INERTIA AND CHANGE IN TRANSFORMATION OF THE IT-FUNCTION IN LARGE ORGANIZATIONS: A PATH THEORY LENS

Knut H. Rolland, University of Oslo and Kristiania University College

Kathrine Vestues, NTNU

Abstract:

Digitalization and digital transformation of large incumbent organizations, more often than not, require substantial changes to how the IT-function conducts and sources software development. This, however, often involves battling strong organizational and technical inertia making radical changes hard to implement. Yet, radical changes are often a prerequisite for digitally transforming the firm. In this regard, we present a longitudinal case study of a large governmental organization that over a relatively short period managed to radically change their IT-function in spite of strong inertia related to its software development methods and practices, sourcing arrangements, and legacy systems. In explaining this, the paper presents two contributions. First, we build on path theory to explain how IT organizations can develop organizational inertia in terms of path dependence related to software development methods, souring models, software delivery routines, and legacy systems. Second, we identify three interdependent mechanisms of breaking out of such path dependency: use of digital platforms for software development, attracting and securing competence, and establishment of cross-disciplinary software development teams. The paper concludes by giving some directions to further Information Systems (IS) research on this topic.

1. INTRODUCTION

In both practitioner-oriented and research it is often emphasized that new digital technologies and solutions provide organizations with a substantial potential for transformation. Especially technologies like Internet of Things, Big data analytics, digital platforms, machine learning, and combinations of these are typically expected to have transformational effects. Recent IS literature on digital transformation confirms that under certain conditions when organizations maintain a capability to change and utilize such technologies in their particular contexts, they may produce novel solutions and innovation (Jonsson et al., 2018; Svahn et al., 2017). In fact, much literature reflects a fundamental assumption that digital technologies are different in the sense that they are more malleable and evolvable than their more physical counterparts (Arvidsson and Mønsted, 2018). At the same time, however, many incumbent firms and organizations are also struggling to transform their business models and ways of working. Incumbent firms are often limited by legacy systems that are hard to discontinue (Mehrizi et al., 2019), current ways of sourcing IT (Law, 2018), and transformation towards more agile ways of working can to be long-drawn and paved by uncertainty (Dikert et al., 2016). Given these typical circumstances in incumbent organizations, it is likely that the way that the IT-function is organized and the way it operates is important for the organization's capability to undertake digital transformation.

This article posits that fundamental changes in the way that the IT-function is organized and practiced are necessary prerequisites for successful digital transformation and digitalization to take place. To put it bluntly, the IT-function requires transformation, in order to be able to facilitate digital transformation of the firm. However, besides studies on the normative models of the IT-function (Guillemette and Paré, 2012) and numerous studies of various IT governance models (Gregory et al., 2018), there are few empirical studies that investigates the actual transformation process of the IT-function in incumbent organizations in contexts of digital transformation. Hence, the main objective in this paper is to investigate to what extent current organizing and practices of the IT-function produces different forms of technical and organization inertia and how such inertia can be broken in order to promote transformation and change. In so doing, we build on insights from path dependence theory (Sydow et al., 2009) in order to theorize a specific form of inertia – namely path dependency and explain why intended changes typically are so hard to implement. More accurately, we conceptualize path dependency of the IT-function as an unintended cause of historical and contingent decisions regarding IS development methods, delivery models, sourcing strategies, and organizing, that over time generate self-reinforcing dynamics that reinforce rather than transform existing ways of working. While path dependency theory has been used in various forms and ways in previous IS research (e.g. Mehrizi et al., 2019; Sing et al., 2015), we are primarily interested in explaining how path trajectories can be broken, and the specific mechanisms and practices that are likely to trigger transformation of paths. In contrast to Sydow et al. (2009) who favors organizational and managerial practices as sources for path breaking, we see path dependency in context of IT-organizations as socio-technical, and hence that mechanisms for breaking a path can be both technical and organizational, as well as their interaction.

Empirically, we present a longitudinal case study of an IT organization in a governmental organization in Norway. Our case organization had for many years struggled with limited progress in digitally transforming itself and to digitalize its service for citizens. Our analysis reveals that changing the IT organization at the government organization were challenging due to the inertia of large projects and outsourcing of software development, a complex digital infrastructure making deployment of new software risky, and a lack of IT and development competence. Against all odds, however, after a major failure of a large software development programme in 2012-14, the government organization managed to turn around its IT organization and establish cross-disciplinary teams, deliver new software continuously, and re-organizing and excising outsourced projects. As a consequence, the government organization has been able to digitally transform a much larger part of their citizen services than before.

Grounded in this analysis the paper suggests that although IT organizations are likely to struggle with inertia and in particular path dependency it is possible to break out of exiting trajectories and transform themselves. Henceforth we theorize that in context of IT organizations that are characterized by outsourcing of software development, legacy IS, and few and big deliverables, some practice seems to be more likely than others to break path dependency. In particular the paper elaborates on three mechanisms for breaking path dependency in such context that relates to: 1) use of digital platforms in software development; 2) attract new competence; and 3) establishment of cross-disciplinary teams.

The paper is structured in the following way. First, we review relevant literature concerning inertia and change of IT organizations, and then we in Section 2.2. develop a framework for conceptualizing inertia in IT organizations as path dependency and mechanisms for breaking with paths. Section 3 covers the description of the case study as well as some brief insights on data collection and analysis. Furthermore, in Section 4 we present the case narrative. Next, in the concluding section, we discuss our contributions and some concluding remarks for further research on the topic of inertia and change in IT organizations.

2. BACKGROUND LITERATURE AND THEORY

2.1 Inertia and change in IT organizations in context of digital transformation

The concept of *digital transformation* reflects a substantial change in the role that digital technologies play for individuals, organizations and the society as a whole. In a recent paper by Hinings et al. (2018: p. 52) relates digital transformation explicitly to digital innovation, practices, at various levels of analysis, by defining it as: "*the combined effects of several digital innovations bringing about novel actors (and actor constellations), structures, practices, values, and beliefs that change, threaten, replace or complement existing rules of the game within organizations, ecosystems, industries or fields*". In this paper, we comply to Hinings et al.'s definition of digital transformation, as a context for what organizations are trying to achieve through their IT-function.

As much as digital transformation implies radical changes in terms of both technologies and organization, inertia has been put forward as a significant barrier for digital transformation. Vial (2019) finds that inertia is one of the "most significant barriers of" digital transformation. In the context of digital transformation, lack of organizational capabilities and resources, as well as path dependency across supply chains in specific industries are identified as typical sources of inertia. However, inertia can also refer to the limited changeability of digital technologies in use in organizations, due to path dependency because of high switching costs (Shapiro and Varian, 1999), digital debt (Rolland et al., 2018), entrenched legacy IS (Mehrizi et al., 2019), and complexity of the installed-base of large-scale infrastructures (Aanestad and Jensen, 2011).

In terms of the IT-function and especially regarding software development, the literature on agile software development proposes not only that IT organizations should adopt agile methods but a more comprehensive agile transformation (Dikert et al. 2016). This shift requires a tighter integration of business strategizing, and all phases and types of work involved in software development and deployment are needed. In order to achieve this, a new organizational structure has been proposed (Fitzgerald and Stol, 2017). First, DevOps incorporates the two words of development and operations, bridging the gap between the two. It emphasizes effective and unified collaboration (Ebert et al. 2016; Lwakatare et al. 2015). This shift means a change towards cross-functional teams working with continuous feature deliveries as opposed to organizational silos performing these functions separately (Ebert et al. 2016). Further, DevOps signifies the demand for a closer connection between the development of software and the deployment into production (Fitzgerald and Stol, 2017). Although agile methods and continuous development practices have become appealing alternatives for many organizations, it has proven challenging to introduce such methods in larger organizations and enterprises. Henceforth, recent literature has frequently pointed at failures and inherent challenges of agile transformation processes (Dikert et al., 2016; Korvonen, 2013; Paasivaara et al., 2018), challenges in adopting DevOps (Kuusinen et al., 2018), and specific CD practices (Chen, 2015). In a systematic literature review of large-scale agile transformations Dikert et al. (2016) have identified a number of challenges. These include change resistance, coordination challenges in multi-team environments, hierarchical management and organizational boundaries, and requirements engineering challenges.

Paasivaara et al. (2018) studied large-scale transformation at a global telecom company, Ericsson, and confirm that such processes are not without organizational problems and challenges. In meeting these challenges, the authors recommend: 1) an experimental approach to transformation, 2) evolutionary implementation of new ways of working, 3) increased specialization of teams' work, and 4) to use a common agile framework for the entire organization. However, since most studies are largely descriptive, the current literature is relatively weak on theorizing why certain transformations succeed and why others fail. Henceforth, also agile transformations of IT organizations can become a barrier due to inertia of existing software development practices.

Highly relevant for transforming IT organizations is also the level and content of outsouring of different IT functions. Law (2018) gives insights how IT sourcing decisions have inherent inertia and path dependencies in the way that it influences competence, ways of working, and organizing of the IT function. Grounded in a longitudinal case study, the author shows how sourcing of IS development has path dependencies (Law, 2018). Moreover, the study also shows how organizations can break with current paths and generate new paths through resource commitment, transferability of competence, and mobilizing for expanding the scope of options for selecting superior alternatives.

With this backdrop, this paper argues that there is a need for understanding the role of the IT-function in relation to digital transformation. As digital transformation involves combinations of different technologies (Hinings et al., 2018), specific organization capabilities (Li et al., 2018), agile ways of working across and integrating business strategy, software development and IT operations (Chan et al., 2019; Fitzgerald and Stol, 2017), and new governance regimes (Gregory et al., 2018), it is necessary that the competencies and practices of the IT-function changes as well.

2.2 Path dependency as a lens for conceptualizing transformation of IT organizations

In order to conceptualize and explain transformations of software organizations, such as the IT organization in the Government Organization, we adopt a lens of path theory (Garud et al., 2010; Meyer and Schubert, 2007; Singh et al., 2015; Sydow et al., 2009; Sydow et al. 2012). Path theory, or path constitution theory, comes in different flavors, and has been developed in multiple streams of literature. In general, path theory attempts at integrating the (somewhat) competing theories of path dependency and path creation. Path dependency and the related concept of lock-in to a situation with no remaining decision options for renewal, were initially coined in evolutionary economics as a theoretical basis for explaining why actors select less than optimal solutions in constituting the path of complex technologies (Arthur 1989; David 1985). In short, path dependency implies that historical choices and events narrow actors' choice options in the present (Meyer and Schubert, 2007). As partly a critique and an extension, Garud and colleagues (Garud et al. 2010; Karnøe and Garud 2012) developed path creation as an alternative view emphasizing that actors can mindfully deviate from existing paths. Applying a path theory perspective on IT organizations, then, opens the analytical possibility for explaining *both* why even quite radical changes in for example development approaches and practices (Laanti et al. 2011) and sourcing strategies (Law, 2018) takes place, and why similar changes in other cases do not seem to materialize – at least not to the same extent (Bick et al. 2017).

Perhaps most relevant for transformations of IT organizations is Sydow et al. (2009)'s operationalization of path theory on an organizational-level. They develop a framework for analyzing path constitution across three distinct phases: 1) preformation phase; 2) formation phase; and 3) lock-in phase. They describe path constitution as a process that at some point reaches a critical juncture that marks the transition from a preformation phase into a formation phase. At this point, a specific pattern or organizational path starts emerging, and a self-reinforcing mechanism kicks in. Because of the self-reinforcing mechanism, the options for organizational actors to deviate from the organizational path are increasingly diminishing.

Central in Sydow et al.'s (2009) conceptualization is that path dependency and the constitution of paths are driven by *self-reinforcing mechanisms*. The authors develop a framework consisting of four more specific occasions in organizations or organizational units, where self-reinforcing effects can take place. First, there are *coordination effects* caused by actors adopting the same organizational routine or rule, so that it becomes increasingly attractive for others to adopt the same routines and rules. Thus, coordination gets increasingly more efficient and less costly. The second element in the framework is *complementary effects* which refers to self-reinforcing effects arising out of "the interaction of two or more separate but interrelated resources, rules, or practices" (Sydow et al., 2009: p. 699). Complementary effects are generated through combinations of multiple routines and practices, so that it becomes increasingly attractive to adopt all of them as an "institutional cluster" (David 1994). Third, there are self-reinforcing *learning effects* at different levels in organizations. Obviously, the more often a specific task is performed, the easier it becomes to perform it, and the more efficient it is executed. Arguably, all these three variants of self-reinforcing mechanisms are relevant for IT organizations. In terms of learning effects and coordination effects related to choices in development approaches (i.e. waterfall, hybrid, or agile), development platforms (i.e. AWS, Google, or Azure), programming languages (i.e. Java or Python), and deployment routines (i.e. seldom or continuous). Moreover, as illustrated in a case study by Law (2018), sourcing models do indeed carry strong path dependencies. Having first outsourced software development activities, makes it progressively harder to break this setup due to self-reinforcing mechanisms of complementary effects (i.e. sourcing model, competence, and contracts). A fourth aspect where self-reinforcing mechanisms can play casual part, is the so-called *adaptive expectation effects.* Since individual preferences or choices are affected by other relevant actors' expectations, it becomes increasingly preferable to choose a certain solution that is perceived as 'right' by a growing number of relevant actors. This is also a question of legitimacy: "…individuals or subsystems not subscribing to the mainstream practices are afraid of losing legitimacy and – if associated with failure – of becoming stigmatized as "outsiders"" (Sydow et al., 2009: p. 700). This mechanism is highly relevant in relation to large IT efforts that have inherent risks and are often prestigious (Fitzgerald and Russo 2005; Flyvbjerg et al. 2003). Whereas these organizational phenomena have the potential – through a self-reinforcing mechanism – to generate path dependency and lock-in effects, there are also contextual conditions of ambiguity and complexity that can increase the possibility for a self-reinforcing mechanism to be triggered (Sydow et al, 2009). For IT organizations this issue is also highly relevant as they often serve various user groups with diverging interests and needs, and thereby increasing the ambiguity in expectations towards new software solutions. IT organizations are often also struggling with already complex digital infrastructures that may lead to conservative deployment schemes there users only seldom get new software and upgrades.

However, the self-reinforcing mechanism does not explain how breaking of a path leads to the possible formation of new paths. The original version of path dependency theory argued that paths could only be broken by external shocks or extraordinary events (Myers and Schubert, 2007). In contrast, path creation as coined by Garud et al. (2010) is perhaps the most elaborate attempt at describing how actors can *deliberately* break paths. In a path creation perspective, agency is seen as a largely distributed underscoring of possibilities in which many different actors in different situations and contexts can contribute to deliberately 'breaking' existing paths. Furthermore, path creation emphasizes that novel paths can stem from improvisation and bricolage, as well as actively cultivating serendipity as for example the case of innovating the famous Post-it Notes (Garud et al., 2010). Law (2018) shows how path dependencies can be broken by strategically mobilizing resources for creating new paths. Rolland et al. (2015) use the terms architectural path dependencies and architectural hacking, in order to theorize evolution of enterprise architectures both is influenced by path dependencies and gives opportunities for path creation. In this paper, we will suggest the term *path breaking* as a collective notion explaining the practices and digital technologies that render existing organizational paths obsolete. We summarize the theoretical concepts and their relevance to IT organizations in Table 1.

Table 1. Conceptualizing Transformation of IT organizations

Concept	Definition	Mechanisms	Relevance
IT organizational path dependency	The continuation of inflexible and possibly inefficient action patterns that are shaped by the unintended consequences of former decisions and positive feedback processes (Sydow et	There are four types of self-reinforcing mechanisms that can lead to organizational path dependency (Sydow et al. 2009): 1. Coordination effects 2. Learning effects 3. Complementary effects	Coordination, learning and adaptive expectation effects potentially causing path dependency in relation to: ISD methods, delivery routines, sourcing models, tools and infrastructure (Law, 2018). Complementary effects related to successful combinations of software development methods, delivery routines, sourcing models, tools and

	al., 2009)	4. Adaptive expectation effects	infrastructure.
Path breaking in IT organizations	The transformation of an existing organizational path that radically breaks with historical action patterns (Djelic and Quack, 2007)	There are three types of mechanisms found in the literature for breaking and possible generating new paths: 1. Experimenting with new practices and technologies (Rolland et al., 2015) 2. Paradoxical intervention (Sydow et al., 2009) 3. Mobilizing resources for collective action (Garud et al. 2010)	Recent evidence that many IT organizations manage go from traditional waterfall-like approaches to agile or hybrid development methods even in large-scale projects (Laanti et al., 2011). Also, path dependencies and near lock-in regarding sourcing models can be broken (Law, 2018).

3. METHOD AND CASE DESCRIPTION

3.1 Case description

The fieldwork was conducted within the IT organization of a large public Norwegian welfare organization. The Government Agency (GA) is one of the pillars in the Norwegian welfare system and is responsible for redistributing one third of the national budget through schemes such unemployment benefits and pensions. The organization was established in 2006, following the reform of the welfare system. With the reform, the formerly independent Employment agency and National insurance agency were merged in to a single unit. In addition, the reform also involved the establishment of a formal collaboration between GA and the municipal social services. In total, the organization employs approximately 19,000 people, of which 14,000 are central government employees and 5,000 are employed in the municipalities. In 2016, the IT organization at GA was relatively small compared to the number of employees with 510 employees, where approximately 200 are working with IT operations. In addition, there was approximately 400 consultants working on small and large IT-projects.

Shortly after GA was established, the Parliament elected a reform of the Pension system. To realize the reforms, GA had to develop a new system for the management of age pensions. The old system was based on dated technology and could not support the required changes. The resulting project was large, with an estimated development cost of several hundred million Euros. GA neither had the resources nor the competence to run a project of this magnitude. The project was therefore outsourced to consultant companies. The solution was delivered on time and within budget, and although the IT organization was skeptical many of the architectural decisions made by the consultant company, the project was considered a success among users and business experts. The outsourcing of software development would last for a decade. A waterfall approach to software development was institutionalized across the IT organization. This methodology was well documented and supported by training schemes, reporting routines, and tools. Deployment was organized as large coordinated releases that could approach 150 000 development hours. Henceforth, there was path dependence related to the competence (e.g. learning effects) in the IT organization as they were experts on controlling and planning outsourced projects and IT operations, but they had less expertise in actual development work.

3.2 Data collection

The paper draws on a longitudinal case study (Pan and Tan 2011; Yin 2009). Data was collected through in-depth semi-structured interviews (Myers 2013), observations, and document analysis. A total of 41 interviews were conducted (Table 2). Interviews lasted approximately 60 minutes. Data was collected both from the line-organization (section for IT Architecture), and from ongoing projects – with particular emphasis on an ongoing mega project (the Parental Benefit project). Data collection started in February 2017 and lasted to February 2019.

Table 2. Overview of Interviews

Roles	The project	Architecture unit	Other	Total
Managers	12	2	1	15
Developers	11		3	14
Architects	6	4	1	11

3.3 Data analysis

The data analysis was iterative and overlapped with data collection, thus granting flexibility to respond to emergent themes (Eisenhardt 1989). This followed Pan and Tan's (2011) process approach, with a framing cycle, followed by an augmenting cycle. During the framing circle, we reviewed documents, did participant observation, and conducted the first batch of interviews. The data material was then written up in a case narrative covering the past 12 years. The narrative provided an overview of events leading up to the agile transformation. The data material was analysed thematically recognizing both self-reinforcing and path breaking mechanisms.

Based on the case narrative we identified defining episodes in the transformation of GA. We describe these stages more fully in the case results section. For each phase, we identified mechanisms that either reinforced or broke with the existing path.

4. CASE NARRATIVE

In our analysis of the case, we have chosen to zoom in on four phases that represents four important attempts at transforming GA's IT organization over a time period of seven years. This is summarized in Table 3 below.

Table 3. Four Phases in the Transformation of GA's IT organization

Phase	Description	Outcomes
Phase 1 (2011-2015) – Failed IT program	A major IT program to modernize and substitute existing legacy systems fails. Management responded by reverting to the old strategy, thereby reinforcing the existing IT organizational path.	**Reinforcing existing IT organizational path:** • Outsourcing of software development continued. • Staged waterfall-type software delivery method with agile practices within each stage. • Large coordinated releases continued. • Application platform allowed for semi-automated provisioning and deployment
Phase 2 (2016-17) – Successful piloting of new software development practices	In the wake of the failed IT program, a new CEO and CTO are appointed. A project is established to pilot cross-disciplinary development teams and continuous deliveries.	**Breaking with current IT organizational path:** • The Pilot project represented a break with current practice and demonstrated continuous delivery and cross-disciplinary development teams as a doable alternative. • The project develops new functionality and has few dependencies to other systems and projects.
Phase 3 (2017) – Mobilizing a new sourcing strategy	NAV realizes a new sourcing strategy by attracting talents from the IT community. The IT organization is re-organized giving room for different ways of working.	**Generating a new IT organizational path:** • Insourcing of software development. • Large projects still employ staged waterfall process and large coordinated releases. • Introduction of second-generation application platform based on Kubernetes, an extensible open source container platform that allows for fully automated deployment and provisioning and monitoring of software
Phase 4 (2018) – Fully implementing BizDevOps	New delivery model based on multidisciplinary teams that have responsibility for entire process within a "product area". Cross-disciplinary teams and continuous delivery practices	**Reinforcing existing IT organizational path:** • Continued insourcing strategy. • Mixture of staged and incremental delivery methods. • Mix of continuous and coordinated deployment.

	introduced into existing projects with high complexity and many dependencies.	• Increasing number of applications migrated to new application platform.

4.1 Phase 1: Reinforcing the existing IT organizational path after a failure project

As the merger was completed and all front-line offices had been established in 2010, GA was ready to plan its digital future. Hence, in 2011, they begun to plan a large modernization program. The program would replace one of the oldest and largest management systems - a benefit management system which had been in operation since 1978. The system was built on dated technology and had become difficult to change and maintain. The system would be replaced through three consecutive projects, where the commencement of one project would depend on the successful completion of former. Each project would last for approximately 2 years with an estimated cost of close to 340 million Euro. Due to an elected reform of legislation governing payments of disability benefit, the first project would develop functionality for the payment of disability benefit. Disability benefit is a welfare benefit that is paid to persons who have a permanently reduced earning capacity due to illness or injury. The reform was elected by Parliament in 2012 and would take effect 1. January 2015. Hence, this date marked an absolute deadline for the project.

At this point in time GA had a long history of outsourcing software development. Since the project was considered strategically important, it was however run by internal resources. The project was perceived as an opportunity to fix “everything” that was malfunctioning. Unfortunately, an ambitious scope, combined with strict deadlines and lack of experience led to failure. The project was terminated after just 6 months on ground of “unforeseen complexity”. The failure led to an open parliamentary hearing, and eventually triggered the resignation of the GA Director and was later followed by the resignation of the IT director. To be able to support the legislative changes initiated by the reform, the organization returned to the delivery strategy that had successfully in the past: The project was outsourced in its entirety to external consultants, and the attempt at transforming the organizational path ended up reinforcing it: Disability benefits were implemented as part of the Pension system, using the same methodology as they had done in the Pension project. Henceforth, the failed project also created strong adaptive expectation effects for other projects to follow the outsourcing strategy to software development.

4.2 Phase 2: Breaking the existing IT organizational path by experimenting with new development practices

Although the IT organizational path remained unchanged after the Disability project, it had sparked several initiatives. Among these were the development of an application platform, and the development of digital follow-up activities related to sick leave. The project, which was called DigiMed, departed from existing development practices, delivering software continuously and incrementally. Even though it was small, the project represented a new way of developing and managing software and became a pilot for a new delivery strategy. In place of handovers and the waterfall type method, software was developed by cross-disciplinary teams that were responsible for the entire software development cycle. Software was released frequently, triggering continuous feedback from sponsors and users. In the aftermath of the failed IT program, GA also appointed a new IT Director. The new Director immediately started establishing a new IT organization and proclaimed that GA should change its strategy and rely less on consultants and more on internal expertise. The promise of working with high-end technologies and the radical strategy of the IT Director – breaking with a focus on outsourcing development and working in multidisciplinary BizDevOps teams also attracted experienced developers to GA's IT organization. This contributed to breaking the current organizational path dependency as the new developers mobilizing support for this way of working. The IT director later described the DigiMed project as an important front-runner for changing the existing organizational path: "*It demonstrated the value [of working like this]*". In this way the new IT Director managed to mobilize resources for breaking with the existing ways of conducting software development in cross-disciplinary teams and practicing Continuous Delivery. As the DigiMed-project was successful, it represented a superior way of working compared to the old-fashioned waterfall-approach.

4.3 Phase 3: Mobilizing a new IT organizational path

In 2017, GA decided to change their outsourcing path, and went from outsourcing to insourcing of software development. The insourcing strategy was accompanied by a reorganization of the IT department. The hierarchical structure was replaced by a flat organization, and the old management team was replaced by managers that believed in the strategic shift. The DigiMed pilot project had proven successful, and the IT Director- wanted to scale cross-disciplinary development teams to the rest of the IT organization. The ambition was to take leadership and ownership of their own solutions and developing expertise in-house – thereby assuming larger responsibility for systems development, maintenance, and operations.

Unlike the prior organization, where there were handovers between departments, the new strategy gave teams responsibility for the entire product development cycle. This was achieved by establishing autonomous, cross-functional teams, having different expertise such as architects, developers and testers in one team as opposed to the former organization, where specification, quality assurance and operations were performed as own units or departments in the organization. Consequently, establishing cross-disciplinary development teams, where each team had to know everything about the solutions that support their product, and what framework conditions they operated under, as well as being responsible for own architecture and the relationships to other applications.

Since GA lacked internal development competence, consultants had to be replaced gradually. This was done by establishing internal product development teams that were staffed with consultants, but with a GA team lead. The consultants in the product teams would be replaced as GA succeeded with their employment strategy. Close to 200 developers would be employed within the next two years and responsibility contracts where consultant firma were responsible for development and maintenance were gradually replaced by capacity contracts, limiting the use of consultants to peak periods and specialist competence.

Further, to facilitate the transformation of IT organization, GA developed an open source framework based on Kubernetes. The platform provided automated provisioning and deployment and allowed development teams to deploy and manage their own applications without the involvement of the Operations department. *"The platform takes the world's best operations person and automates him"* (Member of platform team). Hence, the use of container-based platforms strengthened the new organizational path, as it enabled a distributed governance structure where cross-disciplinary teams could work in relative isolation. The new platform, together with cross-disciplinary development teams, and an insourcing strategy also slowly sparked coordination effects and complementary effects for the new overall strategy. In addition, the use of new platform technology was an effective way of attracting IT competence in a competitive market.

By the end of 2017, the organization was split between two competing organizational paths. On the one side stood proponents of the old organizational path; the managers and coordinators that had specialized in procurement and control. On the other side stood developers and managers who believed in cross-disciplinary development teams and CD.

4.4 Phase 4: Reinforcing a new IT organizational path by fully implementing cross-disciplinary teams

The IT organizational path supporting insourcing was reinforced by applying the new delivery model to an ongoing mega project. Up-until then, the new delivery model had only been used in smaller projects with a limited number of dependencies. The IT department recognized that to succeed with the strategic shift, they needed to implement the new strategy across projects, including the large and ongoing parental benefit project. Attempting to implement CD in an ongoing mega project was however challenging. To initiate the change, the mega project begun by establishing a single cross-disciplinary team that would work alongside the traditional development teams. Even through the cross-disciplinary team only developed a small and isolated part of the system, they demonstrated the benefits of working in multidisciplinary team. The establishment of the cross-disciplinary team helped convince project sponsors that the entire project could deliver better value be working in multidisciplinary teams.

One of the essential initiatives to provide better conditions towards CD was to terminate the deployment process characterized as “main deliverables”. GA and the parental money project were used to coordinating all applications, projects and environments and bundle them into a main deployment schedule. In order to terminate this deployment process, it was necessary to separate and isolate applications and systems for quicker deployment and decreasing the number of dependencies. This required isolation between applications, reducing dependencies so deployment did not need to be coordinated and delivered at the same time. Also, if applications and systems were broken up, and could rotate independent of each other, it could reduce the need for plan-build-run organizing. This would replace handovers between departments with autonomous teams, operating independently, taking responsibility for the entire system development cycle by being cross- functional. As applications and teams had varying environments and context to work by, the control system of white, grey and black-listing helped to distinguish which teams and applications were mature enough for CD. Teams and applications that managed to isolate themselves were allowed to be white-listed and could adopt CD. If the application had too many dependencies, they needed to be coordinated into main deliverable, bundled with other applications as the ability to modify these systems were low.

5. DISCUSSION AND CONCLUDING REMARKS

This paper aims at empirically describe and theoretically explain both inertia and change in IT organizations. Insight into such transformations are important as organizations are moving towards becoming increasingly “digital organizations” involving a fusion between IT strategy and business strategy (Bharadwaj et al. 2013). This shift often requires IT organizations to transform their sourcing models, IT-organization, and move towards cross-disciplinary teams continuously deploying new software and systems (Fitzgerald and Stol, 2017). In order to theorize and explain outcomes of such processes in IT organizations, we draw from path theory (Garud et al. 2010; Mahoney, 2000; Meyer and Schubert, 2007; Sydow et al., 2009) and show how IT organizations can develop path dependency as well as breaking long enduring path trajectories. Grounded in our path analysis of a longitudinal case study, we provide two distinct contributions.

First, we theorize how IT organizations tend to develop path dependency related to interaction and interdependencies between development activities and methods, IT sourcing practices, and software delivery routines. These important aspects of the IT organization tend to become largely complementary entities making each individual entity harder to change because of interdependencies and self-reinforcing mechanisms. As seen in the case study, although many of the involved consultants and internal employees in GO's IT organization were strong proponents of a transition towards cross-disciplinary teams and continuous development, learning and coordination effects prevented a transformation during Phase 1. Furthermore, the complementary effects of a historical decision to follow a strategy for outsourcing software development while simultaneously insourcing IT operations, made cross-disciplinary teams very hard to implement as it required experts from IT operations, developers from consulting companies, and business experts to work together in teams. Finally, also adaptive expectation effects were present in Phase 1 of the case. Top management of the GO organization and other Government departments expected the IT organization to return to the old regime of outsourcing software development that had worked out previously. Paradoxically, the attempt at transforming the way the IT organization worked towards more agile practices, GO ended up reinforcing the existing organizational path characterized by outsourcing, waterfall development, and conservative deployment rates. Furthermore, we theorize that a setting with complex digital infrastructure with legacy systems seems to lower the bar for development of path dependency. In this way, organizations with more modularized and standardized components and IT systems are likely to be less vulnerable to path depended as for instance outsourcing of some of the systems and development activities will not be so interdependent with the rest of the IT organization.

A second contribution is related to the practices and mechanisms that helped break the path trajectory characterized by outsourced software development, waterfall methods, and long-drawn routines and infrequent delivery of new software. Consistent with a path lens (e.g. Garud et al., 2010; Singh et al., 2015; Sydow et al., 2009), our case shows that IT and software organizations can indeed break with existing organizational paths and transform themselves. Especially, based on the longitudinal case presented in this research, we theorize that three practices were essential in breaking the IT organizational path dependency. First, we theorize that experimenting with and implementing different kinds of cross-disciplinary development teams is a path breaking mechanism. In the case of GO, introducing cross-disciplinary development teams was to conduct a paradoxical intervention (Sydow et al., 2009) as there was a stringent division between the line organization and the software development project. This led to a more effective software development process avoiding numerous hand-overs that involved misunderstandings and immature design decisions. Cross-disciplinary teams involved a fusion of different competencies improving technical decision making, and integrated insights from the line organizations with the development expertise. A second mechanism was related to how the IT organization attracted and secured competence. In the case of GO, the new IT director did this by publicly announcing in newspapers and at practitioner's conferences that GO was turning 180 degrees to implement insourcing, autonomous teams, and continuous delivery. This opened the scope of action as it made it possible for the IT organization to attract individuals that had experience and competence with cross-disciplinary development teams and continuous delivery from high-end platform companies. This, however, came as a surprise for consultant companies delivering on the existing outsourced projects. But, for other actors at

GO it provided an opportunity to re-adjust the way that current projects were run. So, during Phase 4, the outsourced projects and the consultants working on the customer-side of the projects did not have many choices as they were offered new favourable contracts complying to the new way of working. Consequently, we theorize that in order to transform an IT organization attracting and securing competence becomes crucial. Thirdly, the introduction of digital platform for new software development t virtualizing infrastructure resources and automating many IT operations tasks is crucial for the transformation of IT organizations. Thus, we theorize that an essential path breaking mechanism not indicated in previous literature, was in our case related to an applications development platform configured by a team of newly hired experts. This platform made it easier to introduce cross-disciplinary development teams that could develop and deploy new software in a relatively autonomous manner as it provided more standardized interfaces to some of the complex integration and infrastructure issues. The new platform also automated previously complex tasks related to monitoring and testing. Moreover, the platform was very important for attracting new developers and technical architects to GO as it was seen as the current state-of-the art for developing complex software systems. In this way, the platform can be argued to be a mechanism that in interaction with the other two path breaking mechanisms helped establish cross-disciplinary teams and continuous delivery practices on a more permanent basis. The use of new technologies and platform technologies are not much recognized in existing literature on agile transformation. For example, Dikert et al., (2016) that offers a comprehensive literature review does not mention this as part of their list of success factors. A different turn is to look at how new digital technologies and layered modular architectures (Yoo et al. 2010) such as digital platforms has a potential to become a mechanism for establishing cross-disciplinary teams and continuous delivery practices – given certain contextual conditions. For example, Rolland (2018) explain how a large media company managed to renew their digital infrastructure and process in media production by continuously leveraging a digital platform and its ecosystem.

Furthermore, it is interesting how these three mechanisms seem to have worked well together in a complementary fashion that in the end lead the IT organization down a very different path trajectory characterized by insourcing of software development, highly agile and cross-disciplinary teams, short and frequent delivery routines, and over time the discontinuation of some of the most crucial parts of a legacy system.

Taken together this provides us with new implications for digital transformation of incumbent organizations. Our research shows that it is essential to transform the IT organization in order to be able to digitally transform the organization as a whole. This essential part of digital transformation processes seems to be nonexciting in current research on the topic. Regrettably, current IS research on digital transformation also fails to underscore how digital technologies both can facilitate change and as well be a source of inertia. The literature on digital innovation and transformation provides numerous examples of how different kinds of digital technologies, like Internet of Things, big data analytics, and robots facilitate changes in organizations (e.g. Barrett et al. 2012; Jonsson et al., 2018; Lehrer et al., 2018; Nicolescu et al., 2018). However, there are only a small number of publications investigating at a finer level of granularity what components and aspects of digital technologies are changed over time and how they can reinforce inertia and path dependency as well as a flexibility to change.

The research also shows that IT organizations in incumbent organizations are likely to suffer under path dependence which make it hard to change. Henceforth, although IT organizations can be organized and structured in multiple ways as portrayed by Guillemette and Paré (2012), moving from one ideal model to a different one seems problematic because of path dependency.

To conclude, a path lens analysis, enables us to explain *both* why radical changes in for example development approaches and practices can be deliberately implemented and why similar changes in other cases do not seem to materialize – at least not to the same extent (Bick et al., 2017). It also underscores that it is not enough to look for single success factors or independent factors that challenge transformations of IT organizations (Dikert et al., 2016), but to look at how different interdependent issues and mechanisms of the IT organization are producing or breaking organizational paths. Furthermore, a path analysis explains how the transformation process unfolds over time, and how actors' actions may reinforce or transform the current IT organization (Singh et al., 2015). On a more practical side, our case describes how it is possible for also hierarchical and complex organizations to introduce cross-disciplinary development teams and CD successfully. Along with Paasivaara et al. (2018) we would recommend organizations to introduce cross-disciplinary development teams and CD carefully in an evolutionary manner. However, grounded in our study we would recommend IT organizations to unlock the interdependencies across sourcing models, software development methods and delivery schemes. Also, utilizing development platforms need to be considered in association with cross-disciplinary development teams and CD in order to succeed.

References

Aanestad, M., & Jensen, T. B. (2011). «Building nation-wide information infrastructures in healthcare through modular implementation strategies." *The Journal of Strategic Information Systems*, **20(2):** 161-176.

Arvidsson, V., & Mønsted, T. (2018). Generating innovation potential: How digital entrepreneurs conceal, sequence, anchor, and propagate new technology. *the Journal of strategic information systems*, *27*(4), 369-383.

Barrett, M., Oborn, E., Orlikowski, W. J., & Yates, J. (2012). Reconfiguring boundary relations: Robotic innovations in pharmacy work. *Organization Science*, *23*(5), 1448-1466.

Bharadwaj, A., El Sawy, O. A., Pavlou, P. A., & Venkatraman, N. (2013). Digital business strategy: toward a next generation of insights. *MIS quarterly*, 471-482.

Bick, S., Spohrer, K., Hoda, R., Scheerer, A., and Heinzl, A. 2017. "Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings," *IEEE Transactions on Software Engineering*), pp. 1-1.

Chan, C. M., Teoh, S. Y., Yeow, A., & Pan, G. (2019). Agility in responding to disruptive digital innovation: Case study of an SME. *Information Systems Journal*, *29*(2), 436-455.

Chen, Lianping. 2015. "Continuous delivery: Huge benefits, but challenges too." *IEEE Software* 32, no. 2 (2015): 50-54.

David, P. A. 1994. "Why Are Institutions the 'Carriers of History'?: Path Dependence and the Evolution of Conventions, Organizations and Institutions," *Structural change and economic dynamics* (5:2), pp. 205-220.

Dikert, K., Paasivaara, M., and Lassenius, C. 2016. "Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review," *Journal of Systems and Software* (119), pp. 87-108.

Djelic, M. L., & Quack, S. (2007). Overcoming path dependency: path generation in open systems. *Theory and society*, *36*(2), 161-186.

Ebert, C., Gallardo, G., Hernantes, J., and Serrano, N. 2016. "Devops," *IEEE Software* (33:3), pp. 94-100.

Eisenhardt, K. M. 1989. "Building Theories from Case Study Research," *Academy of management review* (14:4), pp. 532-550.

Fitzgerald, B., and Stol, K.-J. 2017. "Continuous Software Engineering: A Roadmap and Agenda," *Journal of Systems and Software* (123), pp. 176-189.

Fitzgerald, G., and Russo, N. L. 2005. "The Turnaround of the London Ambulance Service Computer-Aided Despatch System (Lascad)," *European Journal of Information Systems* (14:3), pp. 244-257.

Flyvbjerg, B. 2006. "Five misunderstandings about case-study research". *Qualitative inquiry*, Vol. 12, issue 2, pp. 219-245.

Garud, R., Kumaraswamy, A., and Karnøe, P. 2010. "Path Dependence or Path Creation?," *Journal of Management Studies* (47:4), pp. 760-774.

Gregory, R. W., Kaganer, E., Henfridsson, O., & Ruch, T. J. (2018). "IT Consumerization and the Transformation of IT Governance." *Mis Quarterly*, **42(4)**: 1225-1253.

Guillemette, M. G., & Paré, G. (2012). Toward a new theory of the contribution of the IT function in organizations. *Mis Quarterly*, 529-551.

Hinings, B., Gegenhuber, T., & Greenwood, R. (2018). "Digital innovation and transformation: An institutional perspective." *Information and Organization*, **28(1):** 52-61.

Jonsson, K., Mathiassen, L., & Holmström, J. (2018). Representation and mediation in digitalized work: evidence from maintenance of mining machinery. *Journal of Information Technology*, *33*(3), 216-232.

Karnøe, P., and Garud, R. 2012. "Path Creation: Co-Creation of Heterogeneous Resources in the Emergence of the Danish Wind Turbine Cluster," *European Planning Studies* (20:5), pp. 733-752.

Korhonen, K. 2013. "Evaluating the Impact of an Agile Transformation: A Longitudinal Case Study in a Distributed Context," *Software Quality Journal* (21:4), pp. 599-624.

Kuusinen, K., Balakumar, V., Jepsen, S. C., Larsen, S. H., Lemqvist, T. A., Muric, A., Nielsen, A. O., and Vestergaard, O. 2018, "A Large Agile Organization on its Journey towards DevOps," in *44th Euromicro Conference on Software Engineering and Advanced Applications*, T. Bures and L. Angelis, Eds., ed, pp. 60-63.

Laanti, M., Salo, O., and Abrahamsson, P. 2011. "Agile Methods Rapidly Replacing Traditional Methods at Nokia: A Survey of Opinions on Agile Transformation," *Information and Software Technology* (53:3), pp. 276-290.

Law, F. (2018). Breaking the outsourcing path: Backsourcing process and outsourcing lock-in. *European Management Journal*, *36*(3), 341-352.

Lehrer, C., Wieneke, A., Vom Brocke, J. A. N., Jung, R., & Seidel, S. (2018). How big data analytics enables service innovation: materiality, affordance, and the individualization of service. *Journal of Management Information Systems*, *35*(2), 424-460.

Li, L., Su, F., Zhang, W., & Mao, J. Y. (2018). Digital transformation by SME entrepreneurs: A capability perspective. *Information Systems Journal*, *28*(6), 1129-1157.

Lwakatare, L. E., Kuvaja, P., and Oivo, M. 2015. "Dimensions of Devops," *International conference on agile software development*: Springer, pp. 212-217.

Mahoney, J. (2000). Path dependence in historical sociology. *Theory and society*, *29*(4), 507-548.

Mehrizi, M. R., Modol, J. R., & Nezhad, M. Z. (2019). "Intensifying to cease: unpacking the process of information systems discontinuance." *MIS Q*, **43:** 141-165.

Meyer, U., and Schubert, C. 2007. "Integrating Path Dependency and Path Creation in a General Understanding of Path Constitution. The Role of Agency and Institutions in the Stabilisation of Technological Innovations," *science, technology & Innovation studies* (3:1), pp. 23-44.

Myers, M. D. 2013. *Qualitative Research in Business and Management*. Sage.

Nicolescu, R., Huth, M., Radanliev, P., & De Roure, D. (2018). Mapping the values of IoT. *Journal of Information Technology*, *33*(4), 345-360.

Paasivaara, M., Behm, B., Lassenius, C., and Hallikainen, M. 2018. "Large-Scale Agile Transformation at Ericsson: A Case Study," *Empirical Software Engineering*), pp. 1-47.

Pan, S. L., and Tan, B. 2011. "Demystifying Case Research: A Structured–Pragmatic–Situational (Sps) Approach to Conducting Case Studies," *Information and Organization* (21:3), pp. 161-176.

Rolland, K. H., Ghinea, G., and Grønli, T.-M. 2015. "Ambidextrous Enterprise Architecting: Betting on the Future and Hacking Path-Dependencies," *ECIS*.

Rolland, K. H., Mathiassen, L., & Rai, A. (2018). "Managing digital platforms in user organizations: the interactions between digital options and digital debt." *Information Systems Research*, **29(2):** 419-443.

Shapiro, C., Carl, S., & Varian, H. R. (1999). *Information rules: a strategic guide to the network economy*. Harvard Business Press.

Singh, R., Mathiassen, L., and Mishra, A. 2015. "Organizational Path Constitution in Technological Innovation: Evidence from Rural Telehealth," *Mis Quarterly* (39:3).

Svahn, F., Mathiassen, L., & Lindgren, R. (2017). Embracing Digital Innovation in Incumbent Firms: How Volvo Cars Managed Competing Concerns. *Mis Quarterly*, *41*(1).

Sydow, J., Schreyögg, G., and Koch, J. 2009. "Organizational Path Dependence: Opening the Black Box," *Academy of management review* (34:4), pp. 689-709.

Sydow, J., Windeler, A., Müller-Seitz, G., and Lange, K. 2012. "Path Constitution Analysis: A Methodology for Understanding Path Dependence and Path Creation," *Business Research* (5:2), pp. 155-176.

Vial, G. (2019). "Understanding digital transformation: A review and a research agenda." *Journal of Strategic Information Systems*, **28**: 118-144.

Yin, R. K. 2009. "Case Study Research: Design and Methods. Essential Guide to Qualitative Methods in Organizational Research (Vol. 5)," *The Information Systems Research Challenge (Harvard Business School Research Colloquium). London: Sage*.

Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "Research Commentary—the New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information systems research* (21:4), pp. 724-735.