Principles for enabling deep secondary design

Pries-Heje, Jan; Hansen, Magnus Rotvit Perlt

Published in:
Nordic Contributions in IS Research

DOI:
10.1007/978-3-319-64695-4_6

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Pries-Heje, J., & Hansen, M. R. P. (2017). Principles for enabling deep secondary design. In S. Stigberg, J. Karlsen, H. Holone, & C. Linnes (Eds.), *Nordic Contributions in IS Research: 8th Scandinavian Conference on Information Systems* (pp. 67-82). Springer. https://doi.org/10.1007/978-3-319-64695-4_6

Principles for Enabling Deep Secondary Design

Jan Pries-Heje(✉) and Magnus Rotvit Perlt Hansen

Institute of People and Technology, Informatics,
Roskilde University, Roskilde, Denmark
{janph, magnuha}@ruc.dk

Abstract. User-based redesign after implementation has been studied in many contexts gone by many different names, such as appropriation of technology, malleable design and secondary design. The phenomenon of redesigning content has mainly revolved around technologies such as Facebook, Twitter, or Wikipedia or portal-based technology with configuration abilities, with very little focus on technologies where users can change both functionality, content and the level of technology complexity. We coin this type of secondary design *deep secondary design*. In this paper, we investigate how to enable deep secondary design by analyzing two cases where secondary designers fundamentally change functionality, content and technology complexity level. The first case redesigns a decision model for agile development in an insurance company; the second creates a contingency model for choosing project management tools and techniques in a hospital. Our analysis of the two cases leads to the identification of four principles of design implementation that primary designers can apply to enable secondary design and four corresponding design implementation principles that secondary designers themselves need to apply.

Keywords: Principles · Design theory · Secondary design · Case study

1 Introduction

Years ago, von Hippel [1] claimed that a type of users called *Lead Users* were using IT products in innovative ways, and next generations should be based on the practices of these users. As such, literature on user-changed reinvention has existed since the late 80s in the shape of e.g. participatory design [2] and emergent organizational workarounds [3]. Germonprez et al. [4] observed that IT systems can intentionally be constructed so that it is easy for the design "to be tinkered with and tailored for the creation of systems where people actively reflect on and engage with their local contexts, tasks, and technologies" [4, p. 665]. Today, many IT products are designed in a way that users can easily take the design into use and add their own content and use cases. The users redesign in a manner so they become designers of the original primary design; they become *secondary designers* of the product. This has been found to carry with it new functions for social interaction in non-organizational settings with the argument that secondary design of function in organizational settings can actually inhibit the tailorability and possibilities for users as secondary designers [4]. Furthermore, common design theory and principles, e.g. originating from Design Science Research, have primarily sought to provide prescriptions of how functionality of a

S. Stigberg et al. (Eds.): SCIS 2017, LNBIP 294, pp. 67–82, 2017.
DOI: 10.1007/978-3-319-64695-4_6

design should be in order to attain a solution [5]. However, as secondary design has been defined as a process and not as an artifact only, we still seek to find a proper answer to how and when users take on the role of secondary designers who redesign the functionality, underlying logic and the level of technology complexity for a new target group or use situation. Prior research has focused on the users as *contributors*, while we still need knowledge of how to enable users to become *differentiators* of the content and function of the design [4]. We call this *deep secondary design* and distinguish it from tailoring content by preferences or new features of a product [6], and we seek to identify what enables these changes and how to enable it. Thus, the research question being answered in this paper is: *"How can secondary design be enabled?"*

We answer the research question by presenting two case studies of completed secondary (re-)designs, all concerned with relatively simple IT artifacts on different complexity levels (from manual paper-based scorings to spreadsheet macros and visualization). First, we review the existing literature on related research on secondary design and identify areas for potential research. Second, we explain our research method and methodology used on the two cases. Third, we analyze the cases to infer principles for enabling secondary design. Fourth, we discuss findings and contributions to the field, and finally we conclude the paper.

2 Previous Research

In this section, we show a gap in the field of research how to enable the phenomenon of secondary design. We argue that rather than providing principles of physical attributes of the design, we should focus more on the principles of process that can help primary designers successfully enable secondary design.

2.1 Intentionality and Problem-Solving

Design Science Research (DSR) aims to solve problems through designing solutions and contributing with prescriptive design theories to solve classes of problems [5, 7, 8]. Contributions to DSR require general and prescriptive theories to share and assist designers in solving specific design problems. As such, the DSR field is a strong contender for helping theorize of principles for secondary design. Design principles are a central part of design theories [9] and also a central part of "nascent design theory" (theories in progress) [10]. Gregor and Jones [11] specifically note that principles can be denoted as "form and function", explaining the physical attributes and constructs of the artifact, and "principles of implementation"; how to create the artifact. Recently, examples have been given of malleability to a design, divided into levels of customization (of certain preferences of the technology), integration (between other technologies), and extension (adding new capabilities of the artifact) [6]. The literature on DSR converges in agreement that design theories should focus on design and principles as inherent attributes of the artifact, or something the designers do to create the artifact with little regard to the outside context. Of special interest is the element of "principles of implementation", which indicates prescriptions for how to create the

design artifact. We use this element to argue that in order to answer our research question, we need principles of design implementation for secondary design [11].

2.2 Tailorable Technology Design

In tailorable technology design, the designer intends the users to modify the design after use [12]. The process of tailorable technology occurs in two processes, a *default state* and an *ongoing act of tailoring* where the users gradually change the design as they see fit. Design principles for tailorable design include designers to provide flexibility of abstract tasks and using modifiable components to be reused and re-arranged without establishing best practices [12]. The concept of secondary design by Germonprez et al. [4] is conceptualized by defining a *functional* and a *content* layer. Secondary design of the functional layer is defined as users combine hardware and software solutions to solve problems. The content layer is defined as users being freely able to change the content to support various types of expressions, use ad interaction with other users. Common for the notion of tailorable technology design is that principles rely on the attributes, structures and technology choices of the artifact, similar to the notion of what DSR has noted. We see a gap between understanding how to distinguish artifact attributes and the processes that enable users to move beyond contribution of content to the artifact and into becoming a differentiator of function and content.

2.3 Relationship Between Designers and Users

In order to combine principles of the artifact and principles of design implementation of the process leading up to secondary design, it is imperative to position us in relation to other research areas revolving around design with stakeholders. Examples of research areas where designers engage with stakeholders to create new designs are for example the concept of co-realization [13], or the wide-spread participatory design [2, 14]. Co-realization is the notion of designing technologies-in-use over longitudinal periods based on an ethno-methodological approach so that the final design better corresponds to the reality it eventually is placed in. The principles of participatory design include that designers should have hands-on experience with the domain and that those who will use the design should also have decision power over what the final design should be. Common for these approaches is that users are seen as co-designers who learn design and that the designers are more facilitators than experts. Secondary design has been positioned as being different from this, as secondary designers are defined as autonomous once they begin their design journey [4] and only little or no interaction with the primary designers. We argue that this makes it crucial to focus on the very fleeting moments where a relationship can exist between primary designers and users in secondary design.

2.4 Organizational Appropriation of Technology

On an organizational level, unintended use of technology can be deemed as "appropriations of technology". These typically involve changes to processes post-implementation that are difficult to plan for or entirely unintended. Seminal research is that of Orlikowski and Hofman [3] who distinguished between emergent changes (new and difficult-to-observe changes to how technology is being used) and opportunity-based changes (changes captured and formally implemented by the organization). Davern and Wilkin [15] also created a matrix of types of changes that were either circumventions or innovations based on new, emergent practices. Richter and Riemer [16] defined malleable end-user software (MEUS) as software in the organizational space that changes how users act. With the previously defined layers of content and function, one can say that MEUS is designed specifically with the intention of content layer manipulation. Furthermore, this practice has been coined as *"as a social process of appropriation in which the software is interpreted and "placed" within the context of existing work practices"* [16, p. 196].

Common for the literature of appropriation of technology is that it focuses on what users do with and around the software, and how the organization can benefit from this. As such, it somewhat excludes the relationship between designers and users, as well as the opportunities for changes to the technology itself.

3 Method

We studied the above concept of secondary design in a multiple case study aiming at establishing a "replication logic" for contrasting results [17]. The two cases represent a diverse and different set of organizations and contingencies for secondary design. For each case, primary design and final secondary design were compared, and observations and interviews were made whenever possible. Furthermore, we observed the use of the secondary design in practice. Due to access limitations, case 1 only contained observations and artifact analysis. Table 1 provides details of the two organizations and settings represented in our cases.

For our analytical lens of the two cases, we were inspired by the concepts defined by Walls et al. [9] on how to understand IS design theories and combined it with the anatomy of a design theory by Gregor and Jones [11]. Walls et al. [9] define components of design theories to consist of meta-requirements that are answered by a meta-design. Both meta-requirements and meta-design are based on so-called *kernel theories,* referring to theories from the knowledge base of existing research. While the model was originally conceptualized as a prescriptive way to form design *theories*, we found that by abstracting an instantiated artifact into a similar version of their model, one can infer the overall components of from that specific artifact. We combine the concepts of meta-requirements and meta-design with "principles of implementation" by Gregor and Jones [11] who use that type of principle to derive at the specific product of the design. We combine these two design theory models to identify two sets of principles for secondary design: one describing the design and on describing how to use the design, the implementation guidelines (see Fig. 1). Hence, we use it to describe the

Table 1. Case details

Case	Name (pseudonym)	Secondary design characteristics	Data collection
1	Insurance Company	Fundamentally redesigning a decision model for choosing agile or plan-driven development in a specific project	Observation of knowledge transfer (of primary design) Participation in three design workshops Observing use Analysis of secondary design artifacts
2	Hospital	A contingency model for choosing project management tools and techniques based on contingencies	Observation of secondary design process over 9 months 6 status interviews held Analysis of secondary design artifact

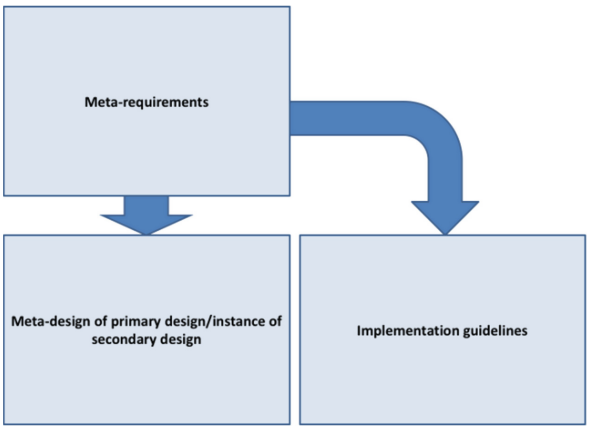

Fig. 1. Inferred components of an IS artifact design, combined from Walls et al. [9] and Gregor and Jones [11].

primary design and we use it show the secondary design giving us an overview of what secondary design involves. We specifically focus on meta-requirements, meta-design and implementation guidelines in this paper.

4 Analysis of the Two Cases

We first present the two cases' primary design and their secondary design counterparts, then we show the similarities by comparing them using the analytical lens and present the design implementation principles and their grounding.

4.1 The Insurance Company Case

The primary design in this case was a tool to assist an IT project manager in choosing between agile tools and techniques or a more classic plan-driven approach. The primary design was inspired by another designed artifact for the same purpose and used as a kernel theory [18]. The meta-requirements for the derived primary design came out of studies in many organizations considering agile but working in a classic plan-driven way. In these organizations, agile development was found to be suitable for only certain IT-projects, and there was a need for choosing between agile and plan-driven methods [19]. Another meta-requirement was that the choice of agile tools and techniques or plan-driven tools and techniques needed to be made early in an IT project again leading to the requirements that IT project managers need to be aware of core project characteristics early on. In Fig. 2 we have elicited the primary design from our analytical lens.

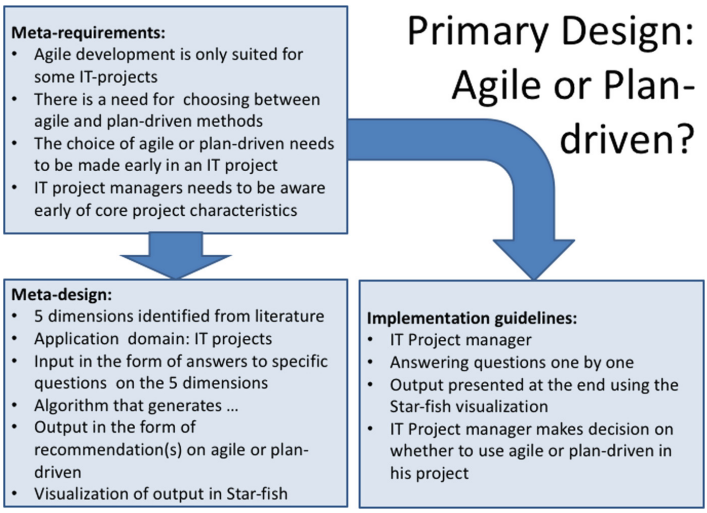

Fig. 2. The primary design in case insurance company

The primary meta-design (lower left box, Fig. 2) was based on a literature survey of what impacts the agile development: "requirements stability", "project size", "complexity", "project team", and "criticality". For example, "complexity" involved the ability to pre-define system requirements and scope as the central constraint [19]. An unclear scope makes requirements hard to define and an agile approach may be preferable because it will allow requirements to persist in near or full ambiguity (Fig. 3).

In the secondary design, the primary actor was a manager responsible for projects who needed to adapt the tool into the IT development department of the insurance company. The meta-requirements did not change as the organization still needed to be able to differentiate between choosing agile or plan-driven projects. The biggest

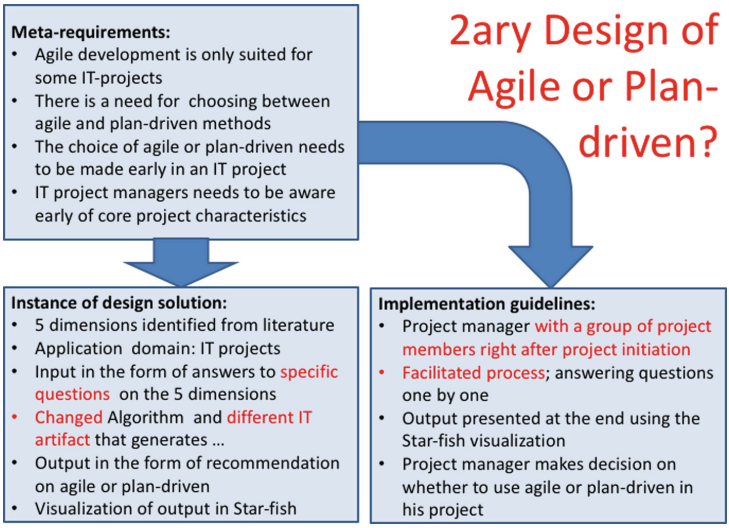

Fig. 3. Solution of the secondary design for case 1.

changes were the instantiation of the meta-design and its input in terms of the questions that were asked. For the secondary design to work and be comprehensible to the users, the questions needed to change, and included new questions on whether the project was staffed with full-time or part-time participants, and whether the result of the project was to be used by employees or customers. Furthermore, the algorithm for calculating a dimension was changed to assign different percentages to different dimensions and was built into an interactive spreadsheet.

The implementation guidelines were changed to be used in a workshop with the project manager and a chosen number of project participants. Furthermore, the workshop was facilitated using a projector so everyone could see and by one person from the "method and project office" who would fill out the interactive spreadsheet.

4.2 The Hospital Case

In Case 2, the primary design was called "the project radar" and was originally used as a tool to identify problematic issues of an existing project in the organization of Danske Bank [20]. The meta-requirements included the assumption that project management is a tool-heavy discipline and this can make it difficult for a project manager to use the right tool for the right problem in a project with varying variables such as size, aim, and number of stakeholders.

The artifact was an open technology incorporated into an interactive spreadsheet. A project manager would answer questions related to 8 dimensions including "task", "knowledge about", "individual and background", "environment", "team", "calendar time", "stakeholders", "quality/criticality" (identified from literature on project management). An algorithm for benchmarking would then generate recommendations for

the project as a whole. The dimension of "individual and background", for example, could be identified by a high score based on the amount of time available to project participants. Solutions included proposals for documenting decisions, or aim at uninterrupted, successive work days for all members in the project. The output was a radar chart visualization where the project manager could see the problematic areas.

The implementation guidelines included to have the project manager answer questions and get an output visualization afterwards. The tool worked as a reflection tool providing the project manager with relevant suggestions and techniques (Fig. 4).

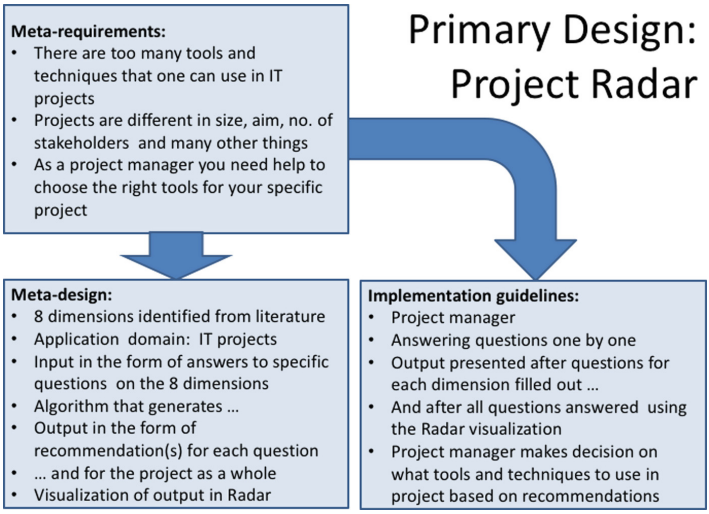

Fig. 4. Figure of the "project radar" primary design.

Two project managers that followed an Executive Master in Project Management were inspired by the primary design and felt that it was applicable in their own practical domain; a health care setting at a hospital. Both project managers had more than 20 years of working experience. The primary issue with how the hospital handled projects was that project teams were put together from random people from different departments. The requirements were more often than not based on their healthcare professional experience and competence and not on project management competences. As a result, they compared meta-requirements from the primary design with requirements from their own domain and found that only little knowledge existed on what project management tools were and how to select appropriately among them.

They tested out the primary design, the project radar, on 4 different projects. The two project managers would facilitate the test of the primary design by engaging in dialogue with the team members to help understand any project management specific jargon and also reshape the jargon used for the secondary design. The meta-design of the primary design was changed so first of all, a complex IT artifact was too technical for the target audience and instead, the project managers opted for evaluating by using

pen and paper and operate an interactive spreadsheet for inputting values (without the project team members knowing). They removed the dimensions of "individual and background" and "environment", and replaced them with "implementation" and "communication", as well as rephrasing new questions for these dimensions. Rather than showing the proposals for improvement, they would manually write up a report of their results with proposals for how to solve the identified problems. The reason was that a certain level of formality was needed when the team members would spend time away from their daily work in order to help with the tests of the secondary design. Furthermore, a formal report would increase the likelihood of receiving management support for their project. The purpose of the tool was changed to inspire the project group rather than letting their project manager decide on a course of action, since many of the projects did not have a dedicated project manager (Fig. 5).

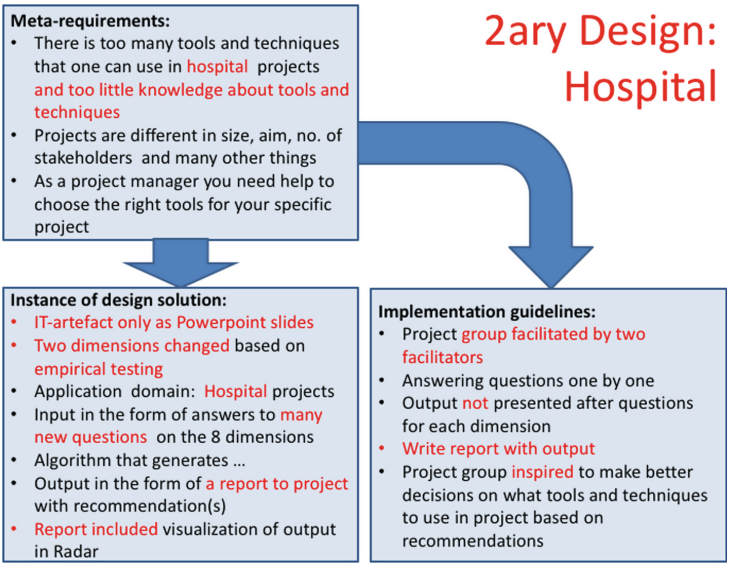

Fig. 5. Secondary design solution of case 2, hospital

5 Results – The Principles

From the two cases, we elicit two sets of four design implementation principles. The principles are dyadic in the sense that one set focuses on how primary designers can design the implementation of a primary design artifact to enable deep secondary design (changing both content and function). The other set is focused on how secondary designers should react so they can perform secondary design (shown in Table 2).

Table 2. Table of principles

Principle #	Design implementation principles for primary designer	Design implementation principles for secondary designer
1	Presenting meta-requirements	Understand meta-requirements and compare to own situation
2	Specifying relevance and advantage	Identify advantage and relevance to own domain
3	Unlocking meta-design and implementation guidelines	Understand all details of meta-design and guidelines and decide how supplementary needs can be met
4	Creating opportunities for trialability	Try out primary design and explicitly identify learning opportunities

5.1 Primary Design Implementation Principle 1: Presenting Meta-Requirements

This principle is the central enabler of deep secondary design. The primary designer is meant to present the meta-requirements grounded in the theoretical and empirical background, as well as show the domain-in-use and the compatibility. This principle is basically the design rationale that lets potential secondary designers understand the fundamentals for the design in the first place. It is up to the primary designer to convince the secondary designers that there is a problem or a need both theoretically and empirically. In Case 1, the primary designer had identified a need for assistance with choosing between agile and plan-driven methods. The right time to make this choice was identified as early as possible in the based on the dimensions identified in meta-requirements shown in Fig. 1. In Case 2, the meta-requirements were presented as part of the executive education in project management that problematized the standard use of project management tools and the need for deciding which of many tools were needed for a specific project. One supporting factor was the fact that the tool was a part of the education where the potential secondary designers already had a fundamental technical understanding of the project management field. The meta-requirements were thus presented as a review of existing knowledge in the project management field. The design rationale was presented through traditional class teaching, with the primary designers as facilitators responsible for creating an understanding of the artifact. Furthermore, the compatibility of the primary design was also grounded in empirical work, both grounded in prior sessions of the project management education as well as in peer-reviewed papers where the tool had been applied.

5.2 Secondary Design Implementation Principle 1: Understand Meta-Requirements and Compare to Own Situation

The logical extension for design implementation principle 1 for secondary designers is to make an effort into understanding the meta-requirements. This means thoroughly

researching the theoretical and empirical backgrounds for the design rationale. While it is not given that secondary designers will always strive for this, it seems that the motivation for understanding the meta-requirements is correlated to how well the primary designers have fulfilled principle 1. In Case 1, the secondary designers decided to adapt a number of the questions of the five dimensions based on their own situation in the insurance company. Another example was to change the whole implementation process to be facilitated in a group instead of individual use by a project manager. This again was based on what had worked before in the insurance company and on who initiated the secondary design. In Case 2, the fact that standardized project management tools had been presented and used already was a central factor for how thorough the secondary designers understood the primary design: *"It could be really interesting if we had the possibility to change some of those questions […] because we do not have projects with 300–400 employees, we have a completely different context." – Secondary designer, Hospital case 2.*

5.3 Primary Design Implementation Principle 2: Specifying Relevance and Advantage

The principle of specifying relevance and advantage includes that (a) the primary design must show its relevance by having a likeliness to the potential secondary designers' background, either through a common purpose within the same field of applicability or through linking a shared knowledge base (in this case: project management), and (b) the primary design must produce a solution that also solves a problem for the secondary designers. The principle was followed in Case 1 by being directly related to the secondary designers' current background; project management methodologies. The relevance aspect was solved by providing a simple decision: when to use agile or when to use plan-driven methodologies. As such the value in terms of relevance was simple in helping the users answer a problematic question. The principle was used in Case 2 through simplifying the variables of projects into simple answers that most project members could answer on a scale. The aspect of relevance was shown by having a list of relevant solutions of actions tailored to the specific projects that users needed: *"We need to be able to find a score and recommend x and y, because that is what helped us; how did it look in practice?" – Secondary designer, Hospital case 2.*

5.4 Secondary Design Implementation Principle 2: Identify Advantage and Relevance to Own Domain

While it is necessary for the primary designers to show relevance and advantages, it is impossible to specify completely due to the generic nature of primary design. The secondary principle following this is the principle of identifying relevance and advantages for the secondary designers' own domain. The secondary designers should compare the relevance and advantages to find areas where they could benefit from tweaking. In Case 1, the primary design had been presented to the insurance company with examples from the company – Danske Bank – where it was originally developed.

The insurance company recognized the need for a similar tool and some similarities between banking and insurance – administrative IT projects – that made it relevant to adapt to the insurance domain. In Case 2, an immediate need was identified and it was established that projects in the hospital were initiated based on healthcare related competences and not on the project competences. As a result, the primary design was seen as a better alternative to expensive, standardized project management education.

5.5 Primary Design Implementation Principle 3: Unlocking Meta-Design and Implementation Guidelines

The principle of unlocking meta-design and method indicates an open approach to technology. The principle covers four layers: (1) the content layer, (2) the algorithmic layer, (3) the complexity layer, and (4) the procedural layer. Unlocking the content layer makes it possible to change what is shown and what the purpose of the design is. Unlocking the algorithmic layer involves re-arranging the components and/or scoring of values between them. Unlocking the complexity layer means that the level of technology can be scaled all the way from paper-based, manual completion to high-tech, full-fledged IT application. Unlocking the procedural layer involves being able to change the use setting and the actors involved in this, e.g. from individual use to collective use, or to assessing a context area individually or collectively. In Case 1, we saw how the meta-design was flexible enough to include changes in questions and presentation, as well as how the practice of assessment was flexible enough to be used either as facilitative dialogue or personal reflection. In Case 2, the initial primary design was based on 8 dimensions but could easily be changed to more or fewer, with also the questions and benchmarking being open to change. Likewise, the different levels of complexity of the technology were already present in that several versions were created to show the various applications. One version was based on slideshows and manual, paper-based filling out questions, while a more embedded and structured version was also created to automatically visualize the results in a spreadsheet.

5.6 Secondary Design Implementation Principle 3: Understand All Details of Meta-Design and Guidelines and Decide How Supplementary Needs Can Be Met

For a secondary designer to take advantage of primary design implementation principle 3, it is necessary for the secondary designer to understand the underlying technology itself and decide how supplementary needs can be met in the mentioned layers. This can be done from experience or by being inspired by various versions of the primary design. In Case 1, supplementary needs were primarily met through changes in the specific questions, how the underlying algorithm was changed to take into account different benchmarking and the level of complexity of the IT artifact. Furthermore, they realized a need for a facilitated workshop process as opposed to an individual stand-alone process. In Case 2, two new dimensions replaced older ones to accommodate the specific domain, and new questions were included to reflect this. The

secondary design tool was assessed with facilitators and rather than having immediate results with solutions, the designers decided to formalize a written report with specific suggestions.

5.7 Primary Design Implementation Principle 4: Creating Opportunities of Trialability

This principle means that the primary designer should actively design so that secondary designers get experience with the primary design from beginning to end. As artifact complexity increases, the designated workflow can also be difficult to overview without having tried it out. This entails that the primary design should be kept simple as high complexity can make it difficult for secondary users to get the required experience to comfortably change it. In Case 1, this was done by testing the artifact (spreadsheet) with the (updated and changed) questions and benchmarking three carefully selected projects. In these projects, the project manager had his assumptions challenged because the project management methodology has more or less been chosen in advance. In Case 2, the primary design was tested out in a class setting with a project that the students were already familiar with: *"My great "Heureka" moment was after using [the tool] […] and we were visually able to see where our challenges were and confirm our suspicions we had when we were part of the project." – Secondary designer B, Hospital case 2.*

Since the primary design did not require real life subjects or real data, it could easily be tried out. Furthermore, the testing of the design also created opportunities for the secondary designers to identify relevance and advantages.

5.8 Secondary Design Implementation Principle 4: Try Out Primary Design and Explicitly Identify Learning Opportunities

The principle that follows here is that of actively trying out the primary design and explicitly identifying learning opportunities. The principle requires that the secondary designers gain hands-on experience with how the design is structured and used to properly estimate what needs to change. This principle further supports changing the design to accommodate a new domain-of-use.

In Case 1, the secondary designers learned that their three projects had a high interdependence and that if one project changed to agile, the other projects would also have to change to the same methodology. As a result, the level of interdependencies to other projects was built into the artifact as an addition to the secondary design during the final two workshops.

In Case 2, the two secondary designers saw that testing the primary design helped them gain the confidence they needed to assess required changes: *"If we are going to test it out for real and redesign it, adapt it to our world, we need to do [prototyping]. We cannot just change it. Now we have knowledge and experience which make it possible for us to start somewhere, and much more qualified." – Secondary designer, Hospital case 2.*

6 Discussion

We have now provided four dyadic "design implementation principles" that need to be applied in unison to enable *deep* secondary design. We contribute with the concept of deep secondary design by defining it as a redesign of a primary design that goes beyond original intentions in both domain-of-use, as well as include both changes to the function and content layer. We further contribute with findings of secondary design within an *organizational* domain, which hitherto has been described as being inhibitive for the secondary design process [4]. On the contrary, we found that by following the principles, secondary design in organizations certainly is viable.

However, the principles strongly relate to taking advantage of when to create a relationship between primary and secondary designers. Prior research on relationship between designers and users, (e.g. in co-realisation [13] or participatory design [14]) have noted the requirements of a flat and symmetrical relationship. While the primary designers were both consultants (in Case 1) and educators (in Case 2), the principles still created a short, facilitative role of the primary designers that enabled the secondary design. Especially the two principles of presenting the meta-requirements and testing out the primary design seemed to be important factors. One of the reasons for this was that rather than focusing on redesigning the product, the primary designers focused on assisting with the process of individual adoption and redesign, with no expectations of benefit realization of the product. The lack of expectations or straightforward cost-benefit measurement is also an important point made by Richter and Riemer [16] when supporting malleable end-user software. Our findings supported this and we also call for more research on applying secondary design principles as a diffusion process in organizational settings.

Our findings also extend the understanding of what "principles" are within DSR. Much literature within DSR has focused on principles of artifact design features, while the importance of having various editions of the artifact on different complexity levels (e.g. from paper to structured and embedded web-apps) have not been an important of the final design. This is seen in Gill et al. [6] who suggests that artifact designs make use of "openness" to enable malleability or in Germonprez et al. [12] who underscore the importance of componentization of the design. While this is important, it overlooks the importance of the use domain where secondary designers and their users might not have high technical design competences.

It is also possible that our findings only extend to the class of information system of our cases, a type of facilitative decision support system. As a result, we also call for more research on deep secondary design with other types of technologies to.

7 Conclusion

We have now defined deep secondary design as a phenomenon where secondary designers change function and content, as well as the complexity level of the technology. We have described and analyzed two cases - all from the project management domain - where secondary designers fundamentally changed both meta-design and implementation guidelines as the result of how the primary designers implemented the

primary designs. Our analysis of the two cases led to the identification of four principles of design implementation that primary designers can apply to enable secondary design and four corresponding principles that secondary designers themselves can apply. We contribute with these two-by-four principles that form a "nascent" theory on deep secondary design. Our practical contribution can help primary designers reflect on what they do with their design rather than how they designed its features. For users with the potential to become secondary designers, we practically propose actions that they need to perform to better grasp how to become better designers.

References

1. von Hippel, E.: Lead users: a source of novel product concepts. Manag. Sci. **32**, 791–805 (1986)
2. Kensing, F., Blomberg, J.: Participatory design: issues and concerns. Comput. Support. Coop. Work **7**, 167–185 (1998)
3. Orlikowski, W.J., Hofman, D.J.: An improvisational model for change management: the case of groupware technologies. In: Malone, T.W., Laubacher, R., Scott Morton, M.S. (eds.) Inventing the Organizations of the 21st Century, pp. 265–282. Sloan School Management, Cambridge (1997)
4. Germonprez, M., Hovorka, D., Gal, U.: Secondary design: a case of behavioral design. J. Assoc. Inf. Syst. **12**, 662–683 (2011)
5. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**, 75–105 (2004)
6. Gill, T.G., Hevner, A.R.: A fitness-utility model for design science research. ACM Trans. Manag. Inf. Syst. (TMIS) **4**(2), 5 (2013)
7. Gregor, S.: The nature of theory in information systems of theory in information systems. MIS Q. **30**, 611–642 (2006)
8. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. MIS Q. **37**, 337–355 (2013)
9. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an information system design theory for vigilant EIS. Inf. Syst. Res. **3**, 36–59 (1992)
10. Heinrich, P., Schwabe, G.: Communicating nascent design theories on innovative information systems through multi-grounded design principles. In: Tremblay, M.C., VanderMeer, D., Rothenberger, M., Gupta, A., Yoon, V. (eds.) DESRIST 2014. LNCS, vol. 8463, pp. 148–163. Springer, Cham (2014). doi:10.1007/978-3-319-06701-8_10
11. Gregor, S., Jones, D.: The anatomy of a design theory. J. Assoc. Inf. Syst. **8**, 312–335 (2007)
12. Germonprez, M., Hovorka, D., Collopy, F.: A theory of tailorable technology design. J. Assoc. Inf. Syst. **8**, 351–367 (2007)
13. Hartswood, M., Procter, R., Slack, R., Voß, A., Büscher, M., Rouchy, P.: Co-realisation and participatory design. Scand. J. Inf. Syst. **14**, 9–30 (2002)
14. Simonsen, J., Robertson, T.: Routledge International Handbook of Participatory Design. Routledge, London (2013)
15. Davern, B.Y.M.J., Wilkin, C.L.: Evolving innovations through design and use. Commun. ACM **51**, 133–137 (2008)
16. Richter, A., Riemer, K.: Malleable end-user software. Bus. Inf. Syst. Eng. **5**, 195–197 (2013)
17. Yin, R.K.: Case Study Research: Design and Methods. Sage Publications, Thousand Oaks (2009)

18. Boehm, B.W., Turner, R.: Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley, Boston (2004)
19. Baskerville, R., Pries-Heje, J., Madsen, S.: Post-agility: what follows a decade of agility? Inf. Softw. Technol. **53**, 543–555 (2011)
20. Avison, D., Pries-Heje, J.: Flexible information systems development: designing an appropriate methodology for different situations. In: Proceedings of the International Conference on Enterprise Information Systems 2007, pp. 212–224 (2008)