

Growing platform-based enterprise systems through ‘modular’ and ‘architectural’ acts of customizing: a case study

Knut H. Rolland¹ and Margunn Aanestad²

1) Norwegian School of IT, Oslo, Norway 2) Department of Informatics, University of Oslo, Norway

rolknu@nith.no, margunn@ifi.uio.no

Abstract. Previous studies on enterprise systems have empirically illustrated and theorized the tensions (or misfits) between the standardized features of such systems and the specific needs of organizations. Often implementation studies have focused on immediate and local workarounds and not the long-term mutual shaping process of work practices and standardized software systems. In contrast, this paper draws from an on-going case study of a prolonged implementation effort of an enterprise system sequentially based on three different versions of Microsoft SharePoint software platform (2007, 2010, 2013). Our study shows that while workarounds flourish, standardized features of the system simultaneously also provide learning and boundary resources for incrementally customizing the system further. We contribute by conceptualizing implementation processes as growing an installed base through ‘modular’ and ‘architectural’ acts of customizing. Architectural acts of customizing refer to configurations, modifications, and additions of a new system in attempts to aligning it to the existing information infrastructure. These architectural acts of customizing, if successful, establish resources and learning for further modular acts of customizing incrementally modifying the system.

1 Introduction

Precisely because enterprise systems are standardized, generic software, and frequently implemented enterprise-wide, a particularly demanding aspect is adapting to specific organizational contexts. Not surprisingly then, adaptation difficulties have been substantially reported in qualitative studies in information systems research. For example, studies have uncovered: difficulties of adapting to contextual a diversity (Wagner and Newell, 2004); different types of misfits (Strong and Volkoff, 2010); how users turn to various workarounds and improvisations to accommodate post-implementation misfits (Boudreau and Robey, 2005; Ignatiadis and Nandhakumar, 2009); the organizational consequences of the tensions between generic modules and communities with strong identity and historically entrenched practices (Pollock and Cornford, 2004), and how users re-invent generic features across multiple sites of a global organization (Rolland and Monteiro, 2002). These – and many more studies have been important for explaining and theorizing upon the many unintended and even

contradictory organizational consequences, and hence the failure to radically transform and standardize work practices and organizations. As Fleck (1994) reminds us, we do not need battalions of examples for refuting the proposition that standardized software solutions provide near perfect fit with organizations' existing work practices and portfolio of information systems, and hence largely fails to radically transform work and organizations.

More recently however, it has been argued that the one-sided focus on overly situated accounts of workarounds and local appropriations has left a blind spot in current theorizing (Kallinikos, 2004; Monteiro and Rolland, 2012; Williams and Pollock, 2012). Is it really the case that standardized software solutions deployed enterprise-wide, if actually used, are so malleable that users often are comfortably able to work around the inscribed templates and rigorous workflows? As Williams and Pollock (2012: p. 4) explain, in light of the authors revisiting a previous case of a 'successful' enterprise system implementation producing the familiar workarounds allowing existing practices to flourish, now some years later "had all but disappeared and that practices and processes across the university had now mostly become aligned with those originally embedded within the ERP templates". Arguably, standardized features of the technologies, through which users' collaborative interaction engage with over long periods of time, slowly transforms practices across time and space – not to become identical – but similar enough (Monteiro and Rolland, 2012). Enterprise systems typically undergo much customizing during implementation (e.g. Brehm et al., 2001; Light, 2001; Markus and Tanis, 2000; Sawyer and Soutwick, 2002), but current conceptualizations largely fail to explain how the customization together with ongoing incremental modifications after implementation mutually shapes work practices and organizing in the long run.

Research on enterprise systems has also been concentrated around ERP-systems (i.e SAP), leaving out similar – but not identic technologies like Microsoft SharePoint, an ECM software platform that is equally diffused among private as well as public organizations. Combining rigorous document management with more fluid social collaboration, ECM-systems involve potentially conflicting features and views of organizing as opposed to ERP-systems that embed an all-embracing procedural logic (Kallinikos, 2004; Munkvold et al., 2006). Thus, in this paper we ask: *how does a platform-based enterprise system evolve through a prolonged implementation process involving customizations and different versions of a software platform?*

Empirically we draw from an on-going case study of a prolonged in-house development effort of an ECM-system based on various versions of the Microsoft SharePoint software platform (hence the emphasis on platform-based). Through a period of 5 years (2009-2014) and still on-going, 'Bergen Drilling' a Norwegian medium-sized oil and gas company operating internationally, has gradually been implementing the system through numerous modifications and extensions of three subsequent major versions of the SharePoint software platform (denoted 2007, 2010 and 2013). We draw from an information infrastructure perspective (Hanseth and Lyytinen, 2010; Hepsø, Monteiro and Rolland, 2009) as well as Henderson and Clark's notions of modular and architectural innovation (Henderson and Clark, 1990). In the next section we present relevant literature in order to construct our analytic framework. Then we offer information about the research study and the case before we present the empirical narrative in section 4, focusing on the history of modifications and customization of the system. In section 5 we analyse the narrative and distinguish between what we call architectural and modular acts of customizing.

2 Conceptualizing platform-based enterprise systems and change over time

The increased interconnectivity of information systems has spurred an interest in conceptualizing networks or layers of such systems as information infrastructures (Bygstad, 2010; Ciborra et al. 2000; Edwards et al., 2009; Hanseth, 2002; Hanseth and Lyytinen, 2010; Hepsø, Monteiro and Rolland, 2009; Rolland and Monteiro, 2002)¹. According to Hanseth and Lyytinen (2010: p. 4) information infrastructures can be defined as “a shared, open (and unbounded), heterogeneous and evolving socio-technical system (which we call installed base) consisting of a set of IT capabilities and their user, operations and design communities”. Taking a more temporal perspective and analysing how platform-based enterprise systems typically get intermeshed with a wider network of other systems over longer periods of time we argue there is much to gain from the insights in the literature on information infrastructures.

From the lens of information infrastructure, enterprise systems are typically not configured and implemented in one discrete activity or project. As pointed out by Edwards and colleagues (2007: p. 7), they rather tend to grow in a distributed fashion:

Since infrastructures are incremental and modular, they are always constructed in many places (the local), combined and recombined (the modular), and they take on new meaning in both different times and spaces (the contextual). Better, then, to deploy a vocabulary of “growing,” “fostering,” or “encouraging” in the evolutionary sense when analyzing cyberinfrastructure.

Firstly, the growing metaphor gives a more apt description of the process than ‘implementation’, which sort of expects that the process will complete. To emphasise growing rather than implementation, is to focus on the evolutionary process over longer periods of time that perhaps never is fully completed. In contrast, much of the literature on enterprise systems continue to have a strong emphasis on what happens immediately after implementation of one particular version or configuration (e.g. Boudreau and Robey, 2005; Strong and Volkoff, 2010). From a perspective of information infrastructure this is unfortunate, because an enterprise system potentially can have different organizational consequences over time as the infrastructure ‘grows’. Although not spelled out in details in their paper, Williams and Pollock (2012) point out that in one particular case where they studied ERP implementation in a university, the initial workarounds and inertia were over time more harmonized. Thus there is at least a possibility that relatively important organizational consequences change over time. Theoretically, taking an information infrastructure perspective, it would be surprising to find the ‘same’ consequences over time. Surely, the Internet did not have the ‘same’ social consequences for people 20 years ago as now – partly because it is more often used by greater numbers of people, but definitely also because its structure and functionality has ‘grown’ into something different.

Secondly, information infrastructures imply a complex network of different dependencies between various technologies and systems (Ciborra et al., 2000), as well as between distributed work practices (Monteiro and Rolland, 2012). From this we can expect also that enterprise systems need to connect to a wider installed base of system in order to ‘work’ and

¹ There is not enough space here to go into detail and give a complete review of all relevant literature on information infrastructures. For this we recommend the reader to consult Henfridsson and Bygstad (2013).

not to become an isolated island. For example, in studying the implementation of a SharePoint infrastructure in NorthOil, Hepsø et al. (2009) illustrate how the SharePoint infrastructure was only moderately successful because it did not connect well to existing installed base of niche systems highly important for engineers' work practices. Furthermore, it seems likely to expect the enterprise system to get integrated with more systems over time, and thus potentially having more profound consequences.

A third point highly relevant to our discussion is the issue of customizing. According to the literature, different forms of customizing or tailoring plays a large part in implementing enterprise systems (Brehm et al., 2001; Light, 2001; Markus and Tanis, 2000; Pollock and Cornford, 2004). Notably, Light (2001) studied customizing in two different case organizations, and found that customizing could be valuable to the organization. However, Light (2001) develops a taxonomy of different customizations, and argues that there are some types that have a larger impact on maintenance than others. For example, while he finds adding a 'new report' to have less implications whereas 'change functionality' as higher implications for maintenance. Furthermore, he distinguishes between 'generic customizations' that are done for all users in an organization, and 'local customizations' that are only used locally. Interestingly the author mentions that one of the main reasons for implementing an enterprise system is to get rid of a large collection of often badly integrated legacy systems. Drawing from the information infrastructure literature, we see that this is seldom the case – and more often than not the new system has to co-exist and integrate in various ways to the installed base (Ciborra et al., 2000; Hanseth and Lyytinen, 2010; Hepsø, Monteiro and Rolland, 2002). In the case of a Maritime Classifications Company, Monteiro and Rolland (2012) describe how an enterprise system was customized and re-customized to accommodate updates of a legacy system and the underlying client and server operating systems. Consequently, the more complex the existing systems and work practices are, the more customizing need to be done to 'fit' the new system with the socio-technical installed base. More over, customization is not only focused on 'functional fit' (Strong and Volkoff, 2010) or 'change functionality' (Light, 2001), but on something we can referred to as 'installed base fit'.

Customizing is considered all but a straight-forward process of simply selecting the modules and clicking the right boxes for installing the software. Fleck (1994: p. 649) notes how such processes are "*a matter of 'learning by struggling to get it work', or perhaps more concisely 'learning by trying'. This does not merely involve fine-tuning to obtain incremental improvements in the operations of an already functioning system. More fundamentally, it involves the construction and development of new viable configurations in the first place*". In such learning processes, which also Fleck describes as a radical innovation, involves high levels of uncertainty and risk. Not only because the underlying software platform can be configured in many ways and involves selecting from numerous components, but also that the new configurations must connect socially as well as technically to a larger installed base (Hanseth and Lyytinen, 2010).

To sum up, we see implementation as a stretched-out process of mutual adaptations between technology and the user environment, triggered by misalignments that become resolved over time (Leonard-Barton, 1988). Hence, we focus on acts of customization, and in order to increase the granularity of analysis, we will seek to identify difference in the nature of these processes. We push Light's (2001) taxonomy of customizations further by using Henderson and Clark's notions of modular and architectural innovation (Henderson and

Clark, 1990). Thus we aim to contribute by conceptualizing platform-based enterprise system implementation as growing an installed base through two different but equally important acts of customizing. *Architectural acts* of customizing refer to the customizing that is directed to fitting the new system to an existing architecture of systems and practices. In contrast to existing accounts of customizing and implementation processes, this underscores that implementing such systems also involve customizing not only because of functional misfits but also because what we can refer to as an architectural misfit. Architectural acts of customizing also establish new resources for more confined and incremental acts of customizing referred to as *modular acts*.

3 Case study of a platform-based enterprise system in Bergen Drilling

3.1 Company context and research method

Bergen Drilling is a privately owned medium sized company located in the western part of Norway. Currently, the company has approximately 300 employees located at nine different locations in America, Asia and Oceania, including larger offices in Houston, US, Perth, Australia and a HQ in Bergen, Norway. The company has over the past 25 years or so been through a remarkable journey with substantial organization changes, mergers, splits, and collaborations resulting in numerous advanced technological innovations used around the globe by some of the largest companies in the oil and gas industry.

The company started off in a small village (which in Norway translates to a few farms, an old community house, a wooden church, and a petrol station) on the wet and windy west coast of Norway by a team of three friends in 1987. The local entrepreneurs started their business by cleaning various kinds of equipment used in the production of oil in the North Sea. In the 1990s the company won a contract with a major international oil and gas company on conducting technical inspections of equipment and the operations of maintenance. This spurred a considerable expansion through various mergers and acquisitions. Over the years, the company expanded internationally, and in 2011 the small start-up had grown to a corporation (Bergen Oil & Gas Group) with 1800 employees with a HQ in Australia. At this moment in time the corporation spanned a wide collection of products and services including petroleum production with complete crew, engineers, equipment and floating rigs, subsea installations, maintenance and drilling. In close relation to one of the major companies in the oil and gas industry, several production technologies used in advanced drilling operations and oil and gas production in the North Sea have been invented. For example, an advanced system used on the seabed for returning mud and cuttings to the offshore rig.

In 2012 parts of the company was sold to an American company. The remaining part was then a bit later on, in 2012, split in three different companies. The drilling part of Bergen Oil & Gas Group was then re-established as Bergen Drilling with approximately 200 employees mainly focused on selling advanced production technologies and the necessary expertise for installing and operating these around the globe.

This research is part of an on-going longitudinal case study historically tracing the process of implementing Microsoft SharePoint and currently also following the process further. The case of Bergen Drilling was selected because of the emphasis on out-of-the-box approach in conjunction with a turbulent business environment. In this respect it can be regarded a unique

case in that it represents a rather extreme approach to implementation given a highly turbulent environment (Eisenhardt and Graebner, 2007; Flyvbjerg, 2006). A total of 15 in-depth qualitative interviews lasting from 1,5 to 2,5 hours have been transcribed and analysed. These were mostly contextual interviews, involving interviewing while observing how the systems were used in the natural context of work. Additional 5 more informal discussions have also been conducted while visiting the research site. Here notes were written down shortly after or during the encounters. Collections of relevant documents like a technical overview of the IT infrastructure, IT strategy, and general information of the company have also been analysed. Data analysis has been conducted following open coding and selective coding as inspired by grounded theory (Urquhart, 2013). However, this is not a grounded theory study as we were inspired by the literature on information infrastructures as well as the concepts from the Boudreau and Robey's study (2005) in our research design, analysis and theorization. More over we have used the "temporal bracketing strategy" as suggested by Langley in order to analyse how the platform-based enterprise system has been mutually shaped by context over time (Langely, 1999).

3.2 Case background on the platform-based enterprise system

Having expanded globally, with major sites in Norway and Australia, there was in the late 2009 increasing pressure for coordinating operations and projects between the two locations. Especially, there was an envisioned need for standardizing document management and archiving across projects typically involving employees located in both Norway and Australia. This included for example technical drawings and operational procedures explaining how to install and operate technical equipment on oilrigs and crucially important for both contractual and safety/environmental reasons as explained by a project manager:

[A procedure] includes a description of the equipment to be used and various authorizations, and then a detailed description of what and how to carry out the tasks. First testing of the equipment, then fill out a checklist, and then increase mud pump to 1000 gallons per minute [gpm], and so on. If you do not follow a correct procedure the customer has about 100-150 million dollars in expenditures and lost production for the next 40 years... [We] write this together with the customer. Procedures are very important – also because it is a way of securing our operations financially. So if something goes wrong it should be [a major international oil and gas company's] responsibility – and not ours. (Project manager)

Consequently, in order to have better control over such documentation as well as improving collaboration both within departments and across countries, the managing director of IT located in Bergen, Norway decided to implement an enterprise-wide system based on Microsoft's SharePoint 2007 software platform. Launched together with the Office 2007 package, SharePoint 2007 was part of Microsoft's digital ecosystem fully integrated with Outlook/Exchange server email and the Word text editor. The SharePoint software platform (2007) is built on top of Windows SharePoint Services (WSS) 3.0 that provides a full-fledged development platform based on existing technologies like ASP.NET 2.0 for developing web applications. In short, SharePoint 2007 provides a set of standardized features and templates covering six different functional areas: collaboration, portal, enterprise search, enterprise content management, business process & forms, and business intelligence. In line with current

challenges facing the fast growing company the following was done, as the IT-director explains:

We had something called [Xerox] DocuShare [for document management] – that was vividly referred to as Docu-Scare and was definitely not working according to the intention. Nobody could find anything there, it was not properly customized and implemented in the organization, and thus it tended to be used as a file server. There was a total lack of competence on it – and as usual everybody was busy with external projects and customers... [We] decided to implement SharePoint [2007] instead and hired a consultant. He was a competent consultant – and had knowledge about the special features and architecture of the product. We wanted to have a new intranet, team-sites and document management based on SharePoint. It was also the initial idea to integrate existing niche systems for supporting maintenance [of offshore oilrigs] (IT director)

The project was established as an agile software development undertaking involving a team of one in-house SharePoint developer, one external SharePoint developer, and the HMSQ director and the IT director as product owners. As the case with other packaged enterprise solutions (e.g. Light, 2001; Pollock and Cornford, 2004), a pressing issue – much contested by practitioners and users alike, is whether or not these standardized features should be customized – and if so, how and to what extent. In short, there are at many ways of customizing SharePoint. First, you can configure existing web parts (i.e. software modules). A second option is to extend existing web parts writing your own code. Then, another alternative is to write your own web parts from scratch. And, finally you could either use a third-party developed web part under a type of open source license or purchase components from different commercial vendors. In addition, since SharePoint is basically web-based – you could always customize pages and templates directly using java-scripts and HTML/CSS coding. These and other possibilities heightened the expectations considerably, as one manager underscored that “[W]e had a view that SharePoint was like manna falling down from the sky that could be used for almost anything”. In practice, however, the project decided to go with Microsoft’s main advice and go for a so-called out-of-the-box strategy:

[The consultant] was keen on following the out-of-the-box strategy, which was the hype of time. In other words, the focus was on doing as little as possible – basically install the software. And our focus was to get this over with as quickly as possible. So the idea was to do as little as possible customization. You have to understand that management wanted as much as possible from the money they spent on the project. In addition, we did not have a lot of competence in the organization on SharePoint, so we did not want it to become too complex to maintain and upgrade over time. But he [the consultant] stayed with us for nine months... (IT director)

But what was first envisioned as a rather small and straight-forward IS-project, became a tedious process lasting for over 9 months. After first developing the platform-based enterprise system based on the SharePoint 2007 platform, and trying it out in the IT-department as well as two other departments, the entire solution was migrated and re-configured based on the SharePoint 2010 platform. Eventually since the project went on for much longer than the initial plan top management stopped the official project in late 2010. However, the system was implemented across the organization during 2010 and early 2011, and especially the news publishing portal was well received by employees throughout the company. Interestingly, a group of in-house developers and also a small group of super-users continued to re-configure,

modify and extended the solution over a prolonged period of 5 years. At the time of writing, a new project has been established in order to migrate and develop a new version based on the new SharePoint 2013 software platform.

<i>Period</i>	<i>Events</i>	<i>Technologies</i>
Q3 2009	IT-department experimenting and pilot testing team-sites in SharePoint locally.	SharePoint 2007
Q4 2009	Top management of Bergen Oil and Gas Group agrees to start a project on improving document management and collaboration within departments and across countries.	
Q1 2010	Consultant hired and project initiated with the goal of implementing a platform-based enterprise system comprising a new intranet news site, document management, and project sites using an out-of-the-box strategy.	SharePoint 2007 and Lightening tools
Q2 2010	Major discussion regarding customizing the information architecture of the department and project sites internally in the project and the among the superusers: flat versus hierarchical structuring of content.	SharePoint 2007
Q2 2010	Project decides to substitute the out-of-the-box enterprise search feature with third-party developed customized solution called Lightning tools.	SharePoint 2007 and Lightening tools
Q2 2010	Development of a new news publishing site and a common front page for a new intranet	SharePoint 2007 and Lightening tools
Q2 2010	Major adoption of the platform-based enterprise system fails	
Q2 2010	Migration and re-configuration of the system to new version of SharePoint. Standard version of enterprise search re-installed. Front page still based on old version, but eventually get re-developed.	SharePoint 2010/ SharePoint 2007
Q3 2010	Top management stops project due to escalating costs	
Q3 2010	The system is implemented in the organization and 15 training sessions are carried out with employees.	SharePoint 2010
Early 2011	The system is extended with a customized APP (the RUN database) for reporting well operations and tracing deficiencies.	SharePoint 2010, RUN database
Early 2011	The system is extended with several customized workflows for improving document management.	SharePoint 2010, workflows
Early 2012	Splitting the companies in different parts and re-establishing Bergen Drilling.	
Q2 2012	The company changes its department structure and thereby unintentionally influences the identity management for the platform-based enterprise system	Active Directory, HR-system, SharePoint 2010, workflows
Q1 - Q3 2013	The corporate infrastructure including the platform-based enterprise system is migrated to a new network domain.	Entire corporate infrastructure
Q1 – Q3 2013	Several features including existing workflows and the RUN database do not work appropriately anymore, and hence a Consultant company is hired to re-configure the solution.	SharePoint 2010, workflows, RUN database

Q4 2013	Additional consultant company hired for migrating existing solution to new version of SharePoint.	SharePoint 2010, SharePoint 2013, workflows, RUN database
---------	---	---

Table 1. The overall timeline of major events

4 Growing the platform-based system over time in Bergen Drilling

4.1 First act: Learning-by-trying (2009-2010)

As mentioned above, in compliance with the entrepreneurial culture of the company as well as Microsoft's and the external consultant's advice, an out-of-the-box strategy was pursued in order to "get the new system implemented as fast as possible" as underlined by the IT manager. In contrast to this, the project proved to be far more time-consuming and complex than first anticipated. What was envisioned a relatively straight-forward installation of SharePoint 2007, became a comprehensive learning-by-trying process involving multiple acts of customizing. The built-in flexibility of the SharePoint software platform provides multiple ways of organizing content on sites and an excessive number of standardized web-parts to be configured for each site. Much to the developers' surprise, the product owner and particularly engineers found the out-of-the-box way of structuring documents in document libraries tedious and bewildering, which in turn spurred a controversy:

To our surprise the transition from a hierarchical structure of documents to a flat structure became a struggle. This implied to figure out and settle on a common metadata structure – and nobody managed to decide on this. Everyone was asking: why can we not have more folders? And my answer was always, but I though we decided not having folders. It was very hard for me to explain and to talk them into organizing documents according to a flat structure using metadata. (Developer)

After much discussion between the product owner, HSEQ managers, engineers, and developers, the process ended – somewhat unintended, in a hybrid-solution representing a compromise between the proponents of hierarchical structure, the proponents of a flat structure and the capabilities offered by the software platform. The document library web part was customized so that uploaded documents were structured in a similar way as a typical hierarchical file structure and at the same time also could be viewed by end users as a flat structure.

Although this type of customization can hardly be perceived as technically complex as such, it involved a complete re-definition of classifying documents that were critical to how the company operated. As both engineers and project managers were used to a strictly hierarchical structure from existing systems – like a niche information system for technical documents (PDM) and drawings, the former Xerox DocuShare, and file servers – it required not only customizing of SharePoint functionality part of the generic platform, but more importantly how it should connect to a wider installed base of digital information important for assembling advanced drilling installations. The new platform implied a new classification system for the management of various digital documents that are deeply embedded in the social organizing of the work around projects, assembly of physical drilling products at the

shop floor, and meeting contractual as well as safety obligations in actual drilling operations offshore. Thus, the new system implied an architectural re-arrangement of digital content. This re-classification was in this way not neutral but embedded in a wider social context. This *act of customizing* was also fundamentally different from how Microsoft and the involved consultants perceived as ‘best-practice’ and how SharePoint was configured out-of-the-box. More over, it underscores the intrinsic dynamic to customizing where the issue is not simply that there is a misfit. What is going on here is a more complex learning-by-trying process where the flexibility of the SharePoint software platform offers many different ways of organizing document management and henceforth a tedious process of both technically and socially experimenting with different alternatives. On the other hand, this act customizing required that whenever a document was uploaded to the particular library, the user had to fill-in a long string of predefined metadata about the project name, project type, customer, etc. Obviously, this rigidity generated numerous workarounds and improvisations:

Well, there are lot of metadata for tagging – but not the right ones for my use. In our procedures we are supposed to do a risk assessment, so we produce a report there risks are analysed. And, then in the system there is no tag for ‘risk assessment’ or for something other relevant, so it typically gets tagged as a ‘report’ (HMSQ responsible)

It is possible to upload several documents at the same time in SharePoint – but if you do all the documents get the same metadata. Many did this, and then they forgot – did not care to, modify the metadata on each document. Thus, most documents have the correct project number and customer – but not the correct activity code. (Document controller)

Accordingly, after introducing the platform-based enterprise system into the organization, the immediate outcome was inertia in the sense that many users continued to use existing systems, and various kinds of workarounds flourished. Consequently, at this particular moment in time, the situation is very similar to what found in studies of other enterprise systems (e.g. Boudreau and Robey, 2005; Pollock and Cornford, 2004; Wagner and Newell, 2004).

4.2 Second act: more confined modifications (2010)

After much experimentation some departments early 2010 used a first version of the platform-based enterprise system. However, further adoption and bootstrapping of the new system relied on further customization. In particular, the search functionality provided out-of-the-box in SharePoint 2007 was not considered appropriate for the organization. Since much of the documentation regarding drilling equipment and technical drawing typically are used across projects, searching was vital for engineers and project managers. In addition, as a side effect of the overwhelming number of metadata – documents were not always categorized correctly making them hard to track down through simple navigation. The out-of-the-box search feature in SharePoint 2007 required additional customizing in order to index and crawl documents in a PDF-format, which most technical documents and drawings were. The project therefore decided to get hold of a third party software module in order to improve searching. Through a process of searching for and experimenting with different alternatives, the project finally settled on “Lightening tools for SharePoint 2007”.

This is an example of a type of customizing that builds on the previous experiences and learning process in the previous act customizing described above. It is more confined in the sense that it provides a more incremental change in the system by substituting and configuring

one single module. Interestingly, it builds on the early experiences with users' practices in use working around the intended categorizing (i.e. tagging) of documents.

4.3 Third act: migrating to a new version of the SharePoint platform (2010-2011)

Some 6 months after customizing the search feature of the software platform, a new version of SharePoint was introduced. This version, 2010, came for the first time with enhanced search facilities based on Microsoft's acquisition of the FAST search engine. Consequently, migrating from 2007 to 2010 made the initial customization of the search feature obsolete as the new out-of-the-box feature outperformed the third-party module. On the other hand, this required an act of customizing migrating from SharePoint 2007 to SharePoint 2010 – not a simple task at all:

I did a migration. First, I installed a test version of SharePoint 2010 to just take a look at it. When, I found a tutorial on the Internet for how to upgrade from 2007 to 2010. A late Friday night I started the procedure and copied the SQL database from 2007 to 2010 SQL database, among other things – it was a long shot. Eventually, I found new versions of nearly all the web parts [software modules] we were using. However, there was two web parts that could not be substituted, and those we just had to scrap in order to proceed with the migration. Then, we discovered a rather funny problem when I first launched the new version 2010. Suddenly the main page got all white [screaming out loud]. In the end we found out that this was because Microsoft had substituted content types [definitions describing what content to be shown in different containers on a web page] on the main page, so that the page specified in the 2007 version did not exist in the 2010 version. So then I had to develop the front page all over again. But in the meantime we hacked a version partly on 2007 and partly on 2010 (Developer)

Thus, again, the learning-by-trying describes well this first step of customizing. Secondly, the project had to re-configure the previously customized modules in the platform-based enterprise system now based on SharePoint 2010, to use the newly available searching features, and then to configure this to index existing sites and ways of presenting search results. This customization is more substantial in scope and more architecturally embedded redefining how different parts of the information system are put together. This suggests that what is at play here is a different form of learning beyond singular components or modules, more related to an architectural knowledge of how different components can – or can not work together.

4.4 Fourth act: Further growth and harmonizing (2011-2014)

With the new SharePoint 2010 as a foundation as well as users increased competence, the proceeding version of the platform-based enterprise system spurred some unanticipated functional extensions. Although the official project was now long finished, one developer in the IT-department, a group of so-called super users where constantly customizing the system. An increasing number of international projects and increasing complexity in offshore operations, made it necessary to have more structured procedures and systems for operational analysis and tracking. Previously, following up offshore operations had been done with an excel sheet, but as one informant explains in detail this changed:

[Previously] we did not have any tracking of our operations – how badly or good do we perform? There was no way we could know what equipment experienced re-occurring problems and how many hours of non-production we had... So with a helping hand from the IT people and use of google, we found out how to do it [develop the RUN database]... Here you see [pointing at the screen] – it is interesting to recognize that we had some hours of non-operational time for the last 18 months. And they all happened after a lot of activity had been conducted towards these wells. And both times in Canada. Looks as if there is a correlation. This is a kind of insight we did not previously have. So we got something out of the system [the platform-based enterprise system]. (HMSQ responsible)

Building on the local boundary resources of the new SharePoint 2010 platform and previous experience, users and developers collectively managed to customize a new application by extending the existing system. This new situation also shows that there has been a considerable shift in current work practices of at least project managers since a prerequisite of the new application now widely in use across the Norwegian-part of the organization, was exactly a quite detailed tagging of reports and information. Other users also acknowledged this as they explained to us, that over time, took it for granted that SharePoint requires meta-data – otherwise it will not work. Over the years, these transformations have been followed by a shift in focus as underscored by the previous product owner who had been one of the promoters for organizing documents according to a hierarchical structure:

Nobody could foresee that we would need this more comprehensive solution when we started. Obviously we were still thinking along with the folder-structure as the only way of organizing documents. So we ended up with developing a quasi-solution (Manager and former product owner)

More such leveraging applications were also made for improving workflows and implementing verification procedures for technical drawings, order lists and other official documentation.

5 Modular and architectural acts of customizing and its consequences over time

The Bergen Drilling case illuminate that implementation of enterprise systems such as systems based on various versions of the SharePoint platform is more complex than typically anticipated. By analysing the case we see that various forms of customizing are necessary although the strategy was to implement an out-of-the-box solution. More specifically, based on the case we can distinguish between two types of customizing, namely *architectural acts of customizing* and *modular acts of customizing*².

5.1 Architectural acts of customizing

Based on the case study, we see that the introduction of a new versions of the SharePoint software platform imply an architectural act of customizing. During the first act, the platform-based enterprise system not only had to be customized in terms of functionality, but also required a re-structuring of digital information stored in different existing niche systems, re-configuring the Active Directory, the existing DocuShare system, as well as various scripts.

² The concepts of architectural and modular acts of customizing were main categories coming out of the qualitative analysis of the data material in the case.

Furthermore during the third act, migrations from one version of the SharePoint software platform to another also involved architectural customizing. This type of customizing is hence more related to re-arranging different systems or modules in a wider architecture – and not so much in the specific modules themselves. This is in contrast to existing conceptualizations of customizing where specific functionality and components/modules of one system is involved (e.g. Light, 2001). Since architectural acts of customizing involve many parts of an information infrastructure, this can henceforth have large-scale consequences for work practices and organizing. As expected based on the current literature (Boudreau and Robey, 2005; Pollok and Cornford, 2004; Strong and Volkoff, 2010), this introduction did not immediately lead to any radical transformation of work practices and organizing, but spurred various forms of inertia (e.g. continuing to use legacy systems rather than SharePoint) and workarounds (e.g. not following the intended metadata structures, but reinventing them). Interestingly, the concept of architectural acts of customizing is also similar to what Henderson and Clark (1990) denotes architectural innovation. They use the term to explain a particular type of technological change where the way that different modules in a technology product is re-arranged and connected compared to similar products. Since architectural arrangements are so interconnected with knowledge practices and organizing changes becomes hard to implement in practice. Similarly, in the case of Bergen Drilling since some of the acts of customizing is related to architectural changes in deeply embedded knowledge practices and organizing, they tend to spur workarounds and inertia. However, architectural acts of customizing provide, standardized (local) boundary resources for further modifications and extensions. An architectural act can be directed to several levels of a digital technology. For example, it can be related to both new digital representations of information and the way functionality is distributed across layers (Van Schewick, 2010).

5.2 Modular acts of customizing

On the other hand, modular acts of customizing builds on existing (local) boundary resources and learning, and configures, modifies and/or extends the existing system. Modular acts of customizing are more incremental changes that are based on, or inspired by, existing workarounds or simply through (re-) discovering new software modules provided by the existing platform. Applying Henderson and Clark's definition of the concept again, modular innovations refer to those incremental changes where the basic architecture of the product is stable – but only its modules are changed (Henderson and Clark, 1990).

As the case study shows modular acts of customizing in the second and fourth act were crucial for the organization's adoption of the platform and shaped the work practices of users. Through modular acts of customizing, users and developers in Bergen Drilling utilized the capabilities of the software platform in order to customize the solution further, also after the implementation project was officially done. The modular acts of customizing as substituting the searching functionality and adding the RUN database component were both crucial acts that increased the value of the previous architectural acts.

Software platforms as part of larger ecosystems typically also provide extensive selections of third-party developed modules (Ghazawneh and Henfridsson, 2013). Modular acts of customizing also flourish on (global) boundary resources of the (global) software platforms such as SharePoint.

6 Implications and concluding remarks

By conceptualizing implementation of platform-based enterprise systems as an on-going process of growing an installed base through continuous acts of customizing gives us a more nuanced understanding. In the introduction we stated the main research question was: *how does a platform-based enterprise system evolve through a prolonged implementation process involving customizations and different versions of a software platform?*

Firstly, enterprise systems based on a rich software platform which is part of a larger digital ecology, are not likely to be just installed in one go during the lifetime of a project. Rather, platform-based enterprise systems are grown in a learning-by-trying kind of process involving a long string of various acts of customizing. Even in a case where the aim was to implement the platform-based enterprise system “out-of-the-box”, this implied numerous architectural and singular acts of customizing over several years. This insight is important because in much current literature on customizing (e.g. Light, 2001; Pollock and Cornford, 2004) you get the impression that you can choose not to customize and that customizing is only related to functionality of the enterprise system in isolation. Our case study and theorizing imply that this is not likely to be the case, since implementation would involve architectural acts of customizing. The notion of architectural acts of customizing focuses on how a particular configuration of an enterprise system also implies re-configuring a wider infrastructure. This to some extent explains why many such implementation projects tend to fail in terms of meeting budgets and schedules. In more practical terms, because the change is architectural it involves re-arranging components and stakeholders that are not part of a typical project.

Secondly, with the concepts of architectural and modular customizing it is possible to explain why enterprise systems can have shifting consequences for work practices and organizing over time. Not evident in current literature, is the aspect that the same standardized features of an enterprise system that provides inertia and workarounds, over time also can generate modular acts of customizing extending the enterprise system in unanticipated ways. For example, in Bergen Drilling an application for reporting of operations on wells was established. In turn, these extensions over time helped making the inscribed ways of working and classifying digital content more established, and rendered some of the most salient workarounds. Especially this harmonized work practices in projects were a project manager used the new application to gather new insights on operations – especially those based in Norway, but did not have the same effect on work practices in offices in Australia for example. This can be explained in many ways, but many offices outside Norway tended to have their own systems because the underlying network infrastructure was not scaled for working on SharePoint. This adds to the existing literature on enterprise systems implementation (Boudreau and Robey, 2005; Howcroft et al. 2004; Ignatiadis and Nandhakumar, 2009; Pollock and Cornford, 2004; Williams and Pollock, 2012), and explains how workarounds can over time be curbed. Conceptually, linking architectural acts of customizing with modular acts of customizing provides a plausible explanation for how this process unfolds. What we argue here is not that extensive and diverse workarounds do not occur or immediately disappear – but that through continuous modular acts of customizing over longer periods of time they become less pronounced and to some extent harmonized. Adopting a logic of opposition (Robey and Boudreau, 1999), we are not arguing that there is a stringent cause-effect relationship – but rather that it is possible that more radical

transformations will emerge over time as new singular acts of customizing are conducted and users are increasingly dependent of the enterprise system.

Thirdly, modular acts of customizing are central for the further growing of the enterprise system. Modular acts of customizing are important for building an installed base of users and thereby ‘bootstrapping’ the new solution across the organization (Hanseth and Lyytinen, 2010; Monteiro et al. 2013). Thus, inertia is not only transformed through improvised learning, but also through modular acts of customizing making the platform-based enterprise system more attractive in various ways. For example, as seen in the case by improved features for searching there a change of the search module made it more valuable for a critical mass of users. Especially this is the case with enterprise systems that has collaborative features as Microsoft SharePoint – if only a few users upload documents the solution only becomes extra work for those who choose to upload documents and few advantages for anyone. These more infrastructural aspects of enterprise systems are underestimated in current conceptualizations as well as in practitioners’ approaches for implementing such systems.

To conclude, our study has extensive implications for practitioners. There is not much space do go into details here, but it is evident that organizations should focus more of their resources on planning and conducting small incremental changes after the implementation of a major new enterprise system or version of a platform. Secondly, it is also important to focus more on the processes around architectural alternatives involving different actor with different perspectives.

References

- Boudreau, M. and Robey, D. (2005). “Enacting integrated information technology: a human agency perspective”. *Information Systems Research*, (16:1), pp. 3–18.
- Brehm, L., Heinzl, A. and Markus, ML. (2001) “Tailoring ERP systems: a spectrum of choices and their implications”. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences. IEEE*.
- Ciborra, C. and associates. (2000) *From Control to Drift – The dynamics of corporate information infrastructures*. Oxford University Press, Oxford.
- Edwards, P.N, Jackson, S.J, Bowker, G.C. and Williams, R. (2009) “Introduction: An Agenda for Infrastructure Studies”. *Journal of the Association for Information Systems* (10:5), pp. 364-374.
- Edwards, P. N., Jackson, S. J., Bowker, G. C., & Knobel, C. P. (2007) “Understanding infrastructure: Dynamics, tensions, and design” *Report of a Workshop on “History & Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures”*.
- Eisenhardt, K. M., & Graebner, M. E. (2007). “Theory building from cases: opportunities and challenges”. *Academy of management journal*, (50:1), pp. 25-32.
- Fleck, J. (1994). “Learning by trying: the implementation of configurational technology”. *Research Policy*, (23), pp. 637-652.
- Flyvbjerg, B. (2006). Five misunderstandings about case-study research. *Qualitative inquiry*, 12(2), 219-245.
- Ghazawneh, A. and Henfridsson, O. (2013) “Balancing platform control and external contribution in third-party development: the boundary resources model”, *Information Systems Journal*. (23), pp. 173–192.
- Hanseth, O (2002) “From systems and tools to networks and infrastructures - from design to cultivation. Towards a theory of ICT solutions and its design methodology implications”, available at: http://heim.ifi.uio.no/oleha/Publications/ib_ISR_3rd_resubm2.html, accessed April 4th, 2014.
- Hanseth, O and Lyytinen, K. (2010) “Design theory for dynamic complexity in information infrastructures: the case of building internet”, *Journal of Information Technology* (25), pp. 1–19.

- Henfridsson, O. and Bygstad, B. (2013) "The generative mechanisms of digital infrastructure evolution" *MIS Quarterly*, (37:3), pp. 907-931.
- Henderson, R.M. and Clark, K.B. (1990). "Architectural innovation: The recognition of existing product technologies and the failure of established firms". *Administrative Science Quarterly*, (35: 1), pp. 9-30.
- Hepsø, V., Monteiro, E. and Rolland, K. (2009) "Ecologies of e-infrastructures" *Journal of the Association for Information Systems*, (10:5), pp. 430–446.
- Howcroft, D., Newell, S., and Wagner, E. (2004) "Understanding the contextual influences on enterprise system design, implementation, use and evaluation". *The Journal of Strategic Information Systems* (13: 4), pp. 271-277.
- Ignatiadis, I and Nandhakumar, J. (2009) "The Effect of ERP System Workarounds on Organizational Control: An Interpretivist case study". *Scandinavian Journal of Information Systems*, (21:2), pp. 59-90.
- Kallinikos, J. (2004) "Deconstructing Information Packages: Organizational and behavioural implications of large scale information systems". *Information Technology and People*, (17:1), pp. 8–30.
- Langley, A. (1999) "Strategies for theorizing from process data." *Academy of Management review* (24:4), pp. 691-710.
- Leonard-Barton, D. (1988) "Implementation as mutual adaptation of technology and organization" *Research Policy*, 17, pp. 251-267.
- Markus, M. L., & Tanis, C. (2000) "The enterprise systems experience—from adoption to success" In: *Framing the domains of IT research: Glimpsing the future through the past*, 173, pp. 207-173.
- Monteiro, E., Pollock, N., Hanseth, O., & Williams, R. (2013). "From artefacts to infrastructures". *Computer supported cooperative work (CSCW)*, (22:4-6), pp. 575-607.
- Monteiro, E. and Rolland, K.H. (2012) "Trans-situated use of integrated information systems". *European Journal of Information Systems*, (21:, 608–620 (2012)
- Munkvold, BE., Päivärinta, T., Hodne, AE., and Stangeland, E. (2006) "Contemporary Issues of Enterprise Content Management". *Scandinavian Journal of Information Systems*, (18:2), pp. 69-100.
- Pollock, N. and Cornford, J. (2004). "ERP systems and the university as a 'unique' organization", *IT & People*, (17:1), pp. 31-52.
- Robey, D. and Boudreau, M. (1999) "Accounting for the contradictory organizational consequences of information technology: Theoretical directions and methodological implications", *Information systems research* (10:2), pp. 167-185.
- Rolland, K.H. and Monteiro, E. (2002). "Balancing the Local and the Global in Infrastructural Information Systems". *Information Society*, (18:2), pp. 87-100.
- Sawyer, S., & Southwick, R. (2002) "Temporal issues in information and communication technology-enabled organizational change: Evidence from an enterprise systems implementation". *The Information Society*, (18:4), pp. 263-280.
- Strong, D.M and Volkoff, O. (2010) "Understanding organization–enterprise system fit: a path to theorizing the information technology artifact" *MIS Quarterly* (34: 4), pp. 731-756.
- Urquhart, C. (2013) *Grounded Theory for Qualitative Research*. Sage: Los Angeles
- Van Schewick, B. (2010) *Internet architecture and innovation*. The MIT Press.
- Wagner, E. and Newell, S. (2004) "Best for whom? The tension between "best practice" ERP packages and diverse epistemic cultures in a university context". *J. Strat. Inform. Systems*, (13), pp. 305–328.
- Williams, R. and Pollock, N. (2012) "Research Commentary: Moving Beyond the Single Site Implementation Study: How (and Why) We Should Study the Biography of Packaged Enterprise Solutions". *Information Systems Research*, (23:1), pp. 1–22.