IMPROVING THE AGILITY OF IT SERVICE NETWORKS

Jan Vlietland

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School of Information and Knowledge Systems.

SIKS Dissertation series No. 2015-

This research has been partially sponsored by Search4Solutions B.V. Parts of this research have been performed in collaboration with Search4Solutions B.V.

Promotion committee:
prof.dr. J. Bosch (University of Gothenburg/Chalmers, Gothenburg, Sweden)
prof.dr. S. Brinkkemper (Utrecht University, Utrecht, The Netherlands)
prof.dr. R. Kusters (Technical University Eindhoven, Eindhoven, The Netherlands)
prof.dr. R. van Solingen (Technical University Delft, Delft, The Netherlands)
prof.dr. H. Reijers (VU University, Amsterdam, The Netherlands)

ISBN 978-94-6295-202-7

NUR 082 (Informatica & Management)

Subject headings: Improving the Agility of IT Service Networks

Printed by Uitgeverij BOXPress || ’s-Hertogenbosch

Illustrations by Stefanie van den Herik and Roxanne Vlietland

Typeset in Calibri by the author

VRIJE UNIVERSITEIT

IMPROVING THE AGILITY OF IT SERVICE NETWORKS

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op donderdag 18 juni 2015 om 15.45 uur
in de aula van de universiteit
De Boelelaan 1105

door

Jan Vlietland

geboren te Goedereede

promotor: prof.dr. J.C. van Vliet

Acknowledgements

Many people warned me in advance about the massive amount of required time and energy to obtain a PhD degree. It is one of the rare moments in life that I consider myself lucky afterwards for not listening. Working 20 to 30 hours a week for six years on a doctorate concurrently with full-time international consultancy and three entrepreneurial initiatives sometimes felt as a mission impossible. Now reaching the end of the endeavor I look back on the past six years with relief, pride and humility. I am so grateful for having met and worked with so many fascinating people. It is because of the help of all these people, I was able to realize my dissertation. I feel grateful for having these people as friends, colleagues and clients; they are forever in my heart.

First of all I thank Hans van Vliet for his everlasting support in getting me to the finish line. I am so grateful for having him as my mentor, coach and teacher. Furthermore I thank Paul Ghijsen and Rob Kusters for their mentoring and support during my Master and first stage of the PhD. I also thank Rini van Solingen for bringing me in touch with companies and for his supplementary advice. Lastly I thank my research colleagues for their discussions, reviews and also for just being around.

Furthermore I thank the decision makers and all the great people of the organizations under study. Unfortunately I cannot personally express my gratitude to these people here due to non-disclosure agreements. Still they so much deserve a personal expression of gratitude. By giving me the opportunity to collect the required empirical data I was able to contribute to science and develop my dissertation.

I also thank my parents and family for listening to my endless (research) stories and their support whenever needed. I also thank my daughters Roxanne and Jennifer for their patience and understanding for hardly having me around for so long. I also thank Susan for being a great mother and taking care of our kids. To conclude, I thank Roxanne for her crisp illustrations about the research topic.

Dankwoord

Veel mensen hebben me vooraf gewaarschuwd voor de enorme hoeveelheid tijd en energie die het vergt om een doctorsgraad te behalen. Het is een van de zeldzame momenten dat ik mezelf achteraf gelukkig prijs dat ik niet heb geluisterd. Het zes jaar lang werken aan een PhD voor 20 tot 30 uur per week met daarnaast voltijds internationale consultancy en drie ondernemingen voelde af en toe als een onmogelijke missie. Nu het einde nadert kijk ik terug op de laatste zes jaar met opluchting, trots en nederigheid. Ik ben dankbaar dat ik zoveel fascinerende mensen heb mogen ontmoeten en met hen heb mogen samenwerken. Dankzij de hulp van al deze mensen was ik in staat om mijn proefschrift te realiseren. Ik voel dankbaarheid om al deze mensen als vriend, collega en/of klant in mijn hart te kunnen sluiten.

Allereerst dank ik Hans van Vliet voor zijn niet aflatende ondersteuning en zijn begeleiding naar dit moment. Ik ben ontzettend dankbaar om hem als mijn mentor, coach en leraar te mogen hebben. Ook dank ik Paul Ghijsen en Rob Kusters voor hun begeleiding en ondersteuning tijdens mijn Master en het eerste deel van mijn PhD. Rini van Solingen dank ik dat hij mij in contact bracht met bedrijven en natuurlijk voor zijn aanvullende advies. Tenslotte dank ik mijn wetenschappelijke collega's voor onze discussies, hun reviews en ook voor de gezelligheid en collegialiteit.

Verder dank ik de beslissers en alle andere mensen in de organisaties die zijn onderzocht. Helaas kan ik deze mensen hier niet persoonlijk bedanken, wegens de geheimhoudingsverklaringen. Niettemin verdienen zij allen een persoonlijk dankwoord. Doordat ze me de mogelijkheid gaven om de benodigde gegevens te verzamelen heb ik waarde aan de wetenschap kunnen toevoegen en dit proefschrift kunnen ontwikkelen.

Ik dank mijn ouders, familie en vrienden voor het luisteren naar mijn eindeloze (onderzoeks-)verhalen en hun ondersteuning wanneer ik dit nodig had. Mijn dochters Roxanne en Jennifer ben ik ongelofelijk dankbaar voor hun begrip en geduld terwijl ik er nauwelijks voor ze was. Ook dank ik Susan voor haar fantastische moederschap en het zorgen voor onze kinderen. Als afsluiting dank ik Roxanne voor haar verfrissende illustraties over het onderzoeksonderwerp.

Contents

List of figures

List of tables

Chapter 1

Introduction

The Industrial Revolution transitioned manufacturing processes from hand production methods to machines. To achieve that transition, work was split into discrete unambiguous tasks (Taylor, 1914). These discrete tasks allowed a controlled execution by machines and low educated staff (Taylor, 1914). To organize the work, the machines and staff were grouped into departments, each with predefined responsibilities, placed in a hierarchical structure (Mintzberg, 1989). Since each staff member was responsible for one or more discrete tasks they lost sight over the overall workflow process and were unable to organize the work themselves. The organization of the work was therefore done by separate supporting staff and the techno structure (Mintzberg, 1989). Nowadays many large companies still hierarchically organize work.

Hierarchical structures are good for reaching top-down decision making among large numbers of people in a timely fashion (J.R Galbraith, 1977). Though, in today's fast moving business, creativity and operational autonomy are important to survive competition (G. Lee & Xia, 2010). To that end staff working in different departments needs to spontaneously collaborate and share information (Beck, 2010). In a traditional hierarchical structure, however, information flows from workers up and down via the chain-of-command, involving many people, obstructing efficient collaboration and information sharing (Cummings & Worley, 2014). Information technology (IT) overcame the dysfunctional effects, by peer-to-peer, low boundary information sharing between staff without involving management (Melville, Kraemer, & Gurbaxani, 2004). That direct point-to-point information sharing allows efficient information sharing and collaboration in workflow processing that cross many departments (Davenport & Short, 2003). Large companies therefore highly depend on IT, for maintaining their competitiveness in today's fast moving business (Melville et al., 2004). The IT that supports these workflow processes consists of many interdependent IT components (Zachman, 2002). These IT components have typically critical interdependencies; only in case all interdependent IT components function correctly, workflow processing is enabled. For instance a financial services App on a mobile device requires many interdependent IT components for executing a payment transaction between two banking accounts.

To keep the IT components operational, human activities from IT staff are required; for instance an IT failure that requires replacement of a physical component by maintenance staff (Kumbakara, 2008). The combination of predefined IT and human activities for delivering IT are defined as IT services (Beck, 2010; van Bon, Jong, & Kolthof, 2007). Based on van Bon et al. (2007) in this dissertation an IT service is defined as: "a mix of predefined automated and human activities, enabled by hardware and software". This definition fits the three service characteristics as opposed to those of physical products (Baltacioglu, Ada, Kaplan, & Yurt, 2007; Moeller, 2010; Niessink & van Vliet, 2000; Parasuraman, Zeithaml, & Berry, 1985, 1988): (1)

Intangibility, services cannot be seen, touched, smelt or tasted, as they are 'performances' rather than 'things'. (2) Simultaneity; reflecting the fact that users must be present for the service to be provided. (3) Perishability; if a service is not consumed when available, the unused capacity is lost.

IT services are delivered by IT service providers (ISPs) (Kumbakara, 2008). Some ISPs deliver specialized IT services to a single company, such as a finance department. Other ISPs deliver IT services to a worldwide customer base, such as Google Drive. Many ISPs also deliver IT services to other ISPs. For instance Microsoft delivering a cloud based application hosting platform to an internal ISP of a company. Together these ISPs form networks, each delivering interdependent IT services.

IT services in that interdependent network are continuously updated, upgraded and renewed, to adapt to the fast moving economy (Beck, 2010). The faster an IT service network realizes these changes, the better an IT service network can adapt to changes in the business environment. Performing such changes on the IT services is a complex task, as IT services (and the IT components of the IT service) have many interdependencies spanning multiple ISPs. A change therefore typically impacts multiple IT services delivered by different ISPs (Plugge & Janssen, 2009; TFSC, 2011). Given these interdependencies, staff from different IT services staff needs to collaborate between and within ISPs; for instance during impact analysis and testing activities or during IT failure analysis to find the root-cause. With improved collaboration within and between ISPs, IT services can be changed faster (Costa, Cataldo, & de Souza, 2011; Sharp & Robinson, 2008).

Changes in IT services often lead to IT outages, which causes interdependent IT services to fail (Vlietland & van Vliet, 2014c). In other words, in case an ISP fails to deliver an IT service to a dependent ISP, the latter ISP can also not deliver its IT service. Currently many large companies suffer from these interdependent IT outages. For instance Research In Motion (RIM) experienced a devastating blow after their Blackberry service failed and users in Europe were unable to communicate for days (Thomson & Miller, 2012). Other examples are Lloyds, TSB and HBOS that were unable to service customers for a three-hour period when a major IT glitch hit Lloyds Banking Group (Flinders, 2014) and RBS presenting incorrect balances after a software failure that stopped processing payments (Scott, 2014). Many IT outages daily disrupt business workflows resulting in potential loss of market share and deteriorated operational profit (Evolven, 2014; Lerner, 2014). To limit the negative impact a fast response to IT outages is required.

To respond faster to events, large ISPs increasingly transfer to Agile methods (VersionOne, 2013). Agile methods promote continuous adaption instead of detailed planning upfront, having its origin in the social constructionism philosophy. Agile is based on Lean thinking and has been primarily adopted by the software development community (VersionOne, 2013). The Agile manifesto laid out the underlying concept of Agile software development (Beedle et al., 2013). Agile primary targets software development (Beedle et al., 2013). Yet, IT operations and IT infrastructure departments maintain existing software and distribute new software to production. Agility in the IT Operations and IT infrastructure environment is therefore needed to prevent bottlenecks in the deployment process to production. To mitigate these bottlenecks Agile thinking has been introduced to IT operations, with DevOps and Continuous Delivery (Humble & Molesky, 2011; Loukides, 2012). The word 'DevOps' is a portmanteau of "development" and "operations", bridging software development, operations, and services (Loukides, 2012). DevOps encourages collaboration between IT development, IT operations and IT infrastructure, by building liaison relationships between teams and the automation of boundary spanning activities (Feitelson, Frachtenberg, & Beck, 2013; Humble & Molesky, 2011; Loukides, 2012). Continuous Delivery entails the automaton of the software manufacturing process, from concept to cash (Humble & Farley, 2010; Poppendieck & Poppendieck, 2007). With the implementation of Continuous Delivery software is automatically integrated, tested and deployed to production, to evade repetitive work and human error.

1.1 Main research question

In the previous section is explained that IT service networks transcend ISPs and cover software development and IT operations. Since Agile methods improve the performance of software development (Bosch & Bosch-Sijtsema, 2011), the question is how Agility can be utilized to improve the performance of an IT service network. The main research question for this dissertation is therefore defined as:

Main research question: How to 'improve' the 'Agility' of 'IT service networks'?

To answer that question an initial literature search on IT service networks was conducted. Table 1 shows four Google scholar search strings that represent the initial search and the Google Scholar results (hits). Beck (2010) states that theory in IT service research is underdeveloped. Also authors in the 'service supply' research field mention the lack of theory about networks and changes of services (Baltacioglu et al., 2007; Zailani & Kumar, 2011). Based on the statements of these authors and the small number of search results, was decided to further study IT service networks.

Table 1, Search strings and Scholar Google hits

Search string	Hits	Ranked hits reviewed
"IT service network"	19	All
"IT service network" performance	8	All
"IT service network" improve	9	All
"IT service network" Agility	2	All

1.2 Literature review

To answer the main research question three literature reviews were conducted, in three moments in time, which is explained in the remainder of this section. Section 1.2.1 explains the literature review process. Section 1.2.2 provides the results of the first and second literature review. Section 1.2.3 provides the results of the third literature review.

1.2.1 Literature review method

The literature review process has three phases: (1) planning the review, (2) executing the review and (3) concluding the review, based on Saunders, Lewis, and Thornhill (2011). Each phase was completed by a closing discussion with the supervisor. The phases of the third literature review were more iteratively performed, without explicit closing discussions.

Phase 2 of the literature review process was split in three stages to identify, select, assess and extract the information in the literature: (1) initial review online, (2) elaborate review after downloading and (3) storage and usage in EndNote. As selection criteria (a) the literature needed published by a scientific publisher and (b) the literature needed to help answering the research question. Books and business articles were considered supplementary sources, to support knowledge building about the research field.

Stage 1: At the first stage the literature was initially reviewed online, by reading the abstract and scanning the conclusion section. Google Scholar was used as search engine for the literature search. Full access to scientific libraries (e.g. IEEEExplore, ScienceDirect and SpringerLink) via Google Scholar was achieved with the library account of the university.

The Google search algorithm carries a risk of missing the least cited scientific literature (Google, 2015), as citation count is the highest weighted factor in the ranking algorithm (Beel & Gipp, 2009), which strengthens the Matthew effect (Merton, 1968).

The risk was mitigated by snowballing (tracing back references in the identified literature). The residual risk was acknowledged and accepted.

The number of ranked search results (on which the initial review was carried out) usually varied between 30 and 80 hits. The number of initially reviewed results depended on the contribution of answering the search question in the higher ranked reviewed articles (search results). In case the search resulted in many articles that helped answering the research question, the search string was combined with other keywords to limit the number of hits. A used search string was stored in Excel, including the number of hits. Literature that met the selection criteria was downloaded and locally stored in a directory.

Stage 2: At the second stage the downloaded articles were reviewed in more detail, by reading the method section, the results section, and then the rest of the article, if applicable. To speed up the reading process, keyword searches in the reader were performed, based on the search strings. With snowballing techniques additional related work was identified.

Stage 3: The articles that passed the second stage of the review were categorized and uploaded in Endnote, for referencing in own work. At that stage key-sentences with the reference were copied in Word documents for future usage.

1.2.2 First and second literature review results

The first literature review was performed in 2010 at the beginning of the Master thesis (Vlietland, 2011), taking approximately three months. The second literature review was performed in 2011 at the start of the PhD, taking one month, to increase the rigor of the literature review. Both literature reviews aimed identifying literature about performance improvements in the IT service network research field.

Based on the initial search results the search string ‘IT service network’ was loosened to “IT service” and ‘network’. Of that search the first 100 search results were reviewed, which resulted in four classifications of related work: (1) Technical IT services (e.g. Chen, 2008; C. Lee & Helal, 2002; Meshkova, Riihijarvi, Oldewurtel, Jardak, & Mahonen, 2008), (2) IT outsourcing (e.g. Currie & Seltsikas, 2001; Earl, 1996), (3) IT service management (e.g. David, Schuff, & St Louis, 2002; Lemoine & Dagnæs, 2003; McNaughton, Ray, & Lewis, 2010; Tan, Cater-Steel, & Toleman, 2009a; Winniford, Conger, & Erickson-Harris, 2009) and (4) Supply chains and logistics (e.g. Min & Zhou, 2002). Literature about technical IT services was not taken into account, as that literature has a technical perspective on IT services, while the research in this dissertation uses a human collaboration perspective.

A subsequent literature review was then performed in the remaining three fields of related work: (1) outsourcing - as multiple outsourcings relationships resembles a service network, (2) IT service literature – as IT services are the actual objects that are delivered in the network and (3) non-IT related supply chains and supply networks - given the potential similarity with IT service networks. To that purpose alternatives for the main keywords ‘IT service network’ and ‘Improving’ were identified in the three fields of related work. Table 2 shows the overview of the alternative keywords.

Table 2, Typical keywords and conclusion identified literature

Year	Main keyword	Alternative keywords	Conclusion identified literature
2010 – 2011	IT service network	IT service, provider, supply chain, network, outsourcing, performance	Lack of IT service network perspective IT service limited to a dyadic perspective
	Improve	Improving, improvement, quality, information, sharing, visibility	Lack of IT service network perspective Closest match improvements in supply chains
2013	Agility	Agile, Scrum, Continuous Delivery, DevOps	Lack of IT service network perspective Limited to a software development context

For the subsequent literature review 96 search strings were compiled, based on the combination of ‘IT service network’ and ‘improving’ or alternative keywords. A new combination was compiled based on the search results of previous combinations. For instance the combination “information visibility ‘service supply chain’ ” resulted in 83 hits about service supply chains. The next search string was therefore further tightened to: “ ’information visibility’ ‘service supply chain’ ” resulting in 14 hits. During the first and second literature review a total of approximately 500 papers passed stage 1 and 169 papers passed stage 2 for uploading in EndNote.

While a large body of literature about improvements in supply chain, outsourcing and IT service settings was identified, the literature was insufficient for answering the main research question. Literature about supply chains provided insufficient answers because of the differences between product delivery and IT service delivery (Jansen & Cusumano, 2013; Parasuraman et al., 1985). In the field of outsourcing and IT services the literature provided insufficient answers because of the dyadic perspective.

Based on the results of the literature review the decision was made to perform empirical research to answer the main research question. The remainder of this section provides a representative overview of the literature that was identified during the literature review.

Related literature as result of alternative keywords for 'IT service network'

Related literature exists in the area of Supply Chain Management and Service Supply Chains (Ahlstrom & Nordin, 2006; Zailani & Kumar, 2011). The developed service supply chain framework of Baltacioglu et al. (2007) models the core processes for service supply chains. Although these processes have similarities with IT service processes (van Bon et al., 2007) it remains ambiguous whether the proposed processes fit IT service networks. Another case study with a supply chain perspective in the IT industry was conducted by Yu, Suojapelto, Hallikas, and Tang (2008). Their article identifies the potential complexity of IT service networks and similarities with supply chain networks. However their results have insufficient detail for answering the main research question.

In the area of outsourcing an overview of outsourcing literature is provided by Lacity, Khan, and Willcocks (2009). Many of the supply chain articles in that overview were reviewed during the literature review. All reviewed articles have a dyadic perspective, not a network or chain perspective. Additional Google scholar searches in this research field also resulted in literature with a dyadic perspective (Hatonen & Eriksson, 2009), including multi-vendor outsourcing literature (Cohen & Young, 2006). That dyadic perspective also exists in IT service delivery literature (Kumbakara, 2008; Rosa, 2012). Moreover, IT service literature typically targets IT processes in the IT operations field (Jantti, 2011; Jantti & Suhonen, 2012; Tan, Cater-steel, & Toleman, 2009b; van Bon et al., 2007), without having an Agile view (Kang & Bradley, 2002; Niessink & van Vliet, 1999).

Related literature as result of the literature review on 'Improving'

Literature about performance improvements exists in various related fields. One related improvement field is identified in supply chain literature. A factor for improving supply chain performance is information and knowledge sharing (Prajogo & Olhager, 2011; Rashed, Azeem, & Halim, 2010; Sahin & Robinson, 2005). Another factor related to information and knowledge sharing for improving supply chain performance is information visibility (Bartlett, Julien, & Baines, 2007; Caridi, Crippa, Perego, Saianesi, & Turmino, 2010b). A lack of visibility also causes bullwhip effects, resulting in large fluctuations in inventory, extending the time of delivery in supply chains (Bhattacharya & Bandyopadhyay, 2011; H. L. Lee, Padmanabhan, & Whang, 1997; Viswanadham, Desai, & Gaonkar, 2005). The third related factor targets collaboration improvements in supply chains (Cao & Zhang, 2011), which is related to shared information and shared knowledge (Banbury, Helman, Spearpoint, & Tremblay, 2010; Fuks et al., 2008; Wood & Gray, 1991).

A large body of improvement literature in the area of IT service delivery was identified (Jantti, 2011; Jäntti, 2012b; Jäntti & Järvinen, 2011; Niessink & van Vliet, 1998; Tan et al., 2009b). The work of Niessink (2001) resulted in the ITS-CMM framework (Niessink & van Vliet, 1999) for the support of IT service improvements. While the model acknowledges linking IT services (e.g. by the term 'Integrated Service Management'), the model does not have an IT service network perspective. After publication of ITS-CMM the service delivery perspective was integrated in the Capable Maturity Model Integrated (CMMI),with CMMI for Services (Team, 2010b), without any guidance for improving chains and networks of IT services.

1.2.3 Third literature review results

Answering the first three research questions (see section 1.3) resulted in new scientific insights. Those new insights initiated a third literature review in 2013, performed in the field of IT service network Agility. The literature review with a one month duration resulted in the third category of related work, as shown by the typical keywords in Table 2. Even though there is a large body of Agile literature, no literature was identified that uses an IT service network perspective. Moreover Agile literature targets software development, not the full IT service lifecycle (e.g. IT operation). The remainder of this section provides the representative overview of the identified literature in this field.

<u>Related literature as result of the literature review on 'Agility'</u>

Agile literature is identified in the field of distributed software development (S. Lee & Yong, 2010; Sutherland, Schoonheim, Rustenburg, & Rijk, 2008). Sutherland, Viktorov, Blount, and Puntikov (2007) consider three models for collaborating Scrum teams in a distributed context. Sutherland, Schoonheim, and Rijk (2009) propose a setup for multiple collaborating Scrum teams. All identified research on such collaborations (Bosch & Bosch-Sijtsema, 2010; Hildenbrand, Geisser, Kude, Bruch, & Acker, 2008; Oppenheim, Bagheri, Ratakonda, & Chee, 2011; Sharp & Robinson, 2010) do not have an IT service network perspective.

That lack of IT network perspective also exists in scaled Agility literature (Larman & Vodde, 2008; Schnitter & Mackert, 2011; Sutherland, 2001). Rautiainen, von Schantz, and Vahaniitty (2011) study the introduction of portfolio management to support scaled Agile development, without clearly describing whether the studied case has a network setting. Other authors identify coordination as a variable that affects scaled Agile effectiveness (Begel, Nagappan, Poile, & Layman, 2009; Costa et al., 2011; Paasivaara, Lassenius, & Heikkila, 2012). Larman and Vodde (2013) use feature teams and liaison relations (J.R. Galbraith, 1971) with communities of practice (CoP) for exchanging knowledge and coordination between teams. All identified literature on coordination do not mention networks as a scaled Agile setting.

Ambler (2009) uses a more abstract view on scaled Agile by identifying eight scaling complexity factors: (1) team size, (2) geographical distribution, (3) regulatory compliance, (4) domain complexity, (5) organizational distribution, (6) technical complexity, (7) organizational complexity and (8) enterprise discipline, while also lacking an IT service network perspective for a scaled Agile setting.

Several other authors discuss Continuous Delivery for enabling Agility by IT process automation (also named IT4IT), without mentioning a network setting (Humble & Farley, 2010; Olsson, Alahyari, & Bosch, 2012). Literature on Agile/DevOps, being a collaboration philosophy between dependent staff, was either not peer reviewed (Humble & Molesky, 2011; Phifer, 2011), or not having a IT service network perspective (Humble & Molesky, 2011).

The relationship between Agile projects and software process improvement is studied by Salo and Abrahamsson (2005). To improve the adoption of Agile methods Qumer and Henderson-Sellers (2008) present a framework. Several other authors also study the adoption of Agile methods (Pikkarainen, Salo, & Still, 2005; Ringstad, Dingsøyr, & Moe, 2011). An elaborate review of prominent work in the area of Agile improvement is performed by Akbar, Hassan, and Abdullah (2011). None of the Agile improvement literature uses an IT service network perspective, which is necessary to answer the main research question.

1.3 Research questions

To answer the main research question the decision was made to conduct empirical research. To that purpose the main research question was split into five (sub) research questions. Each of these questions was empirically studied. Answering these sub questions built sufficient knowledge and insight to answer the main research question. Since interdependence in IT service networks likely affect IT service network performance (Cao & Zhang, 2011; Wilhelm, 2011), the research starts by studying the interdependencies between ISPs, teams and staff. Social network theory helps modeling the interdependencies. According to social network theory, interdependent ISPs, teams and staff form networks, consisting of nodes and links (Freeman, 1979). The ISPs, teams and staff are represented by nodes, while the interdependencies are represented by links. Figure 1 shows an abstraction of an IT service network with nodes and links.

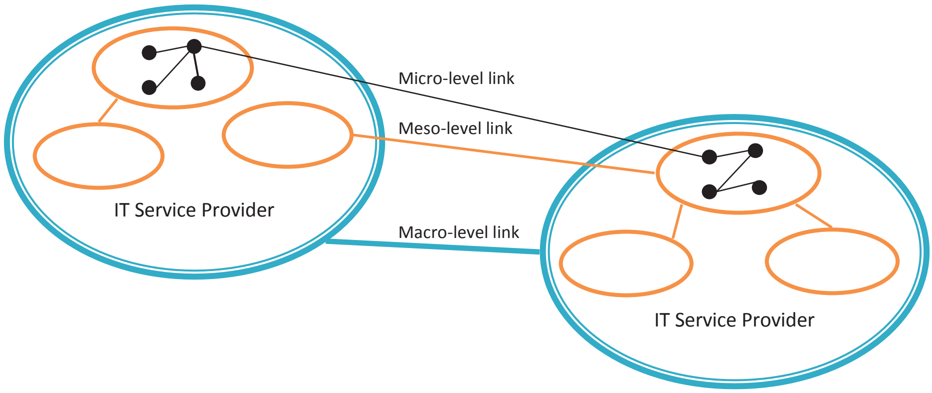

Figure 1, graphical representation of an IT service network

Macro-level nodes represent ISPs, illustrated by blue ovals. Blue lines illustrate the interdependencies between ISPs. Meso-level nodes represent teams within ISPs, illustrated by orange ovals. Orange links illustrate the interdependencies between teams. Micro-level nodes represent staff within teams, illustrated by black dots. Black links illustrate the interdependencies between staff. In order to enhance the understanding of the impact of these macro-level, meso-level and micro-level links the first research question is defined as:

RQ 1: What IT service network interdependencies affect IT delivery in IT service networks?

This research question triggered the study of literature that is related to IT service networks, being supply chain literature. In the supply chain literature Service Supply Chains (SSC) consists of multiple supply chain partners jointly delivering added value (Basole & Rouse, 2008; Sugumaran & Arogyaswamy, 2003). Supply chain partners resemble ISPs and teams in three ways: (1) Services delivered by supply chain partners resemble IT service delivery by ISPs (Yu et al., 2008), (2) both supply chain partners and ISPs have interdependencies and (3) tasks placed on backlogs of teams within ISPs resemble product stocks in supply chains. Given the resemblance between supply chains and ISPs, existing supply chain theory is utilized for this dissertation.

Research in SSCs shows that information needs to be shared between supply chain partners (Bartlett et al., 2007; Wei & Wang, 2010). Such information sharing requires that information is 'visible' to supply chain partners. The visibility of information is a well-known concept in the supply chain literature (Barratt & Oke, 2007; Bartlett et al., 2007; Caridi, Crippa, Perego, Saianesi, & Turmino, 2010a; Caridi et al., 2010b; Francis, 2008; Wei & Wang, 2010; Zhang, Goh, de Souza, & Meng, 2011). Supply chain visibility is defined by Francis (2008) as "the identity, location and status of entities transiting the supply chain, captured in timely messages about events, along with the planned and actual dates/times for these events". Based on the definition of Francis (2008) visibility is in this thesis defined as: “*the quality of known information characterizing predefined entities in a predefined IT domain*". Information visibility helps for instance interdependent partners to reduce the fluctuation of inventory by optimizing the flow of goods (Banbury et al., 2010; Bhattacharya & Bandyopadhyay, 2011). Given the similarities between IT service networks and supply chains information visibility is probably also applicable to IT service networks. The question is which information needs to be visible. Research in the supply chain industry has identified information that needs to be shared (Baltacioglu et al., 2007; Barratt & Oke, 2007) in supply chain networks. That information is not automatically equal to the information that needs to be shared in IT service networks, since supply chains are product oriented, while IT service networks are service oriented. In order to find the information that needs to be visible the following research question is defined as:

RQ 2: What information needs to be visible for IT delivery in IT service networks?

That research question triggered the question whether information visibility can be utilized for improving performance. Research in supply chain networks shows that enhanced levels of information visibility improve supply chain performance (Bartlett et al., 2007; Wei & Wang, 2010).

The performance of IT service networks can be measured in different ways. One way is using business value, which is considered a prime measure for IT performance by many (Alleman, Henderson, & Seggelke, 2003; Melville et al., 2004; Sutherland & Schwaber, 2013). ISPs deliver IT services to enable business processes that result in business value. Melville et al. (2004) study literature and model the relationship between IT services and business value that shows that the relationship is indirect and dependent on many (external) factors. Beccalli (2007) and Lin (2007) support Melville et al. (2004) regarding that indirect relationship. In this thesis the influencing (external) factors are eliminated by measuring IT service performance at the ISP side.

IT service performance of an ISP can be measured with subjective perceptions and by objective indicators. As perceptions are difficult to compare between case studies, the performance in this thesis is measured with objective indicators, allowing comparison over multiple case studies. For the objective performance indicators the common supply chain performance variables: flexibility, output, time, resources and flexibility (Cai, Liu, Xiao, & Liu, 2010; Huijgens, Solingen, & Deursen, 2014) are reused. The variable ‘*flexibility*’ refers to the service supply chain responsiveness to changes in the external environment that requires an IT service to change (Angerhofer & Angelides, 2006; Beamon, 1999). The variable '*output*' represents the delivered results against the requirements (Beamon, 1999; Gunasekaran, Patel, & McGaughey, 2004; Huijgens & Solingen, 2014; Shepherd & Günter, 2006). The variable '*resources*' represents the required resources to deliver the output (Bolstorff, 2003; Gunasekaran et al., 2004). The variable '*time*' represents the duration to deliver the output (Bolstorff, 2003; Shepherd & Günter, 2006). When the same output is achieved in less *time*, against equal *cost* with equal *resources* the performance of an IT service network is improved.

Stable IT provider networks with fixed *resource*s are subject of study. Fixed *resources* prevent interference with the variables '*time*' and '*resources*', since adding resources impacts the duration (time) and output (results) (Bentley, Jones, Atkinson, & Ferguson, 2009). IT service networks with predefined *output* is subject of study. In case of IT failures the *output* is predefined by a restored operational IT service. In case of changes the *output* is predefined by the requirements that define the changed IT service. The variable '*flexibility*' is implicitly covered by measuring the responsiveness to change with the variable time and output. The variable *'flexibility'* is therefore not taken into account as an independent performance variable.

Given the similarity between supply chain networks and IT service networks, the causality between information visibility and performance probably also applies to IT service networks. Visibility of information eases information sharing resulting in improved IT service network performance (Bartlett et al., 2007). For instance information between IT teams about their performance enables teams to influence each other's performance (Banbury et al., 2010). Such influencing activities result in improved performance of the IT service network (Viswanadham et al., 2005). In order to test the hypothesized impact of visibility on IT service network performance, the following research question is defined as:

RQ 3: To which extent does visibility of information improve the performance of IT service networks?

That research question triggered the question which other factors affect IT service network performance. Larger ISPs have many teams that deliver IT. Each of these teams has staff with specialized skills that cannot be easily shared (Paasivaara et al., 2012; Christoph J Stettina, Heijstek, & Fægri, 2012). For instance one team changes and delivers the IT servers and a second team uses the servers to host a database platform. Together the IT results in an IT service that is delivered by the ISP (Chen, 2008). The combination of specialized skills and interdependencies require teams between and within ISPs to collaborate while handling IT failures and achieving IT changes (Vlietland & van Vliet, 2014b, 2015b). Several factors that are related to such collaboration, probably impact IT service network performance.

Delivering IT services in a shorter timeframe is a measure for (IT service network) performance (Bolstorff, 2003; Shepherd & Günter, 2006) and a measure for (IT service network) Agility (Beedle et al., 2013; G. Lee & Xia, 2010). *Time,* or *duration,* is therefore identified as the performance variable for Agility in this dissertation. In order to find the collaboration related factors the following research question is defined as:

RQ 4: What collaboration related factors impede the Agility of IT service networks?

Since the collaboration related factors impede the Agility of an IT service network, the factors are defined as collaboration related issues (Vlietland & van Vliet, 2014a, 2015b). Alleviating these collaboration related issues likely improve the Agility of an IT service network. In order to test the impact of the alleviated collaboration related issues the following research question is defined as:

RQ 5: To which extent does alleviating collaboration issues improve the Agility of IT service networks?

The answer of RQ 1 and RQ 2 offer an in-depth insight into IT service networks which is used for answering RQ 3. Answering RQ 3 enhances the understanding of the impact of information visibility on IT service network performance. The knowledge about IT service networks, gained with RQ 1-3, is applied in the study of collaboration related issues in IT service networks, to answer RQ 4. The gained knowledge of answering RQ 1-4 is applied for developing a set of intervention actions for improving the Agility of an IT service network, answering RQ 5. By answering the last research question sufficient knowledge and insight has been acquired to answer the main research question.

For answering the main research question, the answers of RQ 1-5 are combined and generalized to develop a framework that improves the Agility of IT service networks. Such framework development is supported by Soundararajan and Arthur (2009) that argue that Agile practices need to be structured. The developed framework in this thesis contains intervention actions for improving IT service network Agility. Each of the intervention actions enhances IT service network Agility. The intervention actions have intended change as organizational change paradigm (Weick & Quinn, 1999). Intended change is an organizational change paradigm that impose interventions actions to change a subject to a desired state. Existing organizational change literature is used as point of reference for developing the intervention actions (Cummings & Worley, 2014; Gersick, 1991; Hannan & Freeman, 1984; Kolb, 1984; Qumer & Henderson-Sellers, 2008; Weick & Quinn, 1999; Yamakami, 2013). The intervention actions are packaged into a framework. With that framework the main research question: "How to improve the Agility of IT service networks?" is answered.

1.4 Research methods

The presented research in this dissertation performs all phases of the empirical cycle (Groot & Spiekerman, 1969): observation, induction, deduction, testing and evaluation. O*bservation* puts emphasis on the collection of empirical data. *Induction* aims to create propositions on the basis of the observations that explicate the observations. In the *deduction phase* propositions and additional related work are developed into hypotheses with dependent and independent variables to predict empirical contexts. During the *testing* phase the hypotheses are tested by collecting and analyzing new empirical data. *Evaluation* interprets specified hypotheses and theories and is interpretative by nature by generating new ideas for research.

The research methods in this dissertation are exploratory and confirmatory case studies. Exploratory case studies investigate distinct phenomena characterized by a lack of detailed preliminary research, especially formulated hypotheses that can be tested, and/or by a specific research environment that limits the choice of methodology (Mills, Durepos, & Wiebe, 2009). This form of case study is applied as a preliminary step for the development of a causal or explanatory research design. Confirmatory case studies test hypotheses, an outcome of predictions that are made before the measurement phase (Mills et al., 2009).

The data collection and analysis techniques include archival record study, surveys, interviews, transcriptions, coding techniques and focus groups. Interview results are analyzed by transcribing the recorded data. A three step qualitative coding technique is used to analyze transcripted data (Dul & Hak, 2012; Yin 2009). Subsequent quantitative analysis is performed on the code quantity in information categories (Saunders, Lewis, & Thornhill, 2009). Focus groups are setup after the events or experiences (Krueger & Casey, 2008) to evaluate the impact after the improvement interventions.

Table 3 shows (1) the overview of chapters, (2) the research questions that are covered in each chapter, (3) the segment of the empirical cycle that is performed, (4) the research design and (5) the used data collection and analysis methods in each chapter.

Table 3, Research overview

Chapter	Research questions	Empirical cycle	Research design	Data collection and analysis
1. Introduction				
2. Information sharing for effective IT incident resolving in ISP networks	RQ1 RQ2	Observation Induction	Case study	Archival record study Interviews Transcription/coding
3. Improving IT incident handling performance with information visibility	RQ3	Deduction Testing Evaluation	Case study Survey	Archival record study Interviews Correlation analysis
4. Towards a governance framework for chains of Scrum teams	RQ4	Observation Induction Deduction	Case study	Archival record study Interviews Transcription/coding
5. Delivering business value faster by sets of codependent Scrum teams	RQ5 RQ6	Deduction Testing Evaluation	Case study	Archival record study Interviews/focus groups
6. Improving the Agility of IT service networks	Main RQ	Induction	N/A	N/A
7. Conclusions				

1.5 Chapters overview

Chapter 2: Information sharing for effective IT incident resolving in ISP networks
IT service networks need information to resolve IT incidents in their delivered IT services. The objective of the research is to identify the set of information that needs to be visible within IT provider networks to effectively resolve IT incidents. To this end, we conducted an inductive case study in a network of nine interdependent IT service providers. We found that the required information is distributed over multiple technological stores and that operational IT staff in the network need visibility over these technological stores. Operational staff also needs visibility over the social network of incident handling staff, given the tacit nature of the required information. We therefore premise that better information sharing and enhanced knowledge reuse in the service network has a positive impact on incident handling in IT service provider networks. The main contribution of this chapter is a structured set of information types that positively impacts IT incident handling performance in the IT service network. That structured set has been packaged into a conceptual model, answering research question RQ1 and RQ2.

Chapter 3: Improving IT incident handling performance with information visibility

We hypothesize that the knowledge IT teams in an ISP have of the agreed upon and realized incident handling performance of themselves and other teams will impact their performance. We tested this hypothesis at a large financial institute, using log data from the IT service management application and a survey to measure the knowledge of teams. We found (1) a significant positive correlation between incident handling performance of a team and the knowledge a team has of its own performance; (2) no correlation between the knowledge of agreed upon performance and realized performance within a team; (3) that teams have very little knowledge of agreed upon or realized performance of other teams; and (4) that improving the knowledge a team has of the agreed upon and realized performance of that team and dependent teams results in higher incident handling performance. The results show that increasing information visibility within and across teams in large IT providers is one way to improve incident handling performance. The results answer RQ3 in an intra-ISP context.

Chapter 4: Towards a governance framework for chains of Scrum teams

IT functionality in large enterprises is typically delivered by a portfolio of interdependent software applications involving a chain of Scrum teams. In this study we identify the collaboration related issues in a chain of Scrum teams. We used a qualitative approach with transcripted interviews from three case studies that were coded and analyzed to identify the issues. We identified six collaboration issues; coordination, prioritization, alignment, automation, predictability and visibility. These six issues answer research question RQ4. The synthesis of the issues with existing theory resulted in nine propositions. These nine propositions have been combined into a conceptual model. We used the conceptual model as a starting point to develop the Agile frameworks.

Chapter 5: Delivering business value faster by sets of codependent Scrum teams

In this study we develop a governance framework that packages five empirically tested intervention actions that alleviates the collaboration issues in sets of codependent Scrum teams. 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. 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. The main contribution of this chapter is a governance framework for sets of codependent Scrum teams that support a value chain.

Chapter 6: Improving the Agility of IT service networks

Agility in networks of IT service providers helps to swiftly adapt interdependent IT services to changing business needs. In this chapter a set of intervention actions is developed to improve the Agility of these IT service (provider) networks. The intervention actions are based on Agile literature, organizational change theory and empirically confirmed collaboration related factors in Agile IT service networks. The intervention actions are packaged into an Agile 5+1 intervention action framework. The result is an Agile 5+1 framework to improve the Agility in networks of IT service providers.

1.6 Origin of chapters

The research in this dissertation has been published previously at conferences and in journals. The publications are included in subsequent chapter as-is, with the exception of some minor corrections. The research reported in this dissertation has been performed by Jan Vlietland as the prime researcher.

Parts of chapter 2 have been previously published as: (Vlietland & van Vliet, 2013, 2014b):

– Vlietland, J., & van Vliet, H. (2013). *Visibility and Performance of IT Incident Handling: A Control Theory Perspective.* Proceeding of the Joint Conference of the 23nd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IWSM-MENSURA), Ankara, IEEE Computer Society, 203-212.
– Vlietland, J., & van Vliet, H. (2014). Improving IT incident handling performance with information visibility. *Journal of Software: Evolution and Process*. doi: 10.1002/smr.1649

Parts of chapter 3 have been previously published by JSME as: (Vlietland & van Vliet, 2014c)

– Vlietland, J., & van Vliet, H. (2014c). Information sharing for effective IT incident resolving in IT service provider networks: A financial service case study. *Journal of Software: Evolution and Process.* doi: 10.1002/smr.1697

Parts of chapter 4 have been previously published as: (Vlietland & van Vliet, 2014a, 2015b)

- Vlietland, J., & van Vliet, H. (2014a). *Alignment Issues in chains of Scrum teams.* Proceedings of the 5th International Conference on Software Business, Cyprus, *C. Lassenius and K. Smolander (Eds.), Springer LNBIP 182,* 301-306.
- Vlietland, J., & van Vliet, H. (2014d). Towards a governance framework for chains of Scrum teams. *Journal of Information and Software Technology, Volume 57* (January 2015), Pages 52–65. doi: 10.1016/j.infsof.2014.08.008

Parts of chapter 5 are currently in process of being published by JSS as: (Vlietland, van Solingen, & van Vliet, 2015):

- Vlietland, J., van Solingen, R., & van Vliet, H. (2015). Delivering business value faster by sets of codependent Scrum teams: a governance framework. *Journal of System and Software*.

Parts of chapter 6 have been previously submitted to ICSOB 2015 as: (Vlietland & van Vliet, 2015a):

- Vlietland, J., & van Vliet, H. (2015b). *Improving the Agility of IT Service Networks.* Proceedings of the 6th International Conference on Software Business, Portugal.

Chapter 2

Information sharing for effective IT incident resolving in IT service provider networks

Information technology enabled financial services are typically delivered by a network of interdependent IT service providers. Such networks need information to resolve IT incidents in their delivered IT services. The objective of this research is to identify the set of information that needs to be visible within IT provider networks to effectively resolve IT incidents. To this end, we conducted an inductive case study in a network of nine interdependent IT service providers. We found that the required information is distributed over multiple technological stores, and operational IT staff in the network need visibility over these technological stores. Operational staff also needs visibility over the social network of incident handling staff, given the tacit nature of the required information. We therefore premise that better information sharing and enhanced knowledge reuse in the service network has a positive impact on incident handling in IT service provider networks. The main contribution of this chapter is a structured set of information types that positively impacts IT incident handling performance in the IT service network, packaged into a conceptual model.

2.1 Introduction

Business services in large enterprises are enabled by a combination of IT services, delivered by multiple IT service providers. Each of the IT service providers is contractually accountable for one or more IT services (Allen & Chandrashekar, 2000; Niessink, 2001; Susarla, 2003) and uses IT services from other IT service providers, resulting in a network of interdependent IT service providers (Vlietland & van Vliet, 2013). The interdependencies between the IT service providers have a critical nature. A disruption in one of the IT services immediately disrupts the interdependent IT services, resulting in a disrupted overall IT service delivered to the business partner. Failure of such IT service results in failing business services towards clients, potentially leading to extensive financial damage (Oppenheimer, Ganapathi, & Patterson, 2002).

The financial payments industry is a typical example of having such critical interdependencies. A disruption in the payments processor for instance immediately disrupts the payment services of dependent banks (Cheney, Hunt, Jacob, Porter, & Summers, 2012). The critical interdependencies make each of the IT service providers a single-point-of-failure risk in the network of IT services (Nagasubramanian & Rajagopalan, 2012). Next to the delivery of technology an IT service provider acts on events, for instance during IT service disruptions (Bardhan, Demirkan, Kannan, Kauffman, & Sougstad, 2010; Jantti, 2011). Hence, a delivered IT service consists of information technology and additional human activities (Ellram, Tate, & Billington, 2007; Peppard, 2003). The interdependency between IT services results in collaborating IT service staff between IT service providers to jointly resolve disruptions (Jäntti, 2012a; Vlietland & van Vliet, 2013). Effective collaboration is therefore critical to handle IT disruptions and restore the business service swiftly.

We argue that at least two factors obstruct such inter-provider collaboration. First, IT workflow processes are typically implemented on an intra-provider rather than an inter-provider level. This argumentation is grounded in the ITIL literature that targets IT organizations instead of IT service provider networks (OGC, 2007; van Bon et al., 2007). Second, involved staff from different IT service providers typically resides in different locations, disabling face-to-face communication. These two factors introduce collaboration and information sharing impediments, obstructing effective handling of IT disruptions (Shachaf, 2008).

In this study we follow the ITIL standard and define an IT disruption as an IT incident, which is: *“an event which is not part of the standard operation of a service and which causes or may cause disruption to or a reduction in the quality of services and customer productivity”* (OGC, 2007; van Bon et al., 2007). An IT incident that is discovered by IT staff is typically registered in and tracked with an IT Service Management (ITSM) application. Registered IT incident information in an ITSM application includes an IT incident description, the incident registration timestamp and incident resolving timestamp. The registration and resolving timestamp are used to determine the incident handling duration. ITSM applications also use registered IT incident information to generate aggregated reports about the average realized incident handling performance. IT incident handling performance is typically contractually agreed. An IT incident that occurs needs to be handled within the maximum duration that has been contractually agreed. The term contract in this chapter is considered equivalent to a service level agreement (SLA).

Our prior research targeted the impact of performance information visibility on IT incident handling performance in IT provider network settings (Vlietland & van Vliet, 2013) in which we found that enhancing the visibility of IT incident handling performance by members of an IT service team positively impacts the performance of that team (Vlietland & van Vliet, 2014b).

As that research was limited to enhancing the visibility of performance information, the research question emerged whether other information also positively impacts IT incident handling performance. In this chapter we answer that research question, by identifying the broader set of information that needs to be visible for effective IT incident handling. The research is performed in a case study involving 9 interdependent IT providers in the payment industry.

We found that existing information is scattered over multiple technological and cognitive stores, which results in hardly accessible, highly needed IT incident handling information. In addition, as part of the information has a tacit nature, visibility over the human network supports accessibility to cognitive and (indirectly to) technical stores.

The main contribution of this chapter is a structured set of information types that positively impacts IT incident handling performance in the IT service network, packaged into a conceptual model. That model can be used to improve IT incident handling in IT provider networks and minimize financial damage due to failing business services.

The remainder of this chapter is organized as follows. Section 2.2 explains the used model for the study. Section 2.3 covers related work. Section 2.4 explains the research design used for the case study. Section 2.5 covers our results. Section 2.6 discusses the results, derives propositions from these results and uses a theoretical lens to explain the propositions. Section 2.7 elaborates on the threats to validity and limitations of the research. Section 2.8 concludes the research, deduces the implications and suggests future research avenues.

2.2 Model building

In this paragraph we build the conceptual model that is used as foundation for the research design. The conceptual model is based on control theory. Our prior research used control theory to theorize the relationship between the contractually based incident handling goal, IT incident handling activities and realized incident handling performance (Vlietland & van Vliet, 2013, 2014b). We found in that research that sharing incident handling performance information within an IT provider positively impacts IT incident handling performance of that IT provider. Control theory consists of three fundamental concepts, as shown in Figure 2. The first concept is goal setting; in our case, the goal is predefined by the contract. The second concept is feedback; in our case feedback of the achieved performance level. The third concept is the function (C) that compares output and input; in our case comparison of actual performance and performance goal. The compared result enables the selection of (adapted) action of involved staff to reach the incident handling goal (Andrei, 2006; Forssell & Powers, 2009).

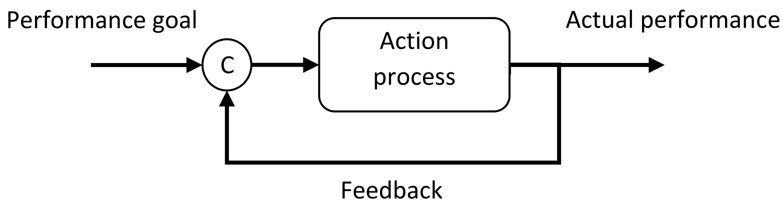

Figure 2, Control theory model

Our prior research used visibility as catalyst for feedback in the control theory model. Visibility of information results in accurate comparison of actual performance and the performance goal. However in that research visibility was limited to performance information (Vlietland & van Vliet, 2014b). In the current research we premise that visibility of other information also enhances incident handling performance (Gregory, Beck, & Carr, 2011) as information is also directly required in the action process next to the performance feedback into the comparison function.

Decision theory, which is related to control theory, uses such a broader view of information to accurately decide on decision making and action taking. Decision theory explains the relationship between action, a set of information and decision making to achieve a set goal (Duffy, 1993; Goodwin & Wright, 2007). A related concept explaining the relationship between information and the action process is the Observe, Orient, Decide Act (OODA) loop of John Boyd (Fadok, 1995). The concept, which is related to control theory, was originally applied to military operations, using raw information to orient the operation for decision making and action taking.

We premise that the action process in our case requires information from different providers in the IT service network, which is supported by academic work in the IT outsourcing (Blumenberg, Wagner, & Beimborn, 2009; Hamid & Salim, 2011) and supply chain management industry (Rashed et al., 2010; Sahin & Robinson, 2005).

In the sequel, the term 'tiers' is used to indicate the distance in terms of the number of edges between IT providers (Caridi et al., 2010a). A first tier relationship indicates a direct interdependency (edge) between two IT providers (nodes). A second tier relationship indicates two IT providers that are connected via an intermediate IT provider. Hence, the tier level indicates the minimum number of edges that information has to travel between two nodes. Certain information might be needed from the zero-tier, which is the IT provider that takes incident handling action. Other information might be needed from first tier, which are directly connected IT providers. Information might also be needed from the second tier, which is an IT provider that delivers IT services to a first tier IT service provider.

Figure 3 shows the resulting conceptual model. Zero, first and second tier information is required in the incident handing action process. The feedback loop is not included in the model as feedback and the comparison function of control theory are applicable to performance information only.

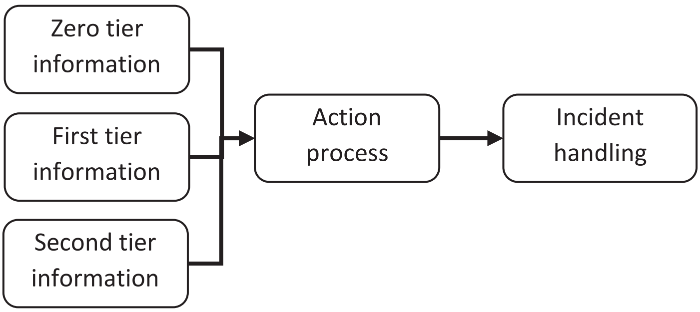

Figure 3, Information – action – performance model

2.3 Related work

A search for related work on information required to effectively resolve incidents in IT provider networks revealed that academic literature regarding this research area is scant. We found and discuss the closest three related research areas: (1) IT service management frameworks, (2) IT sourcing and (3) service supply chain management.

The service operation cluster of the IT Infrastructure Library (ITIL) version 3 framework includes the IT process incident management. ITIL incident management covers identification, prioritization, investigation and solving incidents to restore an IT service (OGC, 2007; Talk, 2013). ITIL Incident management specifies the following information items: incident management rules, incident reports, incident prioritization guidelines, escalation information, incident details and incident status information (Jäntti, 2012a; OGC, 2007; Talk, 2013; van Bon et al., 2007). The related ITIL process service level management refers to the information items: service level agreements, service level reports and existing (infrastructural) IT configuration. A related framework is the Control Objectives for Information and related Technology (Cobit) framework. Cobit targets first line incident management and to a limited extent second and third line support (ITGI, 2007). The covered activities are registering, communicating, dispatching and analyzing incidents of service desks. Both ITIL and Cobit use an intra-provider, not an inter-provider perspective, which is insufficient to extract a reliable set of information items for inter-provider constellations.

An inter-provider perspective is used in the sourcing area although the perspective is limited to dyadic relationships. We did not find any academic literature in the sourcing research area that uses a network perspective. Blumenberg et al. (2009) studied knowledge transfer processes in IT outsourcing relationships, in which their dyadic oriented research mentions the information items: formalized procedures, rules, contracts, SLAs and technical terms. Savić (2008) covered IT incidents in operational outsourcing constellations, although the paper does not include information items that impact IT incident risks. Bartolini et al. (2006) of the Hewlett Packard research lab conducted IT incident handling research, though no information to be shared in service networks is mentioned. The sourcing literature seems not to provide a set of shared information items for IT incident handling within IT provider networks.

The third area we studied is the service supply chain research area. Service supply chains (SSC) also require collaboration and therefore information sharing between providers. The SSC perspective was popularized by Ellram et al. (2004). In their paper they compare the applicability of supply chain models in the services area; however the IT service industry is not covered in their research. Other literature in the service supply chain area does not cover IT provider networks either, let alone a specification of the required information to be shared. One close non-IT match is the paper of Zailani and Kumar (2011), which reports a literature study on service supply chain information flows. The paper mentions the attributes customer request and service planning, however are not necessarily attributes for IT incidents and operational IT plans. We therefore conclude that literature in the service supply chain area does also not answer the main research question.

2.4 Research design

We performed qualitative (Saunders et al., 2009) case study research (Dul & Hak, 2012) at a multinational financial service provider and its supplying IT organizations to gain an in-depth understanding of the used and needed information and get our main research question answered. We developed the following set of research questions for our study:

- What are the IT providers in the selected IT provider network?
- What are the stakeholders in the selected IT provider network?
- Which of these stakeholder roles are highly involved in daily incident handling?
- Which typical incident handling activities are performed by these roles?
- What types of information is shared and/or required by these roles to perform these activities?
- What information is shared and/or required by each of these roles?
- What information is shared and/or required on a first and/or second tier level?

The used research design is based on Eisenhardt (1989), split into two stages as shown in Figure 4. Stage A targets the IT provider network and role mapping, laying the foundation for stage B, that aimed to identify the used and needed information.

The research starts with case selection (step 1), in which we selected a single direct debit card transaction performed by a customer in a domestic retail shop. The scenario requires instant data processing from multiple IT providers. Each of the provided IT services is critically important for successful payment processing (TFSC, 2011). The interdependent IT providers are identified in step 2, by studying archival records and conducting semi-structured interviews at IT providers of the network (Myers & Newman, 2007). The prepared open-ended questions are used to guide the interviews (see stage A questions in the appendix). The interviews are annotated in an interview log and audio recorded for verification purposes. Next to these interviews we use supplementary interviews with candidates working in the IT provider network to collect the information via a snowballing technique. The criteria to participate in the interviews are: (1) having a role in the IT provider network, (2) having information about others in the network, (3) being recommended as interviewee by a manager in the hierarchical structure, (3) being an author of an applicable document (e.g. SLA) and/or being part of the hierarchical structure. The interviews are annotated in interview logs. In addition a data collection log is used to record the activities, the time stamp and the results.

The interview recordings, interview logs, data collection logs and archival records are used to answer the open-ended questions of stage A, shown in the appendix. Using both archival records and semi-structured interviews enables data triangulation (step

3). The network is subsequently mapped in step 4, based on the found interdependencies. Steps 2 till 4 are repeated until all existing interdependent IT providers are included. The involved IT service operation roles are mapped in step 5. Multiple roles are used to enable within-case data analysis. Subject matter experts working in the network are consulted to increase the validity of the mapped network and mapped roles (Gibbert & Ruigrok, 2010).

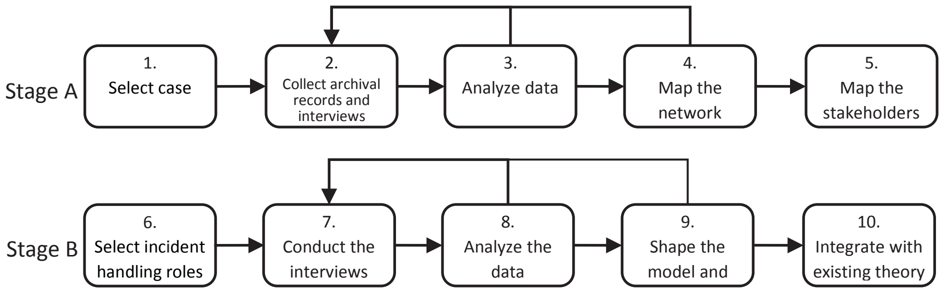

Figure 4, Research design

In the first step of stage B (step 6) the highly involved IT incident handling roles are selected and subsequently interviewed in step 7. Prepared open-ended questions are used to guide the interviews, as shown in stage B of the appendix. The interviews are digitally recorded for content- and construct validity purposes (Mentzer & Flint, 1997). Each interview is analyzed before the next interview, which allows adaptation of the interview script for successive interviews. The iterative approach mitigates the potential problems of Myers and Newman (2007), although the changes in the script were limited.

The setup of all interviews is based on the dramaturgical model, using the metaphor of a theatre to explore social life (Myers & Newman, 2007). The dramaturgical model is based on the general theory of Goffman (1959) that sees social interactions as a drama, with actors that perform in a variety of settings using a script that guides behavior. Both the interviewer and the interviewee play an acting role. The researcher plays the part of an interesting interviewer, the interviewee the part of a knowledgeable person. During the interview a delicate balance is kept between providing direction and getting unbiased answers. Table 4 shows the potential problems of interviews as summarized by Myers and Newman (2007) and our mitigations. The objectives and topics of the interview are set at the beginning of the interview.

Table 4, Potential problems during interviews

Potential problem	Mitigation
Artificiality of the interview	Approach the interview as an interesting conversation. Use a limited structured approach and move gently back to the topic if required.
Lack of trust	Communicate the background of the research, the benefits for the interviewee and the confidentiality of the interview results.
Level of entry	Prepare the interviews by first gaining sufficient knowledge and insight during stage A of the empirical study.
Elite bias	Map the network and the involved roles to ensure that the right IT staff is selected for the interviews.
Constructing knowledge	Triangulate the analysis with archival records, include verifying questions, validate the results with subject matter experts and code one of the transcriptions with a peer.
Ambiguity of language	Include verification questions to verify the interpretation of the used terminology. Mirror by using the words and phrases of interviewees.

The interview results are analyzed in step 8 by transcribing the recorded data. Qualitative analysis techniques are used to analyze the transcripted data (Dul & Hak, 2012; Yin 2009). The qualitative analysis starts with identifying and tagging quotes in the transcriptions (Saunders et al., 2009). The quotes are identified by looking for words and phrases that answer the open-ended questions. The quotes are subsequently tagged with open codes. The open coding process is iteratively performed (Saldaña, 2012). In case a quote applies to more than one open code all applicable codes are linked to that quote. For instance the quote *'Service level agreements are used to share information between roles'*, is related to the codes *'ServiceLevelAgreement'* and *'InformationServiceLevelAgreement'*. Open coding proceeds line-by-line, and proceeds until patterns emerge. These patterns are formalized by grouping the open codes into categories. The categorizing process is done in three ways. First, codes are grouped by prefixes. For instance all codes which are related to information are grouped by the prefix '*Information*'. Second, codes are grouped in code families. For instance the information related codes needed by an '*Incident Manager*' are placed in the code family '*InformationIncidentManager*'. Third, supercodes, predefined queries to retrieve a set of codes, are created for quotes that are related to more than one code or code family. For instance information that is related to the first tier is labeled *'first tier'* and can thus be queried with a supercode. The coding process of the first two interviews is independently performed by a peer and compared with the results of the researcher to minimize analysis bias.

Data collection continues until a new transcription does not significantly contribute to knowledge and insight (Dul & Hak, 2012). Significantly contributing is quantified by determining whether an additional interview results in more than 5% new or modified codes, which is in line with Sandelowski (1995) and Marshall (1996).

Axial coding is subsequently performed to verify the relationship between information (the cause) that results into effective action (the consequence). After the axial coding process has been completed the grouped information and action codes are clustered into the main categories. These main categories are then clustered to concepts to detail the conceptual model (Birks & Mills, 2011). All steps of the data analysis are recorded in Atlas TI, a CAQDAS package (Gibbert & Ruigrok, 2010; Saunders et al., 2009).

A quantitative analysis is performed on the number of codes in each information category for first- and second tier information and for each interviewed role (Saunders et al., 2009). The number of codes in the first- and second tier is used as an indication of the importance of sharing that category of information within the first tier and/or second tier staff. The number of codes for each interviewed role is used as an indication of the importance of the information category for that role.

In step 9 the conceptual model is detailed and the propositions are defined. During step 10 academic literature is consulted to compare the propositions with existing theory.

2.5 Results

The result section elucidates the results, including typical quotes that enrich understanding. The section is organized as follows. The network of participating service providers and the existing incident handling roles are discussed in subsection 2.5.1. Subsection 2.5.2 discusses the information that all maintenance engineers involved in daily incident handling and incident managers of the most central service provider use and require from themselves (zero tier) or adjacent service providers (first tier information). In a similar vein, subsection 2.5.3 discusses the second tier information that is used and required in the most central service provider. Subsection 2.5.4 discusses the information storage, retrieval, and transfer processes that support information sharing in the provider network. Finally, subsection 2.5.5 discusses the overall conceptual model.

2.5.1 Overview

We identified three networks: the contractual, the technical and the human network. The contractual network consists of the contractual interdependencies between the IT service providers that deliver the IT services. The technical network consists of the technical interdependent IT systems, delivered by the service providers. The human network consists of incident handling staff that collaborate and share information. A total of 15 planned interviews and 86 supplementary interviews were conducted to collect the information.

Figure 5 shows the mapped network based on the contractual agreements. The full network consists of nine IT service providers. The business partner of the financial service provider has a contract with an internal IT service provider that in turn has a contract with an internal IT hosting provider. The IT hosting provider uses IT infra services from four external IT providers. The business partner has contracts with three external IT providers. Each contract includes a description of the services, and the maximum duration of IT incidents. The IT providers are categorized in SaaS, Paas and IaaS layers. The arrows illustrate the contractual flow of delivery.

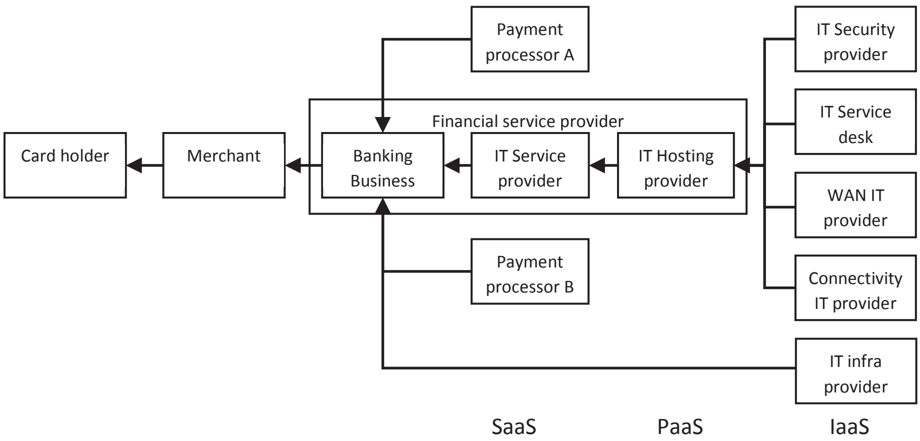

Figure 5, IT provider network map of the research domain

The technical IT network is the second identified network. The technical network is shaped by the technical interdependencies between the IT systems. An electronic payment at a merchants store for instance is sent via the merchant point-of-sales machine to payment processor A, which sends the payment to payment processor B, which sends the payment to the IT service provider. The example shows that the technical information flow differs from the contractual delivery flow (Figure 5).

The third identified network is the human network consisting of collaborating incident handling staff. Each involved member has a defined role in the IT incident handling process, predefined in a standardized role description. The role map in Figure 6 shows five IT service providers, one in each column. Vertically the figure shows for each IT service provider the involved IT incident handling roles, sorted from operational at the bottom to tactical at the top. Note that the card holder, the merchant, the IT infra provider, Payment processor B, the WAN IT provider and the Connectivity IT Provider are excluded from Figure 6 for simplicity reasons.

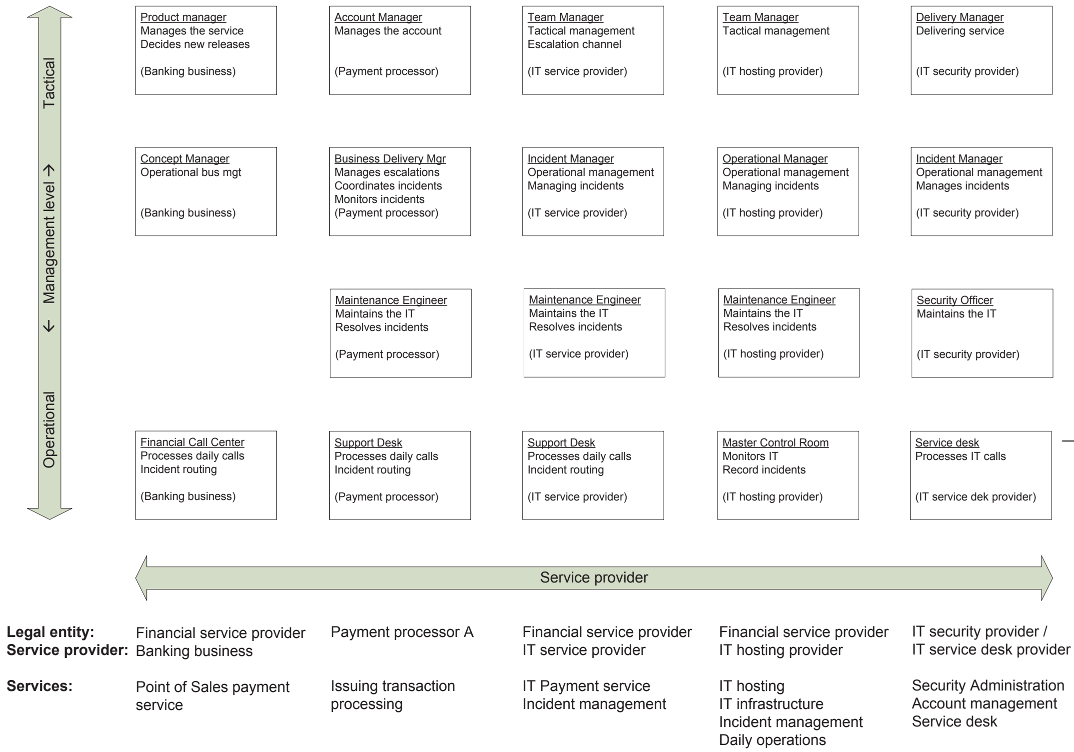

Figure 6, Involved IT service provider roles

Communication between roles varies between IT incidents as each IT incident might occur in different parts of the IT service network, involving different roles and requiring a unique mix of information from different origins. Two incident handling scenarios illustrate possible communication flows that exist during incident handling.

In the first scenario an IT incident is identified by the *Merchant.* The *Merchant* contacts the *Financial Call Center,* positioned at the bottom-left of Figure 6 to report the incident. The *Financial Call* Center subsequently contacts the *Support* Desk of *the Financial Service* provider to report the incident. The *Support Desk* registers the incident in the ITSM application. When the incident gets registered the *Incident Manager* of the *IT service provider* receives a notification of the registered incident through the ITSM application. This triggers the *Incident Manager* to start monitoring the incident handling process. The *Incident Manager* subsequently involves *Maintenance Engineers* from the applicable *IT service providers* to analyze the IT incident. In this scenario the *Maintenance Engineers* discover that payment messages are not received from the *Payment processor*. The *Support Desk* of the *Payment processor*, handling calls, is subsequently contacted to determine whether the incident is caused by the *Payment Processor*.

In the second scenario the *Master Control Room* of the *IT hosting provider* notices an IT incident via an IT service event monitoring application and registers the incident in the ITSM application. The *Incident Manager* is triggered and involves *Maintenance Engineers* to analyze the incident. As the incident in this case seems to be caused by a malfunction in the security service, the *Incident Manager* contacts the *Operational Manager* of the IT hosting provider. The *Operational Manager* subsequently takes care of incident registration at the *IT security provider,* which triggers the *IT security provider* to start IT incident handling after registering the IT incident in their ITSM application.

Data analysis showed that maintenance engineers and incident managers are highly involved in IT incident handling. Incident managers manage the incident handling process by making sure that all incidents are handled within the contracted service levels. Maintenance engineers analyze and resolve incidents. The incident managers and maintenance engineers roles as shown in Figure 6 were specified in detail by role descriptions based on the ITIL standard.

We interviewed all maintenance engineers and incident managers working in the IT service provider of the financial service provider delivering the selected financial service, since the provider is centrally positioned in the network thus having the largest number of second tier provider dependencies.

Interviews with the incident managers revealed the following typical incident management activities:

- Monitor the duration of open incidents
- Analyze incidents
- Allocate incident to maintenance engineers
- Coordinate incident handling
- Escalate to management in case service levels are at risk
- Distribute daily incident lists including incident handling performance
- Communicate to stakeholders
- Report aggregated incident handling performance

Interviews with the maintenance engineers revealed the following maintenance engineering activities:

- Monitor IT system operation
- Analyze incidents

– Resolve the incident and restore the IT service
– Verifying (sample wise) incident registration quality in the ITSM application
– Maintaining procedures and IT system documentation

The interviews show that the incident handling process is complex, difficult, involves many stakeholders and is frequently time consuming. The following quote illustrates a typical situation during a high priority incident:

> *"I received a phone call that the payments are not being processed. As of that moment you start looking for the cause of the issue. We discovered that the first signs showed up hours before but it takes time to gather everybody. During the subsequent conference call with all involved IT departments and incident managers we were unable to determine the cause, however by reasons that are not known the payments process was restored a few hours later. We later found an error code in a log file although we have not been able to find the root cause".*

For high priority incidents a temporary unique task force with subject matter experts is assembled to resolve the IT incident. The task force is unique as each incident involves different subject matter experts, depending on the nature of the incident.

2.5.2 Required zero- and first-tier information for incident handling activities

This subsection presents the qualitative overview of zero and first tier information, used or needed by incident managers and maintenance engineers. The next subsection presents the qualitative overview of second tier information.

The presented overviews are sorted in the information type contractual, technical and human. Within each type the information is presented from highest to lowest importance, based on the number of quotes (see Table 5). Information categories that are not considered necessary are not covered in the two subsections. First tier information that needs to be visible on a second tier level as well is covered in the next subsection and therefore not covered in this subsection.

The quotes in this subsection and the next subsection are tagged with the information categories from the conceptual model in Figure 7 to support understanding of the result section. Information categories tagged with '(-)' are perceived as missing information.

C3: IT Service level (-)
Contractually based IT incident information is typically stored in the ITSM application and used to track the performance of the incident handling process against the agreed service levels. Registered IT incident information is often insufficient for the incident manager to analyze the incident. Incident managers lack proper understanding of the business impact of the incident, requiring incident managers to collect information from maintenance engineers to determine the contractual impact:

> *Q1: "Maintenance engineers are for me the source to determine the priority",* incident manager
> *"We often need to align with others to determine what is exactly going on",* incident manager

IT service specifications, which are part of the service catalog, are used as benchmark to compare the actual performance with the specified contractually agreed performance.

> *Q2: "What are the parameters of the technical service, what is the uptime of the service and what are the performance agreements of the service. For instance how many transactions should the service process and what is the response time of the service?",* incident manager

T5: Design of IT system (-)
Besides technical information stored in the ITSM application technical information is also used from other stores, to determine for instance which hardware and/or software components are affected by the incident. The interviews show that such information is often perceived as outdated and conflicting:

> *Q3: "Documentation often gets outdated and as a result useless within 6 months. Moreover it is hardly possible to get the latest version of documents, as we usually work with documents that have been delivered by projects and are ineffectively versioned. Documented information in such state is very annoying during incident analysis activities in the middle of the night",* maintenance engineer

System logs are also a technical source of information to determine the root cause of incidents. Maintenance engineers express their need to analyze production logs, while they have no access to such production information. Their need transcends the own IT provider, as maintenance engineers also express their need to have access to logs of interdependent IT providers:

Q4: "We do not have access to production data that would really help to analyze incidents. Sometimes we receive complaints that have roamed around for hours because nobody sees the relationships between de symptoms", maintenance engineer

2.5.3 Required second tier information for incident handling activities

This subsection presents the qualitative view of second tier information that is used or needed by incident managers and maintenance engineers. The presented overview is categorized in technical, human and contractual information. The subsection ends with change related information from a cross sectional network perspective.

C1: Incident in IT service (-)
Incident managers in particular perceive a need for sharing contractually based IT incident management information throughout the chain, while the ITSM application lacks possibilities for sharing such incident related information between IT providers:

Q5: "Before the IT security provider is allowed to spent time on handling the IT incident the IT security provider needs to duplicate the IT incident in their ITSM application. However such duplication is hard as information gets lost over the involved links in the chain", incident manager

Q6: "Our supplying IT service provider uses a different ITSM application that is connected via an interface. This interface is error-prone. To route incidents without errors we need to fill out our ITSM application in a very specific way. The service desk is the only party which is able to do that", incident manager

Incident managers expressed that determining business (end user) impact of an incident against the contract is also problematic, which causes misjudgments during impact analysis, as shown by the following quote:

Q7: "Maybe the internal business partner is also not aware of the severity as potentially 10.000 banking customers can be impacted but only 100 customers have performed this specific action", incident manager

C2: Network of IT services (-)
Information about (the overview of) the technical network and their position in the network is needed by both incident managers and maintenance engineers, to be able to interpret an incident:

Q8: "A maintenance engineer should understand the network and his position in the network… the bigger picture… which suppliers are involved. Such overview *helps to understand incidents as not everything is an incident",* incident manager

T1: Technical system process (-)

On an intra-provider level service monitoring applications are used to monitor the service performance of the service provider. As monitoring is restricted to intra-provider level maintenance engineers expressed their need for having inter-provider overview monitoring capabilities over all components:

> *Q9: "We monitor the IT services very closely. Continuously inspecting the monitoring screens and respond immediately if something weird happens",* maintenance engineer

T2: Change in a critical IT system (-)

The cross sectional view on changes in the networks showed that particular IT change related information is considered essential in reducing IT incidents: interviewees stated that 75 - 90% of the incidents are caused by technical changes. Having an overview over all planned and deployed technical changes in the network enables staff to determine incident causes and impact. Both incident manager and maintenance engineer expressed their need for visibility over all technical changes in the network:

> *Q10: "Often we know what is going to happen within our own organization, but not what is happening at interdependent IT service providers. I am totally unaware what is happening at most of the infra providers, for instance. It would really help to have access to their IT change calendars, to understand their planned changes for the coming weekend",* incident manager

Details about IT changes including test results is considered essential information to determine planned IT change impact and consequential IT incidents. Such missing information disables providers to prepare for possible incidents and proper decision making about IT change deployment in production:

> *Q11: "We would like to have a proper view over what has been tested and whether our acceptance criteria have been met. Under commercial pressure management often decides to deploy changes into production while nobody has a proper overview over the consequences",* incident manager

T3: Change in technical capacity (-)

Also unexpected change of payment transaction load leads to IT incidents, which can be prevented in case such information is known upfront:

Q12: "Product management releases a product to a new market segment. Obviously such change has consequences for the capacity of the system. I consider it important that we know what the impact is of such change", incident manager

T4: Network of IT systems (-)
Only high-level technical configuration-items are registered in the ITSM application and the ITSM application is accessible only within the IT service provider. Interviewees therefore perceive a lack of relation understanding between the IT systems and IT components in the technical network:

Q13: "We need to know where a transaction starts and ends, and which interfaces & systems are used. With such information we are able to determine the affected groups during an incident. Information about the technical network is very important", maintenance engineer

H1: Network of human resources (-)
The human network is formed by collaborating staff in the involved IT service providers. The network is utilized for information exchange during incident handling. Accurate contact information is required by incident handling staff for collaboration purposes. Interviewees repeatedly expressed that information about whom and when to contact is lacking, impeding incident handling:

Q14: "It took us more than one and a half day to understand that the point-of-sales machines were causing the issue. At that time we did not have direct access to staff at the payment processor. After we finally discovered who we needed to contact and we contacted that person the problem was solved in no-time", maintenance engineer

H2: Contact details of resources (-)
Contact information includes contact details, such as phone numbers, which is typically missing, as shown in the quote below:

Q15: "Most important is that I can contact you directly to solve the incident instead of getting to you via three points of contact. Next time I would like to call you directly, especially during a priority 1 incident", incident manager

H3: Changes in resources (-)
Also staff changes are considered important information as such information helps to adapt the human network, for instance by communicating the changed contact list, while such information is considered missing:

Q16: "Suddenly we have somebody on the phone that does not even understand which screen to select. Normally we get person x, y or z who knows exactly what we are talking about. However x, y or z are unexpectedly replaced by somebody that lacks knowledge and context, which is obviously less efficient", incident manager

2.5.4 Information storage, retrieval and transfer

Data analysis revealed a various mix of cognitive and technological information stores that are used for incident handling. The technological stores are typically accessible within the IT service provider. Information between providers is almost only shared by means of ad-hoc conference calls, mails and telephone calls during the incident handling process. Documented information that is shared between providers is typically hard to understand, as the information lacks context sharing which is required to properly interpret information:

Q17: "We all should be able to access the same information. However a mainframe document is unusable in a Windows environment… completely unreadable… a different world. Same is true for a Unix environment. So every provider must be able to interpret the information", maintenance engineer

The large amount and dynamic nature of information exceeds the cognitive capacity of the individuals, which brings a need for information sharing and collaboration during incident handling. To quickly access information staff relies on their mental directory of cognitive and technological sources:

Q18: "That implies that I have to know where I need to get my information, which service teams do I need to contact and which systems are maintained. That is why contact information is so important. You need to know where to get the information. That is the most important to me. I expect that my information sources have that information", incident manager

Such interpersonal information dependency has multiple information loss challenges. The information enquiry must be correctly understood by the requestor (staff member 1) and correctly articulated to the provider (staff member 2). The provider has to subsequently correctly understand the enquiry and collect the correct information. The information has then to be correctly communicated and correctly interpreted by the requestor.

Q19: "The essence is that we have access to the information sources ourselves. For example we request information from another IT service provider because we do not have access ourselves. However what we get back is 'yes it looks *good', while we need to exactly know what the other sees on his screen and how fast this is shown. Incidents often take longer because of the lack of information sharing",* maintenance engineer

Also the information selection process might be flawed. The following quote shows a business partner that receives a notification from an IT supplier that plans to change the format of data delivery. Since the notification is not correctly interpreted and therefore not passed to the IT service provider, time to process the necessary changes in the IT system is lost, ultimately leading to a major IT incident:

Q20: "Suddenly we did not receive the transactions and our business partner escalated to us that the system was malfunctioning. We started to analyze the incident but were unable not find the root cause. So we sent a mail to the supplier and registered an incident but did not receive any response. After we started calling it turned out that a letter was sent months before to our business partner. However since the business did not understand the importance of the letter we did not take action. When we finally received the letter it notified us that the message format would change and that the receiving IT system should change their message format", incident manager

Also organizational proxies between IT providers impedes effective communication between IT service providers, as additional communication links are introduced and proxies are often unaware of the (technical) context:.

Q21: "The infra supplier are a difficult world to reach. For instance we get information about a malfunctioning of a firewall in a very late stage. The infra provider never notifies such incidents. We also do not have direct access, so telephone calls need to be rerouted to reach them. A very tricky world" maintenance engineer

2.5.5 Conceptual model

The coding clustering process resulted in the identification of three interrelated networks. The first identified network is structured by the contractually agreed IT services, as shown in Figure 5. Visibility over the contractual network is needed to understand the big picture and the position of the IT service in the provider network. The second network is structured by the interrelated technological systems and components that exchange data. Visibility over the technical network is needed for the IT incident analysis process. The third network is the network of incident handling staff, consisting of incident managers and maintenance engineers. Visibility over the human network enables effective collaboration and information sharing.

Table 5 shows for each information category the number of found quotes. The number of first tier quotes for each category is shown in column 2 and for the second tier in column 4, the number of quotes by maintenance engineers is shown in column 6 and for incident managers in column 7. Column 3 shows the % of total first tier quotes for each first tier information category and column 5 the % of total second quotes for each second tier information category. The categories can be either static or dynamic: static information entails the current state of the contracts, technology and human network, dynamic information concerns changes in service levels, service specifications, technology and staff.

Each of the found information categories is clustered into the human, contractual and technical information concept, which is used to build the conceptual model, as shown in Figure 7. The figure shows for the incident manager and maintenance engineer role the dependency on the information categories; the number in each relationship indicates the dependency based on the number of quotes. Information categories tagged with '2' are used or needed by second tier IT providers, information categories tagged with '(-)' are perceived as missing.

Table 5, Number of quotes per information category

Row Labels	1e tier	1e tier %	2e tier	2e tier %	ME Quote	IM Quote
Human	37	14%	38	39%	38	37
Network of human resources	5	2%	26	27%	15	16
Contact details of resources	17	7%	10	10%	13	14
Change in resources	3	1%	2	2%	0	5
Human process	9	3%	0	0%	9	0
Human role	3	1%	0	0%	1	2
Contractual	96	37%	26	27%	59	63
Incident in IT service	54	21%	15	15%	27	42
Network of IT services	15	6%	8	8%	13	10
IT service level	24	9%	3	3%	16	11
Change in supplier service	3	1%	0	0%	3	0
Technical	126	49%	33	34%	98	61
Technical system process	34	13%	9	9%	33	10
Change in a critical IT system	41	16%	9	9%	19	31
Change in technical capacity	20	8%	7	7%	20	7
Network of IT systems	13	5%	6	6%	13	6
Design of the IT system	18	7%	2	2%	13	7
Grand Total	259	100%	97	100%	195	161

1e tier = first tier quotes; 2e tier = second tier quotes; ME = Maintenance Engineer; IM = Incident Manager

Figure 7 shows that maintenance engineers are most dependent on the technical network. Technical system processes are monitored on an intra-provider level, while there is a visibility need over the full network. We also found a need for inter-provider visibility over the network of the codependent IT systems, while such information is currently limited to an intra-provider level. Improved inter-provider visibility over the technical network will likely improve incident handling. In addition, as most of the incidents are caused by IT changes, technical change related information needs to be shared on an inter-provider level. Given the findings in this study we propose:

> *Proposition 1: Technical network information known to incident managers and maintenance engineers in the IT provider network positively impacts IT incident handling performance.*

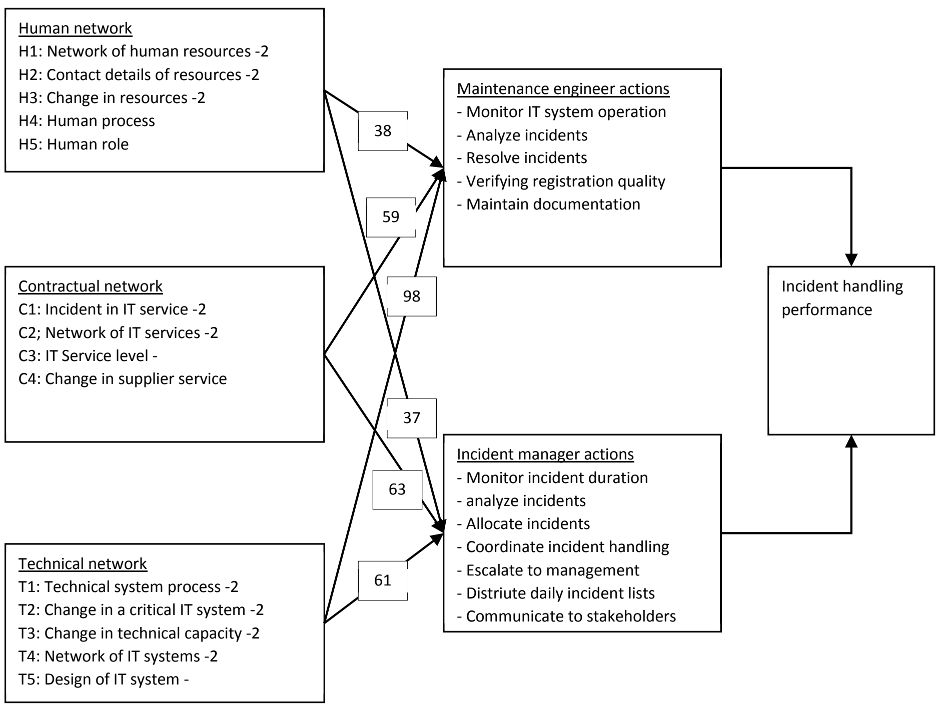

Figure 7, Conceptual model

Regarding the human network, both roles have a need for visibility over (1) the existing network of interacting incident handling staff, (2) contact details about whom to contact and (3) information about staff changes. Interviewees do not perceive a lack of visibility over the IT(IL) roles and IT(IL) processes. Incident handling staff rather has contact details about staff. Moreover we found that the human network acts as an information sharing mechanism, next to the existing technological stores. We therefore propose:

> *Proposition 2: Human network information known to incident managers and maintenance engineers in the IT provider network positively impacts IT incident handling performance*

IT incident information is considered the most important information in the contractual network, while this is often lacking (Q7) and bound to an intra-provider level (Q5 and Q6). Interviewees also express a need for overview over the network of IT providers to understand their position in the network (Q8). Based on these findings we propose:

> *Proposition 3: Contractual network information known to incident managers and maintenance engineers in the IT provider network positively impacts IT incident handling performance*

2.6 Discussion

This section uses a theoretical lens on the built conceptual model and the three propositions. Technical Information is cognitively and technologically stored and accessed by IT incident handling staff in the network. Accessibility to such stores requires a strong social network that empowers knowledge sharing. The knowledge management research area is related to these findings. Several knowledge management studies support the importance of information sharing in provider networks (Mesmer-Magnus & DeChurch, 2009; Rashed et al., 2010; Vlietland & van Vliet, 2013). Even though Rowley (2007) distinguishes information from knowledge by defining knowledge as information that is processed, transformed and/or enriched, we use both terms interchangeably in the sequel of this chapter.

Knowledge management studies the identification, creation, representation, distribution and retrieval of information (Nonaka & Takeuchi, 1995). Nonaka and Takeuchi (1995) identify two very different types of knowledge. At one end of the spectrum there is explicit knowledge being formal and systematic that is easily retrieved, transferred and stored in technological stores. We follow Frické (2009) and define such explicit knowledge as information as it can be easily stored. Tacit knowledge at the other end of the spectrum is highly personal knowledge, hard to formalize and consequently difficult to share. Business context of the IT provider for instance is hard to transfer between IT providers, while such context supports correct interpretation of shared information, as shown by the quotes Q1, Q4, Q7, Q8, Q13, Q17 and Q19. The lack of such knowledge can result in incorrect decision making (e.g. Q11, Q20).

We identified two major impediments for information sharing via technological stores. First, technologically stored information is typically bounded to an intra-provider level (e.g. Q6, Q14, Q15, Q21). Second, changes in the network decrease the reliability of technological stored information (e.g. Q3, Q10, Q12). Third, as discussed above, information might be incorrectly interpreted due to lack of contextual knowledge (e.g. Q7, Q17). Staff working in different IT service providers mitigates these impediments with verbal information sharing. The dependency on verbal information sharing implies that staff needs to know whom to contact and how to contact, explaining the importance of the human network (Huysman & De Wit, 2004).

One of the philosophical theories that fits our findings is Transactive Memory Systems (Nevo & Wand, 2005). Transactive Memory System (TMS) theory was first published by Wegner et al. (1991) who researched closed relationships in married couples. A TMS is a collective cognitive store in a group of staff that consists of (1) subject matter information and (2) an information directory of the information stored in the group (Wegner, 1987). The information directory consists of metadata such as people, systems, location, availability, accessibility, reliability and subject matter information (Hamid & Salim, 2011). Incident handling staff has a directory that points to both cognitive and technological stores.

Information directories require less storage than the actual information, thus are easier to encode, store and retrieve, while the member still has access to the information. The downside of information directories is the dependency on information sources (e.g. Q5, Q15, Q16, Q18, Q21); such as cognitive stored information in a staff member that is on leave. This can result in significant information gaps that obstruct incident handling (Q19). Second order information solely based on cognition has the highest chance of such obstruction as multiple linked cognitive stores are involved.

Bug fixing is another area that provides valuable insights. Like incident handling bug fixing (Guo, Zimmermann, Nagappan, & Murphy, 2011) depends on technically and cognitively stored information (Aranda & Venolia, 2009). Tools have been developed to support the human networks in sharing their knowledge. Distributed software development for instance uses tools to bridge the physical distance between software developers (Schuler & Zimmermann, 2008). The tools enhance visibility over the bug fixing process and code changes, which helps developers in their bug fixing endeavor (Begel & DeLine, 2009; Minto & Murphy, 2007). Such boundary spanning technology can also be used to support incident handling. One notable tool is Codebook that discovers transitive relationships between people, code, bugs and other related artifacts (Begel, Khoo, & Zimmermann, 2010). These tools in the bug fixing area might inspire new ways to improve incident handling.

2.7 Threats to validity and research limitations

This section provides the overview of the threats to validity and our mitigations (Dul & Hak, 2012; Golafshani, 2003; Mentzer & Flint, 1997).

The reliability of the research was enhanced by (1) recording the data collection activities in a log, (2) digitally recording and transcribing the interviews, (3) annotating the interviews in additional interview logs, (4) using a CAQDAS package to analyze the collected data, (5) triangulate the results with organizational documental records and (6) utilizing our prior published research about performance and visibility (Vlietland & van Vliet, 2013, 2014b).

The construct validity of the research was increased by validating the data collection results of stage A with subject matter experts, to mitigate the potential pitfalls of Meyers & Newman (2007). Triangulation of archival records and transcriptions was also used to increase construct validity (Saunders et al., 2009). An independent researcher coded one of the interviews which was compared with the researcher that coded all interviews to minimize coding process bias. The validity of the collected and generated content was increased by using an iterative research design that involved subject matter experts throughout the data collection and data analysis process. Collected content and analysis results of stage B were used in successive iterations to validate the performed data analysis.

Internal validity was improved by theoretically rooting the relationship between the information and incident handling concepts. Traceability of the causal relationship between the concepts was qualitatively performed by the hierarchical coding method and quantitative performed by determining the number of overlapping codes.

External validity was enhanced with interviews at multiple IT providers. We also validated whether role descriptions were similar throughout the network. Nevertheless the external validity in this research has limitations. First, we concentrated the interviews at the IT provider with the highest number of second tier dependencies, arguing that incident managers and maintenance engineers working in this central part of the network have the highest need of second tier information. Second, there are other incident related incident handling roles that can have other information needs, such as the support desks. Third, the interviews naturally exclude information that is not considered relevant or deliberately withheld by the interviewees.

2.8 Conclusion

This study aimed to find information that is shared and needs to be shared within networks of IT providers for effective incident handling. As the academic literature in this area is scant we studied an IT provider network consisting of nine interdependent IT providers in the payment industry. We conducted 15 planned interviews and 86 supplementary interviews in two subsequent stages. With coding techniques we analyzed and clustered the found information in categories. The strength of the relationship between the information categories and two main roles have been quantitatively analyzed. The information categories are clustered to information concepts to build the conceptual model, used to derive three propositions to theorize the impact of information on incident handling performance.

The results show that human contact information enables access to information and knowledge stores required for incident handling. The results show a perceived lack of most found information categories. Staff relies on information directories as alternative for individually stored subject matter information, resembling a transactive memory system. Staff relies on a cognitive information directory that point to cognitive and technological stored information, rather than have all knowledge cognitively stored.

The results of this study can be utilized for enhancing knowledge sharing capabilities in service provider networks. ITSM applications are likely candidates for developing such capabilities as multiple service providers in a network can use a common (SaaS-enabled) ITSM application. A typical functionally might be a graphic representation of the full provider network (Van Der Aalst, Reijers, & Song, 2005; van der Aalst et al., 2007), with (1) designated staff, (2) all involved IT systems and components, (3) a specification of the involved IT services and (4) the planned and processed technological changes. Developing such ITSM capability is a potential future research avenue, aided by development in the bug fixing industry.

Since we found that IT incidents are usually caused by deployed IT changes, a second research avenue might be the application of TMS to prevent such change related incidents. TMS can offer new ways to better analyze the impact and decide how to deploy such changes with minimal impact.

A third research opportunity is the study of the correlation between (1) cognitive and technological stores and (2) the ‘importance’, ‘urgency’ and ‘dynamic nature’ of the information. This offers knowledge and insight about the natural preference of the type of storage and the possible improvement to improve knowledge sharing.

As the study covered only one IT service network a fourth research avenue is replication of the study in other IT service networks.

2.9 Appendix

Stage A: Mapping the network

Selecting case

The table below shows the pre-defined questions for the first step in the data collection process that aims to understand the environment of the IT service providers.

Question	Argumentation
To what business is the most downstream IT service provider delivering?	To gain understanding about the business and the dependency from the IT service provider.
What IT services are being delivered by the most downstream IT provider?	Predefined services are required to be able to use the IT service perspective and defining the IT chain.
Have the IT services been contractually agreed?	Contractual agreements helps to identify the IT services and the relationship between two IT service providers (Poppo & Zenger, 2002)
What is the geographical location of the IT workers in the IT service provider?	To what extend the IT service providers in the IT chain are working with IT tooling to bridge geographical distance (Shachaf, 2008), e.g. email, SharePoint, HPSM.
How has the IT service provider been structured?	Insight in the organizational structure of the provider as this defines the distribution of roles and responsibilities.

IT provider network mapping

The table below shows the pre-defined questions for the second step in the data collection process that aims to map the full primary IT chain.

Question	Argumentation
Which upstream IT providers are delivering services to the IT service provider?	The list of providers helps to map the IT chain.
To which downstream IT providers is the IT provider delivering?	Verifying the validity of the structure of the IT chain by using the opposite perspective.
What are the financial flows in the IT chain?	The financial flows increases our understanding of the formal structure and accountability of the contracted IT service in the chain.
What are the operational flows in the IT chain?	The operational flows can differ from the contractual flows, such as an outsourced service desk function.
What is the geographical location of each IT worker in the IT service chain?	To what extend the IT service providers in the IT chain are hindered by geographical distance.
Is each IT service provider in the IT chain a separate legal entity?	Verifying to what extent TCE is present (Thouin, Hoffman, & Ford, 2009)
How is each of the IT service providers internally structured?	Insight in the organizational structure of the provider as this defines the distribution of roles and responsibilities.

Roles mapping

These questions help to define the roles to operationally deliver the IT service.

Question	Argumentation
To which IT services do you contribute?	IT services are the anchor of the interview and it verifies whether the interviewee is aware of the delivered IT service
What is your position in the IT chain?	This shows whether the interviewee is aware of the position of the IT service provider in the IT chain and whether the interviewee has understood the provided information.
What are your tasks and responsibilities regarding the IT service?	This makes clear whether the interviewee understands his role in the IT service.
Which other roles can be identified within the IT service provider?	Follow up question to define the roles.

Validation of IT provider network and roles

These questions ensure that the mapped IT provider network and the roles are valid.

Question	Argumentation
Do you recognize this IT service chain?	To confirm that the right suppliers have been involved in the supply chain.
Do you recognize the services?	To confirm that the services are recognized and fit the existing mental framework.
Do you recognize the IT roles / functions?	To confirm that the roles are recognized and can be used for finding the information that needs to be shared.
Which IT-roles with distinct tasks & responsibilities are involved in regular IT service delivery?	This helps to understand the perception of the interviewee about the roles involved in the chain and to verify the completeness of the IT chain.
Which IT-roles with distinct tasks & responsibilities are involved in restoring the <IT service>?	This helps to understand the perception of the interviewee about the roles involved in the chain and to verify the completeness of the IT chain.

Stage B: Answering questions

Introduction at the start of the interview:

- Purpose of the research
- Explain how we define information
- Service chain with the description of each node and service names
- Stakeholder map with roles and functions
- Clearly explain the distinction between daily operations (blue) and incident handling (red).

Part 1: What is the role of the interviewee?

Question	Argumentation
What is your role during daily IT operation?	To verify if our understanding is valid and it also refines our understanding of the position of the interviewee.
Can you give an example of a not working <IT service>?	Providing an example lowers the artificiality of the interview and therefore contributes to mitigate some of the potential pitfalls of Meyers & Newman (2007).
What is your role during handling such disruption from start to end?	To verify if our understanding is valid and it also refines our understanding of the position of the interviewee.

Part 2: Which information is important for high-performing IT service networks?

Question	Argumentation
Which information do you need to deliver the <IT service>?	Provides a view on what information is considered important to deliver the IT service.
What are the sources of that <information>?	This enables us to better understand the current visibility of the interviewee.
Which loss of information would result in more incidents?	Looking at the opposite perspective to see whether the answers correlate with the previous answer.
Which additional information would enable you to deliver a more reliable IT service?	Let the interviewee look at the incident from a fictional higher performance perspective to see which other information is important. This also gives us a first glance on the important >1 tier information.
Where should that <information> come from?	This helps us understanding the required visibility of the interviewee.
Which information do you need to restore the <IT service>?	Provides a view on what information is considered important to deliver the IT service.
What are the sources of that <information>?	This enables us to better understand the current visibility of the interviewee.
Which loss of information would create more of such incidents?	Looking at the opposite perspective to see whether the answers correlate with the previous answer.
Which additional information would help you to solve the incident faster?	Let the interviewee look at the incident from a fictional higher performance perspective to see which other information is important. This also gives us a first glance on the important >1 tier information.
Where should that <information> come from?	This helps us understanding the required visibility of the interviewee.

<u>Part 3: Which information of second tier nodes is important for high-performing IT provider networks?</u>

The following information needs to be shared with the interviewee at the beginning of this stage of the interview:

– Explain the concept of tiers in the IT provider network
– Explain the concept of a second tier node

Question	Argumentation
Can you give an example of information that you uses from second tier nodes during regular IT service delivery?	Providing an example lowers the artificiality of the interview and therefore contributes to mitigate some of the potential pitfalls of Meyers & Newman (2007).
Which information (from tiers that are not directly connected to your tier) do you use during regular IT service delivery?	Collects existing information >1 tier information that is being used by the interviewee
Which other information (from tiers that are not directly linked to your tier) will help to improve the performance during regular IT service delivery?	This inspires the interviewee to look from a fictional higher performance scenario.
Can you give an example of information that you uses from second tier nodes during incident handling?	Providing an example lowers the artificiality of the interview and therefore contributes to mitigate some of the potential pitfalls of Meyers & Newman (2007).
Which information (from tiers that are not directly connected to your tier) do you use during incident handling?	Collects existing information >1 tier information that is being used by the interviewee
Which other information (from tiers that are not directly linked to your tier) will help to improve incident handling of the IT service the most?	This inspires the interviewee to look from a fictional higher performance scenario.

Chapter 3

Improving IT incident handling performance with information visibility

In large internal IT organizations, multiple teams are often involved in handling incidents, so these teams come to depend on one another. We hypothesize that the knowledge these teams have of the agreed upon and realized incident handling performance of themselves and other teams will impact their performance. We tested this hypothesis at a large financial institute, using log data from the IT service management application and a survey to measure the knowledge of teams.

We found (1) a significant positive correlation between incident handling performance of a team and the knowledge a team has of its own performance, (2) no correlation between the knowledge of agreed upon performance and realized performance within a team, (3) that teams have very little knowledge of agreed upon or realized performance of other teams, and (4) that improving the knowledge a team has of the agreed upon and realized performance of that team and dependent teams, results in higher incident handling performance. The results show that increasing information visibility within and across teams in large IT providers is one way to improve incident handling performance.

3.1 Introduction

Internal information technology (IT) organizations of large multi-nationals often have more than 2000 employees. Such large scale IT allows to utilize economies of scale and centralizing activities to specialized IT service-teams. These IT service-teams are typically positioned in one of three service layers: (1) Infrastructure as a Service (IaaS), (2) Platform as a Service (PaaS), or (3) Software as a Service (SaaS). A service-team in a layer delivers technology based services to the next layer in the chain, as depicted in Figure 8, ultimately to the (internal) business partners.

What a service-team offers is usually predefined in a service catalog (Hiles, 2002). A service catalog typically defines the offered technology, the predefined maximum duration of IT incidents and the minimum average availability. What service-teams in a layer have to deliver to the next layer in the chain is predefined in Service Level Agreements (Sallé & Bartolini, 2004). A Service Level Agreement (SLA) formalizes the dependencies between the layers, including IT incident handling (Hiles, 2002).

Dependencies between service-teams in the chain are often critical. For instance disruption of an IaaS service causes disruption of the IT services on PaaS and SaaS level. A shared task of service-teams is therefore to handle IT incidents that occur during their service delivery (Bartolini et al., 2006; Team, 2010b). When an IT incident occurs the first service-team that discovers the IT incident assigns the task to resolve the IT incident to the service-team that seems to cause the incident. However, the assigned service-team may not be the one that has caused the incident, or is not able to completely resolve the incident. As a result IT incident handling tasks get routed between service-teams. Effective routing of incidents allows service-teams to solve IT incidents swiftly.

Yet, other variables affect effective incident handling, as service-teams need to be capable of handling assigned incidents. For instance a service-team may have a large backlog of incidents that prevents swift incident handling. In that case it is insufficient to just route the incident to a service-team. A subsequent phone call or the temporary reallocation of resources might be required. To be able to take such actions, visibility of incident handling performance by members of the service-team is required.

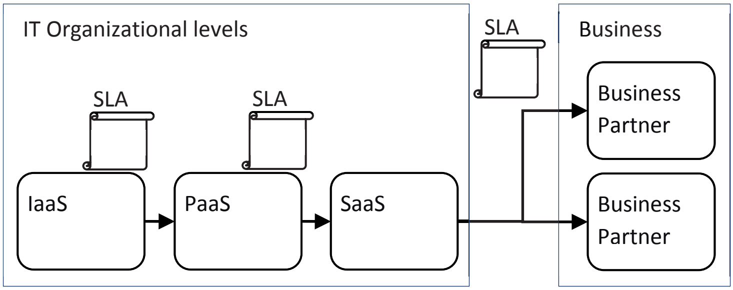

Figure 8, Service chain of an IT service provider offering services to business partners

Research in supply chains shows that improved supply chain visibility improves performance of the supply chain (Bartlett et al., 2007; Caridi et al., 2010b). Supply chain visibility for instance mitigates large fluctuation in inventory by optimizing the flow of goods (Disney & Towill, 2003; H. L. Lee et al., 1997). Supply chain visibility is defined by (Francis, 2008) as "the identity, location and status of entities transiting the supply chain, captured in timely messages about events, along with the planned and actual dates/times for these events".

As routing in supply chains resembles routing of IT incident handling tasks, we use the supply chain visibility concept in our research. We define visibility as: *"the quality of known information characterizing predefined entities in a predefined IT domain*". For this research this means: "*incident handling performance information known by IT service-team members that are part of IT service chains*".

We hypothesize that visibility of incident handling performance information positively correlates with incident handling performance. We argue that incident handling performance information is the primary type of information that has to be visible to IT staff, as this type of information indicates how well incidents should be handled and are handled by service-teams.

We investigate performance of incident handling in chains of service-teams. Based on our visibility concept we measure how much a service-team 'sees' of incident handling.

The visibility is measured in terms of (1) the 'seen' realized incident handling performance and (2) the 'seen' agreed incident handling performance defined in SLAs. The two visibility measures are correlated with the realized incident handling performance, to evaluate the hypothesis. Control theory is used to express the hypothesized relationship between visibility and performance.

Research was conducted in an IT organization of a multinational financial institute, at seven interdependent service-teams. We found that visibility of incident handling performance of a service-team significantly correlates with the incident handling performance of that team. We also found that knowledge about the performance of other, dependent, service-teams is extremely low. We did not find any correlation between the visibility of SLAs and incident handling performance. The results of this first case study are reported in (Vlietland & van Vliet, 2013).

In a follow-up case study in the same organization, we tested the usage of information visibility to *improve* incident handling performance. For one service-team, we used visibility based interventions to change the team's perception of the realized incident handling performance. The interventions are performed by the central incident management team by making the realized incident handling performance visible, for instance through incident handling reports. Over a period of 10 months in which we gathered empirical data, we found incident handling performance of this team improved from less than 10% to over 80%.

The remainder of this chapter is organized as follows. Section 3.2 discusses service-teams and incident handling. Section 3.3 describes the research design of both case studies. Section 3.4 covers the results of the first case study, in which we investigate the correlation between visibility of incident handling performance and actual incident handling performance. Section 3.5 elaborates on the results of the second case study, which focuses on improving incident handling performance by improving visibility. Section 3.6 elaborates on the threats to validity and the limitations of this research. Section 3.7 conclusion concludes the research, deduces the implications and suggests future research avenues.

3.2 Service-teams and Incident handling

As explained in the previous section, service-teams are positioned in layers (Bartolini et al., 2006). A team in a layer uses technology from other teams, typically from a lower layer in the chain. Figure 9 shows a chain segment.

In this figure, service-team Intel servers (IaaS) delivers computing capacity to a Windows hosting service-team (PaaS) that uses it to host a Windows webserver. This hosting service is delivered to a Business application service-team to host a web-enabled business application (SaaS). The arrows indicate the direction of service delivery, flowing from left to right. Each service-team in the chain enriches the service and offers it to the next layer in the chain.

A team does not solely deliver technology as it should also act on events, for instance when delivered technology gets disrupted (Bardhan et al., 2010; Jantti, 2011). In that case a team delivers an act or a deed next to technology(Jäntti, 2012a). We use the term IT services as a combination of technology and performed actions (Ellram et al., 2007; Peppard, 2003). As a service-team has an agreed responsibility with service-teams higher in the chain it has to take care of its supplying IT services, next to managing its own IT service(s) (Niessink, 2001).

The following example illustrates the involvement of multiple service-teams:

> *A SaaS service-team notices a disruption in its service. The team records the disruption as an incident in an IT Service Management (ITSM) application that is used to manage and route incident information. The team subsequently starts investigating the root-cause. The team discovers that the disruption is caused by a failed Windows hosting service and routes the recorded incident to the Windows hosting service-team. This is indicated by the dotted arrow and the icon in Figure 9. The hosting team receives the incident and starts investigating the root- cause. The team discovers that the hosting server is down and routes the recorded incident to the IaaS service-team. This team solves the incident and brings the services back online.}*

Many incidents are typically assigned to service-teams. As a consequence, a service-team has a backlog of assigned IT incidents as shown in Figure 9. The size of the backlog varies based on the amount of (related) incidents. Each service-team has to organize incident handling in such a way that all assigned incidents are resolved within the time constraints of the SLA.

Incidents can differ in priority and each registered incident is therefore tagged with an incident priority. High priority incidents must be resolved immediately; low priority incidents may be resolved later. Incident priority is typically based on user urgency and business impact. The maximum allowed duration of an incident for each priority is predefined in the service catalog.

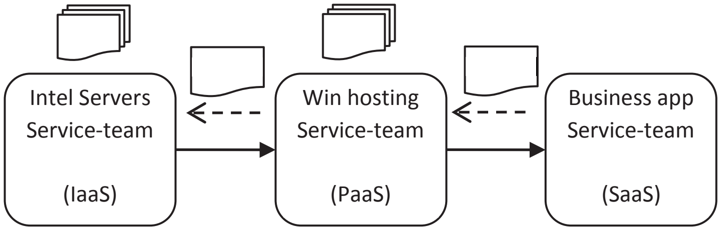

Figure 9, Routed IT incident between service-teams

The registration, assigning and tracking of incidents might be centralized to an Incident Management (IM) team. The central IM team tracks the handling of incidents on behalf of the technology oriented service-teams. Next to tracking, the central IM team provides incident handling performance reports (Jäntti, Lahtela, & Kaukola, 2011). Incident handling performance reports contain, for each incident priority, the percentage of incidents that are handled within the maximum duration by that service-team.

As all IT service-team members are potentially involved in IT incident handling we reason that the (1) agreed incident handling performance and the (2) realized incident handling performance should be known to all members of the service-team.
As mentioned earlier the dependencies between service-teams are typically critical; if a service is disrupted, dependent services get disrupted too. For instance in Figure 9 the business application gets disrupted when the windows hosting service is down. Given these critical dependencies we argue that knowledge about incident handling performance and agreed service levels should not be limited to the own service-team. The service-team members should also know the realized incident handling performance and agreed incident handling performance of interdependent service-teams.

3.3 Research design

Our literature research did not result in any literature about the correlation between incident handling visibility and incident handling performance. Nevertheless a large body of related literature was found that we discuss and use to build the model and shape the hypotheses.

We follow OGC for the definition of an IT incident: "an event which is not part of the standard operation of a service and which causes or may cause disruption to or a reduction in the quality of services and customer productivity" (OGC, 2007; van Bon et al., 2007).

We use control theory to theorize the relationship between performance goal, action and realized performance. Although control theory is historically used as a mathematical model to explain the behavior of physical systems, the basics can be also applied to human actors (Andrei, 2006; Wiener, 1965). Control theory consists of three fundamental concepts, as shown in Figure 10

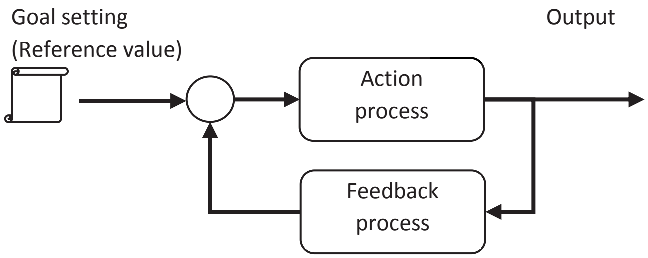

Figure 10, Control theory model

The first fundamental concept is goal setting, which in this case is predefined by the SLA for the service-team. The members of IT service-teams take action to achieve the incident handling goals. The second concept of control theory is feedback. Feedback enables the IT service-team members to know the realized performance. The third concept is the comparison function that compares the realized performance and the set goal. The difference between the two values is fed into the action process initiating (adapted) action to reach the goal.

The time between a change in output of the comparison function and the response of this change fed back into the comparison function is the time constant of the feedback loop. This time constant characterizes the response time of a system (Andrei, 2006).

The constellation of staff in a service-team that controls IT incident handling is abstracted with social network theory. Social networks are defined as nodes with links to other nodes (Freeman, 1979). In our case the nodes are human actors. A service-team is a group of linked nodes, forming a micro-level network, as shown in Figure 11.

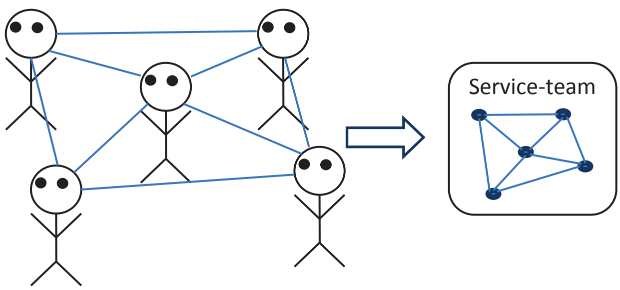

Figure 11, Linked human actors in a service-team

A node from a micro-level network can also be linked to a node of another micro-level network. These connections form meso-level links as shown in Figure 11. IT incident information is exchanged between nodes in different service-teams through the meso-level links. The information included in recorded incidents is exchanged via the ITSM-application. Additional information is be shared via telephone and email.

A notably meso-level link between nodes is the action of one node to influence incident handling performance of another service-team. For instance service-team B appeals to supplying service-team A to speed up handling of an incident so that the incident is resolved within the SLA constraints of service-team B.

The links (see Figure 12) at the micro-level and meso-level are used for hypothesis building further in this section.

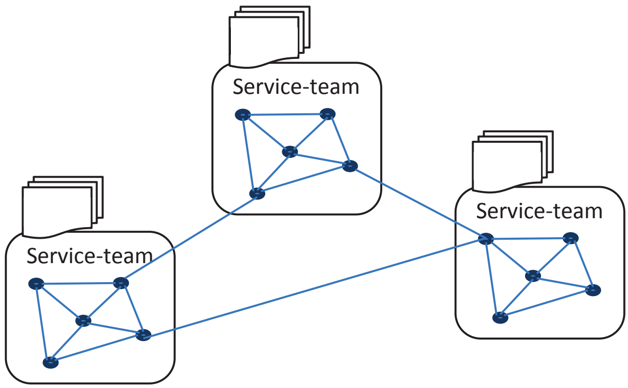

Figure 12, Linked service-teams

The visibility of agreed and realized performance information needs to be high by IT service-team members for effective control in and between service-teams. High visibility enables effective comparison of the agreed and realized value, triggering effective consequential action. High visibility information is indicated with the dashed line in Figure 13.

The supply chain visibility concept is used to measure the visibility of the known information in the control model. Caridi et al. (2010a) use the dimensions accuracy, quantity and freshness to measure visibility. For this research we simplify the visibility measure to the dimensions accuracy and freshness. Accuracy is defined as the knowledge of the node about agreed and realized incident handling performance information. For freshness we use the time constant parameter of control theory.

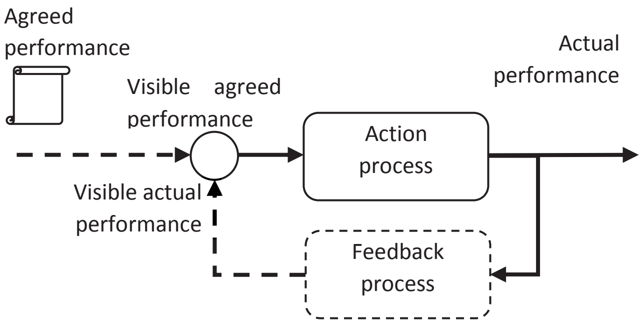

Figure 13, Control theory model augmented with visibility

Based on control theory and social network theory we argue that visibility leads to effective action in service-teams and the utilization of the social network links within and between service-teams to improve incident handling performance.

Visibility of incident handling performance of the own service-team is defined as: own visibility. Visibility of incident handling performance of other service-teams in the meso-level network is defined as: surrounding visibility. The dotted arrows in Figure 14 represent team members that have visibility over agreed and realized incident handling performance of the own and surrounding service-teams. Visibility of agreed incident handling performance helps members to understand their service-team goals and initiate action in the service-team to enable service delivery in accordance with these agreements:

> *[H1] We hypothesize that visibility of agreed incident handling performance values of the service-team positively correlates with incident handling performance of that service-team*

The same line of thought is applied to the visibility of realized performance. Visibility of realized performance metrics allows nodes improve their decision making and define necessary action to handle incidents:

> *[H2] We hypothesize that visibility of realized incident handling performance of the service-team positively correlates with incident handling performance of that service-team.*

Visibility of agreed performance of other service-teams in the meso-level network enables nodes to mitigate differences in agreed service levels. For instance service-team A knows that the agreed service levels of interdependent service-team B are lower than the agreed service-level of the own service-team. Knowing this enables service-team A to take mitigating actions to secure their own service-levels:

[H3] We hypothesize that visibility of agreed incident handling values between service-teams in the meso-level network positively correlates with incident handling performance of the service-team that has that visibility.

In the same line of thought visibility of realized performance values of other service-teams improves the control cycle by enhanced feedback, which improves decision making and action taking.

[H4] We hypothesize that visibility of realized incident handling performance of other service-teams in the meso-level network positively correlates with incident handling performance of the service-team that has that visibility.

If a correlation is found between visibility of realized performance and incident handling performance, it becomes relevant to investigate whether increasing visibility will positively impact performance. By increasing visibility the control cycle is (re)enabled as visibility brings new facts into the team comparison and action process. The control cycle is then utilized to improve incident handling performance. This leads to a fifth hypothesis:

[H5] Visibility based interventions positively impact incident handling performance.

Figure 14 shows the modeled relationship of visibility, performance and the nodes. The nodes have visibility over agreed and realized performance of the own service-team and dependent service-teams. The dotted arrows represent visibility. The arrows have a reference to the hypotheses and an indication of the visibility type (e.g. own surrounding). For instance a node of Service-team C has surrounding visibility over the agreed and realized performance of best-known Service-team A.

The financial institute that was subject to our research has a centralized IT organization of 4,000 fte. The organization is split into IT Service Delivery Centers (SDC). Each SDC delivers predefined IT services to a single internal business partner. An SDC consists of IT service-teams that each deliver technology enabled IT services to groups of end users. Incident handling monitoring and reporting is centralized to a supporting incident management team.

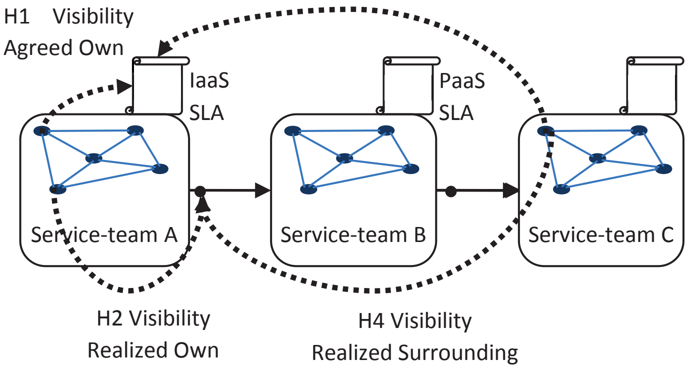

Figure 14, Hypothesized visibility and performance

The IT services of each software service-team are predefined in a service catalog. The service catalog contains a service description of each IT service covering the offered technology and the predefined maximum duration of IT incidents. The service catalog defines four incident priorities each with a maximum duration which is applicable for all service-teams in the SDC. Each incident priority is based on user urgency and business impact. The services to the business partner are formalized with SLA's, based on the service descriptions in the service catalog.

The team manager of a service-team is accountable for the agreed IT service delivery. The team manager takes care of overall team management. Accountability for incident handling performance is delegated to an incident coordinator which is part of the service-team.

An ITSM application is used to record and manage IT incidents between a service-team and the supplying service-teams. Each incident that is recorded in the ITSM application is tagged with an incident priority and a timestamp. Each incident that is solved is tagged with (1) the name of the team that solved the incident and (2) the closing timestamp of the incident.

The incident handling performance of a service-team is monitored by the centralized incident management team. This centralized team uses the ITSM application to generate reports that are monthly sent to the manager of the SDC.

We first tested hypotheses H1 --- H4, using data from a collection of interdependent service-teams. This visibility correlation case study is described in Section 3.4. Next, a second case study was performed to test hypothesis H5, by conducting visibility-based interventions in a single service-team, while measuring the effect thereof on the performance of the team. The latter case study is described in Section 3.5. Both case studies were done within the same network of service-teams.

The first case-study consists of five phases, as defined by Runeson and Höst (2009): (1) Case study design, (2) preparation for data collection, (3) collecting evidence, (4) analysis of collected data, and (5) reporting. The phases of the second case study also match those defined in Runeson and Höst (2009), albeit that the collection and analysis of data go hand in hand in the latter, and are grouped under "interventions and data collecting". The case study design of both studies is defined in this section, while the other phases are discussed in Sections 3.4 and 3.5, respectively.

3.4 Visibility Correlation Case Study

3.4.1 Preparation for data collection

In the first stage one software service (SaaS) is selected, based on archival record study and interviews with involved management staff. Selection criteria are: (1) the services of all interdependent service-teams are clearly defined, (2) the incident definitions (e.g. priorities) are the same for all service-teams and (3) one ITSM repository is used in all service-teams. Criteria (2) and (3) are chosen to ease the subsequent analysis. After the selection, the meso-level network is analyzed and modeled, based on the SLAs and technical dependencies between the service-teams.

The meso-level network selected consists of seven service-teams, shown in Figure 15. The service-teams are distributed over the SaaS, Paas and IaaS layers. The links in Figure 15 illustrate the chains of delivery. The numbers between brackets show the number of team members of each service-team.

Each service-team in the network delivers services, predefined in a service catalog. The service catalog contains service descriptions of all services in the network, including incident handling and four incident priority definitions. Each incident priority has its maximum duration: 2 hours for priority 1, 8 hours for priority 2, 3 business days for priority 3 and 10 business days for priority 4 incidents, regardless of the interdependencies between the IT service-teams.

The SLAs contain the predefined services that are agreed between the layers of service-teams (Hiles, 2002). For instance, the Infra Server service-team provides IaaS that is used by the Unix service-team to host a Unix webserver and the hosting platform is used by the SaaS service-team to host the web-enabled financial application. The SaaS service-team delivers the hosted application to the internal business partner.

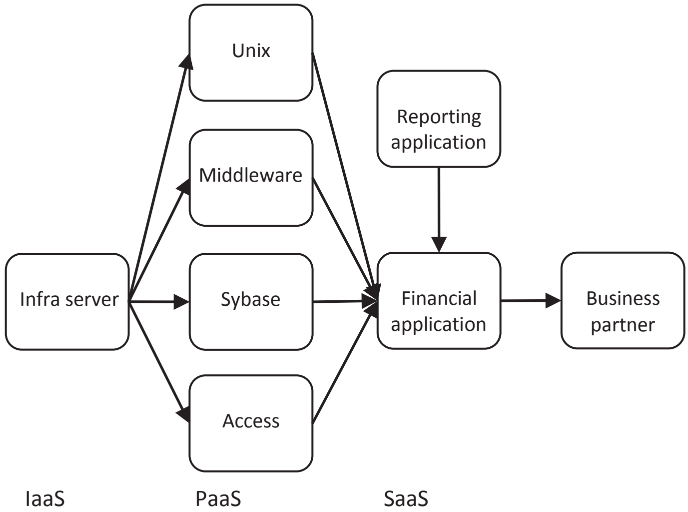

Figure 15, Network constellation of research domain

The network has 107 nodes (team members), distributed over the seven service-teams. Each of the service-teams has a team manager that has accountability for service delivery as agreed in the service catalog and SLAs.

Service level agreements are agreed and documented by a separate contract management team; the service-teams do not have an active role in this. The contract management team is positioned in a separate branch in the organizational structure and is located at another geographical location. This incurs the risk that agreed upon service levels are known to members of the contract management team and not to members of the service-teams.

Incident registration as result of user phone calls is performed by an outsourced helpdesk. The helpdesk handles phone calls, registers the incident in an ITSM-application and routes them to the applicable service-team. Service-team staff uses the ITSM-application to pick the high ranking incidents from the backlog.

Correct usage of the ITSM-application and recorded data is monitored by incident managers. These incident managers are part of a separate incident management team located in a separate hierarchical branch. The centralized incident management team distributes incident management reports to the team managers of the service-teams on a monthly basis. The fact that a helpdesk takes care of customer interaction and incidents are monitored by a centralized team may negatively impact performance visibility.

3.4.2 Collecting evidence

Quantitative data about the agreed service levels are extracted from the service catalog and SLAs. Data about the incident handling performance of the last six months are collected from the ITSM repository. The data are collected one month after the survey is completed to be able to measure the effect of visibility on performance. We performed a walkthrough of the data and cross month data check for consistency purposes. To prevent validity issues we used the raw dataset without a data cleanup.

We collect quantitative data about the existing visibility in the network with a survey as shown in the Appendix. The survey measures the existing knowledge of the agreed and realized performance of nodes. Data is collected for each of the incident priorities.

With a pretest to test the understandability of the questionnaire we found that visibility of surrounding performance was very low. As we need a reasonable level of visibility to enable correlation with incident handling performance we limited the measurement of surrounding visibility to that of the best known service-team.

The first part of the questionnaire collects the data about visibility of the own service-team. Question 1-4 collect data about the visibility of the agreed incident handling performance of the own service-team. Question 6-9 collect data about the visibility of the realized incident handling performance of the own service-team.

The second part collects the data about visibility of surrounding service-teams in the network. Questions 10-11 collect data about the agreed incident handling performance of the best-known, surrounding service-team. Question 12-16 collect data about realized incident handling performance of the best-known, surrounding service-team.

We aim to measure the factual knowledge of service-team nodes, so we use multiple-choice questions with one correct, four incorrect and a 'don't know' option (questions such as 'do you know?' extract perceptions, not factual knowledge).
The questionnaire includes one open question to collect the respondents' thoughts and comments about the incident handling process, with the aim to enhance our understanding of the answers to the multiple choice questions.

Table 6 summarizes the four visibility measurements, categorized in (1) agreed and realized service levels and (2) visibility of the own and best known surrounding service-team.

Table 6, Used variables for performance visibility measurement

Visibility	Agreed	Realized
Own service-team	Agreed service levels of the own service-team	Realized average incident handling duration of the own service-team.
Best known service-team	Agreed service levels between the own team and the best known service-team.	Realized average incident handling between the own team and the best known service-team.

Performance data was collected in October. The survey was conducted from the third week of September to the second week of October. The survey covered all 107 nodes in the network and resulted in 92 responses.

All performance and visibility variables used in this case study are listed in Table 7, together with the corresponding metrics. The same set of variables is used in the second case study, reported in section 3.5. All these variables have an implicit variable denoting the team. That is, they are computed for each team separately.

Table 7, Definition of variables and their metrics

Variable	Definition	Formula
t	time of survey	
m	Month (-4, -3, -2, -1, 0, 1)	
p	Priority (1, 2, 3, 4)	
n	Incident number n	
P(n)	P = 1 (incident n solved in SLA time); P = 0 (incident n not solved in SLA time)	
N	Total number of incidents for one priority	
P(p, m)	Average performance for priority p incidents in month m	$\mathrm{P(p,m)} = \sum_{n=1}^{N} \frac{P(p,m,n)}{N}$
r	Respondent number r	
R	Total number of respondents	
VA(r)	V=1 (answer of the respondent correspond with SLA performance variable); V=0 (answer of the respondent does not correspond SLA performance variable)	
VAO(p)	Average visibility of all respondents on the agreed performance of the own service-team for incident priority p	$\mathrm{VAO(p)} = \sum_{r=1}^{R} \frac{VA(p,r)}{N}$
VAS(p)	Average visibility of all respondents on the agreed performance of the best known service-team for incident priority p	$\mathrm{VAS(p)} = \sum_{r=1}^{R} \frac{VA(p,r)}{N}$
VR(r,m)	V=1 (answer of the respondent correspond with realized performance variable); V=0 (answer of the respondent does not correspond realized performance variable)	
VRO(p,m)	Average visibility of all respondents on the realized performance of the own service-team in month m for incident priority p	$\mathrm{VRO(p)} = \sum_{r=1}^{R} \frac{VR(p,m,r)}{N}$
VRS(p,m)	Average visibility of all respondents on the agreed performance of the best known service-team in month m for incident priority p	$\mathrm{VRS(p)} = \sum_{r=1}^{R} \frac{VR(p,m,r)}{N}$

3.4.3 Analysis of collected data

Table 8 shows the number of handled incidents in the months May (t-4) to October (t+1).

Table 8, Number of incidents per month

Service-team	May	Jun	Jul	Aug	Sep	Oct	Total
Infra Server	626	673	579	494	607	592	3571
Middleware	25	32	16	24	32	34	163
Financial application		112	171	24	65	46	418
Reporting app	119	15	58	72	58	162	484
Sybase	21	26	766	703	628	777	2921
Unix	905	1022	1005	747	743	835	5257
Access	450	36	878	784	923	908	3979
Total	2146	1916	3473	2848	3056	3354	16793

The duration of each incident is determined by comparing the registration timestamp and the resolving timestamp. This duration is then compared with the predefined maximum duration of that incident, to determine whether the incident was solved in time. The maximum duration is, for each incident priority, predefined in the service catalog.

Incident handling performance of a service-team is defined as the percentage of incidents solved in time. This percentage is determined for each priority, for each month separately. Variable $P_{(p,m)}$ represents the incident handling performance, where *p* is the priority and *m* is the month.

The performance data of October $P_{(p,t+1)}$ is used for the correlation analysis. Table 9 shows the incident handling performance in October for each incident priority for each service-team.

Table 9 shows for instance that Infra Server service-team (delivering infrastructure as a service), has a weighted average incident handling performance of 0.85 in October, which implies that 85% of the 592 recorded incidents have been handled within the agreed service-levels.

Table 9, IT incident handling performance $PR_{(p,t+1)}$

Service-team	1	2	3	4	Total
Infra Server	0.50	0.51	0.92	0.64	0.85
Middleware		0.40	0.71	1.00	0.68
Financial application		0.00	0.57	1.00	0.54
Reporting app				0.97	0.97
Sybase	0.67	0.46	0.87	0.74	0.85
Unix	0.50	0.60	0.90	0.76	0.87
Access	0.00	0.17	0.10	0.18	0.11
Total	0.36	0.50	0.67	0.71	0.66

The empty cells imply that not all service-teams recorded priority 1, 2 or 3 incidents that month. We validated the missing figures with respondents. The respondents explained that staff tends to spend time on resolving incidents, rather than logging them. This is particularly applicable to priority 1 incidents as they need to be solved in 2 hours.

Incident handling performance of low priority incidents (prio 4) is higher than high priority (prio 1) incidents. Probable causes are (1) the lack of knowledge about the agreed incident handling duration, (2) the inability to handle priority 1 incidents in 2 hours and (3) the lack of logging all priority 1 incidents. Additional research is required to determine the actual cause(s).

The table shows that 66% of the incidents in the network were handled within the SLA in October. Notable is the bad performance of the Access service-team, while performance of the other teams was rather high.

Visibility of agreed incident handling performance is determined for each incident priority, by comparing each answer to the questionnaire with the applicable value in the service catalog. The visibility of own agreed incident handling performance in the service-team is defined as the percentage of nodes that correctly answered the question. Variable $VAO_{(p)}$ represents the average Visibility of the Agreed incident handling performance of the Own service-team. The variable *p* represents the priority, as the visibility is determined for each priority separately.

Visibility of realized incident handling performance is determined for each incident priority, by comparing the answers of the questionnaire with the applicable realized incident handling performance value $P_{(p,m)}$, for each of the six months for which data is collected. Visibility of own realized incident handling performance is defined as the percentage of nodes that correctly answered the question for month *m*.

Variable $VRO_{(p,m)}$ represents the average Visibility of Realized incident handling performance of the Own service-team, for each of the six months (*m*). The variable *m* runs from four months back in time to one month ahead in time, based on the month of the survey *(m=t-4… t+1; t=0=month of survey)*.

Visibility of agreed incident handling performance of the best known service-team by a node is evaluated in the same way as for the own service-team. Naturally the agreed and realized levels of the best known service-team are used, instead of the agreed and realized levels of the own service-team. The variable $VAS_{(p)}$ represents the average agreed visibility of the best-known service-team. The variable $VRS_{(p,m)}$ represents the average realized visibility of the best-known service-team.

The correlation between visibility and incident handling performance is evaluated with Pearson correlation analysis as the first analysis showed a linear relationship between performance and visibility.

For performance we use the performance figure $P_{(p,t+1)}$ that is one month after closure of the survey *(m=t+1)*. This performance figure is used to assess the impact of visibility on incident handling performance.

The correlation is first performed for each of the six *(m = t-4 … t+1)* visibility datasets $VRO_{(p,m)}$ and $VRS_{(p,m)}$ to determine the dataset that has the highest correlation with performance $P_{(p,t+1)}$. The visibility dataset with the highest correlation is used to determine the time constant of the feedback loop, the freshness of the information.

The correlation analysis showed that a significant correlation existed between the performance dataset of October and the average of the visibility datasets of August and September. Visibility datasets of prior months did not result in a significant correlation.
The visibility results of Figure 16 are therefore based on the performance datasets of August and September.

3.4.4 Reporting

Figure 16 shows the average visibility of agreed own performance (VAO), realized own performance (VRO), average visibility of agreed best-known (surrounding) performance (VAS) and the realized best-known (surrounding) performance (VRS).

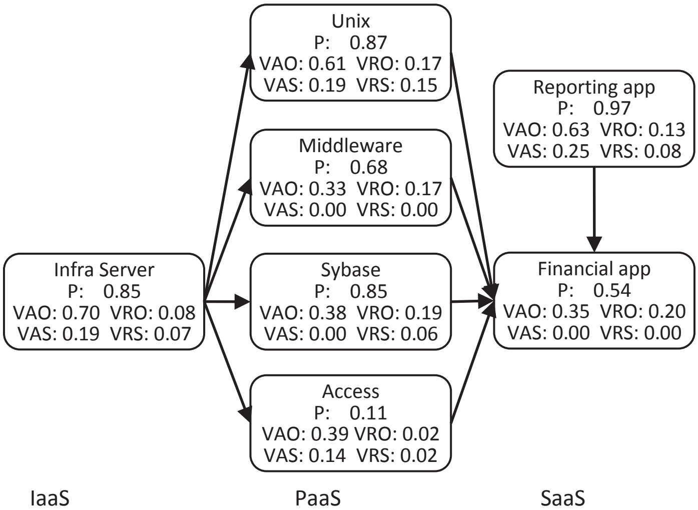

Figure 16, Performance and visibility per service-team

The results in Figure 16 show that the overall VRO is very low and the overall VAO is more than three times higher than VRO. Notable is the Access service-team which has hardly any visibility (0.02) on its own realized performance while the agreed performance is much higher (0.39). This notable difference triggered us to verify the cause: the manager of the team clarified that the team performance suffered from a severe lack of resources due to the implementation of a new policy. As the performance is very low as well (0.11, see Table 9), we suggest that team capacity should be added to the model as presented in the research design (Section 3.3).

The answers to the qualitative question provide additional interesting results. One respondent shares his need for *"a great team of incident-coordinators and a better overview of all incidents including meetings with all coordinators.... just like before"*. This confirms that incident coordinators have a lack of visibility over incidents.

The results in Figure 16 show that best-known (surrounding) visibility hardly exists. Three service-teams have no best-known surrounding visibility at all, not on agreed incident handling performance and not on realized incident handling performance of the best-known service-team.

The results are confirmed by respondents *"the own SLA is already hardly known, not to mention the SLA of another"*. Another respondent explains that *"in long complex chains my experience is that different members of the chain cannot find each other. In these cases incidents take weeks to solve which is very frustrating"*. A third shares *"I* *completely have no idea of SLAs from other departments"*. Another respondent proposes *"a bi-monthly broadcast in the ITSM-system on individual team performance so that all HPSC users can keep track of performance of the incidents allocated by each team; measures could be in terms of % meeting SLA"*. This confirms that incident handling performance is not visible between teams.

The result of the correlation analysis to determine the time constant implies an information freshness of 1.5 months, as the difference in duration between the average of August and September and October is 1.5 months. As the survey was held from the third week of September to the second week of October, the survey covered the reporting period of September. We reason that not all team members were already informed about the performance of September, being the cause for the high correlation between the performance visibility dataset of August and the performance of October. The results seem to confirm that the monthly incident handling performance report has a positive impact on incident handling performance.

Table 10 shows the Pearson correlation analysis between incident handling performance and the four visibility variables. The significance is given between brackets.
The correlation between own realized visibility VRO and incident handling performance is the highest (0.56) with a high significance ($p<0.01$). The correlation between P and VAO is not significant.
The correlation between P and VAS is not significant and the correlation is low (0.13). The correlation between P and VRS is significant at the $p<0.05$ level, which is caused by service-teams that handle incidents quite well and have zero visibility on surrounding nodes, as shown in Figure 17.

Table 10, Correlation matrix of performance and visibility

	P	VAO	VRO	VAS	VRS
P	1.00				
VAO	0.10 (0.33)	1.00			
VRO	0.56 (0.00)	-0.22 (0.13)	1.00		
VAS	0.13 (0.28)	0.45 (0.01)	-0.06 (0.40)	1.00	
VRS	0.41 (0.03)	0.12 (0.27)	0.43 (0.01)	0.43 (0.01)	1.00

High performance seems to be possible without any surrounding visibility. Nevertheless visibility seems to contribute to performance as the non-zero visibility values correlate with incident handling performance. Additional research is required to determine the underlying causes.

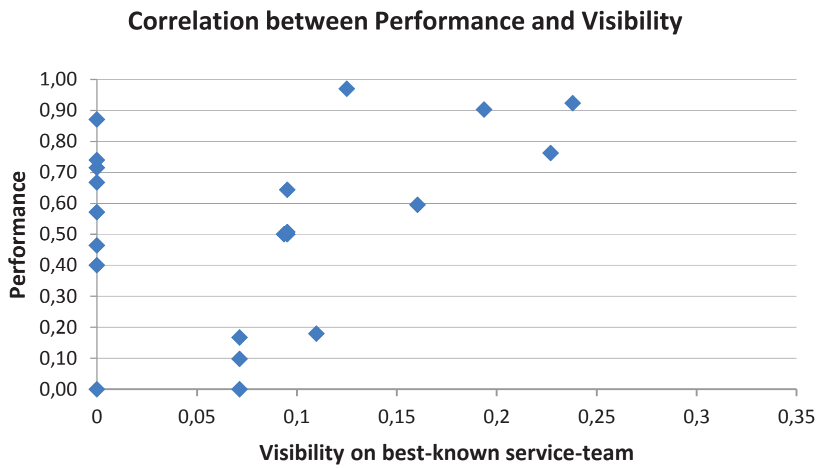

Figure 17, Performance and surrounding visibility

3.5 Visibility Intervention Case Study

3.5.1 Preparation for data collection

As service-team we selected the team that delivers the IT service to the business partner, as this team is has, direct and indirect, interdependencies with all other service-teams in the network (the team labeled "Financial application" in Figure 15). This team also complied with the criteria from Table 11.

Visibility of service-team performance is measured with the first part of the questionnaire as included in the Appendix, and archival record study. The survey measures visibility of agreed and realized service-levels. The answers of each of the respondents are compared with the average realized performance of the last two months. Each answer is scored correct (1) or incorrect (0). Performance visibility of the service-team is evaluated as the average of the scores of all respondents. The pre-condition of the team needs to be set to be able to test the hypothesis. The researcher is involved in setting the preconditions. Table 12 shows the pre-conditions and their rationale.

Table 11, Research service-team criteria

Team criterion	Rationale
Team member cross-location	Cross-location implies limited possibilities for face to face contact and reliance on email, and conference call contact, enabling us to track (the effect of) the interventions.
Service-chain	The service-team is part of a service-chain having interdependencies with other teams.
Accepted services	The agreed services include incident handling used for objective setting.
ITSM application	The team uses an ITSM application to enable measurement and reporting of team performance.
Communicate via email and conference	The team is familiar with email and conference call communication, which implies their usage for the interventions.
Low visibility	Low performance visibility allows us to conduct visibility based interventions and measure their impact on incident handling performance.
Low performance	Low performance allows enhancing service-team performance.

Table 12, Setting service-team pre-conditions

Team condition	Rationale
Objectives	Incident handling objectives are shared to set the reference level for the control cycle. Without set objectives the team members do not understand the objectives and cannot work towards these objectives
ITSM application configuration	The application is preconfigured, such as team names and incident priority definition to ensure that the ITSM application can be used effectively for tracking incident handling performance.
Reporting preparation	The central team of incident managers compiles clear incident handling performance reports. Clear incident handling reports are required to maximize possible effects of the interventions.
Incident coordinator	The incident coordinator role and name is communicated by the team manager. This is required to make team members aware of the role and responsibilities and understand the actions of the incident coordinator.

To measure incident handling performance the duration of each incident is compared with the maximum agreed duration, based on the service catalog and the SLA. The incident duration is defined as the difference between the incident registration and the incident closure timestamp. Incident handling performance of the service-team is determined by the percentage of incidents that have been closed within the maximum agreed duration.

For the measurement of the incident handling performance all incidents that are *controlled* by the software service-team are included. Controlled implies that the team is accountable for handling the incident in time, even when the incident is assigned to a supplying service-team. This is different from the notion of incident handling performance in the first case study, where we only count the incidents actually handled by the team.

Performance data is collected from the ITSM-application. The data collection includes the following information:

– Incident registration timestamp
– Incident closing timestamp
– Priority of the incident
– Service-team that controlled and monitored handling of the incident
– Service-team that solved the incident

Visibility based interventions are used to change the team perception about the realized incident handling performance. The interventions are conducted by the central incident management team by making the realized incident handling performance visible by means of incident handling reports. The report contains, for each incident priority, the total number of incidents and the number of incidents not handled within the SLA (breached); see Table 13 and Figure 18.

Table 13, Sample performance report

Priority	Within time	SLA Breached	Total	Performance (%)	Average duration
1	7	2	9	77	1.2h
2	30	15	45	67	6.7h
3	100	20	120	83	2.8 days
4	45	7	52	86	18 days

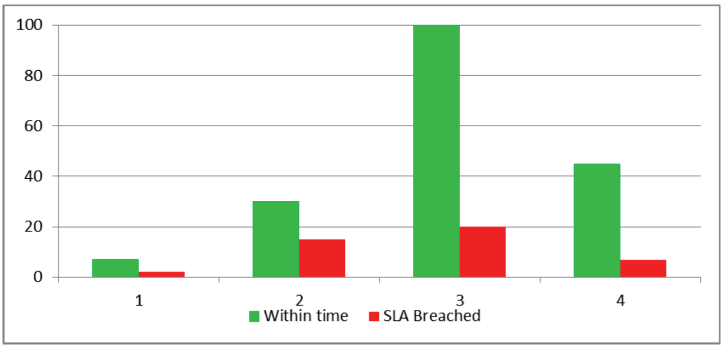

Figure 18, Performance per incident priority

The reporting frequency is increased from monthly to weekly and further increased to a maximum of daily depending on the performance improvement effects. The reporting intensification aims to increasingly challenge the existing incident handling performance perception. The reports are shared with the team manager, the incident coordinator and the other members of the service-team.

Weekly conference calls are held. Participants are the incident coordinator, the centralized incident monitoring representative, the researcher and optionally the team manager. The meetings have a prepared agenda. The topics and agreed actions are recorded in the minutes of the meeting which are distributed to the participants.

During the actual visibility based interventions the IT service-team and the centralized incident management team interact independent from the researcher, notwithstanding that the researcher receives carbon copies of emails. Furthermore the researcher is involved in the weekly progress meetings and if required compiles the minutes to ensure the improvement actions are well recorded. The researcher does not have a further role, such as managing actions to closure.

The empirical research for the second case study started right after the first case study ended, in November, and lasted till October the next year. At the start of the intervention period (November) the team had a performance of <10%, agreed performance visibility of 35% and realized performance visibility of 10%.

Archival email records showed that IT service-team members were informed about the agreed service levels in the service catalog (earlier in the year), as shown in the following snippet[1]:

> *"Main topic of the meeting was to discuss the current SLA and determine to what degree the SLA actually fits current practice and processes. It has been agreed to take more time to study the SLA, Service Catalogue and other relevant documentation in order to be able to define detailed questions which also will be discussed in the light of incident management in a separate meeting with << incident coordinator of the central team>>".*

This pre-condition has therefore been met although the overall knowledge was still quite low.

---

[1] The italic texts between quotes are the recorded texts in the email communication system. This email system was used for daily ad hoc communication by the involved staff next to the ITSM application. For confidentiality reasons, the actual names of persons are replaced by their roles, put between << and >>.

The second pre-condition, the ITSM application configuration, was also achieved earlier in the year. The application was able to support incident registration, prioritization, control and assignment. The remaining pre-condition were set between November and March of the next year. The third pre-condition was achieved by the centralized incident management team, with the realization of basic incident management reports as shown in the following email:

> *"The reporting instruments have been configured in <<ITSM application>>. Basic reports are now possible, with color codes Green = SLA met, Yellow = SLA in danger, Red = SLA breached.*
> *Everything should be now in place to perform your role as incident coordinator".*

The fourth pre-condition was achieved by assigning the incident coordinator in the service-team. Activation of the role was stimulated by communicating the incident coordinator name to all stakeholders. The incident coordinator was also invited in the weekly conference calls.

3.5.2 Interventions and data collecting

Visibility interventions were performed between March and September.
The incident management reports were provided by the central incident management team on a periodic basis as shown in the following quote:

> *"Please be informed of the periodic progress update: Link or see pics below. Remarks: Limited progress noticeable (1% p/day). Please provide your input how things can be improved and/or let us know what obstacles you encounter(ed)".*

The first reports showed that the number of registered incidents was very low. Reporting the results to clients of the IT service resulted in questions from clients about the number of reported incidents:

> *"We receive questions from <client> about the usage of the ITSM application by the service-team. There seem to be only a few incidents logged".*

This quote shows that staff in the vicinity of the service-team starts to ask critical questions about the low number of incidents after the first report was published. These questions subsequently induced improvement actions within the team.

The resulting analysis showed that a secondary incident backlog was in use, invisible to the environment. This triggered the decision to merge the secondary backlog in the central ITSM application, which resulted in a significant increase of registered incidents in the next months.

The report also triggered action with regard to the role description and reporting line of the incident coordinator. The role description turned out to be in conflict with incident coordinator tasks given by the hierarchical manager. The incident coordinator was discharged from these tasks:

> *“<< Resource manager >> has confirmed that the incident coordinator activities are managed by << functional manager >>. << Resource manager >> will not have an operational role. << Resource manager >> will only take care of the HR-performance cycle based on the input from the << functional manager >>.*

The discharge of these tasks provided a clear set of incident coordinator tasks and a clear reporting line to the functional service-team manager.
Closing the secondary backlog and the discharge of tasks are examples of discontinuous change that broke the inert state and enabled the team to improve incident handling performance.

At a later stage the reporting frequency was increased to weekly and at specific moments even to daily:

> *“For the progress meeting of today herewith the update to reflect the development since yesterday: the number of open tickets increased by 34, the number of close to breach tickets increased from 19 to 23, the number of breached tickets increased from 28 to 29”.*

The high frequency of the information, thus high visibility, helped the service-team to recognize the relationship between the incident handling actions of the team members and the incident handling performance of the team.

To enhance understanding of the performance values the full list of incidents was shared next to the incident reports:

> *“As discussed I hereby send you a complete list of all open incidents of the << service-team >> … judging by the contents I see lots of incidents which are no longer an issue/have been fixed. In comparison, << another service-team >> currently has 12 open incidents”.*

Detailed information in addition to the standard incident handling reports helped to determine the actions to improve incident handling performance.
The following quote from an email sent by the incident coordinator shows the role of visibility in the mental comparison function of the incident coordinator:

> *“Currently there are 261 open incident registrations for << service-team >>, which is >100% growth in 10 days … What's wrong?”.*

This resulted in an analysis to find the causes of the increased number of open incidents. The analysis showed that the queue was not effectively managed:

> *"…to get in control of the backlog the remaining action is to … close/transfer last << old >> 44 incidents in the ITSM application".*

Some of the incidents had a logging date older than a year. The incident coordinator therefore actively chased team members to close the incident in the ITSM application after resolving the IT incident:

> *"… please have a serious go at cleaning up the queue. We should be moving towards increased working from ITSM application and to keep our assignment group clean (i.e. housekeeping) is an important aspect of working well".*

To evaluate incident handling performance and to determine the appropriate action a weekly conference call was held. The meeting was prepared through the distribution of the latest incident management report and the meeting agenda. The meeting results and agreed actions were recorded in minutes and shared with the participants and related stakeholders. Additional stakeholders were added to the distribution list at a later stage to increase visibility of the actions. These stakeholders also received the incident handling reports.

The meeting supported the performance improvements in two ways. First the meeting stimulated the members to compare the incident handling reports with the performance targets and to define improvement actions. Second the distributed minutes engaged additional stakeholders to support the improvement actions.

At the start of the intervention period the incident coordinator executed his role in an reactive way. However after four months of high visibility the incident coordinator executed the role very proactively. The incident coordinator made an inventory of service-team staff that was involved in incident handling activities and sent an instruction to the involved staff. This instruction clarified how to record, prioritize, select, solve and close incidents. The incident coordinator also communicated actively to the involved staff and the stakeholders to handle recorded incidents. Furthermore the incident manager took care of the decommissioning of the secondary backlog and the migration of the incidents to the primary backlog.

The incident coordinator also started to proactively request incident reports to enhance understanding of the incident handling process:

"… could you << the central incident management team >> please provide a weekly (probably Monday) copy of the ITSM application reporting for the << service-team >> to me?".

The results show the importance of visibility in the control cycle. We noticed that motivation and ability also play a role in the control cycle. Motivation determines whether actual action is initiated to move closer to the goal and ability determines the effectiveness of these actions. Incident management reporting was effective only when the report content was accepted by involved staff and staff was able to execute the improvement actions. We concluded that the central incident management team was, organization wise, too far departed from the service-team to effectively manage incident handling. Centralizing incident management to a separate incident management team might remove essential management control causing low incident handling performance.

3.5.3 Reporting

With the above visibility interventions and the consequential social responses performance started to increase, which was noticed by the stakeholders of the service-team:

"Good to see the ITSM application has been configured …".
"Good to see things move to the right direction (In all honesty though, the credit is not mine)…".

The number of tickets dropped between July and August because of a bulk closure:

"Bulk closure of tickets was processed this morning. As a result the open tickets reduced from 300+ to 120".

The bulk closure involved old already resolved incidents that were not closed in the ITSM application. The bulk closure was performed by the central incident management team, in close cooperation with the incident coordinator.
From that moment the number of incidents started to decrease since the service-team started to discover causes of recurring incidents:

"…we please ask the help of << member1 >> and/or << member2 >> to assist us in reducing the number of calls".

This led to less time spent on phone calls and more time spent on resolving remaining incidents, which further increased incident handling performance.

In September the number of new incidents and closed incidents became in balance, implying that the amount of closed existing incidents equaled the recorded new incidents:

> *“Below the update regarding << service-team >> queue. Looks like ticket 'growth' and closure are in balance. Let us agree on goal to reduce backlog before end this week so you/colleagues 'only' face the task to resolve/close the tickets logged per day (between 10-30)”.*

The result of the incident performance development of the service-team is shown in Figure 19. The research started in November which is 0 on the x-axis.

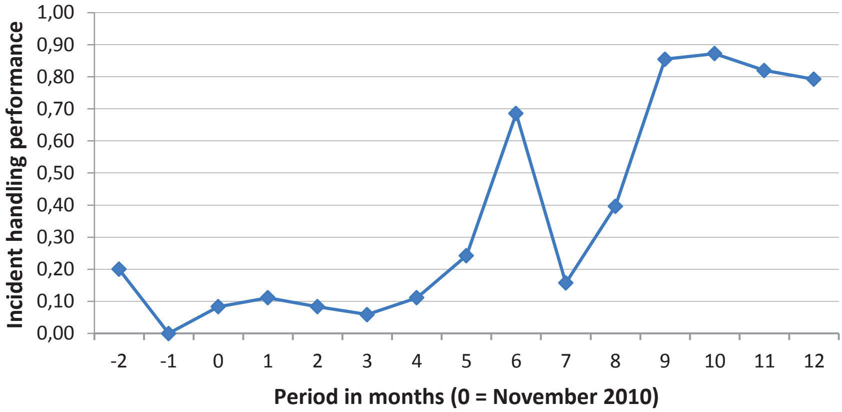

Figure 19, Incident handling performance trend

During the research period, performance increased from below 10% to almost 90% and seems to stabilize at approximately 80% (see Figure 19). The drop in performance between April and May was caused by decreased attention after the initial performance improvement. The performance drop was caused by a lack of closure actions, which was corrected by the bulk closure. After the initial drop the performance increased to the new stable state.

The central incident management team distributed the following text by email:

> *“With these results the team has become an example for other teams. We thank everybody that contributed to achieve this result”.*

Figure 20 shows the amount of registered incidents for each month. The figure shows that the low performance was initially caused by a lack of registered incidents in the

ITSM application. Once the visibility increased the number of registrations started to increase, enabling the control cycle.

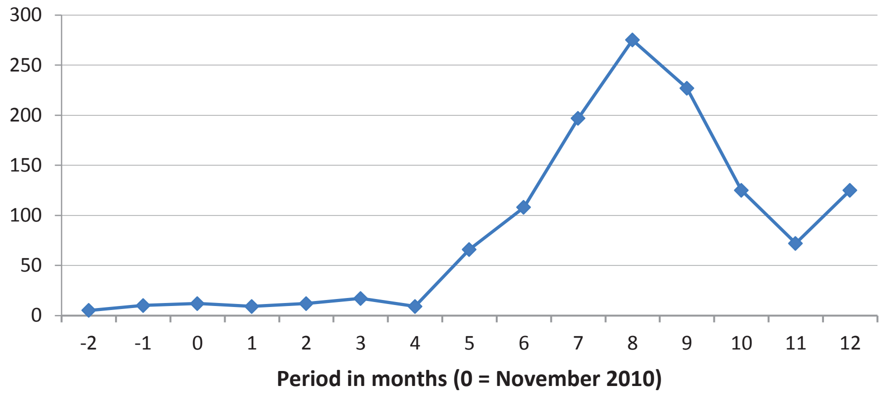

Figure 20, Number of recorded incidents in each month

3.6 Research validity

In this section we discuss the validity of the research. We discuss construct validity, internal validity, external validity and reliability.

3.6.1 Construct validity

During network mapping we validated that the services of the service-teams were defined in the service catalog. We also verified whether the same definitions of incident priorities were used within the full meso-level network. We included a question to verify the involvement of nodes in incident handling to increase population validity.

Understandability of the questionnaire was pre-tested on a service-team of another network, which led to changes in the formulation of questions. The pretest confirmed that visibility of service-team in the network was rather low which led to the decision to only measure visibility of the best-known surrounding service-team.

Archival records were used to triangulate data analysis. A second researcher participated in data analysis to minimize researcher bias. The data collected by the central incident management team to compile the incident handling reports was sample wise verified by the researcher. The visibility based interventions were conducted by the central incident management team to reduce researcher bias.

Construct validity of visibility is limited to the agreed and realized incident handling performance information. Other information, for instance resource capacity planning and service architectures, may also benefit incident handling performance. Another limitation of the construct is the dependency on logged performance information. We used an ITSM repository while some staff members mentioned that not every incident is logged.

3.6.2 Internal validity

The hypotheses with the related variables are built from supply chain management and control theory and subsequently empirically tested. The first study limits the test to correlation analysis. The actual causality of visibility is tested and confirmed in the intervention case study.

The measured visibility of each respondent is limited to the best-known surrounding service-team, which is a rather limited view of incident handling performance of surrounding service-teams. The limited measurement implies that the sum of surrounding visibility on other nodes might be larger than the visibility on the best-known service-team.

3.6.3 External validity

A limitation of the research is the coverage of only one service network in the financial industry. To increase external validity, the research needs to be repeated in other service networks.

3.6.4 Reliability

The researchers have taken several measures to enhance research rigor as explained in the research design. Nevertheless, a researcher bias might be introduced while performing visibility interventions.

3.7 Conclusion

We have empirically researched the relationship between visibility and incident handling performance. Based on existing related literature we built five hypotheses to verify the relationship:

> *[H1] We hypothesize that visibility of agreed incident handling performance values of the service-team positively correlates with incident handling performance of that service-team.*

[H2] We hypothesize that visibility of realized incident handling performance of the service-team positively correlates with incident handling performance of that service-team.

[H3] We hypothesize that visibility of agreed incident handling values between service-teams in the meso-level network positively correlates with incident handling performance of the service-team that has that visibility.

[H4] We hypothesize that visibility of realized incident handling performance of other service-teams in the meso-level network positively correlates with incident handling performance of the service-team that has that visibility.

[H5] Visibility based interventions positively impact incident handling performance.

To verify the first four hypotheses we performed empirical research at seven interdependent service-teams of an internal IT organization of a multinational financial institute.

The results show a significant relationship ($p<0.01$) between realized performance visibility of the own service-team and performance of that service-team in the following 1.5 month, so hypothesis 2 is supported. Visibility of realized incident handling performance seems indeed to influence service-team performance. As the own visibility is low (VRO = 14%) it suggests that enhancing VRO will improve incident handling performance.

The results do not show a significant correlation between agreed visibility and incident handling performance, so hypothesis 1 is not supported.

Surrounding visibility hardly exists; visibility of performance agreements of the best-known service-team is only 11% and realized performance even lower: 5%. This implies that only 5% of the given answers are correct. Given these low levels of visibility we were not able to confirm hypotheses 3 and 4.

Hypothesis 5 was empirically tested, and confirmed, in a subsequent case study within the same environment. This research shows that visibility based interventions can be a useful instrument for performance oriented improvements. In the end state the team actively manages the incident backlog to achieve high levels of incident handling performance. The results indicate that a more prominent role of visibility might benefit performance improvements initiatives.

3.7.1 Future work

Our study is a starting point and one of the future research opportunities is a follow-up study that includes a more detailed analysis, such as correlations per team and incident priority.

Another opportunity is to research the impact of ITSM-applications on visibility and performance.. Research in this area may gain understanding about the impact of ITSM-applications.

We also advocate supplementing the model with capacity. Staff may be unable to improve incident handling performance, due to a lack of staff and/or skills, as was found in the Access service-team. Another research avenue is to study which visible information has the most effect on incident handling performance. This is reported in (Vlietland & van Vliet, 2014c). One type of information might for instance be a dashboard that shows the overall network of interdependent service-teams with existing performance, utilization of resources and planned changes. A fourth research opportunity is to study the impact of information visibility on the performance of development teams.

3.8 Appendix

The appendix contains the questionnaire that was used to survey the visibility.

3.8.1 Part 1.1: Agreed performance of your own group/department

- What (do you think) is the agreed maximum resolution time of a priority 1 incident for your group? (1 hr, 2 hrs, 4 hrs, 8 hrs, 16 hrs, don't know)
- What (do you think) is the agreed maximum resolution time of a priority 2 incident for your group? (1 bus day, 2 bus days, 3 bus days, 4 bus days, 5 bus days, don't know)
- What (do you think) is the agreed maximum resolution time of a priority 3 incident for your group? (1 bus day, 2 bus days, 3 bus days, 4 bus days, 5 bus days, don't know)
- What (do you think) is the agreed maximum resolution time of a priority 4 incident for your group? (3 bus days, 7 bus days, 10 bus days, 15 bus days, 20 bus days, don't know)
- What has been the average number of incidents per month which you have worked on this year? (0-10, 10-25, 25-100, 100-250, 250+)

3.8.2 Part 1.2: Realized performance of your own group/department

- What (do you think) is the average realized resolution time of a priority 1 incident of yourgroup? (1-4 hrs, 4-8 hrs, 8-16 hrs, 16-32 hrs, more than 32 hrs, don't know)
- What (do you think) is the average realized resolution time of a priority 2 incident of your group? (1-2 bus days, 2-3 bus days, 3-4 bus days, 4-5 bus days, more than 5 bus days, don't know)
- What (do you think) is the average realized resolution time of a priority 3 incident of your group? (1-2 bus days, 2-3 bus days, 3-4 bus days, 4-5 bus days, more than 5 bus days, don't know)
- What (do you think) is the average realized resolution time of a priority 4 incident of your group? (3-5 bus days, 5-10 bus days, 10-15 bus days, 15-20 bus days, more than 20 bus days, don't know)

3.8.3 Part 2.1: Agreed performance of your best known group/department

- Select the department/group for which you know the SLA best (other than your own).
- (Infra Server, Unix, Middleware, Sybase, Access, Reporting App, Financial App)
- Do the agreements (SLAs and OLAs) with that department differ from the agreements with your client? (only 0-20% differ, yes 20-40% differ, yes 40-60% differ, yes 60-80% differ, yes 80-100% differ, don't know)

3.8.4 Part 2.2: Realized performance of your best-known group/department

- What is the average realized resolution time of a priority 1 incident of that group? (1-4 hrs, 4-8 hrs, 8-16 hrs, 16-32 hrs, more than 32 hrs, don't know)
- What is the average realized resolution time of a priority 2 incident of that group? (1-2 bus days, 2-3 bus days, 3-4 bus days, 4-5 bus days, more than 5 bus days, don't know)
- What is the average realized resolution time of a priority 3 incident of that group? (1-2 bus days, 2-3 bus days, 3-4 bus days, 4-5 bus days, more than 5 bus days, don't know)
- What is the average realized resolution time of a priority 4 incident of that group? (3-5 bus days, 5-10 bus days, 10-15 bus days, 15-20 bus days, more than 20 bus days, don't know)
- What do you think is needed to increase the performance of incident handling and IT changes? < open question; free text format >

Chapter 4

Towards a governance framework for chains of Scrum teams

Context: Large companies operating in the information intensive industries increasingly adopt Agile/Scrum to swiftly change IT functionality because of rapid changing business demands. IT functionality in large enterprises however is typically delivered by a portfolio of interdependent software applications involving a chain of Scrum teams. Usually, each application from the portfolio is allocated to a single Scrum team, which necessitates collaboration between the Scrum teams to jointly deliver functionality.

Objective: Identify the collaboration related issues in chains of Scrum teams.

Method: We used a qualitative approach with transcripted interviews from three case studies that were coded and analyzed to identify the issues.

Results: We identified six issues in chains of codependent Scrum teams; coordination, prioritization, alignment, automation, predictability and visibility. The synthesis of these issues with existing theory resulted in nine propositions. These nine propositions have been combined into a conceptual model.

Conclusion: We propose this conceptual model as a starting point for a governance framework to manage chains of Scrum teams that addresses the identified issues.

4.1 Introduction

Large companies operating in the information intensive industries experience rapid changing business demands that require swift delivery of new IT functionality. To be able to deliver such IT functionality swiftly internal IT development centers increasingly adopt Agile methods. A common Agile method is Scrum which aims to empower IT development centers to deliver customer focused IT functionality in a fast pace.

IT functionality in large companies however is delivered by a portfolio of interdependent applications, not just a single application. Each application in the portfolio supports a business function in the front to back business process. Typical front to back business functions are: front-office, mid-office, back-office and finance. Figure 21 illustrates a typical front to back business process with business functions for the mortgage business line.

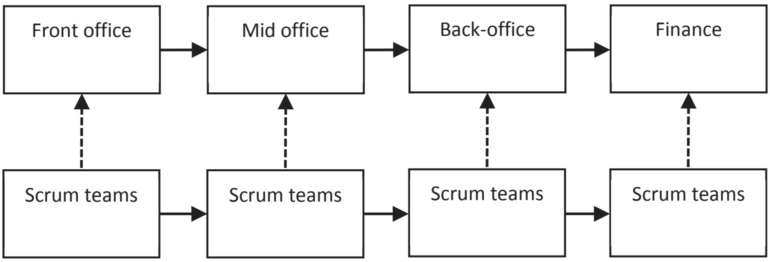

Figure 21, Front to back business process supported by Scrum teams

The front-office – in this example – performs customer facing processes, such as a mortgage client contact center. The mid-office calculates risk by a credibility check of a new client. The back-office performs the mortgage and settlement process, such as account opening. The finance function takes care of the actual funds provisioning, typically performed for multiple business lines.

Scrum is an Agile based method for incremental software development that uses low boundary cross-functional collaboration in software development teams that work toward a set team goal (Schwaber, 2004, 2011). 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. Scrum defines 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. A Scrum team has a small size (max 10).

The small team size eases intra-team knowledge sharing and utilizing the self-organizing ability in professional teams (Takeuchi & Nonaka, 1986). These self-organizing practices are encouraged by a structure, containing a product backlog, sprint backlog, sprint planning, daily-standups and a sprint review (Moe, Dingsoyr, & Dyba, 2008). The team has the task to develop software based on the sprint backlog (Rising & Janoff, 2000).

Scrum teams can be mapped in different ways onto the application landscape. Some prefer to have one Scrum team for the whole front to back chain. However two constraints make such front to back coverage difficult. First, the amount of involved IT staff then easily exceeds the generally agreed upon maximum Scrum team size of 10 members. Second, changes require highly specialized skills that cannot be shared easily. These two constraints result in dedicated Scrum teams for each business function in the front to back process, as shown in Figure 21. Each of these dedicated Scrum team delivers application functions that merged together results in features that automates the front to back business process. We define features as: 'intentional distinguishing characteristics of the application landscape that can be used by a business user', such as the mortgage registration feature.

Given the interdependencies in the application chain, multiple Scrum teams then need to jointly deliver new or changed features, as shown in Figure 21. Joint delivery implies that Scrum teams need to collaborate. Particularly the high frequency of deliveries which are common in Scrum settings likely makes efficient collaboration an important performance factor (Dorairaj, Noble, & Malik, 2012). Yet, due to the nature of Scrum teams, such collaboration might not happen naturally. A Scrum team has specific characteristics, such as a maximum of 10 members, multidisciplinary team, typically owned IT applications, high-frequency deliveries and focus on a single backlog. The focus on the single backlog in combination with the 'owned' IT applications likely results in a bounded Scrum team focus rather than a feature delivery focus. Such focus likely results in collaboration (related) issues.

The application of Agile methods in large organizations has been subject of research for more than ten years (Cockburn, 2006; Dingsøyr, Itkonen, & Fægri, 2013). The majority of reported results are experience reports and there is a need for developing theory that answers burning practitioner questions (Dingsøyr & Moe, 2013; Freudenberg & Sharp, 2010; Jalali & Wohlin, 2012; Martini, Pareto, & Bosch, 2013b).
In this study we empirically identify the issues in chains of Scrum teams and develop new theory. We used a qualitative approach with transcripted interviews from three case studies, that were coded and analyzed to identify the issues. We identified six issues: (1) a lack of coordination in the chain (2) mismatches in backlog priority between teams, (3) alignment issues between teams, (4) a lack of IT chain process

automation, (5) unpredictability of delivery to commitment and (6) a lack of information visibility in the chain. We subsequently build a first version of a conceptual model with these six issues. This conceptual model can serve as a starting point for the development of a governance framework to mitigate the identified issues in chains of Scrum teams. The chapter is an extended version of Vlietland and van Vliet (2014a), in which we presented a preliminary analysis of two case studies, and identified four issues. The present chapter contains a more elaborate discussion of these two cases as well as an additional case, resulting in a larger and refined set of issues. Also, the synthesis of these issues into a conceptual model is new material. Validation of the conceptual model is a topic for future research.

The remainder of this chapter is organized as follows. Section 4.2 covers related work. Section 4.3 explains the research method. Section 4.4 elaborates on the case studies and the issues identified. Section 4.5 synthesizes these findings into a conceptual model. Section 4.6 elaborates on the threats to validity. Section 4.7 concludes the study, deduces implications and suggests future research avenues.

4.2 Related Work

As we did not find literature addressing chains of Scrum teams, we expanded our literature study to Agile related issues in the enterprise. Our literature study acknowledged three categories of related work: scaling issues, 3C issues and automation issues. We identified multiple issues in each category.

Table 14 shows the overview of the identified issues in each category of related work. In general, different publications identify different issues, as explained in more detail in subsection 4.2.1 to 4.2.3. Since large body of literature offers solutions for Agile related issues, we also discuss the solution based literature in this section.

Table 14, Overview of identified issues in each related work category

Category	Identified issues / solutions
Scaling issues	Priority mismatches (3x), Test effort and coverage, Increased management overhead, Increased configuration management effort, Communication issues, Lack of information, Requirements gathering problems, Limited business feedback, Dependencies between Definitions of Done.
3C issues	Soft-standardizing managing work (2x), Operational visibility and transparency (2x), Unclear requirements and design, Reduced informal contact, Inconsistent work practices, Coordination dependencies, Scrum of Scrums
Automation issues	Sensitivity of integration mistakes blocking integration automation, Traditional tooling making development unnecessary complex, Heterogeneous configurations, Lack of transparency and a short feedback cycle, Challenging cultural shift to new values, Struggle with traditional processes hindering 3C and continuous improvements, Team specific test environments being incompatible with the integrated test environment.

A part of the related work concerns distributed contexts. Ågerfalk et al. (2005) identified three main issues arising with the distribution of software development practices: (1) spatial separation, (2) time-zone differences and (3) cultural differences. Spatial separation leads to exacerbation of communication and coordination (Ågerfalk et al., 2005). As communication and coordination is related to collaboration we include related work about distributed contexts in our literature study. We exclude the cultural and time-zone issues (Woodward, Surdek, & Ganis, 2010), since our study environment has a shared culture and time-zone.

Given the Scrum specific characteristics we narrow the literature study to Scrum setups. Sutherland et al. (2007)consider three models for collaborating Scrum teams in a distributed context: (1) Isolated Scrums – teams are geographically isolated, (2) Distributed Scrum of Scrums – teams are geographically isolated and integrated by Scrum of Scrums and (3) Totally integrated Scrums – Scrum teams are cross-functional with members distributed across geographies. Our literature study excludes the latter, given the characteristics of our study.

The remainder of this section is organized as follows: subsection 4.2.1 discusses related work about Agile scaling issues, subsection 4.2.2 discusses specific collaboration, coordination and communication related issues and subsection 4.2.3 discusses issues with regard to the automation of the software development processes.

4.2.1 Scaling issues when applying Scrum in large enterprises

The large enterprise with collections of Scrum teams is far more complex than a single Scrum team and faces many scaling issues (Ambler, 2012). Some researchers studied these Agile scaling issues when adopting Scrum in the enterprise. These scaled Agile studies identified a diverse set of issues, except for priority (related) issues which are mentioned by several studies.

Petersen and Wohlin (2009) studied a case of Agile projects with Scrum characteristics, and identified issues with (1) creating and maintaining the priority list, (2) effort to setup and maintain a test basis that covers a sufficient part of the applications, (3) increased management overhead due to a high number of teams that requires coordination and communication and (4) increased configuration management effort due to an increased number of releases. The paper does not indicate whether the issues are identified in a Scrum chain setting.

The issue with prioritization has also been identified by Lehto and Rautiainen (2009), who studied Scrum issues in a mid-size software company, while the setting of the

Scrum teams is ambiguous. In their study they conclude that prioritization of the high-level goals was unclear which impedes organizing and tracking development work. They also identified traceability issues from high level goals to detailed plans and a lack of information about progress which made it impossible to take corrective actions in time. Waardenburg and van Vliet (2012) studied two enterprises with Scrum teams. While the paper does not elucidate whether the teams are part of a chain, they also identified issues with prioritization, as a consequence of the lack of business involvement. Other identified issues are (1) requirements gathering problems, (2) limited business feedback, (3) communication problems and (4) dependencies between Definitions of Done.

Other related work report experience with scaling. Saddington (2012) reports a success case about scaled product ownership in multiple codependent Scrum teams. They state that visibility and alignment of vision, goals, teams and workload were essential elements to the success of the project. Rautiainen et al. (2011) describe a company that implements an Agile portfolio management structure to solve prioritization issues. Listing the projects in priority order on a single backlog creates visibility about ongoing projects, which benefits the coordination between Scrum teams.

Even though Agile principles aim to introduce flexibility the need for plans and structure remains (Talby & Dubinsky, 2009). Batra, Xia, VanderMeer, and Dutta (2010) conducted a case study on a large project and conclude that combining Agility with traditional plan-driven methods is essential for having both flexibility and control. Soundararajan and Arthur (2009) argue that Agile practices need to be structured to develop large software systems. Their proposed soft structured framework consists of a requirements gathering approach and a tailored development process. A comprehensive structure for a scaled Agile application in the enterprise is the Scaled Agile Framework (SAFe) of. The framework targets seven areas to achieve parallel Scrum development: (1) cross-functional teams, (2) standardized planning and tracking, (3) standardized iterations, (4) smaller, frequent releases, (5) concurrent testing, (6) continuous integration and (7) regular reflection and adaptation.

4.2.2 3C issues when applying Scrum in large enterprises

Codependent chains of Scrum teams (see Figure 21) develop features in parallel (Sutherland, 2005), which requires collaboration, coordination and communication (3C) between the Scrum teams (Sharp & Robinson, 2010). In this subsection we discuss Agile related work that identifies 3C issues.

We follow existing literature to define 3C. Collaboration is: "*the process of two or more people working together on a task*" (Henneman, Lee, & Cohen, 1995; Sharp &

Robinson, 2010). Communication is: "*the exchange of information or knowledge through verbal or non-verbal means between two or more people*" and coordination is: "the *process of managing dependencies among activities*" (Calvert, 1995; DeSanctis & Jackson, 1994; Sahin & Robinson, 2005).

Oppenheim et al. (2011) studied two Agile cases of collaboration between enterprises and present an architecture for such collaboration. They identified three challenges: (1) a standard way to manage similar work, yet allowing local variations, (2) overall operational visibility and transparency and (3) clarity about requirements and design. The need for visibility is also identified by Hildenbrand et al. (2008) which systematically analyzed the distributed software development scenario. They conclude that the distribution of teams hinders collaboration due to the virtual environment that limits body gestures and facial expressions. These limitations might also exist in chains of Scrum teams as the teams are more or less isolated and rely on virtual environments. (S. Lee & Yong, 2010) studied a combination of a global product team and three local IT teams and highlighted successful practices and challenges. They identified three main issues (1) a lack of planning and communication between teams, (2) a combination of low priority and inexperienced staff and (3) mixed responsibilities. Sharp and Robinson (2008) studied the operational collaboration of three Agile teams in different companies. They conclude that even though teams work in the same time-zone they face the problem of maintaining an informal but disciplined collaboration and coordination structure. Such need for structure is supported by the scaled Agile literature (see subsection 4.2.1). Collaboration between Scrum teams seems to be limited by design, because a Scrum team's goal is to realize its own backlog items. Schwaber (2011) supports this view by mentioning the unlikeliness of Scrum teams to collaborate and discusses a case of using product integration teams for spanning team boundaries.

We earlier referred to Sutherland et al. (2007), who consider three models for collaborating Scrum teams in a distributed context. Reported research on such collaborations (Hildenbrand et al., 2008; Oppenheim et al., 2011; Sharp & Robinson, 2010) do not unambiguously identify the model they consider. Our experience from practice is that, generally, aspects of the 'Distributed Scrum of Scrums' and 'Totally integrated Scrums' models apply.

Codependencies between Scrum teams seems to need coordination (Larman & Vodde, 2013). Coordination between Scrum teams is typically done by Scrum of Scrum meetings in which Scrum Masters from all teams participate. However such a Scrum of Scrums can be problematic. Paasivaara et al. (2012) conducted a multiple case study on how a Scrum of Scrum is applied, and concluded that they seem to work poorly in case of too many participants (15-20) and disjoint interests and concerns. A way to enable coordination between Scrum teams is to implement more elaborate structures, such as mentioned by Schnitter and Mackert (2011). They define product teams for coordinating work over multiple Scrum teams. These product teams consist of a product manager, product team Scrum Master, software architect, delivery manager, knowledge management & documentation expert, UI designer and a stakeholder representative. For managing more than seven Scrum teams an intermediate organizational layer is suggested between the product teams and Scrum teams to cater for the necessary coordination (Schnitter & Mackert, 2011). We found no other studies about coordination in chains of Scrum teams (Sutherland et al., 2008).

Yet, other related work is found in the area of global software development that report issues or experiences between groups. Bannerman, Hossain, and Jeffery (2012) identified four types of coordination challenges from global software development literature: (1) reduced informal contact leading to a lack of task awareness, (2) inconsistent work practices that impinge coordination. The two other items are related to time-zone and socio-cultural distance and are therefore not in our scope. Begel et al. (2009) surveyed 775 software engineers at a multinational software company and identified artifacts for coordination between development teams. They identified the artifacts release schedule, features, API's, bugs, documentation, code, prioritized work items and status, as the top coordination dependencies. Schedules and features are considered the common objects to coordinate work. Strode, Huff, Hope, and Link (2012) provide a model to coordinate Agile distributed projects. The model is based on three principles: (1) synchronization of activities, (2) an inter-team structure and (3) boundary spanning activities and artifacts.

Related work also exists in the area of communication, though the work does not specifically research communication issues in Scrum chains. Martini, Pareto, and Bosch (2013a) investigate, through a survey, communication factors affecting both speed and reuse in 3 large companies, identifying five Agile team interfaces: (1) system engineers, (2) product management, (3) distributed teams, (4) inter-project teams and (5) sales unit.

Green, Mazzuchi, and Sarkani (2010) studied the need for communication in the development phases of a distributed project. They conclude that in a distributed Agile setting communication is particularly indispensable during the beginning of the development project. The argumentation is that the beginning of the project is most turbulent and therefore needs most communication to understand (changing) requirements.

Building a shared mental model might be a more fundamental reason for the need for communication in that phase (Mathieu, Heffner, Goodwin, Salas, & Cannon-Bowers, 2000), as such a shared mental model benefits the interpretation of shared information.

Mishra and Mishra (2011) studied a complex development project with multiple teams. The large project size brought the necessity of (1) having sufficient communication between Scrum teams to synchronize work, (2) informing each other about progress and (3) discussing unresolved issues leading to adjusted plans. They mention brainstorming and iteration planning meetings as instruments to share information. Another way to achieve communication is using regular cross-team meetings for informing each other about progress and issues. They conclude that (1) the large size of the project required certain levels of control and (2) flexibility and architectural designs helped to communicate a clearer picture of the entire system. Most of their implications for practice concern business-IT alignment, not communication between Scrum teams. A practical way to organize cross team communication is presented by Kniberg and Ivarsson (2012). Their article reports about the use of tribes at Spotify to cater for the necessary communication. They describe an organizational setup for multiple teams to stimulate knowledge sharing and communication between the teams.

4.2.3 Automation issues when applying Scrum in large enterprises

One of the key successes of information technology (IT) is the automation of business processes. IT can much quicker execute repeated tasks and process vast amounts of data. The same principle is now increasingly applied to software engineering processes. Automation of software engineering processes enables very short time to market of new software, such as at Facebook (Feitelson et al., 2013). The Agile community underlines the importance by making software engineering process automation one of their core Agile principles (Beedle et al., 2013). A. W. Brown, Ambler, and Royce (2013) used a software cost perspective to explain the benefits of automation. They simplified a typical software cost model with dozens of parameters to three essential parameters. They conclude that software process automation is one of the essential parameters to reduce the time to market of software releases.

The automation of software engineering processes concerns the automating of the integration, testing and deployment processes. The key characteristics are Continuous Integration, Continuous Testing and Continuous Deployment, part of the umbrella Continuous Delivery (Beedle et al., 2013; Humble & Farley, 2010).

Continuous Integration stands for the immediate build and integration of a new piece of software as soon as it is ready and checked into the configuration management system (Garg, 2009; Humble & Farley, 2010; Jyothi & Rao, 2011; Ståhl & Bosch, 2013, 2014). Kim, Park, Yun, and Lee (2008) briefly mention two Continuous Integration related issues in their paper: (1) the sensitivity of integration mistakes by the packaging maintainer which blocks the integration automation and (2) developers using own testing environments which are incompatible with the integrated testing environment, blocking the integration test.

Continuous Testing is synonym for the automated execution of a series of automatic tests, verifying the integrity of the updated system (Candea, Bucur, & Zamfir, 2010; Humble & Farley, 2010; Muslu, Brun, & Meliou, 2013). The papers about Continuous Testing that we found during the literature study, such as the experience papers of Muslu et al. (2013) and Candea et al. (2010), do not mention any Continuous Testing related issues.

Continuous Deployment is the discipline of automated distribution of the new version of the system from the development stage to the production stage (Feitelson et al., 2013; Hardion et al., 2013; Humble & Farley, 2010). Olsson et al. (2012) present a multiple-case study in which key-barriers associated with the transition towards Continuous Deployment are identified, being (1) the struggle with traditional processes that hindered 3C and continuous improvements, (2) traditional tooling that made development unnecessary complex and heterogeneous configurations, (3) a challenging cultural shift to new values and (4) a lack of transparency and a short feedback cycle.

Recently, Fitzgerald and Stol (2014) stretched the ‘Continuous’ concept and practice with the introduction of ‘Continuous *’. The Continuous * concept includes Continuous Planning, which involves dynamic open-ended plans that evolve in response to changes in the business environment, and thus involve a tighter integration between planning and execution (Fitzgerald & Stol, 2014; Knight, Rabideau, Chien, Engelhardt, & Sherwood, 2001). The Continuous Planning concept involves planning and tracking of the software delivery process, while Continuous Delivery comprises the automation of the software manufacturing and software delivery itself. We found no papers that mention any Continuous Planning related issues.

We conclude that most of the existing related work about software engineering automation concerns case studies about automation technology. Issues in software engineering process automation are scarcely mentioned and we found no studies about automation issues in chains of Scrum teams.

4.3 Research Method

As existing academic literature provides insufficient answers to our research question, we selected and performed (Saunders et al., 2009) a holistic multiple-case study at multinational service providers, to gain an in-depth understanding of the issues in interdependent chains of Scrum teams and develop new theory (Dul & Hak, 2012). A method for inductive research that builds new theory is presented by Eisenhardt (1989). Her work syntheses previous work on qualitative methods, case study research and grounded theory building (Glaser & Strauss, 1967; Huberman & Miles, 1984; Yin 1981). She presents a process of building theory from case study research that has the steps: (1) getting started, (2) selecting cases, (3) crafting instruments and protocols, (4) entering the field, (5) analyzing data, (6) shaping hypotheses, (7) enfolding literature and (8) reaching closure.

We use Eisenhardt (1989) for our research method. We slightly adapted the method to fit our needs by splitting the design into two sequential stages as shown in Figure 22. Stage A targets Scrum chain selection and role mapping, which prepares stage B that aims finding the issues.

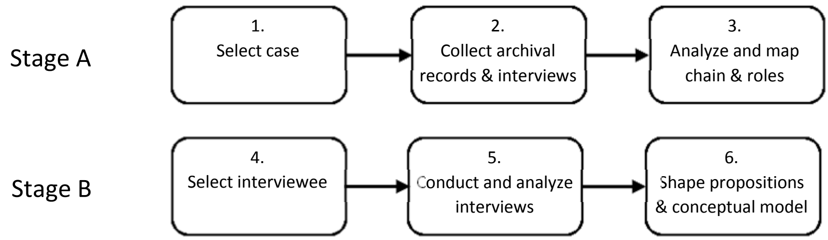

Figure 22, Research design

The research starts with selecting an interdependent chain of Scrum teams by using selection criteria as shown in Table 15. We follow the project level contextualization attributes of Kruchten (2013) to develop the criteria for our study. The attributes of Kruchten help to identify the context of an Agile environment. The attributes are size, architecture stability, business model, team distribution, change rate, system age, criticality and governance. The selection criteria enable the identification of the unique characteristics of a chain of Scrum teams and enhance the content validity of the research.

Table 15, Case study selection criteria

Selection category	Selection criteria for each team in the chain
Application interdependencies	Applications have interdependencies with other applications in the front to back chain.
Chain setup	Scrum teams are part of a chain and each application under development is allocated to one Scrum team.
Team distribution	Studied Scrum team members are working in the same country.
Culture	Studied Scrum team members have the same nationality.
Scrum roles	Scrum roles product owner, Scrum Master and Scrum team are assigned.
Scrum processes	Scrum sprints, sprint planning meeting, daily standups, sprint demo and sprint retrospectives are performed
Scrum artifacts	Product backlog and Scrum backlog is available and product increments are delivered

The chain of Scrum teams is selected in step 2, by collecting archival records and conducting interviews by phone. These interviews are annotated in an interview log. The data is subsequently analyzed to map the chain and identify the involved roles. Steps 2 and 3 iterate until the mapping is complete. Subject matter experts working in the network are involved to secure the validity of the mapped results (Gibbert & Ruigrok, 2010).

The interviewees are selected in step 4, based on the identified roles in step 3. The interview process is flexible; in case another involved role is identified during the role can be included in the interviews.

The interviews itself are conducted in step 5. Open-ended questions are prepared to guide the interviews, as shown in Table 16. We use Galbraith's (1995) framework to study a broad range of aspects in the chain of Scrum teams. Galbraith's framework has five dimensions to assess organizations: (1) strategy, which determines direction, (2) structure, which determines the location of decision-making power, (3) processes have to do with the flow of information; they are the means of responding to information technologies, (4) rewards provide motivation and incentives for desired behavior and (5) people, which is about the selection and development of the right people.

Each of Galbraith's dimensions is translated to our research context. We use the (1) strategy dimension to find the issues in translation of strategy to operational plans and the (2) structure dimension to understand and find the issues in the existing roles and responsibilities. We use Galbraith's (5) people dimension to assess the issues in mindset and competences. Galbraith's (4) reward dimension is kept out of the scope of this research, as analyzing the issues in human resource performance management does not fit in this study. We use Galbraith's (3) process dimension to find process related issues between Scrum teams. Galbraith's process dimension is specified in terms of collaboration, coordination and communication (see Table 16), which have a key role in Agile settings as discussed in subsection 4.2.2.

Table 16, Predefined interview questions

Question	Grounding
What are the main strategic to operational level translation issues with the Scrum chain setup?	Galbraith's strategy
What are the main roles and responsibilities related issues with Scrum chain setup?	Galbraith's structure
What are the main inter Scrum team coordination issues with the Scrum chain setup?	Galbraith's process, 3C coordination
What are the main inter Scrum team collaboration issues with the Scrum chain setup?	Galbraith's process, 3C collaboration
What are the main inter Scrum team communication issues with the Scrum chain setup?	Galbraith's process, 3C communication
What are the main mind-set and competence related issues with the Scrum chain setup?	Galbraith's people

Step 5 starts with one interview that is subsequently analyzed. Follow-up interviews are used to refine the script and the analyzed results. Such iterative approach allows script optimization for successive interviews and the mitigation of potential problems as summarized by Myers and Newman (2007). The interviews are digitally recorded for content- and construct validity purposes (Mentzer & Flint, 1997).

The interview setup is built on the dramaturgical model, using the metaphor of a theatre (Myers & Newman, 2007). The dramaturgical model is based on the general theory of Goffman (1959) that sees social interactions as a drama, with actors that perform in a variety of settings using a script that guides behavior. During the interview a delicate balance is kept between providing direction and getting unbiased answers, while mitigating the potential interviewing pitfalls as summarized by Myers and Newman (2007). The objective and the topics of the interview are set at the beginning of the interview, to set a framework that can be gently referred to in case the interview moves off-topic.

The interview results are transcripted and qualitative analysis techniques are used to analyze the transcripted data (Dul & Hak, 2012; Yin 2009). The qualitative analysis starts by identifying quotes in the transcriptions (Saunders et al., 2009). The quotes are tagged with open codes. The open coding process is performed in a number of iterations (Saldaña, 2012). If a quote applies to multiple open codes the quote is tagged with all applicable codes. For instance the quote: '*The application that is developed by the team is connected with an interdependent application via an enterprise service bus",* is tagged with the open codes *'ApplicationInterdependencies'* and *'ApplicationDevelopment'*.

Open coding proceeds line-by-line, until patterns emerge. These patterns of open codes are grouped into categories, the axial coding process. The axial coding process is done in two ways. First, codes are grouped by prefixes in the tags. For instance all codes which are related to issues are tagged by the prefix '*Issue*'. Second, codes are grouped in code families. For instance performance metric related codes are placed in the code family '*PerformanceMetrics*'.

Data collection continues until a new transcription does not significantly contribute to knowledge and insight (Dul & Hak, 2012). Significantly contributing is quantified by determining whether an additional interview results in more than 5% new or modified codes. In case an interview results in more than 5% new or modified codes an additional interview is conducted. This approach is in line with Sandelowski (1995) and Marshall (1996).

After the axial coding process the open codes tagged with five or more quotes, are clustered into the main categories for which supercodes are used. Supercodes are queries for retrieving a selected set of codes. For instance a query can be built to retrieve all quotes that are linked to open codes that have the word '*Issue*' in their name at one company. The supercodes are used to ground the concepts, which are the (key) issues. All steps of the data analysis are recorded in Atlas TI, a CAQDAS package (Gibbert & Ruigrok, 2010; Saunders et al., 2009).

The issues are used during step 6 to define the propositions. The relationships between the issues that result in the propositions are grounded in existing theory by an additional study of related work. The propositions are then used to build the conceptual model (Birks & Mills, 2011).

4.4 Issues identified

Using our professional network, we identified five cases at large multi-national organizations on which we applied the selection criteria. Three cases passed the criteria and were subsequently used in our research. We performed one case study at a telecommunication company (9 interviews in two overlapping chains), one at a retail bank (6 interviews in one chain) and one at an insurance company (3 interviews in one chain). Each of the companies has a centralized IT organization with 250-1500 IT development employees who offer application services to front to back business functions.

For the interviews, we selected the key roles Product Owners, line managers and Scrum Masters. We noticed in the preliminary interviews that the role of Scrum Masters was overlapping with that of the team members, in that the Scrum masters also performed development work. Also the Scrum literature considers Scrum Masters part of the development team. We therefore decided that interviewing the Scrum Master was sufficient and no other Scrum team members were interviewed. We transcribed the interviews and coded them as described in Section 4.3. Table 17 shows for each identified issue a brief description and the grounding of each issue by the number of quotes.

Table 17, Code grounding of each issue

Issue description	Issue	Nr of quotes
A lack of coordination in the chain	Coordination	124
Mismatches in backlog priority between teams	Prioritization	122
Alignment issues between teams	Alignment	99
A lack of IT chain process automation,	Automation	72
Unpredictability of delivery to commitment	Predictability	56
A lack of information visibility in the chain.	Visibility	38
	Total	511

Table 18 shows for each issue the percentage of quotes in each case study. The table shows that most of the coordination issues occur in case 1. Most prioritization and alignment issues occur in case 2. The highest percentage of predictability and visibility issues occurs in case 3. Automation issues are (relatively) equally often mentioned in all three cases.

Subsections 4.4.1 – 4.4.3 contain, for each case, key quotes for each concept. The issues in each subsection are sorted in the order in which they appear in Table 18. The quotes are labeled between the brackets '[ ]'. These labels are used in section 5 to refer back to the quotes

Table 18, Cross-case analysis of the issues

Issue	Case 1	Case 2	Case 3
Coordination	29%	16%	18%
Prioritization	22%	35%	17%
Alignment	20%	22%	13%
Automation	14%	15%	15%
Predictability	11%	5%	17%
Visibility	4%	7%	20%
Total	100%	100%	100%

4.4.1 Case study 1: Retail banking front to back application chain

The retail bank has a centralized IT organization with 150 Scrum teams that deliver web application services to the various business lines that deliver the services to the banking customer. The web applications are developed by front-end Scrum teams that are clustered around different channels (e.g. internet, call center). The web applications interact with back-office applications that are clustered around financial products. The back-office applications are mainly commercial of the shelf (COTS) packages. These packages are configured by Scrum teams. The back-office packages interact with finance applications that are developed by waterfall organized development teams. The organization has allocated an integrator role, next to the regular Scrum-roles. The integrator is responsible for, monitors and coordinates front to back feature development.

Coordination: Issues regarding coordination typically express themselves during testing and deployment, because in the end of the development lifecycle all teams should have delivered their piece of the puzzle. To ensure that the feature properly works in production, an integration test is executed that needs to be coordinated between multiple teams that each deliver their piece:

> *[C1a] "A lot of communication is required to ensure that each piece of the puzzle is available at the same time in the test environment to ensure that we can properly test the new feature",* Integrator

The interdependencies require extensive coordination between teams, which is performed by an integrator role. The integrator role however experiences a lack of influencing power while coordinating the Scrum team activities. The lack of influencing power makes coordination much more labor intensive and difficult:

> *[C1b] "Scrum teams are stimulated to close the hatches and concentrate on and realize what is on top of their backlog….. I'm drowning in coordination activities*

because teams need to do it themselves while they are only focusing on their own team", Integrator

Prioritization: One of the key difficulties is having the activities properly prioritized on the product backlog in each of the codependent Scrum teams. Only if the priorities of all codependent teams match, a front to back feature can be delivered. However, priorities are often mismatched. Since an integrator is responsible for delivering a front to back feature he gets stuck in case of mismatching priorities as he has not much influencing power to change the priority in one of the backlogs:

[P1a] "The product owner does not want to cooperate in delivering inter-dependent functionality…. I do not have any influencing authority", Integrator

The responsibility of the integrator however brings expectations that have to be fulfilled. The integrator therefore tries to fix the front to back prioritization which is considered an exhausting exercise:

[P1b] "Each of the Scrum teams needs to have the required activities on their backlog, such as having the test environment available and determine test cases…. To achieve this cost an tremendous amount of energy", Integrator

Product owners determine their backlog priority based on the set strategic objectives in their management line. However, the management lines have different strategic priorities leading to mismatched backlog priorities:

[P1c] "The managing directors in our board have different priority settings. The managing director would like to improve his business process, while the manager of the Internet Channel needs to resolve a number of compliance issues and the manager of Marketing and Sales want to develop software for new product offering. They all think that their objective will be achieved", IT development manager

Such mismatches leads to a situation that each Scrum team develops software for its own product owner, while at the end of the sprint nothing works front to back.

A shared design between teams helps to determine the backlog items that enable a smooth delivery of features. Architects and groups of designers therefore play a significant indirect role in the prioritization process:

[P1d] "The design is very important, created by the architects. But also lead developers. For instance architects define the services and interfaces between the applications of the Scrum teams", IT integrator

Alignment: One of the alignment issues is the difference in deadline between Scrum teams. In case teams work towards a shared deadline less coordination is required. However in case the deadlines differ, a team that is ready needs to repeat activities in the next sprint, while the product backlog of that team has a different focus by that time:

> *[M1a] "If one team is ready and the other team still needs to develop their functionality, the team that is ready needs to store their functionality and has to process new changes before feature testing starts. In that case the discussion starts all over again for the team that was ready, because the flow differs between the teams",* IT manager

Alignment issues are in particular perceived during front to back feature testing. Testing is a complex, labor intensive and highly interdependent exercise. During testing teams need to jointly prepare and execute front to back feature testing, requiring the alignment of the interdependent activities, definitions and terminology.

> *[M1b] "A lot of work needs to be redone to synchronize test data to ensure that the correct test data is available in the test environment that needs to be used in the internet channel",* Integrator

Automation: An automation issue concerns the tooling used for status and progress tracking of the backlog items over the chain. The used tool is particularly useful for tracking individual Scrum teams, but not considered useful to track progress of front to back features:

> *[A1a] "Jira is not really useful on chain level. The tool needs to be adapted to make it work on a chain level. It is more an item driven tool",* IT manager

The issue of status and progress tracking is also mentioned by an IT integrator. Next to the suitability of the tool itself the interviewee also mentions the accuracy of the entered information as issue, which hinders coordination activities:

> *[A1b] "Entering information has limited attention. The teams are mainly working on development, administration gets less attention. It is (therefore) very hard to grasp the development progress… They tried to achieve this with Jira. In theory this works fine but in practice it does not work. I have far too limited possibilities to track such progress",* IT integrator

Predictability: One of the key predictability issues is the development of a single software package, by multiple Scrum teams in parallel. The software package was not designed to support such parallel development, resulting in potential interference between Scrum teams:

> *[U1a] "We have one monolithic software package that is changed by multiple Scrum teams. How do I know that the development in one Scrum team does not conflict with the development that is performed by another Scrum teams. This has to be well thought of",* IT manager

Visibility: The lack of information visibility about the status and progress is also perceived as an issue. Interviewees mention that there is a lack of supporting IT tools that can provide such visibility:

> *[V1a] "Tooling about status and progress is far too limited and too little available at the moment. Such lack makes the work of the integrator very difficult",* Integrator

"Such tools help to keep overview over the development cycle", IT manager

4.4.2 Case study 2: Telecommunication front to back application chain

The telecom company has a centralized IT organization having 34 Scrum teams. The IT organization has a cluster of front-end Scrum teams which develop applications that interact with Scrum teams of Operation that interact with Scrum teams for Billing and Finance. The involved applications are interdependent, implying that a feature can only be delivered by the constellation of front to back applications. We interviewed Product Owners, Scrum Masters, IT development managers and project managers.

Coordination: Also in the telecom case coordination issues are mentioned. Both the project manager and the IT manager recognize the inward team focus instead of a focus on the front to back chain. Such inward focus makes it difficult for boundary spanning roles such as a project manager to motivate collaboration. As a result the team delivers their specific components and not an operational feature at the end of the sprint.

> *[C2a] "I find the biggest challenge still the end-to-end chain. I perceive that a lot of teams still have a component story mentality. But, we have that XML delivered! Great! But in that case you are mentally not working in a chain",*
> IT manager

> *[C2b] "For me it is still difficult to get all those people together. For me it is very important that we have refinement sessions with the front to back product owner and the story owners of each involved team. For instance the story*

owners should provide a coarse estimation of the effort…. It is very important that everybody is available. It has no value to do this with each team separately”, Project manager

Prioritization: Priority mismatches between interdependent teams are mostly mentioned in this case (35%). Such priority mismatches are caused by multiple business projects concurrently being executed, while having executives being responsible for their own profit and loss. For instance the front-office, mid-office and back-office each have their own executives, each with their own unique conflicting targets and also the different business lines have their own specified targets

[P2a] “We are executing 15 programs concurrently…. Each program has its executive at business side which has its own profit & loss. So, who determine priority over the chain when each executive wants to have their features with highest priority”, IT manager

[P2b] “The minimal shippable product is a real challenge and creates a lot of tension. Because commerce, operation, IT and network all have a different definition what should be completed. Commerce want a many features as possible and operations want to have a stable solution”, IT manager

[P2c] “Commerce consumers want their new product on the market asap, while that new product has no value for commerce business and want their new product on the market.”, IT manager

For high priority features the needed stories in the interdependent teams are placed as highest priority items on the team backlogs. However each involved executive influences the priority setting of a subset of product owners in the chain, resulting in priority mismatches over the front to back chain. Such conflicts result in impeded delivery of the front to back features:

[P2d] “In team A an end to end feature has highest priority, at team B however a dependent story has low priority due to interdependent stories that need to be delivered first”, IT development manager

[P2e] “Sometimes in one team a story has high priority and in the other team the story has low priority while both stories are needed for a feature. In that case a team uses best-effort stories; they think they can deliver this. With a feature this is difficult since at the end nothing is delivered”, Project manager

Alignment: We also found alignment issues between teams in this case study, such as difficulties to align the definition of done. A definition of done is defined per team and not aligned over all involved teams:

> *[M2a] "Yes we have a definition of done, but try to align that over the full chain with clear requirements and acceptance criteria. This is still very hard.",* Product owner

Another alignment issue mentioned are the test stubs that functionally differ from the simulated interface. Such differences result in impediments at the end of the sprint that results in failing delivery of a feature.

> *[M2b] "If team A and team B agree that team A creates an interface that sends 'ABC' and team B understands 'ABD' then team A is happy with their delivered 'ABC' functionality. However at the end of the sprint there is a nasty surprise, when everything has been integrated. That is the disadvantage of stubs",* Project manager

Automation: We also found a lack of automation about the tracking of backlog items. This is largely managed by using Excel sheets. Jira is also used but only limitedly on an intra-team level:

> *[A2a] "We mostly use Excel sheets. We have Jira. However there is no proper support on Jira, so this is not properly configured. We want to do this but we already trying to achieve that for 9 months. Even burn-ups and burn-downs are not yet properly configured",* Product owner

> *[A2b] "Using Jira over the chain is very difficult. I'm managing the front-office, with all my teams. Siebel is located in the middle. We cloned the higher level items in Jira to reuse them. However in this case way we disconnected the linkages between teams. If later a story owner said that a new interface was implemented the other team reacted only with… Oh??",* IT manager

Predictability: A small percentage of quotes (5%) concerned unpredictability. The main issue we found concerns the dependencies between teams. In case one team cannot deliver, the deployed feature is delayed:

> *[U2a] "Misalignment in timing between teams happens regularly. Recently I had a feature which was on the list of each involved team. However one of the teams that had delays earlier now experienced test defects, while having a due deadline of the code freeze. At the end of the sprint nothing was delivered", project manager*

Visibility: We found visibility issues that are related to automation. One typical issue is the status of backlog items that is often not visible throughout the chain, because most of the agreements are made informally and not documented:

> *[V2a] "A lot of decisions are made in an informal way, by verbal haggling at the coffee machine. Not much is documented", Product Owner*

4.4.3 Case study 3: Insurance front to back application chain

The insurance company has a centralized IT organization that delivers web application services to the various business lines that deliver the services to the insurance customer. The web applications and user interface components are developed by a cluster of five codependent front-end Scrum teams. When the produced story meets the Definition of Done the product is handed over to a separate integration team, which tests and deploys the developed product to production. We interviewed Scrum Masters and an Epic Product Owner in one cluster.
The cluster of front-end Scrum teams works with a single prioritized backlog. Each of the Scrum teams picks the highest priority story from the backlog and develops the story. Less prioritization conflicts within the cluster of Scrum teams occur, compared to the other two cases, as a result of having a single backlog (see Table 18).

Coordination: Chain coordination is achieved by project managers. One typical coordination issue is the lack of control that project managers have over the Scrum teams. Project managers are used to have full control over project resources. However in the Agile way of working project managers have no direct control, and agree the work with product owners that prioritize the work.

> *[C3a] "For the project manager it is challenging as he needs to get work done from the Scrum teams, while he has no direct influence on the work in the teams, in contrary to the classical waterfall approach in which a project manager has direct control over project resources",* Scrum Master

Coordination issues between Scrum teams are also mentioned. A Scrum team does not have influencing capabilities on the activities of another Scrum team, which results in rigidly limiting accountability to their own team.

> *[C3b] "Classically our stakeholders try to hold us responsible for the front to back chain, including connectivity. However it is unpleasant to be responsible for something that you cannot influence. We therefore make explicitly clear that this is our responsibility and if you want to bring something life other items need to be realized as well",* Scrum Master

Prioritization: Also in this case we found evidence of priority mismatches between Scrum teams caused by conflicting objectives at the strategic level. One Scrum Master states that senior management is simply not familiar with negotiating and decision making:

> *[P3a] "On senior management level managers are unfamiliar with such choices. They are not used to come together with six people and listening to each other's needs and jointly decide on priority",* Scrum Master

In this case a single prioritized backlog over multiple teams is used. However conflicting objectives at the strategic level obstruct an unambiguous prioritization of the backlog:

> *[P3b] "And let us know what we need to start with first. But… sometimes that is not clear. Sometimes there are two conflicting objectives that each needs to be completed first…. At senior management level they are unfamiliar with these kind of choices and sitting at the table together making decisions which goal is accomplished first. They all want that of course",* Scrum Master

Alignment: A typical alignment issue that was mentioned in this case was the different way of working of each of the teams that need to work together to jointly deliver features. To achieve collaboration in such situation is considered difficult:

> *[M3a] "The requirements have been distributed over the different teams, which each needs to adapt their software… the chains however are heterogeneous, with different technologies and a different way of working. During chain integration testing we discover whether everything is working. That is a very tiring process",* IT manager

Automation: Also in this case we found the need for reliable status and progress tracking over the chain of Scrum teams. The company has an integrated suite to automate status tracking, including event handling of the integration software:

> *[A3a] "You need to take care of the work items. To achieve that is a case of discipline, because an engineer programs his lines of code. If he checks in the source, his change is integrated in the code base and the work item is updated. It is all very data warehouse like",* IT manager

Predictability: We found that delivery unpredictability can be caused by misjudgments in the planning. Such misjudgment leads to a lot of extra work at the end of the sprint that can easily lead to an unrecoverable delivery impediment:

[U3a] "We did not do something well in the planning. We missed a very important component, which led to straightening out the back-office integration at the last moment", Scrum Master

To mitigate delivery unpredictability the front-end team uses sequential back to front development, implying that the front-office waits until the required back-office application has been delivered:

[U3b] "The process is asynchronous; first there must be a back-office application to be able to define the amount of work", Scrum Master

Even though the predictability of feature delivery might be increased, such mitigation slows down time to market significantly. For instance a feature that needs to be developed in a chain with four teams requires four sprints to deliver the feature.

Visibility: The found visibility issues concern a factual overview of the backlog items' status in the various Scrum team. Instead of using relative indicators the company is working on making the delivery pipeline transparent to stakeholders, by offering factual reports about the tracking of (requirement) backlog items:

[V3a] "His requirement is still in the backlog. He can call us of course, asking when we will start picking up his requirement…. Our management want to understand when something start to happen", Scrum Master

[V3b] "Whatever a red or green smiley is, it is just an undetermined judgment of somebody. And we then start discussing the meaning of such indicators, not about the real progress. So we are now working on making the production line of IT transparent…. The test is 67% succeeded and 33% failed. Two bugs are still open and the progress of a user story", Scrum Master

4.5 Towards a conceptual model integrating the issues

We identified six issues in chains of codependent Scrum teams: (1) a lack of coordination in the chain (2) mismatches in backlog priority between teams, (3) alignment issues between teams, (4) a lack of IT chain process automation, (5) unpredictability of delivery to commitment and (6) a lack of information visibility in the chain. See also Table 17. The number of quotes counted in the interviews is taken as the weight of the issues.

In the following subsections we discuss each of these issues. We synthesize the empirical results and existing theory to build propositions, which we then use to build the conceptual model. While doing so, we refer back to the related work discussed in section 4.2. The subsections are sorted in accordance with the design of the conceptual model derived in subsection 4.5.7. The items between brackets '[ ]' refer back to the quotes in section 4.4.

4.5.1 Predictability

The first issue is the unpredictability of whether a feature will be delivered by the interdependent chain of Scrum teams (56x grounded). Each team has to deliver its application functionality for the feature. In case one of the teams does not deliver its functionality, the feature will not be delivered or is at least delayed [U1a, U2a, U3a]. The risk of one team not delivering its functionality increases with the number of codependent teams. For instance if each team delivers the necessary functionality in 9 out of 10 cases and the chain consists of 10 codependent teams, the chance of a feature delivered in the sprint is $(0.9)^{10}$, which is less than 35%. In case a feature is not delivered in a sprint, all involved teams need to spend time on that feature in the next sprint. The team that did not deliver needs to manufacture the functionality in the next sprint and the teams that did deliver need to do rework, such as retesting and bug fixing. Unpredictability is therefore considered a major cause of cost increase and longer time to market. Yet, in the existing literature as discussed in section 4.2 predictability is mentioned implicitly or the Scrum framework is asserted as having a positive impact on predictability (Dove & LaBarge, 2014).

4.5.2 Coordination

We identify lack of coordination between Scrum teams as the most often mentioned issue in chains of Scrum teams (124x grounded). The lack of coordination expresses itself during the full development lifecycle [C1a, C2b, C3a]. In the first case study an integrator role coordinates the activities between Scrum teams [C1b] and in the second and third case business project managers take care of these coordination activities [C2b, C3a]. In all three cases these coordinating activities are performed by an actor (integrator or project manager) which is not part of any Scrum team. Each of these interviewed roles perceives a lack of influencing capabilities. The interviewees mention that the Scrum teams only focus on their own backlog and not on delivery of (front to back) features [C1b, C2a, C2b, C3a, C3b, P1a]. The lack of influencing capability results in unpredictability whether a feature gets delivered.

Scrum of Scrum meetings are typically used to coordinate activities between Scrum teams and issues with these Scrum of Scrum meetings do occur, as discussed in subsection 4.2.2 (Paasivaara et al., 2012). However, hardly any issues with regard to these meetings are mentioned by the interviewees.

Coordination manages the interdependent activities between the codependent Scrum teams (Malone & Crowston, 1990; Sharp & Robinson, 2010). Coordination theory helps to understand the deeper concept of coordination. Coordination theory is described by Malone and Crowston (1990) and defined as: *"a body of principles about how activities can be coordinated that is, about how actors can work together harmoniously"*. The components of their definition of coordination are: (1) goal identification, (2) activity mapping on goals, (3) actor assignment on activities and (4) interdependency management. The Scrum framework contains these components on a Scrum team level. For a chain of Scrum teams however these components are not supported by the Scrum framework, while they are of critical importance.

Scheerer, Hildenbrand, and Kude (2014) use coordination theory to compile a conceptual framework in Agile settings with three forms of coordination: (1) mechanistic coordination - coordination by plan or rules with little communication, (2) organic coordination - coordination by means of mutual adjustment or feedback via interaction, which can be formal and planned or informal and spontaneous and (3) cognitive coordination - based on explicit and tacit knowledge the actors have about each other, such as a shared mental model (Mathieu et al., 2000). Their conceptual framework shows that coordination should be embedded at the organic and cognitive level, next to mechanistic coordination.

Mapping the conceptual framework of Scheerer et al. (2014) with the coordination practices of the three empirical case studies reveals that only mechanistic coordination has been implemented. Organic and cognitive coordination between Scrum teams is hardly existent due to the explicit Scrum policies, such as the focus on the team specific backlog, fixed sprint cycles and predefined roles, with limited focus on front to back coordination. The mechanistic form of communication seems also ineffective, since coordination is performed by a coordinator role that has limited mandate. We argue that coordination practices should be deeply embedded within and over the Scrum teams by implementing the components from Malone and Crowston (1990) on a Scrum chain level, while utilizing all three forms of coordination as proposed by Scheerer et al. (2014). Deeply embedded coordination will result in better coordination between teams resulting in increased predictability in each of the teams and therefore more delivery predictability. We therefore propose:

> ***[P1]*** *Embedded coordination practices within and between Scrum teams positively impact delivery predictability*

4.5.3 Prioritization

Priority issues have been identified by multiple researchers (Begel et al., 2009; Lehto & Rautiainen, 2009; Petersen & Wohlin, 2009; Waardenburg & van Vliet, 2012), as discussed in section 4.2. However the related work is unclear about the research environment and interpretation of the research results is therefore difficult.

In our case studies, mismatched backlog priority is considered the second largest issue in a Scrum chain (122x grounded). Priority of a backlog is set by a product owner. The product owner sets his priority based on the input the product owner receives, typically derived from the strategic objectives. The goals however differ at strategic level [P1c, P2a, P2b, P2c, P3a]. For instance the goals of the front-office function (e.g. sales) naturally differ from those of a finance function. When each product owner sets his backlog priority based on their (strategic) goals [P1a, P2b, P2c], prioritization over the front to back chain gets mismatched. Mismatched priority setting makes each Scrum teams concurrently developing different functionality for different features [P1b, P2d, P2e], and at the end of the sprint likely no feature gets delivered. Priority mismatches consequently result in increased delivery unpredictability, leading to our next proposition:

> ***[P2]*** *Matching priority over the front to back chain positively impacts delivery predictability*

Prioritization is a way to set goals (Gregory et al., 2011; Locke & Latham, 1990), while goal setting is one of the components of coordination theory (Malone & Crowston, 1990). Matching priority over the front to back chain implies a single goal for all Scrum teams, embedding one of the coordination theory components. We therefore propose:

> ***[P3]*** *Matching priority improves front to back coordination practices*

We found additional related literature in the area of strategic decision making. Strategic decision making is a bounded rational process (Eisenhardt & Zbaracki, 1992). The bounded rational process is caused by the cognitive limitations and motivational and emotional factors (Bazerman & Moore, 2009). These cognitive limitations result in a biased perception of reality and biased decision making.
A decision maker is also influenced by social and political factors (Boonstra, 2003). For instance the business manager (strategic decision maker 1) might be put under pressure to reach a sales goal and therefore gives a risk mitigation program lower priority, while the chief risk offer (strategic decision maker 2) might have the strategic goal to lower overall financial company risk. The product owner of the front-office Scrum team will likely be put under pressure to prioritize the backlog in accordance with the sales goal, while the product owner of the finance Scrum team will likely prioritize in accordance with the risk reduction goal.
We argue that priority matching over the front to back chain will be improved, in case the priorities match at the strategic level. Such strategic priority matching requires the implementation of strategic decision making strategies, such as (1) using decision-analysis tools, (2) acquiring expertise, (3) debiasing judgments, (4) analogical

reasoning, (5) taking an outsider view and (6) enhancing the understanding biases of peers (Bazerman & Moore, 2009; Eisenhardt & Zbaracki, 1992). Supported by bounded rationality theory and decision making strategies we argue that:

> ***[P4]*** *The implementation of decision making strategies improve matched priority setting*

4.5.4 Alignment

Another issue is the misalignment between the codependent Scrum teams (99x grounded) [M1a, M3a]. Although Scrum has a prescribed structure, the working processes can be implemented in many different ways, leading to such misalignments.
An identified misalignment is the definition of done [M2a]. One team defines 'done' as delivered before system testing and another team defines 'done' as delivered including system testing. A second identified issue is the misalignment of the start, the finish and duration of the sprints. One Scrum team has a two week cycle and another team has a monthly cycle. A third identified issue is the misalignment of test activities and test results between Scrum teams [M1b, M2b].

The need for alignment is also reported by Saddington (2012) (see section 4.2). However Saddington (2012) targets the alignment between product owners which is covered as priority matching in our case study.
The misalignment between codependent Scrum teams causes unpredictability and delivery delays. For instance if the sprint of team B ends two weeks later than team A the delivery of the feature is delayed until team B has completed its sprint. The delivery even gets blocked in case team A is has no feature testing item on the backlog of their next sprint. Because when team B is ready to test their functionality, team A is not available for testing. We therefore propose:

> ***[P5]*** *Alignment between Scrum teams positively impacts delivery predictability*

The Scum framework does not support alignment between Scrum teams. A Scrum team will optimize its working processes for its own purposes. Only in case the teams have an interest in an aligned way of working, and are able to understand and influence each other's way of working, alignment is enabled. Supported by coordination theory we argue that (1) a common shared goal and (2) a coordination mechanism improve alignment between Scrum teams.
(1) The key goal of a Scrum team is to add business value, which is one of the control theory components (Malone & Crowston, 1990). With codependent feature delivery the value is realized by codependent Scrum teams. The primary goal of a Scrum team should therefore be to realize a front to back feature, instead of realizing top priority items on their backlog. The goal on realizing features will face team members with

misalignments and motivate them to proactively align their way of working. We therefore propose:

> ***[P6]*** *Matched priority setting positively impacts the alignment between Scrum teams.*

(2) A proper coordination mechanism should be implemented as conceptualized by Scheerer et al. (2014). Mechanistic coordination is achieved by the implementation of policies. Organic coordination can be achieved by implementing communities of practice (CoP), such as a test CoP with the objective of aligned front to back testing. Cognitive coordination is achieved by implementing a shared mental model and transactive memory (Jonker, van Riemsdijk, & Vermeulen, 2011; Lewis & Herndon, 2011; Wegner, 1987). Misalignment in the shared mental model between teams causes misunderstanding and misinterpretation (see also Hildenbrand et al. (2008) in subsection 4.2.2). The more of the mental model is shared, the more cohesive the chain can operate (Mathieu et al., 2000). Given these three contributing types of coordination we propose:

> ***[P7]*** *Coordination practices positively impact the alignment between Scrum teams*

4.5.5 Visibility

We also identified the lack of information visibility as an issue (38x grounded). Interviewees mention the need for visibility over the prioritized backlogs, the development status of a feature and the underlying functionality [V1a, V2a, V3a]. Such information enables teams to take mitigating actions and manage expectations. A lack of visibility disables teams to take appropriate action, which leads to uncontrollable impediments later in the sprint. An example illustrates the need for visibility: a Scrum team needs to realize a backlog item for a feature. After a week the realization of the item is impeded, putting feature delivery at risk. In case codependent teams have visibility over the backlog they become aware of the impediment and can take the necessary mitigating actions.

Control theory theorizes the role of visibility in the chain and the positive impact on coordination practices between the Scrum teams. Although control theory is historically used as a mathematical model to explain the behavior of physical systems, the basics can be also applied to human actors (Andrei, 2006; Vlietland & van Vliet, 2014b; Wiener, 1965). Control theory consists of three fundamental concepts. The first one is goal setting, which in this case is set by the prioritized backlog items. The members of Scrum teams take action to realize the backlog in case of sufficient level of visibility over the set goals; see also Saddington (2012) in section 4.2. The second concept of control theory is feedback. Feedback informs the actors about the actual status and progress of codependent teams, again in case of sufficient visibility. The third concept is the comparison function that compares the actual value with the goal. The difference between the two values is fed into the action process, resulting in adapted interdependent action by the team members in the codependent teams.

Looking from a visibility perspective reveals that the components of coordination theory, as described by Malone and Crowston (1990), are strongly related to the concepts of control theory. Both theories have the components goals and (interdependent) activities. Goals are used to direct the activities and the coordination activities are used to achieve the goals. We therefore argue that visibility is an essential ingredient for effective coordination practices. For instance, coordination practices are impeded without proper levels of visibility over the goals, because the activities cannot be directed towards the goal. For the same reason a lack of visibility over the interdependent activities impedes coordination practices. Information visibility therefore enables coordination practices, leading to the proposition:

[P8] *Information visibility positively impacts coordination practices*

4.5.6 Automation

Automation is the sixth identified issue (72x grounded). One of the mentioned automation issues concerns the lack of backlog status and progress information of the codependent teams in the tracking tool [A1a, A1b, A2a, A2b, A3a]. The lack of such automated information sharing hinders the necessary mitigating activities [A1b, A2b] in the Scrum teams. Automated information sharing of item status and progress falls under Agile Continuous Planning (see subsection 4.2.3), although we found no related work about the lack of such information in Agile settings.

We use related literature in the area of Supply Chain Management to build our last proposition. Supply Chain Management (SCM) is the management of interconnected network, channel and node businesses involved in the provision of product and service packages required by the end customers in a supply chain (Harland, 1996). Workflow in supply chains, which is similar to workflow in Scrum chains, is managed by means of coordination rather than centralization (Mentzer et al., 2001).

Coordination in supply chains requires the visibility of operational (e.g. status and progress) information (Datta & Christopher, 2011; Wei & Wang, 2010). Visibility of such information is enabled by information technology, which is widely used in the area of supply chain management. Given the similarity of supply chain management with a chain of Scrum teams we argue that automation will also benefit visibility in our Scrum chains, which leads to our last proposition:

*[**P9**] Automation of status and progress tracking in the chain positively impacts information visibility.*

4.5.7 Conceptual model

We use the propositions to construct the conceptual model shown in Figure 23. Each proposition connects two issues. For instance the proposition [P9] connects the two issues: (1) '*Automation*' and (2) '*Visibility*'. Each of these two issues is represented as a rectangle in Figure 23. The rectangle, that represents the issue *'Strategic Decision Making'* has a dashed line because the issue was solely grounded in theory (see section 4.5.3) and not identified as an issue through the empirical cases.
An arrow represents the relationship between two issues. The direction of the arrow indicates the dependency between the two issues. For instance the issue '*Automation'* positively impacts the issue '*Visibility'*, as explained in subsection 4.5.6.

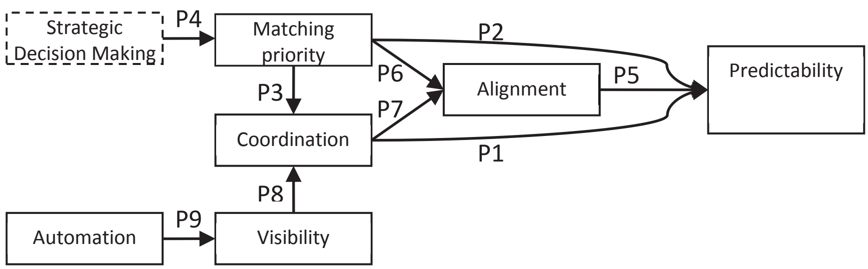

Figure 23, Resulting conceptual model

The conceptual model can serve as starting point for the development of a governance framework to mitigate the identified issues in chains of Scrum teams. We propose a framework that extends the existing Scrum framework with roles that have authority and accountability over the front to back chain. The framework should include a set of coordination and alignment processes for managing front to back feature delivery. But governance is more (A. E. Brown & Grant, 2005):

> *"IT governance is not about what specific decisions are made. That is management. Rather, governance is about 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. Good IT governance draws on corporate governance principles to manage and use IT to achieve corporate performance goals".*

The framework should therefore also include decision making processes to allow matching priority over the front to back chain. For each strategic decision making it should be clear (1) who is the decision making authority over priority setting, (2) who provides input about a decision and (3) how these roles are jointly held accountable (A. E. Brown & Grant, 2005).

The governance framework potentially fulfills the need for structure as mentioned by Talby and Dubinsky (2009), Soundararajan and Arthur (2009) and Batra et al. (2010) (see subsection 4.2.1). Since the governance framework might lead to less Agility in the IT development center, it should comply with the Agile manifesto (Beedle et al., 2013) by having the right mix of plan-based and agile strategies (Batra et al., 2010; Port & Bui, 2009; Soundararajan & Arthur, 2009). The existing models of Kniberg and Ivarsson (2012), Leffingwell (2010) and Scheerer et al. (2014) can be utilized as starting principles for such governance framework.

4.6 Threats to validity

We used case selection criteria and multiple cases to enhance research rigor and external validity. The external validity can be improved further by studying and comparing additional cases in the same area and in other areas, such as IT incident handling chains (Vlietland & van Vliet, 2014c) and service supply chains (Baltacioglu et al., 2007).

For content validity purposes we grounded the interview questions in the dimensions of Galbraith and 3C following the definitions of Sharp and Robinson (2010). We furthermore tested the actual involvement of the standard Scrum Master and Product Owner roles in the Scrum chain and we interviewed supplementary roles where applicable. Bias in the coding and aggregation process was reduced by analyzing the root-causes of the quantitative differences and similarities between the cases (see Table 18). The result of that analysis led to improvements in the coding and aggregation process.

We triangulated the empirical research results by studying archival records, conducting 18 in-depth interviews, conducting verification interviews by phone and using field experts to verify the results.

The inductive nature of the research has several research limitations. First our six issues were identified in three cases, while the preliminary analysis based on two cases identified four issues. Additional case studies might lead to additional issues and extensions or alterations in the conceptual model, since the conceptual model is based on the six identified issues.

Second the qualitative nature of the analysis always results in some levels of analysis bias, even though we used cross-case and quantitative analysis for triangulation. Third the interviews will not collect information that the interviewees consider not necessary to share or consider too sensitive to share.

4.7 Conclusion

We identified six issues in chains of codependent Scrum teams: (1) a lack of coordination in the chain (2) mismatches in backlog priority between teams, (3) alignment issues between teams, (4) a lack of IT chain process automation, (5) unpredictability of delivery to commitment and (6) a lack of information visibility in the chain. The synthesis of these issues with existing theory resulted in nine propositions. These nine propositions have been combined to a conceptual model.

The results show that the application of the Scrum framework in an interdependent application chain can be challenging. Several issues result in unpredictable feature delivery, while the Scrum framework provides little support to mitigate such issues in Scrum chains. Coordination is perceived difficult as coordination is largely allocated to individual coordinators that have little mandate. Priority mismatches result in different Scrum teams developing functionality for different features in parallel. To enable matching priority on operational level the priority needs to be matched at the strategic level at the first place, while priority setting at the strategic level is influenced by emotional, sociological and political factors. The Scrum framework also offers little guidance on the alignment of working processes between Scrum teams. We also found that a lack of automation and visibility of the status and progress of codependent backlogs items impedes collaboration in the chain.

The results imply that the application of Scrum in an interdependent application portfolio needs to be governed. We propose a governance framework to manage chains of Scrum teams in the enterprise that addresses the identified issues, while complying with the Agile manifesto. The framework should include decision making processes for matched priority setting. As a distributed context hinders collaboration between teams, the governance framework should include tailored support for distributed teams, depending on the applied distribution model Sutherland et al. (2007).

Our future research avenue is to empirically test the conceptual model in an existing chain of Scrum teams and to elaborate the conceptual model towards a governance framework that supports the mitigation of the six issues.

A second avenue is to empirically research the strength of the causal relationships between the issues, as well as alternative ways to measure the strength of the issues. A third future research possibility is to analyze additional cases and refine the set of issues and the conceptual model. Another possible future research avenue is to apply the conceptual model in other chains, such as the service supply chain industry.

Chapter 5

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

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.

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.

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).

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

> ***<u>IA:</u>*** *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.*
>
> ***<u>IA:</u>*** *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 codependencies 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.*

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).

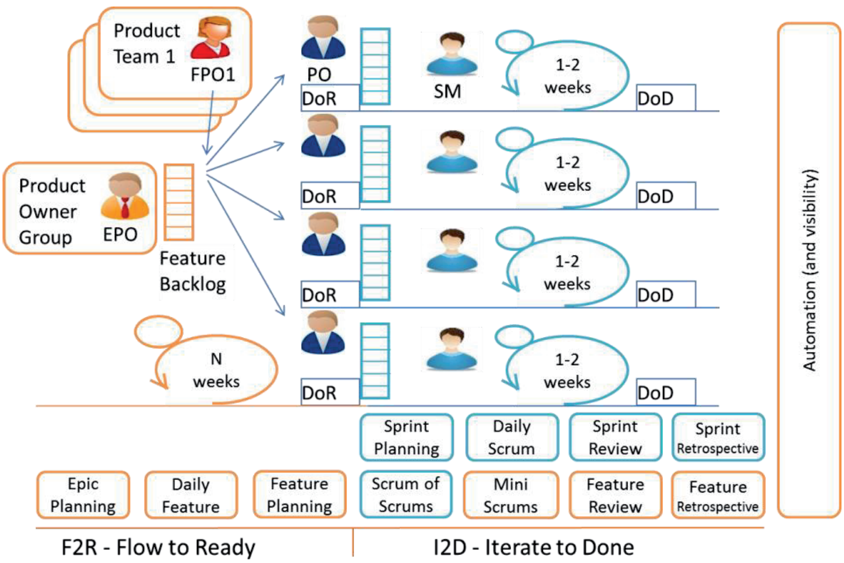

Figure 24: Scrum Value chain Framework (SVF)

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 ϵ 1 – 6
O	Week number of the week that a story is opened	O ϵ Week number
C	Week number of the week that a story is closed	C ϵ Week number
N (C, T)	Amount of stories for team (T), closed in week (C)	N ϵ 0 – m stories
n (C, T)	Story identifier for team (T) closed in a week (C)	n ϵ 1 – N
SD (n)	Amount of days that story (n) is open	DoneDate (n) – TodoDate (n)
ASD (C,T,n)	Average number of open days for all stories by team (T) closed in week (C)	$\sum_{n=1}^{N} SD_{(C,T,n)} / N$
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 ware closed by any team. Scrum team Beta and Gamma are combined in one graph because the two Scrum teams use one combined Product backlog.

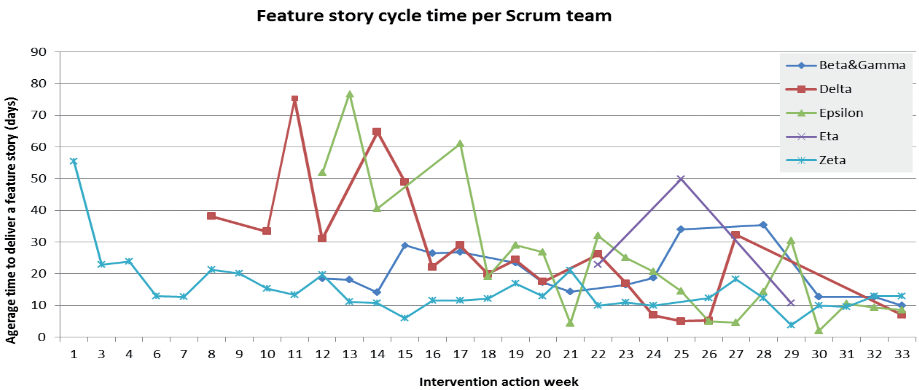

Figure 25: Trend of the Feature story cycle time

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:

> *[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*

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 Scum 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 the Scrum teams.

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*

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			
Focus group	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.

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.

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.

Chapter 6

Improving the Agility of IT Service Networks

Agility in networks of IT service providers helps to swiftly adapt interdependent IT services to changing business needs. In this chapter a set of intervention actions is developed to improve the Agility of these IT service (provider) networks. The intervention actions are based on Agile literature, organizational change theory and empirically confirmed collaboration related factors in Agile IT service networks. The intervention actions are packaged into an Agile 5+1 intervention action framework. The result is an Agile 5+1 framework to improve the Agility in networks of IT service providers.

6.1 Introduction

In today's fast evolving economy business processes of large companies are continuously adapted to survive competition (Takeuchi & Nonaka, 1986). These business processes are to a large extent enabled by information technology. To keep the information technology (IT) operational, the IT staff needs to perform activities. These human activities, combined with the delivered IT is defined as IT services (Beck, 2010; van Bon et al., 2007). IT services are delivered by IT service providers (ISPs). Some ISPs deliver specialized IT services to a single company. Other ISPs deliver to a worldwide customer base (e.g. Google Drive). Many ISPs also deliver IT services to other ISPs. For instance, Microsoft delivering a cloud based platform to an internal ISP of a company for application hosting. As a result ISPs and IT services form networks (Vlietland et al., 2015; Vlietland & van Vliet, 2014c). The IT services in that interdependent (IT service) network are continuously updated, upgraded and renewed by the ISPs, driven by business requirements. The faster changes in an IT service network is achieved, the better the IT service network can follow the changing business environment. To achieve these changes collaboration between staff of different ISPs and different teams in the ISPs is required. That collaboration is clustered in IT software development processes (van Bon et al., 2007). These (workflow) processes flow within and throughout ISPs (J.R Galbraith, 1977).

In order to speed up IT software development of IT services, many internal ISPs transfer to Agile methods (VersionOne, 2013). Agile methods promote continuous change, rather than detailed planning upfront (Beedle et al., 2013). Achieving Agility in IT service networks involves many teams and ISPs, in which staff of the different teams and ISPs needs to interact and collaborate (Vlietland & van Vliet, 2015b). Such interaction and collaboration between staff has been extensively studied in software development contexts (Dorairaj et al., 2012; Paasivaara, Durasiewicz, & Lassenius, 2009; Sharp & Robinson, 2008; Sutherland et al., 2009). To support interaction and collaboration Soundararajan and Arthur (2009) argue that Agile software development practices need to be structured, for which they develop a soft-structured framework. Also other authors developed Agile software development frameworks to structure interaction and collaboration in large scaled Agile settings (Ambler, 2009; Larman & Vodde, 2013; Leffingwell, 2010; Qumer & Henderson-Sellers, 2008; Vlietland & van Vliet, 2015a). The (existing) Agile frameworks do not have an IT service network perspective and hardly cover non-software development processes, such as IT incident handling. For instance existing Agile frameworks are based on iterations, while IT incident handling is a continuous process that does not fit these iterations. A more generic Agile framework for IT service networks, that targets improvements and transcends software development is required. The objective of this research is to develop such framework: Agile 5+1. Agile 5+1 contains intervention actions to alleviate the most common collaboration related issues in IT service networks. The generic nature of Agile 5+1 offers intervention action tailoring to the specific IT service network context. Agile 5+1 is aligned with other scaled Agile frameworks, such as the Scrum Chain Framework of Vlietland et al. (2015). The solid theoretical foundation of Agile 5+1 contributes to the need as mentioned by Freudenberg and Sharp (2010) and Dingsøyr and Moe (2013).

The remainder of this chapter is organized as follows. Section 6.2 discusses the related work. Section 6.3 develops the Agile 5+1 intervention actions. Section 6.4 develops the Agile 5+1 framework. Section 6.5 discusses Agile 5+1, including an operationalization example. Section 6.6 elaborates on the threats to validity. Section 6.7 concludes the study, deduces implications and suggests future research avenues.

6.2 Related work

Three categories of related work are covered. First an overview of scaled Agile frameworks is provided. Next, the essential elements of an IT process are identified, which are applied to Agile 5+1. The section closes with an overview of the collaboration related issues in Agile teams.

6.2.1 Overview of scaled Agile frameworks

In the last decade various scaled Agile framework have been developed. Leffingwell (2010), author of the SAFe framework, promote a three level framework, with the levels, (1) team, (2) program and (3) portfolio. Larman and Vodde (2013) use feature teams and liaison relations (J.R. Galbraith, 1971) with communities of practice (CoP) for exchanging knowledge and coordination between teams. They recently published their Large-Scale Scrum (LeSS) framework based these feature team and liaison principles (Larman & Vodde, 2015). Kniberg and Ivarsson (2012) also utilize the added value of liaison relations, by introducing chapters and guilds. Chapters are groups of people that share expertise. Guilds resemble more free-format special interest groups.

Ambler (2009) describes eight scaling factors to determine scaling complexity: (1) team size, (2) geographical distribution, (3) regulatory compliance, (4) domain complexity, (5) organizational distribution, (6) technical complexity, (7) organizational complexity and (8) enterprise discipline. These scaling factors can be applied to tailor the Disciplined Agile delivery (DAD) process decision framework for scaled Agile applications (Ambler & Lines, 2012). The DAD framework has similarities with the Rational Unified Process framework, both having inception, construction and transition phases (Ambler & Lines, 2012).

Sutherland (2001), the co-author of Scrum (Sutherland & Schwaber, 2013), publishes a 'Scrum at scale' framework via Scrum Inc. The 'Scrum at scale' framework has similarities (e.g. a similar lifecycle) with the Scrum framework (Sutherland & Schwaber, 2013). Scrum framework elements were also applied in the 'Enterprise Scrum' (Greening, 2010), for longer term direction with weekly standups and quarterly sprints. 'Agility path', a scaled Agile framework from the author of Scrum (Schwaber, 2011), aims transitioning the enterprise towards Agility (Schwaber, 2015).

The identified scaled Agile frameworks have several characteristics that do not fit the context of this study. The first misfit is that scaled Agile frameworks lack an IT service network perspective, while many ISPs and teams operate in a network constellations (Vlietland & van Vliet, 2014b, 2014c). Secondly, existing scaled Agile frameworks target software development practices, with fixed iterations. Meanwhile IT incident handling also requires structured Agile collaboration (Vlietland & van Vliet, 2014b, 2014c), while sprints do not fit such continuous process. Thirdly, most scaled Agile frameworks lack described intervention actions (Ambler, 2009; Leffingwell, 2010), while these intervention actions assist staff in achieving Agility (Vlietland et al., 2015). Lastly, Agile frameworks lack a theoretical foundation (Dingsøyr & Moe, 2013; Freudenberg & Sharp, 2010; Jalali & Wohlin, 2012).

6.2.2 Elements of IT processes in IT service networks

In this section the essential elements of an IT process are determined. These elements are targeted by the intervention actions (see section 6.3). Baltacioglu et al. (2007) defined a set of workflow processes for service chains. The defined processes flow through nodes in service networks. A set of processes that exist in the IT service industry is defined by the ISO standardized ITIL v3 framework (van Bon et al., 2007). In each of these IT processes activities are performed that requires collaboration between members in the IT service network.

A process is theorized by Ilgen et al. (2005): input is processed resulting in output. An IT process however has more 'elements'. Journalists describe and report about processes occurring in various forms and events. To describe these processes Journalists use six communication questions (6W), 'Why', 'Who', 'When', 'What', 'Where' and 'With' (Spencer-Thomas, 2012). These six questions have a sufficient abstract nature to define the essence of an IT process and determine the essential elements. Zachman (2002), the author of the generic enterprise architecture framework, uses 6W for segmenting his framework.

The element ‘Who’ refers to the entity that performs workflow activities. Roles and Teams are examples of ‘Who’ elements. Since staff has one or more roles and an ISP has teams, also staff members and ISPs fall under the ‘Who’ element.

The element ‘When’ refers to time; the time to execute workflow activities. Event and Lifecycle are examples of ‘When’ elements.

The element ‘What’ refers to the performed activities in an IT process and the result of the activities. Artifacts and deliverables also fall under the ‘What’ element.

The element ‘With’ refers to the means that support the workflow activities. IT4IT is an example of ‘With’ elements. IT4IT, or ‘IT for IT’ stands for tooling that automates IT processes, which is synonym for Automation in this chapter.

The element ‘Why’ refers to the rationale. In this study the ‘Why’ element refers to the objective of the IT process, which is already covered by the (Agile) priority setting process. The term ‘Where’ refers to geographical spread, which is a context specific operationalization element (see section 6.5). The elements ‘Why’ and ‘Where’ are therefore excluded as separate elements for this study.

6.2.3 Collaboration related factors

Vlietland and van Vliet (2015b) identified six collaboration related factors in chains of codependent Scrum teams that impact Agility: (1) backlog priority between teams, (2) coordination in the chain, (3) alignment between teams, (4) IT chain process automation, (5) information visibility in the chain and (6) delivery predictability. A subsequent study confirms that these factors impact the Agility of a chain of Scrum teams (Vlietland et al., 2015).

In this study these collaboration related issues are considered the most important issues in IT service networks for the following reasons: (1) The study of Vlietland et al. (2015) is executed in a chain of Scrum teams that characterizes a network with nodes and links. (2) Several authors conclude that supply chains and supply networks are similar in terms of collaboration (Cropper, 2008; Wilhelm, 2011). (3) IT incident handling tasks are based on prioritized backlogs (van Bon et al., 2007; Vlietland & van Vliet, 2014c) similar to Agile software development tasks (Sutherland & Schwaber, 2013). (4) Visibility has been also identified in the IT incident handling field and confirmed to be a factor for improved Agility in handling IT incidents (Vlietland & van Vliet, 2014b). For these collaboration related issues in IT service networks intervention actions are developed to enhance IT service network Agility (Vlietland et al., 2015).

6.3 Agile 5+1 intervention actions

To alleviate the collaboration related issues in the IT processes a set of intervention actions is developed. Each intervention action targets the combination of one of the four process elements (Who, When, What and With) and one of the six collaboration related issues (Coordination, Prioritization, Alignment, Visibility, Predictability and Automation). Since Automation is already covered by the 'With' element, no (separate) intervention actions are developed for the collaboration related factor 'Automation' (Vlietland & van Vliet, 2015b). What remains are '4 x 5' intervention actions. Each intervention action is theoretically grounded and discussed in the remainder of this section. The intervention actions are sorted in the order of the collaboration related issues, and labeled with '**[…]**'.

6.3.1 Coordination intervention actions

Coordination is defined by Van de Ven, Delbecq, and Koenig Jr (1976) as: "the process of linking together different parts of an organization to accomplish tasks collectively" (DeSanctis & Jackson, 1994). The organization (in this case the IT service network) needs to coordinate tasks that overarch ISPs and/or teams. Scheerer et al. (2014) compiles a conceptual framework with three types of coordination: (1) mechanistic coordination - coordination by formal plans and rules, (2) organic coordination - coordination by means of mutual adjustment and interactive feedback and (3) cognitive coordination - based on explicit and tacit knowledge sharing between actors, building a shared mental model (Mathieu et al., 2000).

Scheerer et al. (2014) argues that Agile environments require higher levels of organic and cognitive coordination, than mechanistic coordination. An higher level of organic and cognitive coordination better supports Agile organizations in adapting to a changing environment, than with mechanistic coordination. Thus, one intervention action in Agile IT service networks is:

> ***[Pco]*** *Embed organic and cognitive coordination practices between staff members of teams and ISPs.*

The capacity to reflect on past experience is one of the key principles for reflective practices and continuous learning (Holz & Melnik, 2004; Salo & Abrahamsson, 2005). Kolb (1984) abstracts the experiential learning process to the phases (1) concrete experience, (2) reflective observation, (3) abstract conceptualization and (4) active experimentation. A recurring, steady lifecycle with predictable reflection events supports such learning process. Upcoming planned events stimulate staff to reflect on their role and activities, which embeds learning. That argument is supported by Qumer and Henderson-Sellers (2008) arguing that knowledge engineering embedded in the delivery process stimulates learning and improves Agility. Given the above the next intervention action is defined as:

> ***[Pcn]*** *Determine recurring coordination events fitting the delivery lifecycle.*

Cataldo, Bass, Herbsleb, and Bass (2007) define schedules and tasks as artifacts for coordinating work between members in a software development environment. Begel et al. (2009) also identifies schedules and prioritized work items as artifact to coordinate work between teams, along with status of artifacts and context specific artifacts such as bugs and interfaces.
Creating, changing and updating coordination artifacts, requires coordination action (Ilgen et al., 2005). For instance a daily standup is a (coordination) action to coordinate work between team members in an Agile team. During the daily standup three predefined questions are answered to stimulate information sharing between members (Sutherland & Schwaber, 2013). The next intervention action is thus to define and deploy coordination artifacts and necessary actions:

> ***[Pct]*** *Define and deploy coordination artifacts and necessary actions between staff members of teams and ISPs.*

IT service networks typically consists of multiple ISPs with many teams, implying that IT service network members are often physically dispersed (Paasivaara et al., 2012). To coordinate work in these dispersed settings IT4IT is applied, such as continuous planning (Fitzgerald & Stol, 2014). IT4IT enables staff members to instantly coordinate work items with their peers in other teams and ISPs. IT4IT also enables the storage of artifacts that are accessible, independent of location and time. IT4IT furthermore automates workflow processing, such as integration, testing and deployment (Beedle et al., 2013; Humble & Farley, 2010), which otherwise requires manual time consuming coordination activities (Humble, 2010; Vlietland & van Vliet, 2015b). The next intervention action is therefore defined as:

> ***[Pch]*** *Support coordination activities with automation.*

6.3.2 Prioritization intervention actions

Priority issues in Agile settings have been identified by multiple researchers (Begel et al., 2009; Lehto & Rautiainen, 2009; Petersen & Wohlin, 2009; Vlietland & van Vliet, 2015b; Waardenburg & van Vliet, 2012). Also in network organizations priority is often mismatched and ambiguous between interdependent staff, resulting in delayed delivery (Vlietland & van Vliet, 2014a, 2015b). A characteristic of a network organization is confederation – a loose and flexible coalition (Mentzer et al., 2001). To coordinate priority setting in a network organization, a hub can be setup that guides the priority setting process (Webster Jr, 1992). The hub has interdependent staff from teams and ISPs to perform the distributed priority setting process (Kniberg & Ivarsson, 2012). Priority setting processes are driven by decision making processes (Eisenhardt & Zbaracki, 1992; Vlietland & van Vliet, 2015b). For the priority decision process it should be clear (1) who is are the authorized decision makers over priority setting in the network, (2) who provides input about a decision and (3) how these roles are jointly held accountable (A. E. Brown & Grant, 2005; Eisenhardt & Zbaracki, 1992). Matching priority over teams and ISPs stimulates teams to simultaneously work on a common goal, resulting in more delivery predictability and more efficient execution (Lehto & Rautiainen, 2009; Vlietland & van Vliet, 2015b; Waardenburg & van Vliet, 2012). Matching priority also eases coordination of interdependent activities (Vlietland & van Vliet, 2015b). Thus, the next intervention action is defined as:

> ***[Ppo]*** *Assign authorized roles for priority decision making in the network.*

Distributed decision making is a bounded rational process (Eisenhardt & Zbaracki, 1992), which is caused by the cognitive limitations and motivational and emotional factors (Bazerman & Moore, 2009). These cognitive limitations result in a biased perception of reality and biased decision making (Eisenhardt & Zbaracki, 1992). Biased perception can be alleviated by merging the perceptions of different decision makers (Duffy, 1993), resulting in common ground for matched priority setting. Perceptions can be merged with recurring perception sharing and discussion events (Stout, Cannon-Bowers, Salas, & Milanovich, 1999). The need for such perception sharing and shared priority decision making events leads to the following intervention action:

> ***[Ppn]*** *Determine recurring perception merging and prioritization events that fit a lifecycle.*

The prioritizing and decision making process have activities that result in output (Ilgen et al., 2005). In case of prioritization activities the output is the prioritization result, an artifact containing a prioritized list with high level backlog items (Leffingwell, 2010). Also the decision making activities require artifacts, such as a list with objectives, that each includes the 'estimated cost of delay'; which can for instance be calculated with Weighted Shorted Job First (WSJF) (Leffingwell, 2010). The artifacts help focusing the prioritization process (Stout et al., 1999). For instance a prioritized list with ISP objectives that is shared between ISPs during the prioritization event assists the prioritization discussion. One intervention action is therefore to define and establish the artifacts, along with determining the critical prioritization activities. Hence, the next priority intervention actions is:

__[Ppt]__ Define the decision making and prioritization artifacts and necessary actions.

Decision analysis and decision making tools assist the prioritization process (Bazerman & Moore, 2009; Eisenhardt & Zbaracki, 1992). These decision analysis tools include the support of decision making workflow. These tools also support the methods for identifying, representing, and assessing important aspects of a decision. The next intervention action is therefore:

__[Pph]__ Support the decision making and prioritization activities with automated decision analysis and decision making.

6.3.3 Alignment intervention actions

Staff in the IT service network needs to collaborate to deliver interdependent IT services (Vlietland & van Vliet, 2014c). Such collaboration requires a shared mental model between staff (Jonker et al., 2011; Lim & Klein, 2006; Mathieu et al., 2000). Shared mental models are stimulated by grouping people together and stimulate communication and feedback (Stout et al., 1999). A shared mental model is described as cognitive coordination by Scheerer et al. (2014). Next to cognitive coordination, Scheerer et al. (2014) also identifies mechanistic coordination. Mechanistic coordination refers to coordination with formal rules and plans. These rules and plans can be used to develop the shared mental model (Stout et al., 1999), which eases the collaboration. The next intervention action is therefore defined as:

__[Pmt]__ Align the artifacts and workflow over the full IT service network.

Workflow activities are executed by roles. Using similar roles over the IT service network helps collaboration in the network, as members easily recognizes the role of each other (Vlietland & van Vliet, 2014c). Aligned roles also assist in developing the shared mental model (Scheerer et al., 2014; Stout et al., 1999). Aligning the workflow over the service network therefore includes role alignment, leading to the next intervention action:

__[Pmo]__ Use similar roles over the full IT service network.

A shared mental model improves collaboration between staff in a network (Lim & Klein, 2006). Such shared understanding is achieved by information sharing between staff (Vlietland & van Vliet, 2014c). That information sharing is empowered by close collaboration (Scheerer et al., 2014). In many cases staff however cannot work closely together since staff is part of different teams or even different ISPs. These teams and ISPs are often located at different locations, impeding the development of organic alignment (Scheerer et al., 2014). Given the impeded organic alignment, mechanistic coordination is required to share information and build a shared understanding (Scheerer et al., 2014; Stout et al., 1999). Such mechanistic coordination is achieved by setting up events with predefined members (roles). The information that these members share during the events builds the shared mental model (Jonker et al., 2011; Stout et al., 1999). The next alignment intervention action is therefore:

***[Pmn]** Plan and organize alignment events over the full IT service network.*

Multiple ISPs and teams take part in IT service networks. These ISPs and teams are typically distributed and requires IT4IT to collaborate effectively (Vlietland & van Vliet, 2015b). Such automation brings the opportunity of aligning the automated workflow over the IT service network. For instance by having workflow states that are identical over the full IT service network. Such predefined alignment helps developing the shared mental model (Jonker et al., 2011; Stout et al., 1999), enabling members to quickly understand the shared workflow information (Vlietland & van Vliet, 2014c). Our last alignment intervention action is therefore:

***[Pmh]** Align the automated workflow processes in the IT service network.*

6.3.4 Visibility intervention actions

Several researchers concluded that visibility of information improves supply network performance (Bartlett et al., 2007; Vlietland & van Vliet, 2014b). Vlietland and van Vliet (2015b) studied IT service networks confirming information visibility as a factor to improve IT service network Agility. The information that needs to be shared in IT service networks was studied by Vlietland and van Vliet (2014c). They identified three networks; the human, the contractual and the technical network (Vlietland & van Vliet, 2014c). In the human network they identified the following information categories that need to be shared: (1) the network of human resources, (2) contact details of the resources, (3) changes in resources, (4) the process (human activities) and (5) the roles of the resources. Based on these studies the next intervention action is defined as:

***[Pvo]** Share information about the human roles in the full IT service network.*

Vlietland and van Vliet (2014c) conclude that information about the human process needs to be shared in IT service networks. In human (workflow) processing input is transformed to output (Ilgen et al., 2005). In virtual environments workflow processes result in virtual artifacts. These artifacts are stored as information in technological stores. Vlietland and van Vliet (2013) research the impact of information visibility on the performance of workflow processes that run through multiple collaborating teams. In a related research Vlietland and van Vliet (2014c) identify the information that needs to be visible. Many of the identified items refer to artifacts, such as recorded IT incidents, technical designs and service level agreements. That need for visibility of such artifacts leads to the next intervention action:

> ***[Pvt]*** *Share information about the artifacts throughout the full IT service network.*

In product supply networks information visibility is enabled by information technology, for instance for status and position tracking of items (Zhang et al., 2011). Items in the software engineering that are tracked are tasks on a backlog (Sutherland & Schwaber, 2013). The information of these tasks (e.g. status, content, deadline) can be made visible with Continuous planning software (Fitzgerald & Stol, 2014). Also in the IT operations field information visibility is achieved with information technology (Jäntti, 2012a). The next intervention action is therefore:

> ***[Pvh]*** *Support information visibility in the IT service network with information technology.*

Empirical research in the supply chains shows that information sharing helps improving performance (Rashed et al., 2010). Information sharing can be stimulated with (sharing) events. In the Agile software development field Scrum of Scrums and daily standups are examples of events that stimulate information sharing between members (Paasivaara et al., 2012). These information sharing events help building a mutual understanding, such as the achieved results, the status of activities and the impediments (Sutherland & Schwaber, 2013). In case information about these events are visible in the IT service network, staff can subscribe to these events, and contribute in information sharing. Since these events help in sharing knowledge, the last visibility intervention action is:

> ***[Pvn]*** *Share information about the events throughout the full IT service network.*

6.3.5 Predictability intervention actions

Interdependency between IT services implies that a disruption in one IT service disrupts interdependent IT services (Vlietland & van Vliet, 2013). The risk of disruptions increases with the number of interdependencies (Vlietland et al., 2015). For instance in case of a network with 10 interdependent IT services, each delivering with 90% predictability, results in 35% (0.9^{10}) overall (IT service) predictability. One way to increase predictability is reducing the number of interdependencies, leading to the first predictability intervention action:

***[Put]** Remove interdependencies between the IT services in the IT service network.*

The second way is removing the causes of unpredictability in each of the interdependent IT services. Removing the unpredictability requires the responsible members to alleviate unpredictability. Agile teams have self-organizing, group-learning and instability characteristics (Takeuchi & Nonaka, 1986). These characteristics allows Agile teams in making delivering on commitment a team effort (Schwaber, 2004). One of the intervention actions is therefore to allocate clear responsibility for predictable delivery in the team:

***[Puo]** Allocate clear responsibility in the team to deliver with high predictability.*

Agile teams need to have information about the target and realized delivery predictability (Vlietland & van Vliet, 2013). Comparing the realized predictability with the predictability goal, initiates (adapted) action to reach a more predictable IT service (Andrei, 2006; Wiener, 1965). Since information technology enables the workflow process (Fitzgerald & Stol, 2014), the predictability of each IT service (and predictability improvement) can be measured with information technology (Vlietland & van Vliet, 2014b). The next predictability intervention action is therefore:

***[Puh]** Visualize the predictability of each team in the IT service network with automation.*

A workflow process is executed in a timeline with intermediate delivery events and milestones (Mahnic & Zabkar, 2012; Vlietland & van Vliet, 2014b). The intermediate events and milestones result in intermediate artifacts. Predictability of artifact delivery during these intermediate events is an indicator for predictable IT service delivery (Shalloway, Beaver, & Trott, 2009). One of the intervention actions is therefore to monitor the delivery predictability of the intermediate artifacts:

***[Pun]** Monitor the delivery predictability of the artifacts during the intermediate events.*

6.4 Agile 5+1 framework

In this section the intervention actions are packaged into the Agile 5+1 (intervention action) framework. The Agile 5+1 framework has two dimensions. One dimension is based on the four IT process elements (Who, When, What and With). The second dimension of the framework is based on the collaboration related issues (Vlietland & van Vliet, 2015b). The framework has '5' collaboration related factors. '+1' refers to the 'Automation' collaboration factor, that is indirectly covered by the 'With' element (see start section 6.3). The Agile 5+1 framework is shown by Table 25. Each cell in the table contains one intervention action for enhancing IT service network Agility.

Table 25, Intervention action framework for improving IT service network Agility

↓Factor	Who	When	What	With
Coordination	**[Pco]** Embed organic and cognitive coordination practices between staff members of teams and ISPs.	**[Pcn]** Determine recurring coordination events fitting the delivery lifecycle.	**[Pct]** Define and deploy coordination artifacts and necessary actions between staff members of teams and ISPs.	**[Pch]** Support coordination activities with automation.
Prioritization	**[Ppo]** Assign authorized roles for priority decision making in the network.	**[Ppn]** Determine recurring perception merging and prioritization events that fit a lifecycle.	**[Ppt]** Define the decision making and prioritization artifacts and necessary actions.	**[Pph]** Support the decision making and prioritization activities with automated decision analysis and decision making.
Alignment	**[Pmo]** Use the same roles over the full IT service network.	**[Pmn]** Plan and organize alignment events over the full IT service network.	**[Pmt]** Align the artifacts and workflow over the full IT service network.	**[Pmh]** Align the automated workflow processes in the IT service network.
Visibility	**[Pvo]** Share information about the human roles in the full IT service network.	**[Pvn]** Share information about the events throughout the full IT service network.	**[Pvt]** Share information about the artifacts throughout the full IT service network.	**[Pvh]** Support information visibility in the IT service network with information technology.
Predictability	**[Puo]** Allocate clear responsibility that enhances delivery predictability in the team.	**[Pun]** Monitor the delivery predictability of the artifacts during the intermediate events.	**[Put]** Remove interdependencies between the IT services in the IT service network.	**[Puh]** Visualize the predictability of each team in the IT service network with automation.

The Agile 5+1 framework is iconized by the Agile 5+1 model shown in Figure 26. The model illustrates the human members (blue) in the IT service network. The 'orange buttons' illustrate the 5 collaboration related issues. Each button contains four intervention actions to alleviate one collaboration related issue. Automation is illustrated as a surrounding button (+1), with the intervention actions covered by the 'With' element.

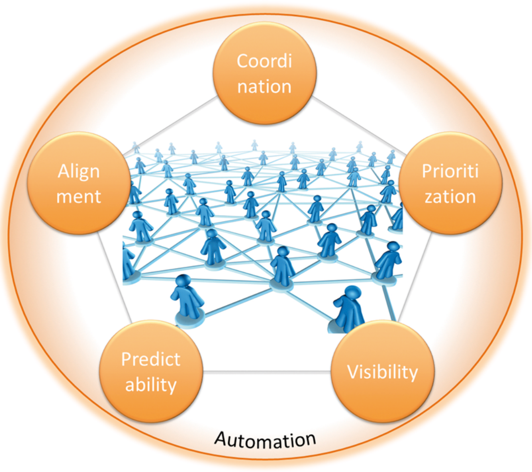

Figure 26, Icon of Agile 5+1

6.5 Discussion

Agile 5+1 targets the full lifecycle in Agile IT service networks; from initiating new ideas to IT incident handling. The intervention actions in the framework need to be tailored to a workflow process in an IT service network. An example of such tailoring is presented by Vlietland et al. (2015). They operationalize Agile 5+1 for a set of codependent Scrum teams, leading to the Scrum Value (chain) Framework. For instance the intervention action Agile 5+1: "*[Pct] Define and deploy coordination artifacts and necessary actions between staff members of teams and ISPs.*", is operationalized by the artifacts *feature backlog* and *feature description* in the Scrum Value (chain) Framework (Vlietland et al., 2015). A second example that fits Agile 5+1 are the visibility based intervention actions in an IT service network in the IT operations field, performed by Vlietland and van Vliet (2014b).

The remainder of this section discusses an IT incident handling scenario in an IT service network, with multiple ISPs delivering interdependent IT services. The scenario describes the IT service network after having multiple Agile 5+1 intervention actions operationalized. In the scenario we refer back to the intervention actions, with the bracketed labels '[…]'.

The scenario starts with a critical IT failure in one of the ISPs, resulting in an avalanche effect of IT failures in the interdependent IT services. The IT failures are automatically detected [Pvh], recorded and placed on the applicable backlogs. Prioritization of the IT failure on the backlogs is supported with decision analysis tooling [Pbh], and manually adjusted by the authorized decision makers [Ppo]. The incident handling work items, which links to the recorded IT failure, are placed on the backlogs with highest priority.

The work items and workflow is visible in all involved teams [Pvn][Pvt], enabled with IT4IT [Pvh]. The teams also have visibility over the involved IT services and underlying IT components [Pvt]. The applicable teams are notified about the new high priority backlog items and instantly start working on the IT failure. The IT4IT [Pvh][Pch], supports active online discussion and root-cause analysis [Pco]. The roles are aligned over the full network [Pmo], minimizing the need for information sharing about the ‘way of working’, which avoids misunderstanding. The roles (members) are easily accessible via the IT4IT [Pmh][Pvo]. Once the root-cause is identified the failing IT service is fixed and brought back online, which is notified by the other teams. The work items are closed in the IT4IT and the IT failure is resolved.

6.6 Threats to validity

The (inductive) Agile 5+1 framework has been largely based on the six collaboration related issues, which were identified in a multiple case study in the software development environment (Vlietland & van Vliet, 2015b). Only visibility as Agile improvement factor was also identified in the IT operation environment (Vlietland, 2011; Vlietland & van Vliet, 2013, 2014c). Yet, whether the six issues exist in other IT service network environments remains hypothetical.

Furthermore, the impact of the collaboration related factors have been tested in a single case study with codependent Scrum teams in the software development area (Vlietland et al., 2015). That case study did however not test the effect of the individual intervention actions (Vlietland & van Vliet, 2015b). Only visibility as Agile improvement factor was also confirmed in an improvement case study in the IT operation environment (Vlietland & van Vliet, 2014b).

Agile 5+1 furthermore has an abstract nature. The intervention actions need to be tailored to the specific context as discussed in the previous section. Such tailoring process involves many organizational contextual factors. These factors need to be identified and linked towards applicable organizational design theory (Daft, 2009; J.R Galbraith, 1977), while taking the Agile principles and objectives into account.

6.7 Conclusion

The objective was to develop a set of intervention actions that improves the Agility of IT service networks. To that end results of existing literature were abstracted, including the identified collaboration issues in IT service networks. That literature was used to develop the set of intervention actions, which was subsequently packaged in an Agile 5+1 (intervention action) framework with two dimensions. The intervention actions aim improving Agility in IT service networks.

The Scrum Value (chain) Framework (Vlietland et al., 2015) and the visibility interventions (Vlietland & van Vliet, 2014b) can be seen as incarnations of Agile 5+1. The Scrum Value (chain) Framework incarnation deploys (tailored) intervention actions that target a codependent set of Scrum teams. Even though the intervention actions of the Scrum Value (chain) Framework have been developed prior to the intervention actions in Agile 5+1, both target the same issues. Also the visibility intervention actions for improving IT incident handling performance (Vlietland & van Vliet, 2014b) fit Agile 5+1.

A limitation is the limited number of IT service networks in which the collaboration related issues have been identified. One future research opportunity is therefore to perform confirmatory case studies in various IT service network settings, to validate the existence of the collaboration related issues.

Another limitation is the hypothetical nature of Agile 5+1. Even though Agile 5+1 have been incarnated with the Scrum Chain Framework and with the visibility interventions in IT operation, Agile 5+1 itself has not been tested. A future research avenue is therefore to test Agile 5+1 and the individual intervention actions in various IT service networks. These results might enhance Agile 5+1 with a sequenced set of intervention actions, depending on the contextual conditions.

A third limitation is the abstract nature of Agile 5+1. A future research opportunity is therefore to develop tailoring guidelines based on applicable organizational design theory (Daft, 2009; J.R Galbraith, 1977).

A fourth limitation of Agile 5+1 is the improvement focus on process, roles and deliverables. These ‘hard’ aspects, of operationalizing Agile 5+1 typically require a shift in mindset and behavior. These investments can be significant depending on the existing organizational culture. A future research opportunity might therefore be to expand Agile 5+1 with guidelines for changing mindset and behavioral aspects.

Chapter 7

Conclusion

7.1 Looking back

The objective of this dissertation was finding ways to improve the Agility of IT service networks, leading to the main research question:

> **Main research question**: How to 'improve' the 'Agility' of 'IT service networks'?

To answer the research question IT incident handling and software development processes in IT service networks have been studied, in the telecom and the financial industry. The research included a number of case studies ranging from 2010 to 2015. Given the existing body of related work some of the case studies are inductive in nature. With the case studies and surveys new theory has been induced and tested. The dissertation started by studying and developing the visibility and IT service network performance concepts. The theory was subsequently elaborated with additional concepts, while moving to IT service network Agility. With the concepts an Agile 5+1 framework has been developed, guiding Agility improvements in IT service networks. The effectiveness of Agile 5+1 has been (indirectly) confirmed with two incarnations.

7.2 Revisiting the research questions

In the introduction chapter the research is split into individual research questions. In this section the individual research questions of chapter 1 are revisited, with the results of the studies in chapter 2, 3, 4, 5 and 6.

The first research question aims to enhance understanding of the impact of the macro-level, meso-level and micro-level links onto the Agility of IT service networks. Macro-level links represent interdependencies between the ISPs, meso-level links represent interdependencies between teams and micro-level links represent interdependencies between staff. The first research question is defined as:

> **RQ 1**: What IT service network interdependencies affect IT delivery in IT service networks?

The answer to this question is given in chapter 2, by conducting an inductive case study in a network of nine interdependent IT service providers. In the case study three types of networks with interdependencies are identified: the human network, the contractual network and the technical network. The interdependencies in these networks and between these networks are to a large extent based on the information that is needed by staff to deliver the IT services. The needed information is distributed over multiple technical and human stores, creating dependencies between staff. To access the information stores, the staff needs an overview over all stores. Since information is partially stored in human stores, staff needs overview over the human network. The answer to the research question includes a conceptual model with the three network types.

The next question is which information needs to be visible in each of the three identified networks, resulting in the following research question:

RQ 2: What information needs to be visible for IT delivery in IT service networks?

The second research question is also answered in chapter 2, by providing the needed information categories of each network type. Regarding the human network, information about human resources, contact details, resource changes, performed processes and human roles needs to be shared. With regard to the contractual network, information about the events in the IT service, network of IT services, IT service levels and changes in the supplier services needs to be shared. As the IT incident handling process is studied, the identified events are IT incidents. Regarding the technical network, information about the technical system process, critical IT system changes, capacity changes, IT system network and IT system design needs to be shared. The study identifies a lack of almost all information categories in the studied IT service network. The study shows that most of the information categories need to be shared beyond the first 'tier' in the IT service network. The tier level (first, second, etc.) indicates the minimum number of edges that information has to travel between two nodes (Caridi et al., 2010a). A first tier relationship indicates a direct interdependency (edge) between two IT providers (nodes).

With a subset of the identified information in the contractual network the hypothesized impact of visibility on IT service network performance was tested, based on the research question:

RQ 3: To which extent does visibility of information improve the performance of IT service networks?

The answer to that question is provided in chapter 3, by a confirmatory case study in an IT service network. The results of the case study show that IT service network performance can be improved by enhancing information visibility with visibility-based interventions. In the end state, the team achieves high levels of incident handling performance.

The interdependencies require teams between and within ISPs to collaborate while handling IT failures and achieving IT changes. Question was which collaboration related factors impact IT service network performance. In order to find these collaboration related factors the following research question was defined:

RQ 4: What collaboration related factors impede the Agility of IT service networks?

The answer to research question 4 is provided in chapter 4, by defining six collaboration related factors: (1) coordination, (2) priority, (3) alignment, (4), automation, (5) predictability and (6) visibility. The factors are based on the six collaboration related issues that were empirically identified in multiple IT service networks. The six identified issues are: (1) a lack of coordination between nodes (2) mismatched priority between nodes, (3) alignment issues, (4) a lack of IT process automation, (5) unpredictable delivery and (6) a lack of information visibility. The synthesis of the factors with existing theory resulted in nine propositions. These nine propositions were subsequently combined to a conceptual model.

With the results the question emerged whether the collaboration related factors impact IT service network Agility. In order to test the impact the following research question was defined:

RQ 5: To which extent does alleviating collaboration issues improve the Agility of IT service networks?

The question is answered with the case study in chapter 5. In the case study the Agility of an IT service network is improved by alleviated collaboration related issues. For the improvement a set of intervention actions is developed, based on the collaboration related factors. The intervention actions are subsequently deployed to mitigate the collaboration issues in a codependent set of Scrum teams, and to validate the effectiveness of the intervention actions. While the intervention actions are deployed the cycle time of new features is reduced from 29 days to 10 days. Participants in focus groups confirmed the causality between the observed improvements and intervention actions. The validated intervention actions are packaged in a governance framework for codependent sets of Scrum teams, the Scrum Value (chain) Framework.

Based on the results of chapter 2-5 the main research question is answered in chapter 6, by developing the Agile 5+1 (intervention action) framework for improving IT service network Agility. Based on the related work 20 intervention actions have been developed to improve the Agility. These intervention actions have been packaged in the Agile 5+1 intervention action framework with two dimensions. One dimension is based on four elements of an IT process (Who, When, What and With). The second dimension is based on the collaboration related factors. The Agile 5+1 framework is iconized by the Agile 5+1 model. The model and the framework together are abbreviated as 'Agile 5+1'.

Two incarnations of Agile 5+1 have been (indirectly) tested, by developing intervention actions based on the collaboration related factors. The most elaborate incarnation was tested in chapter 5 in a software development context, by developing intervention actions packaged in the Scrum Value (chain) Framework (Vlietland et al., 2015). The incarnation in the IT incident handling context was validated in chapter 3, with the development of the visibility-based interventions, deployed in an IT service network (Vlietland & van Vliet, 2014b).

7.3 Contribution and implications

The results of the experiments in chapter 3 and 5 confirm that the intervention actions based on the identified collaboration related factors can improve the Agility of IT service networks. The two incarnations support the validity of the hypothetical Agile 5+1, while the abstract nature allows appliance in many different IT service network contexts. Even though the nature of Agile 5+1 is abstract, the practical set of questions used by Journalists (Spencer-Thomas, 2012) allows straightforward tailoring of the intervention actions. For instance 'Who' can be straightforwardly translated to roles in an IT service network and 'What' to workflow activities and deliverables.

The studies advance the field of Agile software engineering in different ways. In the first place by identifying three different networks within and between IT service providers, which need to be analyzed to understand the interdependencies between IT service providers. These interdependencies affect collaboration between and within IT service providers. Studying the interdependencies and collaboration advances the knowledge of network based IT ecosystems (Jansen & Cusumano, 2013; Jansen, Finkelstein, & Brinkkemper, 2009; Riedl, Böhmann, Rosemann, & Krcmar, 2009), that consist of staff, teams, ISPs and information technology.

Secondly, the studies enhance knowledge of Agility improvements in network constellations, by identifying the collaboration related issues that impede Agility in network settings. These identified collaboration related issues assist in developing ways to improve the collaboration in network settings, such as in complex distributed outsourcing contexts (Paasivaara et al., 2012; Ralph, Shportun, & Bloomberg, 2013; Sutherland et al., 2009).

A third implication is the development of intervention theories to improve the Agility in IT service networks. By developing theory and intervention actions, and subsequently testing these intervention actions in the IT industry, help developing our understanding about organizational change and complexity theories in IT eco-systems (Cummings & Worley, 2014; Jansen et al., 2009; Stacey, 1995; Stelzer & Mellis, 1998).

7.4 Limitations and future research

The dissertation has various limitations and opportunities for future research. One limitation is the inductive nature of the studies in the dissertation. The main research question was answered with the development of the Agile 5+1 (intervention action) framework. The Agile 5+1 framework has been based on the collaboration related factors, which were identified and confirmed by case studies in the software development environment (Vlietland & van Vliet, 2015b). Though the factors were tested and confirmed by only one case study and the effect of the individual intervention actions was not tested (Vlietland & van Vliet, 2015b). Even though the 'visibility' improvement factor was identified and confirmed in the IT operation environment (Vlietland, 2011; Vlietland & van Vliet, 2013, 2014c), the other factors have not been researched in the IT operations context. Hence, Agile 5+1 and the impact of the factors onto Agility of IT service networks has not been (fully) empirically confirmed and remains hypothetical. A future research avenue is therefore (1) to study more IT service networks and validate the identified factors and (2) to test Agile 5+1 and the individual intervention actions in various IT service networks. These results might lead to an understanding of the intervention action dependencies, based on the characteristics of the IT service network. Such understanding might help predicting the feasibility of the intervention actions, similar to the IT service maturity levels of Niessink and van Vliet (1998). A related future avenue is conducting experiments in IT service networks with 'automation' (Continuous Delivery) deployed over multiple ISPs. Automated software development processes (Humble & Farley, 2010), probably impact other collaboration related issues as modeled by Vlietland and van Vliet (2015b).

Another limitation is the abstract nature of the Agile 5+1 framework, while targeting process, roles and deliverables. Next to these 'hard' aspects, the Agile 5+1 intervention actions typically require a shift in mindset and behavior. To achieve that shift the intervention actions need to be tailored to these specific contextual factors. To enable such tailoring the contextual factors need to be identified and linked to the applicable organizational design theory (Daft, 2009; J.R Galbraith, 1977), while taking the Agile principles and objectives into account. Identifying the factors and tailoring the intervention actions can be a significant investment, depending on the existing organizational culture. Currently Agile 5+1 does not provide any guidance in tailoring the intervention actions, leaving tailoring to the interpretation of the user. An opportunity for research is therefore to develop a number of tailoring principles for the Agile 5+1 intervention actions. These tailoring principles guide change agents in developing intervention actions for the specific IT service network context. These tailoring principles can for instance take into account the organization and social culture, perceived distance between staff and governmental, political and architectural constraints (Ambler, 2009).

A third limitation is the relationship between performance and Agility. At the start of the dissertation the dependent variable was IT service network performance. Performance was defined with objective (supply chain) performance indicators. After answering RQ3, IT service network performance was redefined as IT service network Agility, based on similar (objective) indicators. Yet, such definition of Agile is rather narrow, compared to other definitions that include awareness, flexibility, productivity and adaptability (Plummer & McCoy, 2006). Moreover, the definition of Agility in this dissertation is based on contractual (supply chain) parameters (Vlietland, 2011), while the Agile manifesto advocates collaboration over contracts. The definition of Agile seems therefore misaligned with the Agile manifesto. A future research avenue is developing a more comprehensive definition for IT service network Agility. In that comprehensive definition social network theory (Freeman, 1979). might be useful to explain the impact of the three identified networks (Vlietland & van Vliet, 2014c), on IT service network Agility.

The studies in for this dissertation have been carried out in IT service networks in the telecom and financial industry. Applying future research in other industries might benefit these industries, while enhancing the understanding of Agility improvements in IT service networks. Future research can also be in the direction of adapting Agile 5+1 to a model that assists enterprises to become a ‘responsive enterprise’. The responsive enterprise is a philosophy explaining how companies can adapt, learn and respond to our evolving world (ResponsiveOrg, 2014). Such adaption of Agile 5+1 transcends the scope to service networks (Viswanadham et al., 2005), possibly benefiting the much broader service network field.

Chapter 8

Summary

Traditionally working processes are to a large extent hierarchically organized. Such hierarchical setup is aimed on efficiency and stability. That aim on efficiency and stability fits a world that develops itself with limited speed. Nowadays the world develops itself in a much faster pace. That fast pace makes companies producing via hierarchical structures, less effective in the response to market changes. In the last decennia a development unfolds in which companies organize their work via network structures. These business networks are to a large extent enabled by modern information technology. Examples of such technology are internet and mobile telephony. With that technology collaboration structures can be setup in a very short time. Technology is therefore an essential part of our modern network oriented society.

Such technology is delivered by many IT providers, as IT services. Each IT service provider delivers one or more of these IT services. Multiple IT service providers then combine the IT services to composed interdependent IT services, which are delivered to a business network. Different IT service providers consequently form networks of IT services. Such network of IT services results in the term 'IT service network'.

Given the fast developing markets the IT services in that network are subject to many changes. To realize these IT service changes faster, initiatives emerged in the beginning of this century. These initiatives aimed to let organizations operate with increased flexibility. The initiatives are known under the 'Agile' umbrella, meaning 'responsive', 'flexible'. Remarkable successes have been achieved with 'Agile execution'. Yet, these successes are based on small scaled applications. The Agile execution of organizations at a large scale brings serious challenges due to the many dependencies in networks of IT services.

This dissertation is the result of multiple studies aiming to answer the question how the Agility of networks of interdependent IT services can be improved. The first study (see chapter 2) aimed to identify the codependencies between the parties that deliver these IT services. These parties are the IT service providers and the teams in each of the IT service providers. This study revealed three types of networks with codependencies: (1) the human network, (2) the contractual network and (3) the technical network. The study shows that the dependencies in these networks are to a large extent based on the information that staff shares. Staff needs that information to deliver their IT services. The follow up research (see chapter 2) aimed to identify the information that needs to be 'visible' for the staff. This research resulted in an overview of information for each of the three types of codependencies. The case study revealed a lack of almost all information that is needed by the staff.

A logical related question was whether increasing the 'visibility' of information within and between the teams that deliver the IT services improves Agility. The resulting research (see chapter 3) was executed in a large financial institute. The study shows improved Agility as a result of the increased information visibility. Agility in the study was defined as the duration of (IT incident handing) tasks that teams complete.

The follow up research question that emerged was which other factors influence the Agility of IT service networks. To answer that question a case study was performed about the issues that staff in IT service networks experience (see chapter 4) to deliver Agile IT services. From these issues six factors were identified, including the visibility of information.

The subsequent case study tested whether influencing these factors improves the Agility in an IT service network. The case study was performed at a financial institute (see chapter 5), in which we deployed intervention actions for that context. The study confirmed the relationship between the factors and the IT service network Agility; the Agility increased by a significant reduction of the delivery time of new software.

With the results of the studies (chapter 2-5) the Agile 5+1 framework was developed. The Agile 5+1 framework consists of 20 generic intervention actions to improve the Agility of IT service networks.

Still, a lot is to be studied in the field of IT service networks. A limited number of cases have been studied which limits the generalizability of the results. This opens the possibilities for future research in the field. A natural avenue for future research is the empirical confirmatory study of the Agile 5+1 framework. Naturally these studies will take place in the IT industry. However, also outside the IT industry I see possibilities to improve the flexibility between and within collaborating parties, based on the results of this dissertation.

Chapter 9

Samenvatting

Traditioneel zijn werkprocessen in hoge mate hiërarchisch georganiseerd. Zo'n hiërarchische inrichting is gericht op efficiency en stabiliteit. Deze gerichtheid op efficiency en stabiliteit paste bij een wereld die zich ontwikkelde met beperkte snelheid. Tegenwoordig ontwikkelt de wereld zich in een veel sneller tempo. Dit maakt dat veel bedrijven die langs hiërarchische structuren produceren niet effectief genoeg kunnen inspelen op veranderingen in hun markt. De laatste decennia is dan ook een ontwikkeling gaande waarbij bedrijven het werk in toenemende mate in netwerkenstructuren organiseren. Deze netwerken met bedrijven zijn voor een belangrijk deel mogelijk gemaakt door moderne informatie technologische hulpmiddelen. Voorbeelden van deze hulpmiddelen zijn het internet en mobiele telefonie. Met deze hulpmiddelen is het mogelijk om in zeer korte tijd samenwerkingsverbanden te realiseren die samen waarde creëren. Technologie is dus een essentieel onderdeel geworden van onze moderne netwerkgeoriënteerde maatschappij.

Deze technologie wordt door vele IT leveranciers geleverd als IT diensten, in het Engels IT services genoemd. Elke IT leverancier levert een of meerdere IT services. Deze IT services combineren IT leveranciers dan tot samengestelde IT services, die aan een netwerk met bedrijven worden geleverd. Voor een IT service aan een klant zijn dus verschillende IT leveranciers nodig die samen netwerken van IT services vormen. Zo ontstaat de term 'IT service netwerk', netwerken van IT diensten.

Wegens de snel ontwikkelende markten zijn IT services binnen netwerken aan allerlei veranderingen onderhevig. Om veranderingen in IT services sneller te kunnen realiseren, zijn rond de eeuwwisseling nieuwe initiatieven ontplooid. Deze initiatieven hadden tot doel om organisaties flexibeler te laten opereren. Deze initiatieven zijn geschaard onder de term 'Agile', dat zoveel betekent als 'lenig', 'flexibel'. Met het 'Agile werken' zijn al mooie successen behaald, maar de successen zijn steeds gebaseerd op kleinschalige toepassingen. Het op grote schaal Agile laten werken van organisaties blijkt een grote uitdaging door de vele afhankelijkheden in het netwerk van IT services.

Dit proefschrift is het resultaat van meerdere onderzoeken met als doel om antwoord te vinden op de vraag hoe je de Agility van netwerken van deze onderling afhankelijke IT services verbetert. Een eerste onderzoek (zie hoofdstuk 2) had tot doel om de afhankelijkheden tussen de partijen te identificeren die de IT services leveren. Deze partijen zijn de IT leveranciers zelf en de teams in elk van deze IT leveranciers. Uit dat onderzoek blijkt dat er drie typen netwerken met afhankelijkheden zijn: (1) het intermenselijke netwerk, (2) het contractuele netwerk en (3) het technische netwerk. Het blijkt dat de afhankelijkheden in deze netwerken voor een belangrijk deel zijn gebaseerd op de informatie die medewerkers delen. Deze informatie hebben de medewerkers nodig om de IT services in het netwerk te leveren.

Het vervolgonderzoek (zie hoofdstuk 2) richtte zich daarom op de informatie die ‘zichtbaar’ moet zijn voor deze medewerkers. Dit onderzoek resulteerde in een overzicht van informatie voor elk van de drie typen afhankelijkheden. In de onderzochte case bleek een gebrek aan bijna alle geïdentificeerde informatie.

Een logische gerelateerde vraag was of het verhogen van de informatiezichtbaarheid binnen en tussen de teams die de IT services leveren bijdraagt aan de Agility. Het hieruit voortvloeiende onderzoek (zie hoofdstuk 3) is uitgevoerd in een grote financiele instelling. Dat onderzoek laat een verbeterde Agility zien als gevolg van een verhoogde zichtbaarheid van informatie. Agility is in het onderzoek gedefinieerd als de tijdsduur van (IT incident) taken die teams afhandelen.

De onderzoeksvraag die daaruit ontstond was welke andere factoren de Agility in IT service netwerken beïnvloeden. Om deze vraag te beantwoorden hebben we onderzocht welke problemen medewerkers in IT service netwerken ervaren (zie hoofdstuk 4). Uit deze problemen zijn zes factoren geïdentificeerd, waaronder de zichtbaarheid van informatie.

Vervolgens is onderzocht of het beïnvloeden van deze factoren leidt tot een verbeterde Agility in een IT service netwerk. Daarvoor is een case study uitgevoerd bij een financiele instelling (zie hoofdstuk 5), waarin interventie acties voor die context werden uitgevoerd. In dat onderzoek werd het veronderstelde verband tussen de factoren en de Agility van het IT service netwerk bevestigd. De Agility werd verhoogd door een significante verkorting van de levertijd van nieuwe software.

Met de resultaten van de onderzoeken (hoofdstuk 2-5) is tenslotte een Agile 5+1 raamwerk ontwikkeld. Dit Agile 5+1 raamwerk bevat 20 generieke interventie acties om de Agility van IT service netwerken te verbeteren.

Er is nog veel te onderzoeken op het gebied van IT service netwerken. Ten eerste is er maar een beperkt aantal cases onderzocht waardoor de mogelijkheid van generalisatie van de resultaten beperkt is. Dit biedt mogelijkheden voor toekomstig onderzoek in het vakgebied. Een voor de hand liggende richting van toekomstig onderzoek is het empirisch toetsen van het Agile 5+1 raamwerk. Logischerwijs vinden deze empirische onderzoeken plaats in de IT industrie. Maar ook buiten de IT industrie zie ik mogelijkheden om de flexibiliteit tussen en binnen samenwerkende partijen te verbeteren, op basis van de resultaten in dit proefschrift.

Chapter 10

Curriculum vitae

Personal data

Full name:	Jan Vlietland
Date of birth:	November, 16th, 1971
Place of birth:	Goedereede
Father of:	Roxanne Vlietland and Jennifer Vlietland

Education

2011 – 2015	PhD at VU University, Computer Science
2009 – 2010	MSc at Open University, Management Science, Organizational change
1995 – 1998	BSc at Hogeschool West Brabant, HTS technical computer science
1988 – 1992	Zadkine college Rotterdam Zuid, MTS elektronica
1984 – 1988	W.C. van As Middelharnis, LTS elektro

Working experience

2008 – present	Managing partner Search4Solutions B.V.
2013 – present	Strategic Agile consultant, top 500 companies
2008 – 2013	Global Implementation manager, ING, Asia, Europe, USA region
2005 – 2008	Project manager, Achmea, The Netherlands
2002 – 2005	IT consultant, Achmea, The Netherlands
1992 – 2002	Software developer, Origin, ICT Automatisering

Other endeavors

1983 – present	Software development
2013 – present	Micro credits company in Ghana
1998 – 2006	Bike rally raids on international level

Chapter 11

Bibliography

Ågerfalk, P. J., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B., & Conchúir, E. O. (2005). *A framework for considering opportunities and threats in distributed software development.* Proceedings International Workshop on Distributed Software Development, 47-61.

Ahlstrom, P., & Nordin, F. (2006). Problems of establishing service supply relationships: Evidence from a high-tech manufacturing company. *Journal of Purchasing & Supply Management, 12*, 75-89.

Akbar, R., Hassan, M. F., & Abdullah, A. (2011). A Review of Prominent Work on Agile Processes Software Process Improvement and Process Tailoring Practices *Software Engineering and Computer Systems* (pp. 571-585): Springer.

Alleman, G. B., Henderson, M., & Seggelke, R. (2003). *Making agile development work in a government contracting environment-measuring velocity with earned value.* Proceedings of the Agile Development Conference, 2003. ADC 2003, 114-119.

Allen, S., & Chandrashekar, A. (2000). Outsourcing services: the contract is just the beginning. *Business Horizons, 43*(2), 25-34.

Ambler, S. (2009). The agile scaling model (ASM): adapting agile methods for complex environments. Retrieved from www.ibm.com website:

Ambler, S. (2012, Summer 2012). Agility at scale Survey Results, from http://www.ambysoft.com/surveys/stateOfITUnion201209.html

Ambler, S., & Lines, M. (2012). *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*: IBM Press.

Andrei, N. (2006). Modern Control Theory. *Studies in Informatics and Control, 15*(1), 51.

Angerhofer, B. J., & Angelides, M. C. (2006). A model and a performance measurement system for collaborative supply chains. *Decision Support Systems, 42*(1), 283-301.

Aranda, J., & Venolia, G. (2009). *The secret life of bugs: Going past the errors and omissions in software repositories.* Proceedings of the 31st International Conference on Software Engineering, 298-308.

Baltacioglu, T., Ada, M., Kaplan, M., & Yurt, O. (2007). A New Framework for Service Supply Chains. *Service Industries Journal, 27*(2), 105-124.

Banbury, S., Helman, S., Spearpoint, J., & Tremblay, S. (2010). *Cracking the bullwhip: team collaboration and performance within a simulated supply chain.* Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 1620-1624.

Bannerman, P. L., Hossain, E., & Jeffery, R. (2012). *Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?* Proceedings 45th Hawaii International Conference on System Science (HICSS), 2012, 5309-5318.

Bardhan, I. R., Demirkan, H., Kannan, P., Kauffman, R. J., & Sougstad, R. (2010). An interdisciplinary perspective on IT services management and service science. *Journal of Management Information Systems, 26*(4), 13-64.

Barratt, M., & Oke, A. (2007). Antecedents of supply chain visibility in retail supply chains: A resource-based theory perspective. *Journal of Operations Management, 25*(6), 1217-1233.

Bartlett, P. A., Julien, D. M., & Baines, T. S. (2007). Improving supply chain performance through improved visibility. *International Journal of Logistics Management, The, 18*(2), 294-313.

Bartolini, C., Sallé, M., & Trastour, D. (2006). *IT service management driven by business objectives An application to incident management.* IEEE/IFIP Network Operations and Management Symposium, Vancouver, Canada, 45-55.

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the AIS, 2*, 2-32. Retrieved from

Baskerville, R. L., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Technology, 11*(3), 235-246.

Basole, R., & Rouse, W. (2008). Complexity of service value networks: conceptualization and empirical investigation. *IBM Systems Journal, 47*(1), 53-70.

Batra, D., Xia, W., VanderMeer, D., & Dutta, K. (2010). Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry. *Communications of the Association for Information Systems, 27*(1 (article 21)).

Bazerman, M., & Moore, D. A. (2009). *Judgment in managerial decision making*: John Wiley.

Beamon, B. (1999). Measuring supply chain performance. *International journal of operations and production management, 19*, 275-292.

Beccalli, E. (2007). Does IT investment improve bank performance? Evidence from Europe. *Journal of Banking & Finance, 31*(7), 2205-2230.

Beck, R. (2010). The nature of IT services from a management and IS research point of view *Proceedings of SIGSVC Workshop* (Vol. 10): Sprouts.

Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., A., H., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Schwaber, K., Sutherland, J., & Thomas, D. (2013). Principles behind the Agile Manifesto, from http://agilemanifesto.org/principles.html

Beel, J., & Gipp, B. (2009). *Google Scholar's ranking algorithm: the impact of citation counts (an empirical study).* Research Challenges in Information Science, 2009. RCIS 2009. Third International Conference on, 439-446.

Begel, A., & DeLine, R. (2009). *Codebook: Social networking over code.* Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on, 263-266.

Begel, A., Khoo, Y. P., & Zimmermann, T. (2010). *Codebook: discovering and exploiting relationships in software repositories.* Software Engineering, 2010 ACM/IEEE 32nd International Conference on, 125-134.

Begel, A., Nagappan, N., Poile, C., & Layman, L. (2009). *Coordination in large-scale software teams.* Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 1-7.

Bentley, C., Jones, E., Atkinson, D., & Ferguson, A. (2009). *Managing successful projects with PRINCE2*: TSO (The Stationary Office).

Bhattacharya, R., & Bandyopadhyay, S. (2011). A review of the causes of bullwhip effect in a supply chain. *The International Journal of Advanced Manufacturing Technology*, 1-17.

Birks, M., & Mills, J. (2011). *Grounded theory: A practical guide*. Thousands Oaks, California: SAGE Publications Ltd.

Blumenberg, S., Wagner, H., & Beimborn, D. (2009). Knowledge transfer processes in IT outsourcing relationships and their impact on shared knowledge and outsourcing performance. *International Journal of Information Management, 29*(2009), 342-352.

Bolstorff, P. (2003). Measuring the impact of supply chain performance. *Logistics Today, 12*.

Boonstra, A. (2003). Structure and analysis of IS decision-making processes. *European Journal of Information Systems, 12*(3), 195-209.

Bosch, J., & Bosch-Sijtsema, P. (2010). Coordination between global agile teams: From process to architecture *Agility Across Time and Space* (pp. 217-233): Springer.

Bosch, J., & Bosch-Sijtsema, P. M. (2011). Introducing agile customer-centered development in a legacy software product line. *Software: Practice and Experience, 41*(8), 871-882.

Brown, A. E., & Grant, G. G. (2005). Framing the Frameworks: a Review of IT Governance Research. *Communications of the Association for Information Systems, 15*, 696-712.

Brown, A. W., Ambler, S., & Royce, W. (2013). *Agility at scale: economic governance, measured improvement, and disciplined delivery.* Proceedings of the 2013 International Conference on Software Engineering, 873-881.

Cai, J., Liu, X., Xiao, Z., & Liu, J. (2010). Improving supply chain performance management. *Decision Support Systems, 2009*(46), 512-521.

Calvert, R. (1995). The rational choice theory of social institutions: Cooperation, coordination, and communication. *Modern political economy: Old topics, new directions*, 216-268.

Candea, G., Bucur, S., & Zamfir, C. (2010). *Automated software testing as a service.* Proceedings of the 1st ACM symposium on Cloud computing, 155-160.

Cao, M., & Zhang, Q. (2011). Supply chain collaboration: Impact on collaborative advantage and firm performance. *Journal of Operations Management, 29*(3), 163-180.

Caridi, M., Crippa, L., Perego, A., Saianesi, A., & Turmino, A. (2010a). Do virtuality and complexity affect supply chain visibility. *Journal of Production Economics, 127*(2), 372-383.

Caridi, M., Crippa, L., Perego, A., Saianesi, A., & Turmino, A. (2010b). Measuring visibility to improve supply chain performance: a quantitative approach. *Benchmarking: An International Journal, 17*(4), 593-615.

Cataldo, M., Bass, M., Herbsleb, J. D., & Bass, L. (2007). On Coordination Mechanisms in Global Software Development. *ICGSE, 7*, 71-80.

Chen, H.-M. (2008). *Towards service engineering: service orientation and business-IT alignment.* Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, 114-114.

Cheney, J., Hunt, R., Jacob, K., Porter, R., & Summers, B. (2012). The efficiency and integrity of payment card systems: industry views on the risks posed by data breaches. *FRB of Philadelphia-Payment Cards Center Discussion Paper*(12-04).

Clarke, P., & O'Connor, R. V. (2013). An empirical examination of the extent of software process improvement in software SMEs. *Journal of Software: Evolution and Process, 25*(9), 981-998.

Cockburn, A. (2006). *Agile software development: the cooperative game*: Pearson Education.

Cohen, L., & Young, A. (2006). *Multisourcing*: Harvard Business School Press.

Costa, J. M., Cataldo, M., & de Souza, C. R. (2011). *The scale and evolution of coordination needs in large-scale distributed projects: implications for the future generation of collaborative tools.* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 3151-3160.

Cropper, S. (2008). *The Oxford handbook of inter-organizational relations*: Oxford Handbooks Online.

Cummings, T., & Worley, C. (2014). *Organization development and change*: Cengage Learning.

Currie, W. L., & Seltsikas, P. (2001). Exploring the supply-side of IT outsourcing: evaluating the emerging role of application service providers. *European Journal of Information Systems, 10*(3), 123-134.

Daft, R. L. (2009). *Organization theory and design*: South-Western Pub.

Datta, P. P., & Christopher, M. G. (2011). Information sharing and coordination mechanisms for managing uncertainty in supply chains: a simulation study. *International Journal of Production Research, 49*(3), 765-803.

Davenport, T. H., & Short, J. E. (2003). Information technology and business process redesign. *Operations management: critical perspectives on business and management, 1*, 1-27.

David, J. S., Schuff, D., & St Louis, R. (2002). Managing your total IT cost of ownership. *Communications of the ACM, 45*(1), 101-106.

DeSanctis, G., & Jackson, B. M. (1994). Coordination of information technology management: Team-based structures and computer-based communication systems. *Journal of Management Information Systems, 10*(4), 85-110.

Dingsøyr, T., Itkonen, J., & Fægri, T. (2013, 7 June 2013). *An Overview of Research on Large-Scale Agile Software Development.* Proceedings of the 1st Workshop on Researcch Challenges in Large-Scale Agile Development, Vienna.

Dingsøyr, T., & Moe, N. B. (2013). Research challenges in large-scale agile software development. *ACM SIGSOFT Software Engineering Notes, 38*(5), 38-39.

Disney, S. M., & Towill, D. R. (2003). The effect of vendor managed inventory (VMI) dynamics on the Bullwhip Effect in supply chains. *International Journal of Production Economics, 85*(2), 199-215.

Dorairaj, S., Noble, J., & Malik, P. (2012). Understanding team dynamics in distributed Agile software development *Agile Processes in Software Engineering and Extreme Programming* (pp. 47-61): Springer.

Dove, R., & LaBarge, R. (2014). Fundamentals of Agile Systems Engineering–Part 2.

Duffy, L. (1993). Team decision making and technology. *Individual and Group Decision Making: Current Issues. Hillsdale, NJ: Lawrence Erlbaum Associates*, 247-266.

Dul, J., & Hak, T. (2012). *Case study methodology in business research* (1st edition ed.). Oxford: Elsevier.

Earl, M. J. (1996). The risks of outsourcing IT. *Sloan management review, 37*, 26-32.

Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research *Guide to advanced empirical software engineering* (pp. 285-311): Springer.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of management review, 14*(4), 532-550.

Eisenhardt, K. M., & Zbaracki, M. J. (1992). Strategic decision making. *Strategic Management Journal, 13*(S2), 17-37.

Ellram, L. M., Tate, W. L., & Billington, C. (2004). Understanding and managing the services supply chain. *Journal of Supply Chain Management, 40*(4), 17-32.

Ellram, L. M., Tate, W. L., & Billington, C. (2007). Services supply management: The next frontier for improved organizational performance. *California Management Review, 49*(4), 44-66.

Evolven. (2014). Downtime, Outages and Failures - Understanding Their True Costs. Retrieved from http://www.evolven.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html

Fadok, D. S. (1995). John Boyd and John Warden: Air Power's Quest for Strategic Paralysis: DTIC Document.

Feitelson, D., Frachtenberg, E., & Beck, K. (2013). Development and Deployment at Facebook.

Fitzgerald, B., & Stol, K.-J. (2014). *Continuous Software Engineering and Beyond: Trends and Challenges.* Proceedings Rcose 2014.

Flinders, K. (2014, TechTarget). Lloyds is the latest bank to experience major IT outage, *ComputerWeekly.com*. Retrieved from http://www.computerweekly.com/news/2240213200/Lloyds-latest-to-bank-to-experience-major-IT-outage

Forssell, D. C., & Powers, W. T. (2009). *Perceptual Control Theory: Science & Applications-a Book of Readings*: Living Control Systems Pub.

Francis, V. (2008). Supply chain visibility: lost in translation? *Supply chain management: An international journal, 13*(3), 180-184.

Freeman, L. C. (1979). Centrality in social networks conceptual clarification. *Social networks, 1*(3), 215-239.

Freudenberg, S., & Sharp, H. (2010). The top 10 burning research questions from practitioners. *Software, IEEE, 27*(5), 8-9.

Frické, M. (2009). The knowledge pyramid: a critique of the DIKW hierarchy. *Journal of Information Science, 35*(2), 131-142.

Fuks, H., Raposo, A., Gerosa, M. A., Pimentel, M., Filippo, D., & Lucena, C. (2008). *Inter-and intra-relationships between communication coordination and cooperation in the scope of the 3C Collaboration Model.* Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on, 148-153.

Galbraith, J. R. (1971). Matrix organization designs How to combine functional and project forms. *Business Horizons, 14*(1), 29-40.

Galbraith, J. R. (1977). Organization Design: An Information Processing View. *Organizational Effectiveness Center and School*, 21.

Galbraith, J. R. (1995). *Designing organizations: An executive briefing on strategy, structure, and process*: Jossey-Bass.

Garg, A. (2009). Agile Software Development. *DRDO Science Spectrum*.

Gersick, C. J. G. (1991). Revolutionary change theories: A multilevel exploration of the punctuated equilibrium paradigm. *Academy of management review*, 10-36.

Gibbert, M., & Ruigrok, W. (2010). The ‘‘What’’and ‘‘How’’of case study rigor: three strategies based on published work. *Organizational Research Methods, 13*(4), 710-737.

Glaser, & Strauss. (1967). *The discovery of grounded theory: Strategies for qualitative research*: Aldine de Gruyter.

Goffman, E. (1959). *The presentation of self in everyday life*. Edinburgh: University of Edinburgh.

Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The qualitative report, 8*(4), 597-607.

Goodwin, P., & Wright, G. (2007). *Decision analysis for management judgment* (4th edition ed.). New York: Wiley.

Google. (2015). Google Ranking algorithm, from http://en.wikipedia.org/wiki/Google_Scholar#Ranking_algorithm

Green, R., Mazzuchi, T., & Sarkani, S. (2010). Communication and Quality in Distributed Agile Development: An Empirical Case Study. *International Journal of Information Technology, 6*(1).

Greening, D. R. (2010). *Enterprise Scrum: Scaling Scrum to the executive level.* System Sciences (HICSS), 2010 43rd Hawaii International Conference on, 1-10.

Gregory, J. B., Beck, J. W., & Carr, A. E. (2011). Goals, feedback, and self-regulation: Control theory as a natural framework for executive coaching. *Consulting Psychology Journal: Practice and Research, 63*(1), 26.

Groot, A., & Spiekerman, J. (1969). *Methodology: Foundations of inference and research in the behavioral sciences*: Mouton The Hague.

Gunasekaran, A., Patel, C., & McGaughey, R. E. (2004). A framework for supply chain performance measurement. *International Journal of Production Economics, 87*(3), 333-347.

Guo, P. J., Zimmermann, T., Nagappan, N., & Murphy, B. (2011). *Not my bug! and other reasons for software bug report reassignments.* Proceedings of the ACM 2011 conference on Computer supported cooperative work, 395-404.

Hamid, N. A. A., & Salim, J. (2011). A conceptual framework of knowledge transfer in Malaysia e-government IT outsourcing: an integration with transactive memory system (TMS). *International Journal of Computer Science, 8*, 51-64.

Hannan, M. T., & Freeman, J. (1984). Structural inertia and organizational change. *American sociological review*, 149-164.

Hardion, V., Spruce, D. P., Lindberg, M., Otero, A. M., Lidon-Simon, J., Jamroz, J. J., & Persson, A. (2013). Configuration Management of the control system. *THPPC013*.

Harland, C. (1996). Supply chain management, purchasing and supply management, logistics, vertical integration, materials management and supply chain dynamics. *Blackwell Encyclopedic dictionary of operations management. UK: Blackwell, 15*.

Hatonen, J., & Eriksson, T. (2009). 30+ years of research and practice of outsourcing. *Journal of International Management, 2009*(15), 142-155.

Henneman, E. A., Lee, J. L., & Cohen, J. I. (1995). Collaboration: a concept analysis. *Journal of advanced Nursing, 21*(1), 103-109.

Hildenbrand, T., Geisser, M., Kude, T., Bruch, D., & Acker, T. (2008). *Agile methodologies for distributed collaborative development of enterprise applications.* Proceedings International Conference on Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008, 540-545.

Hiles, A. (2002). *The Complete Guide to IT Service Level Agreements: Aligning IT Services to Business Needs*: Rothstein Associates Inc.

Holz, H., & Melnik, G. (2004). Research on Learning Software Organizations–Past, Present, and Future *Advances in Learning Software Organizations* (pp. 1-6): Springer.

Hoyle, D. (2001). ISO 9000: quality systems handbook.

Huberman, A. M., & Miles, M. B. (1984). *Innovation up close: How school improvement works*: Springer.

Huijgens, H., & Solingen, R. v. (2014). *A replicated study on correlating agile team velocity measured in function and story points.* Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics, 30-36.

Huijgens, H., Solingen, R. v., & Deursen, A. v. (2014). *How to build a good practice software project portfolio?* Companion Proceedings of the 36th International Conference on Software Engineering, 64-73.

Humble, J. (2010). Agile Release Management: ThoughtWorks.

Humble, J., & Farley, D. (2010). *Continuous delivery: reliable software releases through build, test, and deployment automation*: Addison-Wesley Professional.

Humble, J., & Molesky, J. (2011). Why Enterprises Must Adopt Devops to Enable Continuous Delivery. *Cutter IT Journal, 24*(8), 6.

Huysman, M., & De Wit, D. (2004). Practices of managing knowledge sharing: towards a second wave of knowledge management. *Knowledge and process management, 11*(2), 81-92.

IEEE. (2008). IEEE SA - 829-2008 *Standard for Software and System Test*: The Institute of Electrical and Electronics Engineers.

Ilgen, D. R., Hollenbeck, J. R., Johnson, M., & Jundt, D. (2005). Teams in organizations: From input-process-output models to IMOI models. *Annual Review of Psychology, 56*, 517-543.

ITGI. (2007). *Cobit 4.1* (Vol. 4.1). Rolling Meadows: IT Governance Institute.

Jalali, S., & Wohlin, C. (2012). Global software engineering and agile practices: a systematic review. *Journal of Software: Evolution and Process, 24*(6), 643-659.

Jansen, S., & Cusumano, M. A. (2013). Defining software ecosystems: a survey of software platforms and business network governance. *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, 13.

Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009). *A sense of community: A research agenda for software ecosystems.* Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on, 187-190.

Jantti, M. (2011). *Improving Incident Management Processes in Two IT Service Provider Companies*. Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on, 26-30.

Jäntti, M. (2012a). *Examining Challenges in IT Service Desk System and Processes: A Case Study.* ICONS 2012, The Seventh International Conference on Systems, 105-108.

Jäntti, M. (2012b). Improving IT service desk and service management processes in finnish tax administration: a case study on service engineering. *Product-Focused Software Process Improvement*, 218-232.

Jäntti, M., & Järvinen, J. (2011). Improving the Deployment of IT Service Management Processes: A Case Study. *Systems, Software and Service Process Improvement*, 37-48.

Jäntti, M., Lahtela, A., & Kaukola, J. (2011). Establishing a Measurement System for IT Service Management Processes: A Case Study. *International Journal on Advances in Systems and Measurements, 3*(3 and 4), 125-136.

Jantti, M., & Suhonen, A. (2012). *Improving Service Level Management Practices: A Case Study in an IT Service Provider Organization.* Advanced Applied Informatics (IIAIAAI), 2012 IIAI International Conference on, 139-144.

Jonker, C., van Riemsdijk, M., & Vermeulen, B. (2011). Shared mental models. *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, 132-151.

Jrad, R. B., Ahmed, M. D., & Sundaram, D. (2014). *Insider Action Design Research a multi-methodological Information Systems research approach.* IEEE 8th International Conference on Research Challenges in Information Science (RCIS), Marrakesh, Marocco, 1-12.

Jyothi, V. E., & Rao, K. N. (2011). Effective implementation of agile practices. *International Journal of Advanced Computer Science and Applications, 2*(3).

Kang, H., & Bradley, G. (2002). Measuring the performance of IT services:: An assessment of SERVQUAL. *International Journal of Accounting Information Systems, 3*(3), 151-164.

Kim, S., Park, S., Yun, J., & Lee, Y. (2008). *Automated continuous integration of component-based software: An industrial experience.* Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering, 423-426.

Kniberg, H., & Ivarsson, A. (2012). Scaling Agile @ Spotify. Retrieved from https://dl.dropbox.com/u/1018963/Articles/SpotifyScaling.pdf

Knight, R., Rabideau, G., Chien, S., Engelhardt, B., & Sherwood, R. (2001). Casper: Space exploration through continuous planning. *Intelligent Systems, IEEE, 16*(5), 70-75.

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development* (Vol. 1): Prentice-Hall Englewood Cliffs, NJ.

Kruchten, P. (2013). Contextualizing agile software development. *Journal of Software: Evolution and Process, 25*(4), 351-361.

Krueger, R. A., & Casey, M. (2008). *A practical guide for applied research*.

Kumbakara, N. (2008). Managed IT services: the role of IT standards. *Information Management & Computer Security, 16*(4), 336-359.

Lacity, M., Khan, S., & Willcocks, L. (2009). A review of IT outsourcing literature: Insights for practice. *Journal of Strategic Information Systems, 2009*(18), 130-146.

Larman, C., & Vodde, B. (2008). *Scaling lean & agile development: thinking and organizational tools for large-scale Scrum*: Pearson Education.

Larman, C., & Vodde, B. (2013). Scaling Agile Development. *CrossTalk*, 9.

Larman, C., & Vodde, B. (2015). Less - More with LeSS, 2015, from http://less.works/

Lee, C., & Helal, S. (2002). Protocols for service discovery in dynamic and mobile networks. *International Journal of Computer Research, 11*(1), 1-12.

Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly, 34*(1), 87.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). The bullwhip effect in supply chains. *Sloan Management Review, 38*, 93-102.

Lee, S., & Yong, H.-S. (2010). Distributed agile: project management in a global environment. *Empirical Software Engineering, 15*(2), 204-217.

Leffingwell, D. (2007). *Scaling software agility: best practices for large enterprises*: Pearson Education.

Leffingwell, D. (2010). *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*: Addison-Wesley Professional.

Lehto, I., & Rautiainen, K. (2009). *Software development governance challenges of a middle-sized company in agile transition.* Proceedings of the 2009 ICSE Workshop on Software Development Governance, 36-39.

Lemoine, W., & Dagnæs, L. (2003). Globalisation strategies and business organisation of a network of logistics service providers. *International Journal of Physical Distribution & Logistics Management, 33*(3), 209-228.

Lerner, A. (2014). The Cost of Downtime. Retrieved from http://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/

Lewis, K., & Herndon, B. (2011). Transactive memory systems: Current issues and future research directions. *Organization Science, 22*(5), 1254-1265.

Lim, B. C., & Klein, K. J. (2006). Team mental models and team performance: a field study of the effects of team mental model similarity and accuracy. *Journal of Organizational Behavior, 27*(4), 403-418.

Lin, B. W. (2007). Information technology capability and value creation: Evidence from the US banking industry. *Technology in Society, 29*(1), 93-106.

Locke, E. A., & Latham, G. P. (1990). *A theory of goal setting & task performance*: Prentice-Hall, Inc.

Loukides, M. (2012). *What is DevOps?* : O'Reilly.

Mahnic, V., & Zabkar, N. (2012). Measuring Progress of Scrum-based Software Projects. *Electronics and Electrical Engineering, 18*(8), 73-76.

Malone, T. W., & Crowston, K. (1990). *What is coordination theory and how can it help design cooperative work systems?* Proceedings of the 1990 ACM conference on Computer-supported cooperative work, 357-370.

Marshall, M. N. (1996). Sampling for qualitative research. *Family practice, 13*(6), 522-526.

Martini, A., Pareto, L., & Bosch, J. (2013a). *Communication factors for speed and reuse in large-scale agile software development.* Proceedings of the 17th International Software Product Line Conference, 42-51.

Martini, A., Pareto, L., & Bosch, J. (2013b). Improving businesses success by managing interactions among agile teams in large organizations *Software Business. From Physical Products to Software Services and Solutions* (pp. 60-72): Springer.

Mathieu, J. E., Heffner, T. S., Goodwin, G. F., Salas, E., & Cannon-Bowers, J. A. (2000). The influence of shared mental models on team process and performance. *Journal of Applied Psychology, 85*(2), 273.

McNaughton, B., Ray, P., & Lewis, L. (2010). Designing an evaluation framework for IT service management. *Information & Management, 47*(4), 219-225.

Melville, N., Kraemer, K., & Gurbaxani, V. (2004). Information Technology and Organizational Performance: An Integrative Model of IT Business Value. *Management Information Systems Quarterly, 28*(2), 283-322.

Mentzer, J. T., DeWitt, W., Keebler, J. S., Min, S., Nix, N. W., Smith, C. D., & Zacharia, Z. G. (2001). Defining supply chain management. *Journal of Business Logistics, 22*(2), 1-25.

Mentzer, J. T., & Flint, D. J. (1997). Validity in logistics research. *Journal of Business Logistics, 18*(1), 199-216.

Merton, R. K. (1968). The Matthew effect in science. *Science, 159*(3810), 56-63.

Meshkova, E., Riihijarvi, J., Oldewurtel, F., Jardak, C., & Mahonen, P. (2008). *Service-oriented design methodology for wireless sensor networks: A view through case studies.* Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08. IEEE International Conference on, 146-153.

Mesmer-Magnus, J. R., & DeChurch, L. A. (2009). Information sharing and team performance: A meta-analysis. *Journal of Applied Psychology, 94*(2), 535.

Mills, A. J., Durepos, G., & Wiebe, E. (2009). *Encyclopedia of case study research* (Vol. 2): Sage Publications.

Min, H., & Zhou, G. (2002). Supply chain modeling: past, present and future. *Computers & Industrial Engineering, 43*(1), 231-249.

Minto, S., & Murphy, G. C. (2007). *Recommending emergent teams.* Mining Software Repositories, 2007. ICSE Workshops MSR'07. Fourth International Workshop on, 5-5.

Mintzberg, H. (1989). *Mintzberg on management*: Free Press.

Mishra, D., & Mishra, A. (2011). Complex software project development: agile methods adoption. *Journal of Software Maintenance and Evolution: Research and Practice, 23*(8), 549-564.

Moe, N. B., Dingsoyr, T., & Dyba, T. (2008). *Understanding self-organizing teams in agile software development.* Proceedings of the 19th Australian Conference on Software Engineering, Perth, Australia, 76-85.

Moeller, S. (2010). Characteristics of services–a new approach uncovers their value. *Journal of Services Marketing, 24*(5), 359-368.

Moniruzzaman, A., & Hossain, D. S. A. (2013). Comparative Study on Agile software development methodologies. *arXiv preprint arXiv:1307.3356*.

Morgan, M., Gibbs, S., Maxwell, K., & Britten, N. (2002). Hearing children's voices: methodological issues in conducting focus groups with children aged 7-11 years. *Qualitative research, 2*(1), 5-20.

Muslu, K., Brun, Y., & Meliou, A. (2013). *Data debugging with continuous testing.* Proceedings of the 9th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering, new ideas track, ESEC/FSE.

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization, 17*(1), 2-26.

Nagasubramanian, R., & Rajagopalan, S. (2012). Payment Gateway Innovation in Multiple Payments. *International Journal of Computer Applications, 59*(16), 33-43.

Neely, S., & Stolt, S. (2013). *Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy).* Agile Conference 2013, Nashville, USA, 121-128.

Nevo, D., & Wand, Y. (2005). Organizational memory information systems: a transactive memory approach. *Decision Support Systems, 39*(4), 549-562.

Niessink, F. (2001). *Perspectives on improving software maintenance.* (PhD), Vrije Universiteit te Amsterdam, Amsterdam.

Niessink, F., & van Vliet, H. (1998). Towards Mature IT Services. *Software Process - Improvement and Practice, 4*(2), 55-71.

Niessink, F., & van Vliet, H. (1999). The Vrije Universiteit IT Service Capability Maturity Model. *Amsterdam, Vrije Univeristeit Amsterdam*, 1-77.

Niessink, F., & van Vliet, H. (2000). Software Maintenance from a Service Perspective. [Review article]. *Journal of Software Maintenance, 12*(2), 103-120.

Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. New York: Oxford University Press, USA.

OGC. (2007). *ITIL Service Operation*. UK: The Stationary Office.

Olsson, H. H., Alahyari, H., & Bosch, J. (2012). *Climbing the" Stairway to Heaven"--A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software.* Proceedings of the 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Cesme, Izmir, Turkey, 392-399.

Oppenheim, D., Bagheri, S., Ratakonda, K., & Chee, Y.-M. (2011). *Agility of enterprise operations across distributed organizations: A model of cross enterprise collaboration.* Proceedings SRII Global Conference (SRII), 2011 Annual, 154-162.

Oppenheimer, D., Ganapathi, A., & Patterson, D. A. (2002). *Why do Internet services fail and what can be done about it?* : Computer Science Division, University of California.

Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2009). *Using scrum in distributed agile development: A multiple case study.* Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on, 195-204.

Paasivaara, M., Lassenius, C., & Heikkila, V. T. (2012). *Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrums really work?* Proceedings of the 6th IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2012, Lund, Sweden, 235-238.

Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1985). A conceptual model of service quality and its implications for future research. *The Journal of Marketing, 49*(4), 41-50.

Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1988). SERVQUAL: A Multiple-Item Scale for Measuring Consumer Perceptions of Service Quality. *Journal of Retailing, 64*, 1.

Peppard, J. (2003). Managing IT as a Portfolio of Services. *European Management Journal, 21*(4), 467-483.

Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software, 82*(9), 1479-1490.

Phifer, B. (2011). Next-Generation Process Integration: CMMI and ITIL Do Devops. *Cutter IT Journal, 24*(8), 28.

Pikkarainen, M., Salo, O., & Still, J. (2005). Deploying agile practices in organizations: a case study *Software Process Improvement* (pp. 16-27): Springer.

Plugge, A., & Janssen, M. (2009). Managing change in IT outsourcing arrangements: An offshore service provider perspective on adaptability. *Strategic Outsourcing: An International Journal, 2*(3), 257-274.

Plummer, D. C., & McCoy, D. W. (2006). Achieving agility: Defining agility in an IT context. *Gartner Inc., ID*(G00137819).

Poppendieck, M., & Poppendieck, T. (2007). *Implementing lean software development: from concept to cash*: Pearson Education.

Poppo, L., & Zenger, T. (2002). Do formal contracts and relational governance function as substitutes or complements? *Strategic Management Journal, 23*(8), 707-725.

Port, D., & Bui, T. (2009). Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications. *European Journal of Information Systems, 18*(4), 317-331.

Prajogo, D., & Olhager, J. (2011). Supply chain integration and performance: The effects of long-term relationships, information technology and sharing, and logistics integration. *International Journal of Production Economics*.

Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software, 81*(11), 1899-1919.

Ralph, P., Shportun, P., & Bloomberg, L. (2013). Scrum Abandonment In Distributed Teams: A Revelatory Case.

Rashed, C. A. A., Azeem, A., & Halim, Z. (2010). Effect of Information and Knowledge Sharing on Supply Chain Performance: A Survey Based Approach. *Journal of Operations and Supply Chain Management, 3*(2), 61-77.

Rautiainen, K., von Schantz, J., & Vahaniitty, J. (2011). *Supporting Scaling Agile with Portfolio Management: Case Paf. com.* Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS), Hawaii, USA, 1-10.

ResponsiveOrg. (2014). The Responsive Organization Retrieved 2014-12-31, from http://www.theresponsiveorg.com/

Riedl, C., Böhmann, T., Rosemann, M., & Krcmar, H. (2009). Quality management in service ecosystems. *Information Systems and E-Business Management, 7*(2), 199-221.

Ringstad, M. A., Dingsøyr, T., & Moe, N. B. (2011). Agile process improvement: Diagnosis and planning to improve teamwork *Systems, Software and Service Process Improvement* (pp. 167-178): Springer.

Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *Software, IEEE, 17*(4), 26-32.

Romanelli, E., & Tushman, M. L. (1994). Organizational transformation as punctuated equilibrium: an empirical test. *Academy of Management Journal, 37*(5), 1141-1166.

Rosa, M. M. S. (2012). *A Method for Identifying IT Services Using Incidents.* (PhD), Universidade Tecnica de Lisboa, Lisboa.

Rowley, J. (2007). The wisdom hierarchy - representations of the DIKW hierarchy. *Journal of Information Science, 33*(2), 163-180.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131-164.

Saddington, P. (2012). *Scaling Product Ownership Through Team Alignment and Optimization.* Proceedings Agile conference, 123-130.

Sahin, F., & Robinson, E. P. (2005). Information sharing and coordination in make-to-order supply chains. *Journal of Operations Management, 23*(6), 579-598.

Saldaña, J. (2012). *The coding manual for qualitative researchers* (1st edition ed.). Arizona: SAGE Publications Limited.

Sallé, M., & Bartolini, C. (2004, 19-23 April 2004). *Management by contract.* NOMS, Seoul, Korea, 787-800.

Salo, O., & Abrahamsson, P. (2005). *Integrating agile software development and software process improvement: a longitudinal case study.* International Symposium on Empirical Software Engineering (ISESE), 2005, Queensland, Australia, 10 pp.

Sandelowski, M. (1995). Sample size in qualitative research. *Research in nursing & health, 18*(2), 179-183.

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for business students* (5th edition ed.). Essex: Pearson Education Limited.

Saunders, M., Lewis, P., & Thornhill, A. (2011). *Research methods for business students, 5/e*: Pearson Education India.

Savić, A. (2008). Managing IT-related operational risks. *Ekonomski anali, 53*(176), 88-109.

Scheerer, A., Hildenbrand, T., & Kude, T. (2014). *Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective.* Proceedings of the 47th Hawaii International Conference on System Science, Hawaii, USA.

Schnitter, J., & Mackert, O. (2011). Large-scale agile software development at SAP AG *Evaluation of Novel Approaches to Software Engineering* (pp. 209-220): Springer.

Schuler, D., & Zimmermann, T. (2008). *Mining usage expertise from version archives.* Proceedings of the 2008 international working conference on Mining software repositories, 121-124.

Schwaber, K. (2004). *Agile project management with Scrum*: O'Reilly Media, Inc.

Schwaber, K. (2011). *The enterprise and scrum*: Microsoft Press.

Schwaber, K. (2015). Scrum.org - Path to Agility, 2015, from https://www.scrum.org/About/All-Articles/articleType/ArticleView/articleId/691/The-Path-to-Agility

Scott, J. (2014, 25 June 2012). RBS enters fifth day of software failures, *ComputerWeekly.com*. Retrieved from http://www.computerweekly.com/news/2240158575/RBS-still-feeling-knock-on-effects-of-software-issues

Shachaf, P. (2008). Cultural diversity and information and communication technology impacts on global virtual teams: An exploratory study. *Information & Management, 45*, 131-142.

Shalloway, A., Beaver, G., & Trott, J. R. (2009). *Lean-agile software development: achieving enterprise agility*: Pearson Education.

Sharp, H., & Robinson, H. (2008). Collaboration and co-ordination in mature eXtreme programming teams. *International Journal of Human-Computer Studies, 66*(7), 506-518.

Sharp, H., & Robinson, H. (2010). Three 'C's of agile practice: collaboration, co-ordination and communication *Agile Software Development* (pp. 61-85): Springer.

Shepherd, C., & Günter, H. (2006). Measuring supply chain performance: Current research and future directions. *International Journal of Productivity and Performance Management, 55*(3-4), 242-258.

Soundararajan, S., & Arthur, J. D. (2009). *A soft-structured agile framework for larger scale systems development.* Proceedings of the 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), 187-195.

Spencer-Thomas. (2012). Writing a Press Release, from http://www.owenspencer-thomas.com/journalism/media-tips/writing-a-press-release

Stacey, R. D. (1995). The science of complexity: An alternative perspective for strategic change processes. *Strategic Management Journal, 16*(6), 477-495.

Ståhl, D., & Bosch, J. (2013). *Experienced benefits of continuous integration in industry software product development: A case study.* The 12th IASTED International Conference on Software Engineering,(Innsbruck, Austria, 2013), 736-743.

Ståhl, D., & Bosch, J. (2014). Modeling continuous integration practice differences in industry software development. *Journal of Systems and Software, 87*, 48-59.

Stelzer, D., & Mellis, W. (1998). Success factors of organizational change in software process improvement. *Software Process: Improvement and Practice, 4*(4), 227-250.

Stettina, C. J., Heijstek, W., & Fægri, T. E. (2012). *Documentation work in agile teams: the role of documentation formalism in achieving a sustainable practice.* Agile Conference (AGILE), 2012, 31-40.

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.

Stout, R. J., Cannon-Bowers, J. A., Salas, E., & Milanovich, D. M. (1999). Planning, shared mental models, and coordinated performance: An empirical link is established. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 41*(1), 61-71.

Strode, D. E., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software, 85*(6), 1222-1238.

Sugumaran, V., & Arogyaswamy, B. (2003). Measuring it performance:" Contingency" variables and value modes. *Journal of Computer Information Systems, 44*(2), 79-86.

Susarla, A. (2003). *Understanding the organization of managed service providers: an analysis of customer satisfaction and contracting in markets for hosted IT services.* (Doctor in Philosophy), The University of Texas at Austin.

Susman, G. I., & Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative Science Quarterly*, 582-603.

Sutherland, J. (2001). Agile can scale: Inventing and reinventing scrum in five companies. *Cutter IT Journal, 14*(12), 5-11.

Sutherland, J. (2005). *Future of scrum: Parallel pipelining of sprints in complex projects.* Proceedings of the Agile Conference 2005, Washington, DC, USA, 90-99.

Sutherland, J., Schoonheim, G., & Rijk, M. (2009). *Fully distributed scrum: Replicating local productivity and quality with offshore teams.* 42nd Hawaii International Conference on System Sciences (HICSS), Hawaii, USA, 1-8.

Sutherland, J., Schoonheim, G., Rustenburg, E., & Rijk, M. (2008). *Fully distributed scrum: The secret sauce for hyperproductive offshored development teams.* Agile, 2008. AGILE'08. Conference, 339-344.

Sutherland, J., & Schwaber, K. (2013). The Scrum Guide TM

Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). *Distributed scrum: Agile project management with outsourced development teams.* Proceedings 40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007, 274a-274a.

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review, 64*(1), 137-146.

Talby, D., & Dubinsky, Y. (2009). *Governance of an agile software project.* Proceedings of the 2009 ICSE Workshop on Software Development Governance, Washington, DC, USA, 40-45.

Talk, A. (2013). Incident Management Retrieved March 2, 2013, from http://wiki.en.it-processmaps.com/index.php/Incident_Management

Tan, W., Cater-Steel, A., & Toleman, M. (2009a). Implementing it service management: A case study focussing on critical success factors. *Journal of Computer Information Systems, 50*(2), 1.

Tan, W., Cater-steel, A., & Toleman, M. (2009b). Implementing IT Service Management: A case study focussing on critical success factors. *Journal of Computer Information Systems*.

Taylor, F. W. (1914). *The principles of scientific management*: Harper.

Team, C. P. (2010a). CMMI for Development, version 1.3

Team, C. P. (2010b). CMMI for Services, version 1.3 (pp. 533). Retrieved from ftp://ftp.sei.cmu.edu/pub/documents/09.reports/09tr001.pdf

TFSC. (2011). The Changing Face of Payments - A Review of Current Payments Infrastructures, Drivers for Change and Implications for the Future. Retrieved from

Thomson, A., & Miller, H. (2012, Sep 21, 2012). RIM Falls as BlackBerry Service Disrupted in Europe, *Bloomberg News*. Retrieved from http://www.bloomberg.com/news/2012-09-21/rim-says-blackberry-service-disrupted-in-europe.html

Thouin, M. F., Hoffman, J. J., & Ford, E. W. (2009). IT outsourcing and firm-level performance: A transaction cost perspective. *Information & Management, 46*(8), 463-469.

Unterkalmsteiner, M., Gorschek, T., Islam, A. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and measurement of software process improvement—A systematic literature review. *Software Engineering, IEEE Transactions on, 38*(2), 398-424.

Vacanti, D., & Vallet, B. (2014). Actionable Metrics at Siemens Health Services.

van Bon, J., Jong, A., & Kolthof, A. (2007). *Foundations of IT Service Management Based on ITIL, Volume 3*: Van Haren Publishing.

Van de Ven, A. H., Delbecq, A. L., & Koenig Jr, R. (1976). Determinants of coordination modes within organizations. *American sociological review*, 322-338.

Van Der Aalst, W. M., Reijers, H. A., & Song, M. (2005). Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW), 14*(6), 549-593.

van der Aalst, W. M., Reijers, H. A., Weijters, A. J., van Dongen, B. F., De Medeiros, A. A., Song, M., & Verbeek, H. (2007). Business process mining: An industrial application. *Information Systems, 32*(5), 713-732.

Van Tiem, D. M., Karve, S., & Rosenzweig, J. Hidden Order of Human Performance Technology. *Handbook of Human Performance Technology*, 1251.

VersionOne. (2013). 7th Annual State of Agile Development Survey. Retrieved from

Viswanadham, N., Desai, V., & Gaonkar, R. (2005). *Bullwhip effect in integrated manufacturing and service networks.* Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, 2994-2999.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology, 53*(1), 58-70.

Vlietland, J. (2011). Information visibility and performance of global IT service networks: a study in the banking industry.

Vlietland, J., van Solingen, R., & van Vliet, H. (2015). Delivering business value faster by sets of codependent Scrum teams: a governance framework. *Journal of System and Software*.

Vlietland, J., & van Vliet, H. (2013). *Visibility and Performance of IT Incident Handling: A Control Theory Perspective*. Proceeding of the Joint Conference of the 23nd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IWSM-MENSURA), Ankara, 203-212.

Vlietland, J., & van Vliet, H. (2014a). *Alignment Issues in chains of Scrum teams.* Proceedings of the 5th International Conference on Software Business, Cyprus, 301-306.

Vlietland, J., & van Vliet, H. (2014b). Improving IT incident handling performance with information visibility. *Journal of Software: Evolution and Process, 2014*(26), 1106-1127 doi: 10.1002/smr.1649

Vlietland, J., & van Vliet, H. (2014c). Information sharing for effective IT incident resolving in IT service provider networks: A financial service case study. *Journal of Software: Evolution and Process* doi: 10.1002/smr.1697

Vlietland, J., & van Vliet, H. (2015a). *Improving the Agility of IT Service Networks.* Proceedings of the 6th International Conference on Software Business, Portugal.

Vlietland, J., & van Vliet, H. (2015b). Towards a governance framework for chains of Scrum teams. *Journal of Information and Software Technology, Volume 57* (January 2015), Pages 52–65. doi: 10.1016/j.infsof.2014.08.008

Waardenburg, G., & van Vliet, H. (2012). When agile meets the enterprise. *Information and Software Technology, 55*(12), 2154-2171.

Walsham, G. (1995). The emergence of interpretivism in IS research. *Information Systems Research, 6*(4), 376-394.

Webster Jr, F. E. (1992). The changing role of marketing in the corporation. *The Journal of Marketing*, 1-17.

Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind *Theories of group behavior* (pp. 185-208): Springer.

Wegner, D. M., Erber, R., & Raymond, P. (1991). Transactive memory in close relationships. *Journal of Personality and Social Psychology, 61*(6), 923.

Wei, H.-L., & Wang, E. T. (2010). The strategic value of supply chain visibility: increasing the ability to reconfigure. *European Journal of Information Systems, 19*(2), 238-249.

Weick, K., & Quinn, R. (1999). Organizational change and development. *Annual Review of Psychology, 50*(1), 361-386.

Wiener, N. (1965). *Cybernetics-: or the Control and Communication in the Animal and the Machine* (Vol. 25): MIT press.

Wilhelm, M. M. (2011). Managing coopetition through horizontal supply chain relations: linking dyadic and network levels of analysis. *Journal of Operations Management*.

Winniford, M., Conger, S., & Erickson-Harris, L. (2009). Confusion in the ranks: IT service management practice and terminology. *Information Systems Management, 26*(2), 153-163.

Wood, D. J., & Gray, B. (1991). Toward a comprehensive theory of collaboration. *The Journal of Applied Behavioral Science, 27*(2), 139-162.

Woodward, E., Surdek, S., & Ganis, M. (2010). *A practical guide to distributed Scrum*: Pearson Education.

Yamakami, T. (2013). *Self-Innovation Skill-Based Change Management: An Approach Toward Flexible Organizational Management.* Proceedings of the 3rd International Conference on Cloud and Green Computing, 256-260.

Yin , R. K. (1981). The case study crisis: some answers. *Administrative Science Quarterly, 26*(1), 58-65.

Yin , R. K. (2009). *Case study research: design and methods* (2rd edition ed.). London: Sage Publications.

Yu, L., Suojapelto, K., Hallikas, J., & Tang, O. (2008). Chinese ICT industry from supply chain perspective - A case study of the major Chinese ICT players. *International Journal of Production Economics, 2008*(115), 374 - 387.

Zachman, J. (2002). The zachman framework for enterprise architecture. *Zachman International*.

Zailani, S. H. M., & Kumar, K. M. (2011). Service Supply Chain (SSC): Proposed SSC Practices Measurement Items for Empirical Testing. *Journal of System and Management Sciences Vol, 1*(5), 11-18.

Zhang, A. N., Goh, M., de Souza, R., & Meng, F. (2011). Conceptual modelling for supply chain inventory visibility. *International Journal of Production Economics*.