Sustaining IT Usefulness – Re-defining end users’ role as contextual designers

Nathan Lakew

Mid Sweden University, Sundsvall, SE
nathan.lakew@miun.se

Abstract. A framework for understanding and interpreting IT usefulness and fitness attributes is presented. This framework is grounded on a relationship that exists between organisms and their landscape. The concept draws on the notion that sustainable relationship between two systems (such as IT and end-users) can be achieved through structural coupling results from mutual perpetuations. In this setting, while contextual usefulness is established in the end-users’ environment, IT designers perpetuate fitness into the conceptual environment. Their relationship suggests that usefulness feeds essential input that enables to create a sustainable fitness attribute. Based on the empirical evidence, the paper demonstrates that end-users are better equipped with defining contextual *usefulness* of IT systems while IT designers’ role to create fitness attribute enables a long-term use of IT artifacts.

Keywords: Usefulness, Fitness, structural coupling, conceptual and contextual space

1 Introduction

Recently, IS research “has found its legitimacy” [1] in the design science research where the main focus has been designing artifacts in the form of methods, instantiations and IT systems. However, literature that are specifically concerns with developing guidelines to design IT artifacts [2-4] recognized that the process of developing an IT artifact will not be finalized once it is appropriated by end-users. In fact, the success of IT artifacts are subjected to rigorous “quality evaluation and efficacy” [2] in post-implementation.

As Gill and Hevner [5] point out, IS research has been widely used IT artifact’s usefulness attribute as the main evaluation criteria for IT success[6]. In this setting, IT is mainly evaluated based on functions and properties it possesses to match end-users’ requirements. But recent research [7, 8] has shown that end-users’ environment is always in the course of change where organizations are portrayed as ‘self-design’ [8] systems. To sustain IT systems’ usefulness, IS design practice needs what Gill and Hevner[5] referred to as a fitness attribute that involves adaption and evolution of IT artifacts.

The purpose of this paper is to illustrate the application of fitness attribute to sustain the usefulness of IT artifacts. The paper develops an analytical framework to demonstrate the relationship between these two attributes and their figurative domain space. Based on the analytical framework, the paper questions end-users' role in the process of IS design which is traditionally limited to requirements knowledge base. It draws on a case study of newly implemented Learning Management System (LMS) called Moodle at Mid Sweden University in three different campuses. The study includes an in-depth interview with the so-called Moodle champions, who were responsible for the smooth transition of LMS in their perspective departments, IT developers' team, Moodle deployment project administrators, and Learning Resource Center (LRC).

Organization of the paper: First a summary of IS literature that concerns with long-term use of IT systems is presented as a theoretical background. Second, conceptual founding of fitness and usefulness attributes in IS design will be presented. Third, based on the conceptual foundation, an analytical framework is developed to illustrate the relationship between these two attributes. Finally, after presenting a case study, I further discuss these attributes and their contribution toward long-term use of IT artifacts.

2 Theoretical foundation

With regard to IS practice that aims for a long-term use of IT systems, a number of researchers proposed different methods, conceptual frameworks, and IS design theories. However, much of the work focused on IS practices that occurs after the deployment of IT systems. Such IT practices mainly include, but are not limited to, rigorous evaluating mechanisms of IT artifacts and robust IT training programs.

Mendoza et al.[9], for example, report that the role of usefulness in the process of adopting and adapting technology fades once end-users become what the authors referred to as long-term users. After the initial adoption, long-term users' focus shifts from technology features that satisfy functionality needs to features that enable users to manipulate the technology to suit their ever-changing work practices. After initial adoption, new aspects of usefulness emerges that concern with the ability of a system to facilitate exploring and constructing new work practice. The authors argue that well-established and on-going IT trainings play a crucial role to sustain a long-term usefulness of IT artifacts.

Tyre and Orlikowski [10] developed a model to describe different stages that a technology will go through during IT implementation. With three different cases they have presented, the authors consistently observed that end-users' effort to adapt a technology into their daily activities gradually diminishes as routine gets its way. Initial adaption to technology mainly aims to routinize technology with daily activities than aiming for using artifacts for long-term use. But subsequent adaptations triggered by different events such as new work practices can lead to finding ways to sus-

tain usefulness of an IT artifact. Tyre and Orlikowski's model highlights enabling and constraining factors that influence adaption, most of which are associated with the process of new IT system adoption.

Carroll et al.[11] introduced Model of technology Appropriation (MTA) to describe the appropriation process they claim to include long-term use of IT artifacts. MTA describes technology appropriation at three levels: when it is adopted and evaluated based on its usefulness, when it becomes routinized and adapted to existing work practices, and when it is considered stabilized and enforced as a result of technology entanglements with existing practices. At times, the authors suggested, end-users' activities can change and these changes affect technology's way of being from 'technology-as-designed' to 'technology-in-use'. They concluded that long-term use IT artifacts' factors are 'unlikely to be uncover' with the usually short and targeted tests organized in the beginning of new IT system's adoption.

The above exemplars of technology appropriation and long-term use of IT systems research show that there is a general consensus regarding the relationship between the initial usefulness of IT artifacts and the success of long-term use. End-users' practice of adapting technology to the existing work practice is mainly based on the usefulness attribute of IT artifacts. In addition, users' active participation seemed to slow down as technology properly placed in existing practice. However, it is still unclear how, in the process of IS design, IS designers can inset attributes that sustain the usefulness of IT artifacts to create long-term use technology.

In exploring this research question, the paper grounds on Gill and Hevner [5] work, who show that IS design can have two focus, a fitness or a usefulness focus. The authors [5] present a fitness-utility model to describe the evolving nature of IT artifacts. Usefulness is presented as having a close tie with other prominent IS researches which consider usefulness as being both a motivation factor and evaluation criteria to select and adopt IT systems. Fitness-utility model is presented as complementary to the well-established usefulness model in describing the adaptive attributes of artifacts.

3 Conceptual foundation of fitness and usefulness attributes

Gill and Hevner [5] suggest that usefulness in its broadest term embodies IT systems' characteristics such as "efficacy in performing the task (including performance), range of task cases performed, ease of use, ease of learning, and cost-benefit in the performance of a task". Here usefulness represents all IT artifacts' attributes that are related to task performance.

Central to usefulness is the task of defining design candidates in the design space. Requirements are searched for a viable design candidate. The well-established Cartesian world view [12] informs the search for design candidates. In this view, end-users' needs (specific design candidates) are thought to be '*representational*' and can be

segmented to the level of operational ends. Once the design candidate's goal is represented in a specific form, the next step is to convert these goals to 'constraints and functions' using IT artifacts.

Gill and Hevner [5] suggested two circumstances where the word fitness can be applied. The first definition is related to its use in the evolution field. Here, fitness of an organism represents its ability to continuously reproduce and evolve for successive generations. The second definition of fitness implies to a shorter and specific fitness scenarios. For example, an organism's ability to overcome a specific situation or danger in a given period of time makes it fit to that specific situation. Individual physical fitness for a specific type of sport can be a good example to the second definition of fitness. In this paper the word fitness refers to the first definition that concern with the evolving ability of IT artifacts for the purpose of sustaining usefulness. Hence, design that focuses on fitness aims to develop the competency of IT artifacts' evolving and adapting ability in users' environment.

A closer look on how organism creates a long-term fitness can be used to illustrate the relationship between fitness and usefulness. Biological organisms do not create *representations* of their environment in order to adapt with it. In fact, their environment is too complex to be represented. Instead, their interaction can be described as patterns of practices that continuously change their own *structure* to adapt their landscape. Maturana and Guilloff [14] describes this interaction using the concept of structural coupling:

When two or more plastic dynamic systems interacts recursively under conditions in which their identities are maintained, the process of structural coupling takes place as a process of reciprocal selection of congruent paths of structural changes in the interacting systems ... [14], p. 139

The definition draws on the idea that two plastic systems can mutually affect each other continuously or at least one repeatedly perturbs the other. A plastic system represents one that can be affected by external events (e.g. organism or environment). The mutual perpetuation of systems (e.g. organisms, software, environment) "leads to a structural fit between the systems" [15].

Each structure is created as a result of perturbed activities, which in turn continuously change perturbation domain (e.g. design space). Fitting to unyielding environment is not a matter of information process or representation of contextual details. Organisms change their *structure* to respond to perturbed changes. The transformation in structure is important to generate an "appropriate change of state triggered by specific perturbing changes" [16]. The process of change in structure, in turn, is affected by the perturbing changes. That is, specific details and changes gradually create "history of responses" [16] leading up to structural formation.

On the one hand, usefulness is specifically concerned with contextual responses where *operational* level fitness is materialized. End-users are the main actors in this activity as they are closer to the application domain. On the other hand, fitness facilitates the contextual respond by providing a platform or structure at a conceptual level. Design that targets a long-term use of IT artifact, thus, focuses on the structure of the artifact than the contextual usefulness of it.

Fitness sees each usefulness demarcations of design candidates as instantiation of users' needs. What usefulness focused IS design assumes to be a design candidate is just another 'temporal regularity' [7] in the eyes of fitness focused IS design. In this sense, fitness can be called instantiations of usefulness. Even though, fitness provides a means to guarantee usefulness, the reverse does not hold true. Structural coupling is the bases for both selection and evolution [16]. While selection of an individual organism depends on surviving each given unyielding circumstance at a time, evolution requires a continuous coupling with other autopoiesis. Structural coupling guarantees both temporal (usefulness) and co-evolutionary situations. Even though, fixing temporal regularities considered essential in the fitness landscape, it is not a sufficient condition for sustaining usefulness.

Hence fitness aims to design sustainable affordances that are malleable to the general design space. Temporary regularities are seen as just another opportunity for a more structural coupling. Usefulness continuously feeds essential input to fitness. There is a feed-back loop like mechanism (described as activities in the analytical framework below) where the structure affects the response and vice versa. In addition, an explanation as to where these two activities are realized can be characterized in two abstract domains: conceptual domain (structure level) and contextual domain (response level). The next section further elaborates the relationship between fitness and usefulness using analytical framework.

4 Analytical framework

The purpose this framework (See figure 4.1) is to demonstrate the application of fitness and usefulness attributes in designing IT artifacts. The framework can be used as analytical tool to identify, organize, and interpret essential activities that involves IS design practices and end users' interaction with IT systems.

Defining space

One of the key assumptions in the framework is that there existing two spaces that the IS practice operates: contextual and conceptual. In the contextual space, end users and implementers strive to embed technological properties and functions to their context. Here, the main goal is to achieve contextual affordances using IT artifact functions and properties. Contextual space represents end-users environment that is emergent and complex in nature. Conceptual space oversees 'perturbed activities' and provide

structure that facilitates fitness based on the feedback from contextual space. In other word, conceptual space provides a means for successful structural coupling between users and their context. In doing so, it does not concern with specific users context or particular affordances. While IS design that focuses on usefulness tries to find representable design candidate within contextual space, fitness focused design leave the allocation of affordances for end-users. This way, end-users are able to define both problems and solutions based on their context.

Figure 4.1 Usefulness and fitness relationship framework

Activities – embedding and de-embedding

There are two activities in the analytical framework represented by arrows in fig. 4. While the embedding activity arrow shows the creation of contextual affordances to

specific needs, de-embedding activity represents the process of ‘abstracting’ general rules [17] to create coupling ability for the artifact.

Embedding represents the process of “place-making” [18] new technological artifacts to existing work practice. End users and implementers are considered to be actors of this process. End-users assemble and scramble affordances accordingly with their context. This is also in line with the framework of Maturana’s structural coupling, where organisms use all available means to survive and continue to fit in a landscape. End-users design contextual ‘placement’ of IT artifacts and become ‘secondary designers’ on their own terms.

As a result of embedding ‘design’, new work practices can emergent. These work practices, including workarounds, are matured toward contextual usefulness. New work practices can be taken as new IS opportunities/problems that need a fitness ‘touch’. That is, IS developers may decide to include coupling mechanism for these new practices in their next ‘version’. Thus, de-embedding activities triggered to create technological properties and functions to sustain new contextual affordances.

Crossing the imaginary threshold between conceptual and contextual space

There is no defined starting point for any design [17]. Different actors play important roles in defining the threshold of these imaginary spaces and activities. In this analytical framework, implementers or sometimes called super-users can share both the embedding and de-embedding activities with developers and end-users. In a sense, they are the mediators for different ‘spaces’. The setting is depend on the organizational strategy, IS design architecture, and other IS method of development issues. Reflecting on the case study that is presented in this manuscript, super-users were used to implement new technology, where developers were continuously adding new functions and properties.

5 Case study

The paper draws on a case study of newly implemented Learning Management System (LMS) called Moodle at Mid Sweden University in three different campuses. The study includes an in-depth interview with the so-called Moodle champions, who were responsible for the smooth transition of LMS in their perspective departments, IT developers’ team, Moodle deployment project administrators, and Learning Resource Center (LRC). Currently Moodle serves more than 17,000 students and the university’s 1500 staff under the supervision of Learning Management Center (LRC).

Research Method

The university has nominated 16 Moodle champions, tasked with supporting the implementation process of Moodle. With the permission of LRC, nine Moodle champi-

ons were agreed to participate in an in-depth interview, each lasts between 30-60 minutes. In addition, 5 more LRC staffs (2 Moodle developers, 1 LRC staff and 2 LRC administrators) have participated in the interview. In total, 14 interviewees were participated in the span of 6 month data collection. It should be noted that a follow-up interview has been conducted to three LRC staffs.

Data were collected using tape recordings and notes. All interviews were transcript to Atlas.ti. Descriptive codes were applied to categorized patterns and specific themes. In coding the data, ‘content analysis’ [19] method is used to examine transcript interviews at the conversation level. The main objective of the analysis was to explore IS practices that affect long-term use of IT artifacts. Two different level of IS practices are emerged from the analysis. A third activity, feed-back loop, is presented as a theme that communicates these IS practices (see table 5.1). In particular, the following steps were followed:

- 1) All the interviews are transcribed and uploaded to Atlas.ti software
- 2) After re-reading all the interviews, 20% of coding have been done manual. In doing so, patterns were observed at conversation level.
- 3) After coding each response, the author examined IS practices and corresponding actors in each of the coded conversations. Three main activities (families as Atlas.ti calls it) have emerged (See table 3). Each family was then compared with the interviewees response based on the original code.
- 4) Based on these families, the paper develops logical relationship with the analytical framework in discussing the findings.

Categories	Descriptions	Examples	Actors
Conceptual Level activities	IT designers inset fitness attribute on to sustain usefulness	<i>“If we feel like it is important (functionality)and a decision is made here in LRC”</i>	LRC staffs and Moodle champions
Contextual Level activities	End-users continuously define both design candidates and contextual affordances using IT functions and properties.	<i>“I have workaround as well, which I created before long time ago, from WebCT limitation, but I continue using it, in fact I adapted my old solution (to) the new system”</i>	End-users’ and Moodle champions
Feed-back mechanism	Both IT designers and end-users exchange input for usefulness and fitness.	<i>“We sent our complaints and raise the issue in the champion meetings”</i>	Moodle champions

Table. 5.1. Three Main categories emerged from coding

6 Findings

The analysis of the case study allows illustrating the relationship between usefulness and fitness attributes and the application of the analytical framework. While end-

users' practices are illustrated in the contextual domain, IT developers' and LRC staffs' activities are represented in the conceptual level of IS practice.

Embedding practices: Creating contextual usefulness

Before Moodle was rolled-out, champions from each department were participated in training workshops to learn how to modify and create affordances accordingly with their respective departments' need. Such affordances mainly include department based forms, subject based data uploading mechanisms (e.g. math numeric assignments), section forms that introduce courses, subject based student forms. In the beginning of Moodle implementation, champions were asked to provide IT affordances that suits the existing practice (*"I used to get question like 'would it be good if I can do this' then I would say, but yes you can do that" Tagged 04 (...)* *"The question is more about the equivalent options they can get in Moodle."* Tagged06). End-users' past entanglements with technological practices are found to be the main driving force of defining new contextual usefulness.

In addition, contextual affordances are found to be emergent and locally confined. Different course given in different seasons, teachers' transfer, organizational changes in departments, introduction of new courses are few list of reasons that are continuously emerge as a new contextual requirements. Moodle champions were constantly changing local artifacts' usefulness and teach end-users to create those contextual affordances. (*"There is always discussion about how to design for different types of courses regarding to the subject and/or if it is a campus or a distance form. Challenges come(s) every week" Tagged 12 (...)* *"They often tell me "but I'll keep that example from the template just in case I need it later", and get surprised when I show them how easy it is to delete it and create it all over again. I think people are used to the fact that the WebCT structure was not as malleable, and since they are not aware of how easy it is to modify things in Moodle" Tagged 04.*

The data also reveal that end-users are proven to be competent designers of their own contextual environment. Contextual affordances may sometimes involve partial use of IT artifacts' function, while workarounds affixed to fill the rest. On this regard, users competently assembled fragments of affordances both from technological functions and perceptual artifacts (workarounds). *"The great limitation is you can't attach files to emails. In my department, the architects want to upload files but then I have to send those from outlook, so it is hard. We find some way around it, we instead start to use a forum to upload the files, but I don't think that is efficient way to do it" Tagged 05).*

De-embedding practices: Creating conceptual fitness

In the quest of creating contextual affordances, users continuously entangles context with IT artifacts' functionalities. As a result, new requirements were continuously emerged. Moodle champions received different functionality requirements and if

necessary let the LRC staffs aware of these new requirements. The case study shows that such requirements were handled to enhance Moodle functions and properties.

First, users' requirement is de-embed from contextual use to develop context neutral IT artifact functions. LRC administrators make the final decision whether to append (in most cases, plug-ins) such functionalities or not. (*"If we feel like it is important and a decision is made here in LRC" Tagged 03*). Secondly, LRC administrators also decided whether such functionalities should be available for all departments. In a way, contextual environment feeds the IT artifacts' structure to continuously evolve with emergent needs.

Feed-back loop – communication between conceptual and contextual space

Moodle champions were playing the role of a 'middleman' between LRC staffs and end-users to create a feed-back loop between usefulness and fitness (*"We sent our complaints and raise the issue in the champion meetings" tagged 09*). Even though, such settings are existed, the case study also shown that there were not used effectively (*"Process from the universities is not usually specific. For example, they goes like Facebook works better, but they don't say what is they need that is similar to Facebook that needs to be there. The request is not correct and the way it comes is not standard as well. Sometimes it comes to me and other times it goes to the LRC" tagged 02*).

In summary, it is evident from the case study that contextual (usefulness) and structural (fitness) changes are implemented in different level of domain space; at conceptual and contextual space. In addition, the analysis has shown that end-users' role as contextual designers can provide competent feedback for long-term use of IT artifacts.

7 Discussion and Conclusion

In this paper, I set out to illustrate the application of fitness attribute to sustain the usefulness of IT artifacts. The paper develops an analytical framework to demonstrate the relationship between these two attributes and their figurative domain space. I argue that while both attributes play an essential role in developing sustainable IT artifacts, fitness based IS design slightly questions traditionally assigned roles given to IT designers and end-users.

On the one hand, conceptual space is used to demonstrate IT designers' role as contextual affordance facilitators. In this capacity, designers are expected to continuously instill the fitness attributes based on the feedback from end-users' IT practices. On the other hand, contextual space is defined as end-users' environment where particulars or daily context based affordances are continuously created and scrapped. Permanent affordances, then, can be used as a candidate for fitness attribute. It is argued that

users are competent to analysis and define their own contextual affordances in a coherent manner. Implementers/super users play an important role in the process of embedding IT systems in the contextual environment. The case study presented in this paper demonstrates these relationships. In addition, it is shown that end-users can deliver constrictive feedback that can be used as a base to continuously produce fitness attribute for existing IT system.

Naturally, usefulness focused IS design bases on a finely developed requirement specifications and particulars results from a well-defined methodological engineering steps [20]. The rationale behind such effort is that well-defined requirement elicitation results in finely tuned IT artifacts with users' environment. IS literature have documented that solutions developed based on design methods that promote 'specific fitness of user-requirement' failed to adapt once they are appropriated by users [21-23]. This is because, as noted previously, end-users' environment is full of articulations and negotiations that may or may not involve IT artifacts. It is implausible to capture all such socio-technical entanglements in a specific timeframe of user requirement analysis. The more a designer focuses on specifics and particulars, the more it misses the holistic user environment that includes socio-technical 'imbrications' [24] such as workarounds and locally implemented work practices. Hence, the rationale behind the emphasis of usefulness may actually result in working against the very design reasoning behind the usefulness of an IT artifact. That is, an over-emphasis on the usefulness can result in a less useful IT artifact.

The paper has also emphasized end-users' input for IT artifacts' fitness attribute. The process by which end-users implement, and at times modify, IT artifacts has been referred to as 'secondary design' [25]. In this setting, primary design demarcates the design process to develop IT systems' properties and functions. An IS design that emphasizes the importance of developing usefulness attribute at the outset leaves less room for secondary design, thus limits the evolving attribute or fitness of IT systems. Design that focus on the fitness attribute transfers the responsibility of defining problem space and creating contextual affordances to the hands of end-users. The assumption is that design candidate in users' environment kept changing contextually and cannot be defined at the outset. Since end-users' environment is considered to be socio-technical, the practice of design, ontologically, falls in the holistic tradition.

In addition, the strong association between usefulness and IT success criteria indirectly affects the search process of a design candidate during users' requirement elicitation. In defining user requirements, usefulness obliged the searching process to be focused on design candidates that can be formulated as a "structure of goals" [26]. Hence, system analysts see users' environment as field of problems that can be converted to operational ends. As it is stated, however, ends are not always clear, but "must be constructed from messy problems"[27] or sometimes called "wicked problems" [28]. The application of usefulness logic to define wicked IS problems as operational ends can result in a short-term use of IT systems.

As contemporary organizations are increasingly becoming complex and their needs are being portrayed as instable[29], applying usefulness at the outset to develop, adopt and evaluate IT artifacts can be limiting. It is factual that the relevance of any product, first and foremost, depends on its contextual usefulness. But predicting contextual usefulness, even as situational practices or as design-in-use, at the outset can affect long-term use of IT artifacts. The findings in this paper advocate end-users' competency to define their own contextual affordances, thus shifting such IT practices into the hands of end-users.

Reference

1. Hovorka, D. and M. Germonprez. *Tinkering, tailoring and bricolage: Implications for theories of design*. in *Proceedings of the 15th Americas Conference on Information Systems*. 2009.
2. Hevner, A.R., et al., *Design science in information systems research*. MIS Quarterly, 2004. **28**(1): p. 75-105.
3. March, S.T. and V.C. Storey, *Design science in the information systems discipline: an introduction to the special issue on design science research*. MIS Quarterly, 2008. **32**(4): p. 725-730.
4. Peffers, K., et al., *A design science research methodology for information systems research*. Journal of Management Information Systems, 2007. **24**(3): p. 45-77.
5. Gill, T.G. and A.R. Hevner, *A fitness-utility model for design science research*, in *Service-Oriented Perspectives in Design Science Research*. 2011, Springer. p. 237-252.
6. Delone, W.H., *The DeLone and McLean model of information systems success: a ten-year update*. Journal of Management Information Systems, 2003. **19**(4): p. 9-30.
7. Truex, D.P., R. Baskerville, and H. Klein, *Growing systems in emergent organizations*. Commun. ACM, 1999. **42**(8): p. 117-123.
8. Weick, K.E., K.M. Sutcliffe, and D. Obstfeld, *Organizing and the process of sensemaking*. Organization Science, 2005. **16**(4): p. 409-421.
9. Mendoza, A., J. Carroll, and L. Stern. *Adoption, adaption, stabilization and stagnation: software appropriation over time*. in *ACIS 2005*. 2005: University of Technology, Sydney.
10. Tyre, M.J. and W.J. Orlikowski, *Windows of opportunity: Temporal patterns of technological adaptation in organizations*. Organization Science, 1994. **5**(1): p. 98-118.
11. Carroll, J., et al., *A field study of perceptions and use of mobile telephones by 16 to 22 year olds*. Journal of Information Technology Theory and Application (JITTA), 2002. **4**(2): p. 6.
12. Descartes, R., *The philosophical writings of Descartes*. Vol. 2. 1985: Cambridge University Press.

13. Stefanou, C.J., *A framework for the ex-ante evaluation of ERP software*. European Journal of Information Systems, 2001. **10**(4): p. 204-215.
14. Maturana, H.R. and G.D. Guilloff, *The quest for the intelligence of intelligence*. Journal of Social and Biological Structures, 1980. **3**(2): p. 135-148.
15. Quick, T., et al. *The essence of embodiment: A framework for understanding and exploiting structural coupling between system and environment*. in *AIP Conference Proceedings*. 2000.
16. Winograd, T.A. and C.F. Flores, *Understanding computers and cognition: A new foundation for design*. 1986: Ablex Pub.
17. Lee, J.S., J. Pries-Heje, and R. Baskerville, *Theorizing in design science research*, in *Service-Oriented Perspectives in Design Science Research*. 2011, Springer. p. 1-16.
18. Riemer, K. and R.B. Johnston, *Place-making: A Phenomenological Theory of Technology Appropriation*. 2013.
19. Klein, H.K. and D.P. Truex, *Discourse analysis: a semiotic approach to the investigation of organizational emergence*. The Semiotics of the Workplace, 1995.
20. Iivari, J., R. Hirschheim, and H.K. Klein, *A paradigmatic analysis contrasting information systems development approaches and methodologies*. Information Systems Research, 1998. **9**(2): p. 164-193.
21. DeSanctis, G. and M.S. Poole, *Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory*. Organization Science, 1994. **5**(2): p. 121-147.
22. Orlikowski, W.J., *Using technology and constituting structures: A practice lens for studying technology in organizations*. Resources, co-evolution and artifacts, 2008: p. 255-305.
23. Orlikowski, W.J., *Sociomaterial practices: Exploring technology at work*. Organization Studies, 2007. **28**(9): p. 1435-1448.
24. Leonardi, P., *When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies*. MIS Quarterly, 2011. **35**(1): p. 147-167.
25. Germonprez et al, *"Secondary design: A case of behavioral design science research"*. Journal of the Association for Information Systems, 2011.
26. Winograd, T. and F. Flores, *Understanding computers and cognition: A new foundation for design*. 1986: Ablex Pub.
27. Schön, D.A., *The reflective practitioner*. Vol. 1. 1999: Basic books.
28. Rittel, H.W., *The reasoning of designers*. 1987: IGP.
29. Truex, D., R. Baskerville, and J. Travis, *Amethodical systems development: the deferred meaning of systems development methods*. Accounting, Management and Information Technologies, 2000. **10**(1): p. 53-79.