

Designing TRiDS: Treatments for Risks in Design Science

John R Venable,

Curtin University, Australia
j.venable@curtin.edu.au

Jan vom Brocke,

University of Liechtenstein

Robert Winter,

University of St. Gallen, Switzerland

Abstract

Design Science Research (DSR) has many risks. Researchers inexperienced in DSR, especially early career researchers (ECRs) and research students (e.g. PhD students) risk inefficient projects (with delays, rework, etc.) at best and research project failure at worst if they do not manage and treat DSR risks in a proactive manner. The DSR literature, such as the Risk Management Framework for Design Science Research (RMF4DSR), provides advice for identifying risks, but provides few suggestions for specific treatments for the kinds of risks that potentially plague DSR.

This paper describes the development of a new purposeful artefact (TRiDS: Treatments for Risks in Design Science) to address this lack of suggestions for treatment of DSR risks. The paper describes how the purposeful artefact was developed (following a DSR methodology), what literature it draws upon to inspire its various components, the functional requirements identified for TRiDS, and how TRiDS is structured and why. The paper also documents the TRiDS purposeful artefact in detail, including four main components: (1) an extended set of risk checklists (extended from RMF4DSR), (2) a set of 47 specific suggestions for treating known risks in DSR, (3) a classification of the treatments identified into 14 different categories, and (4) a look-up table for identifying candidate treatments based on a risk in the extended risk checklists. The treatment suggestions and guidance in TRiDS serve as a supplement to RMF4DSR by helping DSR researchers to identify treatments appropriate for a particular DSR project (or program) and thereby to improve DSR project efficiency and the probability of DSR project success.

1 Introduction

There has been much written and published about research methods for Design Science Research (DSR), especially in the field of Information Systems, since the publication of Hevner et al. (2004). Methods for DSR include those by Vaishnavi and Kuechler (2004, 2015), Peffers et al. (2008), Hevner and Chatterjee (2010), Sein et al. (2011), Bilandzic and Venable (2011), and Pries-Heje et al. (2014b).

Managing risks is an important area for any kind of project, including DSR projects. According to Pries-Heje et al. (2014a, p. 61), "Risk Management is the identification of and response to potential problems with sufficient lead time to avoid a crisis. Thus risk management is proactive." Design Science researchers need to manage DSR project risks proactively, lest they incur substantial, possibly severe, costs. Unless they do so, at best, they risk conducting DSR projects inefficiently (with higher costs and delays for rework or following poor design decisions longer than necessary). At worst, they risk DSR project failure, with a complete waste

of time and other resources and potentially the loss of reputation, career opportunities, etc. Inexperienced DSR researchers, especially early career researchers (ECRs) and research students (e.g. Honours, Masters, or PhD students) are especially vulnerable to DSR risks because they have little experience with research and little exposure to the breadth of the DSR literature and their careers are impacted by project results more directly. To be successful, they need timely and comprehensive guidance for DSR risk identification and treatment.

The DSR literature does provide some pockets of advice on managing risks in DSR. For example, the seven guidelines for DSR research included in Hevner et al. (2004) identify key areas of DSR work for which the quality should be managed. Other DSR literature addresses management of DSR risks through early (“formative” or “ex ante”) evaluation (Baskerville et al., 2008; Sonnenberg & Vom Brocke, 2012a, 2012b; Venable et al., 2012, 2016) – as opposed to solely relying on a final (“summative” or “ex post”) evaluation, which is conducted following completion of the design and implementation of the purposeful artefact, when there is little or no time or other resources for correcting any inadequacies in the artefact and or its performance in meeting its objectives. The most substantial contribution to risk management in the DSR literature is that of Baskerville, Pries-Heje, and Venable (Baskerville et al., 2011; Pries-Heje et al., 2008; Pries-Heje et al., 2014a), who specifically address DSR risk identification, assessment, prioritisation, and management. However, none of this literature identifies more than a very few specific treatments for preventing, overcoming, or alleviating specific DSR risks.

There are many risks to both the efficiency and the success of DSR projects, e.g. as identified by Pries-Heje et al. (2014a). However, researchers (especially novices, such as ECRs and research students) lack important advice and suggestions for how to treat risks to DSR, so that they can manage those risks in a proactive and effective way. Supervisors of DSR researchers could also benefit from a more comprehensive treatment of risk management in DSR. Funding bodies and industry partners engaged in DSR projects also have a major interest in identifying and treating risks in collaborative research projects (Vom Brocke & Lippe, 2010).

This paper reports on the development of TRiDS (Treatments for Risks in Design Science), a purposeful artefact developed by the authors to overcome the above problems. TRiDS suggests specific treatments for every specific DSR risk previously identified in the extensive DSR risk checklists by Pries-Heje et al. (2014a), as well as some other DSR risks identified in this paper. Risks that are not specific to DSR (e.g. general project risks such as missing deadlines or scoping failures) are not addressed in this paper.

The rest of this paper follows the organisation suggested by Gregor and Hevner (Gregor & Hevner, 2013) for a publication reporting DSR research. Section 2 expands on the literature identified above and details what is already in the literature about risk treatment in DSR, as well as literature from other domains that inspired our identification of various other risk treatments. Section 3 describes the research method (Design Science Research Process Model or DSRPM (Vaishnavi & Kuechler, 2004, 2015)) used to develop TRiDS. Section 4 describes the TRiDS purposeful artefact, including requirements for TRiDS and the details of the TRiDS artefact and its components. Section 5 describes how TRiDS was evaluated and the outcomes of that evaluation. Section 6 discusses the findings of the research. Finally, section 7 draws conclusions about TRiDS and the potential for further research.

2 Literature Review

This first part of this literature review covers the DSR literature relating to the problem to be addressed and develops the research gap, i.e. the lack of specific suggestions for how to treat risks encountered in DSR. The second part identifies and reviews other literature that inspired some of the specific treatments that are incorporated into TRiDS.

2.1 DSR Literature Relevant to DSR Risk Management

There are three main areas of the DSR literature that are relevant to DSR risk management: (1) Literature on guidelines for DSR, (2) literature on evaluation as a key activity in DSR, and (3) literature specifically about risk management in DSR. These are covered in sections 2.1.1-2.1.3 below.

2.1.1 DSR Guidelines

The DSR literature provides some guidelines for what makes quality or acceptable DSR. Hevner et al. (2004) provide seven guidelines for DSR (see below).

1. Design as an Artefact
2. Problem Relevance
3. Design Evaluation
4. Research Contributions
5. Research Rigour
6. Design as a Search Process
7. Communication of Research

Venable (2010, 2015) surveyed DSR authors, DESRIST program committee members, and IS basket of eight journal editors concerning their opinions of what criteria and standards are more important in DSR. The findings were that Hevner et al. (2004) guidelines 2, 4, 1, and 3 (in that order) were most important, while other highly rated criteria aligned with the guidelines (e.g. relevance of problem and significance for problem clearly established). Also in the top 10 most important criteria were “clear understanding of why the artefact works (or doesn’t work)” and “depth of analysis and clarity of understanding of the problem and its causes.”

It is a sensible approach to DSR risk management to understand key DSR guidelines and take action to ensure that any relevant (or at least that the most important) guidelines are met.

2.1.2 DSR Evaluation Literature

Baskerville, Pries-Heje, and Venable have also published several papers suggesting that DSR risks of poor usability or of technical or organisational feasibility might be treated through multiple, ex ante (or formative) evaluations of prototypes or partial artefacts (Venable et al., 2012, 2016).

One purpose identified for evaluation in DSR is to “evaluate a designed artefact formatively to identify weaknesses and areas of improvement for an artefact under development” (Venable et al., 2012, p. 426). They also identify that artificial, ex ante (formative) evaluation has the lowest risk to participants in the research, but naturalistic, ex post evaluation has the lowest risk of a false positive, which is especially important for safety-critical systems (Venable et al., 2012, p. 432, figure 2).

In their subsequent paper on FEDS (Framework for Evaluation in Design Science, Venable et al., 2016), they suggest that different evaluation strategies (typical trajectories of evaluation episodes) can be used to reduce risks of technical or organisational feasibility or of insufficient evaluation rigour with respect to effectiveness or efficacy. Technical feasibility and/or rigorous evaluation of efficacy are better achieved with a Technical Risk & Efficacy evaluation strategy, which starts with one or more artificial formative evaluation episode(s), transitions to one or more artificial summative evaluation episode(s), and concludes with one or more naturalistic summative evaluation episode(s). Organisational feasibility, usability, and/or rigorous evaluation of effectiveness are better achieved with a Human Risk & Effectiveness evaluation strategy, which starts with one or more artificial formative evaluation episode(s), transitions to one or more naturalistic formative evaluation episode(s), and concludes with one or more naturalistic summative evaluation episode(s).

Sonnenberg and vom Brocke (2012b) present a conceptualisation for concurrent evaluation according to different aspects of design. They build on prior work describing DSR activities within the overall DSR process, and they argue that, in each of these activities, different progress towards the intended artefacts is made, which offer the potential for concurrent (or formative) evaluation. Such evaluation can mitigate risk, since early feedback on more fine-grained steps leading to the eventual artefact can be incorporated into the design process. Further, they argue that evaluation can be more specific (compared to summative evaluation), and the information to be gained can be more directed, if the evaluation is focussed on particular relevant aspects of the design at the time that related design decisions are made in the design process.

In order to exemplify their approach, they distinguish four evaluation types (Eval 1 to Eval 4), which are derived from typical DSR activities, and which are characterized by the input and output of each activity as well as specific evaluation criteria and evaluation methods.

- Eval 1: Evaluating the problem identification: criteria include importance, novelty and feasibility.
- Eval 2: Evaluating the solution design: criteria include simplicity, clarity and consistency.
- Eval 3: Evaluating the solution instantiation: criteria include ease of use, fidelity with real-world phenomena and robustness.
- Eval 4: Evaluating the solution in use: criteria include effectiveness, efficiency and external consistency.

Stoekli et al. (2017) examined how IS scholars actually apply the above evaluations, as reported in papers citing Sonnenberg and vom Brocke (2012b). They found that few DSR researchers try concurrent evaluation and, if they do, they focus on early phases and apparently find value in doing so.

These early evaluations (Eval 1 and Eval 2) are in accordance with using formative evaluations (Venable et al., 2016) and the principle of “fail early, fail often” (Abraham et al., 2014).

2.1.3 DSR Risk Management Literature

Pries-Heje et al. (2014a) provide extensive DSR risks checklists as part of their Risk Management Framework for DSR (RMF4DSR). Their risk checklists cover six areas (A through F) of potential risks in DSR:

- A. Identifying, selecting, and developing understanding of the business needs to be addressed in the DSR project or program,
- B. Grounding of the DSR in knowledge available in the body of recorded human knowledge
- C. Building design artefacts
- D. Evaluating design artefacts and design theories
- E. Artefact dissemination and use
- F. Adding knowledge to the body of human knowledge

In all, Pries-Heje et al. (2014a) identify 38 different risks specific to DSR and suggest that DSR researchers should use the checklist to identify particular risks relevant to their project as instances of each type of risk on the checklist. Checklists A-F below show the risks identified in Pries-Heje et al. (2014a), grouped by each of the above six categories.

Checklist A: Business needs

- A-1 Selection of a problem that lacks significance for any stakeholder
- A-2 Difficulty getting information about the problem and the context
- A-3 Different and even conflicting stakeholder interests (some of which may not be surfaced)
- A-4 Poor understanding of the problem to be solved
- A-5 Solving the wrong problem, i.e. a problem that isn't a main contributor to undesirable outcomes that motivate the problem solving
- A-6 Poor/vague definition/statement of problem to be solved, with potential misunderstanding by others
- A-7 Inappropriate choice or definition of a problem according to a solution at hand
- A-8 Inappropriate formulation of the problem

Checklist B: Grounding

- B-1 Ignorance or lack of knowledge of existing research relevant to the problem understanding and over-reliance on personal experience with or imagination of the problem
- B-2 Ignorance or lack of knowledge of existing design science research into solution technologies for solving the problem, i.e. lack of knowledge of the state of the art
- B-3 Ignorance or lack of knowledge of existing relevant natural and behavioural science research forming kernel theories for understanding or solving the problem

Checklist C: Build Design Artefacts

- C-1 Development of a conjectural (un-instantiated) solution which cannot be instantiated (built or made real), cf. technical risk in Venable et al. (2016)
- C-2 Development of a hypothetical (untried) solution which is ineffective in solving the problem, i.e. the artefact doesn't work doesn't work well in real situations with various socio-technical complications, cf. human risk in Venable et al. (2016)

- C-3 Development of a hypothetical (untried) solution which is inefficient in solving the problem, i.e. requiring overly high resource costs
- C-4 Development of a hypothetical (untried) solution which is inefficacious in solving the problem, i.e. the artefact isn't really the cause of an improvement observed during evaluation
- C-5 Development of a hypothetical (untried) solution which cannot be taught to or understood by those who are intended to use it, e.g. overly complex or inelegant
- C-6 Development of a hypothetical (untried) solution which is difficult or impossible to get adopted by those who are intended to use it, whether for personal or political reasons
- C-7 Development of a hypothetical (untried) solution which causes new problems that make the outcomes of the solution more trouble than the original problem, i.e. there are significant side effects

Checklist D: Evaluate

- D-1 Tacit requirements (which by definition cannot be surfaced) are not dealt with when evaluating the solution technology, leading to failure of the solution technology to meet those requirements
- D-2 Failure to surface some or all of the relevant requirements leads to those requirements not being dealt with when evaluating the solution technology, leading to failure of the solution technology to meet those requirements
- D-3 Incorrectly matching the articulated requirements to the meta-requirements of the Information Systems Design Theory (ISDT) lead to the testing of the IDST and evaluation of an instantiation of the meta-design in a situation for which neither should be applied
- D-4 Incorrectly matching the meta-design or the design method to the meta-requirements (not following the ISDT correctly) leads to evaluation of something other than the correct solution technology or the ISDT as stated
- D-5 Improper application of the meta-design or the design method (not in accordance with the ISDT) in designing an instantiation leads to evaluation of something other than the correct solution technology or the ISDT as stated
- D-6 Improperly building an instantiation of the solution technology (such that it does not properly embody the meta-design) leads to evaluation of something other than the correct solution technology or the ISDT as stated
- D-7 Difficulties in implementing the solution technology during naturalistic evaluation, due to such things as unforeseen complications within the business/organization, prevent the instantiation of the solution technology from successfully meeting its objectives
- D-8 Success of the solution technology to meet its objectives is not obtained due to dynamic or changing requirements beyond the scope of the solution technology
- D-9 Success of the solution technology to meet its objectives is not achieved due to poor change management practices

- D-10 Determination of success or failure in reaching the objectives of the solution technology is error-prone or impossible due to disagreement about objectives or inability to measure
- D-11 Existing organizational culture, local organizational culture differences (sub-cultures), political conflicts, etc. complicate the evaluation process or weaken the ability to make meaningful measurement of the achievement of the objectives of the solution technology
- D-12 Existing organizational priorities, structures, practices, procedures, etc. complicate the evaluation process or ability to make/measure the achievement of the objectives
- D-13 Emergence of new organizational or individual practices, structures, priorities, norms, culture, or other aspects that complicate the acceptability, workability, or efficiency of the application of the solution technology in a naturalistic setting

Checklist E: Artefact Dissemination and Use

- E-1 Implementation in practice of a solution that does not work effectively, efficiently, and/or efficaciously
- E-2 Misunderstanding the appropriate context for and limitations of the solution
- E-3 Misunderstanding how to apply the solution
- E-4 Inappropriate handling of adoption, diffusion, and organizational implementation

Checklist F: Knowledge Additions

- F-1 Inability to publish or present research results
- F-2 Publication of low significance research
- F-3 Publication of incorrect research
- F-4 Design artefacts prove too unique to disseminate

Pries-Heje et al. (2014a) further suggest that, after identifying specific risks inspired by the checklists, DSR researchers should next analyse and prioritise identified risks and only then determine risk treatments, especially for higher priority risks (those risks that are more likely and with more potential of negative impact). To support the identification of treatments, they propose a DSR risk treatment framework, based on the traditional risk management literature, with a generic treatment type in each of four quadrants: self-insure (for low probability, low impact risks), transfer (for low probability, but higher potential impact risks), self-protect (for higher probability, but low impact risks), and avoidance (for higher probability and high potential impact risks) (Pries-Heje et al., 2014a). These four quadrants are also known respectively as the four T's: (1) tolerate, (2) transfer, (3) treat, and (4) terminate (Hopkin, 2017).

However, Pries-Heje et al. (2014a) do not recommend risk treatments for the different kinds of risks. In fact, they expressly avoid doing so, stating "We could list resolution techniques for each risk item. However, we fear that this may limit the generative process of thinking through what should be done about each risk in the specific context. Hence we prefer—and believe it

to be a better approach—to let the user (DSR researcher) decide the Risk Treatment within broader categories of treatments.” (Pries-Heje et al., 2014a, p. 61).

Nonetheless, in explicating the different classes of risk treatments and in providing a naturalistic case study evaluation of RMF4DSR, they do identify a number of specific risk treatments, such as “Use pilots and prototypes so it becomes clear very fast what the contribution could be” and “Use many diverse problem identification techniques such as document study, observe sourcing-atwork [sic], interview at many levels, etc.” (Pries-Heje et al., 2014a, p. 18). However, their examples do not cover treatments for all of the different risks and are sometimes specific to the particular case study in their evaluation of RMF4DSR.

2.1.4 Problem to Be Solved

The problem addressed in this paper is the lack of suggestions and guidance for specific treatments for the large number and variety of risks in conducting DSR. As noted in Pries-Heje et al. (2014a), there are a significant number of risks that are specific to DSR. For those risks, recommended practices from the general research methods and practice literature may not be adequate. Moreover, that literature has not specifically addressed those risks unique to DSR. Even within the DSR literature in particular, identification of risk treatments is far from comprehensive. Many of the risks in DSR have no specific treatments recommended for them in the literature. The very general types of treatments from the general risk management literature incorporated within Pries-Heje et al. (2014a) are difficult to apply because they are insufficiently specific and rely too much on the ingenuity of the researcher. More guidance and candidate suggestions for risk treatments are needed.

2.2 Non-DSR Literature Relevant to DSR Risk Management

This section identifies and summarises salient aspects of other literature that inspired or informed the design of TRiDS.

2.2.1 General and IS Risk Management

There is a large and rich literature on risk management, both generally and in Information Systems. IS Risk Management is a special case of General Risk Management. Furthermore, we can conceptualise that DSR (project) Risk Management is a special case of Research Project Risk Management, which is a special case of Project Risk Management, which is a special case of General Risk Management. While we cannot possibly cover all of this literature here, we mention a few examples useful for our purposes here.

Among many others, Hopkin (2017) gives an overview of risk management generally (not specific to any domain), covering areas such as kinds of risks, risk management standards, an eight-step process for risk management (the 8 R’s: (1) recognise, (2) rate, (3) rank, (4) respond (deciding how to respond – tolerate, transfer, treat, or terminate – the four T’s), (5) resource (the decided response), (6) reaction planning, (7) report, and (8) review), and four risk control techniques (treatments), including (1) preventive, (2) corrective, (3) directive, and (4) detective. While RMF4DSR analyses risks to determine which of the 4 T’s applies, it does not suggest different controls (preventive, corrective, directive, and detective). Detective treatments identify occurrences of risks, which then need to be followed up with corrective treatments (e.g. redesign) to fix the problem identified.

Iversen et al. (2004) suggest four different types of approaches to Software Risk Management: (1) a risk list, (2) a risk-action list, (3) a risk-strategy model, and (4) a risk-strategy analysis. The

approach in RMF4DSR (Pries-Heje et al., 2014a) is to start with a list of known candidate risks and determine their relevance to form a risk list. After prioritisation, the approach adds risk treatments, to form a risk-action list. Importantly, RMF4DSR lacks suggestions for different kinds of treatments, which suggested to the authors that we should add candidate treatments to the candidate risk list, to form a candidate risk-action list. However, neither RMF4DSR nor TRiDS attempt to aggregate risks or treatments and no overall strategy is created.

2.2.2 2.2.2 Participation in System Development and DSR

There is a long and rich literature about participation in system development, including participatory design/development (e.g. the series of Participatory Design Conferences, (PDC, 2018b)), user-centred design/development, co-design, and aspects of design thinking. The key idea is that participation by users/stakeholders in the design of purposeful artefacts (such as IS) helps ensure that the design meets the users' needs and motivates them to adopt and use the resulting purposeful artefact(s). E.g. see (PDC, 2018a). Modern Agile development methods also include user/customer/client participation as a key concept. E.g., the first value of the Agile Manifesto (Beck et al., 2001) is to value "Individuals and interactions over processes and tools", while the third agile value (of only four) is to value "Customer collaboration over contract negotiation."

Participation is a key part of several DSR methodologies, including Action Design Research (Sein et al., 2011), Participatory Action Design Research (Bilandzic & Venable, 2011), and Soft Design Science Methodology (Pries-Heje et al., 2014b).

Research on Agile Software Development has shown the importance of participation to deal with change in information systems design (ISD) processes. According to Conboy (2009) agility is defined as the "the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment". The Manifesto for Agile Software Development (Fowler & Highsmith, 2001) and approaches such as XP and Scrum (cf. Boehm, 2002; Highsmith, 2002; Highsmith & Cockburn, 2001) make normative statements on how to increase agility in ISD, including aspects of participatory design and early feedback on intermediate artefacts. These approaches rejected highly-formalized approaches to working in favour of faster, user-centric, and more dynamic methodologies that enable continuous delivery, requirements change and reflection. Agile methods have been successfully applied across diverse contexts and domains (Boehm, 2002; Gary et al., 2011; Lindvall et al., 2002), and agile adoption has been attracting increasing interest in research, also beyond ISD (Fitzgerald et al., 2006; Nerur et al., 2005; Wang et al., 2012).

2.2.3 Software Engineering Reviews

The software engineering literature describes the practice and benefits of many different kinds of reviews, including design reviews (and software design reviews) (Jalote, 2008), code inspections (Jalote, 2008), peer reviews (Summers, 2011), software inspections (Summers, 2011), software engineering reviews (Summers, 2011), and formal reviews and audits (Summers, 2011). Summers (2011) suggests six formal reviews or audits: system requirements review, system design review, software specification review, preliminary design review, critical design review, and joint technical review. This literature suggests that reviewing intermediate artefacts as well as the final artefact is a key way to identify problems so that the artefacts can be improved (and risks are avoided).

2.2.4 Literature Reviews

There is a very useful body of work on how to accomplish a good literature review (e.g. Jennex (2015); Vom Brocke et al. (2015); Webster & Watson (2002)). Conducting a proper literature review and following the advice in papers such as these can be a very useful approach to reducing research risks, such as those relating to grounding, building and disseminating the artefact, most specifically as identified in RMF4DSR (Pries-Heje et al., 2014a).

2.2.5 Change Management

Several of the risks identified in RMF4DSR (Pries-Heje et al., 2014a) relate to change management. There is extensive literature on how to manage change well. Planning and designing a purposeful artefact to accommodate change or even making change management recommendations could be effective risk management treatments. Knowledge of change management approaches and concerns, such as contained in the Change Management Body of Knowledge (CMBOK) (CMI, 2014) can inspire risk treatments and practices. Specifically, risks related to the “evaluation” and “use” of the artefact (Pries-Heje et al., 2014a) can be mitigated by appropriate dissemination and communication strategies supporting intended users to effectively and efficiently apply the artefact.

2.2.6 Problem Analysis and Formulation

There is a very rich literature on problem analysis and formulation. Among other areas and approaches, Soft Systems Methodology (Checkland, 1981; Checkland & Holwell, 1999; Checkland & Scholes, 1990) provides extensive advice and techniques (e.g. rich pictures, CATWOE, and root definitions) for analysing problems and obtaining agreement among stakeholders. Methods for causal analysis and reasoning include cognitive mapping (Ackermann & Eden, 2001; Eden, 1988; Eden & Ackermann, 1998, 2001) and its extension to coloured cognitive mapping (Venable, 2005) and the application of coloured cognitive mapping to DSR (Venable, 2014). Also very useful for problem analysis and formulation are approaches to stakeholder analysis, such as the instrumental view in Mitchell et al. (1997) and the systems-thinking-based view of Ulrich (1987). All of these approaches potentially provide useful ways to reduce risks related to poor requirements analysis and formulation. Much of the above literature is also closely related to participation by a variety of stakeholders (see section 2.2.2).

In addition to general approaches, tool-based support using shared representations of a problem and/or solution may be useful for creating the common ground needed for joint inquiry (Hickman, 1998; Hildebrand, 2008) into formulating and solving problems, e.g. through co-design (Steen, 2013). Avdiji et al. (2018) describe principles for such tools using examples such as the Business Model Canvas and the Team Alignment Canvas. Still others have developed canvas tools specifically to support DSR (e.g. Morana et al., 2018; Nagle & Sammon, 2016).

Having identified a number of potentially useful avenues from the literature, we now consider how those ideas were selected and incorporated into TRiDS.

3 Research Method

Research methods and techniques are designed purposeful artefacts. They are designed for the purpose of conducting research efficaciously, effectively, efficiently, and ethically (*cf.* the five E's in Checkland and Scholes (1990)). Venable and Baskerville (2012) asserted that, as

designed purposeful artefacts, research methods and techniques should be developed and evaluated using a Design Science Research approach.

To provide useful guidance to DSR researchers, especially inexperienced ones, on risk treatment in DSR, this research developed a new purposeful artefact (TRiDS: Treatments for Risks in Design Science). TRiDS is an approach for identifying suitable treatments for risks potentially encountered in DSR. Therefore, we adopted a DSR approach to this research.

Our research roughly followed the DSR lifecycle of the Design Science Research Process Model (DSRPM), developed by Vaishnavi and Kuechler (2004, 2015). DSRPM has five stages: (1) Awareness of problem, (2) Suggestion, (3) Development, (4) Evaluation, and (5) Conclusion, as well as the possibility to cycle back to earlier stages. We now summarise each of these five stages and introduce how they were addressed in this research and where they are covered in this paper.

3.1 Awareness of Problem Stage

In the awareness of problem stage of DSRPM, we formulated and agreed upon the problem to be solved, in line with section 1 above. In conducting this research we first considered the problem to be solved and then considered requirements for a new purposeful artefact (TRiDS) that could be used to solve the problem. More than just awareness of a problem, it is important to understand the problem to be solved and for researchers to agree on a definition of the problem. Such problem formulation can be found in many DSR methodologies, including Design Science Research Process Model (Peppers et al., 2008), Action Design Research (Sein et al., 2011), or Soft Design Science Methodology (Pries-Heje et al., 2014b).

Importantly, the scope decided for TRiDS was not the initial scope for the artefact. When the research began, we considered additionally providing further guidance for risk identification, as well as a lifecycle or methodology for risk management covering all the DSR activities.

After some initial design iterations and a clearer idea of the role of RMF4DSR, the research team agreed to focus the scope of the design of TRiDS on identifying appropriate and specific treatments for known DSR risks, not on identifying and prioritising the risks to a particular DSR project, which is already handled well by RMF4DSR. Thus, the aim and scope of TRiDS is to serve as a supplement or further step to the support for risk management provided by RMF4DSR. In addition to setting the scope of the problem, we further formulated requirements for a purposeful artefact to solve the problem, as will be described in section 4 on the design of TRiDS.

3.2 Suggestion Stage

The suggestion stage of DSRPM uses abductive reasoning to identify or inspire one or more approaches to solving the problem and making an improvement. In the suggestion phase, we considered the literature of different kinds of treatments for mitigating risks, including literature from DSR risk management, DSR methodologies, and risk management more generally, research methods more generally, software engineering, and system development. An overview of relevant aspects of these was given in section 2. We further reviewed all the known risks and brainstormed ideas for how to mitigate each risk. More details on the suggestion phase are provided in section 4.

3.3 Development Stage

In the development stage of DSRPM, we conducted analyses, developed ideas, and designed an artefact to include those ideas. First, we reviewed the different risks in the Pries-Heje et al. (2014a) checklists and brainstormed different treatments for each individual risk, based on our understanding of the research literature and our collective experiences as researchers. We did this risk by risk to ensure that every single DSR risk has at least one treatment. In doing so, we also identified a small number of risks not included in Pries-Heje et al. (2014a), which we present as part of TRiDS in section 4. Finally, we needed to design a way to present the outcomes of our analysis so that they are parsimonious and easily understandable. During this process, we went through a number of design iterations in which we discussed and reviewed the different treatments, as well as how to present them, then revised our artefact design accordingly. Details on the artefact design and development are provided in section 4.

3.4 Evaluation Stage

For the evaluation stage of DSRPM, we evaluated our purposeful artefact with a criteria-based evaluation as to whether it meets its requirements, as described further below. Further evaluations for usability, utility, etc., are still needed. Details on the artefact evaluation are provided in section 5.

3.5 Conclusion Stage

The conclusion stage of DSRPM was reached when the artefact reached a state where we judged that the artefact is sufficiently useful for its purpose to conclude the research, report the outcomes (thus far), and contribute to the on-going discourse on DSR risk management in the DSR research community. We trust that further work will lead to even better artefacts for managing risk in DSR.

4 Design of TRiDS (DSRPM stages 1-3)

This section describes the design of the purposeful artefact, as suggested by Gregor and Hevner (2013). To do so, this section follows the structure of the first three stages of the DSRPM methodology (Vaishnavi & Kuechler, 2004, 2015). Section 4.1 completes the Awareness of Problem stage, which was begun in the introduction to this paper, by specifying requirements for TRiDS. Section 4.2 describes the Suggestion stage and section 4.3 provides a description of and rationale for the TRiDS artefact, to convey the Development stage of DSRPM.

4.1 Requirements for TRiDS (DSRPM stage 1)

The scope of the problem to be solved – the lack of comprehensive guidance and advice (especially for novice DSR researchers) concerning treatments for known DSR risks – was described and justified in the introduction to this paper. Based on that scope, we were able to proceed with developing what became TRiDS.

However, in order to solve a problem, it is useful to specify the requirements for solving the problem. Setting out the requirements for a solution and designing a solution to meet them enables solving the problem.

Therefore, we formulated and agreed upon the following five requirements for TRiDS, based on the identified scope of extending RMF4DSR to provide needed guidance to novice DSR researchers on specific treatments for reducing and managing DSR risks.

1. Provide as comprehensive a list of all risks relevant to DSR projects as possible, i.e. possibly extending the checklists of known risks provided in RMF4DSR (Pries-Heje et al., 2014a).
2. Identify treatments for all DSR risks identified (comprehensive coverage of risks), including both those already in RMF4DSR and any other DSR risks identified in point #1 above. For every known risk, at least one potential treatment should be identified.
3. Identify a range of different kinds of treatments or treatment strategies. Having more than one suggestion for potential treatments of a DSR risk would improve the flexibility and utility of the purposeful artefact.
4. Where possible, identify treatments in each of the four risk management quadrants (self-insure, transfer, self-protect, and avoidance) found in RMF4DSR (Pries-Heje et al., 2014a).
5. Support users to easily identify specific candidate treatments for each particular DSR risk.

Later, during further review of relevant research in seeking potential treatments and in reviewing the general risk management literature and classifications of risk treatments, as noted in section 2.3.1, we identified that RMF4DSR does not address the four kinds of controls suggested by Hopkin (2017). This suggests that TRiDS could try to identify treatments to cover the breadth of the four different controls for the self-protect/treat quadrant of controls. This led us to add a sixth requirement for what became TRiDS:

6. Where possible, identify controls of all four types (preventive, corrective, directive, and detective) for use in the self-protect/treat quadrant of assessed risks.

4.2 Suggestion Phase and Solution Alternatives (DSRPM stage 2)

In the suggestion phase, we next considered how to approach the problem. As introduced in section 3, our first approach was to brainstorm ideas for treatments for each and every DSR risk identified in RMF4DSR. Ideas came from many areas in the literature as well as the authors' DSR experience and through considering each of the four quadrants of risk treatments in RMF4DSR (Pries-Heje et al., 2014a), i.e. self-insure, transfer, self-protect, and avoidance, as well as the four kinds of controls in Hopkin (2017), i.e. preventive, corrective, directive, and detective.

Many key ideas were drawn from the extant approaches in the literature on DSR guidelines, as discussed in section 2.1.1, the DSR evaluation literature (Venable et al. (2012, 2016), Sonnenberg & Vom Brocke, (2012b)), as discussed in section 2.1.2, and in the DSR risk management literature, as discussed in section 2.1.3. The idea of a treatment to manage key DSR guidelines followed directly from the DSR guideline literature. Venable et al. (2012, 2016), as well as Sonnenberg and vom Brocke (2012b), provided the idea that risks can be reduced through early evaluation and good evaluation strategies should consider potential areas of risk. The Pries-Heje et al. (2014a) paper on RMF4DSR provided a small number of example DSR risk treatments.

However, we also drew on several other areas of extant literature as reviewed in section 2.2, including literature on general and IS risk management, user participation and participatory development, software engineering (specifically software engineering reviews), how to

conduct literature reviews, change management, and problem analysis and solving. Additionally, we drew on the research and system development knowledge and experience of the authors to identify other candidate treatments.

4.3 Design and Development of TRiDS (DSRPM stage 3)

As described earlier, we developed TRiDS by first brainstorming and documenting all of the different treatments, risk by risk, drawing on RMF4DSR. In doing so, we further identified five other DSR risks not found in RMF4DSR. These were added to the RMF4DSR risk checklists at the appropriate point and relevant treatments identified.

Once we had lists of treatments for each risk, we further considered how to make identification of candidate treatments easy, once a particular risk was selected to be treated.

We also needed to consider how to provide advice to novice researchers on what a specific treatment actually means and how to enact that specific treatment once selected. We noted that a number of specific treatments were similar and some could clearly be applied to a number of different risks. This led us to reflect on the range of different treatments and to group treatments conceptually into different classes. Based on that, we are able to provide a description of a group or class of treatments and general advice and literature on how to enact treatments within that class.

The resulting purposeful artefact, which we have named *Treatments for Risk in Design Science* (TRiDS), has four main parts.

The first part is comprised of six Extended DSR Risks Checklists, which make minor extensions to the checklists in RMF4DSR. Table 1 below shows the additional DSR risks we identified. The intention is that the design science researcher will use these extended checklists in the same manner as in RMF4DSR, to identify, analyse, and prioritise DSR risks for which treatments are needed.

The second part is a DSR Treatment List, which lists and explains the 47 different treatments that we identified. Each treatment is a candidate for application for one or more DSR risks.

The third part is a DSR Risk Treatments Lookup Table. This table matches DSR risks to candidate DSR treatments, but is organised by DSR risk, so a DSR researcher can easily look up potential (candidate) DSR risk treatments for any particular (and relevant, sufficiently high priority DSR risk, as determined using the RMF4DSR procedure.

Finally, the fourth part is a classification of the different kinds of treatments. This facilitates providing a succinct summary of each class of treatment and references to relevant literature. Sections 4.3.1-4.3.4 below describe each of the four components of TRiDS.

4.3.1 TRiDS Extended DSR Risks Checklists

As noted earlier, during the development of TRiDS, we identified several DSR risks that were not expressly included in the DSR Risk Checklists in Pries-Heje et al. (2014a). Table 1 below identifies the additional risks that we have added to the Pries-Heje et al. (2014a) DSR Risk Checklists. Pries-Heje et al. (2014a) divided risks into six different categories (A. Business needs, B. Grounding, C. Build purposeful artefacts, D. Evaluate design artefacts and justify design theories or knowledge, E. Artefact dissemination and use, F. Knowledge additions). The full Extended DSR Risks Checklists (including the original RMF4DSR risks and the additional risks below) are provided in Appendix A.

New Risk Number	Additional Risk Description
A-9	Inappropriate choice of meta-requirements (scoping error)
D-13	Type I evaluation error (false positive) (Baskerville et al., 2011)
D-14	Type II evaluation error (false negative) (Baskerville et al., 2011)
E-5	Unsafe artefact released
F-5	Unsafe artefact design published

Table 1. Additional risks in the TRiDS Extended DSR Risk Checklists

4.3.2 TRiDS Treatment List

During our brainstorming and review and revision of different DSR risk treatments, we identified 47 different treatments, which could be used at appropriate points in a DSR project to address the risks in the Extended DSR Risks Checklists described above. Table 2 below lists and briefly describes each of the treatments, as well as listing the relevant risks to which each treatment could be applied. Treatments in Table 2 are listed in the order of the earliest risk to which they apply, e.g. a treatment that applies to risk A-1 is listed before a treatment that doesn't apply to risk A-1 – and so on. Some of the treatments come from or are adapted from treatments in the DSR literature and references are provided where relevant.

#	Detailed Risk Treatment	Relevant Risks
1.	Systematic literature review about the problem, existing solutions, and technological capabilities that could be used for solution	A-1, A-2, A-3, A-4, A-5, A-6, A-7, A-8, B-1, B-2, B-3
2.	Empirical investigation of the problem (e.g. using survey, case study), <i>cf.</i> Pries-Heje et al. (2014a), p. 15	A-1, A-2, A-3, A-4, A-5, A-6, A-7, A-8, D-1b
3.	Seek co-authors or clients with expertise in and understanding of the problem area and its significance	A-1, A-2, A-3, A-4, A-5, A-6, A-7, A-8, B-1, B-2, B-3, C-5, C-6, C-7, D-1a
4.	Stakeholder analysis	A-3, A-4, A-5, A-6, A-7, A-8
5.	Causal analysis (a.k.a. root cause analysis)	A-3, A-4, A-5, A-9
6.	CATWOE/Root definition	A-6, A-7, A-8, A-9
7.	Requirements choice review	A-9
8.	Link requirements to desired outcomes (causal analysis)	A-9
9.	Update literature review, open automatic query	B-1, B-2, B-3
10.	Seek co-authors or clients with expertise in extant purposeful artefact solutions to the problem	B-2, C-4, C-5, C-6, C-7
11.	Seek co-authors or clients with expertise in technologies to be applied in new purposeful artefact	B-2, C-1, C-3
12.	Seek co-authors with expertise in behavioural theory or other areas of potential kernel theory	B-3
13.	Review solution idea and design with technical experts	C-1
14.	Generate multiple candidate designs and contingency plans, <i>cf.</i> Pries-Heje et al. (2014a), p. 14	C-1, C-2, C-3, C-4, C-5, C-6, C-7, D-1a, D-1b, E-1
15.	Evaluate early and formatively, <i>cf.</i> Pries-Heje et al. (2014a, p. 14), Abraham et al. (2014)	C-1, C-2, C-3, C-4, C-5, C-6, C-7
16.	Review solution idea and design with potential users, <i>cf.</i> Pries-Heje et al. (2014a), p. 15	C-2, C-3, C-4, C-6, C-7, D-1a, D-1b
17.	Review partial prototypes as early as possible with users, <i>cf.</i> Pries-Heje et al. (2014a), 15	C-5, D-1a, D-1b
18.	Review solution idea and design with non-user stakeholders, especially with power and different interests	C-6, C-7
19.	Ask "Devil's Advocate" question: "How and why could use of the artefact make a situation worse rather than better?"	C-7

#	Detailed Risk Treatment	Relevant Risks
20.	Triangulate evaluations using different forms, <i>cf.</i> Pries-Heje et al. (2014a), p. 14	C-7, D-13, D-14, E-1, E-5, F-1, F-3
21.	Ask "Devil's Advocate" question: "How and why might the artefact fail to match the requirements?"	D-2
22.	Design and document the design carefully using template/meta-design-driven design & documentation, <i>cf.</i> Pries-Heje et al. (2014a), p. 15	D-2, D-3, D-4, D-5
23.	Design review	D-3, D-4
24.	Instantiation review	D-5
25.	Early (partial) prototype review	D-6
26.	Post-implementation review	D-6
27.	Ask naturalistic evaluation stakeholders about potential and forecast changes (when investigating the problem for naturalistic evaluation)	D-7, D-12
28.	Develop and deliver the working purposeful artefact for naturalistic evaluation quickly	D-7, D-12
29.	Seek co-authors with expertise in evaluation methods	D-7, D-9, D-10, D-11, D-13, D-14, E-1, E-5, F-1
30.	Plan a good change management practice for naturalistic evaluation	D-8
31.	Involve users early and often (a.k.a. user and stakeholder participation) for naturalistic evaluation, <i>cf.</i> Pries-Heje et al. (2014a), p. 15	D-8, D-9, D-10, D-11, D-12
32.	Identify and resolve disagreements among stakeholders (during problem formulation and/or change management)	D9, D-10, D-11, D-12
33.	Support and guide implementers (post research) to implement the purposeful artefact properly	E-1, E-2, E-3, E-4
34.	Rigorously evaluate for (unsafe) side effects, <i>cf.</i> Pries-Heje et al. (2014a), p. 15	E-1, E-5, F-1, F-3, F-5
35.	Help implementers to conduct proper change management	E-4
36.	Throughout the DSR project, actively clarify and manage (plan, review/monitor, and re-plan) the significance of the problem	F-1, F-2
37.	Throughout the DSR project, actively clarify and manage (plan, review/monitor, and re-plan) the newness/novelty of the artefact	F-1
38.	Throughout the DSR project, actively clarify and manage (plan, review/monitor, and re-plan) the rigour of the evaluation	F-1, F-3
39.	Throughout the DSR project, actively clarify and manage (plan, review/monitor, and re-plan) the relationship of the problem and the extant artefact to extant theory and the literature	F-1
40.	Seek co-authors to distribute the workload and the risk, <i>cf.</i> Pries-Heje et al. (2014a), p. 15	F-1, F-2
41.	Change the scope of the research to something less risky, <i>cf.</i> Pries-Heje et al. (2014a), p. 15	F-1, F-2
42.	Abandon the research if too risky, <i>cf.</i> Pries-Heje et al. (2014a), p. 14	F-1, F-2
43.	Enter into contract/agreement clarifying IP rights and right to publish, <i>cf.</i> Pries-Heje et al. (2014a), p. 18	F-1
44.	Ask "Devil's Advocate" question: "How and why might this research produce incorrect results?"	F-3

#	Detailed Risk Treatment	Relevant Risks
45.	Ask “Devil’s Advocate” question (during design and design review): “How and why could the artefact be (or become) too unique to disseminate?”	F-4
46.	Ask experts whether the design is too unique to be used in other contexts (during design review)	F-4
47.	Revise the scope or topic of the research	

Table 2. TRiDS Treatment List (with relevant risks)

4.3.3 TRiDS Risks and Treatments Lookup Table

Having introduced all the DSR risk treatments that we identified, we inverted table 2 so that it is easily searchable by risk to look up candidate risk treatments for any particular risk. Table 3 provides that capability. Unfortunately, space limitations prevent providing explanations of the risks and treatments in Table 3, so only the risk numbers and treatment numbers are included. All DSR risks identified in RMF4DSR and in this paper are present in table 3 and we have identified at least one risk treatment for all of them. All of the risk treatments are described in detail in table 2 above.

DSR Risk	Risk Treatments	DSR Risk	Risk Treatments
A-1	1, 2, 3	D-3	22, 23
A-2	1, 2, 3	D-4	22, 23
A-3	1, 2, 3, 4, 5	D-5	22, 24
A-4	1, 2, 3, 4, 5	D-6	25, 26
A-5	1, 2, 3, 4, 5	D-7	27, 28, 29
A-6	1, 2, 3, 4, 6	D-8	30, 31
A-7	1, 2, 3, 4, 6	D-9	29, 31, 32
A-8	1, 2, 3, 4, 6	D-10	29, 31, 32
A-9*	1, 2, 3, 5, 6, 7, 8	D-11	29, 31, 32
B-1	1, 3, 9	D-12	27, 28, 31, 32
B-2	1, 3, 9, 10, 11	D-13*	20, 29
B-3	1, 3, 9, 12	D-14*	20, 29
C-1	11, 13, 14, 15	E-1	14, 20, 29, 33, 34
C-2	14, 15, 16	E-2	33
C-3	11, 14, 15, 16	E-3	33
C-4	10, 14, 15, 16	E-4	33, 35
C-5	3, 10, 14, 15, 17	E-5*	20, 29, 34
C-6	3, 10, 14, 15, 16, 18	F-1	20, 29, 34, 36, 37, 38, 39, 40, 41, 42, 43
C-7	3, 10, 14, 15, 16, 18, 19, 20	F-2	36, 40, 41, 42
D-1a	3, 14, 16, 17	F-3	20, 34, 38, 44
D-1b	2, 14, 16, 17	F-4	45, 46
D-2	21, 22	F-5*	20, 29, 34

Table 3. TRiDS Risks and Treatments Lookup Table

The asterisks in table 3 indicate new risks introduced in TRiDS.

4.3.4 TRiDS DSR Risk Treatments Classification and Guidance

The DSR risk treatments in Table 2 are listed in order of risks and similar types of treatments are not grouped together. In this section, we classify different treatments, provide further explanation about them where needed, and provide references to literature for further explanation and guidance for how to conduct the treatments.

Within the list of DSR risk treatments above, some treatments are similar to others and can be grouped into classes. We identified 14 generic classes of DSR risk treatments, including (1) Literature Reviews, (2) conducting Empirical Studies, (3) applying Problem Analysis Techniques, (4) encouraging Participation, (5) conducting Reviews, (6) following established Design Practices, (7) following established DSR Evaluation Guidance, (8) applying Change Management practices, (9) using Project Management techniques, (10) seeking Expert Advice, (11) seeking Co-Authorship, (12) asking (and answering) Devil's Advocate Questions, (13) Managing Critical DSR Quality Guidelines, and (14) Changing Project Scope. Some treatments fall into more than one category. We introduce each of these 14 risk treatment categories and categorise them as self-insure, self-protect, transfer, or avoidance in the remainder of this section.

Literature Reviews (treatments 1 and 9):

Much has been written in the IS field about rigorous literature reviews, including work by Webster and Watson (2002), Jennex (2015) and vom Brocke et al. (2015). Effective search queries are essential (e.g. not relying on only one or a few specific keywords, not limiting the search to a subset of academic outlets, and not ignoring current practices and artefacts available from vendors or consultants or used in industry). Also essential is tracking how the literature search was conducted for future reference and expansion. Modern search systems also provide the feature of open queries or alerts, which notify the user when new publications meeting search criteria become available. See, e.g., Google Scholar. Moreover, a good literature search may identify kernel theories (Walls et al., 1992), or justificatory knowledge (Gregor & Jones, 2007) that through abduction in the suggestion phase (Vaishnavi & Kuechler, 2004, 2015) can be adapted to address the problem to be solved. Conducting thorough literature searches and reviews is a preventive form of self-protection (a.k.a. treat) as it reduces the likelihood and impact of risks.

For example, in ongoing research involving one of the authors on managing complexity in project management, the core of the artefact design was suggested and heavily influenced by literature on the nature of complexity and how to deal with it, including papers from the non-academic, practitioner literature.

Empirical Studies (treatments 2 and 20):

Relying solely on the literature may not be appropriate, particularly for understanding problems (which may change over time). Empirically evaluating purposeful artefacts and design theories needs to be carefully designed, using tried and true approaches such as triangulation rather than relying solely on one evaluation approach or episode. Studying a problem empirically is another preventive form of self-protection. Empirical formative evaluation is a detective form of self-protection, which enables corrective treatment through redesign (or possibly through revising the scope of a DSR study). Designing formative and summative evaluations according to recommended practice is a directive form of self-protection.

For example, in ongoing research involving one of the authors on developing an approach for how to enhance and facilitate university collaboration with not-for-profit organisations, surveys of university staff and students were conducted to identify their willingness to collaborate (e.g. in work-integrated learning (WIL) and collaborative research) and to identify potential issues relating to feasibility and needed features.

Problem Analysis Techniques (treatments 4, 5, 6, 7, 27, and 32):

Understanding problems, which are always perceived, and agreeing about them are important in a problem-solving paradigm like DSR. Much has been written about analysing and achieving shared understanding of problems, including work by Peter Checkland and colleagues on Soft Systems Methodology (Checkland, 1981; Checkland & Scholes, 1990), other work about problem analysis and solving methods (e.g. Flood & Jackson (1991); Rosenhead & Mingers (2001)), and work on causal analysis for DSR (Venable, 2014). Careful problem analysis, from different perspectives, is a form of preventive, but also directive self-protection.

For example, Venable and Travis (2000) made use of stakeholder analysis techniques and rich pictures when developing a proposal for a community-based IS Digital Library system and bringing it to the IFIP community for discussion.

Participation (treatments 3, 10, 11, 16, 17, 18, 27, 31 and 32):

Participation by stakeholders, including users, beneficiaries, and decision makers is useful (if not essential) throughout all phases of DSR. The literature above on problem analysis and solving stresses participation. Various DSR methods have also been developed that stress participation, including Action Design Research (ADR - Sein et al., 2011), Participatory Action Design Research (PADR - Bilandzic & Venable, 2011), Soft Design Science Methodology (SDSM - Pries-Heje et al., 2014b), and a different version of Participatory Action Design Research (PADRE - Haj-Bolouri et al., 2016). Using a participative analysis and design approach is a form of preventive, but also directive self-protection.

For example, in his PhD thesis research, Bilandzic (Bilandzic, 2013) undertook two cycles of PADR, involving extensive participation from (and co-design by) members of a community group, to develop a rich understanding a particular context of connected learning and to develop and evaluate two artefacts, Hack The Evening, a social intervention, and Gelatine, a custom-developed ambient media system (for connected/social learning). Heavy participation from stakeholders ensured that their needs were met and that the artefact was rigorously (and authentically) evaluated by those whom it was intended to benefit.

Reviews (treatments 8, 13, 16, 17, 18, 23, 24, 25, 26, 36, 37, 38, 39, 45, and 46):

Formal (or semi-formal) reviews can be conducted at any stage of a DSR project on any of the artefacts developed along the way. The Software Engineering literature describes practices and standards for the conduct of reviews. E.g., see Pressman & Maxim (2015) and Sommerville (2015). Reviews are a detective form of self-protection. Errors or inadequacies identified through reviews enable redesign, a corrective form of self-protection.

For example, many PhD programs have formal review points during a program, e.g. at the stage of proposal approval, at a mid-way stage, and at a point soon before thesis submission. Moreover, some places organise colloquium style peer reviews. However, such general approaches can be extended to review intermediate design artefacts, such as of problem analyses, solution requirements, initial or detailed designs, prototypes, of components of a larger design. Following prescribed practices and identifying quality criteria beforehand to be applied during a review are good practices.

Design Practices (treatments 14, 20, 23, and 28):

There is extensive literature on how to design quality into software and other artefacts. See the software engineering books listed above for reviews as well as journals such as *Design*

Research. Vaishnavi and Kuechler (2015) suggest a large number of DSR design patterns. Moreover, they also reproduce 40 “inventive principles” (Appendix 6A, pages 146-149) from TRIZ (Altschuller, 2005). Following good design practices is a form of directive self-protection.

For example, Vaishnavi and Kuechler (2015), in their discussion of the suggestion phase of the example case of the Smart Objects DSR project (Vaishnavi et al., 1997), noted how following the engineering design principles of partitioning the system into different objects and separating control information from domain information within each object led to a better design, which allowed domain experts to more easily review the correctness of the implementation of a system supporting a complex hierarchy of rules (e.g. for the control system for a nuclear power plant).

Evaluation Guidance (treatments 15, 16, 17, 20, 25, 27, 28, and 34):

As described earlier, the DSR literature provides substantial guidance for evaluation in DSR. See Venable et al. (2012, 2016), Sonnenberg and vom Brocke (2012a, 2012b), or Prat et al. (2015). Early evaluation is a key lesson (Abraham et al., 2014). Early (formative) and thorough evaluations are a detective form of self-protection. It is also an important form of self-insurance as it allows more time to recover (through redesign, a corrective treatment) should a risk (such as a failed evaluation) eventuate.

For example, the Framework for Evaluation in Design Science (FEDS) (Venable et al., 2016) currently has over 300 citations on Google Scholar.

Change Management (treatments 30, 33, and 35):

The fields of management and information systems have both studied change management extensively and provide guidance for how to accomplish it successfully. See, e.g., By (2005). Following good change management practices is a directive, but also a preventive form of self-protection.

For example, Bider et al. (2012) describe the importance of change management in DSR and provide some useful suggestions for a change management process and various change management practices.

Project Management (28, 41, 42, and 43):

In many important ways, a DSR project is just like any other project, and basic project management practices can be very useful to help ensure successful DSR projects and avoid common DSR project pitfalls. See, e.g., vom Brocke and Lippe (2010, 2013) and Lippe and vom Brocke (2016). Although not specialised to DSR, the Memorial University of Newfoundland provides guidance for managing research projects at <https://research-tools.mun.ca/rpm/>. Good project management is a key form of directive self-protection. Scheduling slack time in the project plan is also a key form of self-insurance for a project.

For example, Kearns and Finn (2017) give an example how not carefully considering whether an activity (publishing a paper on a side topic) was within the scope of the research project led to a four-month delay in a research project. Their excellent book, *Supervising PhD Students*, provides a wealth of straightforward, practical advice about planning, organising, monitoring, and controlling a research project, which is highly applicable whether you are a supervisor or not.

Expert Advice (treatments 3, 10, 11, 12, 13, 29, and 46):

For researchers who are not themselves experts in one or more areas relevant to a DSR project, seeking and obtaining expert advice can be a very useful way to minimise key risks. Key areas of expertise might be in the problem to be solved, known extant approaches to the problem, technologies that might be employed, and DSR research and change management methods. Expert advice is a form of preventive self-protection, but is to some extent also a transfer tactic (by transferring some of the risk to experts).

For example, Reiterer (2018), in preparing his PhD thesis developing and evaluating an ontology of DSR concepts to be used for storing, searching, and retrieving DSR knowledge, sought advice from both DSR experts and ontology experts. He also published preliminary research-in-progress papers and obtained valuable feedback and suggestions from expert conference attendees.

Co-Authorship (treatments 3, 10, 11, 12, 29, and 40):

A common practice used to obtain expertise is to invite an expert to join the project as a co-author. This also has the advantage of distributing the risk (and workload) across multiple individuals. Adding co-authors may complicate working arrangements, but is often well worth it. Co-authorship is a key form of transfer (of some of the risk).

For example, one might bring in an expert in statistical analyses to conduct and write up the analysis part of a DSR paper with a quantitative approach to evaluating the artefact.

Devil's Advocate Questions (treatments 19, 21, 44, and 45):

The Devil's Advocate (Latin: *Advocatus Diaboli*) was a formal position within the Roman Catholic church, whose task was to argue against the canonization of people under consideration for sainthood (Wikipedia, 2018). The idea is extended to mean "someone who pretends, in an argument or discussion, to be against an idea or plan that a lot of people support, in order to make people discuss and consider it in more detail" (Cambridge_Dictionary, 2018). A popular corollary expression to that is to seriously (as opposed to flippantly) ask "What could possibly go wrong?" James A. Senn (Senn, 1981), when considering a system development project, would ask and consider "Under what circumstances would implementing the system actually make things worse rather than better?" In our list, we suggest asking similar "Devil's Advocate" questions about different parts of a DSR project. Devil's advocate questions are similar to evaluations, but through thought experiments; they provide some detective self-protection, but also self-insurance when they are asked early enough to discover problems while there is still time to solve them (i.e. take corrective redesign).

For example, the earlier example of empirical research into staff and student willingness to participate in collaborations between not-for-profit organisations and a university arose out of asking Devil's Advocate questions like "What if academic staff don't want to collaborate with not-for-profit organisations on research, e.g. because they don't have money to fund research?" Similarly, one wonders if the design of automated systems for handling customer queries might have been different if the designers had asked: "How might automated phone-based customer service actually reduce business more than the anticipated cost savings of not having real people answer phones?" The design of early email systems might have been

different if the designers had asked: “How might being able to copy an email once to multiple addressees actually make email less rather than more useful?”

Manage Critical DSR Quality Guidelines (treatments 36, 37, 38, 39):

The DSR literature suggests a number of guidelines and recommendations for what makes criteria for quality (or satisfactory) DSR. The most commonly advocated are the seven guidelines in Hevner et al. (2004). Managing the achievement of critical guidelines is a good way to help overcome risks and ensure DSR project success. Of the seven Hevner et al. guidelines, in accordance with Venable (2010, 2015), we chose guidelines 1, 2, 3, and 4 as critical, but also add a fifth guideline: ensuring that the purposeful artefact is clearly and explicitly related to the literature and relevant theory (i.e. it has theoretical significance). Ensuring that you focus attention on key guidelines for a DSR project is a directive form of self-protection, as well as self-insurance by allowing time to reallocate resources to corrective redesign if a problem is discovered.

For example, Chau (2011) explicitly states how all seven of the guidelines by Hevner et al. (2004) are addressed in the research reported in his paper. Of course, managing the achievement of the guidelines during the project would make it easier to report when writing up a paper. Making explicit statements about achieving the guidelines isn't required, but the achievement of the first four Hevner et al. guidelines – as well as our guideline to ensure the work is clearly related to the literature – must be apparent within any DSR paper, so must have been done during the research.

Changing Project Scope or Topic

Revising (especially reducing) the scope or revising the topic of the research is often an effective (and sometimes the only) successful strategy when encountering significant project risks. Reducing the scope may be used to eliminate a particularly risky area of the project, e.g. by solving a smaller, more specific problem or reducing the scope of the solution being attempted. Doing so may reduce the significance of the project, but the project significance may still be acceptable. Sometimes it may even require abandoning a project altogether and pursuing a different project. Changing the project topic or scope is an avoidance tactic. However, it can only be practised early – either up front or in response to discovering a problem. Planning ahead for flexibility in changing the scope of a DSR project is a prudent way to plan for using avoidance (or termination in the 4 Ts terminology) if it becomes necessary.

For example, Riedmann et al.'s (2013) work on a RIVALE prototype of a virtual case study for learning and practising requirements elicitation originally planned to use voice recognition, but it was decided that voice recognition wasn't reliable enough and was difficult to integrate into a virtual world. Because the project scope was adequate without it (users could simply type their interaction instead), that capability was simply excluded from the research.

In another example, the PhD work mentioned earlier by Reiterer originally intended to write software that would automatically analyse DSR publications to identify instances of the DSR concepts and automatically populate the DSR knowledge base. However, the state of the art of semantic analysis (and the amount of effort required to advance it sufficiently) made such research infeasible within available resources. Therefore, the scope was changed to a more limited scope that was feasible for a PhD project with the available resources.

Changing, especially reducing, the scope can be an effective way to keep a DSR research project on track. Other reasons for changing the scope include (1) you identify new research that already has done what you intended to do, (2) new approaches or technologies become available, or (3) interest in a slightly different topic (particularly if funded!) motivates revising the scope to address the related topic.

5 Evaluation of TRiDS (DSRPM stage 4)

This paper identified six requirements for TRiDS in section 4.1. The six requirements are repeated here for ease of reference.

1. Provide as comprehensive a list of all risks relevant to DSR projects as possible, i.e. possibly extending the checklists of known risks provided in RMF4DSR (Pries-Heje et al., 2014a).
2. Identify treatments for all DSR risks identified (comprehensive coverage of risks), including both those already in RMF4DSR and any other DSR risks identified in point #1 above. For every known risk, at least one potential treatment should be identified.
3. Identify a range of different kinds of treatments or treatment strategies. Having more than one suggestion for potential treatments of a DSR risk would improve the flexibility and utility of the purposeful artefact.
4. Where possible, identify treatments in each of the four risk management quadrants (self-insure, transfer, self-protect, and avoidance) found in RMF4DSR (Pries-Heje et al., 2014a).
5. Support users to easily identify specific candidate treatments for each particular DSR risk.
6. Where possible, identify controls of all four types (preventive, corrective, directive, and detective) for use in the self-protect/treat quadrant of assessed risks.

5.1 Criteria-based evaluation

Criteria-based evaluation (Venable et al., 2016; Venable & Travis, 1996) is a form of artificial (non-naturalistic, i.e. it does not evaluate in real use (Sun & Kantor, 2006)) evaluation. Criteria-based evaluation is appropriate when there are criteria concerning the artefact that can be objectively observed and confirmed or disconfirmed. Each of the six requirements identified above can be objectively confirmed or disconfirmed in the design for TRiDS. We evaluated whether each of the six requirements is met below.

The first requirement is to provide a comprehensive list of risks, including extra risks identified in this paper. The list of extra risks is provided in table 1 and the rest can be found in the checklists in section 2.1.3, which are from Pries-Heje et al. (2014a). While comprehensiveness is a requirement, it is also an issue that such a list should be parsimonious, i.e. not including risks that don't warrant attention, e.g. by being too esoteric, unlikely, or overly specific. The list is comprehensive with respect to sufficiently significant risks in the literature and known to the authors. However, it is impossible to establish that other DSR risks could not be identified in the future.

The second requirement is that all risks must have a treatment that applies to that risk. Table 3 above facilitates confirming this, as all risks are listed. Every risk in Table 3 has at least one treatment.

The third requirement for TRiDS is to identify a range of different treatments. We judge that having identified and included 14 different categories of treatments meets this requirement.

The fourth requirement is to (where possible) identify treatments in all four quadrants of risk treatments: self-insure, self-protect, transfer, and avoidance. While self-protection treatments dominate (mostly by reducing the probability of risks occurring), all four quadrants are represented, but not for every risk, as this is not always appropriate or possible.

The fifth requirement that we decided for TRiDS is that it must support easily identifying treatments for each risk. Table 3 is sorted by DSR risk, allowing easy look-up of relevant treatments. Once a risk is identified (using RMF4DSR), the only thing needed then is to use the treatment number to look up the treatment description in Table 2.

Finally, the sixth requirement we selected for TRiDS is to include candidate treatments from all four types of controls (where possible). The classes of treatments discussed in section 4.3 cover not only all four quadrants, but for self-protection protection treatments, the controls suggested include all four kinds: preventive, corrective, detective, and directive, although not for every specific risk, as this is not always appropriate or possible.

5.2 Evaluation Summary

The criteria-based evaluation presented above demonstrates that TRiDS adequately meets all of the requirements that we established for it early in its design. It is potentially debatable whether the list of risks in the risks checklists (both from RMF4DSR and its extension in TRiDS) is sufficiently comprehensive and parsimonious. We believe that it is. However, we also caution that any list of potential risks may miss out on risks, some of which may only arise in the future. Thus, relying totally on a risk checklist may not be appropriate.

We also note that requirements 4 and 6 concern having a range of optional treatments for each and every risk. TRiDS does not provide options in all cases. For many kinds of risks in DSR, only limited options may be available. That said, many risks have a wide variety of treatments available, as identified in table 3.

Importantly, the above criteria-based, artificial evaluation (meeting requirements) does not yet consider issues of usability, practicality, comprehensibility of potential treatments, and other potential issues. Further naturalistic evaluation is needed with DSR researchers (student or practising) on actual DSR projects. Such work is beyond the scope of this paper.

6 Discussion

This research has developed an approach for identifying risk treatments in DSR (TRiDS), which extends prior research on DSR evaluation and risk management with additional identified risks and recommendations for risk treatments. As discussed in section 5, TRiDS meets the six requirements identified for how to provide more detailed guidance for how to effectively and efficiently treat risks in DSR projects.

This paper identifies new risks in DSR, identifies and classifies treatments for DSR risks into 14 categories, and fills an important gap in the literature.

6.1 Theoretical significance

The knowledge developed in this paper can be formulated as a nascent design theory, including new constructs of the artefact and for meeting the requirements of providing better guidance for how to manage risk.

From a Design Theory perspective, TRiDS provides a *general design* (Baskerville & Pries-Heje, 2010; Venable, 2013) of an artefact for matching DSR risks to candidate DSR treatments, which can later be expanded with additional risks and treatments. The artefact's *general design* (an interacting set of components, e.g. tables) meets the *general requirements* (Baskerville & Pries-Heje, 2010; Venable, 2013) to match risks and treatments through a *utility relationship between them for improved effectiveness and efficiency* (Venable, 2006, 2013). Moreover, the approach could be generalised for use for risk management in other domains than in DSR. Finally, the risks and risk treatments identified could be generalised to other research paradigms than DSR. The design theory includes new constructs of the artefact and increased utility (compared to extant approaches) for meeting the requirements of providing better guidance for how to treat DSR risks.

6.2 Practical significance

Appropriate use of TRiDS, as described in this paper, should provide effective guidance to DSR researchers, especially non-expert researchers, in managing the risks to their DSR projects and/or programs, which should thereby increase the efficient use of resources and the probability of a DSR project/program's success. The use of TRiDS should enhance DSR and help make it more efficacious, effective, efficient, and ethical. DSR, in turn, should better serve society by delivering more and higher-quality contributions to the solution of real-world problems.

6.3 Limitations

As noted earlier concerning the evaluation of TRiDS, while helpful, no checklist is complete (so risks may be missed) and no treatment is guaranteed to work 100% of the time. Users of RMF4DSR and TRiDS must be diligent in their application and consider how their DSR project's context might reduce the effectiveness of risk management approaches.

Another limitation of this work is that, while extending the risk list (Iversen et al., 2004) provided by Pries-Heje et al (2014a) into a more specific risk-action list (Iversen et al., 2004), it does not provide the strategic oversight offered by a risk-strategy model or a risk-strategy analysis (Iversen et al., 2004). Future work might further extend RMF4DSR and TRiDS to provide such strategic oversight.

7 Conclusion

The research reported in this paper extends the advice to DSR researchers on how to manage risks in DSR projects/programs by identifying 47 potential treatments for known risks in DSR. It also provides a straightforward way to work from a relevant risk – the augmented risk checklists – to identify candidate treatments for a particular risk. It has further classified the risk treatments into 14 categories, provided relevant examples, and made reference to relevant literature to guide and support their enactment. Overall, it has extended the existing literature in DSR risk management with some very practical ideas for how to address and treat risks, which should assist DSR researchers to better manage risks in their DSR projects and

programs, and thus to better conduct DSR in terms of efficacy, effectiveness, efficiency, and ethics.

However, further research would be useful. TRiDS could still use a more complete, naturalistic evaluation. It could also benefit by supplementing it with better guidance on how to decide among the treatments identified and how and when to apply the treatments identified for best results.

Furthermore, TRiDS could be extended further. Ultimately, what we envision is an approach to DSR that is risk-aware and identifies and treats risks in an agile and continuous way, developing and evaluating intermediate artefacts rather than waiting for a summative evaluation at the end of a linear research process. Such an approach should lead to DSR that is both more effective (higher quality and more needed outcomes) and more efficient (less wasted effort and rework).

References

- Abraham, R., Aier, S., & Winter, R. (2014). Fail Early, Fail Often: Towards Coherent Feedback Loops Loops in Design Science Research Evaluation. In E. Karahanna, A. Srinivasan & B. Tan (Eds.), *35th International Conference on Information Systems, held in Auckland, New Zealand*, Atlanta, GA, USA: Association for Information Systems.
- Ackermann, F., & Eden, C. (2001). SODA – Journey Making and Mapping in Practice. In J. Rosenhead & J. Mingers (Eds.), *Rational Analysis for a Problematic World Revisited*. Chichester, UK: Wiley.
- Altschuller, G. S. (2005). *The Innovation Algorithm: TRIZ, Systematic Innovation and Technical Creativity*. Worcester, MA, USA: Technical Innovation Center.
- Avdiji, H., Elikan, D., Missonier, S., & Pigneur, Y. (2018). Designing Tools for Collectively Solving Ill-Structured Problems *Hawaii International Conference on System Sciences* held in Waikoloa Village, Hawaii, USA.
- Baskerville, R., & Pries-Heje, J. (2010). Explanatory Design Theory. *Business & Information Systems Engineering*, 2010(5), 271-282.
- Baskerville, R., Pries-Heje, J., & Venable, J. R. (2008). Evaluation Risks in Design Science Research: A Framework. In R. Baskerville & V. Vaishnavi (Eds.), *3rd International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008) held in Atlanta, Georgia, USA*,
- Baskerville, R., Pries-Heje, J., & Venable, J. R. (2011). A Risk Management Framework for Design Science Research *44th Hawaii International Conference on System Science (HICSS 2011), held in Kauai, Hawaii, USA*,
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from <http://agilemanifesto.org/>
- Bider, I., Johannesson, P., Perjons, E., & Johannesson, L. (2012). Design Science in Action: Developing a Framework for Introducing IT Systems into Operational Practice. In F. J. George (Ed.), *Thirty Third International Conference on Information Systems, held in Orlando, FL, USA*, Atlanta, GA, USA: Association for Information Systems.

- Bilandzic, M. (2013). *The Embodied Hybrid Space – Designing Social and Digital Interventions to Facilitate Connected Learning in Coworking Spaces* PhD Queensland University of Technology, Brisbane, QLD, Australia.
- 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) Retrieved from <http://ci-journal.net/index.php/ciej/article/view/786/804>
- Boehm, B. (2002). Get Ready for Agile Methods, with Care. *Computer*, 35, 64-69.
- By, R. T. (2005). Organisational Change Management: A Critical Review. *Journal of Change Management*, 54(4), 369-380. <http://dx.doi.org/10.1080/14697010500359250>
- Cambridge_Dictionary. (2018). devil's advocate. Retrieved from <https://dictionary.cambridge.org/dictionary/english/devil-s-advocate>
- Chau, M. (2011). Visualizing Web Search Results Using Glyphs: Design and evaluation of a flower metaphor. *ACM Transactions on Management Information Systems*, 2(1), 2:1-2:27. <http://dx.doi.org/10.1145/1929916.1929918>
- 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.
- CMI, T. C. M. I. (2014). *The Effective Change Manager: The Change Management Body of Knowledge*. London, UK: The Stationery Office (TSO) Limited.
- Conboy, K. (2009). Agility from First Principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20, 329-354.
- Eden, C. (1988). Cognitive Mapping: a review. *European Journal of Operational Research*, 36(1), 1-13.
- Eden, C., & Ackermann, F. (1998). *Making Strategy: The Journey of Strategic Management*. London: Sage.
- Eden, C., & Ackermann, F. (2001). SODA - The Principles. In J. Rosenhead & J. Mingers (Eds.), *Rational Analysis for a Problematic World Revisited*. Chichester, UK: Wiley.
- Fitzgerald, B., Harnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15, 200-213.
- Flood, R. L., & Jackson, M. C. (1991). *Creative Problem Solving: Total Systems Intervention*. Chichester, UK: Wiley.
- Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. *Software Development*, 9, 28-35.
- Gary, K., Enquobahrie, A., Ibanez, L., Cheng, P., Yaniv, Z., Cleary, K., Heidenreich, J. (2011). Agile Methods for Open Source Safety-Critical Software. *Software: Practice and Experience*, 41, 945-962.
- Gregor, S., & Hevner, A. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2), 337-355.

- Gregor, S., & Jones, D. (2007). The Anatomy of a Design Theory. *Journal of the Association for Information Systems*, 8(5), 312-335.
- Haj-Bolouri, A., Bernhardsson, L., & Rossi, M. (2016). PADRE: A Method for Participatory Action Design Research. In J. Parsons, T. Tuunanen, J. R. Venable, B. Donnellan, M. Helfert & J. Kenneally (Eds.), *11th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2016), held in St John's, NL, Canada*, (pp. 19-36). Berlin, Germany: Springer.
- Hevner, A., & Chatterjee, S. (2010). *Design Research in Information Systems: Theory and Practice*. New York, NY, USA: Springer.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75-105.
- Hickman, L. A. (1998). Dewey's Theory of Inquiry. In L. A. Hickman (Ed.), *Reading Dewey: Interpretations for a Postmodern Generation*. Bloomington, IN, USA: Indiana University Press.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Boston, MA, USA: Addison-Wesley.
- Highsmith, J., & Cockburn, A. (2001). Agile Software Development: The business of innovation. *Computer*, 334, 120-127.
- Hildebrand, D. (2008). *Dewey: A Beginner's Guide*. Oxford, UK: Oneworld.
- Hopkin, P. (2017). *Fundamentals of Risk Management: Understanding, evaluating and implementing effective risk management* (4th ed.). London, UK: Kogan Page.
- Iversen, J. H., Mathiassen, L., & Nielsen, P. A. (2004). Managing Risk in Software Process Improvement: An Action Research Approach. *MIS Quarterly*, 28(3), 395-433.
- Jalote, P. (2008). *A Concise Introduction to Software Engineering*. London, UK: Springer-Verlag.
- Jennex, M. E. (2015). Literature Reviews and the Review Process: An Editor-in-Chief's Perspective. *Communications of AIS*, 36(8)
- Kearns, H., & Finn, J. (2017). *Supervising PhD Students: A practical guide and toolkit*. Adelaide, SA, Australia: ThinkWell.
- Lindvall, M., Basilli, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Zelkowitz, M. (2002). Empirical Findings in Agile Methods. In D. Wells & L. Williams (Eds.), *Extreme Programming and Agile Methods (XP/Agile Universe)*, held in Chicago, IL, USA: Springer.
- Lippe, S., & vom Brocke, J. (2016). Situational Project Management for Collaborative Research Projects. *Project Management Journal*, 47(1), 76-96.
- Mitchell, R. K., Angle, B. R., & Wood, D. J. (1997). Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. *Academy of Management Journal*, 22(4), 853-886.
- Morana, S., Scheid, M., Gau, M., Benke, I., Vom Brocke, J., Fettke, P., & Maedche, A. (2018). Research Prototype: The Design Canvas in MyDesignProcess. In S. Chatterjee, K. Dutta

- & R. P. Sundarraj (Eds.), *13th International Conference on Design Science Research in Information Systems and Technology, held in Chennai, India*: Springer.
- Nagle, T., & Sammon, D. (2016). The Development of a Practitioner Design Science Research Canvas AIS SIGPRAG: *Practice-based Design and Innovation of Digital Artifacts, held in Dublin, Ireland*: Association for Information Systems. Retrieved from <http://sigprag.net/2016/11/>
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
- PDC. (2018a). About PDC. Retrieved from <https://pdc2018.org/about-pdc/>
- PDC. (2018b). FREE Portal to the Complete Archive of Participatory Design Conference (PDC) Proceedings Retrieved from <http://pdcproceedings.org/>
- Peffer, 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.
- Prat, N., Comyn-Wattiau, I., & Akoka, J. (2015). A Taxonomy of Evaluation Methods for Information Systems Artifacts. *Journal of Management Information Systems*, 32(3), 229-267.
- Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach*. New York, NY, USA: McGraw-Hill.
- Pries-Heje, J., Venable, J. R., & Baskerville, R. (2008). A Risk Management Framework for Design Science Research *31st Information Systems Research Seminar in Scandinavia (IRIS 2008)*, held in Åre, Sweden.
- Pries-Heje, J., Venable, J. R., & Baskerville, R. (2014a). RMF4DSR: A Risk Management Framework for Design Science Research. *Scandinavian Journal of Information Systems*, 26(1), Article 3. Retrieved from <http://aisel.aisnet.org/sjis/vol26/iss1/3>
- Pries-Heje, J., Venable, J. R., & Baskerville, R. (2014b). 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.
- Reiterer, E. H. R. (2018). *Design and Evaluation of DSRCRSA: An Ontology-Based Approach to Design Science Research Content Representation and Summarisation* PhD Curtin University, Perth WA Australia.
- Riedmann, P., Venable, J. R., Chang, V., Reiners, T., & Gütl, C. (2013). RIVALE: A Prototype Realistic Immersive Virtual Agent-Based Learning Environment Case Study for Learning Requirements Elicitation Skills. *2013 AIS SIGED: IAIM International Conference on Informatics Education and Research (SIGED 2013)* held in Milan, Italy
- Rosenhead, J., & Mingers, J. (Eds.). (2001). *Rational Analysis for a Problematic World: Problem Structuring Methods for Complexity, Uncertainty and Conflict*. Chichester, UK: Wiley.
- Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Quarterly*, 35(1), 37-56.
- Senn, J. A., personal communication, 1981.
- Sommerville, I. (2015). *Software Engineering* (10th ed.). Harlow, Essex, UK: Pearson.

- Sonnenberg, C., & Vom Brocke, J. (2012a). Evaluation Patterns for Design Science Research Artefacts. In M. Helfert & B. Donnellan (Eds.), *European Design Science Symposium (EDSS) 2011, held in Dublin, Ireland*, (pp. 71-83). Berlin/Heidelberg, Germany: Springer.
- Sonnenberg, C., & Vom Brocke, J. (2012b). Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research. In K. Peffers, M. Rothenberger & B. Kuechler (Eds.), *7th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2012), held in Las Vegas, Nevada, USA*, (pp. 381-397). Springer.
- Steen, M. (2013). Co-Design as a Process of Joint Inquiry and Imagination. *DesignIssues*, 29(2), 16-28.
- Stoekli, E., Neiditsch, G., Uebernickel, F., & Brenner, W. (2017). Towards an understanding of how and why Design Science Research scholars evaluate. In M. Indulska, K. Riemer & V. Tuunainen (Eds.), *Australasian Conference on Information Systems*, held in Hobart, Tas, Australia.
- Summers, B. L. (2011). *Software Engineering Reviews and Audits*. Boca Raton, FL, USA: CRC Press.
- Sun, Y., & Kantor, P. B. (2006). Cross-Evaluation: A new model for information system evaluation. *Journal of the American Society for Information Science and Technology*, 57(5), 614-628.
- Ulrich, W. (1987). Critical Heuristics of Social Systems Design. *European Journal of Operational Research*, 31, 276-283.
- Vaishnavi, V., & Kuechler, W. J. (2004). Design Science Research in Information Systems. 23 October 2013 Retrieved from <http://desrist.org/design-research-in-information-systems/>
- 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.
- Vaishnavi, V. K., Buchanan, G. C., & Kuechler, W. L. (1997). A data/knowledge paradigm for the modeling and design of operations support systems. *IEEE Transactions on Knowledge & Data Engineering*, 9(2), 275-291.
- Venable, J. R. (2005). Coloured Cognitive Maps for Modelling Decision Contexts. In T. Bui & A. Gachet (Eds.), *First International Workshop on Context Modeling and Decision Support, held in Paris, France*: CEUR Workshop Proceedings. Retrieved from http://CEUR-WS.org/Vol-144/03_venable.pdf
- Venable, J. R. (2006). The Role of Theory and Theorising in Design Science Research. *First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)* held in Claremont, CA, USA.
- Venable, J. R. (2010). Design Science Research Post Hevner et al: Criteria, Standards, Guidelines, and Expectations. In R. Winter, J. L. Zhao & S. Aier (Eds.), *5th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2010), held in St Gallen, Switzerland*: Springer.
- Venable, J. R. (2013). Rethinking Design Theory in Information Systems. In J. vom Brocke, S. Ram & M. Rossi (Eds.), *8th International Conference on Design Science Research in*

Information Systems and Technology (DESRIST 2013), held in Helsinki, Finland, Heidelberg, Germany: Springer.

- Venable, J. R. (2014). Using Coloured Cognitive Mapping (CCM) for Design Science Research. In M. C. Tremblay, D. VanderMeer, M. Rothenberger, A. Gupta & V. Yoon (Eds.), *9th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2014), held in Miami, FL, USA*, (pp. 345-359). Berlin, Germany: Springer.
- Venable, J. R. (2015). Five and Ten Years On: Have DSR Standards Changed? In B. Donnellan, M. Helfert, J. Kenneally, D. VanderMeer, M. Rothenberger & R. Winter (Eds.), *10th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2015), held in Dublin, Ireland*, (pp. 264-279). Berlin, Germany: Springer.
- Venable, J. R., & Baskerville, R. (2012). Eating Our Own Cooking: Toward a More Rigorous Design Science of Research Methods. *Electronic Journal of Business Research Methods*, 10(2), 141-153. Retrieved from <http://www.ejbrm.com/volume9/issue2>
- Venable, J. R., Pries-Heje, J., & Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research. In K. Peffers, M. Rothenberger & B. Kuechler (Eds.), *7th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2012), held in Las Vegas, Nevada, USA*, (pp. 423-438). Springer.
- Venable, J. R., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, 25(1), 77-89. <http://dx.doi.org/10.1057/ejis.2014.36>
- Venable, J. R., & Travis, J. (1996). Criteria-Based Evaluation of Information Systems Modelling Languages. In K. Siau & Y. Wand (Eds.), *1st Workshop on Evaluation of Modelling Methods in Systems Analysis and Design (EMMSAD '96), held in Heraklion, Crete, Greece: University of British Columbia, Vancouver, BC, Canada.*
- Venable, J. R., & Travis, J. (2000). Developing a Virtual Community-based Information Systems Digital Library: A Proposal and Research Program. In R. Baskerville, J. Stage & J. I. DeGross (Eds.), *Organizational and Social Perspectives on Information Technology: IFIP TC8 WG8.2 International Working Conference on the Social and Organizational Perspective on Research and Practice in Information Technology, held in Aalborg, Denmark*, (pp. 319-336). Boston, MA, USA: Kluwer Academic Press.
- Vom Brocke, J., & Lippe, S. (2010). Taking a Project Management Perspective on Design Science Research. In R. Winter, J. L. Zhao & S. Aier (Eds.), *5th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2010), held in St Gallen, Switzerland*, (pp. 31-44). Berlin: Springer-Verlag.
- Vom Brocke, J., & Lippe, S. (2013). Identifying and Managing Creative Tasks in Collaborative IS Research Projects. *Project Management Journal*, 44(6), 94-113.
- Vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., & Cleven, A. (2015). Standing on the shoulders of giants: Challenges and recommendations of literature search in Information Systems research. *Communications of ACM*, 37(1), 205-224.
- 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.

Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software, 85*, 1287-1299.

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly, 26*(2), xiii-xxiii.

Wikipedia. (2018). Devil's advocate. Retrieved from https://en.wikipedia.org/wiki/Devil%27s_advocate

Appendix A: Extended DSR Risk Checklists

Checklists below are extended from RMF4DSR (Pries-Heje et al., 2014a).

Five additional risks are annotated with an asterisk.

Checklist 1: Business needs

A-1 Selection of a problem that lacks significance for any stakeholder
A-2 Difficulty getting information about the problem and the context
A-3 Different and even conflicting stakeholder interests (some of which may not be surfaced)
A-4 Poor understanding of the problem to be solved
A-5 Solving the wrong problem, i.e. a problem that isn't a main contributor to undesirable outcomes that motivate the problem solving
A-6 Poor/vague definition/statement of problem to be solved, with potential misunderstanding by others
A-7 Inappropriate choice or definition of a problem according to a solution at hand
A-8 Inappropriate formulation of the problem
*A-9 Inappropriate choice of meta-requirements (scoping error)

Checklist 2: Grounding

B-1 Ignorance or lack of knowledge of existing research relevant to the problem understanding and over-reliance on personal experience with or imagination of the problem
B-2 Ignorance or lack of knowledge of existing design science research into solution technologies for solving the problem, i.e. lack of knowledge of the state of the art
B-3 Ignorance or lack of knowledge of existing relevant natural and behavioural science research forming kernel theories for understanding or solving the problem

Checklist 3: Build Design Artefacts

C-1 Development of a conjectural (un-instantiated) solution which cannot be instantiated (built or made real), <i>cf.</i> technical risk in Venable et al. (2016)
C-2 Development of a hypothetical (untried) solution which is ineffective in solving the problem, i.e. the artefact doesn't work doesn't work well in real situations with various socio-technical complications, <i>cf.</i> human risk in Venable et al. (2016)
C-3 Development of a hypothetical (untried) solution which is inefficient in solving the problem, i.e. requiring overly high resource costs

C-4 Development of a hypothetical (untried) solution which is inefficacious in solving the problem, i.e. the artefact isn't really the cause of an improvement observed during evaluation
C-5 Development of a hypothetical (untried) solution which cannot be taught to or understood by those who are intended to use it, e.g. overly complex or inelegant
C-6 Development of a hypothetical (untried) solution which is difficult or impossible to get adopted by those who are intended to use it, whether for personal or political reasons
C-7 Development of a hypothetical (untried) solution which causes new problems that make the outcomes of the solution more trouble than the original problem, i.e. there are significant side effects

Checklist 4: Evaluate

D-1 Tacit requirements (which by definition cannot be surfaced) are not dealt with when evaluating the solution technology, leading to failure of the solution technology to meet those requirements
D-2 Failure to surface some or all of the relevant requirements leads to those requirements not being dealt with when evaluating the solution technology, leading to failure of the solution technology to meet those requirements
D-3 Incorrectly matching the articulated requirements to the meta-requirements of the ISDT lead to the testing of the IDST and evaluation of an instantiation of the meta-design in a situation for which neither should be applied
D-4 Incorrectly matching the meta-design or the design method to the meta-requirements (not following the ISDT correctly) leads to evaluation of something other than the correct solution technology or the ISDT as stated
D-5 Improper application of the meta-design or the design method (not in accordance with the ISDT) in designing an instantiation leads to evaluation of something other than the correct solution technology or the ISDT as stated
D-6 Improperly building an instantiation of the solution technology (such that it does not properly embody the meta-design) leads to evaluation of something other than the correct solution technology or the ISDT as stated
D-7 Difficulties in implementing the solution technology during naturalistic evaluation, due to such things as unforeseen complications within the business/organization, prevent the instantiation of the solution technology from successfully meeting its objectives
D-8 Success of the solution technology to meet its objectives is not obtained due to dynamic or changing requirements beyond the scope of the solution technology
D-9 Success of the solution technology to meet its objectives is not achieved due to poor change management practices

D-10 Determination of success or failure in reaching the objectives of the solution technology is error-prone or impossible due to disagreement about objectives or inability to measure
D-11 Existing organizational culture, local organizational culture differences (sub-cultures), political conflicts, etc. complicate the evaluation process or weaken the ability to make meaningful measurement of the achievement of the objectives of the solution technology
D-12 Existing organizational priorities, structures, practices, procedures, etc. complicate the evaluation process or ability to make/measure the achievement of the objectives D-III Emergence of new organizational or individual practices, structures, priorities, norms, culture, or other aspects that complicate the acceptability, workability, or efficiency of the application of the solution technology in a naturalistic setting
*D-13 Type I evaluation error (false positive) (Baskerville et al., 2011)
*D-14 Type II evaluation error (false negative) (Baskerville et al., 2011)

Checklist 5: Artefact Dissemination and Use

E-1 Implementation in practice of a solution does not work effectively, efficiently, and/or efficaciously
E-2 Misunderstanding the appropriate context for and limitations of the solution
E-3 Misunderstanding how to apply the solution
E-4 Inappropriate handling of adoption, diffusion, and organizational implementation
*E-5 Unsafe purposeful artefact released

Checklist 6: Knowledge Additions

F-1 Inability to publish or present research results
F-2 Publication of low significance research
F-3 Publication of incorrect research
F-4 Design artefacts prove too unique to disseminate
*F-5 Unsafe purposeful artefact design published

Copyright: © 2019 Venable, vom Brocke & Winter. This is an open-access article distributed under the terms of the [Creative Commons Attribution-NonCommercial 3.0 Australia License](https://creativecommons.org/licenses/by-nc/3.0/australia/), which permits non-commercial use, distribution, and reproduction in any medium, provided the original author and AJIS are credited.

