Choosing a Design Science Research Methodology

John R. Venable
School of Information Systems
Curtin University
Perth, WA, Australia
Email: j.venable@curtin.edu.au

Jan Pries-Heje
User-Driven IT Innovation
Roskilde University
Roskilde, Denmark
Email: jph@ruc.dk

Richard Baskerville
Department of Computer Information Systems
Georgia State University
Atlanta, GA, USA
School of Information Systems
Curtin University
Perth, WA, Australia
Email: baskerville@acm.org

Abstract

Design Science Research (DSR) is a popular new research approach and paradigm, for which a number of research methodologies have been developed. One of the challenges facing researchers wanting to apply this new approach is the choice of research methodology. In this paper we give an account of six DSR methodologies and we compare them using a Design Science Research Methodology Comparison Framework that we adapted from an existing Information Systems Development Methodology Comparison Framework. Based on the outcomes of the comparison, we develop a set of technological rules that forms a contingency-based framework to support Design Science Researchers in choosing an appropriate and well-suited DSR methodology, depending on the contingencies of the situation at hand.

Keywords Design Science Research (DSR), Design Science Research Methodology, Research Design, Methodology Comparison, Design Theory, Technological Rule

1 Introduction

If you cook food for the first time, it is a good idea to have some guidelines or even a cookbook to provide you with a process or method to do it. For example, the “New Nordic Cuisine” provides an underlying philosophy in the form of 10 guidelines to achieve pureness, ethicality, sustainability, health and quality (Byrkjeflot, Pedersen, & Svejenova, 2013; Välimäki, Sørensen, & Dahlgren, 2004). The New Nordic Cuisine Manifesto provides a *Methodology* encompassing a view on the world, a paradigm and some principles but not *Methods* for cooking specific food. But, the New Nordic Cuisine is not alone; there are other schools of thinking – or methodologies – like The French Kitchen, The Japanese Kitchen, and so on.

While Social and Behavioural Sciences seek to understand reality, Design Science Research (DSR) seeks to invent (design) new means for acting in the world in order to change and improve reality. As a result, DSR re-creates reality through creating and evaluating artefacts that serve human purposes and solve human problems (March & Smith, 1995; Simon, 1996).

Like the cooking example, if you are trying to do Design Science Research, it is a good idea to have some guidelines or a cookbook. Just as there are different cuisines or schools of cooking, there are competing methodologies for conducting DSR, based on different worldviews and each with a different set of recommendations. As a new design science researcher, you face a choice of DSR “cookbooks” and methodologies. But, which one should you choose and use? In what situations and based on what contingencies?

This paper aims to answer the research question, “How can a DSR researcher choose an appropriate DSR methodology, well suited to a particular DSR project?” The remainder of the paper is organized as follows. First we give an account of the most commonly used DSR methodologies. Next we choose a Nexus-approach as our research method. We then compare and contrast DSR methodologies and develop a contingency-based framework that can help Design Science Researchers choosing methodology dependent on the situation at hand. Finally we discuss and conclude the paper.

2 A History of DSR Methodologies

The field of information systems has a long history and strong emphasis on research methods and paradigms. The early history of IS saw much interest and activity in research that developed new technology. Early empirical research, such as the Minnesota experiments (Dickson, Senn, & Chervany, 1977), evaluated the effectiveness of such technologies and provided guidelines for their selection and use. Gradually, empirical research, especially positivist research, gained more emphasis. Research method and paradigm discussions established that interpretive research has a significant place in IS research. Further debate established that critical research also is highly relevant to information systems. Such debate and emphasis gradually came to devalue and de-emphasise research that developed new purposeful artefacts (i.e. Design Science Research), such as IS and ICT, but also methodologies, etc.

In reaction to this de-emphasis and to defend DSR as a legitimate research approach, suitable for conducting research worthy of publication in the field of IS, various researchers using such an approach began publishing papers concerning DSR as a research method and paradigm (e.g. Hevner, March, Park, & Ram, 2004; March & Smith, 1995; Nunamaker, Chen, & Purdin, 1990; Walls, Widmeyer, & El Sawy, 1992). The rest of this section identifies and briefly introduces six papers proposing DSR research methodologies.

2.1 Systems Development Research Methodology (SDRM)

The first significant paper in this genre was by Nunamaker, Chen, and Purdin (1990), who propose “A Systems Development Research Methodology” (which we will abbreviate as “SDRM” here. SDRM includes a five-step research process with relevant research issues at each step, as shown in table 1 below.

Research Step	Research Issues
1. Construct a Conceptual Framework	• State a meaningful research question • Investigate the system functionalities and requirements • Understand the system building processes/procedures

2. Develop a System Architecture	• Develop a unique architecture design for extensibility, modularity, etc. • Define functionalities of system components and interrelationships among them
3. Analyze & Design the System	• Design the database/knowledge base schema and processes to carry out system functions • Develop alternative solutions and choose one solution
4. Build the (Prototype) System	• Learn about the concepts, framework, and design through the system building process • Gain insight about the problems and the complexity of the system
5. Observe & Evaluate the System	• Observe the use of the system by case studies and field studies • Evaluate the system by laboratory experiments of field experiments • Develop new theories/models based on the observation and experimentation of the system's usage • Consolidate experiences learned

Table 1: System Development Research Methodology Approach & Research Issues (Nunamaker et al., 1990)

While the SDRM research method is essentially linear in nature, researchers following the method are able to cycle back to an earlier step at any point in the process.

2.2 DSR Process Model (DSRPM)

Vaishnavi and Kuechler (Vaishnavi & Kuechler, 2004, 2007, 2015) propose a different five-step process, the DSR Process Model (which we will abbreviate here as DSRPM) as shown below in table 2.

Research Step	**Cognitive Processes**	**Outputs**	**Knowledge Flows**
1. Awareness of Problem		Proposal	Knowledge contribution (KC) (input)
2. Suggestion	Abduction	Tentative design	-
3. Development	Deduction	Artifact	Circumscription (to KC)
4. Evaluation	Deduction	Performance measurement	Circumscription (to KC)
5. Conclusion	Reflection, abstraction	Results	Design science knowledge (to KC)

Table 2: DSR Process Model Steps, Cognitive Processes, Outputs, and Flows (Vaishnavi & Kuechler, 2004, 2007, 2015)

Like in SDRM (Nunamaker et al., 1990), DSRPM allows for cycling back to earlier steps. However, these expressly include knowledge flows, whether circumscription (constraint knowledge limiting theories identified through building or evaluating the artefact) or design science knowledge, both of which provide a knowledge contribution, which is to the outside world or to the next cycle of research (awareness of a new problem).

2.3 Design Science Research Methodology (DSRM)

Peffers et al. (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2008) propose a six-step Design Science Research Methodology (which they abbreviate as "DSRM") as shown in table 3.

Research Step	**Concerns**	**Output to Next Step**	**Entry Point?**
1. Identify Problem & Motivate	• Define problem • Show importance	Inference	Problem-Centered Initiation
2. Define Objectives of a Solution	• What would a better artefact accomplish?	Theory	Objective-Centered Initiation
3. Design & Development	• Artifact	How-to Knowledge	Design & Development Centered Initiation
4. Demonstration	• Find suitable context • Use artefact to solve problem	Metrics, Analysis Knowledge	Client/Context Initiated
5. Evaluation	• Observe how effective, efficient • Iterate back to design	Disciplinary Knowledge	
6. Communication	• Scholarly publications		

	• Professional publications		

Table 3: Design Science Research Methodology (DSRM) (Peffers et al., 2008)

Like SDRM and DSRPM, DSRM allows cycling back to earlier activities, in particular from (5) Evaluation or (6) Communication back to (2) Define Objectives or (3) Design & Development, depending on the reason for cycling back.

2.4 Action Design Research (ADR)

The Action Design Research (ADR) Method proposed by Sein et al. (Sein, Henfridsson, Purao, Rossi, & Lindgren, 2011) is a methodology specifically combining Action Research (AR) and DSR. ADR includes 4 stages with associated activities and principles as shown in table 4.

Research Stage	**Activities**	**Principles**
1. Problem Formulation	1. Identify and conceptualize the research opportunity 2. Formulate initial research questions 3. Cast the problem as an instance of a class of problems 4. Identify contributing theoretical bases and prior technology advances 5. Secure long-term organizational commitment 6. Set up roles and responsibilities	1. Practice-Inspired Research 2. Theory-Ingrained Artifact
2, Building, Intervention, and Evaluation (BIE)	1. Discover initial knowledge-creation target 2. Select or customize BIE form 3. Execute BIE cycle(s) 4. Access need for additional cycles, repeat	3. Reciprocal Shaping 4. Mutually Influential Roles 5. Authentic and Concurrent Evaluation
3. Reflection and Learning	1. Reflect on the design and redesign during the project 2. Evaluate adherence to principles 3. Analyze intervention results according to stated goals	6. Guided Emergence
4, Formalization of Learning	1. Abstract the learning into concepts for a class of field problems 2. Share outcomes and assessment with practitioners 3. Articulate outcomes as design principles 4. Articulate learning in light of theories selected 5. Formalize results for dissemination	7. Generalized Outcomes

Table 4: Action Design Science (ADR) Stages, Activities, and Principles (Sein et al., 2011)

In ADR, there is an inherent cycle within the Building, Intervention, and Evaluation (BIE) Stage as artefact are constructed, put into action with the AR client, and evaluated in action. Moreover, transitions to Reflection and Learning (stage 3) and back can be made at any time from stages 1 and 2. The learning during the BIE cycle (stage 2) may also lead to cycling back to Problem Formulation (stage 1). Finally, the transition to Formalization of Learning (stage 4) only occurs when the problem is evaluated as sufficiently resolved by the research to develop and use the artefact.

2.5 Soft Design Science Methodology (SDSM)

Soft Design Science Methodology (SDSM) (Baskerville, Pries-Heje, & Venable, 2007; Baskerville, Pries-Heje, & Venable, 2009; Pries-Heje, Venable, & Baskerville, 2014) was inspired by Soft Systems Methodology (SSM) (Checkland, 1981; Checkland & Holwell, 1999; Checkland & Scholes, 1990, 1999) as a way to deal with issues in formulating problems and evaluating solutions. Table 5 shows the eight activities of SDSM and concerns relevant at those stages. Like ADR, SDSM works with a client to solve a specific problem, but generalises on the problem along the way.

Research Activity	**Real World or Design Thinking World?**
1. Learn about specific problem	Real
2. Inspire and create the general problem and general requirements	Design Thinking
3. Intuit and abduce the general solution	Design Thinking
4. Ex Ante Evaluation (General)	Design Thinking
5. Design specific solution for specific problem	Design Thinking

6. Ex Ante Evaluation (Specific)	Real
7. Construct specific solution	Real
8. Ex Post Evaluation	Real

Table 5: Soft Design Science Methodology (SDSM) Activities (Pries-Heje et al., 2014)

Like SSM and the other DSR methodologies above, SDSM includes a cycle over the entire process. Unlike SSM or the other processes above, SDSM includes three evaluation cycles. General ex ante evaluation (stage 4) concerns whether the general design matches the general requirements. Specific ex ante evaluation (stage 6) evaluates the specific design (of the general design) prior to its instantiation and introduction to the client's situation. Ex ante evaluation (after instantiation) evaluates the utility of the artefact in actual use by the client. These three cycles (particularly the ex ante cycles) provide flexibility to learn about issues and overcome them before they become problems for the client.

2.6 Participatory Action Design Research (PADR)

Bilandzic and Venable (2011) proposed the five-stage Participatory Action Design Research (PADR) methodology for developing solutions to problems held by heterogeneous groups of stakeholders, particularly in the context of urban informatics systems for use in public spaces. Like ADR, PADR aims to combine AR and DSR, but for a large group or the public in general, not for a single client organisation. Table 6 shows the five stages and the activities within each stage.

Research Stage	**Activities**
1. Diagnosing and Problem Formulation	Participative problem setting Ethnographic Study
2. Action Planning	Opportunity identification Participative planning
3. Action Taking: Design	Participative design Prototyping and installation Usability evaluation
4. Impact Evaluation	Ethnographic study Participative evaluation
5. Reflection and Learning	Participative client learning Design theorising for Urban Informatics (UI)

Table 6: Participatory Action Design Research (PADR) Stages and Activities (Bilandzic & Venable, 2011)

Like other DSR methodologies, PADR supports cycling back. In particular, cycles within Action Taking: Design (stage 3) are supported for cycling from the usability evaluation (a form of ex ante or formative evaluation) back to participative design and/or prototyping and installation. Cycles are also supported from participative evaluation (in stage 4) to action planning (stage 2) and from participative client learning (in stage 5) to diagnosing and problem formulation (stage 1).

This section provides an introduction to the history and evolution of DSR and the development of six DSR methodologies from which a DSR researcher might choose. DSR methodologies have evolved from a very IT-centric artefact design without client involvement to practices combining AR and DSR with a high level of client engagement. However, the literature provides no guidance to answer the question: How should one decide which DSR methodology to use (if any)? The next section describes the research methodology used to derive an answer to that question.

3 Research Method

One way in which design science differs from social or natural science is its stronger dependence on functional explanations grounded in the relationship between functional requirements and the prescriptive components of the design. In this paper our research question is "How can a DSR researcher choose which DSR methodology he/she should apply and use?" To answer that question we first surveyed existing literature and findings for different DSR methodology approaches. Second, we decided to compare the methodologies using a framework. We decided to apply a framework by Avison & Fitzgerald (2006) originally aimed at comparing Information Systems Development (ISD) methodologies. We did, however, not apply it as-is, but adapted it to the specifics of DSR.

Following the overview created by applying the framework – we decided to try to identify and specify technological rules to answer the research question.

Technological rules are one acknowledged form of design theory (Bunge, 1967; J.E. Van Aken, 2004). Rules prescribe a form of practical action. For example, one contingency (design) theory within management was formalized as technological rules, expressing a decision design as: “A technological rule follows the logic of ‘if you want to achieve Y in situation Z, then perform action X’. The core of the rule is this X, a general solution concept for a type of field problem.” (Joan Ernst van Aken, 2005, p. 23). The “Z” in these technological rules embodies the contingencies.

One challenge facing us when using technological rules is that they need grounding: “Research that intends to ground a technological proposition to explain why and how it produces certain outcomes will typically have to draw on survey-based field studies” (J. Van Aken & Romme, 2009, p. 9). Our grounding in this paper will be the survey and comparison of methodologies using an adapted framework. We use this grounding to identify and specify a set of technological rules that can help design science researchers choose an appropriate DSR methodology. A natural follow-up study to this paper would therefore be an exploratory study of what methodologies are used and why? But that is beyond this paper.

Right now our research method includes a formative evaluation. Meaning that we have developed a set of technological rules that we then have formatively evaluated among ourselves using different examples of DSR projects (of our own). Thus the set of technological rules that we put forward is in itself a contribution we now put forward to you as a reader of this paper.

4 Comparing Methodologies

This section compares the six DSR methodologies introduced above using an adapted version of the Avison and Fitzgerald (2006) framework for comparing Information Systems Development (ISD) methodologies. First we present the existing framework and then how it can be adapted to create a new DSR Methodology Comparison Framework. Finally we apply the new DSR Methodology Comparison Framework to the six chosen DSR methodologies.

4.1 The Avison and Fitzgerald ISD Methodology Comparison Framework

Avison and Fitzgerald (2006) provide a framework for comparing information systems development methodologies. Their ISD Methodology Comparison Framework comprises seven elements, two with sub-elements. Table 7 below shows these seven elements and their sub-elements, together with a brief explanation.

Framework Element or Sub-Element	**Description**
1. Philosophy	What is the weltanschauung or essence of its approach? Four sub-elements below
a. Paradigm	Science vs Systems, Objectivist vs Subjectivist Ontology, Positivist vs Interpretive Epistemology
b. Objectives	What is the goal of the methodology?
c. Domain	Classes of situations where relevant. Narrow problem focus or Broadly systemic?
d. Target	Targeted to specific kinds of situations or general purpose?
2. Model	What is the basic abstraction and representation mechanism used? (1) verbal, (2) analytic or mathematical, (3) iconic, pictorial, or schematic, and (4) simulation
3. Techniques & Tools	What tools and techniques are used in the methodology?
4. Scope	What stages/activities of the systems development life cycle are covered?
5. Outputs	What are the deliverables at each stage and at the end?
6. Practice	Three sub-elements below
a. Background	Commercial or Academic?
b. User Base	Numbers and types of methodology users
c. Participants	What roles participate and what skills are needed?
7. Product	What do methodology purchasers get for their money? Software? Documentation? Training? Help service? Consultancy? Etc.??

Table 7: ISD Methodology Comparison Framework (Avison & Fitzgerald, 2006)

4.2 A DSR Methodology Comparison Framework

This section adapts the Avison and Fitzgerald framework from ISD methodologies to DSR methodologies and produces a new DSR Methodology Comparison Framework. Table 8 below presents the aspects from the ISD methodology comparison framework and explains how each aspect is reflected in the DSR Methodology Comparison Framework.

Framework Element or Sub-Element	**Description**
1. Philosophy	What is the 'Weltanschauung' or essence of its approach? Four sub-elements below (same sub-elements as for ISD, but some adapted as shown)
a. Paradigm	Science vs Systems, Objectivist vs Subjectivist Ontology, Positivist vs Interpretive Epistemology (Same as for ISD)
b. Objectives	Possible goals/objectives for DSR methodologies include: Increasing relevance, Increasing research rigour, Improvement (for whom – client? other stakeholders? those disadvantaged? the public? in what way – Efficacy? Effectiveness, Efficiency, Ethicality), Emancipation/critical perspective, Stakeholder consensus, Solving the "right" problem, Artefact effectiveness, Relation to existing literature, Practical significance, Theoretical significance
c. Domain	No specific client, Single client, Multiple/group of clients, societal client
d. Target	Artefact type: IS/IT, CBIS, ISD method/tool/technique/methodology, product (generally, not only in IS/IT), process (generally, not only in IS/IT)
2. Model	What is the basic abstraction and representation mechanism used? (1) verbal, (2) analytic or mathematical, (3) iconic, pictorial, or schematic, and (4) simulation (Same as for ISD)
3. Techniques & Tools	What tools and techniques are used in the methodology? (Same as for ISD)
4. Scope (DSR activities)	What stages/activities of the DSR process are covered? Activities found in common across DSR methodologies include: (a) Problem assessment, (b) Design/ framing, (c) Design/ making, (d) Evaluation, and (e) Reflection.
5. Outputs	What are the deliverables at each stage and at the end? (Same as for ISD)
6. Practice	Three sub-elements below (same sub-elements as for ISD, but some adapted as shown)
a. Background	Commercial or Academic? (Same as for ISD)
b. User Base	Numbers and types of DSR methodology users (Use citations as a surrogate)
c. Participants	What roles participate and what skills are needed? Researcher, Client, User, Other stakeholder
7. Product	What do methodology purchasers get for their money? Software? Training? Documentation? Help service? Consultancy? Etc.?? (Same as for ISD)

Table 8: DSR Methodology Comparison Framework

4.3 Application of the Adapted Framework to DSR Methodologies

This section applies the adapted DSR Methodology Comparison Framework to the six DSR methodologies introduced in section 2. Each methodology is characterised in terms each of the elements of the adapted framework. Table 9 shows the results of our analysis of these six DSR methodologies.

	SDRM	**DSRPM**	**DSRM**	**ADR**	**SDSM**	**PADR**
1. Philosophy						
a. Paradigm	Science, Objectivist, Positivist	Science, Objectivist, Positivist	Science, Objectivist, Positivist	Systems, Subjectivist, Interpretive	Systems, Subjectivist, Interpretive	Systems, Subjectivist, Interpretive
b. Objectives	New artefact, improvement	New artefact, improvement	New artefact, improvement	New artefact, improvement, client service and relevance	New artefact, improvement, effectiveness	New artefact, improvement, effectiveness, consensus, emancipation
c. Domain	No specific client	No specific client	No specific client	Single client	Single or multiple clients	Multiple /societal clients

d. Target	CBIS (computer-based IS), IT	CBIS, IT, methods	CBIS, IT, methods		Product or Process	Product, Urban Informatics
2. Model	unspecified	unspecified	unspecified	unspecified	unspecified	unspecified
3. Techniques & tools	None	None	None	None	None	None
4. Scope (DSR activities)						
a. Problem assessment	Investigate functionalities and requirements	Awareness of the problem	Identify the problem	Problem formulation	1. Learn about specific problem, 2. Inspire and create the general problem and general requirements	Diagnosing and Problem Formulation (Participative problem setting, Ethnographic study)
b. Design/ framing	Construct conceptual framework	Suggestion	Define objectives of solution	Theory ingrained artifact	2. Inspire and create the general problem and general requirements	Action Planning (Opportunity identification, Participative planning)
c. Design/ making	Architect, analyze & build the system	Development	Design & development	Building & intervening	3. Intuit and abduce general solution, 5. & 7. Design & construct specific solution	Action Taking: Design (Participative design, Prototyping & installation)
d. Evaluation	Observe/ evaluate	Evaluation	Evaluation & Extensive adaptation to daily use	Intervening & Evaluation	4. Ex Ante Evaluation (General), 6. Ex Ante Evaluation (Specific), 8. Ex Post Evaluation	Action Taking: Design; Impact Evaluation: (Ethnographic study, Participative evaluation)
e. Reflection	Develop theories & models, consolidate experience	Reflection & abstraction	Communication	Reflection, learning, formalization	Each of the evaluations includes reflection	Reflection and Learning: (Participative client learning, Design theorising for UI)
5. Outputs	Artefact	Artefact, Theory	Artefact	Artefact, Design Theory	Artefact, Design Theory	Artefact, Design Theory
6. Practice						
a. Background	Academic	Academic	Academic	Academic	Academic	Academic
b. User Base (Google citations on 15/08/2017)	(JMIS 1990) 1293	(Webpage 2004) 725 (book 2015) 509	(JMIS 2008) 2561 (DESRIST 2006) 388	(MISQ 2011) 878	(DESRIST 2009) 157 (DESRIST 2007) 73 (book chapter 2014) 1	(JoCI 2011) 55
c. Participants	DSR researchers, users (evaluators)	DSR researchers, users (evaluators)	DSR researchers, users (evaluators)	DSR researchers, clients, users (evaluators)	DSR researchers, clients, users (evaluators)	DSR researchers, clients, public, users (evaluators)
7. Product	Article	Website, Book	2 Articles	Article	3 Articles	Article

Table 9: Comparison of six DSR Methodologies

Assigning a paradigm to a DSR methodology (row 1a in table 9) is somewhat controversial and deserves some justification. The paradigm of a DSR methodology is reflected largely in the methodology's treatment of problems, stakeholders, and evaluations (which test design theories). In our interpretation, SDRM, DSRPM, and DSRM take a more objectivist, positivistic stance to these activities, while the other three methodologies take a more subjectivist, interpretive stance. SDRM largely treats problems as coming from the literature and focuses (positivist) evaluation using experiments. Similarly, DSRPM identifies research problems from "developments in industry or a reference discipline [or] reading in an allied discipline" (Vaishnavi & Kuechler, 2015, pp. 14-15) while evaluation focuses on performance measurement. DSRM is mute about the source of problems, but seeks clarity of the researcher's understanding of the problem and its significance. For evaluation, DSRM suggests "observe and measure how well the artefact supports a solution to the problem" (Peffers et al., 2008, p. 56), a substantially objectivist and positivist position. In contrast, ADR, SDSM, and PADR all specifically include problem formulation based on local (not literature-based) needs and working with client stakeholders in doing so, as well as in the evaluation, which demonstrates a much more subjectivist, interpretive stance. While a DSR methodology has a dominant paradigm, a DSR methodology user may have a different philosophical stance and adapt and use the methodology in accordance with that stance.

Re. Model (row 2 in table 9), all methodologies specify a *process* model of steps, but none specify a generic model for modelling artefacts, other aspects of the work performed, or outputs design theories.

5 Identifying and Specifying a Set of Technological Rules

To identify technological rules we have carefully analysed the outcome of our comparison of methodologies as found in Table 9. It quickly became clear that only very small and minute details distinguish some of the methods. As an example DSRPM (Vaishnavi & Kuechler, 2015) in column 2 and DSRM (Peffers et al., 2008) in column 3 only differ in one place namely "e. Evaluation" where the latter emphasises demonstration as a way of evaluating. Another example is that ADR (column 4) and PADR (column 6) only differ in relation to client(s). That may not be that surprising if one had been derived from the other. But as they both originate in 2011 and were published in parallel one may conclude that combining Action Research with DSR was 'in the air' that year.

However, our analysis led to the identification of a single characteristic that distinguishes the first three methodologies from the last three; that is the paradigm or Weltanschauung embedded. Thus our first technological rule is exactly concentrating on that.

If you need a DSR methodology for planning and organising your research in a situation where you
• believe that people and society, as well as the physical world, operate according to general (natural) laws
• believe that one design artefact (construct, model, method, or instantiation) or design theory can be found to be best
• believe that scientific results have to be objective
then choose an objectivist, positivist methodology such as SDRM (Nunamaker et al., 1990), DSRPM (Vaishnavi & Kuechler, 2015) or DSRM (Peffers et al., 2008)
If not,
then choose a subjectivist and interpretive methodology
such as ADR (Sein et al., 2011), SDSM (Pries-Heje et al., 2014) or PADR (Bilandzic & Venable, 2011)

If we now take a closer look at the three objectivist/positivist methodologies (columns 1 to 3 in Table 9) we can identify a couple of technological rules to choose between them. The first one can be identified in Table 9, row called "c. Design / making":

O/P-1
If you know that the artefact outcome of your research should be an IT system
Then choose SDRM (Nunamaker et al., 1990)

The second technological rule can be found in Table 9, the row called "d. Evaluation":

O/P-2
If extensive adaptation to daily use is needed
Then choose DSRM (Peffers et al., 2008)

The third technological rule can be found in Table 9, the row named "5. Outputs"

O/P-3
If you want to develop design theory
Then choose DSRPM (Vaishnavi & Kuechler, 2015)

Finally we can abduce a fourth rule:

O/P-4

If none of the technological rules O/P1 to 3 apply
Then choose your preferred methodology based on the O/P-paradigm

In the same way, we can take a closer look at the three subjectivist/interpretive methodologies (columns 4 to 6 in Table 9) and identify a few interesting differences in our categorisation. The first one can be found in the row named "c. Domain":

S/I-1
If you have a single client that wants to engage in a research undertaking with you
Then choose ADR (Sein et al., 2011)

The second technological rule can be found in the row "c. Design / making"

S/I-1
If the generic meta-level theory is more important than the intervention in an organisation
Then choose SDSM (Pries-Heje et al., 2014)

The third technological rule can be found in Table 9, the row named "c. Domain", and combining it with the "Participative" characteristic of PADR

S/I-3
If your domain is society-at-large and you have societal clients that are eager to participate
Then choose PADR (Bilandzic & Venable, 2011)

Finally – again - we can abduce a fourth rule; now for the subjective and interpretive methodologies:

S/I-4
If none of the technological rules S/I-1 to 3 apply
Then choose your preferred methodology based on the S/I-paradigm

6 Conclusion and Contribution

We have now provided an answer to the research question we set out to answer namely "How can a DSR researcher choose which DSR methodology he/she should apply and use?" The answer is found above as a set of grounded technological rules – an acknowledged form of design theory.

The rules we have identified prescribe the practical action of choosing a DSR methodology expressing it as a decision design using the format 'if you want to achieve Y in situation Z, then perform action X'.

The core rule we found was that of distinguishing between two types of paradigm or Weltanschauung embedded in the Methodology; Objectivist / Positivist versus Subjectivist / Interpretive. The six methodologies analysed were divided by this rule in two sets of three. For each of these sets, a number of specific technological rules (different from the specific rules in the other set) were then found and described.

Altogether, the DSR methodology comparison framework (in section 4.2), its application to six DSR methodologies (in section 4.3), and the nine technological rules (in section 5) form our contribution and the technological rules provide an answer to the research question.

Thus far, we have not formally evaluated the approach we have developed in practice. It may be that other ways of formulating the technological rules might be easier to understand and therefore be more effective and possess more utility. However, such considerations can be considered in future research.

7 References

Avison, D., & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, techniques and tools* (4th ed.). Maidenhead, Berkshire, UK: McGraw-Hill.

Baskerville, R., Pries-Heje, J., & Venable, J. R. (2007). *Soft Design Science Research: Extending the Boundaries of Evaluation in Design Science Research*. Paper presented at the 2nd International Conference on Design Science Research in Information Systems & Technology (DESRIST 2007), Pasadena.

Baskerville, R., Pries-Heje, J., & Venable, J. R. (2009). *Soft Design Science Methodology*. Paper presented at the 4th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2009), Philadelphia, Pennsylvania, USA.

Bilandzic, M., & Venable, J. R. (2011). Towards Participatory Action Design Research: Adapting action research and design science research methods for urban informatics. *Journal of Community Informatics, 7*(3).

Bunge, M. (1967). *Scientific Research II: The Search for Truth* (Vol. 3). New York: Springer-Verlag.

Byrkjeflot, H., Pedersen, J. S., & Svejenova, S. (2013). From label to practice: The process of creating new Nordic cuisine. *Journal of Culinary Science & Technology, 11*(1), 36-55.

Checkland, P. (1981). *Systems Thinking, Systems Practice*. Chichester, UK: Wiley.

Checkland, P., & Holwell, S. (1999). *Information, Systems, and Information Systems*. Chichester, UK: Wiley.

Checkland, P., & Scholes, J. (1990). *Soft Systems Methodology in Action*. Chichester: J. Wiley.

Checkland, P., & Scholes, J. (1999). *Soft Systems Methodology in Action: A 30-Year Retrospective*. Chichester, UK: Wiley.

Dickson, G. W., Senn, J. A., & Chervany, N. L. (1977). Research in Management Information Systems: The Minnesota experiments. *Management Science, 23*(9), 913-923.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

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

Nunamaker, J. F. J., Chen, M., & Purdin, T. D. M. (1990). Systems Development in Information Systems Research. *Journal of Management Information Systems, 7*(3), 89-106.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-77.

Pries-Heje, J., Venable, J. R., & Baskerville, R. (2014). Soft Design Science Methodology. In O. E. Hansen, J. Simonsen, C. Svabo, S. M. Strandvad, K. Samson, & M. Hertzum (Eds.), *Situated Design Methods*. Cambridge, MA, USA: MIT Press.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly, 35*(1), 37-56.

Simon, H. A. (1996). *The Sciences of the Artificial* (3rd ed.). Cambridge, Mass.: MIT Press.

Vaishnavi, V., & Kuechler, W. J. (2004, 23 October 2013). Design Science Research in Information Systems. Retrieved from http://desrist.org/design-research-in-information-systems/

Vaishnavi, V., & Kuechler, W. J. (2007). *Design Science Research Methods and Patterns: Innovating Information and Communication Technology* (1st ed.). Boca Raton, FL, USA: Auerbach Publications - CRC Press.

Vaishnavi, V., & Kuechler, W. J. (2015). *Design Science Research Methods and Patterns: Innovating Information and Communication Technology* (2nd ed.). Boca Raton, FL, USA: CRC Press.

Välimäki, H., Sørensen, L., & Dahlgren, M. (2004). Manifesto for the New Nordic Kitchen. In.

Van Aken, J., & Romme, A. G. L. (2009). Reinventing the future: adding design science to the repertoire of organization and management studies. *Organization Management Journal, 6*, 5-12.

Van Aken, J. E. (2004). Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. *The Journal of Management Studies, 41*(2), 219-246.

van Aken, J. E. (2005). 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.

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