

Design Theory Projectability

Richard Baskerville, Jan Pries-Heje

► **To cite this version:**

Richard Baskerville, Jan Pries-Heje. Design Theory Projectability. 5th Working Conference on Information Systems and Organizations (ISO), Dec 2014, Auckland, New Zealand. pp.219-232, 10.1007/978-3-662-45708-5_14 . hal-01331828

HAL Id: hal-01331828

<https://hal.inria.fr/hal-01331828>

Submitted on 14 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Design Theory Projectability

Richard Baskerville¹ and Jan Pries-Heje²

¹ Georgia State University, Atlanta, USA
baskerville@acm.org

² Roskilde University, Roskilde, Denmark
janph@ruc.dk

Abstract. Technological knowledge has been characterized as having a scope that is specific to a particular problem. However, the information systems community is exploring forms of design science research that provide a promising avenue to technological knowledge with broader scope: design theories. Because design science research is materially prescriptive, it requires a different perspective in developing the breadth of applications of design theories. In this paper we propose different concepts that embody forms of general technological knowledge. The concept of *projectability*, developed originally as a means of distinguishing realized generalizations from unrealized generalizations, helps explain how design theories, being prescriptive, possess a different form of applicability. The concept of *entrenchment* describes the use of a theory in many projections. Together these concepts provide a means for comparative discussions of the importance of design theories. Projectable design theories guide designers in the design of artifacts similar in principle, but different in context. These can also help design researchers understand interrelationships between design theories.

Keywords: design science research · design theory · technological knowledge · generalizability · research methodology

1 Introduction

At a recent design science conference Pries-Heje and Pries-Heje [1] presented a 6-by-6 framework based on a design theory regarding physically distributed project teams. The design theory premised that such project teams would achieve improved cooperation when social capital is systematically built in six different ways during six phases of a team's life-cycle. An instance of this framework was evaluated in a banking project, and subsequently new instances are now being rolled out to 100 other projects. In the work below, we consider the question, "does such widespread replication of the theory across many different instances qualify the design theory as an *important* theory?" In what ways does it have consequences? As a prescriptive design theory, it is clearly different from the descriptive theories traditional in social science. Should we instead consider, "is it an important *design* theory?" In what ways does it have consequences for design?

There is a continuing interest in design science that are well anchored to seminal publications [e.g., 2, 3, 4, 5, etc.]. It has a presence in the top journals [6]. But its value as an academic enterprise is debatable [7, 8]. Is design science truly “science”? Is design science a proper activity for leading information systems scholars? Is design science “research”? If the answer to the first question is “yes”, then at least scientists are likely to answer the other two questions “yes” as well. But how do we compare the scholarly importance (or significance) of design science research studies? In this paper we consider one characteristic that can help distinguish an important or significant design science study from a trivial or insignificant study.

Generalizability is one measure of the intellectual usefulness of scholarly knowledge, even though it may take different forms [9]. It provides one indication of the importance of the particular findings from a research study. We usually assume that a generalizable study is important because it offers wider consequences in the future. The usefulness of design science knowledge has a more practical character and is often directed to a narrow context. If design science knowledge is to have wide consequences, there must be a means for assessing its value not merely descriptively in its own design context, but prescriptively for future contexts.

In this paper, we borrow the concept of projectability [10] from the philosophy of science and develop it as a design science alternative to generalizability. As with generalizability in descriptive research, projectability offers a frame for comparing the consequences of such prescriptive research methods as design science research. We develop this concept in the following way. First we distinguish projectability from generalizability. Next we describe the projection of technological knowledge (and design science results in particular). We then illustrate the use of these concepts in comparing the projectability of four design theories. After a brief discussion of the implications of the illustration, we conclude.

2 Generalizability versus Projectability

The scientific enterprise seeks “to discover and to formulate in general terms the conditions under which events of various sorts occur, the statements of such determining conditions being the explanations of the corresponding happenings.” [11, p. 4] The notion of “general terms” is important. Like an experiment, a design can be highly localized and particularistic. But the scientific public is not interested in a particular, past, local design. They want to learn about larger policies and interesting theoretical constructs. They want to connect the results of design science to broad conceptual applicability, requiring “generalization at the linguistic level of the constructs” rather than their operationalization in a particular design [12, p. 18].

Simply borrowing the concept of generalizability from other sciences may be problematic in design science. The phenomena in the natural sciences “have an air of ‘necessity’ about them in their subservience to natural law. The phenomena in design science “have an air of ‘contingency’ in their malleability by their environment.” The genuine problem in design science is to show how general empirical propositions can be made about designs that, “given different circumstances, might be quite other than

they are.” [13, p. xi] This problem does not just inhabit the science aspect of the enterprise, for the wider practical value of design studies lies in their consideration for applicability beyond a single environmental example [14]. For design science to be truly science, research, and a proper activity for leading information systems scholars, it should ideally produce generalizable knowledge.

There are different ways in which researchers may choose to generalize their findings from the study of one phenomenon to explain other, perhaps similar, phenomena [9]. For example, studies that focus on a phenomenon in a sample of instances where that sample has been randomly selected from the population of such instances; such studies will often adopt a statistical frame of generalizability that will project an expectation that characteristics found in the sample will also inhabit the population. For these studies, the characteristics of the population are subject to prediction or controls. The *context* of generalization is subject to prediction or control.

Context, at a fundamental level, might be either theoretical or empirical. Different contexts can shape different forms of generalizability. This diversity leads to different and sometimes conflicting definitions. In information systems, one analysis of the different ways to achieve generalizability in scientific studies involves distinguishing between theoretical and empirical statements [9]. This analysis developed four types of generalizability: generalizing from empirical statements to theoretical statements (Type ET), generalizing from empirical statements to empirical statements (Type EE), generalizing from theoretical statements to empirical statements (Type TE), generalizing from theoretical statements to theoretical statements (Type TT).

But because design science is not the same kind of science as, for example natural science, we should admit the possibility that generalizability of design sciences might be altogether different in nature than previous forms of generalizability. For example, in design science the future context for using design knowledge can be unpredictable and beyond control because it may not yet exist. Design science is materially prescriptive in the sense that its theories prescribe as-yet unconstructed artifacts. In design science research, design theories provide theoretical explanations that are usually functional rather than deductive [13, 15] and theoretical statements in design science tend to be prescriptive rather than descriptive [2, 16]. Generalizability in design science tends to be of a different nature, prescriptive rather than descriptive.

Transferability is a conceptual alternative to generalizability that is sometimes associated with forms of naturalistic inquiry such as action research [17, 18]. But a requirement for transferability is a deep knowledge of both the “sending” and “receiving” contexts in order to determine adequate congruence [18, p. 124]. In contrast, design science assumes that its theories can be applied in designing as-yet immaterialized future artifacts in as-yet unknown contexts.

Existing notions of generalizability and transferability are less suitable for the highly contingent notions of future applicability inherent in the materially prescriptive nature of design science. For example, an underlying principle of generalizability in social or natural science is the principle of abstraction. Abstraction involves a logical determination of the “universal” on the grounds of the “particular” [19, p. 158]. Abstraction is “a putative psychological process of the acquisition of a concept x either by attending to the features common to all and only x 's or by disregarding just the spatio-temporal

locations of x's" [20, p. 3]. Lockean abstraction is a process of discovering the idea of general types of objects from observations about individual material objects, such that a "type" is equivalent to a "generalization" so that a word is, in itself, a generalization [19, p. 403]. Abstraction is a process that separates a particular existence or instantiation from the idea, word, or general name for it and, at the same time, separates this idea from other ideas. We exclude some parts of a particular while retaining some other parts, yielding a general idea [21]. While abstraction operates well in descriptive science, it is less suitable for use in prescriptive science. Loss of the particular existence involves loss of the context of an instance.

While abstraction as a basis for generalizability might be useful in descriptive theory, generalizability for prescriptive theory must take a quite different form in order to promote the integrity of design science as a future source of general technological knowledge. Rather than adopt or adapt the notion of generalizability from the descriptive sciences, design science calls for a prescriptive form of general technological knowledge that can operate across different (perhaps presently non-existent) contexts.

Goodman's [10] concept of *projectability* provides such an alternative. Goodman originally developed this concept as a means of distinguishing realized generalizations from unrealized generalizations. It is useful in design science because it helps explain how design theories, being prescriptive, possess a different form of applicability.

Rather than attempting to focus on the history of regularities or consistencies that currently exist, or predetermination of the relevant abstract characteristics, *projection* involves determining possible regularities or consistencies that could be created in the future. A *projection* is any relevant instance that supports a theory. A theory is *actually projected* when some (but not all) of its possible instances have been examined. A theory is *projectable* if it is capable of being projected, has no known violations (observations that oppose the theory), and not all possible instances have been examined. When the terms of the theory have been used in many projections, it is said to be *entrenched* [10, pp. 80-81].

Goodman's concept fits particularly well for conceptualizing the applicability of design theories. This fit is because empirical design studies will *actually project* a design theory by instantiating it, and future instantiations will serve to *entrench* the design theory.

Thus this focal shift turns our attention away from the act of confirming or falsifying a descriptive theory, and toward the act of entrenching a prescriptive theory using future instantiations (actual projections). For design theories, determining their prescriptive projectability may be more relevant than determining their descriptive generalizability. This determination regards the consequence of the theory for further designs. The status of what we know about a design theory's projectability provides one criterion for the importance of the theory. We may know that a projectable theory is important, but we would know that one that has been projected is more important. Evaluation of an instantiated artifact is critical because it means the design theory is actually projected. Similarly, a design theory that has become entrenched as a result of a wider scale of projection would be known to be more important still. Its consequences are broader.

The projectability of a design theory depends on the way in which the terms of its statements are anchored to actual or possible projections. The most limited design theories may not be projectable beyond the actual projection empirically demonstrated in the originating research. More typically, design theories can be projected to instances defined by a class of possible projections available (but not yet actualized). In some cases, the title of the design theory suggests its projectability. For example, in “Building an information system design theory for vigilant EIS”, the projectability extends (at least) to future instantiations of vigilant executive information systems [5]. Another example is “A theory of decision support system design for user calibration” where the projectability allows for future instantiations of decision support systems [22]. A third example is “A design theory for systems that support emergent knowledge processes”, where the projectability is future instantiations of knowledge management systems [23].

For design science, projectability on a very broad scale implies a design theory that could possibly be projected to an infinite number of future design problems; we characterize such a design theory as a *projectable design theory*.

Returning to the example introduced at the beginning of this paper [1], we propose an answer to the “important theory” question in at least one way. Since the design theory has actually been *actually projected* across many instances of projects in an organization it has proved important in terms of its *projectability* within this organization. We propose that there is a further process to make *projections* in other organizations leading to a process to *entrench* the theory. Hence it is clearly a *projectable design theory*.

3 Projecting Design Theory as Technological Knowledge

We can expect that the nature of a projectable design theory would be different from the nature of a generalizable theory in natural or social science. This is because one way in which design science differs from social or natural science is its stronger dependence on functional explanations. This kind of explanation is grounded on the relationship between functional requirements and the prescriptive components of the design. While authorities in the philosophy of science might disagree about whether functional explanations should be regarded as scientific or non-scientific [cf. 11, 24], functional explanations form the core of theories in design science [13, 15].

Because design theories engage functional explanations, a projectable theory in design science may also be able to encompass a family of other, perhaps more material, design theories that involve values, decisions, games, operations research, etc. These design theories are all operative or technological theories that other sciences (like natural or social science) might regard as dependent on non-scientific or ordinary knowledge:

In a conceptual sense, the theories of technology are definitely poorer than those of pure science: they are invariably less deep, and because the practical man, to whom they are devoted, is chiefly interested in net effects that occur and are controllable on the human scale: he

wants to know how things within his reach can be made to work for him, rather than how things of any kind really are. [24, p.123]

Such representations of technological theories as inferior no longer stand well against critical scrutiny. For example, one acknowledged form of technological or design theory is the *technological rule* [25, 26]. Rules prescribe a form of practical action. One important form of rule, *rules of science and technology*, are the norms for scientific research techniques as well as other advanced production techniques [25, p. 132]. But scientific knowledge itself is not distinguished from ordinary knowledge by its rationality, objectiveness, nor its regard for substance. Ordinary knowledge can also be rational, objective, and substantial. What distinguishes scientific knowledge is its scientific approach: the scientific method and goal [24, p. 6]. Because this scientific method is encoded by technological rules, such a fundamentally distinctive character of science cannot be regarded as “poorer” or “less deep”.

4 Comparing Projectability of Design Theories

In this section, we will compare the importance of four design theory examples. The first two examples are declared design theories in the sense that the underlying research studies explicitly proposed these theories as *design theories*. These illustrate design theories that are *projectable* and *actually projected*. But these examples are too recent to illustrate the concept of a design theory that has become *entrenched* by numerous actual projections. It may be the case that design science is itself too new in information systems to permit such examples. So we will use two further examples of theories that were not originally proposed as *design theories* but can be easily recast as such. This *post hoc* reconceptualization of these as design theories provides an indication of how we might recognize design theories that are projectable, actually projected, and entrenched.

4.1 Vigilant Executive Information Systems

While often cited as the seminal source of design theory itself, Walls et al. [5] also illustrated their proposal with an example of a design theory for a Vigilant Executive Information System (VEIS). The example may never have been *actually projected*, but it held promise for being *projectable* to the range of VEIS as proposed in the descriptions. It was logically evaluated in the research, but not reported as instantiated for evaluation. [The works predated the notions of build and evaluate that followed, 3, 4]

This design theory suggested that the environment of many organizations had become “turbulent” and proposed “vigilance” as newly required functionality in order for executive information systems (EIS) to remain appropriate. The theory prescribed changes in the EIS interface to incorporate new inputs such as templates, triggers, and twitches. A template is the frame of reference with which an executive perceives an issue domain. A trigger is a stimulus causing a template to shift. A twitch is a short movement with a sudden motion leading to a template modification. The theory prescribed changes in the EIS functionality to incorporate such features as rapid response

through open loop control. Open loop control is faster because it does not necessarily depend on a control feedback loop.

While the Walls et al. paper had a tremendous consequence through its proposal for a concept of design theory, the VEIS example seems less important. The authors proposed conservatively that it was projectable so far as adding vigilance to information systems. This VEIS design theory was narrowly projectable, not actually projected, and not entrenched. In terms of its actual consequences, we might conclude that our knowledge so far suggests that the VEIS design theory was not very important.

But as a prescriptive theory, this conclusion could change. It is always possible that the projectability is better than described. For many organizations, the turbulence has only increased. The information available from this environment has also increased with data arising more online activity. There is new science emerging aimed at discovering useful knowledge for decision making in organizations [27]. It is at least feasible that the functions and interface involved in the VEIS design theory may be projectable into this new data science. Its projectability may have increased, and its importance may rise as a result. But we must wait until there is knowledge from actual projections before the consequences might be descriptively known. Actual projections and entrenchment conceptualize the history of projections. The concept of projectability is itself much broader, and encompasses both history and future projections in undetermined new contexts.

4.2 Theory Nexus

A design theory nexus is an artifact that improves the search for design solutions among contrasting alternatives using the principle that these are based on distinct kernel theories [the underlying psychosocial theories, natural theories, computing theories, etc., cf. 5]. Carroll and Kellogg [28] used the term *nexus* to describe the interactive nature of such kernel theories when used to design artifacts. Using examples from human computer interface design, they explain how the use of multiple psychological theories specify in too much detail the designed artifacts, and that the exact effects of an artifact in relation to the theories underlying its design can only be realized by experience with the artifact; or in Goodman's terms, its *actual projection*. The empirical design will actually project the design theory by formalizing it or instantiating it and thereby understanding exactly how the theories determine the artifacts.

A theory nexus emerges when multiple kernel theories drive the design of an artifact. As the artifact becomes formalized, it articulates these theories in such a way that any conflicts that inhabit the theories make the formalization or instantiation of the design problematic. In such settings, designs emerge from an iterative process in which design theories are re-articulated in the presence of the artifact and competing design theories.

Pries-Heje and Baskerville [29] used the nexus concept as a basis for a *design theory nexus* as an approach to designing artifacts that help solve wicked problems. A wicked problem is an incomplete and contradictory problem that changes over time and for which no classic linear decision model can be found. Many social, commercial or financial planning problems will be wicked "because they won't keep still" [30]. Such problems are often refined, rather than solved, by alternative solutions. We can work

towards a solution if not solve it. Wicked problems have some of the following five characteristics [31]: (1) There is no definitive formulation of the problem. We need to understand the problem (better) through working with the solution. (2) There is no stop signal embedded in the problem. This is because the process by which we solve the wicked problem is identical to the process by which we understand the problem. (3) There are no true or false solutions but only solutions of varied goodness. (4) Any solution to a wicked problem is a unique one-time solution. (5) You cannot list – or number – a finite set of solutions.

The design theory nexus pulls the wicked problem itself into the theory nexus. Not only are the kernel theories and the design artifact re-articulated in this nexus, but the problem itself is rearticulated in the process.

The study reports two instantiations of the design theory nexus: One for a wicked problem (organizational change) and one for the problem of user involvement. The two instantiations were intended to demonstrate how the design theory nexus operated with wicked kinds of problems and more normal kinds of problems. In this case, the study is claiming that the nexus design theory is projectable across a very wide range of problem solving settings. It has been actually projected into two instances. The projectability suggests that this theory has potential to be very important. However, the actual projections provide only the limited knowledge about its consequences in two instantiations. While we have more knowledge about this design theory's projectability than we do about the VEIS design theory, neither of these two theories is known to be entrenched.

4.3 Pattern Design Theory

The examples of the explicit design theories above help us differentiate between design theory projectability, actual projection, and entrenchment, and to evaluate the value of such knowledge in determining the importance of the design theory. But neither design theory above was entrenched. In order to gain a sense of how an entrenched design theory might appear, we will consider the concept of a pattern as a design theory. Aside from its explicit use as a design theory [15], Vaishnavi and Kuechler [32] inspire such recognition by applying pattern theory as a framework for design science.

The influential book *Notes on The Synthesis of Form* opens with, "These notes are about the process of design: the process of inventing things which display new physical order, organization, form, in response to function" [33]. This responsive relationship between inventions and function is a *pattern*. For example, Alexander went on to show how to express pattern theory using a pattern language where he *projected* exact methods for constructing practical, safe and attractive designs at every scale, from entire regions, through cities, gardens, buildings, and down to the doorknob of a door in the building [34]. Today, we might characterize Alexander's notions about a meta-language for construction as a design theory, and recognize it as an obvious instantiation of general pattern design theory.

This theory of patterns – providing rules and pictures - is a projectable design theory because it explains why and how certain kinds of subject design theories are generated. For example, pattern theory justifies the use of patterns in the IT development arena

such as the use of reusable solutions by object-oriented software development when encountering a commonly recurring problem [35]. We can even characterize this notion of reusable solutions as a design theory in its own right: A prescriptive framework consisting of *Problems and Constraints* followed by *Solution and Pattern*. Gamma et al. [35] project this fundamental patterns design theory to 23 areas of object oriented development. Two examples of such object patterns include a *Singleton* and a *Proxy*. A Singleton pattern is one in which object creation is restricted. The object class is allowed to have only one instance. In personal records, for example, there will be only one surname, but multiple forenames. The surname object would be a Singleton pattern. A Proxy pattern that provides a placeholder for another object thereby controlling access and allowing substitution of parts of the system as long as they interface in the same way with the placeholder. In shopping systems, a verify credit proxy object might always respond “unverified credit” for privacy purposes until the user is logged in; after which a completely functional verify credit object is engaged instead. From this perspective, a pattern design theory has been actually projected in object oriented design. Object oriented design has subsequently been projected further to innumerable instances of software designs.

Pattern theory inhabits methods in design science research as problem solving patterns for different phases of design. In this sense, pattern theory can support a form of subject design theory for DSR, that will provide “... a methodology for the practice of DSR that is keyed to the patterns” [32, p. 3]. In this work, there are problem solving patterns for different phases of design. For example this work details patterns for the “build” and “evaluate” phases of DSR, and provide meta-patterns (such as brainstorming) that can be used across multiple DSR phases. This study actually projected the pattern design theory into a design science methodology, which in turn has been further projected to instances of design science studies.

In this case, such a supposed pattern design theory has vast projectability and countless actual projections. In the two examples above, the projections themselves have further projections. Such a pattern design theory is solidly entrenched by the variety and enormity of the actual projections. A well-entrenched design theory appears likely to have projectability not only into instances of artifacts, but into other subsequent design theories. An entrenched design theory may have “children” as it were; regenerations of itself as a design theory in newer forms with more current applications.

4.4 Contingency Design Theory

As a fourth example of how a massively projected design theory might become known to be important, we will consider how contingency theory might appear as an entrenched design theory. The core idea in contingency theory is that organizations that want to optimize performance need to adopt the structure that fits best with the situation they are in - the contingencies given them. “At the most abstract level, the contingency approach says that the effect of one variable on another depends on some third variable ...” [36, p. 5]. Contingency theory arises in a broad array of studies. For example, it has been *projected* into vague and ambiguous situations, where the information available could be interpreted in many ways depending on perspective [37]. Contingency

factors have included leadership style [38, 39], formalization and centralization of organizations depending on the uncertainty of the environment [40], communication support, process structuring and information processing [41, 42], as well as task complexity and whether the technology is appropriate for the task [43].

Contingency theory in management is sometimes formalized as technological rules, expressing a decision design as, “A technological rule follows the logic of ‘if you want to achieve Y in situation Z, then perform action X’. The core of the rule is this X, a general solution concept for a type of field problem.” [44, p. 23]. The “Z” in such technological rules embodies the contingencies. Technological rules need grounding: “Research that intends to ground a technological proposition to explain why and how it produces certain outcomes will typically have to draw on survey-based field studies” [45, p. 9]. Hence, grounded technological rules need at least one *actual projection* (the field study).

Our supposed contingency design theory is projected into the concept of technological rules, which in turn can be used as a design theory for designed organizational decision heuristics. This regeneration is the projection of the contingency design theory into technological rules as a design theory. The highly projectable design theory is the overall contingency theory, and it is projected as Van Aken does it – If Y in Z do X – or as Donaldson puts it – effect of 1 or 2 depends on 3. In both cases the underlying level embodies the contingencies in situ and the theory applied for a specific instance. An example of the latter is the expression of different sets of technological rules for different management areas.

Like pattern design theory, such a contingency design theory has vast projectability and is deeply entrenched by countless actual projections. It has regenerated with its actual projections into other design theories. As a result, we have considerable knowledge about the importance of such a theory.

5 Discussion

The examples in the previous section included declared design theories and *post hoc* reconceptualized design theories. The declared design theories were declared and presented as design theories in research. The *post hoc* theories were constructed as an argument to illustrate how we might know about important design theories that are not only projectable, but have been actually projected and entrenched.

The examples have also shown how design theories are not only projectable into material artifacts as design instances, but also into future design theories. It is also possible to project a design theory backward in time in order to gain better knowledge about why a past design or a design theory succeeded. An example in this discussion was our projection of the pattern design theory into object oriented design methodology. This specific projection is interesting because a design theory is normally thought to be prescriptive, but it also has a descriptive component that can be used for better understanding past design successes and failures. This descriptive component involves the functional explanations within design theory that justify the various components in

the design [15]. In the Design Theory Nexus, the theory rearticulation structure is functionally explained by the particular nature of wicked problems.

Additional insight into the projectability of a design theory can arise because actual projections should not be logically reversible. In the nexus example, we considered a post-hoc projection of the design theory nexus that was logically consistent in recasting all psychologically overspecified designs as wicked problems. The projection cannot be reversed because all wicked problems are not psychologically overspecified designs. This one-way nature of actual projects between design theories helps us determine which design theory in the pair is more projectable (and therefore more important).

Similarly, Goodman's notion of projectability helps identify interesting situations where design theories are projectable into instances (actual projections) of another. For example, one interpretation holds that the more projectable contingency design theory inhabits both the vigilant executive information system (VEIS) design theory [5] and the user calibrated decision support system (UCDSS) design theory [22]. In VEIS design theory, the executive process is contingent on triggers from the environment. In UCDSS design theory, the locus of symbolic representation is contingent on problem novelty. Contingency design theory is so projectable that it can be easily projected to instances of UCDSSs instantiated with the design theory for UCDSS as well as to instances of VEISs instantiated with the design theory for VEIS. But the design theory for UCDSS cannot easily be projected into a VEIS design, or vice-versa. Contingency as a design theory is entrenched and more projectable than both UCDSS or VEIS.

It is likely that there are many cases where multiple cases of entrenched design theories actually projected into a less projectable design theory. For example, we might use patterns for contingency, or build methods that rearticulate technological rules in a nexus. The VEIS design theory uses contingency triggers and patterns coded as executive process templates for determining managerial processes. Entrenched design theories are not necessarily mutually exclusive when actually projected into another design theory.

We recognize the two forms of projectability of design theory. In form one, the design theory may be actually projected into instances of some operational artifact. For instance, in our opening example, a six-phase social capital project team design theory has been actually projected into operating instances of more than 100 project teams. In form two, the design theory may be actually projected into another design theory. For instance, we earlier proposed actually projecting pattern design theory into the VEIS design theory. The distinction between forms one and two has parallels in the kinds of generalizability in descriptive science. Actually projecting a design theory to operating instances is parallel to generalizing from theoretical statements to empirical statements [described as type TE generalizability in 9] while actually projecting a design theory to another design theory is parallel to generalizing from theoretical statements to empirical statements [described as type TT generalizability in 9]. However, the materialization of design theories as prescriptive, functional projections makes design science projections rather different in operation than descriptive science generalizations.

Like generalizability, it is unlikely we will ever develop absolute or relative scales of projectability. It cannot be meaningful to decide that VEIS has scale-3 projectability while UCDSS has scale-4 projectability. The prescriptive nature of design science

means that projectability assessments are subject to massive revision in cases where a design theory proves more useful in the future than now. However, Goodman's concept of projectability provides a language for discussing the character of a design theory as general technological knowledge, and to compare those characteristics among quite different design science studies. It provides the means to begin discussing in design science terms the ways in which one design science study could be seen as more important than another.

For example, in the introduction, we asked about the importance of a six-phase social capital project team design theory. We can say that it has been actually projected into operating instances of more than 100 project teams. We can assess that the theory is certainly projectable because it has been actually projected many times. While it has not been actually projected beyond one organization, it is clearly projectable to other contexts. It may even be said to be entrenched (at least in one organization) given the large number of actual projections. We might also characterize it as more entrenched than either VEIS or the Theory Nexus, but not so vastly entrenched as our contingency or pattern design theories.

While using the concept of projectability in design science has many rational justifications discussed earlier, it is also useful for helping to distinguish design science from other forms of science. Projectability is rooted only in the philosophy of science, and is perhaps less confused with the statistical concept of the generalizability of a sample to a population. The class of possible projections cannot be equated to a population, because the class is, at least partly, non-existent. A design theory is not projected to a population. The confusion over the applicability of statistical generalizability concepts in non-quantitative research settings has fueled debate in the field of information systems [46, 47].

6 Conclusion

The distinct and often material way in which design science makes prescriptive design theory projections illuminates two distinctions between our consideration of general technological knowledge and other forms of knowledge. On the one hand, we acknowledge how technological knowledge is different from knowledge in the natural and social sciences. On the other hand, the ability to develop broadly projectable design theories means that design science can develop more general technological knowledge. This change in the development of technological knowledge production is rather fundamental, perhaps even revolutionary. This innovation is actually taking place in the information systems community; occurring so gradually that this important achievement may go unnoticed by the field itself.

The notion of projectability provides a means to assess and discuss the comparative importance of different design science research studies. It is different from generalizability because of its prescriptive and more context-independent characteristics. It enables us to distinguish design theories that feature characteristics of projectability, actual projections, and entrenchment. Actual projections and entrenchment are historical characteristics of the design theory's past projections. Importantly for a prescriptive

science, projectability is itself the conceptualization of the consequence of a design theory in unplanned, uncontrolled, future contexts.

References

1. Pries-Heje, J., Pries-Heje, L.: Designing a Framework for Virtual Management and Team Building. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) *Design Science Research in Information Systems: Advances in Theory and Practice*. LNCS, vol. 7286, pp. 256-270. Springer, Heidelberg (2012)
2. Gregor, S., Jones, D.: The Anatomy of a Design Theory. *Journal of the Association for Information Systems* 8(5), 312-335 (2007)
3. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. *MIS Quarterly* 28(1), 75-105 (2004)
4. March, S.T., Smith, G.: Design and Natural Science Research on Information Technology. *Decision Support Systems* 15(4), 251-266 (1995)
5. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Building an Information System Design Theory for Vigilant EIS. *Information Systems Research* 3(1), 36-59 (1992)
6. Baskerville, R., Lyytinen, K., Sambamurthy, V., Straub, D.: A Response to the Design-Oriented Information Systems Research Memorandum. *European Journal Information Systems* 20(1), 11-15 (2011)
7. Junglas, I., Niehaves, B., Spiekermann, S., Stahl, B.C., Weitzel, T., Winter, R., Baskerville, R.: The Inflation of Academic Intellectual Capital: The Case for Design Science Research in Europe. *European Journal Information Systems* 20(1), 1-6 (2011)
8. Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., Sinz, E.J.: Memorandum on Design-Oriented Information Systems Research. *European Journal Information Systems* 20(1), 7-10 (2011)
9. Lee, A.S., Baskerville, R.L. Generalizing Generalizability In Information Systems Research. *Information Systems Research* 14(3), 221-243 (2003)
10. Goodman, N.: *Fact, Fiction, & Forecast*. Harvard University Press, Cambridge, MA (1955)
11. Nagel, E.: *The Structure of Science: Problems in Scientific Explanation*. Routledge and Kegan Paul, London (1961)
12. Shadish, W.R., Cook, T.D., Campbell, D.T.: *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin, Boston (2002)
13. Simon, H.A.: *The Science of the Artificial*, 3rd edn. MIT Press, Cambridge, MA (1996)
14. Williams, R., Pollock, N.: Moving Beyond the Single Site Implementation Study: How (and Why) We Should Study the Biography of Packaged Enterprise Solutions. *Information Systems Research* 23(1), 1-22 (2012)
15. Baskerville, R., Pries-Heje, J.: Explanatory Design Theory. *Business & Information Systems Engineering* 2(5), 271-282 (2010)
16. Walls, J.G., Widmeyer, G.R., El Sawy, O.A.: Assessing Information System Design Theory in Perspective: How Useful Was Our 1992 Initial Rendition? *Journal of Information Technology Theory and Application* 6(2), 43-58 (2004)
17. Guba, E.G., Lincoln, Y.S.: Epistemological and Methodological Bases of Naturalistic Inquiry. *Educational Communications and Technology Journal* 30(4), 233-252 (1982)
18. Lincoln, Y.S., Guba, E.G.: *Naturalistic Inquiry*. Sage, Newbury Park, CA (1985)
19. Walmsley, J., The Development of Lockean Abstraction. *British Journal for the History of Philosophy* 8(3), 395-418 (2000)

20. Priest, S.: Abstraction. In: Honderich, T. (ed.) *The Oxford Companion to Philosophy*, p. 3. Oxford University Press, Oxford (2005)
21. Walmsley, J.: Locke on Abstraction: A Response to M.R. Ayers. *British Journal for the History of Philosophy* 7(1), 123 (1999)
22. Kasper, G.M.: A Theory of Decision Support System Design for User Calibration. *Information Systems Research* 7(2), 215-232 (1996)
23. Markus, M.L., Majchrzak, A., Gasser, A.: A Design Theory for Systems That Support Emergent Knowledge Processes. *MIS Quarterly* 26(3), 179-212 (2002)
24. Bunge, M.: *Scientific Research I: The Search for System*. Springer-Verlag, New York (1967)
25. Bunge, M.: *Scientific Research II: The Search for Truth*. Springer-Verlag, New York (1967)
26. van Aken, J.E.: Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. *Journal of Management Studies* 41(2), 219-246 (2004)
27. Davenport, T.H., Barth, P., Bean, R.: How Big Data Is Different. *MIT Sloan Management Review* 54(1), 43-46 (2012)
28. Argyris, C.: The Discipline of Management and Academic Defensive Routines. In: Mansfield, R. (ed.) *Frontiers of Management*, pp. 8-21. Routledge, London (1989)
29. Pries-Heje, J., Baskerville, R.: The Design Theory Nexus. *MIS Quarterly* 32(4), 731-755 (2008)
30. Ritchey, T.: *Wicked Problems: Structuring Social Messes with Morphological Analysis* (2011) <http://www.swemorph.com/pdf/wp.pdf>
31. Rittel, H., Webber, M.W.: Dilemmas in a General Theory of Planning. *Policy Sciences* 4, 155-169 (1973)
32. Vaishnavi, V.K., Kuechler, W.: *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*. Auerbach Publications, Boca Raton, FL (2008)
33. Alexander, C.: *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, MA (1964)
34. Alexander, C., Ishikawa, S., Silverstein, M.: *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, NY (1977)
35. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA (1995)
36. Donaldson, L.: *The Contingency Theory of Organizations*. Foundations for Organizational Science. 2001, Thousand Oaks, CA: Sage Publications.
37. Galegher, J., Kraut, R.E.: Computer-Mediated Communication for Intellectual Teamwork: An Experiment in Group Writing. *Information Systems Research* 5(2), 110 (1994)
38. Hersey, P., Blanchard, K.H.: Life Cycle Theory Of Leadership. *Training and Development Journal* 23(5), 26-34 (1969)
39. Hersey, P., Blanchard, K.H.: So You Want to Know Your Leadership Style? *Training and Development Journal* 35(6), 34-54 (1981)
40. Lawrence, P.R., Lorsch, J.W.: *Organization and Environment: Managing Differentiation and Integration*. Harvard University, Graduate School of Business Administration, Division of Research, Boston (1967)
41. Zigurs, I., Buckland, B.: A Theory of Task/Technology Fit and Group Support System Effectiveness. *MIS Quarterly* 22(3), 313-334 (1998)
42. Zigurs, I., Buckland, B.K., Connolly, J.R., Wilson, E.V.: A Test of Task/Technology Fit Theory for Group Support Systems. *Database for Advances in Information Systems* 30(3/4), 34-50 (1999)

43. Van de Ven, A.H., Drazin, R.: The Concept of Fit in Contingency Theory. *Research in Organizational Behaviour* 7, 333-365 (1985)
44. van Aken, J.E.: Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. *British Journal of Management* 16(1), 19-36 (2005)
45. van Aken, J.E., Romme, G.: Reinventing the Future: Adding Design Science to the Repertoire of Organization and Management Studies. *Organization Management Journal* 6, 5-12 (2009)
46. Lee, A.S., Baskerville, R.L.: Conceptualizing Generalizability: New Contributions and a Reply. *MIS Quarterly* 36(3), 749-761 (2012)
47. Tsang, E.W.K., Williams, J.N.: Generalization and Hume's Problem of Induction: Misconceptions and Clarifications. *MIS Quarterly* 36(3), 729-748 (2012)