Management Design Theories

Jan Pries-Heje and Richard L. Baskerville

[1] Roskilde University,
4000 Roskilde, Denmark
[2] Georgia State University,
Atlanta, Georgia, USA

Abstract. This paper elaborates a design science approach for management planning anchored to the concept of a management design theory. Unlike the notions of design theories arising from information systems, management design theories can appear as a system of technological rules, much as a system of hypotheses or propositions can embody scientific theories. The paper illustrates this form of management design theories with three grounded cases. These grounded cases include a software process improvement study, a user involvement study, and an organizational change study. Collectively these studies demonstrate how design theories founded on technological rules can not only improve the design of information systems, but that these concepts have great practical value for improving the framing of strategic organizational design decisions about such systems. Each case is either grounded in an empirical sense, that is to say, actual practice, or it is grounded to practices described extensively in the practical literature. Such design theories will help managers more easily approach complex, strategic decisions.

Keywords: Design science research, management design, decision design, technological rules, design theory.

1 Introduction

This paper proposes a design science research approach to management planning. Design science research (March and Smith 1995) is a *generative* mode of research.

Generative research means that scientific discoveries proceed from the design and creation of artefacts, and from evaluation of such artefacts in use. Design scientists create knowledge by generating designs, generating artefacts from these designs, and studying these artefacts in practical usage. Design science operates with prescriptive rather than descriptive theories because the nature of designs is action oriented. This action orientation arises because designs show how we "do things." Design theories fundamentally relate a general class of design problems with a general class of design solutions.

Design science research has great potential value for management and information systems. It offers a possible improvement in the usefulness of research for management, which is regarded by some authorities as a problem: "academic management

research has a serious utilization problem" (van Aken 2004, p. 219). From this perspective, management research results are too descriptive and historical. For reflective studies for managers facing current problems, the direct usefulness of such histories is questionable. Post mortem analyses of last year's decisions are less relevant than help and advice for the issues managers face. If management research became less descriptive and more prescriptive, and less historical and more design-oriented, the utility of management research could be increased dramatically. Management research could lead to a new form of theory, a design theory consisting of "field-tested and grounded technological rules" (van Aken 2004, 2005b). This notion constitutes a design science research approach to management.

Designing involves developing prescriptive, not descriptive, knowledge. In van Aken's design rules, there are two possible outputs (artefacts or interventions) and three kinds of designs in a professional episode (object-design, realization-design, or process-design). An object-design defines the artefact or intervention. A realization-design defines a plan for implementing the object-design. A process-design defines how the design process itself is carried out.

Van Aken expresses a design in the form of technological rules. "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" (van Aken 2005a, p. 23).

Technological rules must be grounded. "Without grounding, the use of technological rules degenerates to mere 'instrumentalism,' i.e., to a working with theoretically ungrounded rules of thumb (Archer 1995, p. 153).

> In engineering and in medicine, grounding of technological rules can be done with the laws of nature and other insights from the natural and the life sciences (as well as from insights developed by these design sciences themselves). In management, grounding can be done with insights from the social sciences (van Aken 2005a, p. 25).

Just identifying technological rules per se is insufficient, regardless of how helpful they may be to managers. The rules must be properly grounded from a social science perspective.

In discussing technological rules, Pawson and Tilley (1997) raised the issue of causality. Which of the generative mechanism(s) that are used in an intervention actually produces the outcome in a given context? This question leads to the formulation of the CIMO-logic that can be formulated in the following way: "In this class of problematic Contexts, use this Intervention type to invoke these generative Mechanism (s), to deliver these Outcome(s)" (Denyer et al. 2008, p. 395).

Besides detailing the formulation of technological rules by virtue of the CIMO-logic, Denyer et al. (2008) suggest the term *design proposition* instead of *technological rule* arguing that "the latter term suggests—contrary to our intentions—a rather mechanistic, precise instruction."

The empirical cases we are reporting below used the technological rules rather than the CIMO-logic. While perhaps less logically comprehensive, the technological rules were simpler and more accessible for our cases.

The remainder of the paper is organized as follows. We first discuss in general how design science could improve management. Then we discuss design science research

in management and demonstrate it through three grounded cases. Finally, we conclude that it is possible to help managers approach complex, strategic decisions by using the concept of technological rules.

2   Design Science in Management

Simon (1996) defines the science of design as the study of the artificial: "the way in which that adaptation of means to environments is brought about" (p. 113). Vaishnavi and Kuechler (2004) define design science research in information systems as "the analysis of the use and performance of designed artefacts to understand, explain and very frequently to improve on the behavior of aspects of Information Systems." The key concept is the design of an artefact that is meant to be somehow present in reality. While the artefact must be real, is could be a construct, a model, a method, or a material instantiation (March and Smith 1995).

Science and design are related in ways that are complex and contentious. Design is a generative production arising when the faculties of reason align in a way that is different from the analytical productions that are prized in science (see Kant 1908). Conflating science with design will frame the act of designing at a higher level of abstraction. At this level, designs are more universal and address a more general class of problems. We can contrast design science from design itself, which addresses a single, unique design problem. Generality demands theory, and in design science, a design theory is a special form of theory. Design theories share particular characteristics such as principles of form or function, principles of implementation, etc. (Gregor and Jones 2007; Walls et al. 1992).

Design science can be found in a variety of professional disciplines such as architecture, information systems, computer science, and engineering. In management, it arises mostly in decision science. As described earlier, van Aken (2004, 2005a, 2005b) argues that the utility of management research would increase if management research becomes prescriptive and design-oriented. Where van Aken proposed technological rules for designing decisions, these can also be used to express a design theory. This opens the possibility for a design science research approach to management.

We have chosen three cases to demonstrate that it is in fact possible to increase utility and help managers in a way that makes it possible and plausible to make better decisions in complex and/or strategic decision situations. The three cases we use are organizational change management, user involvement, and process improvement. For each case, we show how a new design theory appears using "technological rules" as suggested by van Aken. At the end of each case, we also discuss the grounding in social science as well as the (potential) application by managers and the practical implication(s).

3   Case 1: Process Improvement

Improving organizational processes and managing product and process quality is a particular area where advice for managers is urgently needed. The basic assumption made in this arena is that the quality of products and services is a direct result of the

quality of organizational processes. The principle is simple and proven: Improve the organization's processes, and the goods and services it produces will also improve. Here design theories could prove highly practical as well as academically novel. The "big three" approaches to process improvement are the general International Standards Organization (ISO) standards for quality management systems (ISO 9000), the more technical and product-development oriented Capability Maturity Model (CMM), and the more specific ISO standards for process improvement and capability determination (ISO 15504/SPICE) (Hunter and Thayer 2001).

There are also many variations, enhancements, and proprietary alternatives, as well as opportunities to adapt general quality management models directly for software quality improvement (e.g., Six Sigma and Juran). Like the more general quality improvement approaches, there is little work that provides helpful guidance about which of these approaches to choose for which kinds of software organizations.

Management not only involves administering process improvement efforts, but also deciding which of these myriad approaches should be used to frame the effort within the product development organization at hand. This approach selection process can be viewed as a design problem because it is concerned with the adaptation of available means to an environment (Simon 1996). However, it is a managerial design problem.

We analyzed seven normative models for software process improvement (can be understood as process improvement in software developing organizations): Balanced Scorecard, Bootstrap, Business Excellence, CMM, ISO 9000, Juran, and Six Sigma. We identified common elements that could be used to distinguish and characterize different type of models. The major common element identified we called *agenda*, and it was characteristic that agenda differed among the models. We defined the agenda as the perspective on outcomes (of improving the organization) that drives the entire process improvement effort. Second, normative process improvement models share knowledge generating activities similar to those of research methods viz., observation, analysis, and synthesis. These three are also a kind of common element, but they follow after (or pertain to) agenda (see Figure 1).

In Figure 1, we have shown agenda with the three subsequent common elements underlying the agenda. Together, the four elements define a process theory that delineates what kind of recommendations can be derived for a specific organization. It is, therefore, a working classification system that could be restated as a simple hypotheses. In this way, it is equivalent to the set of research questions (the system of hypotheses) that drive a research project.

Agendas are set according to the process improvement goal and the general management viewpoint on organizations. Goals vary. Some managers seek to achieve a balance in the organizational activities and resources for an optimum performance. Other managers seek to provide a direction for the organization, a path to a future, desirable state. Organizational viewpoints also vary. Some managers see software organizations as quite similar and believe a set of universal solutions can be applied in most organizations. Others see organizations as highly unique instances, intersections of very particular resources and people (see Figure 2).
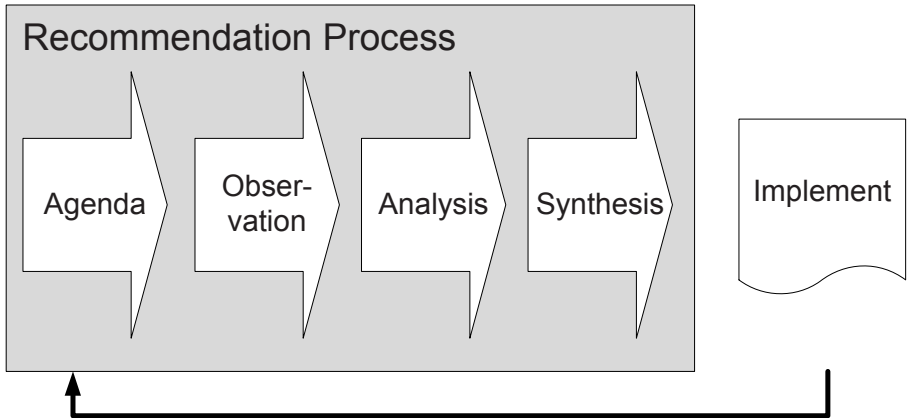
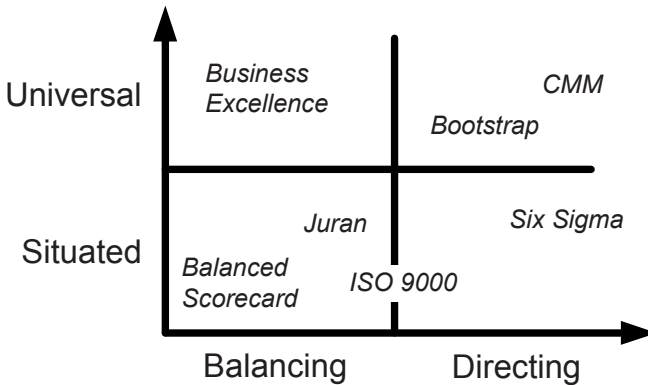**Fig. 1.** Overview of the General Improvement Recommendation Process

Fig. 2. Agenda Examples (from Pries-Heje and Baskerville 2003)

ISO 9000 has a very simple agenda: OK or not-OK. The organization is measured against a standard. The agenda in CMM is to improve processes by moving up to the next step on a five-step scale. The agenda in Bootstrap is partly the same as for CMM; the organization is measured against a number of processes. But Bootstrap focuses more on business goals. The Six Sigma agenda is focused on five sets of deliverables that correspond to each of the five steps in the six sigma roadmap. The agenda in EFQM is a press toward Business Excellence following nine specific criteria. In BSC, the agenda embraces a pronounced vision, mission, and strategy for the organization and sets concerted objectives, measures, targets, and initiatives in each of four perspectives: learning and growth, business process, customer, and financial. Finally, the agenda in Juran is to bring processes under statistical control. There is no specific direction, and it is contingent on what processes are selected for control.

Based on a detailed analysis of each approach mentioned in Figure 2, we can formulate the following technological rules for the agenda. These rules systematically embody a management design theory comprising the first element in the process theory of Figure 1.

If you want to improve software processes in a situation where you
- believe that "best practices" for an improvement area can be identified,
- trust the usefulness of practices from another organizational or national setting,
- agree that your improvement effort will be alike to what other companies have done; you are not special in relation to this,

then choose a *universally* applicable model, such as CMM or Bootstrap. If not, then choose a *situated* model, such as Juran or Six Sigma.

If you want to improve software processes in a situation where you
- need a vision to motivate and give direction to your improvement effort,
- believe there is one and only one path to a future desirable state,
- agree that your improvement effort should be directed by one single vision—and *not* balance many organizational activities and resources as to optimize performance,

then choose a *directing* model, such as Six Sigma, or ISO 9000. If not, then choose a *balancing* model such as Juran or Balanced Scorecard.

Our analysis has been able to develop similar rules for the remaining process elements of the theory depicted in Figure 1: observation, analysis, and synthesis processes. Observation is characterized as either detached or participatory. Analysis is characterized as either statistical or interpretive. In terms of modes of synthesis, we found that the various process improvement approaches either confined developers to a finite recommendation set, or allowed an open-ended, generative style for recommendations. Some approaches used a synthesis approach that involved a distinct model for synthesis. Other approaches had less-defined synthesis stages that were dependent on the tacit knowledge or know-how of the developers.

3.1   Grounding the Process Improvement Case in Social Science

In terms of social science research methodology, this paper reports work that is conceptual in nature; it is design theorizing for the purpose of improving management of process improvement. The theorizing is grounded in an analysis of the published process improvement models, and the theoretical results are expressed as models and technological rules.

The analysis (Pries-Heje and Baskerville 2003) of the published models was done following grounded theory coding techniques (Glaser and Strauss 1967; Strauss and Corbin 1998). In the concrete, grounded theory is a qualitative social science research methodology that takes its name from the practice of discovering theory that is grounded in data. Grounded theory is best used in research where one has relatively

uncharted territory, as was the case with the identification of technological rules implicit and embedded in process improvement models. Grounded theories are inductively discovered by careful collection and analysis of qualitative empirical data. That is, this method does not begin with a theory, and then seek proof. Instead, it begins with an area of study and allows the relevant theory to emerge from that area through a three-step process called open, axial, and selective coding (Strauss and Corbin 1998). Figure 1 is a pictorial representation of what emerged in this first case.

3.2   Practical Implications of the Process Improvement Case

The set of matrices, technological rules, and underlying principles form a *framework* that can help organizations make sense of different normative models, and link them to their organizational and improvement goals. Our framework distinguishes between four significant dimensions: agenda, observation, analysis, and synthesis. These dimensions can be used to examine an organization's needs and then select an appropriate improvement model in an informed and systematic way.

In practice, a manager should try to place his own organization, its values, and its beliefs within the framework. Let us take agenda as an example. A *universal perspective* embeds trust in maturity models such as CMM and Bootstrap. More generally, the universal perspective focuses on models of best practices and, consequently, models of general process problems. A *situated perspective*, on the other hand, focuses on what software practitioners and their managers perceive as problems in the process.

Furthermore, the manager should consider their end goal. Is CMM level 5 regarded as an attractive state-of-the-practice in the organization? In general, models having a *cosmopolitan vision* provide an organizational direction toward better and better development. Such models assume that the organization is in a development "state" and has the opportunity to change to an improved state. Usually, this goal also assumes that there will be further opportunities to improve so the improvement process is seen as progressive. On the other hand, *harmonious standard models* have a clear aim toward building a value system for quality among other important organizational values. A balancing strategy assumes that something is missing in the development organization: some activity, value, or element that must be added or restored in order to improve the software process.

4   Case 2: User Involvement

User participation can be defined as "participation in the system development process by potential users or their representatives" (Barki and Hartwick 1989, p. 53). Many have identified lack of fulfilment of user needs in information technology projects as a major problem. Clavadetcher (1998, p. 30) summarized the problem:

> Quite simply, the software we build does not meet our customers' needs. Those of us who build large software programs fail miserably—90 percent of the time—to deliver what customers want, when they want it, at the agreed-upon price. We fail to adequately manage the software development process.

> User– developer communication breaks down; the requirements control process breaks down; we have runaway requirements, budgets, schedules, and "death march" projects.

Traditionally, user participation has been found to be a major factor in systems' success. This finding builds on theories of participative decision making (Barki and Hartwick 1994) and the user role in organizational change (Baroudi et al. 1986). There is not total agreement on the benefits of user participation. For example, Ives and Olson (1984), in a review of IS research, found mixed results on user participation. Likewise, Cavaye (1995) studied the relationship between user participation and success and found that the relationship was more complex than just more user participation leading to more success.

As with organizational change there are also many different fundamental theories for user participation. Bødker (2004) provides an excellent overview of a number of these techniques for user participation. For certain specific techniques, such as paper prototypes, Bødker recommends user participation early in the development process in order to elicit requirements from the users. Saleem (1996) recommends user participation when task uncertainty is high. However, the combination where one involves users early in projects with high task uncertainty cannot be found in the literature.

The process by which an IT project—typically by the project manager—selects the appropriate approach and time for user participation is often *ad hoc*. Each approach has its advocates and adherents, and there is little comparative research for choosing among such approaches. Thus we set out to develop a framework of technological rules that could be useful in this situation.

The combined findings from a literature study and a field study (see section 4.4 on grounding) describe a management design theory about user participation that comprises three major influences on user participation: complexity, resources, and user identity. For shorthand, we will call this the CRU management design theory for user participation. Below we systematically elaborate each element of the CRU management design theory and formulate technological rules for each of these three elements.

4.1 Complexity

The complexity issue is characterized by six major factors that give rise to complexity. These include domain knowledge, task complexity, size, technical knowledge, perceived change, and the type of system.

The first factor leading to complexity is the degree of knowledge held by the developers in the domain in which the users work. Developers need knowledge about the existing working styles in order to develop the right system for future work. Lack of domain knowledge among developers is one of the three key problems of systems development when the system is large (Curtis et al. 1988). Where developers lack knowledge of the users' work, they need to observe and experience the users while they do their work (Kensing and Munk-Madsen 1993). Where the users' work is well-known to the developers, then reviews are needed to ensure the knowledge is still accurate. A high degree of user participation is needed where developers lack domain

knowledge (Saleem 1996), and this participation is less urgent where developers have strong domain knowledge. Cavaye (1995) notes that user participation is less urgent if user requirements are well known.

The second factor is task complexity. Where tasks are structured at the operational level, the demand for user participation is minimal. Where tasks are unstructured and only described at the strategic level, then the need for user participation is urgent (Cavaye 1995). This determination is complicated by the presence of both operational and strategic tasks. For example, cases where the users involved are not knowledge-able enough about certain tasks to determine if these were operational or strategic may lead to partial failures in the inevitable system (Wilson et al. 1997). When task uncertainty is high, then a high degree of user participation is recommended (Saleem 1996). On the other hand, if a system is well structured and well defined, then it is not necessary to involve users for purposes of system quality (Ives and Olson 1984) but perhaps for system acceptance.

Size is the third factor we found that influences complexity. Size is one of the main influences both on system risk (Applegate et al. 1999) and complexity (Cavaye 1995). User participation is common where the system is perceived as large, such that participation may be impractical if the system is perceived as small (Cavaye 1995). The distinction between large and small will, of course, vary from setting to setting. For our purposes, we asked participants in our field study how they would distinguish large from small systems. For this setting, the users indicated that projects longer than 36 man months (or 24 calendar months) are large, while a project of less than 12 man months or 12 calendar months was perceived as a small system.

The fourth influence factor on complexity was knowledge of the technology. Lack of technological knowledge (i.e., on hardware, operating system, database management, programming language, etc.) increases complexity. This factor is a known area that increases risk (Applegate et al. 1999), and we know that there is a need for a balance between the complexity of the application and the complexity of the technology (Nicholas 1985). User participation may be unsuitable when considerable technical expertise is needed (Ives and Olson 1984).

The fifth influence factor is perceived change for stakeholders. If change is perceived to be considerable, then there is an advantage in involving users (Cavaye 1995). In addition, a larger the number of stakeholders is more likely to have a wider variety of goals for the system.

Finally, the sixth influence factor is the type of system. For transaction based systems the traditional way of involving users is through information given from users to developers. Based on this information, the developers then formulate user needs and requirements. While this may be sufficient for transaction systems, decision support systems involve more complex work flows and a higher degree of user participation may be needed (Hawk and Dos Santos 1991). This finding is consistent with work showing that when developing standard applications, such as a payroll system, a small degree of user participation may suffice (Saleem 1996). One useful way of determining the type of system in this regard is to relate complexity to the number of different kinds of interfaces involved.

A management decision on what to do with user involvement in reaction to complexity could be "designed" through the development of the following technological rule:

> If the
> - degree of domain knowledge by developers is low, or
> - task complexity is high, or
>   - size of system to be build is large, or
>   - technical knowledge is low, or
>   - perceived change is great, or
> - the type of system is decision support (as opposed to transaction based),
>
> then a **higher** degree of user participation is needed.

4.2 Resources

The literature review developed three major influence factors regarding resources.

The first factor was management support. This support will increase the prospect of user participation (Cavaye 1995). Where senior managers are only moderately positive, or worse still, resistant, then risk increases considerably (Applegate et al. 1999). Management must support user participation in both word and deed, along with commanding results from the participation of users.

The second factor involves resources in the form of adequate budget and staff for the project (Cavaye 1995). When a project has limited resources, then users will be less involved simply because user participation is expensive. For example, a workshop with 12 users for two days may involve three or four developers in preparation and follow-up. The total cost of such a workshop may be more than two man months.

The third factor is time, especially with regard to whether the project has a hard or a soft deadline. If a project has a hard deadline, it may be more difficult to find the calendar time for user participation. User participation techniques often require planning well in advance. For example, finding a day for a workshop involving 12 busy people is impossible with short notice. Projects with hard deadlines can exclude effective engagement for user participation (Wilson et al. 1997).

The technological rule that we designed based on these findings (Pries-Heje and Baskerville 2008) was

> If the
> - management support is high, and/or
> - the budget and staff allocated for project is adequate, and/or
> - time pressure is insignificant,
>
> then a **higher** degree of user participation can be advantageous.

4.3 User Identity

The degree to which the users are personally known to the developers depends on the type of development. User identity is classified as "named" or "nameless" users. For example, Grudin (1991) distinguishes between in-house development, custom development, competitively bid and contract development, and product development. For

in-house and custom development, the developers know the users from the very beginning. The most practical measure of this identity is whether the developers can name the users, or at least to obtain the names of the users in advance. In product development, developers do not know their users until the users buy the product. It is possible to identify potential users such as representative users of the last version of a product, but the complete set of users necessarily remain nameless until they acquire the product.

This leads to the simple technological rule:

> If the
> * users are nameless,
> then traditional user participation should be avoided.

Collectively, these three technological rules systematically define the CRU management design theory much as a system of hypotheses might define an explanatory scientific theory.

4.4 Grounding the User Involvement Case in Social Science

To develop a useful framework of technological rules for deciding when to have users participate in an IT project, we carried out a study involving three researchers: one tenured professor (one author of this paper) and two students writing a dissertation on user participation. In order to solve the how and when problem of user participation, we researched methods and techniques for user participation. As step one, we identified and reviewed hundreds of research papers and books on the subject. From this review, we developed a set of popular but alternative methods and techniques for user participation. Based on this set of techniques, we then conducted a field study in ten companies. We call this field study phase 1. In the concrete, we conducted exploratory interviews focusing on how user participation in practice took place. These were followed by semi-structured interviews, using a think-aloud test. After the field study (phase 1) we initiated phase 2, analyzing the alternative approaches discovered in the first phase. This lead to the technological rules presented above.

4.5 Practical Implications of the User Involvement Case

There seems to be widespread agreement that user participation is positive, is of high utility, and can be extremely valuable. However, many IT project managers don't know how and when to do what. They cannot implement user participation in practice in their project. The result being, in many cases, that user participation in practice is just something that is talked about and not practiced.

To solve that problem, we have developed the user participation technological rules that can help IT project managers decide when and how to have users participate. The technological rules were developed based on an extensive literature survey, an interview study, as well as a field study.

The technological rules approach is meant to be used at a workshop in the early phases of an IT project. We rigorously tested the technological rules in practice with

project managers from ten companies. The project managers found that we had designed one possible and (to them) useful answer to the user participation problem (Pries-Heje and Baskerville 2008).

5   Case 3: Organizational Change

How can an organization select the best change strategy from the abundance of different foundational theories for organizational change? Each theory has its advocates and adherents, and there is little comparative research to aid the selection. The theories are so varied that comparisons are usually drawn between only a few alternatives (Tingey 1997). Our next case focuses on this selection issue, the lack of formulated tools to help organizational change managers to select from these change theories. Our intention is to improve the ability for organizational change managers to rationally select the most appropriate change strategies by designing technological rules to guide the decision making.

In connection with our survey of the organizational change literature, we conducted a number of search conferences involving participants from the Danish companies in order to assemble a catalogue of change approaches, which have been used successfully in practice. From the search conferences, we identified a number of high-level overall approaches. We analyzed them to determine their distinguishing characteristics and related them to the theories in the literature. We focused on the essential attributes of the organizational setting and the particular way of approaching change strategy. These are refined into ten prominent change strategies that can be represented as technological rules.

Each of these approaches was founded on the presence of highly specific conditions in the organizational setting, specific goals for the organizational change, and particular reasons for implementing change in the context of the organizational setting. These foundations embody a management design theory based on conditions, goals, and reasons. For shorthand, we will call this the CGR management design theory.

Following this analysis, we set out to elaborate the CGR management design theory to create technological rules to guide change managers in choosing which of the 10 change strategies would be most appropriate in an actual organizational setting. For example, for the change strategy called "commanding," we formulated the following assertions:

- Right now, we need change to happen fast
- It is primarily organizational structures that need to be changed
- In the past, we have had successes in requiring or dictating change

And for the change approach called "optionality," we formulated the assertions:

- Our employees are self-aware and always have an opinion
- We have very knowledgeable employees that know their areas well
- There are vast differences between the tasks of different employees

A management decision to adopt one of these two approaches to organizational change could be "designed" through the development of the following two technological rules. The references indicate examples of the approach recommended in the rule.

If you want to initiate organizational change in a situation where you believe
- that formal structures needs change, and
- change is needed fast,

then choose a *commanding* approach where change is driven and dictated by (top) management; one where management takes on the roles as owner, sponsor, and change agents (Huy 2001).

If you want to initiate organizational change in a situation where you believe
- that target group is very diverse and has large individual differences, and
- the target group are experts,

then choose an *optionality* approach where change is driven by the motivation and need of the individual; it is to a large degree optional (Rogers 2003).

Following similar developments, we defined eight further technological rules.

If you want to initiate organizational change in a situation where you believe
- that the need for change arises among the employees,
- that there is no need for a standardized approach, that the result is more important than the process, and
- an open management style will allow change to arise from the bottom,

then choose an *employee driven* approach where change is driven from the bottom of the organizational hierarchy when needs for change arise among employees (Andersen et al. 2001; Kensing 2003; Kensing and Blomberg 1998).

If you want to initiate organizational change in a situation where you believe
- that dynamic and complex surroundings make it important to explore an open management style that will allow change to arise from the bottom,

then choose an *exploration* approach where change is driven by the need for flexibility, agility, or a need to explore new markets, technology or customer groups (Benner and Tushman 2003; Mintzberg 1983).

If you want to initiate organizational change in a situation where you believe
- there is a need for change in attitudes and/or behavior,
- the organization is talented in learning, and
- relationships between means and goals are unclear,

then choose a *learning driven* approach where change is driven by a focus on organizational learning, individual learning, and what creates new attitudes and behavior (Huy 2001).

If you want to initiate organizational change in a situation where you believe
- there are relatively stable surroundings so measurements from the past can be used to decide the future,

then choose a *metrics driven* approach where change is driven by metrics and measurements (Oakland 2003; Pande and Holpp 2000).

If you want to initiate organizational change in a situation where you believe
- there are relatively stable surroundings, and
- there are many homogeneous resources and work flows,

then choose a *production organized* approach where change is driven by the need for optimization and/or cost reduction (Benner and Tushman 2003; Huy 2001).

If you want to initiate organizational change in a situation where you believe
- a need exists for major change (for example, when organization has ground to a halt),
- nothing new happens,
- decisions are made but not carried out, and
- a crisis is eminent,

the choose a *reengineering* approach where change is driven by fundamentally rethinking and redesigning business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed (Bashein et al. 1994; Boudreau and Robey 1996; Davenport 1993; Hammer 1990; Hammer and Champy 1993; King 1994; Malhotra 1998; Willcocks et al. 1997).

If you want to initiate organizational change in a situation where you believe
- organizational skills and capabilities need to be developed,
- no unhealthy power struggles occur (so people can talk), and
- employees that can be exemplars are available,

then choose a *socializing* approach where change in organizational capabilities is driven by working with social relationships and diffusion of innovations happens through personal contacts rather than through plans and dictates (Huy 2001).

If you want to initiate organizational change in a situation where you believe
- work has vast complexity and variety so there really is a need for special knowledge, and
- there is access to necessary specialists, eventually by in-sourcing them,

then choose a *specialist driven* approach where change is driven by specialists, either with professional, technical, or domain knowledge (Ciborra 2000; Mintzberg 1983; Simon 1973, 1983; Woods 1988; Woods and Hollnagel 1987).

All of these technological rules represent systematic expressions of the CGR theory, conditions in the organizational setting, a stated goal of organizational change, and the reason for implementing change in the context of the organizational setting.

5.1 Grounding the Organizational Change Case in Social Science

We exercised these technological rules in two organizations, using them to design organizational change initiatives in each organization (Pries-Heje and Baskerville 2008). The rules took the form of a query form where managers expressed their degree of agreement or disagreement with the conditions underlying the rule connoted by statements. The degree to which the conditions for change in that organization can be compared to the conditions for each of the 10 rules is presented in Table 1. The fit of each is indicated by the percentage (0 to 100 percent) to which the rule's conditions are present in the particular organization. Take, for example, socializing. Here the rule is three-fold, consisting of answers to the following three statements: (1) we have situations where we believe organizational skills and capabilities need to be developed; (2) we have no unhealthy power struggles occuring (so people can talk); and (3) employees that can be exemplars are available. If the group of managers fully agree (equals 100 percent) with statement (1), partly agree (67 percent) with statement (2), and partly disagree (33 percent) with statement (3). Then the combined fit is calculated as (100 + 67 + 33 / 3).

A fit calculated above 67 percent means that the corresponding change strategy fits the organization well (will be successful). This application led us to change design recommendations in each company to achieve the best-fitting change strategies.

In both companies, the management of the IT division found the results quite positive and considered them very useful. In Company A, the CIO called the results a major "Aha!" experience. The recommendations at Company B led to a hybrid design using the "optionality" strategy on those change initiatives driven by the individual's or group's need and motivation and using the "commanding" strategy for designing change initiatives where they really needed to drive change fast.

Table 1. The Degree of Fit for Each of the 10 Change Strategies in the Evaluations

Company A	Company B
60% Socializing	71 % Optionality
60% Learning driven	65 % Commanding
56% Production organized	59 % Socializing
55% Employee driven	58 % Production organized
54% Optionality	56 % Specialist driven
42% Metrics driven	40 % Metrics driven
37.5% Specialist driven	34 % Learning driven
35% Exploration	29 % Exploration
34,5% Commanding	28 % Reengineering
31% Reengineering	18 % Employee driven

5.2   Practical Implications of the Organizational Change Case

We designed and implemented the organizational change technological rules as a coherent artifact and evaluated it within a research project involving three participating companies. The IT organization in two of these companies was particularly involved in evaluating through an action research field study. In one of the evaluated organizations, the management group committed to the prescribed change strategy—not in detail, but in principle. This result is nearly ideal in relation to the prescriptions from the nexus. In the other organization evaluated, the results were also quite positive and the framework of technological rules was evaluated as very useful.

   Whether the visions for strategic change in the two organizations will be achieved will take another two to three years to develop. At the moment, however, the organizational change framework of technological rules clearly leads to operational management decisions about change strategy.

6   Conclusion

This paper contributes a fresh perspective on how management planning based on design science operates through expressions of a particular type of design theory called management design theory. These types of theories can be expressed through systems of technological rules. By applying design science research as a guide for designing general frameworks for decision making (that is to say, heuristics), we help managers (in their own perception) more easily approach complex, strategic decisions. The approach is built on the concept of simple design theories and technological rules, a simple expression of the design theory that relates a general organizational situation to a general course of action. Three grounded cases—process improvement, user involvement, and organizational change management—illustrate and validate the concepts.

   Our three cases provide practical contributions. However, in general, this design science approach to designing management decisions demonstrates that design concepts have great worth for improving management activities, a field of work that is not usually associated with design. This strategic framing of organizational design decisions contributes to the general core of design research by demonstrating that technological rules are an operational form of managerial design theory.

References

Andersen, C.V., Krath, F., Krukow, L., Mathiassen, L., Pries-Heje, J.: The Grass Root Effort. In: Mathiassen, L., Pries-Heje, J., Ngwenyama, O. (eds.) Improving Software Organizations: From Principles to Practice. Addison-Wesley, Upper Saddle River (2001)

Applegate, L., McFarlan, F.W., McKenney, J.L.: Corporate Information Systems Management: Text and Cases. Irwin–McGraw Hill, New York (1999)

Archer, M.S.: Realist Social Theory: The Morphogenetic Approach. Cambridge University Press, Cambridge (1995)

Barki, H., Hartwick, J.: Rethinking the Concept of User Involvement. MIS Quarterly 13(1), 55–63 (1989)

Barki, H., Hartwick, J.: Rethinking the Concept of User Involvement and User Attitude. MIS Quarterly 18(1), 59–79 (1994)

Baroudi, J.J., Olson, M.H., Ives, B.: An Empirical Study of the Impact of User Involvement on System Usage and Information Satisfaction. Communications of the ACM 29(3), 232–238 (1986)

Bashein, B.J., Markus, M.L., Riley, P.: Preconditions for BPR Success: And How to Prevent Failures. Information Systems Management 11(2), 7–13 (1994)

Benner, M., Tushman, M.: Exploitation, Exploration, and Process Management: The Productivity Dilemma Revisited. Academy of Management Review 28(2), 238–256 (2003)

Bødker, K., Simonsen, J., Kensing, F.: Participatory IT Design: Designing for Business and Workplace Realities. The MIT Press, Boston (2004)

Boudreau, M.-C., Robey, D.: Coping with Contradictions in Business Process Reengineering. Information Technology and People 9(4), 40–57 (1996)

Cavaye, A.L.M.: User Participation in System Development Revisited. Information Management 29, 311–323 (1995)

Ciborra, C.U.: From Control to Drift: The Dynamics of Corporate Information Infrastructures. Oxford University Press, Oxford (2000)

Clavadetscher, C.: User Involvement: Key to Success. IEEE Software 15(2), 30–32 (1998)

Curtis, B., Krasner, H., Iscoe, N.: A Field Study of the Software Design Process for Large Systems. Communications of the ACM 31(11), 1268–1287 (1988)

Davenport, T.H.: Process Innovation: Re-engineering Work through Information Technology. Harvard Business School Press, Boston (1993)

Denyer, D., Tranfield, D., van Aken, J.: Developing Design Propositions through Research Synthesis. Organization Studies 29(3), 393–413 (2008)

Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory, Strategies for Qualitative Research. Aldine Publishers, Chicago (1967)

Gregor, S., Jones, D.: The Anatomy of a Design Theory. Journal of the Association for Information Systems 8(5), 312–335 (2007)

Grudin, J.: Interactive Systems: Bridging the Gaps Between Developers and Users. IEEE Computer 24(4), 59–69 (1991)

Hammer, M.: Reengineering Work: Don't Automate, Obliterate. Harvard Business Review, 104–112 (July-August 1990)

Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business, New York (1993)

Hawk, S., Dos Santos, B.L.: Successful System Development: The Effect of Situational Factors on Alternate User Roles. IEEE Transactions on Engineering Management 38(4), 316–327 (1991)

Hunter, R.B., Thayer, R.H.: Introduction. In: Hunter, R.B., Thayer, R.H. (eds.) Software Process Improvement, pp. 1–4. IEEE Computer Society Press, Los Alamitos (2001)

Huy, Q.N.: Time, Temporal Capability, and Planned Change. Academy of Management Review 26(4), 601–623 (2001)

Ives, B., Olson, M.H.: User Involvement in MIS Success: A Review of Research. Management Science 30(5), 586–603 (1984)

Kant, I.: The Critique of Pure Reason (1781); J. Watson, trans., In: Rand, B. (ed.) Modern Classical Philosophers, pp. 370–456. Houghton Mifflin, Cambridge (1908)

Kensing, F.: Methods and Practices in Participatory Design. ITU Press, Copenhagen (2003)

Kensing, F., Blomberg, J.: Participatory Design: Issues and Concerns. Computer Supported Cooperative Work 7(3/4), 167–185 (1998)

Kensing, F., Munk-Madsen, A.: PD: Structure in the Toolbox. Communications of the ACM 36(6), 78–85 (1993)

King, W.R.: Process Reengineering: The Strategic Dimensions. Information Systems Management 11(2), 71–73 (1994)

Malhotra, Y.: Business Process Redesign: An Overview. IEEE Engineering Management Review 26(3), 27–31 (1998)

March, S.T., Smith, G.: Design and Natural Science Research on Information Technology. Decision Support Systems 15(4), 251–266 (1995)

Mintzberg, H.: Structure in Fives: Designing Effective Organizations. Prentice-Hall, Upper Saddle River (1983)

Nicholas, J.M.: User Involvement: What Kind, How much and When? Journal of Systems Management 36, 23–37 (1985)

Oakland, J.S.: TQM—Text with Cases, 3rd edn. Butterworth-Heinemann, Burlington (2003)

Pande, P.S., Holpp, L.: What Is Six Sigma? McGraw-Hill, New York (2000)

Pawson, R., Tilley, N.: Realistic Evaluation. Sage Publications, London (1997)

Pries-Heje, J., Baskerville, R.: Improving Software Organizations: An Analysis of Diverse Normative Models. Paper presented at the EuroSPI, Graz, Austria (2003)

Pries-Heje, J., Baskerville, R.: The Design Theory Nexus. MIS Quarterly 32(4), 731–755 (2008)

Rogers, E.M.: Diffusion of Innovations, 5th edn. Free Press, New York (2003)

Saleem, N.: An Empirical Test of the Contingency Approach to User Participation in Information Systems Development. Journal of Management Information Systems 13, 145–166 (1996)

Simon, H.A.: The Structure of Ill Structured Problems. Artificial Intelligence 4, 181–201 (1973)

Simon, H.A.: Search and Reasoning in Problem Solving. Artificial Intelligence 21, 7–29 (1983)

Simon, H.A.: The Science of the Artificial, 3rd edn. MIT Press, Cambridge (1996)

Strauss, A., Corbin, J.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, 2nd edn. Sage Publications, Thousand Oaks (1998)

Tingey, M.O.: Comparing ISO 9000, Malcolm Baldrige, and the SEI CMM for Software: A Reference and Selection Guide. Prentice, Upper Saddle River (1997)

Vaishnavi, V., Kuechler, W.: Design Research in Information Systems. Association for Information Systems (2004), http://www.isworld.org/Researchdesign/drisISworld.htm (retrieved October 2, 2004)

van Aken, J.E.: Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. The Journal of Management Studies 4(2), 219–246 (2004)

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 (2005a)

van Aken, J.E.: Valid Knowledge for the Professional Design of Large and Complex Design Processes. Design Studies 26(4), 379–404 (2005b)

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)

Willcocks, L., Feeny, D., Islei, G.: Managing IT as a Strategic Resource. McGraw-Hill, New York (1997)

Wilson, S., Bekker, M., Johnson, P., Johnson, H.: Helping and Hindering User Involvement: A Tale of Everyday Design. Paper presented at the CHI 1997, Atlanta, Georgia (1997)

Woods, D.D.: Coping with Complexity: The Psychology of Human Behavior in Complex Systems. In: Goodstein, L.P., Andersen, H.B., Olsen, S.E. (eds.) Tasks, Errors and Mental Models, pp. 128–148. Taylor and Francis, London (1988)

Woods, D.D., Hollnagel, E.: Mapping Cognitive Demands in Complex Problem-Solving Worlds. International Journal of Man-Machine Studies 26, 257–275 (1987)

About the Authors

Jan Pries Heje is Professor in Information Systems, Department of Communication, Business and Information Technologies, Roskilde University, and head of the User Driven IT-Innovation Research Group. His research focuses on designing and building innovative solutions to managerial and organizational IT problems. Previous and current projects explore process improvement as design, the ability for an organization to improve, and how one can design a process for making better sourcing decisions. Dr. Pries-Heje is co-chair of the joint IFIP 8.2/8.6 working conference on "Human Benefit through the Diffusion of IS Design Science," and, from January 2010, chair of IFIP TC 8 Working Group 8.2 on Information Systems. He can be reached by e-mail at janph@ruc.dk.

Richard L. Baskerville is a Board of Advisors Professor of Information Systems at Georgia State University. His research and authored works regard security of information systems, methods of information systems design and development, and the interaction of information systems and organizations. Dr. Baskerville currently serves as editor-in-chief of the *European Journal of Information Systems.* He is a Chartered Engineer, holds a B.S. (summa cum laude) from the the University of Maryland, and M.Sc. and Ph.D. degrees from the London School of Economics, University of London. He can be reached by e-mail at baskerville@acm.org.