

Chapter 5

Delivering business value faster by sets of codependent Scrum teams: a governance framework

137



Context: Many enterprises that adopt Agile/Scrum suffer from collaboration issues between codependent Scrum teams that jointly deliver functionality for a value chain. These collaboration issues delay the delivery of functionality, deteriorating the business value in these value chains.

Objective: Develop a governance framework that packages empirically tested intervention actions that alleviates the collaboration issues in sets of codependent Scrum teams.

Method: The effectiveness of the intervention actions was validated in a large confirmatory case study with a set of codependent Scrum teams at a multi-national financial institute, by studying the qualitative effects in archival records and measuring the change in cycle time within a specific workflow application. The effectiveness of the intervention actions was triangulated in three focus groups with members that operate in the set of Scrum teams.

Findings: The intervention actions initiated a cycle time reduction from 29 days to 10 days. The participants in the focus groups confirmed the causality between the performance improvement of the set of codependent Scrum teams and the intervention actions.

Result: The main contribution of this chapter is a governance framework for sets of codependent Scrum teams that support a value chain.

5.1 Introduction

Large companies operating in the information intensive industries experience rapid changing business demands that require the swift adaption of the front to back (business) value chains. Since these value chains are automated with IT services, the rapid changing business demand requires flexible IT services. The IT services that enable these front to back value chains, are delivered by a portfolio of interdependent applications, That application portfolio is typically delivered by multiple codependent IT service providers (ISP). IT Service changes therefore often require software development staff of multiple ISPs (Plugge & Janssen, 2009; TFSC, 2011), to jointly execute the fast paced software development process which transcends ISPs (Moniruzzaman & Hossain, 2013; Pikkarainen et al., 2005).

In order to achieve a fast paced software development process, many internal IT development centers increasingly transfer to Agile methods. The most common Agile framework used in industry is the Scrum software development method (VersionOne, 2013). Scrum is an incremental method that uses low boundary cross-functional collaboration in software development teams that work toward a set team goal (Sutherland & Schwaber, 2013). Scrum works with fixed iterations shorter than one calendar month to deliver working and tested increments of working software.

139

Scrum teams can be mapped in different ways onto the (interdependent) application portfolio. Some prefer a single Scrum team for all interdependent applications that support the front to back value chain (Sutherland, 2005). However two constraints make such coverage difficult. Firstly, the amount of involved IT staff (typically from different ISPs) then easily exceeds the generally agreed upon maximum Scrum development team size of 9 members. Secondly, changes typically require highly specialized skills (due to a complex IT landscape with multiple Commercial-off-the-shelf items) that cannot be shared easily within every single team. The solution chosen in companies for the two constraints is setting up dedicated Scrum teams. Each Scrum team then develops one or more applications in the portfolio that automates a part of the front to back value chain (Vlietland & van Vliet, 2015b). The applications developed by multiple Scrum teams, together result in value-adding features. Features are defined as: 'intentional distinguishing characteristics of the application landscape that can be used by a business user' (IEEE, 2008), e.g. a mortgage registration feature.

As feature delivery is the output of multiple Scrum teams, collaboration is needed between the teams. Particularly the high frequency of deliveries which are common in Scrum settings makes collaboration a performance factor (Dorairaj et al., 2012). Yet, due to the nature of Scrum teams, such collaboration might not happen naturally. A Scrum development team has specific characteristics, such as a maximum of 9

members, a multidisciplinary setup, allocated IT applications, high-frequency deliveries and focus on a single product backlog (Sutherland & Schwaber, 2013). These characteristics typically limit the focus of a Scrum team, resulting in collaboration issues (Vlietland & van Vliet, 2015b).

Vlietland and van Vliet (2015b) identified six blocking issues in chains of codependent Scrum teams. The present study develops intervention actions (IAs) that alleviate the issues in a set of codependent Scrum teams that support a front to back value chain. The IAs are packaged into a governance framework. The IAs are validated in a large confirmatory case study with a set of codependent Scrum teams at a multinational financial institute. The case study had a timespan of approximately 9 months. After deploying the IAs the cycle time was reduced from 29 days to 10 days. The improvement effects of the IAs were triangulated with focus groups consisting of members operating in the set of codependent Scrum teams. These focus group confirmed that the cycle time significantly reduced as result of the IAs.

The remainder of this chapter is organized as follows. Section 5.2 covers the related work for developing the IAs and the governance framework. Section 5.3 explains the case study design with the research method. Section 5.4 elaborates on the empirical results. Section 5.5 discusses the results. Section 5.6 elaborates on the threats to validity. Section 5.7 concludes the study, deduces implications and suggests future research avenues.

5.2 Related work

Three areas of related work are studied. First an overview of organizational change literature is given to theoretically embed the IAs. Subsequently, an overview is given of the Agile IA literature. The section closes with related work about Agile governance frameworks. With that framework literature typical core-elements of Agile governance frameworks are identified. With these core-elements the coverage of the Agile framework that is developed in this chapter is validated.

5.2.1 *Organizational change literature*

Three perspectives on change in the organizational change literature are identified: (1) the tempo of change, (2) planned versus spontaneous change and (3) top-down versus bottom-up change. After introducing these three perspectives, a deeper analysis is performed on the combination of change perspectives that fit this case study, while introducing learning theory as catalyst for organizational change. The subsection closes with a summary of the change design for this case study.

Tempo of change: One perspective on organizational change is the tempo of change (Weick & Quinn, 1999). At one end of the spectrum is evolutionary change, which involves a relatively long stream of small changes as reaction to the changing environment, as first modeled by Darwin. Evolutionary change in organizations progresses continuously. Revolutionary change at the other end of the spectrum happens in short bursts (Hannan & Freeman, 1984). One theory in the area of revolutionary change is the theory of inertia and punctuated equilibrium (Romanelli & Tushman, 1994). In case an organization does not evolutionary follow the changing environment, the organization gets disconnected from the environment and tends to an inert equilibrium state (Gersick, 1991). An organization in that state is hard to change. After some time, strategic reorientation is required to realign the organization with the environment, resulting in a revolutionary change. For such revolutionary change the inert equilibrium needs to be punctuated. After the inertia is punctuated the organization experiences a turbulent change to find a new equilibrium closer aligned with the environment.

Planned versus spontaneous change: A related perspective to evolutionary and revolutionary change is planned versus spontaneous change. Spontaneous change occurs without a set purpose. Each individual actor interacts with other actors and the system changes through evolution (Stacey, 1995). At the other end there is planned change. The actors together aim to achieve a planned state.

Top-down versus bottom up: A perspective related to planned change is top-down versus bottom-up change. Yamakami (2013) analyzed organizational change initiatives in the IT industry and identifies three types of initiatives (1) top-down, in which top management takes initiative, (2) bottom-up, in which the work floor staff exercises own initiative to distribute change and (3) a hybrid approach.

Synthesis: Cummings and Worley (2014) elaborate on planned change as a way to change organizations. They identify two planned change strategies (1) a positivistic approach with an unfreezing, moving and freezing phase and an (2) interpretivistic approach with iterations and feedback loops (Jrad, Ahmed, & Sundaram, 2014). Positivistic based change paradigms have long dominated the IT industry, such as CMMI (Team, 2010a) and ISO 9000 (Hoyle, 2001). The positivist paradigm uses a machine metaphor in which input is transformed to output (Ilgen, Hollenbeck, Johnson, & Jundt, 2005; Stelzer & Mellis, 1998). The paradigm stimulated the use of detailed prescribed work processes which can be quantitatively measured, analyzed and controlled (Unterkalmsteiner et al., 2012). A positivistic approach works in areas of high predictability. The intrinsic human intensive activity of software development with high levels of unpredictability and uncertainty however seems a misfit with such a positivistic paradigm (Clarke & O'Connor, 2013). That misfit was answered in the

beginning of this century when the interpretivistic based Agile paradigm got momentum (Akbar et al., 2011). The Agile paradigm uses a bottom-up, continuous change paradigm to utilize human capital in the software development industry (Van Tiem, Karve, & Rosenzweig). Agile is supported with iterations and feedback loops to increase the evolutionary change capability (Qumer & Henderson-Sellers, 2008). Such iterative implementation approach is specified by R. L. Baskerville and Wood-Harper (1996) and R.L. Baskerville (1999). They specify cyclical action research based on the description of Susman and Evered (1978). Their research design consists of the five phases, which are repeatedly executed to allow adaptation of the change strategy during each cycle. The five phases are: diagnosing, action planning, action taking, evaluating and specifying learning.

Learning as catalyst: Experience-based learning can be seen as catalyst for organizational change in Agile environments. Kolb (1984) uses three models of experiential learning for developing a model that combines experience, perception, cognition and behavior. His resulting experience learning model consists of four phases: (1) concrete experience, (2) reflective observation, (3) abstract conceptualization and (4) active experimentation.

For continuous learning in Agile environments, one of the key principles is reflecting on past experience (Holz & Melnik, 2004; Salo & Abrahamsson, 2005). Such reflective practice exists in different development disciplines on individual, team and organizational level. For instance a Scrum team conducts a demo and notices that the Product owner struggles with a drag and drop action. Such observation offers the team to rethink the functionality and experiment another solution. Qumer and Henderson-Sellers (2008) argue similarly that agile knowledge engineering and management approach should be integrated with an agile software development approach and use it for performance improvement, learning and decision making in an agile software development environment.

Change design: This case study fits an evolutionary intervention strategy while having a planned objective. The objective enables us to design IAs in achieving that objective. Given the Agile characteristics it is expected that a hybrid, iterative change approach fits the purpose of the case study. The research design is further elaborated in section 5.4.

5.2.2 *Agile improvement intervention literature*

In this subsection the Agile performance improvement intervention literature is discussed, in the areas of the five collaboration related issues coordination, prioritization, alignment, automation and visibility (Vlietland & van Vliet, 2015b). The

scope of the discussed literature is limited to the literature that is used to design the IAs for this case study.

Coordination: The Scrum of Scrums is a Scrum practice to coordinate collaboration between Scrum teams. That practice comes with challenges. Paasivaara et al. (2012) identified that Scrum of Scrums works poorly in case of too many participants with disjoint interests. A way to further coordinate work is by using product teams. Schnitter and Mackert (2011) outline how Scrum was scaled with liaison relations between Scrum teams, by introducing product teams that are each responsible for up to seven Scrum teams. The characteristics of such a product team are that each member of the product team is a member of a Scrum team and that each product team bears full responsibility (time, cost and result). Kniberg and Ivarsson (2012) report the implementation of a two level structure combined with liaison relations between Scrum teams to coordinate collaboration, similar to a matrix organization. Scheerer et al. (2014) introduce a more conceptual multi-team system perspective with three types of coordination: (1) mechanistic coordination - with plans, rules and programming, (2) organic coordination - with mutual adjustment and feedback and (3) cognitive coordination - by means of similarity configuration. Product teams utilize such coordination, for instance by making plans and rules and responding to feedback (Vlietland & van Vliet, 2014b).

Prioritization: Another way to improve collaboration between Scrum teams is to prioritize the work over multiple Scrum teams (Christoph Johann Stettina & Hörz, 2015). Literature about priority matching between backlogs is scarce. Rautiainen et al. (2011) study the introduction of portfolio management to support scaled Agile development, by prioritizing all projects in a single backlog. Prioritization dramatically reduced the number of ongoing projects, enabling visibility about ongoing projects that assisted coordination. The product teams of Schnitter and Mackert (2011) with linked product owners of Scrum teams are one way to match backlogs of codependent Scrum teams. A way to determine which backlog items need to be prioritized over the Scrum teams is explained by Vlaanderen, Jansen, Brinkkemper, and Jaspers (2011). They introduce a Software product management (SPM) process for managing requirements, defining releases and defining products with many stakeholders.

Alignment: Literature about the alignment of Scrum teams is scarce as well. The literature study did not reveal literature that describes alignment interventions. Scheerer et al. (2014) embedded the alignment concept in coordination. Mechanistic Scrum team alignment can be achieved by implementing plans and rules similar to those promoted by Leffingwell (2007). Leffingwell (2007) promotes an aligned sprint heartbeat and mentions a define/build/test workflow for all teams. Organic and cognitive alignment is achieved with a shared mental model (Jonker et al., 2011; Lim &

Klein, 2006; Mathieu et al., 2000). Shared mental models are implemented by grouping people together and stimulate communication and feedback, such as with Scrum of Scrums practices. Mechanistic alignment focuses on executing prescribed alignment practices, while organic and cognitive alignment focuses on actually embedding these practices.

Visibility: For the visibility intervention literature the Agile and Supply Chain Management research areas were studied. Vacanti and Vallet (2014) explain the IAs at Siemens to shift from traditional Agile metrics to actionable flow metrics. Selecting and visualizing flow metrics opened the way to even greater Agility, improving the predictability and performance. The identified IAs are: (1) defining key goals with key metrics and (2) clearly visualizing these metrics, such as cycle times including predictions of future cycle times. Supply Chain visibility has (Scrum) value chain related characteristics. Banbury et al. (2010) explored the role of collaboration between teams by simulating a supply chain and studying the resulting bullwhip effect. The bullwhip effect results in productivity drop in a chain of suppliers, due to a combination of change in demand and a delayed response to that change (H. L. Lee et al., 1997). The results show that team focused groups need information about the current demand level in the supply chain to minimize the cost, back-orders and bullwhip size and maximize the delivery of orders. Bartlett et al. (2007) investigate the link between visibility and business performance by implementing enhanced visibility of plans, materials and inventory management. Vlietland and van Vliet (2014b) studied the effect of visibility of past performance information onto the actual performance of IT incident handling. Their case study revealed that such visibility has a positive effect on IT incident handling performance.

Automation: In the area of automation of IT processes, being information technology for information technology (IT4IT), literature was identified that describes implementation practices. No identified literature mentions a value chain supported by a set of codependent Scrum teams. Olsson et al. (2012) present a multiple case study on a move from traditional development to continuous delivery. They identified that during the implementation, collaboration and information exchange is poorly supported and old conservative technology restricts the automation of software development practices. Humble and Farley (2010) describe various practices for the implementation of continuous integration, testing and deployment, by focusing on the technical implementation aspects. Neely and Stolt (2013) report their experience with the implementation of continuous delivery practices. Their approach is to use an evolutionary change approach for gradually decreasing the delivery time with one step at the time.

5.2.3 Agile governance framework literature

This subsection provides an overview of the Agile governance framework (AGF) literature. The subsection starts with a definition of a governance framework. Subsequently the subsection identifies and discusses AGF core-elements in related work. These AGF core-elements are used in section 5.3 to validate whether the designed IAs of this study cover all core-elements of AGFs. The identified AGF core-elements are: (1) Role, (2) Event, (3) Team, (4) Artifact and (5) Lifecycle. The subsection is structured in order of these identified AGF core-elements.

Definition of governance: A. E. Brown and Grant (2005) classify governance as: *“Systematically determining who makes each type of decision (a decision right), who has input to a decision (an input right) and how these people (or groups) are held accountable for their role”*. They add that a framework should make clear: (1) who has decision making authority, (2) who provides input about a decision and (3) how these roles are jointly held accountable.

145

Role: According to that classification an AGF core-element are roles with clear responsibilities and authorities. The Scrum framework includes such core-element with three roles: the Product Owner, Scrum Master and other Scrum team members. A Product Owner acts as the single ‘voice of the customer’ collecting and prioritizing customer needs onto a prioritized list of items: the product backlog. The Scrum Master facilitates the Scrum team in achieving its goal. The Scrum team has the responsibility to develop software based on the Sprint Backlog (Rising & Janoff, 2000; Sutherland & Schwaber, 2013). Larman and Vodde (2013) introduce an area product owner as additional role in Agile development to coordinate multiple product owners.

Event: Sprint Planning, Daily Scrums and Sprint Review are team events of the Scrum method (Sutherland & Schwaber, 2013), that support self-organization (Moe et al., 2008). Larman and Vodde (2013) introduce an augmented framework for larger scale Agile development. The augmentation addresses coordination needs by additional events that support cross team coordination: (1) inter-team Sprint Planning meetings, (2) inter-team Daily Scrums, (3) inter-team Product Refinements and (4) inter-team Sprint Reviews. Events are identified as the second AGF core-element.

Team: The development team in Scrum has a small size (max 9). Ambler (2009) defines the (small) size of teams as an Agile scaling factor when Scrum is scaled. The small team size eases intra-team knowledge sharing and utilizes the self-organizing ability in professional teams (Takeuchi & Nonaka, 1986). Larman and Vodde (2013) use feature teams and liaison relations with Communities of Practice for exchanging knowledge and coordination between teams. Schnitter and Mackert (2011) identified product

teams (similar to feature teams) to manage the interdependencies between Scrum teams. Based on the related work small teams (up to 9 members) are identified as the third AGF core-element.

Artifact: Self-organizing practices within Scrum teams are supported by artifacts, such as a Product Backlog with the requirements of a product and a Sprint Backlog with items selected for a sprint (Sutherland & Schwaber, 2013). Leffingwell (2007) and Leffingwell (2010) promote a three level artifact structure consisting of stories, features and epics (cluster of features). Artifacts are therefore identified as the fourth AGF core-element.

Lifecycle: A Scrum development lifecycle normally consists of short (2-4 weeks) iterations, which enables swift feedback from software users and related stakeholders about the developed solution. Soundararajan and Arthur (2009) use two phases in their framework for large scale systems: (1) a generation process to gather requirements and (2) a scaling development process for large scale systems. Hence lifecycle is identified as the fifth AGF core-element.

5.3 Research Method

This section explains the setup of the confirmatory case study. The case study is performed in a large multi-national financial institute, delivering financial services to multinational business customers. The case entails a set of codependent Scrum teams that support a value chain. Each Scrum team needs each other's functionality as part of the whole solution offered to the value chain. The case study has five phases, following Runeson and Höst (2009): (1) Designing the case study and designing the interventions; (2) preparing for data collection; (3) collecting evidence; (4) analysis of collected data; and (5) reporting. This section is organized in that order.

5.3.1 Case study design

A confirmatory case study setup is selected (Easterbrook, Singer, Storey, & Damian, 2008) to test the impact of the IAs onto the cycle time of feature stories, delivered by the Scrum teams. One could argue that reusing existing (traditional) framework, such as CMMI or ITIL (Team, 2010a; van Bon et al., 2007) is the way forward. Agile/Scrum however is based on a philosophy that finds its roots in social constructionism and interpretivism science philosophies (Walsham, 1995). The intervention approach in this case study is aligned with that philosophy, using the perceived issues as departure point. Given the Agile philosophy (Akbar et al., 2011; Qumer & Henderson-Sellers, 2008) a planned, evolutionary intervention approach is chosen (Weick & Quinn, 1999).

The expectation was that a planned, evolutionary intervention approach was the best fit for achieving the planned change.

Each IA is top-down planned and initiated (Yamakami, 2013). The top-down initiation aims to break the existing equilibrium within the organization (Romanelli & Tushman, 1994). Each IA is designed in a way that multiple members are stimulated to iteratively adapt, refine and further deploy the IA, after the top-down initiation. The iterative cycles are stimulated by learning (Kolb, 1984), aiming to deeply embed the organizational change.

Archival records are studied to identify the sociological effects of the interventions onto the people operating in the set of codependent Scrum teams. These effects act as rationale for adapting and refining the IAs (Kolb, 1984). Focus group interviews at the end of the intervention period triangulate the effect of the IAs onto the cycle time.

Based on the scaling factors of Ambler (2009), selection criteria are defined for developing the applicable case study selection criteria, as shown in Table 19. The item between the brackets ‘(..)’ at the end of each description refers to the scaling factors.

Table 19, Case study selection criteria

Selection criterion	Selection criteria description
Application interdependencies	Applications have stable interdependencies with other applications in the front to back value chain (technical complexity, domain complexity).
Chain setup	Scrum teams support a front to back value chain and each application under development is allocated to one Scrum team (organizational distribution, organizational complexity, and technical complexity).
Application experience	Each Scrum team develops each of the allocated applications for at least 6 months (technical complexity, organizational complexity and enterprise discipline)
Team distribution	Studied Scrum team members are working in the same country (geographical distribution).
Regulatory requirements	Non user requirements exist that must be taken into account by the product owners (regulatory compliance)
Culture	Studied Scrum team members have the same nationality (organizational complexity)
Agile transition state	Each of the teams acts in a Scrum setup for at least 6 months (organizational complexity).
Workflow automation	Scrum teams already use a single database to manage the development workflow (organizational complexity).

The selection criteria enable the identification of the unique characteristics of a set of Scrum teams that support the front to back value chain and enhance the content validity of the research. Each of the Scrum teams needs to be experienced and work in accordance with Scrum framework to minimize research bias.

5.3.2 Intervention action design

Vlietland and van Vliet (2015b) identified six issues in chains of codependent Scrum teams: (1) mismatches in backlog priority between teams, (2) a lack of coordination in the chain, (3) alignment issues between teams, (4) a lack of IT chain process automation, (5) a lack of information visibility in the chain and (6) delivery unpredictability. This subsection describes the initial designed IAs for alleviating these issues, except unpredictability. Unpredictability directly impacts the cycle time of new features and is considered the dependent variable of the IAs, following Vlietland and van Vliet (2015b).

The IAs are designed based on the related work of section 5.2. To mitigate a lack of commitment for top-down IAs (Scheerer et al., 2014), top-down and bottom-up interventions actions are combined in a hybrid implementation approach, as identified by Yamakami (2013).

148

The related work of subsection 5.2.1 and 5.2.2 is used to predict the effect of the IAs. Each of the IAs impacts the IT workflow processes of the codependent Scrum teams. Each IA results in deployed 'items' that are indicated in **bold** (e.g. **feature description**). Subsection 5.2.3 identifies the five AGF core-elements. Each of the deployed items refers to these core-elements, to validate whether all core-elements are covered by the set of IAs. A reference to the core-element is indicated by bold brackets '<...>'. The IAs are categorized in accordance with the identified collaboration related issues of Vlietland and van Vliet (2015b)..

Issue 1: Prioritization

IA: *Multiple Scrum teams collaborate for jointly delivering added-value features. Each feature will be described in a **feature description <artifact>**. These feature descriptions are broken down in stories on the Product Backlog of each Scrum team that supports the value chain. Each feature description includes the added value and high-level effort estimation. A feature description consists of a functional feature description and a technical interaction design.*

IA: *The feature analysis and design activities incorporate many uncertainties and can therefore hardly be estimated within a sprint cycle. For this reason Soundararajan and Arthur (2009) is followed by defining two lifecycle phases: (1) a preparation phase that prepares features in the **Flow to Ready (F2R) <lifecycle>** and (2) an execution phase that realizes the features in the **Iterate to Done <lifecycle>**. Feature design and analysis activities take place in the F2R phase that takes 'N' weeks to accomplish (Vlaanderen et al., 2011).*

IA: Each feature is prioritized on the **feature backlog <artifact>** to match the story priority on the Product Backlog of each codependent Scrum team. Prioritization will be based on the added value and effort. Each feature is described by a feature description consisting of a functional feature description and a technical interaction design. The prioritization mechanism is similar to the mechanism of Rautiainen et al. (2011). Rautiainen et al. (2011) describes the prioritization of a portfolio of projects, while in this study a portfolio of features on a feature backlog is prioritized (see Figure 24, feature backlog and the matching arrows to the Scrum team backlogs). Unique feature priority likely also mitigates disjoint interests during Scrum of Scrums (Paasivaara et al., 2012).

IA: Next to the Scrum team Product Owners (PO) that already exist, **Feature Product Owners (FPO) <role>** will be allocated. A Feature Product Owner owns the functionality of a set of (front to back) features.

IA: An **Epic Product Owner (EPO) <role>** will be allocated, being accountable for the unique priority of each of the feature on the Feature Backlog.

IA: All three Product Owner types in scope of the codependent Scrum teams will be part of the **Product Owner Group (POG) <team>**. The POG together discusses and decides about the priority of each feature on the feature backlog. The group is headed by the Epic Product Owner.

IA: The Product Owner group will meet weekly during the **Epic Planning <event>**. Subgroups of Product Owners will meet regularly on an as needed basis to prepare the priority in the weekly meeting. These interacting groups and subgroups of product owners will enable the forming of a shared mental model (Jonker et al., 2011). It is reasonable to expect that such a shared mental model combined with a clear responsibility will stimulate matched priority setting.

Issue 2: Coordination

IA: **Product teams <team>** crossing the Scrum teams will be set up to coordinate the work between the Scrum teams, as outlined by Schnitter and Mackert (2011) and Kniberg and Ivarsson (2012). Product teams consist of product owners, IT architects, functional analysts and interface designers. A product team will be headed by a feature Product Owner. Typically multiple concurrent product teams exist.

A product team elaborates a feature into a feature description that can be broken down into stories. Product teams have similarities with the system teams

of Leffingwell (2010). The functional analysts and interface designers work part-time in their Scrum Team and part-time in their Product Team.

IA: Product teams will meet in **Bi-daily Features <event>** for sharing results, and discussing next actions and impediments. Compared to Kniberg and Ivarsson (2012) product teams focus on sprint preparation activities taking care of dependencies rather than managing such dependencies during the sprint.

IA: Feature Planning <event> meetings will be scheduled that precedes the sprint planning of the Scrum teams. During the feature planning meeting the elaborated features will be used by the Scrum teams for determining and estimating the team specific stories.

IA: A Scrum of Scrums <event> will be implemented to organically manage codedependencies between the Scrum teams. The Scrum of Scrums will be facilitated by a Scrum coach to secure the effectiveness of the meeting and prevent the issues as identified by (Paasivaara et al., 2012). The Scrum of Scrums will be executed weekly.

IA: Interface connectivity between two applications developed by different Scrum teams is enabled by middleware and interface-adapters. The middleware and adapters are developed by a third dedicated Scrum team. For each interface, therefore, three Scrum teams are involved. **Mini Scrums <team>** centered on interface connectivity will be setup with an analyst/designer from each Scrum team to develop the interface designs and coordinate the dependencies. These Mini Scrums mitigate the issue of disinterest in the Scrum of Scrums as identified by Paasivaara et al. (2012). The **Mini Scrum <event>** take place bi-daily to weekly, depending on the need. The Mini Scrums are facilitated by a Scrum Master (SM) of one of the Scrum teams.

IA: During the **Feature Review <event>** the functionality that was developed by the codependent set of Scrum teams is demonstrated. The Feature review will be scheduled by the Epic Product Owner and facilitated by the applicable Feature Product Owners.

IA: During the **Feature Retrospective <event>** a Product team will evaluate the sprint and plan improvements to be enacted during the next sprint.

Issue 3: Alignment

IA: A four week **Aligned Sprint Lifecycle <lifecycle>** duration will be institutionalized over all Scrum teams that support the value chain to make sure

that each team delivers stories within the same expected time-frame, being mechanistic alignment (Scheerer et al., 2014). The sprint duration will be institutionalized via the management team and then implemented via the Product Owners and Scrum Masters to the Scrum teams.

IA: The **Aligned Sprint Start <lifecycle>** will align the sprint heartbeat. All Scrum teams work toward the same point in time, the feature review. A fully aligned sprint heartbeat ensures natural alignment in activities between the Scrum teams.

IA: A common workflow over all teams will be rolled-out, consisting of predefined workflow states for features, stories and tasks. Features, stories and tasks each have their applicable, standardized workflow. Common workflow helps building and utilizing the shared mental model as described by (Jonker et al., 2011).

A story with a 'Ready' state will indicate a story that can be picked up for the sprint planning meeting. The state 'Todo' will indicate that a story is accepted by the Scrum team for a sprint. The ready definition will be bullet wise written down as the **Aligned Definition of Ready (DoR) <artifact>**. Features, stories and interface designs will be developed until the Definition of Ready is met by the product team.

151

IA: A story with the 'Done' state will be the indication for a story that can be demonstrated in a feature review. At that time the story has been realized and system tested, including interfacing and middleware testing. To allow full understanding of the status of a story before the 'Done' stage is reached, the stories test cycle will also be aligned between the teams. Such elaborated **Aligned Definition of Done (DoD) <artifact>** will align the shared understanding (Jonker et al., 2011) between the Scrum teams, helping teams to adapt and mitigate possible delays of other teams.

Issue 4: Automation (IT4IT)

IA: A **Workflow Application <artifact>** will be deployed to support the feature development lifecycle. Each feature will be entered into the application and tagged with a unique priority. The underlying stories will also be entered in the application and linked to the registered feature. Entering and updating the features and the feature workflow status will fall under the responsibility of the product team. Each Scrum team will be responsible for entering and updating the applicable stories and the story workflow state. The development tasks will be entered into the application and linked to a story by the Scrum team.

Issue 5: Visibility

IA: The Workflow Application has to support the feature development lifecycle and enhances visibility over the structure with features, stories and tasks. On each level planning, status, progress and impediments will be visualized for item progress tracking throughout the lifecycle. For instance the application sends e-mail to each user in the set of Scrum teams, in case a story or feature changes state or priority. All information in the workflow application will be accessible by all members. Compact minutes of meeting will be created and shared with the stakeholders. The prioritized list of features will also be shared with all Product Owners, Scrum Masters and IT managers on a weekly basis. Collaboration will be improved by enhanced visibility about the new way of working, as confirmed in the studies of Bartlett et al. (2007) and Vlietland and van Vliet (2014b). Actively sharing the new way of working with all Product Owners, Scrum Masters and IT managers will help punctuating the equilibrium of the existing organizational state (Gersick, 1991).

The IAs are packaged by the Scrum Value chain Framework (SVF). Figure 24 illustrates the SVF with the items as result of the IAs. For instance the intervention element: ‘Epic Product Owner (EPO)’, is shown as an icon with ‘EPO’ underneath. The blue colored items represents the (standard) Scrum framework (Sutherland & Schwaber, 2013), the orange colored items are additions to that framework, resulting in the SVF. Each of the abbreviations in the SVF is explained within each IA description (e.g. F2R, I2D).

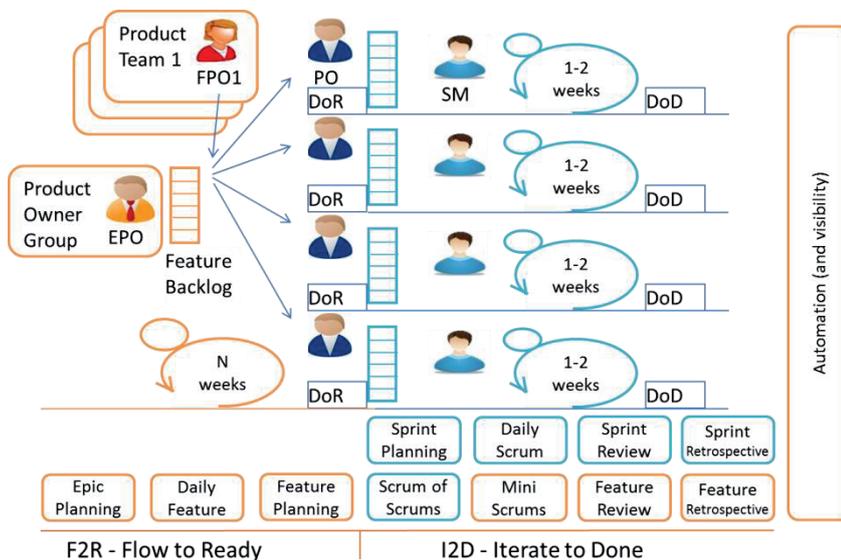


Figure 24: Scrum Value chain Framework (SVF)

Note: As discussed in section 5.5.1 (see remark [P3]), the advised sprint duration has been reduced to 1-2 weeks, instead of 4 weeks as initially defined in the IA. The reduced sprint duration, compensates for the slower feedback cycle, due to the extra ‘Flow to Ready’ phase, that is executed prior to the sprint cycle.

To validate whether the items in the framework cover all five AGF core-elements, each of the items are categorized under the AGF core-elements, as shown in Table 20. Each AGF core-element is covered with at least two (SVF) items (see cells).

Table 20, AGF coverage with (SVF) items

Role	Event	Team	Artifact	Lifecycle
Feature Product Owner (FPO)	Epic Planning	Product Owner Group (POG)	Feature Description	Flow to Ready (F2R)
Epic Product Owner (EPO)	Bi-daily Feature	Product Team	Aligned Definition of Ready (DoR)	Iterate to Done (I2D)
	Feature Planning	Mini Scrum	Aligned Definition of Done (DoD)	Aligned Sprint Lifecycle
	Scrum of Scrums		Workflow Application	Aligned Sprint Start
	Mini Scrum Feature Review Feature Retrospective			

5.3.3 Preparing for data collection and collecting evidence

The mail application and archive collects the typical responses as result of the IAs. Collected information of each response are: the response date, the responding person and the content of the response, with attachments if existent.

The data about story cycle time is extracted from the company workflow application (database). The database registers the stories (issue type Story) per Scrum team, which are exported to Excel with the workflow application web-end. For each of the involved Scrum teams the status change timestamps are extracted from the database with a customized MySQL query and a Putty terminal. The timestamps *ToDoDate* and *DoneDate*, as defined in Table 21, are collected.

Table 21, Variables and description of the cycle time variables

Variable	Description
ToDoDate	Timestamp that a story was committed in a sprint for the first time.
DoneDate	Timestamp that a story was moved to the done state

After analyzing the mail application and archive, and calculating the reductions on cycle-time, focus groups are setup to determine the impact of the interventions onto the cycle time, from the perception of the Scrum team members. The focus groups aim to validate the causality between the observed improvements and the IAs, instead of other interventions, actions or influencing factors. Focus groups have been found useful for generating information and shedding light on data already collected, and can be used prior, during and after events or experiences (Krueger & Casey, 2008). The focus groups in this study will evaluate the impact of the performance improvement after the IAs have been performed. Focus groups with four to six participants are organized. These small focus groups are more comfortable for the participants, as some levels of existing discomfort due to reorganization is expected (Krueger & Casey, 2008; Morgan, Gibbs, Maxwell, & Britten, 2002). The expectation is that these (mini) focus groups deliver more in-depth results, as participants likely have a great deal to share and the discussed topic has a high complexity (Krueger & Casey, 2008). The participants of each group are homogeneously selected to stimulate a focused discussion.

The interventions and activities that - according to the focus group - had the most impact on the shorter cycle time are collected and quantified, by using post-its. The focus groups also discuss and categorize the 'post-it items' for improved contextual understanding of the items. Each focus group session is audio-recorded to reduce analysis bias. A focus group with Product Owners and a focus group with Scrum Masters of the codependent teams were compiled, since these roles are directly involved in coordination, prioritization and alignment activities. A focus group with Feature product owners was compiled, being the actors that perform mechanistic front to back coordination (Scheerer et al., 2014).

5.3.4 Analysis of collected data

The start, duration and content of the IAs were determined by analyzing the mail history and finding typical keyword such as shows in Table 22. Each of the mails is analyzed for key responses (keywords) as result of an intervention.

Table 22, Mail database with typical keywords

Variable	Typical keywords
Coordination	Feature, Owner, Master, Coach, Scrum of Scrums
Prioritization	Group, Priority, Excel, Progress, Daily
Alignment	Definition, Status, Sprint, Time, Date, Workflow
Automation	<Workflow Application Name>
Visibility	<Status update>, Attachments, Minutes

The average cycle time of feature stories (ASD) is determined per week for each codependent team (T), by calculating the average number of open days (SD) of the feature stories (S) that were closed in that week (C). The overview of the performance variables is shown in Table 23. The performance analysis was done once, after the IAs were deployed.

Table 23, Performance variables

Variable	Description	Metric
T	Team identifier in the set of codependent Scrum teams	$T \in 1 - 6$
O	Week number of the week that a story is opened	$O \in \text{Week number}$
C	Week number of the week that a story is closed	$C \in \text{Week number}$
N (C, T)	Amount of stories for team (T), closed in week (C)	$N \in 0 - m \text{ stories}$
n (C, T)	Story identifier for team (T) closed in a week (C)	$n \in 1 - N$
SD (n)	Amount of days that story (n) is open	$\text{DoneDate (n)} - \text{TodoDate (n)}$
ASD (C,T,n)	Average number of open days for all stories by team (T) closed in week (C)	$\frac{\sum_{n=1}^N SD_{(C,T,n)}}{N}$
ASDstart	First measurement week of the average number of open days	4 weeks after start IAs
ASDend	Last measurement week of the average number of open days	6 months after start IAs

The items of the focus groups are analyzed to triangulate the IAs and the performance improvement. The focus group categories are compared with the collaboration related issues. The audio recordings are used as point of reference. Items that do not fit the the collaboration related issues are kept under the categories as defined by the focus groups. The sum of the allocated IA points (during step 3) determines the quantitative impact of a category.

5.4 Results

5.4.1 The case

A case in the banking industry at a large multinational bank which delivers financial services to large business customers was subject of study. The case entails a set of Scrum teams that offers solution delivery services to a high-volume banking value chain at the multinational bank. The case conforms to the selection criteria of Table 19.

The value chain is supported by a set of six Scrum teams with technical interdependencies between the applications under development of the Scrum teams. The front-office application (developed by Scrum team Beta and Gamma) captures the banking transactions, the mid-office application (developed by Scrum team Epsilon) processes the transactions and the back-office application (developed by Scrum team Eta) settles the transactions. Scrum team Gamma develops connectivity for the application that is developed by Scrum team Beta. Scrum team Delta develops generic

connectivity services. Scrum team Zeta develops generic applications that support the applications that automate the business process.

At the start of the IAs all Scrum teams record and track the stories in the workflow application. Each team has its own workflow, while the *Todo* and *Done* statuses are used by all codependent Scrum teams.

5.4.2 Performance development

Figure 25 shows the cycle time development of the feature stories of each Scrum team. On the X-axis the week number is shown. The Y-axis shows the cycle time. Each line represents a Scrum team and each dot the average number of days of the stories that reached the done status for that team in that week. A missing dot in a week indicates that no story was closed by the Scrum team in that week. A missing week indicates that no stories were closed by any team. Scrum team Beta and Gamma are combined in one graph because the two Scrum teams use one combined Product backlog.

156

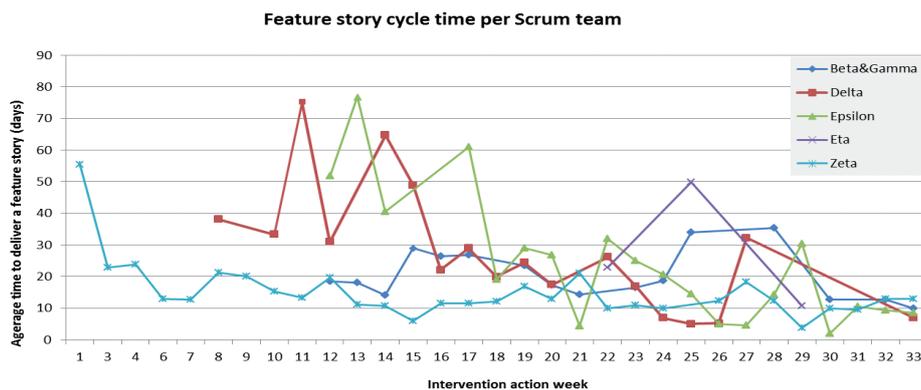


Figure 25: Trend of the Feature story cycle time

The first IAs started in week 4 (see Figure 25) and the last IAs started in week 18. The IAs were deployed quite organically, based on the social responses. Some teams needed extensive coaching and direction to keep the pace compared to other teams.

5.4.3 Intervention results

This section describes the typical responses by the members of the studied case as result of the IAs. The typical responses are illustrated by key quotes, collected by the mail application. The labels (see brackets '[']') are used in section 5.5 for reference purposes. The text between '<>' in the quotes contains edited text because of

confidentiality reasons, for instance in case of mentioned names of departments or members.

Prioritization

Implementing the priority setting framework and setting matched priority over all Scrum teams started in week 13.

[P1] "Many thanks for the lively and constructive discussion in the first Product Owner Group (POG) meeting. Below you find the summarized minutes of meeting. The ultra-short term target for the POG is to understand what has already been developed and start driving the development", Feature product owner

Weekly Product Owner Group (POG) meetings were planned from week 14 onwards to discuss and match priorities between the Scrum teams. Input for the meeting is the backlog that is high-level prioritized by the Epic product owner. The Scrum team product owners, feature product owners and the epic product participate in the POG. The role of the Scrum team product owners is to match the priority of the team backlogs within and after the POG meeting:

157

[P2] "The role of the product owners from each Scrum team is to align the backlogs between the teams. For instance feature X covers <a business function> which requires specific configuration and IT development each by each Scrum team", Feature product owner communicating the role to others

The prioritization process turned out to be complex and involved many stakeholders. Each stakeholder applied their influence for priority setting towards their interest, which often contradicted with the interest of another stakeholder. Priority also needed to be set well before the sprint to prioritize the refinement activities.

Coordination

Product teams started with standups from week 18 onwards, to share the achieved results, the next actions and the impediments. The realized feature stories were reviewed by the product owners:

[C1] "I do not understand the flow of events between the applications. I looks like to messages are send between application X and application Y, while only one time should be needed. Also the feedback from application Z should be an aggregation of all messages that have been sent", reviewing feature Product Owner

CHAPTER 5

A weekly held Scrum of Scrums is institutionalized in week 17-21 to coordinate the work between the Scrum teams. Each team delegates a team member to the Scrum of Scrums which is typically the Scrum Master, or a senior technical person. The Scrum of Scrums allows the teams to discuss their codependent activities, such as interfacing and integration:

[C2] "We have a joint view on organizational impediments, we share and leverage best practices across teams and we provide a sounding board from the shop floor..... As far as I know this is the only 'voice from the shop floor' and also offers direct input to the management team", Scrum coach explaining the typical Scrum of Scrum results

Mini Scrum of Scrum meetings were implemented as of week 12 to support the development of application connectivity between two Scrum teams. Participants of a mini Scrum of Scrum were the interface developer from each of the two Scrum teams and an interface specialist from the generic connectivity Scrum team (Delta). The mini Scrum of Scrum was facilitated by a Scrum Master of one of the Scrum teams.

158

Alignment

Scrum teams Alpha, Epsilon, Eta and Zeta gradually implemented a four weekly sprint heartbeat, as of week 4. Scrum teams Beta, Gamma and Delta implemented a bi-weekly sprint heartbeat fitting in the four weekly sprint heartbeats.

[M1] "Thanks for the presentation. One question about the sprints dates. For me, a sprint takes 4 weeks and not 1 month, which means that the dates I have in mind are slightly different.", Scrum Master correcting support staff

A single development workflow was implemented in all Scrum teams from week 12 onwards. The workflow was extensively discussed and communicated between stakeholders. An example of such communication is shown in the quote below:

[M2] "We earlier agreed that the additional state is required as otherwise too many different testing activities are placed under the 'Done' status. The extra status also better aligns with teams that do not develop via the Scrum framework. Nevertheless we should keep validating the necessity of the extra state because it is a workaround", workflow application manager

The workflow was approved by the managers of the Scrum teams in week 13. The workflow was subsequently discussed with the Scrum teams. As a result the teams aligned the test phases between the Scrum teams and mapped the test phases onto the workflow statuses. The Definition of Done (DoD) was discussed and agreed with the Scrum teams. The DoD was integrated with the existing test phases. 'Done' implied

that the functionality of a story worked in accordance with the feature stories, including the integration of the application connectivity.

Automation

Linkages between features and stories and the workflow statuses were configured in the workflow application from week 4 onwards. Several Scrum teams experienced difficulties in correctly connecting the stories to the features in the workflow application, indicated by the missing lines at the left in Figure 25. Coaching and guidance were required to correct and add the necessary information:

[A1] "We still miss items in the workflow application, such as (1) required Scrum team Beta and Gamma functionality to realize the interface, stories are linked to this feature; (2) interface <X> owned by Scrum team Gamma and (3) required functionality about Scrum team Alpha to process the data.", Feature Product Owner

159

Reporting by the workflow application turned out to be inadequate and Excel was introduced as reporting tool. The Excel report was manually compiled on a weekly basis by using the workflow application and Scrum teams as data source. The Excel sheet was then distributed via mail to all stakeholders.

Visibility

The framework was presented during the Product Owner Group (POG) kick-off meeting to the IT managers under which the Scrum teams operate and the product owners in scope of the Scrum teams. The way of working, including roles and responsibilities was afterwards distributed by minutes of meeting.

The workflow application was accessible by all internal employees and each status update by a value chain member was automatically communicated to all members via collaboration tooling. Access to the workflow application was not possible for a supplier that developed software for one of the Scrum teams.

[V2] "Access is required from <external supplier> to <Scrum team> to be able to have intercompany visibility on dev workflow. This topic was already discussed earlier. It is about providing access to <workflow application> for external employees. We are still investigating the setup. Technically this is possible obviously", workflow application manager

A weekly agenda was distributed to all members of the POG. The agenda included the (1) minutes of last meeting, (2) current status of feature stories and (3) the existing priority of features and (4) the current status of the stories and features in the sprint.

The distribution of the agenda triggered the necessary communication between POG members, such as discussing and prioritizing feature stories.

5.4.4 Focus group results

Scrum Masters, Product Owners and coordinators in the value chain were each allocated to a focus group. The group with Scrum Master and the group with Product Owners have 4 members. The coordinator role coordinates the Scrum team transcending activities in the value chain. That focus group has 5 members.

Each of the groups categorized the items on the post-its. The focus group categorization process was done on a white board by clustering yellow post-its while writing and updating the category names.

The items on the post-its confirmed that the performance improvement was achieved with the IAs. Even though the items were independently categorized from the collaboration related issues, the categories were remarkably similar to the IAs categories. Table 24 shows the sum of the points per category based on the allocated points per item per focus group member. The table also shows for each focus group and category, the percentage of the total number of allocated points. The total column is the sum of the three focus groups.

Table 24, Number of allocated points per category in each team

Category	Number of allocated points			
	One	Two	Three	Total
Alignment	44 (14%)	9 (3%)	73 (22%)	126 (41%)
Prioritization	27 (8%)	34 (10%)	15 (5%)	76 (23%)
Coordination	5 (2%)	38 (12%)	7 (2%)	50 (15%)
Visibility	3 (1%)	10 (3%)	18 (6%)	31 (10%)
Automation	15 (5%)	9 (3%)	2 (1%)	26 (8%)
Performance	6 (2%)	0 (0%)	10 (3%)	16 (5%)
Total	100 (31%)	100 (31%)	125 (38%)	325 (100%)

Focus group Two, with coordinators that coordinate the feature activities between Scrum teams, allocate the most points to the Coordination IAs. Scrum Masters and Product Owners allocate significantly less points to Coordination (2%). Product Owners allocate the most points to Alignment (common sprint heartbeat, workflow, DoR and DoD), the least points are allocated to Alignment by the coordinators. Two focus groups mentioned performance related items, such as: *“Teams are able to pick up more stories”* for which an additional category was created. For a few items was referred back to the participant to further explain the item, next the audio recording.

The focus groups confirmed the importance of learning during the deployment of the IAs. Typical items on the post-its illustrate that learning process: *“The work between Scrum teams has improved”*, *“Maturity of understanding the (collaboration) process”* and *“Better usage of the workflow tool”*. Learning must be seen as inextricably linked to the deployment of the IAs, and the learning related items were therefore categorized under the other categories.

5.5 Discussion

5.5.1 Discussion of the results

In this section the performance development and swift delivery of business value by a set of codependent Scrum teams is discussed and analyzed, by referring to the typical quotes as result of the IAs, and the focus groups. The items between brackets ‘[]’ refer back to the quotes in section 5.4.

The results confirm the effectiveness of the IAs. The cycle time of feature related stories in Figure 25 shows a significant decreasing trend while performing the IAs, ultimately leading to more Agility in the value chain. The focus groups endorse the effect of the IAs. The trend in Figure 25 moves from an average of 29 days cycle time to 10 days cycle time and seems to stabilize at 10 days. A cycle time of 10 days is equivalent to approximately two working weeks, while teams Alpha, Epsilon, Eta and Zeta have a four week sprint cycle [M1]. These results show that stories are delivered faster than the sprint cycle. A driver to deliver faster might be other teams that deliver interdependent stories in a bi-weekly sprint cycle. These teams with a bi-weekly sprint cycle might put social pressure on teams with a four week sprint cycle to deliver faster. Such premised social factor cannot be validated with the current dataset and might be subject for further study.

The prioritization process of a feature affects the feature preparation of the upcoming feature preparation process and the subsequent sprint cycle, as shown in quote [P3]. The quote shows that the preparation process preceding the sprint cycle slows down the feedback loop. For instance, a feature during the sprint cycle cannot be realized due to an unexpected dependency with another feature. The feature priority on the feature backlog has then to be changed. These changes will result in new prioritized features on the backlog, which need to be prepared before that feature can be realized. To mitigate such longer feedback loop a shorter sprint cycle of 1-2 weeks is suggested. The three focus groups confirm the impact of the prioritization IAs as shown in Table 24, such as backlog refinement by slicing work into small sized stories that can be prioritized by a team.

CHAPTER 5

Coordination by means of the mini Scrum of Scrums achieved more in-depth focus than a Scrum of Scrum meeting mitigating the disinterest and superficiality in Scrum of Scrums meetings (Paasivaara et al., 2012). The mini Scrum of Scrums stimulated detailing activities prior to the sprint, preventing impediments during the sprint. The focus group with coordinators allocates 38 points to the coordination category between Scrum teams. The number of points confirm the effectiveness of the coordination IAs, that deeply embedded coordination within and between Scrum teams, confirming the finding of Vlietland and van Vliet (2015b).

Entering the data in the workflow application was perceived difficult [A1], which is confirmed by the jumps between data points in Figure 25 at the start of the IAs. Reliability of the performance graph increases over time, even though one of the selection criteria is a single workflow application used by all teams. The combination of visibility, coaching and increased usage of the workflow application stimulated the increase of data entry reliability.

162

Visibility was limited by the workflow application due to the inaccessibility to external suppliers [V2] and the limitation in reporting which had to be mitigated by the usage of Excel and the mail system. Further improvements in this area will likely help utilizing visibility as factor for swift feedback and mitigating impediments (Vlietland & van Vliet, 2014b, 2015b).

The combination of top-down and bottom-up IAs improved the implementation effectiveness. The top-down implementation gave the teams the necessary focus, for instance the prioritization framework [P1]. The bottom-up implementation confirmed the actual adoption, actual commitment and the state of the mental change by the members in the value chain. The bottom-up implementation also utilized the intelligence on the shop floor and provided the necessary feedback about the feasibility of the top-down intervention actions.

The introduction section explained the collaboration related issues that codependent Scrum teams currently face, that slows down the cycle time of new features (Vlietland & van Vliet, 2015b). The case study presented in this chapter shows that a set of IAs can alleviate these issues, resulting in cycle time reduction. The SVF with its IAs helps achieving that cycle time reduction. Such cycle time reduction improves the Agility of the value chain, enabling swift delivery of business value to the client, possibly resulting in a better competitive position. The SVF aims to comply with the Agile manifesto (Beedle et al., 2013) by having a mix of top-down and bottom-up intervention actions. Such mix is mentioned as a good-practice by other authors (Batra et al., 2010; Port & Bui, 2009; Soundararajan & Arthur, 2009). Based on the findings the premise is that the SVF offers sufficient structure for large scale Scrum as

mentioned by Talby and Dubinsky (2009), Soundararajan and Arthur (2009) and Batra et al. (2010), while maintaining the necessary flexibility as intended by the Agile manifesto (Beedle et al., 2013).

5.5.2 *A caveat*

Many will possibly oppose a setup with multiple codependent Scrum teams in a value chain. Ideally, each Scrum team should cover the end-to-end delivery, to prevent the negative impacts of dependencies in a chain. Looking in perspective at a set of codependent Scrum teams in a value chain, organizations are just installing a new type of waterfall: one of teams instead of one of development phases. Such a waterfall of teams could never have been the intention of the inventors of Scrum. However, in complex environments with complex IT landscapes, there is often no real alternative, as Scrum development teams of more than 9 members are not allowed. In those settings adopting the IAs and/or the framework is a best practice. However, not without emphasizing that organizations should simultaneously put effort into decreasing their complexity, allowing Scrum teams to cover end-to-end delivery.

5.6 Threats to validity

For sure, a practical study with IAs in a real-life setting, involving multiple teams with real people has limitations and brings threats to validity.

First of all, this is just a single case. Though the IAs were implemented in multiple teams and proved their impact, this is still one case-study in one multinational bank. As such the causal relation between the IAs and the performance improvements cannot be generalized. The results can also not be generalized to the financial domain. Given the setting of the teams, we do expect that the domain itself has limited influence. As such, we recommend the repetition of the IAs in more case-studies, so as to increase the generalizability for sets of codependent Scrum teams in general.

Secondly, the impact of the combination of the IAs has been validated. The IAs were packaged into the SVF, to be used in organizations that want to decrease the cycle-time of their Scrum teams in a codependent setting. Though, each individual IA cannot be traced to the reduction in cycle time, since the actual data was extracted after the IAs were performed, and the effect of an individual IA was not recorded. Another experiment with a different setup is required to determine the effect of each IA.

Thirdly, the IAs were developed from related work that contains experience reports with similar empirical case studies. As such the external validity of the IAs seems stronger than just a single case. However, the interrelationships between the actions,

the level of impact of the individual actions and the balance between them have not been studied in the present research. Furthermore, the IAs were not deployed simultaneously in all teams. Even though the teams were selected based on stability criteria, there might be some bias due to individual team learning that influenced the reduced time of the feature stories next to collaboration learning between teams.

Finally, the relationship between the impact of the IAs and the decreased cycle time with the focus groups was triangulated. As such, there is stronger evidence that the IAs did have an impact in the practical case. Measures were taken to prevent bias in the focus groups, by splitting the focus group session in two parts. The first part identified and refined the focus group categories independent from the IAs and the second part determined the impact of each identified category. During the first part the top 10 interventions and activities from each individual was collected before integrating them into categories, thus preventing influence of dominant individuals in the focus groups. The influence of the focus group setup on the confirmations of the IAs is therefore considered to be low.

The SVF needs to be tested in other organizations. For example, the SVF assumes the Epic product owner to be capable to uniquely prioritize all features. This worked in this empirical case but an environment with higher complexity might reduce the decision making effectiveness of the Epic Product Owner. Such decision making effectiveness of the Epic Product Owner requires further study. One might also consider this a generic issue with Scrum by assuming competent role fulfillment.

Finally, this work has been carried out in a practical setting. Participants in the study, especially the Scrum teams involved, understood that the IAs were taken with a specific purpose. Though, it was not the goal in itself to decrease cycle-time specifically, the teams knew that the actions were taken to improve their collaboration, prevent delays and increase the predictability over the complete value chain. As such, this might have influenced the results (Hawthorne effect). Given the observations and participant opinions in this study, these influences are considered rather limited.

5.7 Conclusion

In this study a set of IAs, packaged in the SVF, is validated to alleviate collaboration issues in a set of codependent Scrum teams that delays the swift delivery of business value. The IAs result in a cycle time reduction from 29 days to 10 days. The archival records showed the implementation of the IAs, and the delivery metrics confirmed their impact. The participants in the focus groups confirmed the causality between the

observed performance improvement and IAs. The results indicate the effectiveness of the IAs and the SVF for codependent Scrum teams.

The results indicate that the SVF helps IT service networks to realize IT changes faster, enabling large companies in the information-intensive industry to swiftly adapt to market changes. Since these companies experience rapid changing business demands, the SVF will likely help companies to achieve a better competitive position, as suggested by Melville et al. (2004).

Imposing a set of IAs to be interpreted by teams themselves is likely introducing new challenges, such as misinterpretations, ignoring a specific action, timely attention to an action, and so on. As such, packaging the results into a single SVF is expected to help improving the Agility of Scrum teams in a codependent setting. Besides recommending the application of the SVF in other settings, so as to further validate its effectiveness, we recommend repeating the IAs separately in other empirical settings. This is expected to enhance the understanding of the interdependencies between the actions and the level of impact of the individual intervention actions. A future research avenue is therefore to research the individual IAs, such as qualitatively and quantitatively researching the effect of priority setting onto the cycle time, the predictability and efficiency of the set of codependent Scrum teams. A second research avenue is to study prioritization challenges in larger scale settings with multiple feature backlogs and multiple value chains supported by multiple codependent sets of Scrum teams.

