Soft Design Science Methodology

Richard Baskerville
Georgia State University
CIS Dept
PO Box 4015
Atlanta GA 30302
USA
+1 404 413-7362
baskerville@acm.org

Jan Pries-Heje
Roskilde University
Dept. of Comm., Bus. & IT
Universitetsvej 1
4000 Roskilde
Denmark
+45 72 18 50 00
janph@ruc.dk

John Venable
Curtin University of Technology
School of Information Systems
GPO Box U1987
Perth WA 6845
Australia
+61 8 9266 7054
j.venable@curtin.edu.au

Abstract

This paper proposes and evaluates a soft systems approach to design science research. Soft Design Science provides an approach to the development of new ways to improve human organizations, especially with consideration for social aspects, through the activities of design, development, instantiation, evaluation and evolution of a technological artifact. The Soft Design Science approach merges the common design science research process (design, build-artifact, evaluation) together with the iterative soft systems methodology. The design-build artifact-evaluation process is iterated until the specific requirements are met. The generalized requirements are adjusted as the process continues to keep alignment with the specific requirements. In the end, the artifact represents a general solution to a class of problems shown to operate in one instance of that class of problems. The proposed methodology is evaluated by an analysis of how it differs from, and could have informed and improved, a published design science study, which used a design-oriented action research method.

Categories and Subject Descriptors

A.0, A.m, H.m (Information Systems miscellaneous)

General Terms

Management, Design

Keywords

Research Methodology, Soft Design Science, Design Science Research, Soft Systems Methodology, Action Research

1. INTRODUCTION

Design Science Research (DSR) is used to develop new technologies for solving problems. Such problems and solutions are often socio-technical in nature, which creates problems for DSR in gaining problem understanding, identifying systemically appropriate solutions, and in effectively evaluating new and innovative solutions. Baskerville et al [1] proposed the development of a softer approach to DSR to address these issues. This paper proposes and evaluates such a soft approach to design science research (Soft DSR), which combines aspects of Soft Systems Methodology (SSM) [2-4] with DSR. Soft Design Science provides an approach to the development of new ways to improve human organizations, especially with consideration for social aspects, through the activities of design, development, instantiation, evaluation and evolution of a technological artifact.

This paper looks at some of the literature on DSR, Action Research, and SSM in the next section before developing a design and rationale for a Soft DSR approach in the subsequent section. Following that, the proposed Soft DSR approach is evaluated by analyzing what insights might have occurred and improvements made if the Soft DSR approach had been applied to a published case study, rather than the design-oriented action research method that was applied. Finally, the analysis is discussed and conclusions are drawn.

2. LITERATURE REVIEW

This section reviews literature that serves as the basis for the development of a Soft DSR approach. It considers Design Science Research, Action Research, and Soft Systems Methodology. Action Research as reviewed because of its similarities to both DSR and SSM and because a design-oriented action research study was used in the case study that is analyzed to evaluate potential benefits of a Soft DSR approach.

2.1 Methodology in Design Science Research

Design Science Research (DSR) has been cast as a paradigm rather than a discrete research methodology. Iivari [5] describes its distinct character as a paradigm in terms of its ontology, epistemology, methodology and ethics. He equates the approach to constructive research methodology that is distinct from action research historically, practically, and philosophically.

Methodology in DSR has largely been regarded tangentially in the literature. Indeed, its basic method so closely approximates the common view of the "scientific method" that it seems acceptable as an assumption. The researcher learns about artifacts and natural settings by formulating hypotheses (a design), conducting an experiment (instantiating an artifact), and matching the results to the expectations (evaluating). Perhaps this simple research process (design, build-artifact, evaluation) requires no further elaboration.

Consequently, the major seminal works in information systems DSR have been concerned with its relevance in the philosophy of science [5, 6], the nature of theory in DSR [7, 8], and criteria for evaluating DSR [9]. The research methodology implied by these works seems often to be regarded as an outcome of the philosophy, theory, and criteria.

2.1.1 Iteration in DSR

DSR has not usually been regarded as an iterative process. It is, instead, mostly regarded as episodic. Perhaps its strong conceptual links to the professions of engineering and computer science anchor theory to a set of complex specifications, and the construction process is so complex and expensive that revisions after evaluation are more-or-less regarded as separate design science episodes rather than iterative artifact development.

One simple representation of this episodic method is shown in Figure 1 [10]. This approach consists of four activities: (1) Search, (2) Ex Ante Evaluation, (3) Construction, and (4) Ex Post Evaluation. This approach assumes that problem identification and specification is part of the search process that develops the design. The two major products in the method are the design and the artifact. The "scientific" learning arises from the search process, the construction, and the two evaluations.

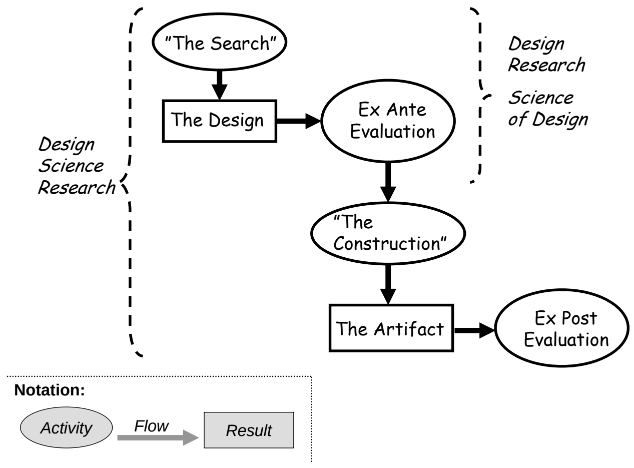

Figure 1. Episodic Design Science Research Method.

Such an episodic approach is appropriate where the design can be relatively complete and final in its specification and the evaluation of the artifact will complete the research project (or episode).

It certainly seems likely that this method can be operated iteratively. Doing so implies that the design and the artifact must be regarded as tentative, and it suggests that these must necessarily be simpler, less complex, and less costly if the process is to be repeated multiple times. Such an approach can be most simply represented by adding a "repeat" arrow to the process. See Figure 2.

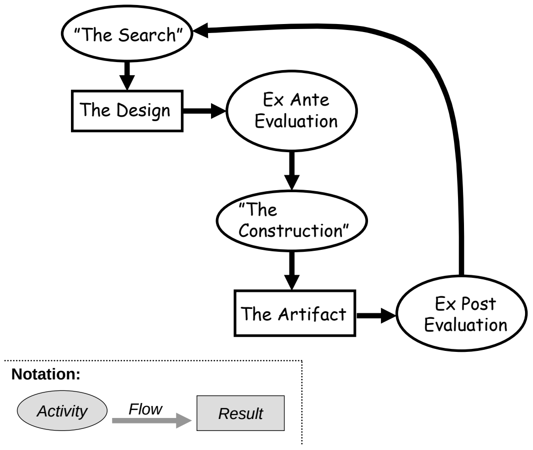

Figure 2. Iterative Design Science Research Method.

2.2 Prototyping

An iterative DSR method raises the similarity of such a DSR method to prototyping, and its more general (and older) form in organizational development, action research. Prototyping is a venerable system development methodology that involves construction and test of prototypes of systems, often for purposes of clarifying vague requirements [11] and often in collaboration with the prospective users [12].

Prototypes take different forms for different purposes. The simplest is a mock-up prototype that models physical aspects of the final system. Another example is an evolutionary prototype which is a modifiable, running model of part of a system. This form of prototyping underlies many release-oriented software products [13]. Since each of these forms of prototyping involve designing and deploying artifacts for the purpose of learning about the artifact or its environment, it seems problematic to exclude prototyping methodologies from use within DSR. Because of the collaborative nature of prototyping, it is regarded as a systems development approach that can also become a mechanism for organizational change [14]. From this perspective, the prototyping potential for organizational development also makes it useful as one form of action research methodology [15].

2.3 Action Research

The similarity between action research methodology and DSR has already sparked a discourse in the literature. On the surface, the two approaches to scientific discovery seem quite distinct. Action research aims at organizational action to create change in order to discover new knowledge in a clinical mode. DSR aims at design to create an artifact in order to discover new knowledge in a generative (or creative) mode. See
Figure 3.

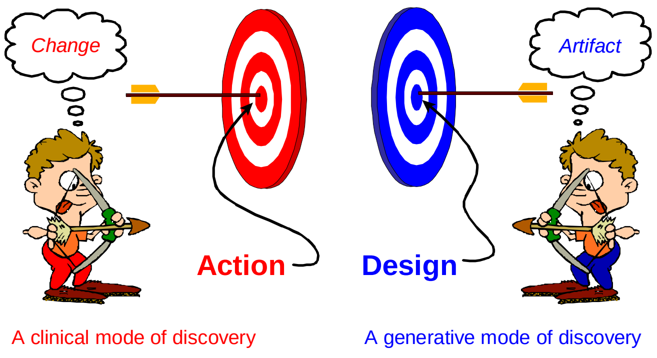

Figure 3. Surface distinctions between action research and design science.

However, these surface distinctions begin to evaporate when design is acknowledged as organizational action, and when situating a new or different artifact in an organizational setting is acknowledged as an organizational change. Likewise, because DSR can produce concepts, constructs, and models, as well as artifact instantiations [6], the intellectual products of both approaches can be the same. In a practical sense, either approach begins to seem subsumable within the other.

The similarities between DSR and action research at a high level of abstraction are undeniable. Both generate scientific knowledge by intentionally modifying a real setting and by carefully evaluating the result [16]. Evidence of the similarity also becomes apparent when evaluating the results of each research approach. In certain cases, each approach can satisfy the accepted criteria for the other [17]. Sein et al [18] (unpublished) are currently developing the "Action Design" approach to further merge action research and design science.

But the similarities, and the ease with which one approach can merge with the other does not mean that these research approaches are the same. Iivari's analysis [5] explains how the approaches arise from historically different roots (action research in the sociotechnical movement and design science in engineering). Their basic practical intentions are quite different (action research in treating social illnesses and design science in construction of artifacts). These also differ ontologically (action research is nominalistic, idealistic, and constructivistic while design science is realistic and materialistic); epistemologically (action research tends to be anti-positivist while design science tends to be positivist); and methodologically (action research idealizes client situations while design science idealizes laboratory situations). Iivari and Venable [19] further show differing levels of overlap depending on their variants – (1) non-overlapping (i.e. disjoint), (2) somewhat overlapping, and (3) significantly overlapping but that they are never exactly the same. The third case could be termed "design-oriented action research", which is the kind considered here.

Clearly there are both similarities and differences between DSR and action research. Further, both approaches also share an ability to adopt prototyping as an underlying approach where implementing and evaluating an information system is an element of the research activity. From our perspective, portions of design science, social science, and computer science are arriving at very similar, though still different, points in their research approaches.

Having considered the literature on DSR and AR, we now introduce the literature on Soft Systems Methodology.

2.4 SOFT SYSTEMS METHODOLOGY

Soft systems methodology (SSM) is a prominent systems science approach to social-technical problems [2-4]. Because it emerged from the juncture of action research and systems science, it is regarded as a form of action research [15]. In practice, SSM is less often used for research per se, and more often used purely as a systems development methodology. Its systems science and systems thinking make it effective for developing systems (including information systems) that succeed in difficult social organizational settings. It is also an iterative approach to systems development (like prototyping), but one that accounts for the social system into which a technical system must integrate. Most importantly, the approach distinguishes thinking in the real world from thinking in an abstract, systems world. See the representation of the seven stage model of SMM [2] in Figure 4.

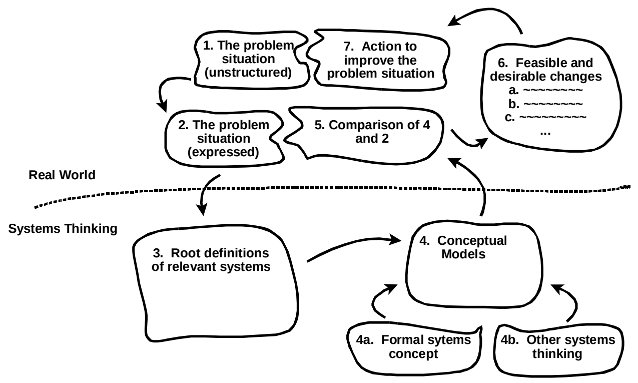
1. The problem situation (unstructured)
7. Action to improve the problem situation
6. Feasible and desirable changes
a. ~~~~~~~~
b. ~~~~~~~~
c. ~~~~~~~~~
...
2. The problem situation (expressed)
5. Comparison of 4 and 2
Real World
Systems Thinking
3. Root definitions of relevant systems
4. Conceptual Models
4a. Formal sytems concept
4b. Other systems thinking

Figure 4. Soft Systems Methodology [2].

There are different published versions of SSM as it has developed over the years. In this section, we give an overview of the older seven stage model [2] as we believe it is more accessible to the novice, with more specific activities in the stages and less generalized iteration than the newer four stage model [4]. In enacting SSM, one uses techniques relevant to the various stages, switches between the real world and the systems thinking world, and iterates where appropriate. We describe the overall process and each stage briefly below.

Stage one is the first of two stages that explore a problematic situation within the real-world frame of thinking. In stage two, the unstructured views of the problematic situation are given more structure and expression, in ways that stakeholders can understand them, that focus on the essence of the situation, and that identify relevant themes. Key aims are to develop a shared understanding of different perspectives and to create a basis for further discussion in later stages. Stage three represents a shift from real world thinking (about perspectives on what is, what is undesirable or desirable, and why) to system thinking (using systems concepts and system-thinking-inspired techniques). Stage three is about debating various viewpoints, defining what stakeholders want in a relevant system (as a solution to the problematic situation), and selecting one (or more) definition(s) for further consideration in stage 4. In stage four, ***conceptual models*** are constructed, based on agreed root definitions of the desired system. A conceptual model represents desired human activities. Stage 5 represents a shift back from system thinking to reconsider how the conceptual models developed in the system thinking world fit into the real world. Comparisons are made between conceptual models and the problem situation expressed. Stage six considers whether the identified areas for improvement and change determined in stage five can be accepted and integrated into the culture. Finally, stage seven should determine the scope of the action, who is to take action, what kinds of action should be taken, in which areas and when.

Following action taking, further iteration may be taken to assess the problematic situation and determine whether sufficient improvement has been made and why or why not. Continuing problems may be the motivation for another cycle (or episode) through the SSM process.

DSR and SSM are similar because they are both concerned with solving problems and are concerned with reasoned design of solutions. Both may involve evaluation of solutions generated and iteration back to earlier activities (or indeed the entire process) where weakness is identified and/or further improvement is needed. The main difference between SSM and DSR is that SSM is primarily concerned with solving particular situated problems for a particular client, while DSR is concerned with solving generalized problems for a generalized class of stakeholders. However, as identified above, SSM is very useful for addressing socio-technical problems. In particular, while both aim to be well grounded in problem understanding and reasoning about design, SSM provides key activities and techniques derived from system thinking for doing these better. In the next section, we consider how SSM might be adapted for use in Design Science Research.

2.5 DESIGN SCIENCE, ACTION RESEARCH, AND SOFT SYSTEMS

SSM is commonly applied in action research situations and has similarities with both action research and prototyping. It is complementary to both action research and DSR. Table 1 illustrates this by comparing and contrasting these three approaches. Given the complementary nature of SSM and DSR, a Soft Systems Methodology approach to DSR may be appropriate where the study subject includes research into the interaction between an artifact and a social system. This last point is critical. The advantage of SSM is its ability to study the artifact in relation to the social system into which the artifact is inserted and evaluated.

Possible relationships between DSR and SSM are not well explored. There has been some exploration of the notion of a soft form of DSR [1]. This work envisioned that a soft approach to DSR would (1) make strong use of behavioral theory, (2) treat problems and problem solving as complex and situated, (3) use interpretive methods for evaluation of artifacts, and (4) examine the entire design and evaluation process during evaluation [1]. Except for point (1) above, SSM naturally accommodates these characteristics. SSM is particularly strong in avoiding simplistic conceptualizations and understandings of problems. Its immersion in a situation supports soft versus hard evaluation approaches well, aligning well with interpretive versus positivist concepts in evaluation and avoiding reductionist (over-) simplifications.

Table 1. Comparison of characteristics of DSR, SSM, and AR.

Characteristic	DSR	SSM	AR
Orientation / Method for …	Research	Practice	Practice and Research
Goal	Problem Solving	Problem Solving	Problem Solving and/or Behavioral Understanding
Specificity	Generalized	Situation Specific	Situation Specific and Generalized
Design Role	Invention / Generative	Application or (Invention and Application)	Application or (Invention and Application)
Outcome	Design Theory or Artifact shown to have utility	Situated Organizational Improvement	Situated Organizational Improvement and (Behavioral Theory or Design Theory)

Having given an overview of the salient aspects of the literature on DSR, Action Research, and SSM, the next section will consider how SSM might inform the design of a soft methodology for DSR. The result is a proposal for an adaptation of SSM for the purposes of a general, socio-technical methodology for DSR (a "Soft DSR" methodology).

3. ADAPTING SSM INTO A SOFT DSR METHODOLOGY

Because of the similarities described above, adapting SSM to DSR may be straightforward. For example, the iterative representation of DSR in Figure 2 can easily be conceptualized as distinguishing between thinking in the real world, and thinking in the abstract systems world with the distinction placed after "The Design" and before "The Construction". Here, the search for the design solution and the evaluation of the design solution are activities that take place in the abstract world of design thinking. Artifact construction and its evaluation are activities that take place in the real world of the social system into which the artifact becomes situated.

Design thinking involves creativity rather than just analysis. It involves inspiration and the process of generating, developing, and testing ideas. Because design ideas must ultimately be executed in the very practical production of artifacts, successful design thinking incorporates systems thinking [20].

This Soft DSR methodology can be elaborated by expressing the SSM activities in the language of parallel DSR activities. See

Figure 5. This Soft Design Science Research methodology (Soft DSR) has seven activities.

1. A specific problem is identified and delineated.
2. This problem must then be expressed as a specific set of requirements
3. In the systems world, the requirements for the specific problem are systemically abstracted and translated into a general problem with both technical and social dimensions. Here the design thinking is about a class of problems rather than the specific problem owned by the client.
4. A general solution design (a class of solutions) for the general problem is derived through systems thinking and expressed in terms of general requirements. This activity involves a combination of design science techniques, such as the search for general components of the solution together with expression using imperative logic.
5. The general design requirements are compared with the specific problem for fit. In this activity the specific problem is re-articulated in terms of the general requirements and the imperative logic.
6. A declarative search is then made for the specific components that will provide a workable instance of a solution to the general requirements. The declarative search is made necessary by difficulties in operating imperative logic.
7. An instance of the specific solution is constructed and deployed into the social system. In this way, the specific problem is changed (hopefully improved), learning is derived, and the cycle begins again.

This cycle continues until the social and technical problems are resolved.

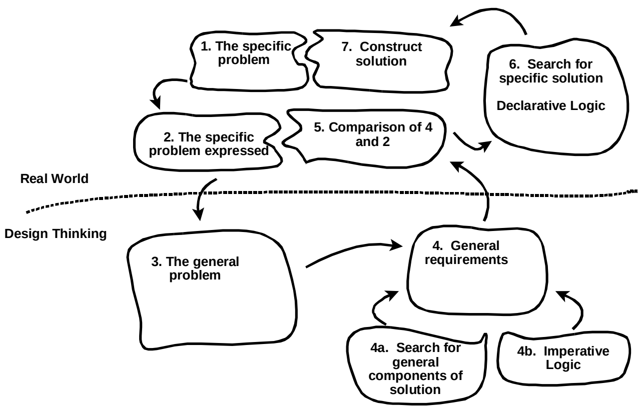

Figure 5. Soft Design Science Research.

This soft systems adaptation of DSR has elements that distinguish it from SSM and DSR. Among these elements are (1) the distinction between design thinking activities and real world activities; (2) clear distinction of the activities surrounding the expression of the general problem and general requirements; (3) the separation of imperative and declarative logic; and (4) the logical collocation of artifact construction and the specific problem to solve the client's specific problem. Design thinking, as a Soft DSR parallel to systems thinking, is different from systems thinking with its emphasis on creativity and inspiration. It is also different from the typical DSR empirical analysis that dominates real-world thinking about specific problems. Enrolling design thinking into Soft DSR helps clarify the distinct positioning of general problems and general requirements in a theoretical and abstract world. The strengths (ease of requirements expression) and limitations (operationalization difficulty) of imperative logic are also suitable to this abstract world. Unlike DSR and SSM, the switch from the abstract world back to the real world entails expressly switching logics. Switching over to declarative logic (ease of solutions expression) for the purposes of a solution search was Simon's solution. In Soft DSR, this logical switchover is expressed and defined more clearly than in either DSR or SSM. One further distinction is the iterative nature of this distinct DSR approach, viz., the collocating of the instance of a design solution with the originating instance of the problem.

General forms arise as abstractions from real-world instances [21]. The abstraction activities in Soft DSR differ distinctly from both SSM and DSR. In SSM, the abstract representation of the system is expressed through the process of defining root definitions and translating these into conceptual models. In DSR, the abstractions arise as design theorizing. In Soft DSR, the abstract processes are more distinctly formed into abstractions-to-general-forms and creative operations using these general forms. The two main general forms in Soft DSR are the generalized problem and generalized requirements. The creative operations include choosing how to abstract the general problem, and deciding which features of the general problem will be used as a basis for the general requirements.

Having defined a Soft DSR methodology that adapts SSM to DSR, the next section will provide a preliminary evaluation of the methodology by analyzing its hypothetical application to an extant DSR study.

4. SOFT DSR IN ACTION: AN ANALYTICAL EVALUATION

Before a proposed research methodology such as Soft DSR is given a "live" evaluation in field trials (i.e. naturalistic evaluation [10, 22]), evidence should be required to provide indications of its likelihood for developing a successful design outcome and significant research discoveries. A workable artificial evaluation [10, 22] to develop evidence prior to conducting live, naturalistic evaluation is to apply Soft DSR constructs in evaluating a completed (and published) DSR example. The purpose of this application is to evaluate how that research might have been improved (or at least done differently) with Soft DSR. Such an evaluation will provide prospective researchers with good indications of potential outcomes.

For this purpose, we selected a DSR project conducted with a large commercial bank that studied diffusion and adoption of IT [23]. This case was selected for several reasons. The details of the original study have been published elsewhere. We had access to background material, participants, and experience from the original case. This case used a design-oriented action research approach that makes it a candidate for consideration as improvement research, an early example of a design science type of study. Finally, as an action research case, it also enables us to distinguish how the design science orientation of Soft DSR would differ from (or improve) an action research study. The contrast between Soft DSR and action research also provides insights into the contrasts between action research and design science in general.

The case involves Danske Bank, a large European bank with more than 20,000 employees. The bank operated a large IT division, Danske Data, which develops applications for banking and insurance. Primarily applications are meant to run on a mainframe, but some applications are for Internet banking, client/server-environments, and PCs. Danske Data mainframe systems are up and running 24 hours a day, and every day 9 million transactions are carried out from 11.000 workstations. The developers typically have IT educations at a bachelor level, or come from a background in banking. Recently more and more employees with a master's degree have been hired.

For the purposes of evaluating the research in terms of Soft DSR, we will organize an evaluation of the events in the research according to the Soft DSR steps in Figure 5.

Soft DSR Step 1: The specific problem

From a combined Capability Maturity Model [24] and Bootstrap [25]assessment, it became clear that many products and processes developed by the IT department were *not* diffused and adopted as intended. The assessment report said: "Why are so many products and procedures well described but not used?" This specific problem understanding was identified and delineated and the need for a solution implied.

Evaluation of Step 1

This initial step is very common for both action research and design science projects. The basic problem situation is defined clearly in such pragmatic research approaches.

Soft DSR Step 2: The specific set of requirements

The assessment report suggested that Danske Bank should make a further analysis of diffusion and adoption, which was subsequently done. Insightful people from all parts of the organization, including project managers, line managers, and people from the methodology department, agreed to be part of a task force. In two workshops the task force diagnosed the diffusion and adoption problem. The cause of the inadequate diffusion and adoption was narrowed down to a number of issues. The problem to be solved was expressed as a specific set of requirements:

1. It is a requirement to ensure diffusion and adoption of products and processes from IT projects in Danske Bank.
2. It is a requirement that the individual project ensures that its new product will be used as expected when it has been completed

Evaluation of Step 2

This step was well thought through and carefully executed through the use of participative workshops. The specific requirements appear explicit and valid.

Soft DSR Step 3: The general problem

As the undertaking at Danske Bank was part of an externally funded research project called "Center for Software Process Improvement", there were numerous attempts to clarify and specify the general problem (cf. [23]). However, there was no formalized explicit translation of the specific requirements into a general problem statement about a class of problems. After studying the problem, the researchers consulted the literature on diffusion and adoption and found inspiration for a framework approach. The framework approach offers a process for organizational implementation of IT innovations (described below). This approach embodies the theory-driven nature of action research. This approach implies that the general problem with IT innovation diffusion was perceived to be disorganized and incomplete innovation implementation processes.

Evaluation of Step 3

Generalization of the problem was not explicitly addressed in the research project. It is an obvious assumption, similar to a Toulminian "warrant" that modifies the translation of the specific requirements to the general requirements [26]. A clearly and explicitly stated, generalized problem would have improved the logical basis for the solution, and might possibly have modified the requirements, if this general problem statement had been made explicit. More generalized statements (not specific to Danske Bank) would be:

1. It is a requirement to ensure diffusion and adoption of products and processes from IT projects in general.
2. It is a requirement that individual projects ensure that what they produce are used as expected.

Soft DSR Step 4a: Search for general components of a solution

The research project had an explicit and quite rigorous search process for available components of the general solution. This search process involved additional workshops and the development of a sub-task force. To answer the specific problems, the infrastructure of the project was changed from a 12-people task force to a small task force group of three (one being an author of this paper), each working half of their available time. The process unfolded as a second workshop one month later, which identified a number of solutions to the problems.

The sub-task force, while studying the organization's successes and failures in the first workshop, realized that attempts to ensure diffusion by adding some additional activities at the end of the project often fail. The group reasoned that starting such attempts early in the project would have an effect on the product itself. The major component for such a solution could be a framework to be used in a one-day workshop for projects focusing on diffusion and adoption right after the requirements to a given product had been defined.

The evidence that such a one-day workshop was promising was an earlier (unusual) successful diffusion effort. This effort included an analysis workshop to bring customers and developers together for agreement on scope and requirements. It also meant that projects within Danske Bank were familiar with the 1-day facilitated workshop concept.

Evaluation of Step 4a

The search process for components of the general solution is evident in the research reports. Quite a lot of time and effort was put into this specific search activity. Many alternative (candidate) components) were identified that ended up not being used as part of the specific solution. The components that were drawn into the solution include the workshop format, the framework, and the early timing for the workshop. This search process fits well with Soft DSR.

Soft DSR Step 4b: Imperative logic

As an action research project, it is not surprising that there was no explicit reference to imperative logic. The imperative logic notion is stronger in design science. However, we can readily express the solution components in imperative terms:

1. Hold an innovation diffusion planning meeting in a workshop format.
2. The workshop must be held early in the project process.
3. Follow a clear framework for the workshop.

Evaluation of Step 4b

While the general solution implies these three imperatives, there was no explicit imperative search for the best components of the solution. Given the problematic issues with imperative logic, Simon [27] describes conducting the search process by searching through declarative statements for solutions. For example, consider the contrasting declarations,

1. "An early workshop leads to IT diffusion and adoption."
2. "A late workshop leads to IT diffusion and adoption."
3. "A workshop anytime leads to IT diffusion and adoption."

Of these three declarations, the researchers explicitly noted that (2) would not lead to a workable solution. Logically, this fact implies that (3) does not hold. Therefore (1) is the only workable declaration and could be expressed as the second imperative in the solution expression.

There was also no consideration of the solution of the general, rather than the specific problem at Danske Bank. Consideration for generality in step 4 of Soft DSR might have guided these researchers to allow for organizational settings that are different than the one at hand. For example, it could be the case that in some other settings, declaration (2) or (3) might operate fine. If these declarations are to hold, then the imperative two would change to "The workshop must be held at the ideal time in the project process." The alternative meanings of "ideal" could be declared and searched in Step 6 (the specific solution search).

In this way, Soft DSR would have guided the researchers to further consideration of the degree to which the general requirements were unnecessarily specific to the immediate problem. Clearly, the specific solution is appropriate for Danske Bank, but the generality of the research results could have been improved by the Soft DSR methodology.

Soft DSR Step 4: The general requirements

As in Step 3, the research progressed toward its general solution without specifically expressing the general solution in terms of general requirements. Instead, the general solution is expressed in terms of remaining problems, which might substitute well for general requirements, and as a general solution that is easily translated into imperatives.

Evaluation of Step 4

Compared with the Soft DSR, the descriptions of the research make rapid progression from the specific problem description toward a specific solution. This is not necessarily inappropriate for action research, which can sometimes take on the nature of a prototype solution. In some settings, quickly changing anything can provide additional learning about the problem setting. It could be that the researchers were adopting a moderately fast-paced timing in this action research project, using due care in grounding their solution to collected data and reference literature, but avoiding unnecessary delays involved in structuring their abstract work. The use of Soft DSR would have drawn more research resources into consideration for expressing the general problem and the general solution.

Soft DSR Step 5: Comparison of general requirements to the specific problem

In most forms of action research, this comparison is left as an implication of action planning. As the action plan is constructed, gaps or mismatches will grow more obvious. The main comparison stage is made in evaluating the outcome of the implementation of the planned action (i.e. post hoc). The research study at hand is not particularly different. Although it may not have been explicit, it is likely that the researchers were constantly comparing the problem and the general nature of the proposed solution as a routine part of the continued development of the solution.

Evaluation of Step 5

If considered from a Soft DSR perspective, the terms of the specific problem, and the terms of the general requirements surface as comparable. See Table 2.

Table 2. Comparison of the specific problem and the general requirements.

Specific Problem from Step 2	General Requirements from Step 4
1. It is a requirement to ensure diffusion and adoption of products and processes from IT projects in Danske Bank. 2. It is a requirement that the individual project ensures that its new product will be used as expected when it has been completed.	1. Hold an innovation diffusion planning meeting in a workshop format. 2. The workshop must be held early in the project process. 3. Follow a clear framework for the workshop (specifics to be determined).

Specific problem 1 (ensure diffusion) and general requirement 1 (diffusion workshop) do align the general solution with the specific problem. Likewise specific problem 2 (project products) and general requirement 2 (early project timing) also align a general solution with the specific problem. The unmatched third requirement seems sensible as a measure to ensure the workability of the other two requirements.

Soft DSR Step 6: Declarative search for specific solution

Concurrently with studying cases from Danske Bank, literature on diffusion and adoption of IT was brought in, and different authors who had analyzed various elements of the issue inspired the specific Danske Bank solution to the general problem above. The specific focus is on the exact framework for the workshops. The specific solution we designed had three phases called analysis, design, and planning, each covering two stages as shown in Figure 6. The details for each of these phases and stages, including the reasons why these were adopted as components of the solution, are included in the appendix at the end of this paper.

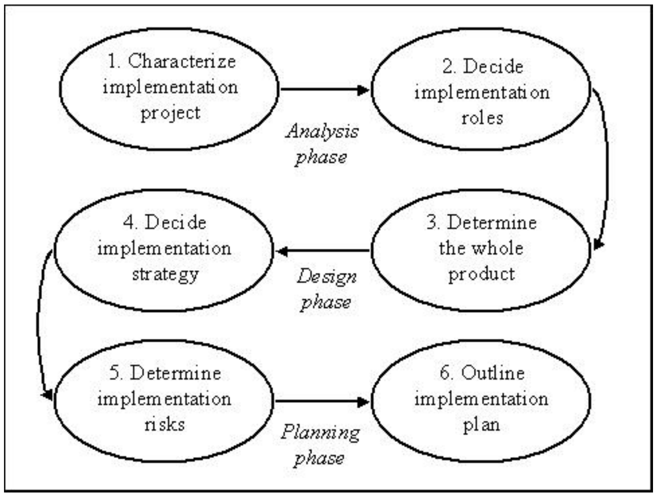
1. Characterize implementation project
2. Decide implementation roles
Analysis phase
4. Decide implementation strategy
3. Determine the whole product
Design phase
5. Determine implementation risks
6. Outline implementation plan
Planning phase

Figure 6. The example specific solution.

Evaluation of Step 6

The search process for components of the specific solution is mentioned in the research reports, but not well defined. The overall process included early workshops and later library work and inspiration by the smaller research team. The components of the specific solution appear to have arisen mostly from this latter, unstructured part of the study. We learned from the study reports why the components were selected and how these should operate, but it appears that there was little attention given to exactly how the literature search and selection process were conducted. For example, it is not clear in the reports what components were considered and rejected, nor what the criteria were for the selection process. Following Soft DSR methodology would have led the researchers to consider the search processes more carefully, and pay closer attention to the component selection criteria. It is likely that a broader search could have produced more workable components and these components might have been more carefully described and expressed. The solution would likely have been improved both by this attention to detail and the development of more diversity.

Because of the action research framework, rather than a Soft DSR approach, it is difficult to distinguish the development of the specific solution from the development of the general solution. To a certain degree, the only clear distinction is that the specific solution, the framework, is specified in more detail. Indeed, it is hard to say whether the more detailed process framework developed above is general (applicable broadly, not just in Danske Bank) or tuned specifically in Danske Bank. Consideration of the details of how to “Follow a clear framework for the workshop” (general requirement 3 from step 2) might have been more appropriately considered at stage 4, then matched to the specific requirements in step 5.

Neither imperative logic (in the general requirements) nor declarative logic (in the specific solution) is expressed in the research reports. We can translate the proposed framework components in Figure 6 into seven imperative statements:

1. Have three phases in the workshop: analysis, design, and planning
2. In the analysis phase, characterize the implementation project
3. In the analysis phase, decide implementation roles
4. In the design phase, decide implementation strategy
5. In the design phase, determine the whole product
6. In the planning phase, determine implementation risks
7. In the planning phase, outline implementation plan

These seem to align to the imperative statement 3 arising out of step 4 (as noted above) in that they are fairly clear and taken together comprise a framework for the innovation diffusion planning workshop. However, the seamless progression from the general to the specific solutions shown is not necessarily optimal. Consideration of alternative declarative statements might have lead to alternative steps for the detailed framework. For example, imperative four (decide implementation strategy), could be expressed declaratively as,

1. “Deciding implementation strategy in the design phase leads to IT diffusion and adoption.”

If the researchers had been following Soft DSR, then other values for this declaration should have been explicitly searched, such as,

2. “Deciding implementation strategy in the analysis phase leads to IT diffusion and adoption.”
3. `“Deciding implementation strategy in the planning phase leads to IT diffusion and adoption.”

It is likely that the researchers gave consideration to the idea represented by these alternative declarations, but the importance of the search process in Soft DSR would have led them to a more formal and more careful process of considering alternative declarations. As a result, a more diverse set of alternative solutions might have been more clearly considered by the researchers and possibly other design decisions made abou the workshop organization framework.

Soft DSR Step 7: Construct solution

The researchers clearly designed the specific solution (which could be interpreted as an instance of a general solution) and deployed it in bank projects. Since this was a design-oriented action research project, the solution was improved over five iterations. The initial deployment in a real Danske Bank project resulted in a number of adjustments, and there were further adjustments after a second iteration in two other projects workshops. After the third iteration, the framework and the workshop had evolved into a form that was acceptable for internal projects. A fourth and a fifth iteration were carried out a year later to adapt the framework to external projects – meaning projects involving customers from the Bank or outside the Bank.

In this way, the specific problem was changed and improved four times, illustrating the cyclical, iterative nature of design science.

Evaluation of Step 7

Both action research and Soft DSR are iterative approaches to research. Each involves a number of phases. The research report is not clear about the nature of the adjustments that proceeded from the iterations. Indeed, it is possible that aspects of the reported solution are those that resulted from the refinements of several iterations, rather than results of an initial solution search. In terms of reporting length, the researchers are probably reporting the final framework, and not the initial one. However, the report does suggest that adjustments were directed to the specific solution, and it is not clear that the general problem or general requirements were revisited. It could be obvious that the general solution was working, and the extra effort to reexamine the abstractions was simply unnecessary. In other words, the iterations may have simply been through Steps 1, 6 and 7, instead of Steps 1–7. Soft DSR might have improved the research process by adding emphasis to the need to constantly re-express the specific problem after the need for adjustments arose, and to reevaluate the general problem, its requirements, and its relationship to the specific problems.

CONCLUSION

The review of a published DSR study, which used a design-oriented action research case, developed a number of key insights about the similarities and differences between the two approaches. With a focus on possible improvements in the research study that might have proceeded from the use of Soft DSR, we have developed evidence of its probable success as a methodology in a future research study. The analysis has identified similarities, ambiguous differences, probable improvements, and possible disadvantages.

Similarities. The design-oriented action research study examined in this paper shares some clear similarities to Soft DSR. Consistent with the Soft Design Science ideals, the action research project seemed to approach problem definition with an aim to clarity and specificity. Explicit, valid and specific requirements are an ideal in both approaches. The design-oriented action research DSR project also engaged in a very prominent and documented solution search process. There was also a good match between the specific problem and the general requirements generated in the action research study. This match is also an ideal for Soft DSR. The iterative nature of the action research project was also appropriate for Soft DSR.

Ambiguous Differences. The action research study examined in this paper is also different from Soft DSR in several ways. However, it is not clear that all differences are improvements. For example, the design-oriented action research study did not clearly distinguish the various stages of solution development, as would have been expected with Soft DSR. As a result, the distinction between the general requirements and the specific solution could have been clearer. While this is a difference between Soft DSR and the project studied, it may be that this quest for a seamless, specific solution, without identifying a general problem and solution, would be appropriate in some research settings (e.g., where time is of the essence) , and other settings (e.g., where an ideal, high quality, or optimal solution is critical) might benefit from the distinctions.

Probable Improvements. Some differences would very likely have improved on the study outcomes if Soft DSR had informed the research approach. In shifting to design thinking, an explicit investigation into the general problem was missing from the action research study. Soft Design Science would involve more careful explication of the general problem, and this additional detail could have changed or improved the direction of the research. Also, Soft Design Science would have promoted more structure in the search process for both the general requirements and specific solutions by expressly using imperative statements and declarative logic. Particularly at the general requirements level, it is quite possible that the researchers may have gained clearer insights into the generality of the solution addressed by the requirements. At the specific search level, the added structure may have enabled the research to discover more alternative components for the specific solution. Soft DSR might also have improved the action research project by constantly reviewing the specific and general representations of the problem and solution spaces. This comparison could detect shifts in underlying constructs that might otherwise have been missed in the action research project.

Possible Disadvantages. A third type of difference offers insight into how the use of Soft DSR to inform the action research study might indeed have degraded its processes and outcomes. It is clear from the Soft DSR evaluation of the action research study that there are more structures incorporated in the more elaborate methodology. While offering rigor, this additional structure could also weigh down researchers with additional complexity in their tasks.

This paper has proposed a Soft DSR Methodology, which incorporates aspects of Soft Systems Methodology into a DSR process. The proposed methodology has some similarities to design-oriented action research. The paper analyses a hypothetical application of Soft DSR to a published DSR study that used design-oriented action research. The analysis identifies similarities and differences, including probable improvements and some possible disadvantages.

Overall the potential improvements enumerated above provide a positive indication of the potential value of using Soft DSR approaches. On the basis of the evidence in this study, future research is warranted for developing and using Soft DSR methodology as an approach to design science research. Research applying Soft DSR in a fresh setting with a fresh research goal would provide a naturalistic evaluation to complement the artificial analysis presented in this paper. The evidence in this paper also suggests that future research into DSR methodology could provide alternative approaches to enable more careful and complete DSR.

APPENDIX: WORKSHOP FRAMEWORK DETAILS

This section provides details from the case study.

The analysis component has two stages. First, we focused on a range of questions in order to characterize the project. The purpose was to create a common understanding of the product that was going to be the outcome of a given IT project. Here our search resulted in a set of questions from [28]. Second, we focused on who was playing which roles in the diffusion and adoption of the project result. Here we designed a *role model* inspired by CATWOE [4] and [29]. The main idea of the role model we designed is that five roles must be filled if implementation is going to succeed. If one of the roles has not been filled, implementation will fail. The five roles were:

1. The target group are the persons who are going to use the product
2. The owner/sponsor is responsible for initiating the implementation and scoping the direction. Towards the end of the implementation the owner /sponsor also is responsible for demanding the results coming out of the implementation.
3. The manager of implementation is the person doing the actual implementation work. Often this role is named the project manager.
4. The champion/ambassador are the persons who actually makes the people from the target group take the innovation into use.
5. “Other secondary stakeholders” consists of all other interested parties not taking any of the four primary roles.

The design component also consists of two stages. First, we try to define “the whole product”. The idea behind “the whole product” is that the user of a new product normally expects to get more than a mere technical solution. Therefore, we are working with three product levels [30]:

1. The core product
2. The whole product
3. The expanded product

The core product is the developers’ idea of what they have to prepare in order to give the customers what they promised. For example, a developer who is going to develop a HomeBanking system may regard the computer program as the core product.

The whole product is the customer’s idea of what he/she will get - i.e. the core product including related supplementary products to ensure that the product is easy to use. I.e. a customer who is buying a HomeBanking system will expect to get the disks with the program in addition to a quick guide, hot-line telephone support, and a tutorial.

The expanded product is not relevant until the whole product has been put into operation and the customer or the developer suggests supplementing the existing product with an additional service. For example, the customer might wish that the HomeBanking system also included a tax calculation feature.

Second, we examined how to organize the implementation of the product. The purpose is partly to clarify what kind of product we have to make in order to improve the chances of success in implementation. And partly to create a framework for an implementation plan that can be used in the subsequent phase. For this purpose we have used a theory on implementation strategies developed by Ken Eason [31]. The theory comprises five different strategies from the most revolutionary strategy – where the users abandon their old system one day and adopt the new system on the following day – to the most evolutionary approach where the users over a period of months or years, little by little, start using more and more of the new system. Eason’s five strategies are called:

1. Big Bang
2. Parallel application
3. Phased introduction
4. Experimental diffusion
5. User-based experiment

Similar to the two previous phases, this phase also has two stages: a risk analysis and an implementation planning stage. In the first stage of the planning phase, we perform a risk analysis in order to ensure that we haven’t overlooked any potential problems. Risk management deals with identifying and reacting to potential problems in timely manner [32]. Risk management often leads to proactive activities. In our risk analysis we focus on the following six issues:

1. Identify risks. Here we use a standard list of the ten highest risks in the organization based on the organization’s own experience.
2. Assess the probability for every risk on a scale from one to five.
3. Assess the consequence of every risk on a scale from one to five.
4. Make a list of priorities by multiplying probability with consequence.
5. Find activities – proactive or supportive – that relate to the three most important risks.
6. List the chosen activities as items that should be included in the implementation plan.

The second stage is to outline an implementation plan. During the workshop all activities mentioned by the project group are being written down on small notes. These notes – which are appropriate implementation activities – are now placed on a blackboard or a table where the benchmarks and the intermediate states are outlined.

The last stage of the workshop provides the project group with a good outline of the implementation plan. It may be applied directly in connection with estimating the implementation phase and as input to the project plan and the final implementation plan of the project.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their useful feedback, which enabled us to improve this paper.

REFERENCES

[1] Baskerville, R., Pries-Heje, J. and Venable, J. *Soft Design Science Research: Extending the Boundaries of Evaluation in Design Science Research*. Claremont Graduate University, Pasadena, 2007.

[2] Checkland, P. *Systems Thinking, Systems Practice*. J. Wiley, Chichester, 1981.

[3] Checkland, P. and Holwell, S. *Information, Systems and Information Systems: Making Sense of The Field*. John Wiley, Chichester, 1998.

[4] Checkland, P. and Scholes, J. *Soft Systems Methodology in Practice*. J. Wiley, Chichester, 1990.

[5] Iivari, J. A paradigmatic analysis of Information Systems as a design science. *Scandinavian Journal of Information Systems*, 19, 2 (2007), 39-63.

[6] March, S. T. and Smith, G. F. Design and natural science research on information technology. *Decision Support Systems*, 15, 4 (Dec 1995), 251-266.

[7] Gregor, S. and Jones, D. The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, 8, 5 (2007), 312-335.

[8] Walls, J. G., Widmeyer, G. R. and El Sawy, O. A. Building an information system design theory for vigilant EIS. *Information Systems Research*, 3, 1 (1992), 36-59.

[9] Hevner, A. R., March, S. T., Park, J. and Ram, S. Design Science In Information Systems Research. *MIS Quarterly*, 28, 1 (Mar 2004), 75-105.

[10] Pries-Heje, J., Venable, J. and Baskerville, R. Strategies for Design Science Research Evaluation. In *Proceedings of the 16th European Conference on Information Systems (ECIS 2008)* (Galway, Ireland, 9-11 June 2008). National University of Ireland.

[11] Mason, R. and Carey, T. Prototyping interactive information systems,. *Communications of The ACM 26*(1983), 347-354.

[12] Ehn, P. *Work-Oriented Design of Computer Artifacts*. Arbetslivscentrum, Stockholm, 1988.

[13] Baskerville, R. and Stage, J. *Prototyping*. Berkshire Publishing Group, Great Barrington, MA, 2004.

[14] Porra, J. Colonial systems. *Information Systems Research*, 10, 1 (Mar 1999 1999), 38-70.

[15] Baskerville, R. and Wood-Harper, A. T. Diversity in Information Systems Action Research Methods. *European Journal of Information Systems*, 7, 2 (1998), 90-107.

[16] Järvinen, P. Action Research is Similar to Design Science *Quality and Quantity* 41, 1 (2007), 37-54.

[17] Cole, R., Purao, S., Rossi, M. and Sein, M. K. *Being Proactive: Where Action Research Meets Design Research*. Association for Information Systems, Las Vegas, Nevada, USA, 2005.

[18] Sein, M. K., Henfridsson, O., Purao, S., Rossi, M. and Lindgren, R. *Action Design Research: A Methodology For Design Of Artifacts In Context*, Kristiansand, Norway, 2009.

[19] Iivari, J. and Venable, J. Action Research and Design Science Research – Seemingly similar but decisively dissimilar. In *Proceedings of the 2009 European Conference on Information Systems (ECIS 2009)* (Verona, Italy, 8-10 June, 2009).

[20] Brown, T. Design Thinking. *Harvard Business Review*, 86, 6 (2008), 84-93.

[21] Lee, A. S. and Baskerville, R. L. Generalizing Generalizability In Information Systems Research. *Information Systems Research*, 14, 3 (2003), 221-243.

[22] Venable, J. A Framework for Design Science Research Activities. In *Proceedings of the 2006 Information Resource Management Association Conference* (Washington, DC, USA, 2006).

[23] Pries-Heje, J. and Tryde, S. Diffusion and adoption of IT products and processes in a Danish Bank. In *Proceedings of the Diffusing Software product and Process Innovations. IFIP TC8 WG 8.6 Fourth Working Conference* (Banff, Canada, April, 2001).

[24] Paulk, M. C., Weber, C., Curtis, B. and Chrissis, M. B. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, Reading, Mass, 1995.

[25] Kuvaja, P., Similä, J., Krzanik, l., Bicego, A., Saukkonen, S. and Koch, G. *Software Process Assessment & Improvement. The BOOTSTRAP Approach*. Blackwell Publishing, Oxford, UK, 1994.

[26] Toulmin, S. *The Uses of Argument*. Cambridge University Press, Cambridge, 1958.

[27] Simon, H. A. *The Science of the Artificial*. MIT Press, Cambridge, Mass., 1996.

[28] Mathiassen, L. and Sørensen, C. *A Guide to Manage New Software Engineering Tools*. Chapman & Hall, London, 1997.

[29] Bendix, J. and Andersen, O. S. *Forandringsledelse - Kommunikation, Adfææerd og Samarbejde ("Change Mangement - Communication, Behaviour and Cooperation")*. Børsens Forlag, Copenhagen, 1995.

[30] Moore, G. *Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers*. Capstone, Mankato, 1991.

[31] Eason, K. *Information Technology and Organisational Change*. Taylor & Francis, London, 1988.

[32] Iversen, J. H., Mathiassen, L. and Nielsen, P. A. Managing Risk in Software Process Improvement: An Action Research Approach. *MIS Quarterly*, 28, 3 (2004), 395-433.