

Why Managers Don't Manage: An Interpretive Case Study of Enterprise Systems Adoption Management

Cecilie Caroline Falch
IT University of Copenhagen
cefa@itu.dk

Oliver Krancher
IT University of Copenhagen
olik@itu.dk

Abstract

Although there is a large amount of research on adoption and benefits management, we know surprisingly little about how and why managers manage adoption of enterprise systems. By conducting an interpretive case study, we explore managers' perceptions of adoption management. Through semi-structured interviews, we uncover that two levels of management expressed surprisingly little action to influence users' adoption towards desired benefits once the application is live. The data analysis found a mental rationalisation pattern that justified the managers' (lack of) adoption action. We identify these ostensive and performative structures as a project-centric mindset to highlight how the rigid boundaries of project completion are prioritised, thus challenging holistic adoption management.

Keywords: Adoption Management, Case Study, Project Thinking, Management Actions, Management Justification.

Introduction

The study of user adoption is unarguably one of the most represented literature topics in Information System (IS) research. However, organisations still struggle to manage adoption towards desired benefits. The resources allocated to IT projects continue to rise, but reports show that on average they return 56% less value than anticipated (Bloch et al., 2012). In the quest to understand the challenges organisations encounter when managing user adoption, we need to differentiate between voluntary and mandatory use environments. Our study focuses on the mandatory use context of enterprise software. Mandatory environments have been shown to be substantially different from voluntary use contexts. When software is voluntary, user attitudes and intention to use have been proven to correlate with actual usage, however as enterprise software is mandated by the organisation, intentions contribute with very little explanatory power (McNally & Griffin, 2010; Bhattacharjee et al., 2018; Beaudry et al., 2020; Nah et al., 2004; Brown

et al., 2002; Koh et al., 2010). Thus, for enterprise software, user adoption embodies a more ambiguous and dynamic behaviour that calls for a different management approach (Seo et al., 2011; Van Offenbeek et al., 2013). Mintzberg's (1971) foundational work on managerial work has shown that it is imperative to understand managers' working processes to improve them. It is then surprising that despite the vast amount of literature on user adoption (Venkatesh et al., 2016; Beaudry & Pinsonneault, 2010) and on benefits management (Ward et al., 1996; Ashurst et al., 2008; Holgeid et al., 2022), relatively little work has been done to understand how managers manage adoption in mandatory environments. Ultimately, it implies that there is a need to know more about what managers do when they manage adoption of enterprise software.

This case study departs from the need to understand perceptions of adoption management and to uncover challenges management faces in practice. This is also to further explore the boundaries of adoption management and explain current obstacles. The case study explores the following research questions: (1) *How do managers perceive adoption management and their actions towards managing adoption?* and (2) *How do managers rationalise and justify their adoption management actions?* In the remainder of this paper, we review literature on adoption, benefits management, and organisational routines, describe our methods and results of an interpretive case study at three organizations, and discuss implications. Our key contribution lies in identifying managerial interpretations and rationalisations that explain why managers may devote limited efforts to managing adoption.

Background Literature

There is very little explicit focus on adoption management in literature and there is a need to delimit the phenomenon in IS terminology. Following a recent literature (Falch & Krancher, 2022), we define adoption as when a system enables usage that solves a given problem with a valuable outcome. We distinguish two broad literature streams that intersect aspects of this definition:

the acceptance and resistance literature and the project and benefits management literature.

Project and Benefits Management

Benefits management as a discipline is largely built on the early work of Ward et al. (1996) where they presented the process model of benefits management that has shaped much of later research. The paper argues that IS/IT does not deliver benefits on its own, but it is through a comprehensive management process that organisations can ensure that desired benefits from IT investments can be realised. The phases of the iterative model include identifying benefits, planning the realisation, executing the plan, evaluating, and analysing the potential for further benefits (Ward et al., 1996). Scholars and professional organisations such as the Project Management Institute have built upon this model and developed similar frameworks that underpin the importance of a structured and often iterative approach (PMI, 2016; Peppard et al., 200; Holgeid et al., 2022; Ashurst et al., 2008). The common pattern among the benefits management literature is that the papers are largely prescriptive and rarely focus on how and why managers should manage user responses to align system usage and business benefits, thus not focusing on adoption. Moreover, in line with the notion of projects as a set of time-limited one-off activities (Cadle & Yeates, 2008; Ashurst et al., 2008), the project and benefits management literature emphasises processes and practices that ensure “benefits are realised as project implementation progresses and finishes”, though some frameworks also acknowledge the importance of sustaining benefits after a project’s end (PMI, 2016).

Acceptance and Resistance

For what can be characterised as acceptance and resistance literature, it is commonly known that research in this field either seeks to explain acceptance (Davis, 1987; Ginzberg 1981; Beaudry & Pinsonneault, 2010) or reduce resistance in users (Hirschheim & Newman, 1988; Kim & Kankanhalli, 2008). Recently, research has studied acceptance and resistance simultaneously to accommodate the complex and paradoxical user behaviour of mandatory system usage. Here, studies are challenging the notion that by obtaining usage, the organisation is also guaranteed to satisfy user needs (Beaudry et al., 2020; Bhattacharjee et al., 2018). Most of the literature is explanatory and/or predictive in nature for a wide variety of mediators e.g., perceived switching cost (Polites & Karahanna, 2012), that is argued to align system usage and user behaviour. However, although this research points to some managerial interventions (e.g., training, deliberate change management initiatives) for

promoting adoption, this research has not explored the full repertoire of actions that managers take to manage adoption, nor the reasons why managers take these actions.

While observing the nature of these two literature streams, it becomes evident that three knowledge gaps exist. First, where benefits management literature provides a tangible management aspect to aligning system implementation and business benefits, the adoption literature provides explanatory and predictive models of how to align system usage and user behaviour. Thus, each literature provides insight into the concept of adoption management, but there seems to be a missing link that holistically connects system implementation, user behaviour and business benefits. Second, we know little about the concrete actions that managers take to manage adoption and about the reasons why they take these actions. Notwithstanding the value of prescriptive frameworks, organisations implementing these frameworks may not gain the expected results without understanding the reasons why managers may or may not use these frameworks or take other actions to manage adoption in practice. Third, both literature streams focus on structures and activities during the implementation phases of projects, thus taking a project perspective. Yet, such a project perspective is increasingly challenged as organisations move from project to product organisations (Wiedemann et al., 2020) and realise that environments may change and new opportunities for creating value from systems arise over the lifetime of a system (Strong et al., 2014; Orlikowski, 1996).

Organisational Routines

As we collected and analysed our data, the theory of routines (Feldman and Pentland, 2003) emerged as a helpful lens for making sense of and organising the emerging concepts. Routines are patterns of recurrent, interdependent actions in organisations. As such, managers’ actions of monitoring and following up on adoption can be seen as a routine to the extent that there are recurrent patterns in these actions. In their theory, Feldman and Pentland build on Latour’s (1986) distinction between ostensive and performative aspects of practice. While the ostensive aspect refers to ideas about a routine, the performative aspect refers to concrete actions that actors take in concrete instances of the routine (Feldman and Pentland, 2003). Drawing on Latour (1986) and ethnographic research (Feldman, 2000), Feldman and Pentland theorise the reasons for stability and change in routines. They argue that the ostensive aspect legitimises and guides the performative, while the performative creates, reinforces, and modifies the ostensive.

Method

We used an interpretive case study (Walsham, 1995) approach to understand how managers perceive adoption management and their actions related to the practice. We chose the interpretive approach because it allowed us to explore the emergence of manifestations and cognitive patterns that are deeply embedded in the subjective understanding of the interviewees' work. Moreover, we chose case study as our research method because it was well suited for exploring an issue that is little understood (Walsham, 1995).

We selected cases that met the following criteria: (1) the company must employ more than 1000 employees, (2) the company must be ready to provide insight to an enterprise software application with more than 100 users, and (3) the application must have been live for more than one year. This enabled us to analyse case companies that would be considered larger enterprises and discuss implementation projects where the software had been operational for a significant number of users with enough time for managers to influence adoption post-implementation. The research departs from three Danish companies that adhere to these criteria. The number of employees in each of the three companies ranges from approx. 1500 to approx. 60.000. The number of users for each of the three applications ranges from approx. 300 to approx. 16.000. The first company is a major Danish retail company. We chose their recently completed implementation of a product-return system as the case project. The second company was also founded in Denmark but has a large international sale and production chain. As a manufacturing company, they proposed their large implementation case of a new machine maintenance system. Finally, with more than 80 vessels operating in the shipping industry, the third company's case project departed from its long-running enterprise procurement system that continues to experience adoption obstacles. In all three cases, we interviewed two levels of management that had been involved in the implementation or are currently responsible for the case project's software application. These roles included top management and the product owner. Furthermore, we interviewed one user for each application to compare the alignment between management perceptions and users' reality. Though each specific job title varied across the companies, we utilise these more discrete categories to illustrate the hierarchical order and still maintain anonymity.

Prior to the interviews, the participants were told that the focus of the study was to understand user reactions, adoption outcomes, and the actionable management of adoption and desired benefits. The participants were motivated to contribute as they had a clear interest in the topic of adoption and managing towards benefits.

They felt the topic was highly relevant and were eager to reflect on their experiences.

Data Collection

Our data sources were interviews with semi-structured focus questions (Charmaz, 2006). We began with questions about the project's status and how it was perceived by the interviewees. As we followed the semi-structured interview approach some questions varied, but the core aim remained the same: to understand management actions and the subjective structures that drove those actions. The key interview questions are shown in Table 1. Since the focus of our study was on managers' interpretations and rationalisations, we preferred interviews as our key data source over alternative data collection strategies like archival data.

Would you say that the users have accepted the application? – Why?
Would you say that the users are compliant with how to use the application? – Why?
Would you say that the application has been adopted? – How do you know?
What actions have you applied post-implementation to manage adoption?

Table 1: Overview of Key Interview Questions

Data Analysis

We applied data analysis techniques and coding strategies known from grounded theory (Charmaz, 2006) for our analysis as we began to make analytic sense of the participants' experiences and the meanings and actions behind them. First, all interviews were transcribed, and those interviews conducted in Danish were transcribed by the authors to English. Second, the findings were revealed through the phases of initial and focused coding. First, we coded the quote sections that helped us to understand how adoption was perceived, what actions management applied to influence the application use after going live, and how they measured or verified their claim that they were successful or not. Within these sections of quotes there began to emerge a pattern of themes, and we moved to focused coding. Here, we applied the most useful initial codes and began applying them consistently across all interview transcripts. The codes included: *usage is adoption, no usage no job, measure mistakes, no structured approach, no news is good news, management justification, project thinking, no baseline to compare, it lives let go, how do you know you are right* and *status check-ins*. As we began to integrate these codes some became more intertwined and could be combined, whilst two natural categories appeared. In the analysis of these categories, we

found that the tangible actions that management applied post-implementation named *management actions* were portrayed in the codes: *measure mistakes*, *no structured approach* and *it lives, let go*. On the other hand, we found that the rationalisation of these actions could fall under the umbrella term *management justification* and include the codes: *usage is adoption*, *no usage no job*, and *no news is good news*. After focused coding, we moved to theoretical coding (Charmaz, 2006). In this stage, Latour’s distinction between the ostensive and the performative emerged as a theoretical description (Bygstad et al., 2016) of our findings. In the next section, we present these findings in detail.

Findings

In our three cases, managers expressed that they took surprisingly little action to manage post-implementation adoption. The inductive examination of the data revealed that managers justified the lack of initiatives based on a pattern of cognitive perceptions regarding adoption. The findings show that both top management and product owners struggle to pinpoint focused attempts to influence user adoption to ensure system usage that solves a defined problem due to a linear and rigid perception of enterprise software adoption. The data analysis revealed that the mental perceptions of adoption and adoption-focused actions are closely intertwined as there seems to exist an interdependent relation through a rationalisation process. The code categories, corresponding sub-codes, and the interrelation between them are represented in Figure 1 below.

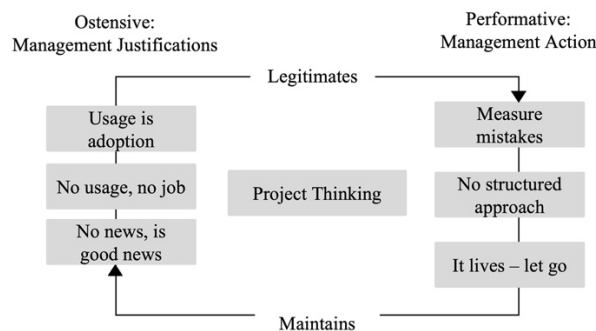


Figure 1: Conceptual Model of Project Thinking Explained by Management Justifications and Management Actions

Looking at the nature of these two categories, we found it relevant to draw on Feldman and Pentland’s (2003) theory of routines. We argue that there is a natural interdependency between managers’ concrete actions (i.e., performative aspect) and the cognitions that justify these actions (i.e., the ostensive aspect). We argue that the way management perceives adoption and

acts accordingly can be characterised as a project-thinking mindset. By utilising this label, we make connotations to how it seems in the interviews that project completion is prioritised over the final product and evaluating the value it created. We argue that when reading the model, there is a constant alternation in the legitimising and maintaining structures across the code categories. We do not describe a linear relationship between e.g., *usage is adoption* and *measure mistakes*. Instead, we wish to emphasise that e.g., the code *usage is adoption* also cross-legitimise the code *no structured approach*. It is important to highlight these cross-sectional notions to underline the socio-technical complexity of how adoption management is perceived. In the following section, we describe the foundation for each code and how the dynamics between the two sides of the model were demonstrated during the interviews and can be characterised as what we label project thinking.

Management Justifications

As we focus on the first part of the model, the ostensive aspect of project thinking shapes the perception of what management justifications are. Project thinking is portrayed in the various ways management justifies their (lack of) actions.

First, there was a noticeable perception among management that once the application went live, the project ended and users utilised the system, the application was implemented and adopted. The distinction between implementation and adoption lacked clarity in terms of semantics, but it was evident that actual usage signified adoption. These quotes were coded as *usage is adoption*:

We use the system and the fact that we have 37,000 cases running through the system is to me a clear indication that it is being utilised. (Retail, Top Management).

Basically, the only way that we looked at it, was how many users out of the total are actually using the system. How many transactions, how much activity is actually being done in the new system compared to the old? We don’t have any other measure. (Manufacturing, Product Owner).

These quotes offer a very rigid perception of adoption and do not go beyond the mere operation of the system, which potentially hindered the realisation of desired benefits. The lack of appropriate measurement techniques for adoption also seemed to pose a challenge for managers to meaningfully influence adoption toward benefit realisation.

And specifically, we are still delivering our products and the supply chain is running. That is a very concrete example of how it is working. (Manufacturing, Top Management)

I don't think the users are thinking – I really get a lot out of using the system. I think they do, what they need to do in the system. So, I think that they can see how it is necessary for their work function and in that way gives them value. But I don't think they feel enriched by it. (Shipping, Top Management)

As the quotes show, management's focus resided on operational and functional use and failed to connect holistically to greater business benefits or user value. By not considering these aspects, companies are at a greater risk of overlooking ambiguous behaviour that is difficult to surface.

Another management justification that emerged in our data was *no usage, no job*. This was used to classify when management themselves expressed how the lack of users' voluntary choice to adopt the given application influenced their adoption actions. These quotes included:

To be honest, if they don't accept the application then they are not able to fulfil their work tasks. And if they don't fulfil their work tasks, then I would hear about it, because then something would be wrong. Then we'd have customers who are not getting help. (Retail, Product Owner).

I think they have adopted the application because there's not really a voluntary choice. At the end of the day, it's the company who decides and you are getting paid to be here, and that means there are some processes you must follow. And this is one of them. (Manufacturing, Top Management).

The quotes follow the logic that employees must adhere to the operational procedures of the company. Following these linear ideas of usage, it was interesting to get the perspective of the users. Here, we asked if they considered their usage as in alignment with the company's desired benefits.

I think there's missing a common understanding of how they want us to use the system, so that we can actually use the system, in the way they want us to use it. (Retail, User).

Yes, we have accepted the application, because we don't have a choice. It is a business decision that we run with this application. And you can have an opinion towards that, in the same way you can have an opinion about all the systems we use. And you can have an opinion about whether it's good or bad. But the fact is, that if you want to work here, then it's this system we all work in. And if you don't like it, then there's not any other way to work. (Shipping, User).

Here, it was evident that the shipping user's experience the *no usage, no job* mindset as problematic, but as a management justification it has permeated to the users and manifested as an understanding that their opinions do not matter as enterprise software is an extensive part of company operations. The retail user continued the

same kind of problematisation as they highlighted that the lack of a common and unified understanding of user behaviour hindered valuable usage. During the interviews, top management and product owners alternated between elaborating on adoption actions and the justification of those actions. In this process, we found that one of the cognitive patterns that was most protruding was what we called *no news is good news*. Here, management rationalised their scarce adoption management actions by asserting that the business is still operational and any major issues that would interfere with that would be brought to their attention. Thus, situated usage as a singular-level and linear idea of production.

We don't necessarily go out and ask a lot of questions. We probably have the more classic approach where if no one is yelling loudly about something that is not good enough, then we probably did okay. That's the approach we've gone with. (Retail, Product Owner).

Actually, the reaction was quite silent. We haven't heard much, to be honest. At least not during the first weeks after the shutdown of the old system. And no information is good information at that point. That's how we looked at it. No one was missing any critical feature. No one was complaining. Everyone could do their work. (Manufacturing, Product Owner).

This level of explicit rationalisation was a surprising finding. The rigidity of this cognitive pattern created an invisible barrier between management and the users. Management did not actively seek systematic feedback and the users were not provided with structured ways of providing feedback, thus creating a continuous circle of unawareness.

When we send out updates to the different owners, there's just radio silence. Perhaps one of the super users react. But there's no community feeling around the application and if we put in effort, we could have a system that worked for us and not against us. (Shipping, Top Management).

I think they are trying. But it is driven by us and that we are reaching out and expressing our frustrations. It is not a proactive approach where they ask whether we've reached the goal. We are the ones picking up the ball and saying – this is not good enough. (Manufacturing, User).

Here, top management perceived the challenge as a lack of community sense toward the application, but the user expressed that the challenge was driven by the lack of proactive actions from management. As we regard the codes for management justification collectively, some clear structural pillars are forming through these cognitive patterns. We argue that the ostensive aspect of management rationalisation, called project thinking, is recognised in the mindset that use and benefits will automatically follow from the completion of a project's activities (e.g., development, training). Thus, we argue

that Figure 1 prescribes to the growing semantic differentiation of project thinking and product thinking. The project-centric mindset is apparent here as once the project has been released it is on to the next project and the companies fail to follow up and understand if the first project met its objectives. Essentially, adoption management follows the argument by Cagan (2017) that discusses how a product-centric mindset aims to meet the needs of the business by acting according to the users' needs. However, this link is not apparent in the cognitive patterns of management.

Management Actions

Redirecting the focus to the performative part of the model, we focus on the specific actions that management expressed to conduct to manage adoption or influence users after the implementation ended and the application went live.

First, in the process of understanding the desired benefits that management hoped to gain from the system, we asked how they validated whether they reached those goals or not. To this, the answer was by looking at the application usage and how many user errors were conducted. This was coded as *measure mistakes*, as in the following quotes:

If we measure that the same store is making the same mistake, then something is wrong. But that's the only data drive we have. If there's mistakes in the process, then we might need to reach out to that person and say – you need to do it the right way. (Retail, Top Management)

Basically, the only way how we looked it at was how many users out of the total is using the new system and how they were using it. How many transactions, how much activity is actually being done compared to the old system? We didn't have any other measurements. (Manufacturing, Product Owner)

Benefits management literature commonly argues that evaluation is a key step in realising desired outcomes and benefits (Ward et al., 1996), and studies show that companies that engage in these types of practices outperform their competitors (Holgeid et al., 2022). However, monitoring user errors, correcting manual mistakes, and ensuring the procedure is followed are rarely considered part of benefit management practices.

We look at the quality of the data that comes in. When I talk to the other departments, then they have a pretty good overview if something totally crazy comes in. (Shipping, Top Management)

So, I've got people who are actually monitoring, how we work with the different parts of the process. So, I have a team for that, they continuously make sure and monitor how the users are working, showing the user

how to do it, and praising them when they're doing a good job. (Shipping, Product Owner)

Though management elaborated on some processes for ensuring procedure and process alignment, we were not able to identify any type of structural or systematic process that followed up on the realisation of desired benefits or any actions that were targeted at managing adoption towards such realisation. To some extent, management was able to describe IT surveys and status meetings that were partly dedicated to the applications' performance and implementation status. But again, the lack of systematic actions was apparent, and the code *no structured approach* emerged as relevant.

We have user satisfaction surveys, which we have been sending out from time to time. But it was not every month, it was more, I don't know, three times a year or something like that. And sometimes it is more focused on a specific feature and sometimes it is the whole system we asked about. (Manufacturing, Product Owner)

In IT we have user surveys a couple of times a year where we ask about everything from how they experience IT service desk. Are they happy with their phone and computers? How did they feel working from home during corona? And what they think about the different applications. And (the case application) always gets heavy critique. (Shipping, Top Management)

These actions do not adhere to the systematic and iterative approach necessary for effects-driven IT improvement (Simonsen & Hertzum, 2022). The code for *no structured approach* also included the following quotes that elaborated on the nature of these management actions that were an attempt to follow up on the implemented application solution.

It's an okay solution and it works on the parameters I want it to work on. But it is not something we are methodologically following up on. We don't have time for it, but it is one of the things one should consider doing. (Retail, Top Management)

We have the traditional approach and have tried with newsletters. And you know, involving the team leaders, that's the way they can give their input. (Manufacturing, Top Management)

This seemingly unstructured approach is also apparent in the final code that we call *it lives, let go*. Here, we categorised the quotes that symbolised how management related to the project's desired benefits and overall adoption after implementation. First, it was expressed how collaborating closely with the users during development ensured that management could take a step back once the application was running.

We include a lot of the users in the design phases. We have 5-6 different heads of departments and different informants – so they are involved. But when we let go, that's when it is implemented, then it's – go do it yourself. (Retail, Product Owner).

I wouldn't say that management has made changes after implementation. There's been some changes from the system supplier. But there's not been any system use or if we are to use it differently. We haven't heard anything afterwards. (Retail, User)

The system is intuitive, and the users have been part of that journey. That whole interaction – I am pretty sure that it has helped. Do I have numbers on it? No, I don't, it's not something we measure, and I don't know if you can see it in our employee satisfaction surveys, that would be amazing if we could. But if we did measure it, I believe that they would actually be more satisfied. (Manufacturing, Top Management)

When we asked management how they knew if the project and application implementation was a success after these extensive efforts and that they stepped back, the code *it lives, let go* was further sustained. One question was directly posed: *how do you know or measure that the application is a success?* To which the answer was: *By pure feedback. And feeling. No data.* (Retail, Top Management). The lack of data-driven success estimation sustains the notion that management takes a step back once the application is running. One success criterion was directly linked to the mere replacement of the old system.

It was a success because the old system was closed. That was the success criterion. That was what we wanted. It wasn't that we needed to save 10 million in costs. It was all about closing the old system. We are there now, and we are beginning to see some drift-offs and are actually seeing some benefits that we hadn't originally used as arguments, but they are surprisingly there. (Manufacturing, Top Management)

In this part of the interview, we continued to circulate the topic of how management aimed to influence usage or user behaviour towards benefits, but the sense of management stepping back continued.

A user might be used to shrugging their shoulders and saying – that's just how the system is. And that's unfortunate because we might be able to fix it. But then we'd have to correct according to 18.000 users: that's a lot of corrections. It is difficult to give an ear to each user. (Retail, Top Management)

We don't train every new employee that comes in. That's up to the individual departments. So, it depends on who sits beside you. We don't have an e-learning universe that tells you where to start and how to follow a procedure or process. (Shipping, User).

The code for *it lives, let go* suggests that the companies are very focused on creating and developing features and completing the project (outputs), instead of focusing on changing user behaviour to drive business results (outcomes) (Seiden, 2019). Hence, we continue to see the relevancy of differentiating between project and product thinking and argue that the findings that support

Figure 1 resemble project thinking. Managers portrayed a project-centric mindset by focusing on project completion and not on the value the product generated. We argue that Figure 1 represents a complex socio-technical ideology that omits project thinking where the cognitive patterns legitimate (lack of) management actions, and in return, the result of those actions does not challenge the perceptions and maintains the mental structures. The codes developed above collectively become instances of project thinking as the cognitive patterns reinforce the idea that enterprise software implementations are successful if the company is operational, and no employees express a lack of features. The project mindset situates operational usage as adoption as users must adhere to these operations to fulfil their job. The project thinking mindset is observed in the action manifestations that rely on measuring errors. The management actions that are further applied are unstructured and there is a sense of management stepping back once the project is implemented.

Discussion

In summary, we analysed three cases of organisations where managers made little effort to manage adoption. In these cases, managers' actions were rigid and unstructured and rationalised by several cognitive patterns. We summarise the process of legitimising actions and maintaining perceptions as project thinking. This is to make connotations to a mindset that prioritises the rigid boundaries of project completion over focus on the product itself and the value it creates. The findings join previous research that navigates the space between benefits management and adoption management. Here, studies also highlight how few benefit-oriented practices are applied post-implementation as the focus remains on delivering a solution that fulfils the specifications, on time and on budget (Ashurst et al., 2008; Pppard et al., 2007). This is despite studies that show that companies that formulate business cases and follow up on them derive greater benefits from their IS projects (Thomas & Fernandez, 2008; Holgeid et al., 2022). The rigidness of the presented statements contradicts other previous studies that have illustrated the importance of acknowledging the relationship between individuals' various emotions, such as happiness and anxiety, and IT use (Beaudry & Pinsonneault, 2010). The portrayed mindset in the interviews seems to disregard the complex facets of user behaviour and application usage, thus challenging studies that proved a reciprocal connection between technology adoption decisions, group interactions, and the degree of positive and negative feelings towards the decision process (Sarker et al., 2005). It also neglects the multilevel nature of usage and the well-established theories that conceptualise effective IT use as

a mechanism through which desired benefits can be reached (Burton-Jones & Volkoff, 2017). The study suggests that successful adoption management entails as much focus on managerial cognition as on prescriptive frameworks and methods. Our informants' project thinking contrasts with current movements such as DevOps, which call for replacing project teams with product teams that are responsible for software over its lifecycle (Wiedemann 2020). Our study adds to this that moving from a project- to a product-centric organisation may involve not only changes in the collaboration between developers and operation units but also changes in the roles that managers see for themselves in software implementations. By summarising the management mindset as project thinking, we offer some explanation for the cases in the study as to why their IT implementation projects struggle to return value. The project thinking mindset seems to be a mental construction of the world in which managers may not find it necessary or worthwhile to follow up on business benefits realisation and user value – or know *how* to. With this model, this paper does not aim to challenge the advantages of project thinking. Instead, the conceptual model points to summarise how project-thinking manifests for some managers and how the mindset can be challenging in the process of managing adoption of enterprise software. According to the Project Management Institute, a project is “a temporary endeavour undertaken to create a unique product, service, or result.” (Ding, 2016) and it is the temporary nature of project thinking that challenges the reiterative evaluation necessity of adoption management.

We hope that managers and other practitioners find inspiration in the findings for how they can become more mindful of their actions and perceptions regarding adoption management. It was surprising how few adoption-focused actions management was applied post-implementation, but by surfacing these rationalisation patterns that drive these (lack of) actions we enable managers to take the first step of transforming from a project-centric mindset to a product-centric mindset thus focusing on holistic adoption management of enterprise software. In the process of becoming more mindful of adoption management perceptions, we also call for managers' and scholars' attention to the consequences of mandatory software. The implementation of enterprise software forces users to comply and utilise the application to adhere to their work tasks. But in the effort to realise benefits and value for both the organisation and users, we need to consider if mere use is desirable if users commit deliberate errors (Ferneley & Sobreperez, 2006) or complain about the system (Markus, 1983). Enterprise software calls for contextualisation of usage as we cannot automatically consider use as representing users' acceptance or satisfaction (Beaudry et al., 2020).

When managing in the space of mandatory software there is a need to consider the adoption of software more holistically to encompass the various acceptance and resistance responses that occur in mandated environments (Bhattacharjee et al., 2018). Even though the managers in the study acknowledged the effects of mandating software, they were not able to influence user adoption accordingly. In mandatory environments, poor software is like a set of blunt building tools. If management provides a builder with unsharpened tools to build your house, it slows down the construction process, compromise the quality of the house, and causes strain on the builder. By not engaging in iterative adoption management practices, organisations miss the opportunity to find out that people are dissatisfied with the software. Thus, management risks providing poor enterprise software for their workers which hinders productivity, stifles the software's full potential, and frustrates the employee. Translating this idea to modern organisations, the claim of being a modern and digital organisation no longer pertains solely to external business operations and satisfying customers. This shift is amplifying attention to the employee experience and the jobs they are to complete, not only within human resource (HR) management (Plaskoff, 2017; Tucker, 2020) but across all functional departments. For IT departments the trajectory ahead seems to entail a focus on not introducing and enforcing solutions that fail to match the user-friendly experiences employees are accustomed to in their personal lives. It is difficult to imagine that young digital natives will lower their expectations regarding how they interact with technology in the future. So, what implications await organisations that fail to monitor and act upon employees' digital experience? Can we realistically envision that digital natives will gravitate towards a work life that entails working with poor software solutions? Studies show that perks are not the sole answer to employee engagement – organisations must co-design experiences with employees that demonstrate care and a deeper understanding of their needs (Plaskoff, 2017). The future of succeeding businesses resides in putting the employee's total experience at the centre. However, it potentially challenges the traditional approach to engage the employee in the workplace, as it goes beyond increasing perks or “funifying” in the workplace (Plaskoff, 2017). In translating such findings from the HR management discipline to an IS implementation setting, we see that mandatory environments call for the same holistic approach to ensure that researchers and practitioners are designing, implementing, and instructing employees to utilise solutions that effectively sustain their needs while aligning with the organisation's desired benefits.

Mandatory environments offer another challenging consequence for management in the transformation process from a project to a product mindset. As the organisation begin to centralise the employees' digital experience, we expect that the challenges of measurability will become evident. In the search for enabling managers to influence IS implementation processes to ensure system usage that solves a given problem with a valuable outcome, guiding indicators must be available and information-driven. In the case study, both levels of management expressed challenges in measuring how the solutions performed and whether they returned value. Following the argument of recent studies that differentiate the management of voluntary and mandatory software (Nah et al., 2004; Brown et al. 2002; McNally, & Griffin, 2010; Bhattacharjee et al., 2018; Beaudry et al., 2020), the conventional ways of measuring IT performance through e.g., frequency of logins, active user count, and duration spent in the application becomes invalid when trying to confirm value realisation. In the context of a voluntary application, like personal use software, where multiple alternatives exist, tracking these metrics indeed makes sense. Then, they help provide an overview of the application's desirability in the reflection of high daily logins, engaged users and extensive usage time. However, when considering an employee who must access an application to fulfil an intricate and time-consuming work task, the meaning of such metrics diminishes and offers scarce insights.

Finally, we acknowledge several limitations of our study. First, the case study offers a restricted scope, as it encompasses only three selected companies, thereby limiting the generalisability of the findings to a broader range of organisations. Thus, we call for future studies to apply this perspective to broader and more longitudinal cases to understand how a project thinking mindset affects adoption management of enterprise software. Second, it would be beneficial to include other data sources to expand the triangulation of actions and per-

ceptions of adoption management. This study only includes 9 interviews, and it would provide further methodological rigour to interview more managers at multiple management levels to understand how different perceptions of adoption impact actual management actions.

Conclusion

In this paper, we explored the actions and rationalisations of managers who made little effort to manage the adoption of enterprise software. While we cannot infer how managers, in general, manage adoption our case study provides insights into why at least some managers do not take an active stance in managing adoption. For these managers, their management actions manifested as unstructured and based on the measurement of user mistakes or manual errors. When these appear, management aims to correct the errors and align the procedure to ensure it does not happen again. The scarcely described actions do not adhere to the adoption management or benefit management definitions of best practice. Furthermore, these actions are legitimised by managers' cognitive perception of adoption and how to manage it in a mandatory environment. Here, usage is perceived as adoption if the business is operational, and the users are enabled to complete and fulfil work tasks. Management acknowledges that users do not have a voluntary choice when it comes to enterprise software, but still maintains the idea that no news is good news. We label these ostensive and performative structures as a project thinking mindset to emphasise that management in the study struggles to influence the adoption process because once the implementation is completed, a new project begins that requires their focus. We hope the presented model inspires future managers to be more mindful of the way they regard adoption and how to manage it, as they potentially risk realising desired benefits.

References

- Ashurst, C., Doherty, N. F., & Peppard, J. (2008). Improving the impact of IT development projects: the benefits realization capability model. *European Journal of Information Systems*, 17(4), 352–370. <https://doi.org/10.1057/ejis.2008.33>
- Beaudry, A., & Pinsonneault, A. (2010). The other side of acceptance: studying the direct and indirect effects of emotions on information technology use. *MIS Quarterly*, 34(4), 689–710. <https://doi.org/10.2307/25750701>
- Beaudry, A., Vaghefi, I., Bagayogo, F., & Lapointe, L. (2020). Impact of IT user behavior: observations through a new lens. *Communications of the Association for Information Systems*, 46, 331–364. <https://doi.org/10.17705/1cais.04615>
- Bhattacharjee, A., Davis, C. J., Connolly, A. J., & Hikmet, N. (2018). User response to mandatory IT use: a coping theory perspective. *European Journal of Information Systems*, 27(4), 395–414. <https://doi.org/10.1057/s41303-017-0047-0>
- Bloch, M., Blumberg, S., & Laartz, J. (2012). Delivering large-scale IT projects on time, on budget, and on value (pp. 1–6). Business Technology Office: McKinsey & Company.
- Brown, S. A., Massey, A. P., Montoya-weiss, M. M., & Burkman, J. R. (2002). Do I Really Have To? User Acceptance of Mandated Technology. *European Journal of Information Systems*, 11(4), 283–295. <https://doi.org/10.1057/palgrave.ejis.3000438>

- Burton-Jones, A., & Volkoff, O. (2017). How can we develop contextualized theories of effective use? A demonstration in the context of community-care electronic health records. *Information Systems Research*, 28(3), 468-489.
- Bygstad, B.; Munkvold, B.E.; and Volkoff, O. (2016). Identifying generative mechanisms through affordances: a framework for critical realist data analysis. *Journal of Information Technology*, 31(1), 83-96.
- Cagan, M. (2017). *Inspired: How to Create Tech Products Customers Love*. John Wiley & Sons.
- Cadle, J., & Yeates, D. (Eds.). (2004). *Project management for information systems*. Pearson education.
- Charmaz, K. (2006). *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. Sage.
- Davis, F. (1987). User acceptance of information systems: the technology acceptance model. *Research Paper Series, Working Paper (529)*, 1-33. University of Michigan. <https://hdl.handle.net/2027.42/35547>
- Ding, R. (2016). *Key project management based on effective project thinking*. Springer-Verlag Berlin Heidelberg.
- Falch, C. C. & Krancher, O., Adoption management: a review of the benefits management and the adoption literature (2023). *Proceedings of the 31st European Conference on Information Systems*. 402. <https://aisel.aisnet.org/ecis2023-rp/402>
- Feldman, M. S. (2000). Organizational routines as a source of continuous change. *Organization Science*, 11(6), 611-629.
- Feldman, M. S., & Pentland, B. T. (2003). Reconceptualizing organizational routines as a source of flexibility and change. *Administrative Science Quarterly*, 48(1), 94-118.
- Ferneley, E. H., & Sobrepey, P. (2006). Resist, comply or workaround? An examination of different facets of user engagement with information systems. *European Journal of Information Systems*, 15(4), 345-356.
- Ginzberg, M. J. (1981). Early diagnosis of MIS implementation failure: promising results and unanswered questions. *Management Science*, 27(4), 459-478.
- Hirschheim, R., & Newman, M. (1988). Information systems and user resistance: theory and practice. *The Computer Journal*, 31(5), 398-408. <https://doi.org/10.1093/comjnl/31.5.398>
- Holgeid, K. K., Jørgensen, M., Sjøberg, D. I., & Krogstie, J. (2022). Benefits management in software development: a systematic review of empirical studies. *IET Software*, 15(1), 1-24. <https://doi.org/10.1049/sfw2.12007>
- Kim, H.-W., & Kankanhalli, A. (2009). Investigating user resistance to information systems implementation: a status quo bias perspective. *MIS Quarterly*, 33(3), 567-582. <https://doi.org/10.2307/20650309>
- Koh, C. E., Prybutok, V. R., & Ryan, S. D. (2010). A model for mandatory use of software technologies: An integrative approach by applying multiple levels of abstraction of informing science. *Informing Science*, 13, 177.
- Latour, B. (1986). *The powers of association*. In J. Law(ed.), 264-280. London: Routledge and Kegan Paul.
- Markus, L. M. (1983). Power, politics, and MIS implementation. *Communications of the ACM*, 26(6), 430-444.
- McNally, R. C., & Griffin, A. (2010). Mandatory adoption of customer relationship management information technology: the role of customer support in an attitude-intention model. *International Journal of Business Innovation and Research*, 4(1-2), 30-47.
- Mintzberg, H. (1971). Managerial work: Analysis from observation. *Management science*, 18(2), B-97.
- Orlikowski, W. J. (1996). Improvising organizational transformation over time: A situated change perspective. *Information systems research*, 7(1), 63-92.
- Peppard, J., Ward, J., & Daniel, E. (2007). Managing the realization of business benefits from IT investments. *MIS Quarterly Executive*, 6(1), 1-11.
- PMI (2016). *Benefits Realization Management Framework*. Thought Leadership Series. November 2016. <https://www.pmi.org/learning/thought-leadership/series/benefits-realization/benefits-realization-management-framework>.
- Polites, G. L., & Karahanna, E. (2012). Shackled to the Status Quo: The Inhibiting Effects of Incumbent System Habit, Switching Costs, and Inertia on New System Acceptance. *MIS Quarterly*, 36(1), 21-42. <https://www.jstor.org/stable/i40068646>.
- Sarker, S., Valacich, J. S., & Sarker, S. (2005). Technology adoption by groups: a valence perspective. *Journal of the Association for Information Systems*, 6(2), 3.
- Seiden, J. (2019). *Outcomes Over Output*. Sense & Respond Press.
- Seo, D., Boonstra, A., & Offenbeek, M. (2011). Managing IS Adoption in Ambivalent Groups. *Communications of the ACM*, 54(11), 68-73. <https://doi.org/10.1145/2018396.2018416>
- Simonsen, J., & Hertzum, M. (2022). Effects-driven IT improvement: pursuing local post-implementation opportunities. *Scandinavian Journal of Information Systems*, 34(1), 2.
- Strong, D. M., Volkoff, O., Johnson, S. A., Pelletier, L. R., Tulu, B., Bar-On, I., ... & Garber, L. (2014). A theory of organization-EHR affordance actualization. *Journal of the association for information systems*, 15(2), 2.
- Thomas, G., & Fernández, W. (2008). Success in IT projects: a matter of definition?. *International journal of project management*, 26(7), 733-742.
- Van Offenbeek, M., Boonstra, A., & Seo, D. (2013). Towards integrating acceptance and resistance research: evidence from a telecare case study. *European Journal of Information Systems*, 22(4), 434-454.
- Venkatesh, V., Thong, J. Y., & Xu, X. (2016). Unified theory of acceptance and use of technology: A synthesis and the road ahead. *Journal of the association for Information Systems*, 17(5), 328-376.
- Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems* (4:2), pp. 74-82
- Ward, J., Taylor, P., & Bond, P. (1996). Evaluation and realisation of IS/IT benefits: an empirical study of current practice. *European Journal of Information Systems*, 4(4), 214-225.
- Wiedemann, A., Wiesche, M., Gewalt, H., & Krcmar, H. (2020). Understanding how DevOps aligns development and operations: a tripartite model of intra-IT alignment. *European Journal of Information Systems*, 29(5), 458-473.