

Making *Use* of Design Principles

Leona Chandra Kruse¹, Stefan Seidel¹(✉), and Sandeep Purao²

¹ University of Liechtenstein, Vaduz, Liechtenstein
{leona.chandra, stefan.seidel}@uni.li

² Bentley University, Waltham, USA
spurao@bentley.edu

Abstract. This paper reports on the results of a study that investigates how design principles are used in design practice. Design principles have become the predominant way to capture abstract knowledge about the design of information systems (IS) artifacts—and as design science researchers, we expect that practitioners will use these outcomes of our work. Our empirical evidence is drawn from the analysis of spoken-out thought processes of designers as they attempted to use a certain set of design principles in a new context. Through our analysis, we identify five key categories conceptualizing the use of design principles: interpreting scope and content, matching with problem space, guesstimating missing information, projecting into solution space, and implanting into design process. We find that design principles do not shut down degrees of freedom, but rather, channel actions from the designer, who acts in a conscious, deliberative manner to creatively apply the design principles. Through our work, we contribute to (a) our understanding of how design principles are incorporated in design processes, and (b) the emergent stream of research about the formulation of design principles.

Keywords: Design principles · Prescriptive knowledge reuse · DSR and practice · Designer knowledge

1 Introduction

Contemporary research has argued that an appropriate outcome from design science efforts is knowledge in the form of ‘design principles’ [1, 2]. Design principles capture knowledge about the creation of other instances of artifacts belonging to the same class [2] and are derived as the researcher moves from the specific instance to the more generic level. An implicit assumption in this argument is that design principles are the right form of abstraction that would allow the community to accumulate knowledge that may then be useful to both, the next cycle of research as well as for design practice.

Unlike other modes of research (e.g., positivist or interpretive), design science inherently builds on the assumption that designing and intervening in authentic settings can generate new knowledge [2] that would be of benefit to the next generation of practitioners. The assumption essentially makes an end-run around the argument for ‘making the research relevant’ by tackling problems that practitioners have encountered or may encounter. The intent of the design science researcher is, then, unabashedly one

of generating knowledge that will aid the next generation of practitioners who may need to address similar problems [3].

As design science researchers, we expect that practitioners will then, use outcomes of our work. An implicit assumption we make here is that they will simply “follow the recipe” we give them. Unfortunately, much prior work tells us that such use (reuse) is not a natural adjunct to creative problem solving. Scholars describe phenomena such as “NIH” (not-invented-here) that have hampered use of prior knowledge in new situations [4]. The expectation that the design knowledge we generate and codify in the form of design principles “will simply be used in new situations” by the practitioners may, therefore, be naïve.

To understand whether and how practitioners will use design principles, a number of conditions need investigation. For example, there are several other forms of design knowledge, e.g., scenarios [5], patterns [6], or heuristics [7]. Each may be more or less useful depending upon the problem context [8]. The designer may bring little or substantial design experience or domain knowledge to the design situation, which will further influence how she will use prior design knowledge such as design principles. Insights from studies of designers and programmers tell us that the breakdowns [9] that occur in design efforts can trigger the search for such knowledge. Empirical studies of design [10, 11] also suggest that the messy nature of design processes can prevent the injection of external sources, including codified prior knowledge. And our understanding of design as a non-routine activity [12] warns us that design, particularly in complex settings, tends to be a situated activity [13] that must respect the knowledge brought to the endeavor by both, users and designers [14] who must then translate and integrate information from a number of sources, one of which happens to be the design principles. We thus ask:

How might (how do) designers use design principles (as a form of design knowledge) in new design contexts?

We believe that a number of designer behaviors are likely to manifest in such an examination. These include searching for sources of information [15], structuring the problem space [16], moving from the problem space to the solution space [16], evaluating the design alternatives, and so on. Each has been examined extensively in prior research. Our interest is *not* to examine these activities in themselves. Instead, our particular focus is on understanding how designers use design principles as they engage in the design process, which may include the above activities. In particular, we are interested in the processes of knowledge access, knowledge translation, and knowledge application that designers engage in—with a focus on design principles as the form of knowledge. Such knowledge is crucial as the design science research community attempts to develop a cumulative body of design knowledge.

We proceed as follows. In the next section, we describe how design knowledge is used in the practice of design, thereby focusing on relevant knowledge components and the understanding of design as a contextual activity. Next, we describe our research design. The analysis section ensues, and we conclude with a discussion of important implications for both research and practice and provide an outlook.

2 Using Design Knowledge in the Practice of Design

Experts and scholars argue and agree that design can never be decontextualized. Scholars in many research streams also agree, starting from Rittel and Weber [17], that design remains a wicked problem that starts with a user concern with no obvious answer, that is, it “is never a process that begins from scratch: to design is always to redesign. There is always something that exists first as a given, as an issue, as a problem” [18, p. 5]. The contextual nature of design poses a challenge for the creation and use of codified design knowledge (e.g., in the form of design principles), and thus for the development of a cumulative body of design knowledge created by design researchers. Codified design knowledge can be represented in various forms such as design patterns [e.g., 19, 20], technological rules [e.g., 21], analysis patterns [e.g., 22], and design principles [e.g., 2, 23]. In this paper, we focus on design principles, defined as “knowledge about creating other instances of artifacts that belong to the same class” [2, p. 39].

The definition suggests that design principles must be codified in some way and must be applicable to a class of problems. We may expect that information systems (IS) practitioners would use such design principles to produce reliable outcomes. On the other hand, the design of IT artifacts is thought to be ongoing and complex, shaped by their organizational context during development and use [2]. As a result, design principles must be understood in relation to the (often novel) contexts in which they are used. To understand how design knowledge is used in design practice and what makes design knowledge usable in practice, we thus need to get at the inherent tension between the formulation design principles that follows a nomothetic approach about how to design a class of things and their idiosyncratic use in highly contextual design practice. We can decompose this tension into two important aspects: (1) design is an activity that involves use of both codified and non-codified (un-codifiable) knowledge and (2) design is an activity that is repeatedly carried out across contexts and time to address similar, but still context-dependent and novel, problems.

2.1 Codified and Non-codified Knowledge in Design

Prior work highlights the epistemological differences in the spectrum of design knowledge, ranging from the more tangible or explicit knowledge to the more intangible or tacit [24–26]. Design Principles (in so far as they are considered a form of design knowledge) represent knowledge that is codified, explicit knowledge, readily accessible as prescriptive statements. Prior to the advances in design science research, such codified knowledge might have resulted from the externalization of tacit knowledge gained from design experience, derived from the combination of existent explicit design knowledge, or both. The design science research community is now engaged in explicating such principles from their efforts to generate IT artifacts in novel situations. Still, the caveats from prior work apply. Not everything can be explicated, “we can know more than we can tell” [27, p. 4], and “a wholly explicit knowledge is unthinkable” [p. 144]. This view suggests that there are limits on what can be expressed through design principles—and implicitly argues that not everything required for design practice can, or should, be explicated.

Using this position as a starting point, we may argue that when designers make use of design principles, they convert explicit knowledge (i.e., prescriptive statements that are shared in the form of design principles) into practice. This conversion is heavily influenced by designers’ tacit knowledge base—both, about the new context as well as the class of problems for which the design principles have been explicated [28]. One approach to better understand the relationship between tacit and explicit knowledge is to use labels such as “know-how,” “know-why,” and “know-what” [29]. Although we know that these categories are not mutually exclusive, there are differences. Garud [29], elaborates these as follows: “in the context of technological systems, knowledge represents an understanding of the principles that underlie their functioning, processes employed to create them, and the uses that these technological systems serve” [p. 83]. Simply put, know-how is usually acquired through learning by doing, know-why through learning by studying, and know-what through learning by using. We suspect that these three categories come together as designers make use of design principles. We anticipate that understanding the interactions of these components in the process of design will, in turn, help us better understand what *should* (and more importantly, *should not*) be captured in design principles.

2.2 Idiosyncrasy and Generalization in Design

Although design is a contextual activity [e.g., 18, 30], design principles are about a class of problems [1] and are thus intended for application across contexts and across time. These contexts *are* different even though they might share certain boundary conditions. Design principles, thus, carry multiple possible meanings as they are interpreted differently in different contexts according to the need and purpose. This characteristic is crucial in understanding how design principles are used in different contexts. In essence, utilizing design principles means that designers must write their own version of those principles—the knowledge contained in these design principles is not “cast in concrete” [31, p. 289].

Understanding the use of design principles thus requires us to investigate the activities that occur when designers “write their own version” as they use design principles in various design contexts. This phenomenon is related to the degree of novelty as well as the complexity of the design situation in knowledge storage, retrieval, and transformation [32, 33]. So, in low complexity situations (such as deterministic change of non-human task through a more efficient algorithm) knowledge might simply be transferred, whereas higher level complexity situations (such that impact human tasks) might require that designers translate or transform knowledge [33]. In more novel situations, the focus shifts from one of efficiently using knowledge to one of effectively creating knowledge [32, p. 1192].

3 Research Approach

To explore how designers make use of design principles in the early phase of the process of designing artifacts, we captured and analyzed their thought processes during as they attempted to use a certain set of design principles in a new context. This

allowed us to gain insights into how they *use* design principles during the process of design. Contrary to the common self-report method, which is dependent upon vagaries of recall and interpretation [34], we collected data while the designers were engaged in active use of design principles, i.e., as they attempted to design a representation of a proposed IS artifact. As the participants thought aloud, their words were recorded.

3.1 Collecting and Analyzing Think-Aloud Protocols

Think-aloud protocol analysis places an emphasis on “externalizing covert thinking without altering it” [34, p. 180]. This means that reactive thinking (describing and explaining—and therefore transforming—thoughts) is not encouraged. The method is based on the assumption that thinking is a sequential process and can therefore be represented as a sequence of thoughts [34, 35]. The closest connection between thinking and verbal reports (and therefore, the least altered externalized thinking) can be established by asking participants to think aloud while completing tasks [34]. This method of eliciting verbalized thoughts of designers while designing has been widely used in studies related to problem solving and cognitive analysis of design activity in various design fields, from architecture [e.g., 36, 37] to usability studies (for a comprehensive review see Cross [30, 37]). Table 1 summarizes key elements of the work we carried out to collect the think-aloud protocols.

Table 1. Collecting think-aloud protocols

Category	Description
Task	Designing a representation of a Clinical Decision Support System (CDSS)
Participants	Professional IT/IS designers, one participant for each session
Stimulus	A set of design principles for CDSS and a problem description
Instrument	Audio recorder, paper and pencil, printed instruction for participant (ensuring all participants receive the same instruction) that incorporates problem formulation and design principles
Duration	37 min (pilot), 35 min (participant P), and 31 min (participant Q), including training phase

The method we used aimed to elicit a verbal protocol [38] from each individual designer (i.e., excluding collaboration) as they made use of design principles in response to a problem scenario. Each session involved one participant and one researcher (in the case of our pilot, two), as described in Table 1. The data collection effort was divided into the four phases: preparation, training, design task, and debriefing. During the **preparation phase** each participant was formally asked for her/his consent to participate in this study. However, the researcher did not provide any information regarding the actual research objectives. The setting chosen for the experiment provided sufficient comfort and tranquility for the participant to feel at ease during the session, because it required them to think (talk) aloud. Participants were asked for their agreement to audio recording the session. Finally, participants were

requested to give a brief description of their professional background, expertise, and experience in designing software or IS. In the **training phase** the participants were guided through a round of exercises for verbalizing their thoughts. When the researcher was satisfied with how the participant thought aloud, they moved to the **design task**. Here, the printed instruction including the problem formulation and the set of design principles were shared with the participants. The participants were also provided with a sketchbook and drawing/writing instruments. Before beginning the design task, participants were reminded to verbalize their thoughts, and encouraged to do so as a way of thinking instead of trying to describe or explain what they were doing, i.e., they were requested to focus on the design task itself. The researcher read the printed instructions and asked the participant to read both problem formulation and design principles aloud. A **debriefing phase** followed at the end of the design session. Participants were informed about the general aim of this study and were again asked for their continued consent, and further correspondence, should questions arise during analysis.

3.2 Design Principles and Scenario

The participants were provided with a scenario that was hypothetical, yet grounded in an authentic design science research project [39]. This ensured that the scenario was similar to what one might encounter in practice. They were also given a set of design principles. Table 2 provides an overview of the scenario and

Table 3 shows two example design principles along with a short description.

Table 2. Problem scenario for the design session

<p>Marco is a lead software designer. His company has just acquired a contract to develop a piece of software that will support doctors who deal with deciding patient treatment at the local Cancer Care Center (CCC). The patient treatment decision is difficult because it requires a combination of perspectives and expertise from a number of specialists. These specialists are able to meet only about once every month, because they travel from different locations. At CCC, their meetings tend to run for about 2 hours and at each meeting, they end up considering as many as 15 patients. The meetings are attended by about 10-12 specialists in different fields who discuss each case and deliberate to finalize the treatment for each patient before moving to the next patient. These decisions can be very critical because they have consequences that may be irreversible.</p>

Table 3. Design principles

Number of design principles	9
Two examples shown	
Principle 3: make available and visible case-relevant knowledge requested	Pre-populate case-relevant knowledge instead of relying on search-on-demand
Principle 8: preserve group memory	Record deliberations and decisions for each case

3.3 Participants

We collected data from three participants, all IT/IS designers with more than ten years of experience. The session with the first participant served as a pilot session and we thus excluded the resulting protocol from our analysis. Still, the pilot session assisted us in improving the clarity of instructions and the readability of both problem formulation and design principles. All sessions were conducted in German. The instructions, problem formulation, and design principles were made available in both German and English. The sessions were, however, conducted in German because we wished to allow the participants structure and describe their thoughts in the language they were most comfortable with, in spite of their proficiency in English.

The profile of the two participants (excluding the pilot) was as follows. The first participant (P) has a background in software design and engineering, but currently works as a business project manager in a large financial institution. P indicated the necessity to understand the logics of both business analysis and systems development in order to perform well in his function. More often than not, this participant must be able to translate the needs and implications from one side to another and vice versa. The second participant (Q) began his career in a software engineering company before eventually started up his own company in the same field. Being one of the leaders in the company, Q is also responsible for maintaining client relationships, and his expertise includes understanding client needs as well as explaining complex software systems in a simple yet intelligible manner.

3.4 Data Analysis

The recordings of participants' spoken thoughts were transcribed verbatim. Due to the exploratory nature of this study, we aimed to achieve the closest textual representation of the recorded audio—including indication of pause, stutter, murmur, and other unintelligible or incomplete words. Through our data analysis, we intended to explore the observed behaviors of designers as they tried to use design principles in a new, given context. In order to remain open, we thus coded the data in the spirit of open coding [40] to identify salient categories related to the use of design principles. We omitted those segments that were unrelated to making use of the design principles, for instance, training segments and instruction segments.

4 Findings

From our analysis, a number of categories emerged that highlight how designers use design principles. Table 4 provides an overview of these categories along with a brief description.

Next, we describe the categories and illustrate them with quotes from the data. Due to constraints on space, we provide only illustrative quotes. Additional details such as frequency or sequencing of our observations are not shown.

Table 4. How designers *use* design principles

Category	Description
Interpreting scope and content	Designers create meaning of, and reframe, the design principles against the given scenario and against their personal background
Matching with problem space	Designers match the design principles with the given application scenario
Guesstimating missing information	As designers experience incompleteness and degrees of freedom they make assumptions and draw analogies
Projecting into solution space	Designers translate design principles into more specific requirements of form, function, and usefulness
Implanting into design process	Designers attempt to embed design principles into a design/software development process

4.1 Interpreting Scope and Content

Our data suggests that designers interpret design principles and create meaning under consideration of the application context. That is, the data suggests that different designers apply the same design principle differently, even when given the same scenario. Participant P, for instance, said with regard to one of the principles:

...so that we can update the data again before the discussion... so timely beforehand still... That's what I think...that's what I make out of "plan for and ensure accurate and current content."

When interpreting the same principle, participant Q mentioned:

Contents that are accurate and actual... that's... based on my experience always difficult, to maintain the data. It's a big topic. Here it can be done through... partly through... user guidance, so that the users tell us all kinds of data they need and so that the data can always be updated.

In this statement it becomes also noticeable that in interpreting design principles, designers further draw on their personal background and experience. The interpretation of design principles is deliberative and conscious, and designers indeed write their own version of those principles. The need to interpret design principles also becomes evident in the language the designers used when thinking loudly, for instance, Participant P made repeated use of the phrase “it means...”:

...it means, the templates and later the decisions... and everything must be an integral part...

Collectively, these quotes suggest the following. As designers use the design principles, they exercise certain degrees of freedom. A generous interpretation of our data suggests that design principles do not prevent such degrees of freedom; instead, they channel the designers’ creative actions towards a narrower band of creativity. This is further elaborated by the categories to which we turn next.

4.2 Matching with Problem Space

Designers match design principles with the problem space. This appears to involve two conscious acts. First, they weigh the design principles under consideration of their domain knowledge [41]. In doing so, they map the abstract form of the design principle against their understanding of the problem domain. Second, they attempt to match an implicitly instantiated form of the design principle against the specific situation at hand. Because design principles are an abstraction, this second act seems to require more effort. Consider, for example, the following statement from Q, that points to both:

... data security can be a discussion, since it's about medical data. [...] the problem lies a bit in the enterprise that manages it and where the data is stored, in which country. (Q)

Here, Q points out that the system design needs to consider aspects of healthcare regulations in terms of legal requirements of the organization or country, thus clearly relating the discussion to his/her domain expertise. Participant P makes similar moves in noticing the time pressure while discussing and making decisions about each patient when matching that aspect of the problem space against another design principle:

It means someone must prepare the case-relevant knowledge...what I'd now implicitly assume is...that each of the cases comes from each of the experts, probably it's still...prepared by each of their employees. Otherwise you can never ever discuss everything in five to ten minutes. (P)

However, we observed fewer traces in the data showing such engagement from the participants—in our case, expert designers—with the problem space than with the solution space. This observation is consistent with some of the findings of novice versus expert studies [e.g., 42] and relative focus of designers in both problem and design spaces [43].

4.3 Guesstimating Missing Information

The developers we studied perceived the design principles as lacking detail and providing certain degrees of freedom. This behavior thus logically follows from the interpretation of scope and content. As designers guesstimate this missing information, design principles appear to ‘come alive.’ In effect, the designers write their own version of those principles closely matched against the situation under consideration. In other words, guesstimating becomes an important process where designers act creatively within the boundaries provided by the design principles. For instance, participant P noted:

... but many details are still open, which I must then still pick up.

and

I need then more detailed information so that I can design the system completely and eventually I will have to go one step back...

It thus becomes noticeable that, in order to design an actual system, more information is required than is provided by the design principles. It is perhaps not surprising

that designers have to make assumptions if the design principles are the only formal source of reference. Participant P said:

...what I'd now implicitly assume is...that each of the cases comes from each of the experts, probably it's still...prepared by each of their employees. Otherwise you can never ever discuss everything in five to ten minutes. (P)

P also noted:

what's not stated in any design principle here...but I take it for granted, that one can't simply access the patient data at any time, but they are released...just in time before the discussion... those are private anyway...

The participant thus uses examples or assumptions from his professional experience (i.e., drawing on design and domain knowledge) and injects those into the description of the artifact/solution domain. Finally, it was expressed that the same design principle may have different meanings, from the participant's point of view:

it means that...well this principle may also mean that we... it can have double meanings... on the one hand it means that one completely allows the access to... (P)

This conscious effort continues on the part of the designers. Not only do they interpret the design principles and deal with the lack of detail, but they also engage in translating the principles into concrete possibilities by considering the solution space.

4.4 Projecting into Solution Space

Both subjects showed a clear tendency to translate the design principles into what we might call “requirements,” that is, physical and functional needs that a particular design, product, or process must be able to perform. This included both: (a) specific functions and their technical implementation details, and (b) envisioned user roles. At this, the degrees of freedom turned out to be important, as the participants identified various potential implementations, for instance:

...so it means everyone has a laptop or something like that, look at the date on his own, and for that the whole group an expert would...ok it's then... now I'd just say it like this...through...a... power point or similar presentation, where maybe keywords are presented and the facts revealed, and then each can still... (P)

This designer mentions the availability of a laptop computer or the use of Power Point as potential, appropriate means to allow for multiple presentation modalities (as stated in one of the principles). Not only did the participants speculate about potential implementations, but also about the quality of the system. For instance:

It means also, that it needs to be easy to use, or that... that the UX...the user experience and design should be relatively simple, because the one it, yeah, broad things... or it has many different users... (Q)

Participant Q highlights the need to design a system that is easy to use [44] because its users will be doctors, who are non-IT experts. This was an interesting observation as none of the design principles expressly states the need to develop an easy-to-use

system. Similarly, Q talks about “usability,” while considering suggestions from another principle (in this case, “principle one”):

...principle one... so, that has much to do with usability, so a usability... problem... so that it becomes an integral part... it happens often that a user... or if the system should be user friendly...

4.5 Implanting into the Design Process

Both participants highlighted that design principles need to be embedded into a more comprehensive design process. The design process they were implicitly considering seemed to involve stages such as user observations, prototyping, and/or testing, for instance:

Afterwards we can actually start with a simple mock up or a click-prototype, that's how we call it, which has the real logic... or that can specify and demonstrate the user experience, how the sequence may look like with the mock up simply... or a few cases, those cases can be invented, and to review the workflow with some doctors. (Q).

Similarly, the same participant elaborates with regard to one of the principles, how users might be included in the design process:

Here it can be done through... partly through... user guidance, so that the users tell us all kinds of data they need and so that the data can always be updated... it can be done through a user guidance in asking questions.

These articulations appeared to indicate that the designers placed the design principles in a web of action that they considered appropriate to deal with the context—regardless of whether such guidance was available within the set of design principles.

5 Discussion and Implications

The data we have presented and the observations tied to this data suggest a number of possibilities. First, with regard to interpreting scope and content, our study raises important questions about whether and how boundary conditions of design principles may be specified. In our example case, the boundary conditions were provided by the scenario, but the design principles themselves did not contain any explicit boundary conditions. Prior literature on abstract design knowledge has highlighted the importance of boundary conditions, or scope, when developing such knowledge [3, 45]. Consequently, there is need to investigate if, and how, boundary conditions and scope might be considered in the development and formulation of design principles. Such knowledge becomes important as designers evaluate the applicability of design principles in specific contexts.

Second, matching the design principles with the problem space surfaces as an important, yet intricate, step in applying design principles. By their very nature, design principles are abstract. Matching them with the problem space requires the designer to act creatively by applying both her expertise and domain knowledge [41] in particular.

In our case, the participants were not experts in the field of healthcare, and did not have the opportunity to rely on such domain expertise. The design principles appeared to fill some of this lack of expertise by providing prescriptive guidelines, which the designers then seemed to weigh against their own knowledge-base.

Third, with regards to guesstimating, our study indicates that the formulation of design principles might lead to ambiguity and various degrees of freedom. Still, there are good reasons to argue that such freedom is required to allow for creativity and innovation in the design process. Design principles are not a simple set of instructions to be followed blindly. In fact, they do not contain procedural knowledge at such level of detail. To understand and enact design principles, designers must possess professional knowledge such as design skills and domain knowledge. It is our contention that it will be important to further investigate the appropriate levels of detail and abstraction that design principles must contain. These decisions would be important as the design science research community engages in the larger goal of (a) developing a cumulative, reusable body of knowledge without (b) compromising creativity and innovation.

Fourth, our analysis highlights that a key challenge is seen in translating design principles into more concrete requirements of form and function, as the participants in our sample struggled to understand ‘what exactly’ some of the design principles meant, and how they might be converted into something tangible like a concrete instantiation [46]. Coupled with the previous observation, this struggle points to the question of ‘what exactly is enough’ specification within a design principle.

Fifth, it became noticeable from the data we collected that designers naturally place the design principles into a design process. One promising approach to further study the use of design principles in practice might thus be through the lens of various software development approaches such as waterfall, spiral, or agile methods [47]. Further, it might be useful to differentiate between different types of design principles, for instance, design principles of form and function and design process principles [45], and expressly consider the stage at which the different design principles are intended to be useful. Such insights will further improve our understanding of how design principles should be specified.

Finally, we hope that our initial effort to understand how designers make use of design principles will be followed by comparative studies that include other forms or representations of design knowledge. Such efforts will contribute to the cumulative body of knowledge about the design of design principles, design knowledge reuse, and might have important implications for the overall design process.

6 Next Steps

This paper continues the ongoing discourse in DSR about studying designers’ activity, i.e., “studying design” [48] or “ethnography to design” [49] as a mode of research. The study we have reported is, however, not yet another study of design processes. Instead, our effort is to consciously bring together the two streams of “doing design research” and “studying design” as argued in [42]. By doing so, we hope to make two contributions. The first is about the design processes themselves, based on our investigation of the (re) use of a new form of design knowledge (design principles). However, as

pointed out earlier, our effort is not to examine the design process in and of itself—but rather, focus on how it incorporates design principles. This is our interest because we wish to implicitly test the assertion that design principles are an appropriate knowledge form that design science researchers should develop. The second contribution is to the growing body of writing about design principles. By understanding how future designers *use* design principles in actual design situations, we hope that we can feed back to design science researchers who can then use the lessons learned to formulate more effective and useful design principles. We hope that the work reported will start this dialog.

Acknowledgements. This research is funded by the Research Fund of the University of Liechtenstein (Forschungsförderungsfonds der Universität Liechtenstein).

References

1. Baskerville, R.P.H.: Explanatory design theory. *J. Bus. Inf. Syst. Eng.* **5**, 271–282 (2010)
2. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research. *Manag. Inf. Syst. Q.* **35**, 37–56 (2011)
3. Gregor, S., Jones, D.: The anatomy of a design theory. *J. Assoc. Inf. Syst.* **8**, 313–335 (2007)
4. Katz, R., Allen, T.J.: Investigating the Not Invented Here (NIH) syndrome: a look at the performance, tenure, and communication patterns of 50 R&D project groups. *R&D Manag.* **12**, 7–20 (1982)
5. Carroll, J.M.: *Scenario-based Design: Envisioning Work and Technology in System Development*. Wiley, New York (1995)
6. Gamma, E.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, India (1995)
7. Pearl, J.: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Boston (1984)
8. Purao, S., Storey, V.C., Han, T.: Improving analysis pattern reuse in conceptual design: augmenting automated processes with supervised learning. *Inf. Syst. Res.* **14**, 269–290 (2003)
9. Guindon, R., Krasner, H., Curtis, B.: Breakdowns and processes during the early activities of software design by professionals. In: Gary, M.O., Sylvia, S., Elliot, S. (eds.) *Empirical Studies of Programmers: Second Workshop*, pp. 65–82. Ablex Publishing Corp. (1987)
10. Gero, J.S., Kannengiesser, U., Pourmohamadi, M.: Commonalities across designing: Empirical results. In: *Design Computing and Cognition 2012*, pp. 265–281. Springer, Netherlands (2014)
11. Tang, H., Lee, Y., Gero, J.S.: Comparing collaborative co-located and distributed design processes in digital and traditional sketching environments: a protocol study using the function–behaviour–structure coding scheme. *Des. Stud.* **32**, 1–29 (2011)
12. March, S.T., Smith, G.F.: Design and natural science research on information technology. *Decis. Support Syst.* **15**, 251–266 (1995)
13. Riemer, K., Seidel, S.: Design and design research as contextual practice. *Inf. Syst. e-Bus. Manag.* **11**, 1–4 (2013)
14. Sanders, E.B.-N., Stappers, P.J.: Co-creation and the new landscapes of design. *Co-design* **4**, 5–18 (2008)

15. Cross, N., Christiaans, H., Dorst, K.: Design expertise amongst student designers. *J. Art Des. Educ.* **13**, 39–56 (1994)
16. Gero, J.S., Gero, J.S., Mc Neill, T.: An approach to the analysis of design protocols. *Des. Stud.* **19**, 21–61 (1998)
17. Rittel, H., Webber, M.M.: 2.3 planning problems are wicked. *Polity* **4**, 155–169 (1973)
18. Latour, B.: A cautious prometheus? A few steps toward a philosophy of design (with special attention to Sloterdijk, P.). In: *Proceedings of the 2008 Annual International Conference of the Design History Society*, pp. 2–10 (2008)
19. Denning, P., Dargan, P.: Action-centered design. In: *Bringing Design to Software*, pp. 105–119 (1996)
20. Borchers, J.O.: A pattern approach to interaction design. *AI & Soc.* **15**, 359–376 (2001)
21. Van Aken, J.E.: *Improving the Relevance of Management Research by Developing Tested and Grounded Technological Rules*. Eindhoven Centre for Innovation Studies, Eindhoven (2001)
22. Fowler, M.: *Accounting Patterns. Analysis Patterns: Reusable Object Models*. Addison, Boston (2007)
23. Denning, P.J.: Great principles of computing. *Commun. ACM* **46**, 15–20 (2003)
24. Nonaka, I., Konno, N.: The concept of “ba”: building a foundation for knowledge creation. *Calif. Manag. Rev.* **40**, 40–54 (1998)
25. Grant, R.M.: Toward a knowledge-based theory of the firm. *Strateg. Manag. J.* **17**, 109–122 (1996)
26. Nonaka, I., Toyama, R.: The knowledge-creating theory revisited: knowledge creation as a synthesizing process. *Knowl. Manag. Res. Pract.* **1**, 2–10 (2003)
27. Polanyi, M.: The logic of tacit inference. *Philosophy* **41**, 1–18 (1966)
28. Vitalari, N.P., Dickson, G.W.: Problem solving for effective systems analysis: an experimental exploration. *Commun. ACM* **26**, 948–956 (1983)
29. Garud, R.: On the distinction between know-how, know-why, and know-what. *Adv. Strateg. Manag.* **14**, 81–101 (1997)
30. Cross, N.: Designerly ways of knowing: design discipline versus design science. *Des. Issues* **17**, 49–55 (2001)
31. Romme, A.G.L., Endenburg, G.: Construction principles and design rules in the case of circular design. *Organ. Sci.* **17**, 287–297 (2006)
32. Carlile, P.R., Eric, S.R.: Into the black box: the knowledge transformation cycle. *Manag. Sci.* **49**, 1180–1195 (2003)
33. Carlile, P.R.: Transferring, translating, and transforming: an integrative framework for managing knowledge across boundaries. *Organ. Sci.* **15**, 555–568 (2004)
34. Ericsson, K.A., Simon, H.A.: How to study thinking in everyday life: contrasting think-aloud protocols with descriptions and explanations of thinking. *Mind Cult. Act.* **5**, 178–186 (1998)
35. Ericsson, K.A., Crutcher, R.J.: Introspection and verbal reports on cognitive processes—two approaches to the study of thinking: a response to Howe. *New Ideas in Psychology* **9**, 57–71 (1991)
36. Suwa, M., Tversky, B.: What do architects and students perceive in their design sketches? A protocol analysis. *Des. Stud.* **18**, 385–403 (1997)
37. Suwa, M., Purcell, T., Gero, J.: Macroscopic analysis of design processes based on a scheme for coding designers’ cognitive actions. *Des. Stud.* **19**, 455–483 (1998)
38. Van Someren, M.W., Barnard, Y.F., Sandberg, J.A.: *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Academic Press, London (1994)
39. Carmelo, G.: *A practical approach to breast cancer knowledge management: a tumor board perspective (RTM)*. The University of Wisconsin (2010)

40. Strauss, A.L., Corbin, J.: *Basics of Qualitative Research. Techniques and Procedures for Developing Grounded Theory*. Sage, London, UK (1998)
41. Amabile, T.: *Creativity in Context*. Westview press, Boulder (1996)
42. Cross, N.: Expertise in design: an overview. *Des. Stud.* **25**, 427–441 (2004)
43. Puroo, S., Rossi, M., Bush, A.: Towards an understanding of the use of problem and design spaces during object-oriented system development. *Inf. Organ.* **12**, 249–281 (2002)
44. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.* **13**, 319–340 (1989)
45. 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)
46. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. *MIS Q.* **37**, 337–355 (2013)
47. Balaji, S., Murugaiyan, M.S.: Waterfall vs. V-model vs. agile: a comparative study on SDLC. *Int. J. Inf. Technol. Bus. Manag.* **2**, 26–30 (2012)
48. Puroo, S., Baldwin, C.Y., Hevner, A., Storey, V.C., Pries-Heje, J., Smith, B., Zhu, Y.: The sciences of design: observations on an emerging field. In: *Harvard Business School Finance Working Paper* (2008)
49. Baskerville, R.L., Myers, M.D.: Design ethnography in information systems. *Inf. Syst. J.* **25**, 23–46 (2015)