Nora O. Ringdal and Babak A. Farshchian (2022): The Role of Boundary Objects in Platformization Practices: A Case Study of Software Testing. In: Proceedings of the 20th European Conference on Computer-Supported Cooperative Work: The International Venue on Practice-centred Computing on the Design of Cooperation Technologies - Exploratory Papers, Reports of the European Society for Socially Embedded Technologies (ISSN 2510-2591), DOI: 10.48340/ecscw2022_ep07

The Role of Boundary Objects in Platformization Practices: A Case Study of Software Testing

Nora O. Ringdal and Babak A. Farshchian
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
nora.o.ringdal@ntnu.no; babak.farshchian@ntnu.no

Abstract. While digital platforms are frequently investigated at technical, societal, and organizational levels, there are relatively few empirical studies of the collaborative practices that are involved when platforms are introduced into organizations. In this paper, we investigate such practices in the context of a large-scale platformization project within healthcare. Because off-the-shelf platforms already possess a stable core, platformization processes often focus on downstream system development activities such as configuration and testing. Our case study is about one such downstream activity, i.e., that of software testing. We frame software testing as a sociotechnical process involving tacit knowledge from a variety of user groups. We use the theoretical framework of boundary objects to demonstrate how test artifacts – mainly the test versions of the product – function as boundary objects, used to transfer knowledge among platform developers, those who configure the platform, and future users of the platform. Our findings show when and how boundary objects function or break down. We discuss the theoretical and practical implications of our findings both with respect to the boundary objects themselves, and the practices that surround boundary objects to support their collaborative properties.

Introduction

I feel like if I had something to say, I should have said it earlier

The above sentence is by a domain expert we interviewed during our study of a platformization process, i.e., the process of introducing a digital platform in her workplace. *Digital platforms* (Tiwana, 2014) are not only transforming our societies within transportation, tourism, education, finance, healthcare, etc. These platforms are also fundamentally altering our everyday work practices by, e.g., redefining the workplace itself (Ajzen and Taskin, 2021), outsourcing human resource-related decisions to algorithms (Jarrahi et al., 2021), and even redefining our organizational models (Faraj and Pachidi, 2021). With such a broad area of impact, it is difficult to define what a platform is. From a global perspective, platforms overlap with digital infrastructures such as the Internet (Plantin et al., 2018). From an organizational perspective, Gawer (2014) differentiates between *industry* platforms (such as the global platforms of Google and Facebook), *supply chain* platforms (with contractual relations among supply chain members), and *internal* platforms (developed within an organization for internal use). A common denominator for all platform definitions is that a platform consists of a *core* and its *periphery*, which creates a generative power that allows platforms to gradually adapt to different settings (Rodon Modol and Eaton, 2021; Tiwana, 2014). In this paper, we want to focus on a supply chain platform – in the form of an off-the-shelf product – and how it is being implemented in and reshaping an organization, here, a healthcare organization.

This process of *platformization* – which was the setting for the opening quotation from our domain expert in the healthcare organization – is multi-faceted. It is, for instance, studied in depth from a societal perspective as a process of "platform imperialism" driven mainly by American platform companies and their impact on various "spheres of life" such as social and cultural (see e.g., Poell et al., 2019). It is also studied from a technological perspective by, e.g., Bygstad and Hanseth (2019), defined as the process of moving from technological silos to platform-oriented infrastructures that connect the silos and facilitate data sharing. In this paper, we look at platformization *practices*, in particular when supply chain platforms are implemented in organizations. We define platformization as "the process of organizational, social, financial and technological transformation that an organization often must go through to effectively utilize a platform model" (Farshchian et al., 2021). Platformization is, therefore, a process that requires stakeholders to cooperate across professional, social, and/or cultural boundaries. Our case study represents snapshots from such a process, told from the perspectives of those users who are involved in this transforming process.

We have chosen boundary objects (Star and Griesemer, 1989) as our theoretical framework. Boundary object (BO) theory has been used in earlier research to describe system development activities (Doolin and McLeod, 2012) and platformization (Islind et al., 2019), and our study adds to this growing body of literature. As we will demonstrate, BO is a useful concept to study some of the

challenges in platformization processes where there is a strong and established platform core acting as both a liberating and constraining force. This is true for our case, where the platform core is an established product purchased from a world-leading vendor. Framing this core as a BO contrasts the view that BOs should be malleable and negotiable in order to be useful (see e.g., Lee, 2007). We show that boundary objects – particularly the product and its graphical user interface – play a central role as educational objects (Akkerman and Bakker, 2011) and as "wake-up calls" for users to see what is coming. The platform core is therefore useful as a "non-negotiable" BO not only for knowledge transfer purposes but also for motivating users (an effect also observed by Ellingsen and Hertzum, 2019). Our findings also complement the views taken by Islind et al. (2019) and partly by Doolin and McLeod (2012), both studying the design of new systems. Our case – being about an existing system designed by others a long time ago – demonstrates the challenges posed by a non-negotiable or "closed" (Islind et al., 2019) BO during downstream design activities.

Our study also demonstrates the strengths and weaknesses of the BO concept in large-scale system development projects consisting of several organizations and hundreds of participants, where formal and top-down project management regimes can create new settings for a BO. This setting is similar to earlier studies of enterprise resource planning (ERP) software (e.g., SAP, see Schreieck et al., 2021) or global packaged software in general (Pollock et al., 2007). Our research builds on this body of literature and focuses on the practices of user participants during the downstream design activity of testing. There are few studies of testing practices in such settings. We demonstrate how testing becomes a central activity in platformization because – as formulated by our domain expert – many design decisions are already taken a long time ago to build the core of the platform (see also Williams and Pollock, 2012). This means that downstream activities such as testing and training play a potentially important role in platformization because they become the only avenue for users to participate in design. Our findings show that boundary objects – mainly in the form of the software product itself but also test plans, project plans, presentations, etc. – take center stage in testing. Our findings also demonstrate the challenge of cooperating with an international platform provider who becomes a permanent service provider and a powerful part of the ecosystem.

In this paper, we pose the question: What happens in downstream design activities in platformization processes? Our empirical data are from a large ongoing platform implementation project in Central Norway. The project has a complex ecosystem consisting of vendors, customers, and intermediaries, in this way implementing what Gawer (2014) calls a supply chain platform. The findings presented in this paper are preliminary and will be refined and updated as we collaborate with the case organizations and collect and analyze more data.

In the rest of this paper, we first discuss some relevant literature in the areas of platformization, boundary objects, and testing. We then provide a short description

of our case and the case study method we are using. We will then describe some of our findings and discuss their implications for theory and practice.

Background

Platformization

A digital platform (also called a platform in this paper) is defined as a "software-based product or service that serves as a foundation on which outside parties can build complementary products or services" (Tiwana, 2014, p. 5). The "foundation," commonly called the core of the platform, is shared across all the users of the platform and is often difficult to change, i.e., it is "entrenched" (Rodon Modol and Eaton, 2021). The complementary products or services – i.e., the "generative" parts of the platform ecosystem (Thomas and Tee, 2021) – are usually referred to as periphery and are connected to the core using boundary resources (Ghazawneh and Henfridsson, 2013) such as application programming interfaces (APIs). As an example from our case study, the healthcare platform in discussion has a core consisting of an electronic health record (EHR) and a basic set of applications and APIs. This core is then configured and further developed with functionality – i.e., the periphery – needed to support the practices in each specific healthcare organization. Although platformization is a broad concept and contains transformations at different levels and scopes (Farshchian et al., 2021), configuring and augmenting the functionality of the technical platform is a central part of this process and is the focus of our study here. Configuring the platform to fit into the organization can be a challenge for several reasons. First, the organization will need to change to fit to the core functionality of the platform software (Strong and Volkoff, 2010). Second, the organization needs to take advantage of the generativity of the platform's periphery to create innovations in its own context. This is a challenge because "While configuration makes some functionalities easy to set up, it also restricts the space of possible functionalities to those envisioned by the designers of the configuration facility" (Ellingsen and Hertzum, 2019, p. 2). Moreover, platformization is not a linear process with design following use. Due to their broad areas of impact, using platforms leads to organizational change that, in turn, requires reconfiguration of the technological base (Leonardi, 2009).

According to Pollock et al. (2007, p. 1) generic systems such as platforms are "brought into being through an intricately managed process, involving the broader extension of a particularised software application and, at the same time, the management of the user community attached to that solution." The majority of existing studies have investigated the process of platformization on technical, societal, and organizational levels. Fewer studies have explored the collaborative practices that are involved when platforms are introduced into organizations (see e.g., Ellingsen and Hertzum, 2019; Farshchian and Thomassen, 2019; Islind et al.,

2019). Our study builds on these and extends them by looking at downstream system development activities of testing

Software testing as a sociotechnical process

According to Myers et al. (2012, p. 2), software testing is a "process, or a series of processes, designed to make sure computer code does what it was designed to do and, conversely, that it does not do anything unintended." According to some studies, testing can take up to fifty percent of the total development costs (Bertolino, 2007). Numerous classifications of software testing exist. The widely used textbook by Sommerville (2016) defines three major types of testing. *Development testing* is testing done by a system's developers during its initial development to discover bugs and defects. *Release testing*, often performed by a separate group than developers, is about making sure that the system as a whole meets the requirements of the system stakeholders. *User testing*, including acceptance testing, is done by end-users or potential end-users in their own environment to document whether the system is useful and usable, i.e., whether its features are understood and accessible by its end-users and whether it supports end-users to do their job. Recent developments in software engineering, such as agile and continuous development, have blurred the distinctions between these types of tests. This is in particular true for release and user testing, which often happen in parallel.

While the vast majority of software testing research is concerned with the vision of fully automated testing (see e.g., Bertolino, 2007; Orso and Rothermel, 2014), software testing is also regarded as a fuzzy concept, dependent on the sociotechnical surroundings of a software project (Rooksby et al., 2009). This fuzziness seems to increase when testing modern types of software - e.g., platforms and mobile systems - characterized by high levels of heterogeneity and configurability Sommerville (2008). As noted by Orso and Rothermel (2014, p. 125): "testing of these systems is often performed in ad-hoc, inadequate ways, which can have dramatic consequences."

There are at least three aspects of testing that are interesting for us. First, due to platforms becoming off-the-shelf products, downstream system development activities such as configuration and testing play a central role in platformization processes. Testing in such settings is not primarily for fixing software bugs, as noted by Sommerville: "Systems rarely failed in the sense that they crashed or produced clearly incorrect output. Rather, the failures could only be detected by users who understood their local processes and who could identify where system support was inadequate" (Sommerville, 2008, p. 7). In this way, testing can be seen increasingly as a source of new requirements for the next version of the platforms, helping organizations and vendors cross the "implementation line" (Leonardi, 2009). Second, testing is a collaborative organizational practice. Although software testing has traditionally been regarded as a purely technical activity performed mainly by programmers, several researchers emphasize the importance of involving other stakeholders such as marketing and end-users

(Kawalek and Wood-Harper, 2002; Mäntylä et al., 2012; Rooksby et al., 2009). Idrus et al. (2019) argue that software testing heavily relies on business process and system knowledge as a "vast amount of knowledge is captured, used, shared, stored and reproduced throughout the testing activities." Ahonen et al. (2004) show how organizational models of the test organization impact the processes and outcomes of testing. Third, software testing is sociotechnical, and artifact-driven as the subject of testing is often the artifacts – here, boundary objects – that result from earlier system design and development activities (Doolin and McLeod, 2012).

Boundary objects in system development

The concept of boundary object (BO) was first introduced by Star and Griesemer (1989), and has since been used in a wide variety of research areas (see e.g., Lee, 2007). Extant literature has emphasized BOs' importance in collaborating across boundaries and distances – such as organizational, social, and/or cultural – between stakeholder groups or communities (Islind et al., 2019). In the book *Sorting Things Out: Classification and Its Consequences*, Bowker and Star (2000) developed the concept further. Boundary objects are "those objects that both inhabit several communities of practice and satisfy the informational requirements of each of them" (Bowker and Star, 2000, p. 16). Boundary objects are thus "both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites" (Star and Griesemer, 1989, p. 393).

In the development of Information systems (IS), BOs are the artefacts that stakeholders interact with in order to facilitate understanding and cooperation across diverse knowledge domains (Doolin and McLeod, 2012). BOs can be technological artifacts such as product versions. They can also be requirement specifications, project timelines, project management methodologies, workshops, drawings, or paper prototypes (Doolin and McLeod, 2012; Islind et al., 2019; Peer and DiSalvo, 2019).

As highlighted by Levina and Vaast (2005), artifacts only become boundary objects when in use by temporarily creating a bridge between groups of different perspectives. In traditional in-house design and development projects, BOs are often used for mutual learning, seeking to overcome the knowledge barriers between those with the technical knowledge and those with domain specific knowledge (Barrett and Oborn, 2010). Islind et al. (2019) suggest that different stages of a platform design process call for different types of BOs to move the collaboration and co-design process forward. Through the different phases the objects move from "open" to "closed", or unstructured to structured. Open BOs are typically used during the early phase of a design process to give rise to questions and create dialog, while structured objects are introduced later in the process as they are not as easy to change and closer to the final solution.

In platformization processes where a platform is being implemented, the core of the platform is already developed. Boundary objects that represent this core might

therefore seem not as malleable as those in the early stages of system development. Moreover, platformization processes often involve multiple communities of practice – e.g., programmers, healthcare personnel, managers – in large numbers, leading to highly regulated platformization processes. At the same time, while the analysis by Islind et al. (2019) – and many others who have studied BOs in system design – describes how BOs evolve in a phased model of system development, modern iterative methods such as agile (Hoda et al., 2018) are based on the assumption that there is no strict "implementation line" (Leonardi, 2009) dividing design and use. This means that the "closeness" and "openness" of BOs might depend on other factors than their current location in the design process. These factors need to be understood, and we believe employing a practice lens can help us understand and recognize the potential new roles that BOs play in platformization processes.

Case and Method

Case description

In March 2019, the municipality of Trondheim (primary healthcare) and the central Norway health authorities (specialist healthcare) together purchased an Electronic Health Record (EHR) system from the American vendor Epic. This was the culmination of a lengthy process that started ten years earlier with the introduction of the so-called collaboration reform in the Norwegian healthcare (Helse- og omsorgsdepartementet, 2009). Norwegian healthcare is characterized by a strong primary healthcare sector, consisting largely of municipal health services and privately operating general practitioners. The collaboration reform aimed at strengthening primary healthcare even further, moving several duties from hospitals to municipalities. This led to Norway now having one of the most distributed healthcare services in the world (Nylenna, 2020). Due to this distribution – and to prepare for the collaboration reform – the government launched the vision of "One citizen – One medical record" in 2012 (Helse- og omsorgsdepartementet, 2012). By implementing a customized version of Epic's EHR product, locally called Helseplattformen (Healthcare platform, or HP in this paper), the hope is that the almost 40 000 employees in the involved organizations will share healthcare data, in this way improving healthcare services for the citizens.

Immediately after the purchase of the new EHR product, a new organization was also set up (Figure 1). A new local implementation company called Helseplattformen AS – the middle box in Figure 1 – was created, owned by the hospital and the municipality (new municipalities and other primary care actors are also joining the project and taking part-ownership in this company as they join). HP AS has hired tens of Application Analysts (AAs). AAs are often previous healthcare workers who are in charge of the customization process of the EHR product. In addition, the user organizations – e.g., the hospital and the municipality

–have each set up their local implementation projects that are in charge of organizing the local implementation tasks.

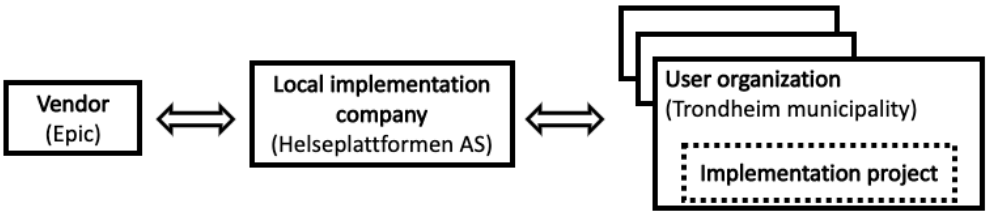

Figure 1. The ecosystem of organizations involved in the digitalization project.

The implementation process for the EHR product consists of six phases as shown in Figure 2. During the specification and development phases, a new version of the EHR product is developed and the assumption is that this new version will fulfill the needs of all the organizations involved. The test and approval phase will test this customized version, and verify its functionality. The study reported in this paper is mainly about the test and approval phase. At the time of writing, all testing is concluded and the training phase has started. The "go-live" date for the product is set for May 1st of 2022.

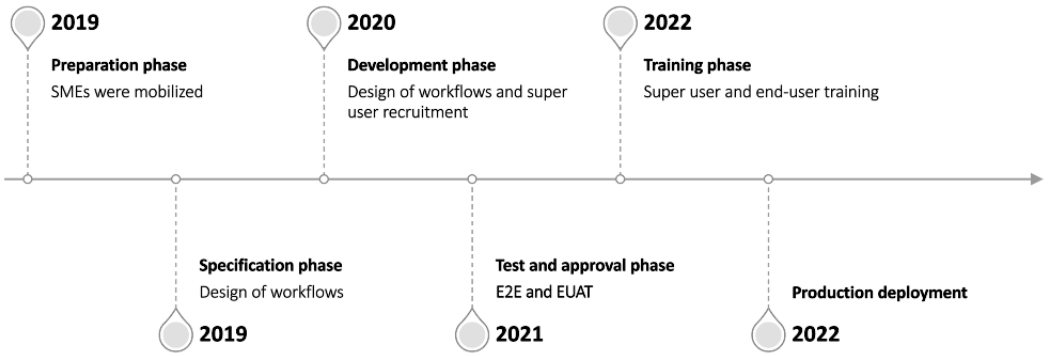

Figure 2. Timeline for the implementation of Helseplattformen.

The above organization and the process of implementing the EHR illustrate a large-scale platformization process as discussed earlier. Early on, it was decided that user participation would be an essential part of the process. This was to avoid past mistakes in e.g., Denmark, where the same product led to a high level of dissatisfaction among healthcare workers after it was implemented. Two types of users have been particularly important for this participation; Subject Matter Experts (SMEs) and Super Users (SUs). Early in the project, the role of SMEs was created to represent domain experts. They have been involved from the preparation phase. More than 400 SMEs, employed in primary or secondary healthcare, have been directly involved in the platform customization during the specification phase. Therefore, it is important to remind the reader that several activities were

conducted with SMEs before the testing phase and that these are not included in the study. In addition, approximately 700 SUs have been recruited to play a key role in training the employees to use the new system.

As discussed previously, testing is a sociotechnical process involving many people in various roles. Three types of tests are in focus in HP (see Table I), End-to-End (E2E) integration tests pass 1 and pass 2, and End-User Acceptance Tests (EUAT). SMEs and SUs have participated in pass 2 of E2E and EUAT. Before the E2E tests, Helseplattformen arranged two separate kick-offs; one for the SMEs and one for the SUs. The aim was to clarify expectations and make them ready to participate in E2E. Trondheim Kommune also arranged a kick-off for the municipal test participants to make sure that they understand their role on behalf of the primary healthcare service. During the kick-off, they were presented with information about how to provide change requests and report on errors. It was also specified that E2E is not training, a demonstration of the solution, or a dedicated test for all possible variations within each integration. The roles of the SMEs and SUs were also explained. The role of the SMEs during E2E is to verify that the adopted solution works, comment on errors and omissions, and reflect on what changes the solution brings to their organization. The SUs are invited, as observers, for two reasons: to build competence before EUAT and to identify changes caused by the solution in their organization. Trondheim municipality also wants to use SUs to identify areas for improvement.

Table I. Overview of the testing activities that involve users in Helseplattformen.

Test	**Purpose**	**Roles**
E2E Pass 1	Make sure that the integrations between applications work before involving the users in order to avoid confusion.	Application analyst (AA), Vendor (V)
E2E Pass 2	Verify the regional solution as agreed upon through requirements and adoption, detect errors so that these can be corrected, and identify change requests and areas of improvement.	Application analyst (AA), Subject matter expert (SME), super user (SU), Vendor (V)
EUAT	The SMEs peform the test to verify that key integrations interfaces are working as expected.	Application analyst (AA), Subject matter expert (SME), super user (SU), Vendor (V)

All tests happen in so-called test sessions organized around a test scenario. A test scenario tells a patient's story and involves several applications, workflows, and relevant integrations. For every scenario, there is a test script that describes how the

procedure is performed. Application analysts from Helseplattformen prepare the test scripts in collaboration with SMEs before the testing. Application analysts are in charge of running and facilitating the test sessions, while representatives from the vendor are present in the sessions to answer questions. Figure 3 shows the different testing activities and the people that are involved in each of them. The total number of participants in each test session varies depending on the number of workflows and integrations involved in the scenario. Due to the pandemic, all the test sessions that we observed were organized and run in Microsoft Teams. After each session, the staff from the implementation project in the municipality would hold debriefing meetings with the SMEs and SUs from the municipality.

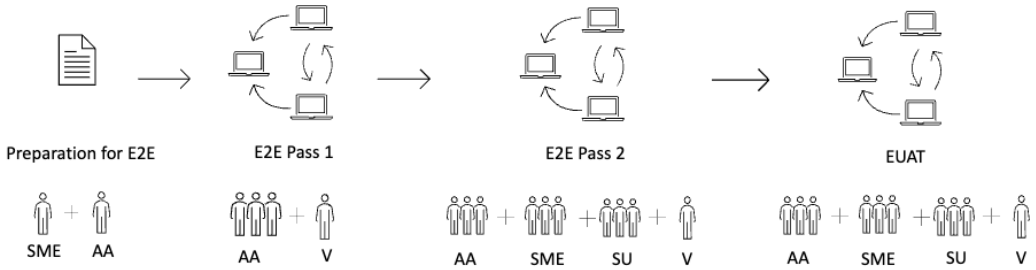

Figure 3. The process of the testing activities in Helseplattformen (see also Table I).

Research method

The strategy to answer the research questions is through an exploratory case study. The research is based on qualitative research methods as the aim is to get an in-depth understanding of boundary objects in the testing phase. Case studies can vary in their approach to time, although this is a short-term study where data was collected over a period of 4 months.

Data was collected through interviews, field observations, and analysis of documents related to the local implementation project in Trondheim municipality. Semi-structured interviews were conducted with stakeholders, including SMEs, SUs, application analysts, and vendor employees. Due to the ongoing pandemic, they were conducted digitally through Microsoft Teams and were, therefore, video recorded by consent. The recordings were then transcribed. Observations were also performed digitally and were documented using extensive field notes. Observations were not recorded.

A qualitative analysis was performed on the data using the qualitative data analysis software, NVivo. The analysis began by identifying codes in the transcriptions, field notes, and documents that were later divided into conceptual categories to identify patterns (Tjora, 2018). These patterns were then used to determine overall themes.

Table II. Data collection methods.

Method	Data
Observation	3 E2E test sessions (8.5 hours in total) 2 Debriefs hosted by the municipality (average of 20 minutes)
Interview	2 SMEs (primary healthcare) 3 Super users (primary healthcare) 2 Application analysts (Helseplattformen AS) 3 Vendor representatives (Epic)
Document analysis	Municipality organizational documentation, public documents, Helseplattformen project documentation, project-related emails, news paper articles

Findings

This section first demonstrates the roles that SMEs and SUs played in testing and what types of breakdowns we observed in their participation in test sessions. We will then present our findings concerning the test version of the products that we observed during the tests and how they supported collaboration.

The role of the end-users in E2E

The goal of E2E testing is to verify the solution as agreed through requirements and adoption. In the SMEs' view, it is difficult to see why they are involved in E2E as they feel it is too late to say something now. However, the vendor gives three reasons why end-users are involved in E2E testing.

> They can say, "I don't think that's right" they can say, "oh, that's confusing, but it's the right thing to do, so I need to make sure that gets into training materials," or "I need to tell my colleague that this is going to be what that is." So that is what we expect from the [SMEs]. (Vendor 1)

According to the vendor, E2E testing is not an activity that end-users are usually involved in, but something they decided to do in this project as a preparation for EUAT.

> The idea is that if end-users are participating in E2E, their comments [and] questions will be addressed earlier than if we waited for End User Acceptance Testing. This is a strategy we have seen before on other installs. (Vendor 1)

Super users are invited to E2E as observers. Involving SUs in E2E is something that the different health authorities and primary healthcare are invited to do. Still, it

is up to them to decide how they want to use this participation in further work within the health authority or the unit. When asked about their role in the test session, one of the super users responded that

> It was to be an observer, so I paid attention to where they [the application analysts] clicked, what they talked about, and then I was asked at the end what I thought about it - how things worked. (Super user 2)

Preparation for E2E

In preparation for E2E testing, SMEs and super users were presented with information about the process and their role through kick-off meetings. All the participants needed to be informed about their role to ensure that testing did not turn into training.

> I have seen this across all the implementations I have worked with, end-users would love to be on the phone the entire time of the implementation, but they also have their own jobs to do. We want to make sure they are still providing patient care, and there is going to be a whole other effort devoted to them getting their training, so making sure that we are using testing to test the system. (Vendor 1)

One of the SUs we interviewed had participated in the kick-off meetings intended to introduce them to the process of E2E. However, it seemed like the information they received led to more confusion than clarity for some.

> During the kick-off, I got the impression - there was kind of two messages [about my role] - one was that I was supposed to be an observer and the other one was that we were supposed to have something to write with and sort of take notes along the way. (Super user 1)

An SME also mentioned that the information was presented in a terminology that they were not familiar with:

> They speak in a language that I don't understand at all with English computer expressions that are completely foreign to me. It might not even be computer expressions, but it's a completely foreign terminology. (SME 2)

The SME, therefore, spent a lot of time in advance to prepare for the test by trying to go through the script several times and look at the workflows that were involved. The foreign terminology used in the kick-off also made it seem like the threshold for making comments during the test was high.

> The threshold for notifying change seemed to be quite high, and they [the test team] talk about systems and ways to notify changes that I have never heard of before, so it gets kind of like s***, how bad does it have to be before one dares to say something? (SME 2)

According to an SME, there was a long sequence at the beginning of the test session where none of the SMEs or SUs dared to say anything, and everyone was "holding their breath" before it gradually became easier to take the floor and speak up.

Breakdowns in the process

The Helseplattformen (HP) project has two process levels: the process introduced by the vendor and the local implementation process (see also Figure 1). Sometimes these are difficult to coordinate. While HP AS arranges the E2E tests, the municipality is invited to involve super users, which leads to these process levels intertwining. The test session did not clear up the uncertainties created by the information received by the end-users in advance. One of the SUs mentioned that the super users were not introduced at the beginning of the test session in the same way as the other participants, which led to further confusion about why they were there.

> There was just a bit of uncertainty about my role and what I was really doing there. Because I sat and took notes, and there was a lot, but then again that might be about content, and that might not be what they were supposed to be testing at that moment. But if that is not what they were testing, there was kind of no need for me to participate. (Super user 1)

After each test session, the SMEs and SUs are invited to a debrief hosted by Trondheim municipality to share their immediate thoughts and experiences. Both SMEs and SUs are also asked to fill in a questionnaire to help them identify changes in their day-to-day work routine using the new solution. Although they express that they find the debrief useful, they are unsure whether the input will be taken further by HP AS and lead to changes. Still, they say that it was nice to be able to share their experiences right away.

During one of the test sessions, an SME discovered that a word that had been translated from English to Norwegian had not been translated to the word that the SME wanted it to be. The SME decided not to say anything about it during the test because it did not have anything to do with the integration with other applications. The questionnaire had no field to report about the translation either, which made the SME unsure of where to provide this feedback. The SME decided to e-mail the test team in Trondheim municipality to ask who should receive such feedback. In the e-mail, the SME stated that there was a need to provide some feedback regarding word choices and some other small changes in the solution and that the SME did not remember how to do it. The SME ended the e-mail by asking, "should I send it to

the application analyst? If so, which one?". After receiving an insufficient response from the municipality, the SME decided to contact one of the application analysts directly. The application analyst thanked the SME for not bringing the questions up during the test and cleared up the SME's uncertainties.

> I could have asked about it during the test, but it didn't have anything to do with the integration between the different applications, so that's why I couldn't bring it up. (SME 1)

According to the SME, a change for that translation was requested a year ago, but the discovery makes the SME uncertain whether it has been implemented since the test version of the product might not include all change requests.

The product in E2E testing

E2E testing in Helseplattformen involves different BOs like the test scripts, test plans, and the version of the product - where the latter seems to be the most important. The E2E test session is the first time the super users see a version of Helseplattformen live. Despite the uncertainty related to their role, all super users reported a growing curiosity and increased motivation to use the new solution after seeing the test version of the product. The super users also thought seeing the application analysts reporting errors so thoroughly was reassuring.

> Now I have seen an example from Helseplattformen, and it looks like we will get a lot of information about a patient on one page. The way the journal system works today, you have to go into separate journal codes to find the information you need. Nothing is gathered in one place. So I am looking forward to getting lots of information gathered in one place. (Super user 2)

Since the scenario that was tested in one of the sessions included both the hospital side and municipal health services, one of the super users mentioned that seeing the product provided them with insight into the information flow between the hospital and the unit.

> For me, it was more insight into what is going on in secondary healthcare, but also the information that is sent from there [the hospital] to the municipality, via the Health and Welfare Office and to the Health Center. So you get some insight into a process that you don't know much about. You know that it is done, but not always what kind of information is sent. Except for a discharge letter or a final note. (Super user 3)

After seeing the product another super user became more curious but states that it may not have been necessary for the super users to attend an entire test session to get the same effect.

> It made me a little curious about what it's like to use it. I got a quick glimpse of the screen, so it's a bit exciting, but I think I would have thought the same even if for example, the leading super user in my department had shown us that we can do this and this [...], and this is what it looks like for us, I think that would have been the same as the test. (Super user 1)

The test is also described as a "wake-up call" by one of the super users saying that

> I believe everyone would benefit from participating in a test like this because [...] the process has lasted for so long, and it is so peripheral to the employees. [...] I am afraid that people forget that this is actually coming. So it is sort of a wake-up call that this is happening and that we just have to keep up. (Super user 2)

Communication through the product

According to a vendor representative, the challenge of communication in a large project appears in the testing phase the most as that is where different pieces of the software and different people are working together really for the first time. An application analyst gives examples of typical comments that emerge during test sessions and says that

> It can be 'why did you press this button and not that one' and the answer is often that it doesn't matter because you are taken to the same [place]. [...] Or it can be that a flow chart is not exactly how it was agreed upon, which might be because it has not been fully developed, but it has been built just enough for it to be tested, so questions, and comments like that may emerge. (Application analyst 1)

During a test, one of the SMEs discovered that there was a person in a patient list that was too young to be in that specific list and asked why that was the case. "It is probably just fill [test data]," the application analyst replied, "but you can press the remove patient button if that is the case." Some of the other end-users comment "good observation" in the chat, to which the SME reply, "well, this is our future, and it is a very realistic situation".

The super user that was taking notes also asked to share a few comments as the end of the test session was approaching, to which the Helseplattformen test team responded, "bring them on". The super user refers to a specific part of the solution and asks whether it is okay to write notes to themselves that are confidential there. An application analyst replies, "yes, but nothing that can't stand the light of day". The super user then asks whether it is okay to write the patient's name or if you have to use a number or ID. "No, you can write their name" the application analyst replies and mention that Helseplattformen is a secure system. One of the test team members proceeded to say that it is common to get a feeling of uncertainty when

seeing the product like this for the first time and that there will be a debrief for the municipality shortly after the test session is finished. The super user responds by saying, "then I will save my comments until then".

Discussion

Our findings demonstrate some well-known challenges related to design artifacts as boundary objects (Doolin and McLeod, 2012; Islind et al., 2019). For instance, we see that BOs, particularly the test product itself, are efficient communication tools. The product engages the users because it communicates to them. Consistent with Islind et al. (2019), the product is a closed BO and triggers more detailed conversations that would likely not have emerged with less concrete BOs. At the same time, due to the fact that our case is about the testing phase of a customized platform, we also see some differences compared to the development of new platforms and systems. These differences relate to both the boundary object itself - here, the test product – and the collaboration processes based on that object.

The role of boundary objects in platform testing

The super users saw the product for the first time in the test sessions. Even though they were confused about what they could comment on and when, all of them reported a growing curiosity and motivation to use the end product. At the same time, this “first meeting” was the source of the confusion; can they comment on the product’s obvious faults that shine through? If yes, how? This situation is illustrative of the discontinuity involved in designing and using large platform products. The BO – the product – is closed, but it was not closed by these users. It was closed mainly by users in earlier projects in USA and the SMEs and developers in earlier phases of this project. While the product as a BO invites modifications – because it is so material and easy to understand and scrutinize by the super users – it is also “protected” by contracts and change management regimes in the project. SMEs who participated in the earlier phases of the project report similar issues with what they could comment on (Zahlsen et al., 2020).

Some of the confusion we observed was related to the fact that in large development projects, the product is continuously being developed and modified – i.e., the BO might be perceived as closed by end-users but is still open for others in the project. The end-users often do not know how mature or complete a product is (a point also raised by Dourish et al., 2020). This might mean that a change that is seen as important by a super user is regarded as negligible by an application analysis or the vendor. Moreover, the platform is a special type of layered BO because it consists of parts that can be changed easily, parts that will cost some resources to change, and parts that cannot practically and financially be changed in one project. However, for users, it is difficult to know which is which. This layered nature leads to a knowledge management problem. Often the distance between those who know the different layers of the BO is so large that it is practically not

feasible to have all of them in the same room. Local discussions can therefore become confused and error-prone as those participating do not know the BO well enough.

Our findings suggest that the phased "closing" process of BOs, as described by Islind et al. (2019), might not be suitable for large-scale platformization projects. This closing process is often punctuated in platformization processes by the passing of time and the involvement of vastly different users in the various phasing of the closing process (Pollock et al., 2007). BOs in use are often understood as "technical" by different users, meaning that they are built using the language of another community of practice. In our case, this was exacerbated by the fact that we are dealing with an American product being adapted to a Scandinavian setting. While in smaller projects developing new platforms, such "technicalities" can be sorted out by the group, in large and longitudinal projects, they often do not get solved immediately and carry on creating confusion in the cooperation. This is partly because the BO can only be modified through a protective shield of "change management" tools and processes, making the threshold to resolve inconsistencies very high. Even if some local understanding is created by some people using ad hoc means, the longevity of the project means that those people are replaced by new ones in the next phase, and the original or new misunderstandings appear again.

Traditionally product testing is regarded as the end-point of a long and laborious design and development process. This is the consequence of following a waterfall development process – often mandated by regulations surrounding public procurement. At the same time, the testing phase is often the first time users see the product. While the vendor is interested in "closing" the BO and signing the final contract papers, the users start their journey first during the product testing sessions. When the end-users get involved, the product becomes the main BO to create a shared understanding. As mentioned by a vendor representative, the challenge of communication in a large project appears in the testing phase the most as this is where the different pieces of the software and the different communities of practice are brought together for the first time. In this sense, product testing needs more attention as a practical and central tool to cross the "implementation line" and open up the black box of needed functionality (Leonardi, 2009).

Collaborative processes surrounding the boundary objects

Due to the size of the project and the number of people involved, access to the BO – i.e., the test product – was controlled by strict and formal processes. These were standardized processes introduced into the project by the vendor. Various kick-off meetings and workshops were held to train the SMEs and super users in the processes. Despite the training, these processes and the associated roles were not understood well by those taking on these roles. The introduction of formal processes created confusion about what could be done with the BO and when. For instance, the SMEs often did not know whether they could comment on the BO directly

during the test session or whether it was "too late." When the user representatives went outside the formal process, e.g., sending emails directly to the test session facilitators, it was not clear what happened to such communication. This, in turn, led to more insecurity about the role of the user representatives in the process.

The ownership of the process by the vendor made it hard for local user organizations to change it to fit their needs. At the same time, valuable knowledge from local processes was not integrated into the formal process and seemed to get lost. Local processes such as the debrief meetings in the municipality often created a more trustful environment for the exchange of ideas. However, we observed numerous breakdowns in the transfer of knowledge from these local processes to the project-wide formal processes. Moreover, although the municipality had implemented additional local processes, these processes were sometimes influenced by the project-wide processes. For instance, the questionnaires that were distributed by the municipality among their SMEs and super users did not open for commenting the BO – even if the debrief meetings demonstrated that there still was a need for it locally. Instead, the questionnaires asked the user representatives to only comment on how they should change their own work processes to accommodate for the new platform. We believe the formal processes in the project could have benefited greatly by taking into account local processes as valuable input to the project.

In large-scale platformization processes, BOs are surrounded by numerous restrictions. Adapting BOs in these projects becomes a costly task. Instead of doing quick changes locally, the project team often needs to propagate the change back to a platform core owner, who needs to calculate the impact of the change for all users of the core and calculate the cost of the change for this customer. Changes are furthermore regulated by formal contracts, which restrict the allowed cost. This means that changes that make sense in a local context are left in a "product backlog" and may not be relevant once the project has moved on and the level of knowledge and maturity has changed, making these changes irrelevant or even harmful. This is far from what happens in small-scale new platform development as reported in e.g., Farshchian and Thomassen (2019) and Islind et al. (2019). Once more, product testing has greater potential than being regarded as the last phase in a project. Regarding testing as a sociotechnical process and the test product as a BO early on can provide a valuable vehicle not only for requirements but also for training and preparedness.

Conclusion

Boundary objects play a central role in testing as educational and motivational objects as well as "wake-up calls" for new users. The product as a BO creates enthusiasm among the user representatives. As opposed to other more abstract BOs such as plans, test scripts, and presentations, the test product seems to be much easier to relate to own work and plays a central role in creating reactions. Lack of local ownership and strict, formal processes reduce a BO's spanning properties. It

is easier for user representatives to feel safe in more local meetings involving other municipality employees or preferably their own unit. Useful information about the process and the BO emerge in such meetings, which might mean that the process of testing should have been more distributed and performed locally by local SMEs and super users. Giving user representative earlier access to the product by starting product testing earlier, and establishing local ownership of the testing processes, can help reduce various risks that often face large-scale platformization projects.

Our data collection and analysis should be considered as ongoing. This paper presents the results of a first round of thematic analysis, and an initial theoretical framing. The hope is therefore that this paper will provide us the opportunity to present and receive feedback on this work in progress.

Acknowledgements

We thank employees from the municipality of Trondheim, Helseplattformen AS, and Epic Systems Corporation for providing us the opportunity to observe and interview them, and for their feedback on earlier versions of this paper. We also thank the members of Forskerfabrikken at NTNU and the anonymous reviewers of ECSCW for their comments, which helped improved this paper. The research reported here was partly funded by the PlatVel project (www.ntnu.edu/platvel).

References

Abraham, R. (2013): 'Enterprise architecture artifacts as boundary objects-A framework of properties'. In: *21st European Conference on Information Systems, ECIS 2013*.

Ackerman, M. S., J. Dachtera, V. Pipek, and V. Wulf (2013): 'Sharing knowledge and expertise: The CSCW view of knowledge management'. *Computer Supported Cooperative Work (CSCW)*, vol. 22, no. 4-6, pp. 531–573.

Ahonen, J. J., T. Junttila, and M. Sakkinen (2004): 'Impacts of the organizational model on testing: Three industrial cases'. *Empirical Software Engineering*, vol. 9, no. 4, pp. 275–296.

Ajzen, M. and L. Taskin (2021): 'The re-regulation of working communities and relationships in the context of flexwork: A spacing identity approach'. *Information and Organization*, vol. 31, no. 4, pp. 100364.

Akkerman, S. F. and A. Bakker (2011): 'Boundary crossing and boundary objects'. *Review of educational research*, vol. 81, no. 2, pp. 132–169.

Barrett, M. and E. Oborn (2010): 'Boundary object use in cross-cultural software development teams'. *Human Relations*, vol. 63, no. 8, pp. 1199–1221.

Bertolino, A. (2007): 'Software testing research: Achievements, challenges, dreams'. In: *Future of Software Engineering (FOSE'07)*. pp. 85–103.

Bowers, J. M. (1991): 'The Janus Faces of Design: Some Critical Questions for CSCW'. In: J. M. Bowers and S. D. Benford (eds.): *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*. Amsterdam, etc., pp. 333–350, North-Holland.

Bowker, G. C. and S. L. Star (2000): *Sorting things out: Classification and its consequences*. MIT press.

Bygstad, B. and O. Hanseth (2019): *Transforming Digital Infrastructures through Platformization*. Association for Information Systems.

Cordella, A. and A. Paletti (2019): 'Government as a platform, orchestration, and public value creation: The Italian case'. *Government Information Quarterly*, vol. 36, no. 4, pp. 101409.

Dahl-Jørgensen, T. C. and E. Parmiggiani (2020): 'Platformization of the public sector: Assessing the space of possibility for participation'. In: *PDC'20: Participatory Design Conference 2020-Participation Otherwise*.

De Reuver, M., C. Sørensen, and R. C. Basole (2018): 'The digital platform: a research agenda'. *Journal of Information Technology*, vol. 33, no. 2, pp. 124–135.

de Reuver, M., C. Sørensen, and R. C. Basole (2018): 'The Digital Platform: A Research Agenda'. *Journal of information technology*, vol. 33, no. 2, pp. 124–135.

Doolin, B. and L. McLeod (2012): 'Sociomateriality and boundary objects in information systems development'. *European Journal of Information Systems*, vol. 21, no. 5, pp. 570–586.

Dourish, P., C. Lawrence, T. W. Leong, and G. Wadley (2020): 'On being iterated: The affective demands of design participation'. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. pp. 1–11.

Drewel, M., L. Özcan, C. Koldewey, and J. Gausemeier (2021): 'Pattern-based development of digital platforms'. *Creativity and Innovation Management*, vol. 30, no. 2, pp. 412–430.

Ellingsen, G. and M. Hertzum (2019): 'User participation in the implementation of large-scale suite systems in healthcare'. In: *Infrahealth 2019-Proceedings of the 7th International Conference on Infrastructures in Healthcare 2019*.

Faraj, S. and S. Pachidi (2021): 'Beyond Uberization: The co-constitution of technology and organizing'. *Organization Theory*, vol. 2, no. 1, pp. 2631787721995205.

Farshchian, B. A., M. Grisot, S. Hochwarter, A. Sigríður Islind, M. Mikalsen, E. Parmiggiani, and P. Vassilakopoulou (2021): 'Platform-Supported Cooperative Work'. In: *Proceedings of 19th European Conference on Computer-Supported Cooperative Work*.

Farshchian, B. A. and H. E. Thomassen (2019): 'Co-creating platform governance models using boundary resources: A case study from dementia care services'. *Computer Supported Cooperative Work (CSCW)*, vol. 28, no. 3, pp. 549–589.

Gawer, A. (2014): 'Bridging differing perspectives on technological platforms: Toward an integrative framework'. *Research policy*, vol. 43, no. 7, pp. 1239–1249.

Gawer, A. (2022): 'Digital platforms and ecosystems: remarks on the dominant organizational forms of the digital age'. *Innovation*, vol. 24, no. 1, pp. 110–124.

Gerson, E. M. and S. L. Star (1986): 'Analyzing due process in the workplace'. *ACM Transactions on Office Information Systems*, vol. 4, no. 3, pp. 257–270.

Ghazawneh, A. and O. Henfridsson (2013): 'Balancing platform control and external contribution in third-party development: the boundary resources model'. *Information systems journal*, vol. 23, no. 2, pp. 173–192.

Helse- og omsorgsdepartementet (2009): 'Samhandlingsreformen: rett behandling - på rett sted - til rett tid'. https://www.regjeringen.no/no/dokumenter/stmeld-nr-47-2008-2009-/id567201/.

Helse- og omsorgsdepartementet (2012): 'Én innbygger – én journal'. https://www.regjeringen.no/no/dokumenter/meld-st-9-20122013/id708609/.

Helseplattformen (2019a): 'Bakgrunn for Helseplattformen'. https://helseplattformen.no/bakgrunn-for-helseplattformen.

Helseplattformen (2019b): 'Prosjektet'. https://helseplattformen.no/om-oss/prosjektet.

Hoda, R., N. Salleh, and J. Grundy (2018): 'The rise and evolution of agile software development'. *IEEE software*, vol. 35, no. 5, pp. 58–63.

Huvila, I., T. D. Anderson, E. H. Jansen, P. McKenzie, L. Westbrook, and A. Worrall (2014): 'Boundary objects in information science research: An approach for explicating connections between collections, cultures and communities'. *Proceedings of the American Society for Information Science and Technology*, vol. 51, no. 1, pp. 1–4.

Idrus, H. M. et al. (2019): 'Tacit Knowledge in Software Testing: A Systematic Review'. In: *2019 6th International Conference on Research and Innovation in Information Systems (ICRIIS)*. pp. 1–6.

Islind, A. S., T. Lindroth, J. Lundin, and G. Steineck (2019): 'Co-designing a digital platform with boundary objects: bringing together heterogeneous users in healthcare'. *Health and Technology*, vol. 9, no. 4, pp. 425–438.

Jarrahi, M. H., G. Newlands, M. K. Lee, C. T. Wolf, E. Kinder, and W. Sutherland (2021): 'Algorithmic management in a work context'. *Big Data & Society*, vol. 8, no. 2, pp. 20539517211020332.

Johansen, R. (1988): *Groupware. Computer Support for Business Teams*. New York and London: The Free Press.

Kawalek, P. and T. Wood-Harper (2002): 'The finding of thorns: user participation in enterprise system implementation'. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, vol. 33, no. 1, pp. 13–22.

Krabbel, A. and I. Wetzel (1998): 'The customization process for organizational package information systems: A challenge for participatory design'. In: *Proceedings of the PDC*, Vol. 98. pp. 45–54.

Lee, C. P. (2007): 'Boundary negotiating artifacts: Unbinding the routine of boundary objects and embracing chaos in collaborative work'. *Computer Supported Cooperative Work (CSCW)*, vol. 16, no. 3, pp. 307–339.

Leonardi, P. M. (2009): 'Crossing the implementation line: The mutual constitution of technology and organizing across development and use activities'. *Communication Theory*, vol. 19, no. 3, pp. 278–310.

Levina, N. and E. Vaast (2005): 'The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems'. *MIS quarterly*, pp. 335–363.

Mäntylä, M. V., J. Itkonen, and J. Iivonen (2012): 'Who tested my software? Testing as an organizationally cross-cutting activity'. *Software Quality Journal*, vol. 20, no. 1, pp. 145–172.

Myers, G. J., C. Sandler, and T. Badgett (2012): *The art of software testing*. Hoboken, N.J: John Wiley & Sons, 3rd ed edition.

Nicolini, D., J. Mengis, and J. Swan (2012): 'Understanding the role of objects in cross-disciplinary collaboration'. *Organization science*, vol. 23, no. 3, pp. 612–629.

Nylenna, M. (2020): *Helsetjenesten i Norge: et overblikk*. Oslo: Gyldendal, 2. utgave. edition.

Oates, B. J. (2006): *Researching information systems and computing*. London: Sage Publications.

Orso, A. and G. Rothermel (2014): 'Software Testing: A Research Travelogue (2000–2014)'. In: *FOSE 2014*. New York, NY, USA, p. 117–132, Association for Computing Machinery.

Peer, F. and C. DiSalvo (2019): 'Workshops as Boundary Objects for Data Infrastructure Literacy and Design'. In: *Proceedings of the 2019 on Designing Interactive Systems Conference*. pp. 1363–1375.

Plantin, J.-C., C. Lagoze, P. N. Edwards, and C. Sandvig (2018): 'Infrastructure studies meet platform studies in the age of Google and Facebook'. *New media & society*, vol. 20, no. 1, pp. 293–310.

Poell, T., D. Nieborg, and J. Van Dijck (2019): 'Platformisation'. *Internet Policy Review*, vol. 8, no. 4, pp. 1–13.

Pollock, N., R. Williams, and L. D'Adderio (2007): 'Global software and its provenance: generification work in the production of organizational software packages'. *Social studies of science*, vol. 37, no. 2, pp. 254–280.

Rodon Modol, J. and B. Eaton (2021): 'Digital infrastructure evolution as generative entrenchment: The formation of a core–periphery structure'. *Journal of Information Technology*, vol. 36, no. 4, pp. 342–364.

Rooksby, J., M. Rouncefield, and I. Sommerville (2009): 'Testing in the wild: The social and organisational dimensions of real world practice'. *Computer Supported Cooperative Work (CSCW)*, vol. 18, no. 5-6, pp. 559.

Schreieck, M., M. Wiesche, and H. Krcmar (2021): 'Capabilities for value co-creation and value capture in emergent platform ecosystems: A longitudinal case study of SAP's cloud platform'. *Journal of Information Technology*, vol. 36, no. 4, pp. 365–390.

Sommerville, I. (2008): 'Construction by configuration: Challenges for software engineering research and practice'. In: *19th Australian Conference on Software Engineering (ASWEC 2008)*. pp. 3–12.

Sommerville, I. (2016): *Software engineering*. Pearson, 10th ed., global ed. edition.

Star, S. L. and J. R. Griesemer (1989): 'Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39'. *Social studies of science*, vol. 19, no. 3, pp. 387–420.

Strong, D. M. and O. Volkoff (2010): 'Understanding Organization—Enterprise system fit: A path to theorizing the information technology artifact'. *MIS quarterly*, pp. 731–756.

Thomas, L. D. and R. Tee (2021): 'Generativity: A systematic review and conceptual framework'. *International Journal of Management Reviews*.

Tiwana, A. (2014): *Platform Ecosystems: Aligning Architecture, Governance, and Strategy*. Morgan Kaufmann.

Tjora, A. (2018): *Qualitative research as stepwise-deductive induction*. Routledge.

Trondheim kommune (2021): 'Helseplattformen'. https://www.trondheim.kommune.no/helseplattformen/.

Vassilakopoulou, P., M. Grisot, T. Blegind Jensen, N. Sellberg, J. Eltes, A. A. Thorseng, and M. Aanestad (2017): 'Building national ehealth platforms: the challenge of inclusiveness'. In: *Proceedings of the International Conference on Information Systems-Transforming Society with Digital Innovation, ICIS 2017*.

Williams, R. and N. Pollock (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*, vol. 23, no. 1, pp. 1–22.

Zahlsen, Ø. K., D. Svanæs, A. Faxvaag, and Y. Dahl (2020): 'Understanding the Impact of Boundary Conditions on Participatory Activities.'. In: *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*. pp. 1–11.

Øverby, H. (2018): 'Digital economics : how Information and communication technology is shaping markets, businesses, and Innovation'.