the face of changing project requirements*: AMACOM/American Management Association.

Creswell, J. W. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*: Sage publications.

Crowder, J. A., & Friess, S. (2015). *Agile project management: managing for success*: Springer.

De Bruijn, H., & Ten Heuvelhof, E. (2008). Management in Networks. On Multi-actor Decision Making. In: London: Routledge.

De Bruijn, H., Ten Heuvelhof, E., & In 't Veld, R. (2010). *Process Management: Why Project Management Fails in Complex Decision Making Processes*: Springer Science & Business Media.

Dym, C. L., Little, P., Orwin, E. J., & Spjut, E. (2009). *Engineering design: A project-based introduction*: John Wiley and sons.

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software, 33*(3), 94-100.

Folke, C., Hahn, T., Olsson, P., & Norberg, J. (2005). Adaptive governance of social-ecological systems. *Annu. Rev. Environ. Resour., 30*, 441-473.

Forsgren, N., & Kersten, M. (2018). DevOps metrics. *Communications of the ACM, 61*(4), 44-48.

Gunderson, L. H., & Holling, C. S. (2002). *Panarchy: understanding transformations in human and natural systems*. Washington D.C.: Island press.

Janse, D., & Bogers, M. (2013). *Getting Teams Done: Een praktische methode voor teamproductiviteit*: Academic Service.

Janssen, M., Chun, S. A., & Gil-Garcia, J. R. (2009). Building the next generation of digital government infrastructures. *Government Information Quarterly, 26*(2), 233-237.

Janssen, M., & Van Der Voort, H. (2016). Adaptive governance: Towards a stable, accountable and responsive government. In: Elsevier.

Jullian, A. R., & Sangeetha, M. (2016). From Dev to Ops – Introduction to Devops on understanding Continuous Integration and Continuous Delivery. *International Journal*

of Innovative Research in Computer and Communication Engineering, 4(6), 12567-12572.

Junior, M. L., & Godinho Filho, M. (2010). Variations of the kanban system: Literature review and classification. *International Journal of Production Economics, 125*(1), 13-21.

Kim, G., Debois, P., Willis, J., & Humble, J. (2016). *The DevOps handbook: how to create world-class agility, reliability, and security in technology organizations*: IT Revolution.

Klievink, B., Bharosa, N., & Tan, Y.-H. (2016). The collaborative realization of public values and business goals: Governance and infrastructure of public–private information platforms. *Government Information Quarterly, 33*(1), 67-79.

Klievink, B., & Janssen, M. (2014). Developing multi-layer information infrastructures: Advancing social innovation through public–private governance. *Information Systems Management, 31*(3), 240-249.

Laanti, M. (2014). *Characteristics and principles of scaled agile.* Paper presented at the International Conference on Agile Software Development.

Lappi, T., & Aaltonen, K. (2017). Project governance in public sector agile software projects. *International Journal of Managing Projects in Business, 10*(2), 263-294.

Luna, A., Kruchten, P., Pedrosa, M., Neto, H., & de Moura, H. (2014). State of the art of agile governance: a systematic review. arXiv preprint. *arXiv, 1411*.

Luna, A. J. d. O., Kruchten, P., & de Moura, H. P. (2015). Agile Governance Theory: conceptual development. *arXiv preprint arXiv:1505.06701*.

Qumer, A. (2007). *Defining an integrated agile governance for large agile software development environments.* Paper presented at the International Conference on Extreme Programming and Agile Processes in Software Engineering.

Robertson, B. J. (2006). Holacracy: A complete system for agile organizational governance and steering. *Agile Project Management Executive Report, 7*(7).

Robertson, B. J. (2007). Organization at the leading edge: Introducing Holacracy™. *Integral Leadership Review, 7*(3), 1-13.

Rubin, H. J., & Rubin, I. S. (2011). *Qualitative interviewing: The art of hearing data*: Sage.

Scaled Agile. (2017). SAFe 4.5 Introduction: Overview of the Scaled Agile Framework for Lean Enterprises. *A Scaled Agile, Inc. White Paper, August 2017*.

Scrum.org, & Vacanti, D. (2018). The Kanban Guide for Scrum Teams.

Soe, R.-M., & Drechsler, W. (2018). Agile local governments: Experimentation before implementation. *Government Information Quarterly, 35*(2), 323-335.

Stettina, C. J., & Hörz, J. (2015). Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management, 33*(1), 140-152.

Sutherland, J., & Schwaber, K. (2013). The Scrum guide. the definitive guide to Scrum: The rules of the game. *ScrumGuides. com*.

Sutherland, J., van Solingen, R., & Rustenburg, E. (2011). *The power of Scrum*: CreateSpace.

Talmy, S. (2010). Qualitative interviews in applied linguistics: From research instrument to social practice. *Annual Review of Applied Linguistics, 30*, 128-148.

The Economist. (2014). The Holes in Holacracy. *Business.* Retrieved on 4 March 2019 from https://www.economist.com/business/2014/07/05/the-holes-in-holacracy

Wang, C., Medaglia, R., & Zheng, L. (2018). Towards a typology of adaptive governance in the digital government context: The role of decision-making and accountability. *Government Information Quarterly*, *35*(2), 306-322.

WhatIs. (n.d.). Definition: holacracy. Retrieved on 4 March 2019 from https://whatis.techtarget.com/definition/holacracy