AN ANALYSIS OF CHANGE SCENARIOS OF AN IT ORGANIZATION FOR FLEXIBILITY BUILDING

Complete Research

Nurdiani, Indira, Blekinge Institute of Technology, Karlskrona, Sweden, indira.nurdiani@bth.se

Fricker, Samuel A., Blekinge Institute of Technology, Karlskrona, Sweden, samuel.fricker@bth.se

Börstler, Jürgen, Blekinge Institute of Technology, Karlskrona, Sweden, jurgen.borstler@bth.se

Abstract

Flexibility is important for software organizations to cope with changes demanded in the business environment. So far, flexibility has been extensively studied from a software product and software development process point of view. However, there is little work on how to build flexibility at the level of the whole software organization. Thus, there is no clear understanding of how to effectively improve the ability of an organization to respond to changes in a timely fashion and with little effort. This paper presents the results of a grounded theory study on how flexibility is built and improved in an IT organization and provides a holistic and explanatory view of how this is achieved. Implications for research and practices are also provided.

Keywords: Flexibility, Software Organization, Grounded Theory

1 Introduction

A software organization is an entity that develops and delivers software to an external mass market, a company, or supports business operations within the same company. An IT organization is a form of software organization that operates, develops, and manages human resources, processes, and infrastructure (both hardware and software) that supports and is of importance for the business (Byrd and Turner, 2000; Knoll and Jarvenpaa, 1994).

Software organizations in general, and IT organizations in particular, need to be flexible and cope with the changes demanded in the business environment. Changes not only originate from the business (customers of an IT organization), they can also originate from suppliers, competitors, governing bodies, and also technological development (Byrd and Turner, 2000; Knoll and Jarvenpaa, 1994). In a turbulent business environment, IT organizations need to develop and improve strategies to be flexible. Without flexibility, an IT organization might lose its ability to support the business' market position and competitive advantage (Byrd and Turner, 2000).

Despite the importance of flexibility, it is not well elaborated how organizational flexibility is built. There are known prescriptive approaches in software engineering, such as Agile methodologies (Beck, 2000; Schwaber and Beedle, 2001), software product line engineering (Clements and Northrop, 2001), and software product management (Fricker and Schumacher, 2012). Agile methodology emerged as an approach that offers flexibility by welcoming changes in customer needs (Beck, 2000). Software product line engineering also offers flexibility through variability management (Chen et al., 2009). The area of product planning decision also offers approaches to achieve software product flexibility (Fricker and Schumacher, 2012). These approaches offer prescriptive suggestions on how to include flexibility in the software product, but not at an organizational level.

In this paper, we elaborated events that took place as an IT organization built and improved its flexibility. These events are then presented as a model that was developed through grounded theory. The data were collected through interviews with managers to identify scenarios where flexibility was built and improved in the organization. The scenarios that took place in the case company were related to change of organization structure, implementation of a new software development method, and implementation of a new legislative regulation. The elaboration of these events can be used as a stepping stone to further improve flexibility in a software organization.

The remainder of this paper is organized as follows. Section 2 presents the background and related work. Section 3 describes the research methodology. Section 4 presents the results and analysis. Section 5 presents the discussions of the result. Lastly, the paper is summarized and concluded in Section 6.

2 Background and Related Work

Organizations can be conceptualized as entities that interact with an environment through inputs, conversions, and outputs (Ilgen et al., 2005; Jones, 2012), as shown in Figure 1. The environment refers to the entities that provide inputs and consume the outputs generated by the organization. The environment includes customers, legislation, suppliers, competitors, and other stakeholders. Inputs refer to the things that the organization obtains from the environment to produce value such as raw materials, human resources, and information. Conversions refer to the constituents that the organization requires to transform the inputs into something of value. For an IT organization to be flexible its constituents need to be built for flexibility. These constituents are workforce, management, processes, organizational structure, and infrastructure (Tapanainen et al., 2008). Workforce refers to human resources and their skills and knowledge. Process refers to concerted activities that take place in the organization, for example a development process. Organizational structure refers to the organization hierarchy and roles and how they relate to each other. Infrastructure refers to technology platforms that are required to conceive, build, release, and evolve products and services. Outputs refer to the elements that the organization releases to its environment and include products, services, and value that are released to the environment.

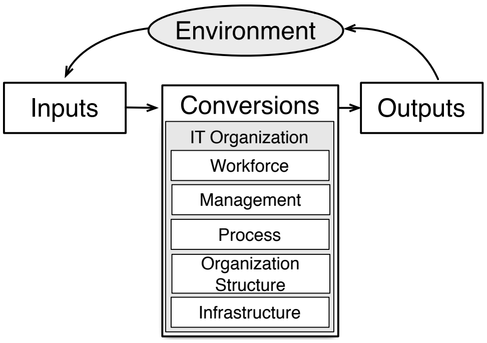

Figure 1: Organization Model Adapted from (Jones, 2012) and (Tapanainen et al., 2008)

Flexibility is a multidisciplinary concept that relates to the ability of an entity to change easily and effectively (De Toni and Tonchia, 1998). There are different definitions of flexibility depending on the context and domain (Sethi and Sethi, 1990). Across the different definitions of flexibility it can be summarized as the ability of an entity to change with: (1) effectiveness, timeliness, and satisfaction with a change (Kara et al., 2002), (2) balance of the amount of change and stability (Regev et al., 2006), (3) minimal difficulty, cost, time, effort, and risk of the change (De Toni and Tonchia, 1998; Huber and McDaniel, 1986; Sethi and Sethi, 1990), (4) and the universality of the entity expected to cope with variations of input (Knoll and Jarvenpaa, 1994).

Manufacturing recognizes that flexibility needs to be built from different aspects of the manufacturing processes, such as, material handling, production line, to marketing process, as well as management and organizational structure (Sethi and Sethi, 1990). Flexibility is required to cope with internal and external uncertainties (De Toni and Tonchia, 1998; Sethi and Sethi, 1990) and achieve competitive advantage (Boyle, 2006; Sethi and Sethi, 1990). To build flexibility, manufacturing literature suggests the importance of identifying the required flexibility derived from the uncertainties, implement the flexibility building decision, and perform audit of the planned and actual achieved flexibility (Boyle, 2006; De Toni and Tonchia, 1998; Gerwin, 1993). However, the intangible, changeable, complex nature of software and the unpredictability of development projects differs from manufacturing of physical goods (Brooks, 1987). Thus, approaches in manufacturing may not be easily transferable to IT organizations.

Literature in management has two perspectives of organization flexibility. The first perspective views organization flexibility as how quickly an organization can change its structure given the changes in the environment (Huber and McDaniel, 1986). This can be achieved by modifying people's roles and responsibilities (Aalst and Jablonski, 2000), or having a manager with openness to change (Sharfman and Dean Jr, 2003). The second perspective views organization flexibility as how the organization structure can remain stable given changes in the environment (De Haan et al., 2011). This can be achieved by employing highly skilled workers (Sethi and Sethi, 1990), and adopt a flat organization structure to allow quick problem resolution (Huber and McDaniel, 1986). Management literature has provided suggestions to achieve organizational flexibility, but there is no descriptive guidance on how to build flexibility.

The need for holistic view of flexibility is also expressed in IS literature. Byrd et al. (2010) proposed a framework of IS flexibility with four dimensions, people, IT, data, and process. Each dimension is associated with various factors that support its flexibility which in turn will support the organization flexibility. Allen and Boynton (1991) discuss two main approaches to achieve IS architecture flexibility, high-road, and low-road. Each approach has its own pros and cons. Thus organizations need to tailor them to their own contexts. Current solutions are still lacking descriptive approach that allows organizations to build flexibility that suits their needs and contexts.

Software engineering has offered approaches on how to build flexibility on the level of the software product. Software product line engineering (SPLE) is intended as an approach to deal with mass-customization of software products (Clements and Northrop, 2001). This is achieved through variability management where commonalities and variabilities of a set of software products' artefacts (e.g., requirements, codes, test cases, etc.,) are identified (Chen et al., 2009). Agile methodologies enable product flexibility through a process that welcomes frequent user requirement changes (Dingsøyr et al., 2012). Agile methodologies include practices that enable flexibility such as short iterations, collective code ownership, and refactoring (Abrahamsson et al., 2002). Short iterations allow changes to be integrated as soon as the need for change is discovered from customer feedback (Petersen, 2011). Collective code ownership increases knowledge sharing and supports flexibility by making it easier to assign people to tasks (Abrahamsson et al., 2002). Refactoring systematizes the actual change of a software system (Fowler and Beck, 1999).

Today, we lack an understanding of how to build flexible software organizations. Software engineering literature addresses product flexibility but does not do so on an organizational level. A holistic organizational coverage is missing. Manufacturing and IS literature confirms this need for a holistic view (Huber and McDaniel, 1986; Sethi and Sethi, 1990). Making one aspect more flexible might lead to other aspects requiring more control (Leeuw and Volberda, 1996; Sethi and Sethi, 1990). Without a holistic view, the trade-offs of flexibility cannot be thoroughly considered (Ferdows and Meyer, 1990; Harris et al., 2009).

3 Research Methodology

We studied an IT Department that serviced one of the business units of a Fortune Global 500 company operating in the financial sector. The company provides services like investment banking, asset and wealth management for private, corporate and institutional clientèle worldwide. The IT Department had around 200 developers who were responsible for developing, hosting, and maintaining the software solutions required by the business unit.

The IT Department had undergone changes in the past decade as part of their efforts in improving flexibility. With the assistance of the IT Department management representatives, we identified **change scenarios** that were (a) sufficiently representative and (b) covered sufficient kind of change variation that they encountered (Miles and Huberman, 1994). In Reorganization 1, the organization shifted from hierarchical to matrix organization. In Reorganization 2, the organization shifted from matrix to pool organization. Both Reorganization 1 and Reorganization 2 were intended to improve resource allocation flexibility. In RUP (Rational Unified Process) Introduction, the organization adopted RUP as the standard development methodology to improve resources flexibility to support different development projects. In Regulatory Change Implementation the organization modified its IT system(s) due to regulatory change that the business had to comply.

Together, these four scenarios covered all five constituents of an IT organization which were mentioned by Tapanainen et al. (2008). Reorganization 1 and Reorganization 2 influenced workforce, organization structure and management. RUP Introduction influenced the development process, and workforce's knowledge of RUP. Regulatory Change Implementation influenced the infrastructure of the organization, as the system becomes part of their infrastructure.

3.1 Data Collection and Analysis

In this research, we aim to reveal a holistic view on how flexibility is built or improved in an IT organization to cope with changes. To achieve our research aim, we adopted grounded theory as suggested by Charmaz (2014) to uncover the events that take place as an IT organization improves its organizational flexibility. The following research question was formulated to guide our study:

RQ1. How does an IT organization build or improve flexibility?

Grounded theory is a well-established research method used in software engineering, IT, and IS research (Coleman and O'Connor, 2007). Grounded theory allows us to develop concepts and theories of building flexibility based on the data, evidence and not through pre-conceived notions (Charmaz, 2014). Grounded theory allows us to remain open and critical to what flexibility building really is.

We performed data collection and data analysis iteratively. Data collection was done through semi-structured interviews and documentation reviews for data triangulation (Robson, 2011). The aim of semi-structured interviews is to get a holistic understanding of the situation in the organization, what changes had been faced, what kind of challenges were faced, what had been done to address the change, and how they were perceived. A preliminary list of open-ended questions was developed. These questions were continuously modified as we progressed with the interviews and concepts were emerging, as suggested in grounded theory (Charmaz, 2014). Documentations were collected to gather standards, policies, and lessons learned that are relevant to get an overview of the flexibility improvement efforts that have taken place in the organization.

We conducted interviews with nine managers in the case organization. Each interview lasted approximately 60 minutes, and they were recorded and transcribed. Prior to performing the interviews with the managers, we conducted interviews with three of the management representatives of the IT organization to get an overview of the organization and why it needed to improve its flexibility. They also provided an overview of recent changes that the organization had to face. From these preliminary interviews, we saw how the different changes influenced different constituents of the organization, as explained by Tapanainen et al. (2008). The representatives then provided a list of managers that we could interview pertaining to these changes, whilst ensuring that we covered wide variation of scenarios based on the impact of changes to different organization constituents.

We followed the coding steps in grounded theory as prescribed in (Charmaz, 2014). In **initial coding**, we adopted line by line coding. It was selected to allow us to see processes and distance us from the interviewees' perspective. We went through the interview transcriptions line by line and assign one code or more as they emerge from each line. From the open coding, we identified reccuring themes across interviewees and scenarios.

Then we performed **axial coding** to categorize the change scenarios based on the reccuring themes such as, triggering uncertainties, impact of the uncertainties to the organization, actions that were taken to address uncertainties, and their respective outcomes with respect to achieved flexibility and relevant trade-offs. The concepts that emerged from the axial coding can be seen in Table 1.

In **theoretical coding**, we analyzed the relationships between the categories. We observed that the categories that emerged from the axial coding were a series of events that took place as an IT organization improved its flexibility. The outcome of the theoretical coding is shown in Figure 4.

3.2 Threats to Validity

This research is subject to validity threats common to grounded theory, such as credibility, resonance, originality, and usefulness (Charmaz, 2014).

Credibility is compromised when the research performed did not have data with sufficient breadth and depth. Thus, the arguments are not strongly supported by the data. This was mitigated through having multiple interviewees and reviews of the organization's documentations. Credibility was also supported by having change scenarios that covered all five of the organization constituents.

Resonance refers to whether the research performed was able to capture the actual experience. We addressed this threat by informing our interviewees of our presence, the purpose of our study, and also the kind of information we would like to elicit from them. Interviewees also participated on their own will and interest in the study.

Originality is a risk that the outcome of the study did not offer new insights of flexibility building. The study that we conducted was built on top of existing work on flexibility from different domains. In this study, we uncovered how the existing knowledge of flexibility building is applicable for an IT organization. Furthermore, in this study we uncovered a series of events in building flexibility.

Usefulness is a risk to the applicability of the results beyond the studied setting. To improve the usefulness, we selected scenarios that covered the different constituents that were identified byTapanainen et al. (2008) which should be present in other IT organizations. We believe that reorganizations, adoption of a new development method, and implementation of a new regulation are typical scenarios in other IT organizations supporting a financial institution.

4 Results and Analysis

We uncovered key concepts in flexibility building that were present in all four scenarios (Reorganization 1, Reorganization 2, RUP Introduction, and Regulatory Change Implementation). The scenarios, flexibility building concepts, and interviewees' statements (marked S1 – S43) are summarized in Table 1. Flexibility building needed a key person or a group of people who initiated the process. We referred as *initiator*. *Situation* referred to the condition of the organization prior to the modification of the constituent. Given some internal or external changes, the organization faced variations or unclear inputs (budget, information, or regulation) referred as *uncertainty*. *Need for flexibility* was the problems or inefficiencies that emerged in the organization due to the uncertainty. *Flexibility building* referred to the modifications made to the specific organization constituent. *Achieved flexibility* referred to the situation after the organization constituent modification and how flexibility was improved. *Trade-off* referred to the identified disadvantages of the selected flexibility building option. The following subsections describe how these concepts contribute to flexibility building of an IT organization.

Table 1: Key Concepts and Relevant Interviewees’ Statement

	Reorganization 1	**Reorganization 2**	**RUP Introduction**	**Regulatory Change Implementation**
Initiator	S1:“[The CIO] was a change manager”	S10:“[The CIO] decided to merge them in one big pool organization.”	S19: “It was from internal governance.”	S28: “It was my [program manager] decision what can be descoped.”
Situation	S2: “We had some groups, and each group was responsible for a product and the people.”	S6	S20: “We had [..] templates, but not really a methodology.”	S29: “There is nothing [regulated service system] on the market available.”
Uncertainty	S3:“Years ago we were really stable in the budgeting [..]. Now, [..] sometimes they like to invest on [application A], a year later they like to invest [..] on a different topic.”	S9	S21:“There was no common way of working, [..] how they work was different from project to project	S30: “The bilateral agreement between [country A] and [country X], [country Y], and [country Z].” S31: “This is a complete[ly] new agreement, [..] nobody really had any experience.” S32:“Unfortunately, [Country X] jumped off few days before going live.”
Need for flexibility	S4: “If [..] an employee [..] would like to move [his] manager can stop it because they did not like to have [..] moves to the other department.” S5: “[The group] SAP implementation had 180 people. After a year you do not really need the [rest] 60 people, but these 60 people could not be relocated, and they did not have anything to do.”	S9	S22: “There was a problem that each project work in their own ways.” S23:“we wanna standardized the way of working so we can support each other.”	S33: “[These] regulatory things you have to deliver, there is no choice.” S34: “[..] to become a compliant financial institution of 2012.”
Flexibility building	S6: “We had a matrix with [..] four pools: project management, business analysis, development, and integration. Across we had the key accounts, and we had the head of this unit called Service Provider Unit.”	S11: “We are organized in the delivery unit in five pools: [..] project leads, business analysts, and we have [two] development pools.” S12: “So if we have some escalations between these two organizations [pools], then you have to go to [the CIO].”	S24: “[..] all projects are running RUP.”	S35: “The software is ready [..].” S36: “We [..] did it on a make [build] approach in our infrastructure.” S37: “We [..] had chosen the option which has the fewest impact on all of the system.”
Achieved flexibility	S7: “People had the possibility to move around, to grow, to have a good future perspective.” S8: “We [the staff] had more power to switch [..] to other solutions.”	S13: “The pool setting offers flexibility to cope with demand from the business. You can focus people on a solution whenever required.” S14: “Whatever the business decides, you can allocate the people to that particular solution.” S15: “Here [pool organization], the responsibilities are clear[er]. You have one organization responsible for product, one organization responsible for deliveries [..].”	S25: “Now we all have the same terminology [..].” S26: “If a new architect coming to your team and knows how you work.”	S38: “We prepared the software in a way we could switch on or off a country.” S39: “In the end the system will fit in our environment.” S40: “We can steer the whole process.”
Trade-off	S9: “He [a manager] could decide himself on an escalation view. The problem [was] all these three SPUs [Service Provider Delivery Units] had small differences in the process.”	S16: “People miss having their home and may be unmotivated.” S17: “Knowledge management is much more complicated [..]. Something gets lost when [people] go away from a solution domain.” S18: “With the pool I [project manager] need time to find people.”	S27: “At the beginning they thought that is [..] just a window to their project.”	S41: “[the option] might not be the best for the clients because clients will get two confirmations out of one transaction” S42: “It [was] not cheaper.” S43: “You’re the only one who fulfil it with a huge investment.”’

4.1 Uncertainty and Need for Flexibility

The initiation of flexibility building was triggered by emergence of uncertainty that the organization could not cope. Such uncertainty led to the need for flexibility.

In **Reorganization 1**, the organization was in hierarchical structure (S2), Figure 2 left. A problem emerged related to the financial institution's budget reduction that led to uncertainty in the focus of the development projects (S3). The need for flexibility emerged as the hierarchical structure of the organization hindered team members of one development section to be reallocated to another one (S4). Preferred staffs were running idle (S5).

In **Reorganization 2** the organization was in matrix structure (S6), Figure 3 left. A problem of inconsistent processes across development units or Service Provider Units (SPUs) emerged due to unpredicted outcome from Reorganization 1. The inconsistency led to uncertainty in the process of project issues escalation (S9). The differences in escalation process resulted in confusion and delays.

In **RUP Introduction**, initially the organization did not have a development methodology (S20). The problem was related to inconsistent development methodology that led to uncertainty on how a project would be performed (S21). This uncertainty limited the possibility to allocate staff into different projects (S22,S23), as they might require time to learn how the other project worked.

In **Regulatory Change Implementation**, new bilateral agreements between financial institution's home country were planned with other countries (S30). There were no IT systems that supported such bilateral agreements (S29). The contents of these agreements varied across countries (S30), and nobody knew the exact rules that were going to be agreed (S31). Furthermore, uncertainty escalated as there was a possibility that a country could withdraw from the agreement (S32). This uncertainty led to a difficulty in determining what services need to be provided for every bilateral agreement, and if they need to be provided at all. This led to the possibility of performing unnecessary work.

From the four scenarios, we learned that an uncertainty could originate from outside the organization, like changes in the business environment, as shown in Reorganization 1 and Regulatory Change Implementation scenarios. An uncertainty could also originate from within the organization and not necessarily caused by changes in the environment, as shown in Reorganization 2 and RUP Introduction. The uncertainty limited the software development organization's ability to operate efficiently or lead to additional or unnecessary work. These limitations then led to the need for flexibility.

4.2 Build Flexibility

Once the need for flexibility was identified, an IT organization built flexibility through modifying one or more of its constituents. The role of a decision-maker was crucial for initiating the required modifications.

In **Reorganization 1**, there was a need for more flexible resource allocation to high priority projects (S4,S5). This rigidity of resource allocation led the Chief Information Officer (CIO) to initiate a reorganization (S1). The option was to shift from hierarchical organization to a matrix organization structure, as shown in Figure 2, where staffs were grouped according to their roles (S6) and not by product development. The new organization structure should accommodate team members' mobility to be easily allocated into different projects.

In **Reorganization 2**, there was a need to overcome the inconsistency of escalation process across SPUs (Q9). The CIO, again, opted to initiate a reorganization (S10). The inconsistency needed to be addressed by having a more centralized escalation process. The option was to shift from was to shift from matrix to pool organization structure, as shown in Figure 3. In the pool organization, the SPUs were merged into different pools, and all pools only reported to the CIO (S11). The centralized reporting should eliminate variations in issue escalation process.

In **RUP Introduction**, there was a need for flexible resource allocation. It was hindered due to variation of development process across projects. The internal governance group decided that a common way of working was required to improve flexibility (S19). The option to address the inflexible resource allocation was by adopting RUP as a standard organization development methodology (S24). A common way of working and terminology should ease communication between staff and transitions between projects.

In **Regulatory Change Implementation**, there was a need for flexibility in IT service provision to support the new bilateral agreements (S30,S32), as well as to remain compliance with the regulation (S31). For this scenario, we uncovered different options on how to build flexibility of the IT service.

First option was whether to *build* or *buy* the regulated service system. The program manager (S28) of the implementation program decided on *build* approach (S36). Prior to executing the decision, the program manager had foreseen the pros and cons of building the IT system in-house. The *build* approach would require less integration effort (S39), and but it would be more expensive (S42).

There was also an option to build the system with high maintainability or high usability. Prior to executing the decision, the project manager had also seen the pros and cons of each option. To have better maintainability, the regulated service system would be decoupled from the rest of the IT system (S37). However this option would sacrifice usability for the clients (S41). It was decided to go with the option that prioritized maintainability over usability.

There were different decision-makers involved in deciding how the constituents were to be modified to build flexibility. They could be from upper level management or middle level management. Uncertainty that involved the whole organization, as in Reorganization 1, Reorganization 2 was handled at executive level, the CIO. The RUP Introduction was initiated by the internal governance responsible for development processes. Regulatory Change Implementation was handled by the program manager of the development program. Furthermore, the need for flexibility influenced which constituents to be modified and how they were modified. Unlike the other scenarios, in Regulatory Change Implementation, the decision-maker was able to foresee the outcome of the different options in improving flexibility.

4.3 Achieved Flexibility and Trade-Offs

After the constituents were modified, the organization achieved the flexibility that they required. However, there were trade-offs associated with the achieved flexibility.

In **Reorganization 1**, the matrix organization structure delivered flexibility in resource allocation. The team members were able to move different solution development or projects (S7,S8). The freedom for team members to move across projects also allowed them to improve their career perspectives in the organization (S7).

However, this new flexibility of resource allocation had a trade-off at the management level. The separation of the organization into different SPUs led to inconsistent issue escalation processes across the SPUs (S9). The variations of issue escalation process led to delays and confusions, and led the CIO to initiate Reorganization 2 (S10). This suggests that the decision-maker could not anticipate the trade-off of resource allocation flexibility from the reorganization to matrix organization.

In **Reorganization 2**, the pool organization solved the problem of project issue escalation, as all pool heads only reported project issues to the CIO (S12). The pool organization also strengthened the flexibility of resource allocation. As team members are no longer permanently designated to a particular product, they could be assigned to projects or initiatives which the business really needed (S13,S14). The pool organization also established clearer responsibility for the team members and managers (S15).

The pool structure allowed team members to be allocated into different projects which eliminated a sense of belonging to a group (S16). There was loss of product knowledge as people were not designated to a particular product anymore and moved between projects (S17). More time was required to find the right people to source projects (S18) because people were spread out in different pools. Decision-makers initiated a mitigation approach to mitigate loss of knowledge problem through sharing of lessons learned. As expressed by one of our interviewees: [*"We have some specialists groups, [..] then the group will communicate this lessons learned to the whole organization."*]

RUP Introduction improved communication and flexibility of the team members to be assigned in different development projects. However, the introduction of RUP practices like project milestones and

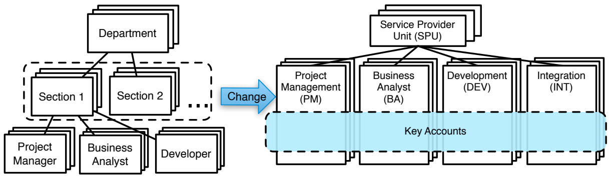

Figure 2: Reorganization 1: Shift from hierarchical to matrix organization

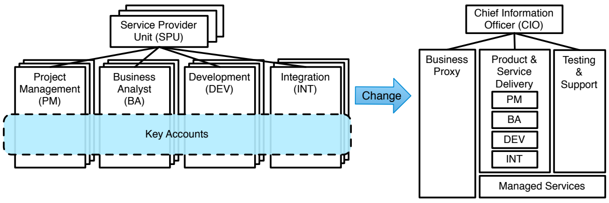

Figure 3: Reorganization 2: Shift from matrix to pool organization

mandatory documentations, made team members felt like they were being watched (S27). This trade-off only occurred in the beginning of RUP adoption. Thus no mitigation actions were initiated.

The result of the **Regulatory Change Implementation** was the deployment of a regulated service system into the infrastructure of the organization (S35). The decision to build the system in-house (S36) minimized the integration effort (S39), and allowed the project team to steer the development process (S40). Furthermore, the decision allowed the design of the infrastructure to be flexible in handling uncertainties of the bilateral agreement (S36). However, the decision to build the system was more expensive (S42) and required a large investment (S43). The decision to prioritize maintainability resulted in a regulated service system which was decoupled from the other IT systems (S37). However the decision compromised the system's usability (S41).

The selected option to modify a constituent allowed the organization to achieve the flexibility required to address the uncertainty. However, each option to improve flexibility could have other negative influence on other constituents. In the case of Reorganization 1, Reorganization 2, and RUP Introduction, it was difficult to properly foresee the outcome of the flexibility building option. However, in the case of Regulatory Change Implementation the trade-offs were consciously made by the decision maker.

4.4 Flexibility Building Model: Revisit RQ1

Using interview data from nine IT managers and grounded theory our research explored how an IT organization improved or built flexibility through strategic change of one of the IT organization constituents. Our research uncovered three states that the organization went through as part of the strategic change: pre-change, transitional state, and post-change. Pre-change was the state prior to the strategic change. Transitional state was the state when the constituent modification was taking place. Post-change was the state after the strategic change was in place. If another strategic change was required, the post-change state would be the pre-change state.

Through these transitional states in the organization, our data show that there were different events that took place in each state, emerging uncertainty, need for flexibility, build flexibility, and achieved flexibility and trade-offs. Across four distinct change scenarios, we observed that the uncertainty emerged in the pre-change state of the organization. The uncertainty triggered the need for flexibility. During transitional state, flexibility was built by modifying one or more of the organization constituents to address the uncertainty. In the post-change state, after the modification was implemented, flexibility was achieved yet with associated trade-offs. Furthermore, as demonstrated in Reorganization 2, flexibility building process could be cyclical, if the associated flexibility trade-off introduces new uncertainty, another round of flexibility building could be initiated.

The result suggests a model that elaborates the different events that took place in the transitional states as shown in Figure 4. The model does not detail the decision making points that managers need to take to build flexibility. However, it identifies and describes the events that took place as an IT organization built flexibility of its different constituents. The model offers a unified view of flexibility building for an IT organization, from ad hoc and latent experience to a well-constructed and explicit point of view.

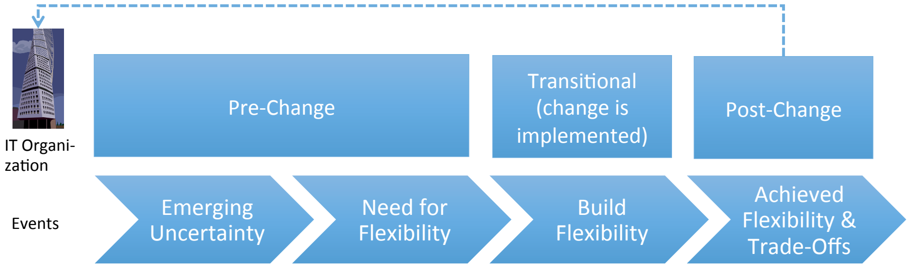

Figure 4: Events in Flexibility Building

Furthermore, we discovered that a flexible organization was not an organization that could quickly make changes to its structure every time it faced an uncertainty. Yet, a flexible organization was an organization that was able to restructure its organization constituents in such a way that they allowed the organization to withstand a specific type of uncertainty.

5 Discussion

This research has discovered the key events that took place as an IT organization built or improved its flexibility. Our data show that building organizational flexibility was done through modifying one or more of the IT organization's constituents to address the uncertainty that exposed the organization's need for flexibility. Through the constituents modification, the organization was able to achieve flexibility and this aided efficient response to changes and uncertainty, and with little additional effort. Our study has provided a holistic explanatory view of the processes in improving organizational flexibility.

Manufacturing literature suggests that building flexibility entails identifying uncertainty, implementing the appropriate decision, and monitoring the achieved flexibility(Boyle, 2006; Gerwin, 1993). Similar view is also shared in organizational change management literature (Orlikowski, 1993). The results of our analysis show similar pattern with existing literature in flexibility and change management. However, our study suggests that a flexible organization is not the one that is capable of constantly change the structure of its organization constituents on the face of uncertainties. Organizational change is one of the tools to build organizational flexibility. Building organization flexibility is about purposefully restructure of the organization constituents so the organization is able to cope with uncertainties. Further research could be allocated on looking at known approaches in organizational change management to complement the current model that was developed in this study.

In software reusability management, reuse can be used from two perspectives "developed for reuse" and "developed with reuse" (Börstler, 1992). Developed for reuse is when software artefacts like documentations and codes are predefined and purposefully developed for later reuse usually by implementing well known design principles like encapsulation, coupling, and cohesion. Developed with reuse deals with the use of the previously predefined artefacts. The data in our study shows that to achieve flexibility organization constituents could also be "developed for flexibility" and "developed with flexibility". In Reorganization 1, Reorganization 2, and RUP Introduction, the organization structure and development process were developed for flexible resource allocation to different projects, whilst each initiated project was developed with flexibility of the organization structure and development process. In Regulatory Change Implementation the infrastructure was developed for flexibility of providing financial services for different bilateral agreements, whilst each supported bilateral agreement was developed with flexibility of the infrastructure. This shows that software engineering has mechanisms for improving flexibility that are applicable at the organizational level, but currently are used at product level. Future research can be directed towards identifying mechanisms and design principles that could be transferred from software artefacts to organizational level to build organizational flexibility.

Real options thinking is often suggested for managers to cope with uncertainty (Fichman et al., 2005). Uncertainty can be addressed by choosing an option. One of these options is *defer* where decisions are delayed for a certain period of time. Our study shows that modifications of organization constituents allowed managers to exercise the *defer* option. In Reorganization 1 and Reorganization 2, the decision to change the organization structure, allowed them to defer the decisions which team members to be allocated to a certain development effort until a project was funded. In the Regulatory Change Implementation scenario, the decision to design the infrastructure to cope with variation of bilateral agreements, allowed them to defer decisions on which services to activate or deactivate. Future research on real options thinking can be pursued to get insights on how to develop criteria to help managers to select the best option in improving organizational flexibility.

Previous studies suggest that flexibility is a trade-off with values, such as control (Harris et al., 2009), performance (Ferdows and Meyer, 1990), and cost (Van Biesebroeck, 2007). Our study shows that flexibility is not necessarily an immediate trade-off with other values. Scenarios clearly show that flexibility trade-offs were the indirect effects of the selected option for the strategic change. For example, in Regulatory Change implementation, the option to build the system in-house was more costly, nevertheless it was preferred over purchasing a system that required large integration effort. In Reorganization 2, ease of team member allocation to projects was preferred over quick development project completion. Our study shows that different options could lead to different trade-offs, however these trade-offs were acceptable given the achieved flexibility. Further research needs to be done to help managers predict and assess the trade-offs of the different options in improving organizational flexibility.

One of the practices in Agile methodologies is iterative development (Abrahamsson et al., 2002). Previous studies have shown that iterative development allows a software organization to reduce rework, improve estimations, and reduce lead time (Petersen and Wohlin, 2010). Our study also shows that flexibility improvement is done iteratively. This was clearly demonstrated in Reorganization 1 and Reorganization 2. In Reorganization 1, the organization structure shifted from hierarchical to matrix organization structure to deal with rigid resource allocation. Despite the achieved flexibility of resource allocation, the new organization structure introduced unanticipated inconsistencies into the issue escalation process. The inconsistency initiated a new cycle of flexibility improvement through Reorganization 2. Our study shows that the iterative nature of achieving organizational flexibility is congruent with the iterations in Agile methodologies. Furthermore, our study shows that reflecting the outcome of a strategic change was an important process for a software organization in achieving flexibility. Future research could be directed to seek other key ideas from Agile methodologies for improving organizational flexibility.

Current approaches in software engineering, like Agile methodologies (Beck, 2000) and SPLE suggest (Clements and Northrop, 2001) that flexibility is built through the products. Our study shows that organizational flexibility could be built into different organization constituents, not only built through the products. Reorganization 1, Reorganization 2, and RUP Introduction show that modifications in organization structure, management, and development process could have influence on the flexibility of project resource allocation which in turn improved the IT organization's flexibility to initiate projects that suited the business needs. Regulation Change Implementation shows that the infrastructure design improved the IT organization's flexibility in coping with what the business requires in terms of providing services for different bilateral agreements. This shows that there are wider range of possibilities how an IT organization can be improve its flexibility than previously assumed and further confirmed the need for a holistic approach to flexibility (De Michelis et al., 1998) not only through products.

5.1 Implications for Practice and Research

Our research suggests a model that provides a holistic view of how an IT organization improves flexibility. This model was uncovered through analyzing the experiences of IT managers in the case organization and has not yet been operationalized. However, the findings of our study yield implications both for researchers and practitioners.

The key events that our study uncovered can be used by practitioners as a foundation for process improvement. For example, flexibility could be added as a process area to existing frameworks such as CMMI. Such a process area would include the identification of uncertainty, the prioritization of which kind of flexibility should be built, the selection of the appropriate flexibility improvement approach, and the analysis of the obtained results. Also, such a process should specify how experiences from flexibility improvement would be collected and shared to support future decisions.

Flexibility improvement processes would need to be supported by suitable tools that enable process institutionalization. For example, the uncertainties that need to be addressed by the organization could be captured in a backlog. Backlogs are an established and commonly used tool in Scrum to enable visibility and prioritization of features that a product needs to deliver (Schwaber and Beedle, 2001). A backlog of uncertainties would enable sharing of flexibility improvement needs, prioritization, and progress tracking, thus allow the organization to manage flexibility proactively and systematically.

Also, knowledge and experience sharing may improve the organization's ability to build flexibility. Knowledge sharing is an essential part of organizational learning that has led to performance improvement in many companies (Garvin, 2000). Here, knowledge sharing should enable the identification of flexibility-building options and improve the understanding of the options' impacts on flexibility and trade-offs. Such knowledge could be expressed as stories or patterns (Hagge and Lappe, 2005). Their use would reduce the risk of unanticipated negative surprises and minimize the need to re-iterate.

Our study provides a stepping stone for future research in understanding how organizational flexibility is built. To support organizations in improving flexibility, a systematic mapping of flexibility-building options with their associated benefits and trade-offs is needed. Experience transfer and evaluation from domains other than software engineering, like management and manufacturing, can be valuable inputs to build such a mapping. Management literature provides suggestions to build flexibility, for example to adopt a flat organization structure (Huber and McDaniel, 1986) and having managers that are open to change (Sharfman and Dean Jr, 2003). Manufacturing literature describes mechanisms and measures to build flexibility through suitable material handling and production process design (Sethi and Sethi, 1990). Concerns that are unrelated to software need to be replaced, however. Nevertheless, understanding the practices of other established domains can prevent "reinventing the wheels".

To formulate a model that maps flexibility-building options and their associated benefits and trade-offs, we can also pursue experience transfer from existing product-level approaches in software engineering to organizational level. Our study shows that organization constituents could be "developed for flexibility" and "developed with flexibility". Also, our study shows that flexibility improvement was done iteratively. These are known concepts from software reusability management (Börstler, 1992) and Agile methodologies (Abrahamsson et al., 2002) with an abundance of empirical studies. Using existing empirical studies from such established approaches can allow researchers to build a map of flexibility-building options and their associated advantages and disadvantages.

6 Summary and Conclusion

We adopted grounded theory as a research approach for understanding how an IT organization improved flexibility. We studied four scenarios of flexibility improvement that our industry partner considered representative for an IT organization that had evolved over the past decade. Data was collected by interviewing the respective decision-maker about the strategic flexibility-related change he was involved in. The scenarios suggest that decision-makers have a central role in managing IT organization flexibility. They are the key authoritative figures that allow which changes to be made in the pursue of flexibility. Furthermore, the selected course of actions that the decision-makers take can yield to different flexibility trade-offs.

The research resulted in a model that describes events that take place as an IT organization improves its flexibility. The model suggests that flexibility is built to answer uncertainty in the organization that can cause delays, unnecessary work, and inefficient use of resources. Furthermore, flexibility can be built into different constituents of the organization, and not just through the product. Future research should be directing towards improving the understanding of how to select the course of action that is best suited to address uncertainty, and better predict flexibility trade-offs.

Acknowledgement

We would like to thank our anonymous industry partner for their support and cooperation.

References

Aalst, W. Van der and S. Jablonski (2000). “Dealing with Workflow Change: Identification of Issues and Solutions.” *Computer Systems Science and Engineering* 15 (5), 267–276.

Abrahamsson, P. et al. (2002). *Agile Software Development Methods – Review and Analysis*. Tech. rep. 478. VTT Publications.

Allen, B. R. and A. C. Boynton (1991). “Information architecture: in search of efficient flexibility.” *MIS quarterly*, 435–445.

Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley Professional.

Börstler, J. (1992). *Feature Oriented Classification and Reuse in IPSEN*. Tech. rep. Aachen University of Technology.

Boyle, T. (2006). “Towards best management practices for implementing manufacturing flexibility.” *Journal of Manufacturing Technology Management* 17 (1), 6–21.

Brooks, F. P. (1987). “No Silver Bullet: Essence and Accidents of Software Engineering.” *IEEE Computer* 20, 10–19.

Byrd, T. and D. Turner (2000). “Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct.” *Journal of Management Information Systems* 17 (1), 167–208.

Byrd, T. et al. (2010). “An Examination of an Information Systems Flexibility Framework.” In: *Proceedings of the 43rd Hawaii International Conference on System Sciences*, pp. 1–10. DOI: 10.1109/HICSS.2010.52.

Charmaz, K. (2014). *Constructing Grounded Theory*. 2nd Edition. Sage Publications Limited.

Chen, L. et al. (2009). “Variability management in software product lines: a systematic review.” In: *Proceedings of the 13th International Software Product Line Conference*. SPLC ’09. San Francisco, California: Carnegie Mellon University, pp. 81–90.

Clements, P. C. and L. Northrop (2001). *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley.

Coleman, G. and R. O’Connor (2007). “Using grounded theory to understand software process improvement: A study of Irish software product companies.” *Information and Software Technology* 49 (6), 654 –667.

De Haan, J. et al. (2011). “Framing Flexibility: Theorising and Data Mining to Develop A Useful Definition of Flexibility and Related Concepts.” *Futures* 43 (9), 923 –933.

De Michelis, G. et al. (1998). “A three-faceted view of information systems.” *Communications of the ACM* 41 (12), 64–70.

De Toni, A. and S. Tonchia (1998). “Manufacturing Flexibility: A Literature Review.” *International Journal of Production Research* 36 (6), 1587–1617.

Dingsøyr, T. et al. (2012). “A Decade of Agile Methodologies: Towards Explaining Agile Software Development.” *Journal of Systems and Software* 85 (6), 1213 –1221.

Ferdows, K. and A. D. Meyer (1990). “Lasting Improvements in Manufacturing Performance: In Search of a New Theory.” *Journal of Operations Management* 9 (2), 168 –184.

Fichman, R. G. et al. (2005). “Beyond Valuation: "OPTIONS THINKING" IN IT PROJECT MANAGEMENT.” *California Management Review* 47 (2), 74 –96. ISSN: 00081256.

Fowler, M. and K. Beck (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional.

Fricker, S. and S. Schumacher (2012). "Release planning with feature trees: Industrial case." In: *Requirements Engineering: Foundation for Software Quality*. Springer, pp. 288–305.

Garvin, D. A. (2000). "Building a learning organization." *Harvard Business Review)* 71 (4), 78–91.

Gerwin, D. (1993). "Manufacturing Flexibility: A Strategic Perspective." *Management Science* 39 (4), 395–410.

Hagge, L. and K. Lappe (2005). "Sharing requirements engineering experience using patterns." *Software, IEEE* 22 (1), 24–31.

Harris, M. et al. (2009). "Control of Flexible Software Development Under Uncertainty." *Information Systems Research* 20 (3), 400–419.

Huber, G. and R. McDaniel (1986). "The Decision-making Paradigm of Organizational Design." *Journal of Management Science* 32 (5), 572–589.

Ilgen, D. R. et al. (2005). "Teams in Organizations: From Input-Process-Output Models to IMOI Models." *Annual Review of Psychology* 56 (1). PMID: 15709945, 517–543. DOI: 10.1146/annurev.psych.56.091103.070250.

Jones, G. R. (2012). *Organizational Theory, Design, and Change*. Prentice Hall.

Kara, S. et al. (2002). "The Role of Human Factors in Flexibility Management: A Survey." *Human Factors and Ergonomics in Manufacturing and Service Industries* 12 (1), 75–119.

Knoll, K. and S. Jarvenpaa (1994). "Information Technology Alignment or "fit" in Highly Turbulent Environments: The Concept of Flexibility." In: *Proceedings of the Computer Personnel Research Conference on Reinventing IS*. ACM, pp. 1–14.

Leeuw, A. D. and H. Volberda (1996). "On the concept of flexibility: A dual control perspective." *Omega* 24 (2), 121 –139.

Miles, M. B. and A. M. Huberman (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. Sage Publications.

Orlikowski, W. J. (1993). "CASE tools as organizational change: Investigating incremental and radical changes in systems development." *MIS quarterly*, 309–340.

Petersen, K. (2011). "Is Lean Agile and Agile Lean?" *Modern Software Engineering Concepts and Practices: Advanced Approaches, IGI Global*, 19–46.

Petersen, K. and C. Wohlin (2010). "The Effect of Moving From a Plan-driven to an Incremental Software Development Approach with Agile Practices." *Empirical Software Engineering* 15 (6), 654–693 (6). ISSN: 1382-3256.

Regev, G. et al. (2006). "Taxonomy of Flexibility in Business Processes." In: *Proceedings of the 7th Workshop on Business Process Modelling, Development and Support*, pp. 90 –93.

Robson, C. (2011). *Real world research*. 2nd. West Sussex, UK: John Wiley & Sons, p. 608.

Schwaber, K. and M. Beedle (2001). *Agile Software Development with Scrum*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR. ISBN: 0130676349.

Sethi, A. K. and S. P. Sethi (1990). "Flexibility in Manufacturing: A Survey." English. *International Journal of Flexible Manufacturing Systems* 2 (4), 289–328 (4). ISSN: 0920-6299.

Sharfman, M. and J. Dean Jr (2003). "Flexibility in Strategic Decision Making: Informational and Ideological Perspectives." *Journal of Management Studies* 34 (2), 191–217.

Tapanainen, T. et al. (2008). "Towards an Agile IT Organisation: A Review of Prior Literature." In: *Proceedings of the 2nd European Conference on Information Management and Evaluation*, pp. 425 –432.

Todnem By, R. (2005). "Organisational change management: A critical review." *Journal of Change Management* 5 (4), 369–380.

Van Biesebroeck, J. (2007). "The Cost of Flexibility." *Assembly Automation* 27 (1), 55–64.