Progress in IS

Jan vom Brocke
Alan Hevner
Alexander Maedche *Editors*

Design Science Research. Cases

Springer

Progress in IS

"PROGRESS in IS" encompasses the various areas of Information Systems in theory and practice, presenting cutting-edge advances in the field. It is aimed especially at researchers, doctoral students, and advanced practitioners. The series features both research monographs that make substantial contributions to our state of knowledge and handbooks and other edited volumes, in which a team of experts is organized by one or more leading authorities to write individual chapters on various aspects of the topic. "PROGRESS in IS" is edited by a global team of leading IS experts. The editorial board expressly welcomes new members to this group. Individual volumes in this series are supported by a minimum of two members of the editorial board, and a code of conduct mandatory for all members of the board ensures the quality and cutting-edge nature of the titles published under this series.

More information about this series at http://www.springer.com/series/10440

Jan vom Brocke · Alan Hevner ·
Alexander Maedche
Editors

Design Science Research. Cases

Springer

Editors
Jan vom Brocke
Department of Information Systems
University of Liechtenstein
Vaduz, Liechtenstein

Alan Hevner
Muma College of Business
University of South Florida
Tampa, FL, USA

Alexander Maedche
Institute of Information Systems
and Marketing (IISM)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

ISSN 2196-8705 ISSN 2196-8713 (electronic)
Progress in IS
ISBN 978-3-030-46780-7 ISBN 978-3-030-46781-4 (eBook)
https://doi.org/10.1007/978-3-030-46781-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

DSR Cases Book: Preface

The Design Science Research (DSR) paradigm has become central to Information Systems (IS) studies in the past 20 years. Simply stated, the goal of DSR is to generate knowledge on how to build innovative solutions to important problems in the form of models, methods, constructs, and instantiations. DSR aims to provide knowledge of how things can and should be constructed or arranged (i.e., designed). In this sense, design knowledge describes means-end relationships between problem and solution spaces.

DSR is ideally positioned to make both research and practical contributions. From a research point of view, it contributes to the technology body of knowledge in the form of innovative design artifacts. Furthermore, it also delivers design theories that extend and generalize the knowledge contribution from a scientific perspective. DSR also contributes practically by delivering actionable innovative solutions that solve real-world problems with prescriptive knowledge.

Despite the huge potentials and increasing impacts of DSR, there is currently no comprehensive collection of successful DSR cases available. This is regrettable because practitioners and scientists, who want to apply DSR are confronted with numerous questions regarding planning and implementation of comparable projects. Exemplary DSR cases offer opportunities to learn from documented experiences of others, and, as such, they complement existing sources.

This book provides a collection and documentation of DSR cases provided by experienced researchers in the field. It gives access to real-world DSR studies together with the reflection of the authors on their research process. These cases support researchers who want to engage in DSR with a valuable addition to existing introductions to DSR methods and processes. Readers will learn from the valuable experiences of a wide range of established colleagues who have extensively conducted DSR in many application contexts.

Moreover, the book also aims to increase the exchange of knowledge in the DSR field, as well as to invite colleagues to engage in this promising form of research. Specifically for IS researchers who would like become familiar with DSR, this book provides many examples illustrating how to plan and conduct DSR. These examples provide both inspiration and a source of reference. The book also showcases the range of DSR projects and gives an overview of colleagues highly active in the field.

Each chapter follows a unified presentation structure that makes the relevant case knowledge easily accessible and transferrable to other contexts:

- **Introduction**: A brief narrative of the entire story to grasp interest in the case is provided.
- **Context**: This section describes the business or the societal context, so that readers can relate the findings to their own context.
- **Journey**: DSR projects typically do not follow not a linear process, but rather a journey of continuous refinement of both problem and solution understanding. In this section of the case this journey is described. Here the DSR process is overviewed with an emphasis on the different types of activities conducted during the DSR project. Specifically, iterations of problem and solution understanding during the design process are presented.
- **Results**: The key results of the journey are documented, covering both scientific and practical contributions.
- **Key Learnings**: Finally, reflections and learnings made during the reported DSR project are documented. Notable successes and key limitations of the research are addressed. Future directions can be provided.

With the unified structure of each case, we hope to support readers effectively accessing the most relevant parts to build on in their own DSR work. The material presented in this book is complemented by online material for teaching, training, and consulting. The website http://www.dsr-cases.com makes available slides and additional content that can be helpful for using the cases both in teaching DSR and in preparing for DSR projects in practice.

We thank the following people and institutions for their continuous support toward the compilation of this book. First, we thank our research teams, specifically Charlotte Wehking and Michael Gau from the University of Liechtenstein and ……. from Karlsruhe Institute of Technology.

We hope you will enjoy reading the book and learning from these DSR cases. We look forward to your feedback on how best to share knowledge and learning from DSR projects.

Vaduz, Liechtenstein — Jan vom Brocke
Tampa, USA — Alan Hevner
Karlsruhe, Germany — Alexander Maedche

Contents

Introduction to Design Science Research

Jan vom Brocke, Alan Hevner, and Alexander Maedche

Abstract Design Science Research (DSR) is a problem-solving paradigm that seeks to enhance human knowledge via the creation of innovative artifacts. Simply stated, DSR seeks to enhance technology and science knowledge bases via the creation of innovative artifacts that solve problems and improve the environment in which they are instantiated. The results of DSR include both the newly designed artifacts and design knowledge (DK) that provides a fuller understanding via design theories of why the artifacts enhance (or, disrupt) the relevant application contexts. The goal of this introduction chapter is to provide a brief survey of DSR concepts for better understanding of the following chapters that present DSR case studies.

1 Introduction to Design Science Research

The Design Science Research (DSR) paradigm has its roots in engineering and the sciences of the artificial (Simon, 1996). It is fundamentally a problem-solving paradigm. DSR seeks to enhance human knowledge with the creation of innovative artifacts and the generation of design knowledge (DK) via innovative solutions to real-world problems (Hevner, March, Park, & Ram, 2004). As such, this research paradigm has generated a surge of interest in the past twenty years, specifically due to its potential to contribute to fostering the innovation capabilities of organizations as well as contributing to the much needed sustainability transformation of society (Watson, Boudreau, & Chen, 2010; vom Brocke, Watson, Dwyer, Elliot, & Melville, 2013; vom Brocke, Winter, Hevner, & Maedche 2020).

J. vom Brocke (✉)
Department of Information Systems, University of Liechtenstein, Vaduz, Liechtenstein
e-mail: jan.vom.brocke@uni.li

A. Hevner
Muma College of Business, University of South Florida, Tampa, FL, USA

A. Maedche
Department of Information Systems, IISM - Information Systems and Marketing, Karlsruhe Institute of Technology, Karlsruhe, Germany

J. vom Brocke et al. (eds.), *Design Science Research. Cases*, Progress in IS,
https://doi.org/10.1007/978-3-030-46781-4_1

The goal of a DSR research project is to extend the boundaries of human and organizational capabilities by designing new and innovative artifacts represented by constructs, models, methods, and instantiations (Hevner et al., 2004; Gregor & Hevner, 2013). DSR aims to generate knowledge of how things can and should be constructed or arranged (i.e., designed), usually by human agency, to achieve a desired set of goals; referred to as design knowledge (DK). For example, DK in the Information Systems (IS) discipline includes knowledge of how to structure and construct a database system, how to model business processes, how to align IS with organizational strategy, how to deliver data analytics for effective decision making (e.g. Becker et al., 2015), as well as how to use information technology to support sustainable practices (Seidel et al., 2013; vom Brocke & Seidel, 2012a, b). DSR results in IS have been shown to create significant economic and societal impact (Gregor & Hevner, 2013; vom Brocke et al., 2013). Beyond the IS field, DSR is a central research paradigm in many other domains including engineering, architecture, business, economics, and other information technology-related disciplines for the creation of novel solutions to relevant design problems.

In the following, we introduce some essential frameworks and conceptualizations that we deem important in order to provide foundations on how to conduct DSR to scholarly standards. The cases presented in this book use such fundamentals in order to structure and document their DSR projects.

2 The DSR Framework

Figure 1 presents a conceptual framework for understanding, executing, and evaluating design science research (Hevner et al. 2004). The environment defines the problem space in which the phenomena of interest reside. It is composed of people, organizations, and existing or planned technologies. In it are the goals, tasks, problems, and opportunities that define needs as they are perceived by stakeholders within the organization. Needs are assessed and evaluated within the context of organizational strategies, structure, culture, and existing work processes. They are positioned relative to existing technology infrastructure, applications, communication architectures, and development capabilities. Together these define the "research problem" as perceived by the researcher. Framing research activities to address real stakeholder needs assures research relevance. The knowledge base provides the raw materials from and through which DSR is accomplished. The knowledge base is composed of Foundations and Methodologies. Prior research and results from reference disciplines provide foundational theories, frameworks, instruments, constructs, models, methods, and instantiations used in the build phase of a research study. Methodologies provide guidelines used in the evaluate phase. Rigor is achieved by appropriately applying existing foundations and methodologies.

DSR studies relevant problems in the real-world environment with various application domains. Research links to a "need" for solutions to be empirically investigated with people in organizations using specific technology. Often, the analysis

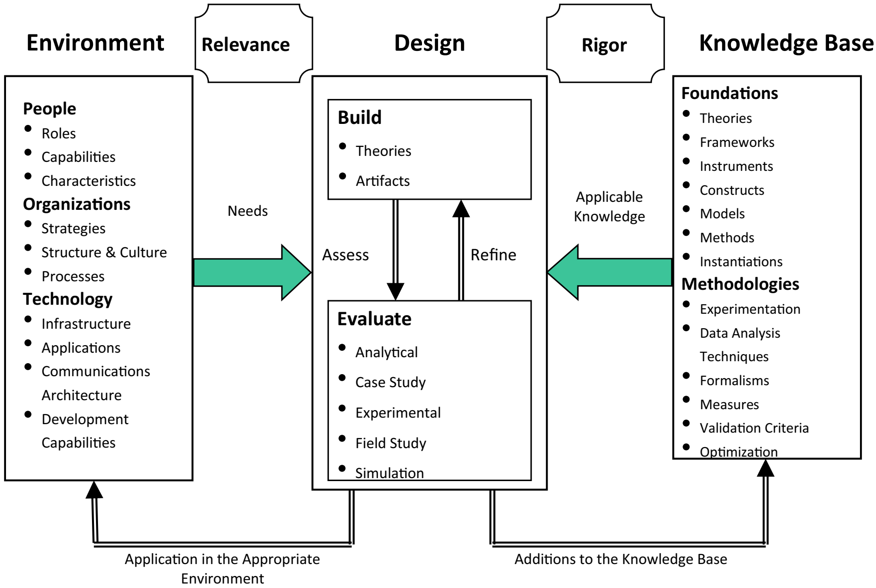

Fig. 1 Design science research framework (Adapted from (Hevner et al., 2004))

of the business environment and the derivation of specific needs to be solved build the starting point of a DSR project. However, also situations exist in which needs have already been studied and can be taken from extant research. DSR analyses the (academic) knowledge base in that it studies to which extent design knowledge is already available to solve a problem of interest. Such knowledge can take the form of theories, frameworks, instruments or design artifacts, such as constructs, models, methods or instantiations. In case knowledge is already available to solve a problem identified, this knowledge can be applied following "routine design", which does not constitute DSR. Else, DSR sets out to create an innovative solution to the problem, which, in most cases, builds on existing parts of a solution and combines, revises, and extends extant design knowledge. The design activities comprise of "build" and "evaluate" activities, typically following multiple iterations. In course of a DSR study, diverse research methods are applied, including those well established in social science research, such as interviews, surveys, literature reviews, or focus groups.

3 DSR Process

The performance of DSR projects has been based on several process models, such as Nunamaker, Chen and Purdin (1991), Walls, Widmeyer and El Sawy (1992), Hevner (2007), Kuchler and Vaishnavi (2008). The mostly widely referenced model is one proposed by Peffers et al. (2007). The design science research methodology (DSRM) process model is shown in Fig. 2. This DSR process includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication; and four possible entry points: problem-centered initiation, objective-centered solution, design and development-centered initiation, and client/context initiation. A brief description of each DSR activity follows.

Activity 1. Problem identification and motivation. This activity defines the specific research problem and justifies the value of a solution. Justifying the value of a solution accomplishes two things: it motivates the researcher and the audience of the research to pursue the solution and it helps the audience to appreciate the researcher's understanding of the problem. Resources required for this activity include knowledge of the state of the problem and the importance of its solution.

Activity 2. Define the objectives for a solution. The objectives of a solution can be inferred from the problem definition and knowledge of what is possible and feasible. The objectives can be quantitative, e.g., terms in which a desirable solution would be better than current ones, or qualitative, e.g., a description of how a new artifact is expected to support solutions to problems not hitherto addressed. The objectives should be inferred rationally from the problem specification.

Activity 3. Design and development. An artifact is created. Conceptually, a DSR artifact can be any designed object in which a research contribution is embedded in the design. This activity includes determining the artifact's desired functionality and its architecture and then creating the actual artifact.

Activity 4. Demonstration. This activity demonstrates the use of the artifact to solve one or more instances of the problem. This could involve its use in experimentation, simulation, case study, proof, or other appropriate activity.

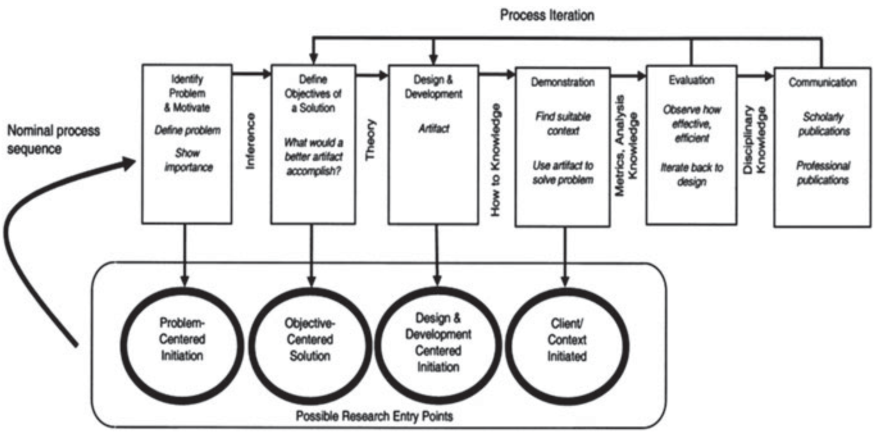

Fig. 2 DSR methodology process model (Adapted from Peffers et al. (2007))

Activity 5. Evaluation. The evaluation measures how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution to actual observed results from use of the artifact in context. Depending on the nature of the problem venue and the artifact, evaluation could take many forms. At the end of this activity the researchers can decide whether to iterate back to step three to try to improve the effectiveness of the artifact or to continue on to communication and leave further improvement to subsequent projects.

Activity 6. Communication. Here all aspects of the problem and the designed artifact are communicated to the relevant stakeholders. Appropriate forms of communication are employed depending upon the research goals and the audience, such as practicing professionals.

4 DSR Evaluation

The process of conducting DSR has been further developed in many ways, specifically paying attention to the evaluation activities and allowing for a more concurrent and fine-grained evaluation of intermediate steps in the design process. While it is well-understood that also the Peffers et al. (2007) process should and would be conducted iteratively, evaluation only takes place after design, development and demonstration activities; missing out on the opportunity to inform the design in an early stage of the research process.

Sonnenberg and vom Brocke (2012) conceptualize concurrent evaluation according to different aspects of design as shown in Fig. 3. They build on prior work describing DSR activities within the overall DSR process, arguing that each of these activities progresses toward the intended artefacts differently and thus offer potential for concurrent (or formative) evaluation. Such evaluation can mitigate risk (Venable, vom Brocke, & Winter, 2019), as early feedback on the minute steps leading to the eventual artefact can be incorporated into the design process. The authors also assert that this type of evaluation can be more specific and better directed if the evaluation focuses on the different aspects of design when relevant decisions are being made during the design process.

To demonstrate, Sonnenberg and vom Brocke (2012) identify four evaluation types (Eval 1 to Eval 4) derived from typical DSR activities. Figure 3 shows a cyclic high-level DSR process that includes the activities of problem identification, design, construction, and use. In addition, Fig. 3 suggests that each DSR activity is followed by an evaluation activity, as follows:

- 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

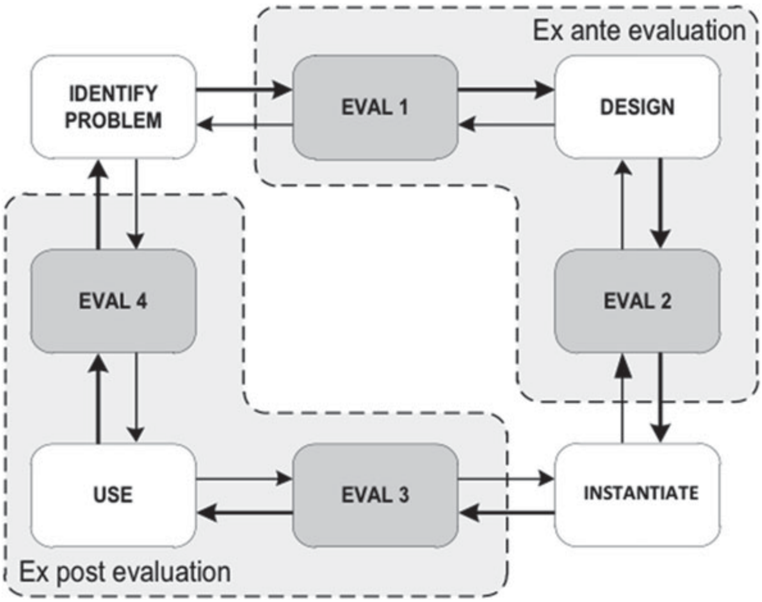

Fig. 3 Evaluation activities within the DSR process (Adapted from Sonnenberg and vom Brocke (2012))

- Eval 4: Evaluating the solution in use; criteria include effectiveness, efficiency, and external consistency.

Depending on when an evaluation occurs, *ex ante* and *ex post* evaluations are distinguished. *Ex ante* evaluations are conducted before the instantiation of any artefacts, while *ex post* evaluations occur after the instantiation of any artefact (Venable, Pries-Heje, & Baskerville, 2016). The DSR process in Fig. 3 indicates that there are feedback loops from each evaluation activity to the preceding design activity. Overall, these feedback loops together form a feedback cycle that runs in the opposite direction to the DSR cycle.

5 Design Knowledge Framework

The design knowledge (DK) produced in a DSR project can be richly multifaceted. DK will include information about the important problem, the designed solution, and the evaluation evidence. Specifically it includes measures of timely progress on how well the problem solution satisfies the key stakeholders of a problem.

We consider these three components to constitute DK: the problem space, the solution space, and the evaluation. While we understand that both problem space knowledge and solution space knowledge exists independently, it is only through putting them in relation to one another that we refer to the respective knowledge as DK. Figure 4 provides a simple model conceptualizing important components of DK.

Information systems research consumes and produces two basic types of knowledge: (1) behavioral science-oriented research activities primarily grow propositional knowledge or Ω-knowledge (comprising descriptive and explanatory knowledge), and, (2) DSR-oriented research activities primarily grow applicable (or prescriptive) knowledge or λ-knowledge (Gregor & Hevner, 2013). Contributions to the λ knowledge base typically comprise knowledge about technological (i.e. digital) innovations that directly affect individuals, organizations, or society while also enabling the development of future innovations (Winter & Albani, 2013). Contributions to the Ω knowledge base enhance our understanding of the world and the phenomena our technologies harness (or cause). Research projects may combine both paradigms of inquiry and contribute to both knowledge bases.

The relationships of design knowledge produced and consumed in DSR projects and the (design) knowledge bases are shown in Fig. 1. This figure is adapted and simplified from (Drechsler & Hevner, 2018) and clearly illustrates paired modes of consuming and producing knowledge between the DSR project and the Ω and λ knowledge bases. The λ-knowledge is further divided into two sub-categories. The *Solution Design Entities* collect the prescriptive knowledge as represented in the tangible artifacts, systems, and processes designed and applied in the problem solution space. The growth of design theories around these solutions is captured in the *Solution Design Theories* knowledge base (Gregor & Hevner, 2013). Knowledge can be projected from the specific application solutions into nascent theories around

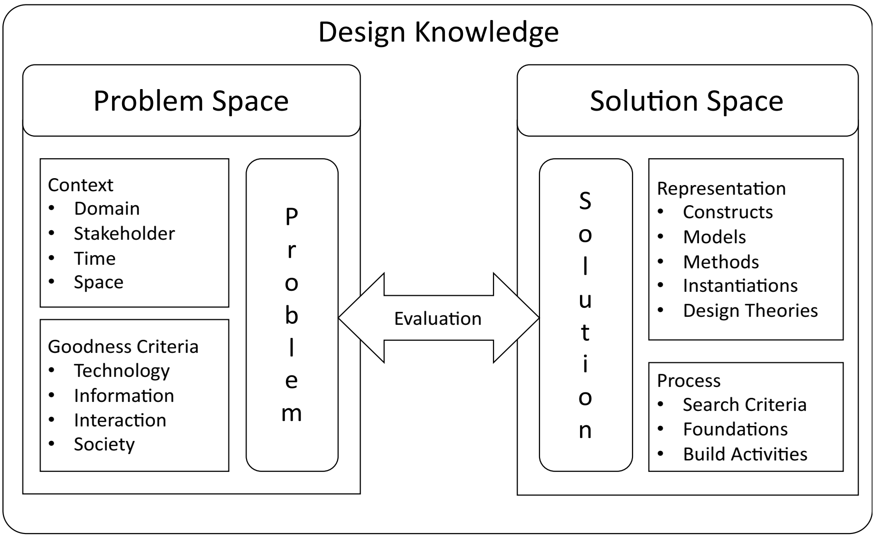

Fig. 4 Components of design knowledge for a specific DSR project

solution technologies, actions, systems, and design processes based on the new and interesting knowledge produced in a DSR project. Thus, we can describe the interactions of a specific DSR project with the extant knowledge bases in the following consuming and producing modes (Fig. 5):

- Descriptive (Ω) Knowledge: Ω-knowledge (or kernel knowledge) informs the understanding of a problem, its context, and the underlying design of a solution entity (Arrow 1). As results of the research project, the design and real-world application of solution entities or design knowledge enhances our descriptive understanding of how the world works via the testing and building of new Ω-knowledge (Arrow 2).
- Prescriptive (λ) Solution Design Entities: Existing solution entities, design processes, or design systems are re-used to inform novel designs of new entities, processes, or systems (Arrow 5) (vom Brocke & Buddendick, 2006). Within a DSR project, effective solution entities, design processes, or design systems are produced and contributed to new λ-knowledge (Arrow 6).
- Prescriptive (λ) Solution Design Theories: Solution design knowledge, in the form of growing design theories, informs the design of a solution entity, a design process or a design system (Arrow 3). Within a DSR project, effective principles, features, actions, or effects of a solution entity or a design process or system are generalized and codified in solution design knowledge (e.g. design theories or technological rules) (Arrow 4).

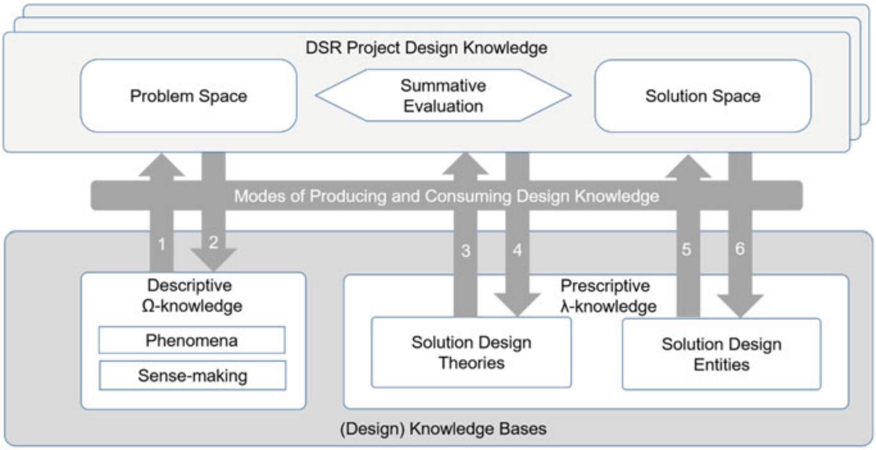

Fig. 5 DSR projects and modes of producing and consuming design knowledge (Adapted from Drechsler and Hevner (2018))

6 Three Types of Design Science Projects

In simple terms, a DSR project can make two types of contributions—it can contribute to design entities or to design theory—and conducting design processes in search of solutions to prob-lems and theorizing about such processes are what lead to these contributions (vom Brocke & Maedche, 2019). The two type of contributions and related activities are illustrated in Fig. 6.

Early contributions to DSR focused on contributions to design entities (e.g., Hevner et al. 2004; Peffers et al. 2007). Gregor and Jones (2007) introduce the idea of DSR projects' producing design theory and conceptualize the anatomy of a design theory by means of six core components: purpose and scope, constructs, principle of form and function, artifact mutability, testable propositions, and justificatory knowledge. Gregor and Hevner (2013) outline how both types of contributions relate to each other and how a DSR project can go beyond the design of design entities to contribute to design theory by theorizing about the design science process and the evaluation result achieved.

More recently, Chandra-Kruse, Seidel and vom Brocke (2019) suggest a third type of DSR project that builds on design processes that are not conducted as part of the DSR project itself but at another place and time. Such research opens DSR projects up to theorize about design that is not motivated by research but by something that happened in, for example, industry or society. Drawing from archeology research, researchers have described methods for investigating design processes and artifacts empirically to generate DK. In short, three types of DSR projects can be differentiated regarding the contribution they intend to make to DK: (1) projects that contribute to design entities, (2) projects that contribute to both design entities and design theory, and (3) projects that contribute to design theory without developing a design entity as part of the same project.

Given the complexity of DSR projects and the various ways a DSR project might contribute to DK, how comprehensively and effectively a DSR project is planned and communicated can affect its likelihood of success. Such planning and communication enables researchers to reflect on and receive feedback about their DSR project in its early stages and to question and update their scope as they progress in the project.

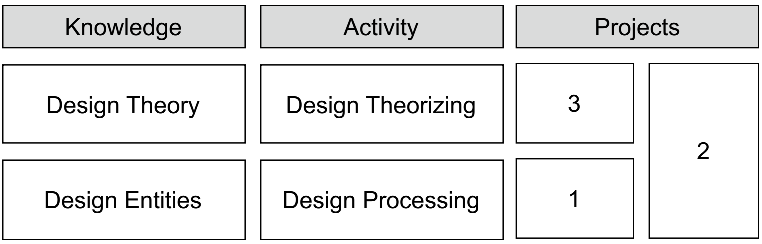

Fig. 6 DSR Projects' contributions to design knowledge

7 The Design Science Research Grid

The DSR Grid (vom Brocke & Maedche, 2019) enables researchers to effectively plan, coordinate and communicate their DSR projects. The DSR grid intends to put an entire DSR project on one page, highlighting its essential components in order to reflect and communicate its scope. Such representation of a DSR project helps to better plan and communicate a DSR project as well as to receive feedback from different stakeholders in an early stage and to question and update the scope as the project progresses. As shown in Fig. 7, the DSR Grid consists of the six most important dimensions of a DSR project.

Problem Description: What is the problem for which a DSR project must identify possible solution? Problems should be formulated by means of problem statements and characterized by positioning the problem in a problem space. Research has identified the context, described by the domain, the stakeholder, time and place, and goodness criteria, the last of which tells when a problem should be considered solved, as necessary to capture the problem appropriately (vom Brocke et al. 2020).

Input Knowledge: What prior knowledge will be used in the DSR project? As introduced above one can distinguish Ω-knowledge and λ-knowledge, the first being descriptive, explanatory, or predictive, and the second being prescriptive (Gregor & Hevner, 2013). Three types input—kernel theories, design theories, and design entities—can be differentiated for high-level communication about DSR projects.

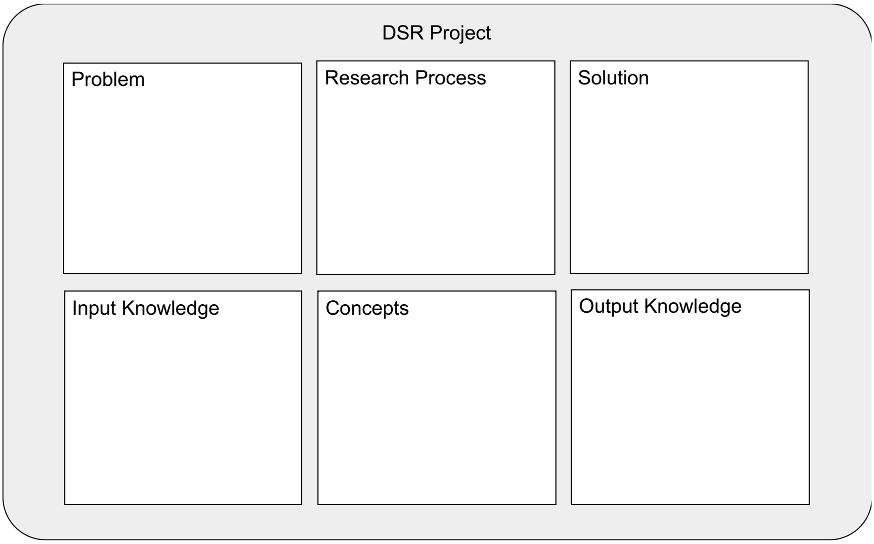

Fig. 7 DSR grid comprised of the six core dimensions of a DSR project

Research Process: What are the essential activities planned (or conducted) to make the intended contribution? When the intended contribution is design entities, the process includes build and evaluate activities (Hevner et al., 2004). In particular, these activities also include grounding the design (vom Brocke et al., 2020) by, for example, conducting literature reviews (Webster & Watson, 2002, vom Brocke et al. 2015), and meta-analysis (Denyer, Tranfield & Van Aken, 2008). In order to support concurrent design and evaluation, it is suggested to plan and document the build and evaluation activities in one. DSR tools have been developed (vom Brocke et al., 2017; Morana et al., 2018) to keep logs of the research process; such logs can complement a high-level list of research activities used to scope the DSR project in the process dimension. The process documented here may also include activities for theorizing about the design. While activities for processing the design can draw from DSR process models like the Peffers et al. (2007) model, activities for theorizing can draw from various research methods and strategies of inquiry, such as qualitative and quantitative empirical research.

Key concepts: What are the most important concepts used in the research performed in the DSR project? The words used to describe the research, such as the problem and solution space that the DSR project focuses on, as well as the concepts used to describe the process and input and output knowledge must be defined clearly. A clear definition of the key concepts is particularly important to ensure a rigorous execution of the evaluation activities.

Solution Description: What is the solution to the problem being investigated by a DSR project? The solution description clearly states the essential mechanisms of the solution (vom Brocke et al. 2020) and how the solution is positioned in solution space by characterizing its representation as a construct, a model, a method, an instantiation, or a design theory.

Output Knowledge: What knowledge is produced in the DSR project? Naturally, DSR projects produce DK, classified as λ-knowledge (Gregor & Hevner, 2013), but in contrast to the solution description, the DK generated through the project puts the problem and solution spaces in relation to each other (vom Brocke et al. 2020). If a DSR project does not intend to generate design theory but to generate design entities, the description of such entities does not constitute DK, as it is only the results of the design entity's evaluation in context that constitute DK. These results are then documented as output knowledge when the project is described.

Factors like the phase of the project (e.g., early planning or documenting completed research) and the stakeholder group (e.g., industry partners or editors) determine the perspectives from which and the detail with which the six dimensions may be described. Multiple versions of the dimensions will usually be created in iterations as a project progresses, but referring to the dimensions helps researchers to consider the core aspects of a DSR project from the outset and to discuss these aspects with stakeholder groups to shape the project's profile further as it goes along.

8 Conclusion

In this chapter, some important DSR concepts and models have been presented to provide a foundation for the planning, performing, and disseminating DK from specific DSR projects. In the following chapters, cases of DSR projects are presented as conducted by experienced researchers in the field. These cases serve as examples from which to learn in order to inform one's DSR projects. These case studies provide invaluable experiential knowledge of how fellow researchers have conducted DSR over the past decades. This case collection is intended to "live" in that we are always very happy to include new cases of diverse application environments. The richer the collection, the more useful for the community. Apart from enjoying to read the cases in the book, authors are cordially invited to get in touch and discuss how to add their own case to this collection.

References

Becker, J., vom Brocke, J., Heddier, M., & Seidel, S. (2015). In search of information systems (Grand) challenges: a community of inquirers perspective. *Business & Information Systems Engineering (BISE), 6*(1), 39–43.

Chandra-Kruse, L., Seidel, S., & vom Brocke, J. (2019). Design archaeology: Generating design knowledge from real-world artifact design. Paper presented at the 14th International Conference on Design Science Research in Information Systems and Technology, Worcester, MA.

Denyer, D., Tranfield, D., & van Aken, J. E. (2008). Developing design propositions through research synthesis. *Organization Studies, 29*(3), 393–413.

Drechsler, A., & Hevner, A. R. (2018). Utilizing, producing, and contributing design knowledge in DSR projects. In S. Chatterjee, K. Dutta, & R. Sundarraj (Eds.), *Lecture Notes in Computer Science* (Vol. 10844, pp. 82–97)., Designing for a Digital and Globalized World Cham, Switzerland: Springer.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association For Information Systems, 8*(5), 312–335.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337–355.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems, 19*(2), 87–92.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems. *MIS Quarterly, 28*(1), 75–105.

Morana, S., Scheid, M., Gau, M., Benke, I., vom Brocke, J., Fettke, P., & Maedche, A. (2018). Research Prototype: The Design Canvas in MyDesignProcess.com, in: DESRIST 2018 Conference Proceedings.

Nunamaker, J. F., Chen, M., & Purdin, T. D. (1991). Systems development in information systems research. *Journal of Management Information Systems, 7*(3), 89–106.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77.

Seidel, S., Recker, J., & vom Brocke, J. (2013). Sensemaking and sustainable practicing: functional affordances of information systems in green transformations. *Management Information Systems Quarterly (MISQ), 37*(4), 1275–1299.

Simon, H. A. (1996). *The sciences of the artificial*. Cambridge, MA: MIT Press.

Sonnenberg, C., & vom Brocke, J. (2012). Evaluations in the Science of the Artificial: Reconsidering the Build-evaluate Pattern in Design Science Research. In K. Peffers, M. Rothenberger, & B. Kuechler (Eds.), *Design science research in information systems and technology 2012, advances in theory and practice (Lecture Notes in Computer Science)* (pp. 381–397). NV, USA, Springer, Berlin, Heidelberg: Las Vegas.

Venable, J. R., Pries-Heje, J., & Baskerville, R. L. (2016). FEDS: A framework for evaluation in design science research. *European Journal Information Systems, 25*(1), 77–89.

Venable, J., vom Brocke, J., & Winter, R. (2019), Designing TRiDS: Treatments for risks in design science. *Australasian Journal of Information Systems (AJIS)*, June 2019.

vom Brocke J., Seidel S. (2012a) Environmental sustainability in design science research: direct and indirect effects of design artifacts. In: K. Peffers, M. Rothenberger, B. Kuechler (eds) Design Science Research in Information Systems. Advances in Theory and Practice. DESRIST 2012. Lecture Notes in Computer Science, vol 7286. Springer, Berlin, Heidelberg.

vom Brocke, J., & Seidel, S. (2012b). Environmental Sustainability in Design Science Research: Direct and Indirect Effects of Design Artifacts. Paper presented at the 7th international conference on Design Science Research in Information Systems, Las Vegas, USA.

vom Brocke, J., Fettke, P., Gau, M., Houy, C., Maedche, A., Morana, S., & Seidel, S. (2017). Tool-Support for Design Science Research: Design Principles and Instantiation (May 23, 2017). Available at SSRN: https://ssrn.com/abstract=2972803 or https://doi.org/10.2139/ssrn.2972803.

vom Brocke, J., & Maedche, A. (2019). The DSR grid: Six core dimensions for effectively planning and communicating design science research projects. *Electronic Markets, 29*(3), 379–385.

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 the association for information systems. *Article, 37*(1), 205–224. Article 9.

vom Brocke, J., Watson, R., Dwyer, C., Elliot, S., & Melville, N. (2013). Green information systems: Directives for the IS discipline. *Communications of the Association for Information Systems (CAIS), 33*(30), 509–520.

vom Brocke, J., Winter, R., Hevner, A., & Maedche, A. (2020). Accumulation and evolution of design knowledge in design science research—a journey through time and space. *Journal of the Association for Information Systems (JAIS), forthcoming*.

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.

Watson, R. T., Boudreau, M.-C., & Chen, A. J. W. (2010). Information systems and environ-mentally sustainable development: Energy Informatics and new directions for the IS com-munity. *MIS Quarterly, 34*(1), 23–38.

Webster J. and Watson R.T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2), xiii–xxiii.

Winter, R., & Albani, A. (2013). Restructuring the design science research knowledge base: A onecycle view of design science research and its consequences for understanding organizational design problems. In R. Winter & A. Albani (Eds.), *Lecture Notes in Information Systems and Organisation* (Vol. 1, pp. 381–397)., Designing Organizational Systems: An Interdisciplinary Discourse Heidelberg, Germany: Springer.

DSR in Information Management

A Novel Approach to Collectively Determine Cybersecurity Performance Benchmark Data

Aiding Organizational Cybersecurity Assessment

Richard Baskerville and Vijay Vaishnavi

Abstract How do we determine cybersecurity performance benchmark data across organizations without the organizations revealing data points involving their frequency of information compromises? Disclosures of underlying data points are fundamentally inhibited by the need for privacy. It is a responsible organizational action to prevent the risk of expected or unexpected damages through data disclosure. Obtaining the data, especially valid and reliable data, necessary to calculate benchmarks, was thus an unsolvable wicked problem. The problem was therefore redefined as: How do we enable a distributed power-base of cybersecurity managers across organizations to c*ollectively* determine their benchmark data *without actually disclosing their own data points*? The core of the solution is a simple creative idea of having a protocol for a network of organizations to calculate benchmarks by distributing such calculations starting with some obfuscating data instead of centrally collecting the constituent data points of each organization. In this way, the confidential data of the organization would never be released beyond organizational systems. The fuller development of the protocol faced the issues of establishing trust in the network and preventing statistical compromises that were addressed through creative use of available technology, leading to the final solution, a distributed peer-to-peer architecture called Trusted Query Network (TQN). Distributed processing can induce trust and privacy into computer networks. In addition: (1) A research group representing multiple strengths and different but complementary backgrounds can be a very powerful asset in solving difficult problems. (2) Use of creativity is central to design science research but is particularly needed in solving apparently intractable problems. A group format can encourage free flow of ideas and brainstorming that are useful in spurring creativity. (3) It is very useful to be visionary in finding and solving challenging problems. Research groups provide the psychological strength to confront existing design challenges and to visualize their out-of-the-box solutions.

R. Baskerville (✉) · V. Vaishnavi
Georgia State University, Atlanta, Georgia
e-mail: baskerville@acm.org

V. Vaishnavi
e-mail: vvaishna@gsu.edu

J. vom Brocke et al. (eds.), *Design Science Research. Cases*, Progress in IS,
https://doi.org/10.1007/978-3-030-46781-4_2

1 Introduction

How well are today's organizations protecting their confidential information? As important as the answer to this question is, we don't really know the answer. It has been historically impossible to answer such a question because the underlying data is impossible to collect. Such a collection is impossible because the individual data points involve revealing the frequency of information compromises in organizations. Organizations are understandably reticent when it comes to admitting their information security compromises. As a result, we can only guess about the general status of our cybersecurity efforts. For example, the Privacy Rights Clearinghouse reports details of more than 10 billion compromised records since 2005.[1] This data comes from government reports or verifiable media sources. In an age of big data, is our average exposure of 800 million records annually a record that is spectacularly bad or spectacularly good? The data reveals that 14 publicly known breaches at Bank of America have exposed 1,894,326 records. By comparison, nine publicly known breaches at Citigroup have exposed 4,769,338 records. Does this mean that Citi (with annual revenue of around US$70b) is more careless than BoA (with annual revenue of around US$90b)? Instead, does it simply mean that Citi has been forced to disclose publicly more of its exposures than BoA? What if both of these records are much better than the average of actual exposures at all banks?

We cannot answer these questions because we lack benchmarks for cybersecurity performance. We lack such benchmarks because companies are understandably reluctant to admit their actual cybersecurity performance. What makes this problem worse is that these companies cannot know themselves whether their performance is comparably better than, or worse than, the cybersecurity performance benchmarks for their industry. Disclosures of underlying data points are fundamentally inhibited by the need for privacy. Disclosure inhibition is an inner impediment to the free expression of information. It is a responsible organizational action to prevent the risk of expected or unexpected damages through data disclosure. Thus the fundamental problem is *The Law of Private Data Disclosure* (Vaishnavi, Vandenberg, Baskerville, & Zheng, 2006):

$$\text{Private Data Disclosure} \Rightarrow \text{Risk}$$

That is, any information disclosure implies a risk to the discloser. The risk may vary in scale from trivial to fatal, but any disclosure involves risk. As a result few organizations share information about their cybersecurity breaches because such information is so sensitive (Vance, Lowry, & Wilson, 2016).

Under today's management theory, the capability to manage high quality processes depends on the availability of good metrics to guide decision making. It is a completely simple notion, like a speedometer for helping a driver manage the speed of the vehicle. The meter indicates speed, and the driver makes informed decisions

[1] Data reviewed on 29 December 2017 at https://www.privacyrights.org/data-breaches.

whether to go faster or slower. Of course, a poor manager may make poor decisions in terms of the metrics, just as a poor driver may precipitate a collision or get arrested.

Dedicated cybersecurity managers, such as a CISO (Chief Information Security Officer) need more information than just the organizational cybersecurity performance. They need cybersecurity performance benchmarks that provide reference points for their performance. Is the organization's number of cybersecurity incidents better or worse than similar organizations? Is the CISO and the security department doing a good job or a poor job relative to their peers? It is similar for driving. Drivers need some indication about how fast is too fast and how slow is too slow. The most obvious speed benchmark for drivers is a speed limit or speed recommendation. For drivers, these speed benchmarks are based on laws or road designs.

Of course, benchmarks for the metrics for cybersecurity managers are more complex. These are rarely set by laws or environmental designs, and more often based on comparative performance of similar organizations. For example, a measure like the rate of return on investment (ROI) may be regarded "good" or "bad" depending on comparative benchmarks. If an investment manager achieves an ROI of 6%, and the average ROI for other comparative investment managers is 8%, then it suggests the manager is making poor decisions and has room for improvement. On the other hand, if the average ROI for other comparative investment managers is 4%, then the 6% investment manager is making good decisions that could lead others to improve.

It is the value of the benchmarks as well as the performance metrics that most help managers to know if their decision making has been good or bad, better or worse, in comparison with other managers facing similar decisions. A focus purely on organizational metrics only solves part of the guidance issues for management decisions. It is the benchmarks that help determine the goals for the metrics values. The calculation of benchmarks across comparative organizations is often further complicated by the confidentiality of the underlying measures. For example, cybersecurity managers may want to use a metric such as the number of server compromises per month. Suppose the metric measure for December is 21. Is this good or bad? The cybersecurity manager can compare to November or January, but this is not as useful as accurately knowing what would be typical for this measure in other organizations. Is this number spectacularly high; or is it a tiny fraction of that normally found in other organizations? It is the benchmark that helps the cybersecurity manager decide if the server compromises are being overmanaged or undermanaged.

2 The Context

The problem with confidentiality of the measures most desirable for benchmarks is the extreme sensitivity of the metrics values for each organization. There has been some operational success with industry-based Information Sharing and Analysis Centers (ISACs)[2] that exchange threat and mitigation information. There has also

[2]https://www.nationalisacs.org/ (last accessed on 8 March 2018).

been similar success with a law-enforcement-based (i.e., FBI) operational information exchange intended to protect critical infrastructures (Infragard[3]). However, none of these organizations track organizational cybersecurity performance information. Attempts to create central databases for such confidential data have not only run afoul of trust, but also risks of legal discovery, and freedom-of-information laws. Examples include Purdue University Cerias Incident Response Data Base (CIRDB)[4] project[5] and the CIFAC project (Rezmierski, Rothschild, Kazanis, & Rivas, 2005). The purpose of these systems is to manage this sensitive point data centrally, rather than sharing benchmark data at a collective level.

The specific setting for this problem was a U.S. state university system that confronted the need to assess the cybersecurity performance across its 30 constituent universities and colleges. Even when the cybersecurity breach reporting was made mandatory, little data was collected, partly because the cybersecurity managers in the various institutions did not know whether they were confessing to incompetence or bragging about their competence. Members of the university system originally formed a team to investigate the creation of a national ISAC-like system for collecting and reporting cybersecurity performance in higher education. Eventually this effort gave way to a recognition that obtaining the data, especially valid and reliable data, necessary to calculate benchmarks was a wicked problem. It involved multiple, conflicting criteria at different levels of the data collection and disclosure process. No one would willingly divulge their own data points until they had the opportunity to compare their own data points to the collective benchmark. Based on this observation, the problem was redefined as:

> How do we enable a distributed power-base of cybersecurity managers to collectively determine their benchmark data without actually disclosing their own data points?

The redefined problem operated under a fundamental assumption that cybersecurity managers, who are distributed across the population, would be motivated to improve their cybersecurity performance as soon as they learned that they are underperforming in relation to the benchmarks. As a natural outcome of the steadily improving quality of the underperforming sector in the population, the benchmarks will rise. As a natural outcome of the behavior above, the overall performance of the population will rise. In other words, the way to improve the cybersecurity performance of any sector is to develop and share benchmarks of performance.

3 The Journey

We will describe the design science research journey of this project in terms of Vaishnavi and Kuechler's general process model for design science research (Vaishnavi & Kuechler, 2015). This model describes an iterative process of problem awareness,

[3]https://www.infragard.org/ (last accessed on 8 March 2018).

[4]https://www.cerias.purdue.edu/site/news/view/cirdb_cassandra/ (last accessed on 8 March 2018).

[5]https://www.cerias.purdue.edu/site/about (last accessed on 12 March 2018).

solution suggestion, artifact development, evaluation, conclusion and knowledge flow circumscription. This model is depicted in Fig. 1.

Awareness of Problem. There can be multiple sources from which an awareness may arise of an interesting practical and research problem. The problem should be interesting because it is proving intractable. Intractable problems are those for which the solutions at hand are unsatisfying. Intractable problems are interesting when we discover that these problems are essentially not of the nature previously assumed. In the example above, we assume that the essence of the problem has to do with overcoming the unwillingness of cybersecurity managers to reveal unpleasant data. The problem becomes interesting when we assume the cybersecurity managers are behaving properly in withholding the data. It is a problem of helping them obtain the information they need to properly manage their cybersecurity operations.

Intractable problems are often interesting research problems because researchers may have been basing their knowledge on the wrong range of theories. Such a misalignment occurs because the practical problem has been misdiagnosed.

The output of the Awareness phase is a proposal, formal or informal, for a new design science research effort.

Suggestion. This phase follows from the proposal. Indeed, it is closely connected with awareness as indicated by the dotted line around proposal and tentative design. Both the Awareness and the Suggestion phases are likely to involve an *abductive reasoning process*. It is a reasoning process in which the designer observes the problem and then creates elements of the most likely solution (tentative design). This tentative design is the output of the Suggestion phase.

Development. In this phase, the tentative design is further developed and implemented. The implementation itself is sometimes very pedestrian. That is, it may not necessarily involve novelty or originality beyond the current state-of-the-art. The novel contribution is usually present in the artifact's design rather than in its

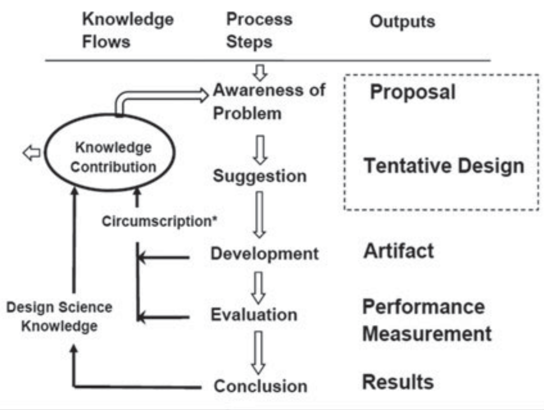

Fig. 1 Vaishnavi & Kuechler's design science research process model (2015, p. 13)

construction. Both the Development and the Conclusion phases (next) involve *deductive reasoning* in deducing the artifact's material characteristics from the tentative design.

Evaluation. In the Evaluation phase, the results of the artifact development are compared with the expectations that are either implicit or explicit in the Awareness and the Suggestion phases. When results embody essential deviations from expectations, we need tentative explanations to determine which further steps to follow next. These results and explanations often provide information that often helps refine our understanding of the problem, the utility of the suggestion, and the feasibility of the originally imagined artifact.

Conclusion. In terms of reasoning, this phase involves *reflection and abstraction.* We give consideration to the meaning of the more important and general outcomes of the previous phases. It is not necessary that the outcomes are optimal, only that they satisfice. But in producing these outcomes, we learn about the nature of the problem, the character of the solution, and the effect of the artifact. In other words, we not only seek to solve the problem, but also to learn about the environment that produces the problem and envelopes the solution-artifact.

Circumscription represents the major feedback loops that drive iteration in the design science research process. It is a rich notion about the common-sense value of knowledge and conjecture. McCarthy defined circumscription as "a rule of conjecture that can be used by a person or program for 'jumping to certain conclusions', namely … the objects they can determine to have certain properties or relations are the only objects that do" (McCarthy, 1980, p. 27). This aspect of the process informs us of the limits or boundaries of the knowledge discovered in each design science research cycle in two ways. First, we discover constraint knowledge about the theories underlying our design. This knowledge involves detecting and analyzing contradictions arising when things do not work according to theory. Second, a problem situation determines our awareness and suggestion, which in turn drive our conclusion to design and develop an artifact. In so doing, we create a new situation and we must again decide what to do. We create and use both knowledge about the specific situation and more general types of knowledge (like common sense). Accordingly, there are two types of arrows in the Fig. 1 representation of this process. Broad white arrows represent knowledge use, and narrow black arrows represent the generation of knowledge.

3.1 Lap 1—"Paper-Based Prototype"

Awareness of Problem. The problem and its origin, as it appeared to the design group, are described above in the introduction and the context sections. The important aspect of this awareness was the realization that it would not be possible to compel organizations to share such sensitive data. This aspect embodied our pre-theory as we approached the suggestion phase (Baskerville & Vaishnavi, 2016). An alternative was needed in which the benchmarks could be developed without any organizational disclosures.

Suggestion. At the design group's first meeting, the suggestion arose that it might be possible for a collective of organizations to calculate benchmarks by distributing such calculations instead of centrally collecting the constituent data points of each organization. In this way, the confidential data of the organization would never be released beyond organizational systems or organizational protection. Our solution to the underlying problem (The Law of Private Data Disclosure) was defined as:

(Disclose no private data) ˆ (Disclose only aggregate data)

The design group devised a paper-based experiment in which slips of paper would be passed around the group members and each person would individually calculate their data's impact on the benchmark.

Development. Because this initial experiment was a simple paper-based prototype it materialized something like a design walk-through. Our first experiment was calculating the average age of the group members without any one person revealing his/her age.

The first person imagined three people of different ages. He then totaled these three people's ages (obfuscating data) with his own. He wrote the number of people ($n = 4$) and their total age on a slip of paper and passed this slip to another (randomly chosen) member of the design team.

The second person added his age to the total and incremented n. He wrote the new total and the new value of n on a new slip of paper. He passed this slip to another (randomly chosen) member of the design team.

The third person similarly added her age and incremented n. She passed along a new slip of paper with her results to the next person, and so on, until all members of the design group had a pass at the calculation.

The paper was returned to the first person (the initiator) who subtracted the total age of the three imaginary people and reduced n accordingly by three. He then divided the remaining total by the remaining n, producing the average age of the design group without anyone actually revealing his/her own age.

This exercise was repeated several times, with ongoing discussions about how to compromise the protocol and how to calculate more complex values.

Evaluation. While this early paper-based experiment seemed very promising, it was easy to imagine very simple compromises. For example, the group could collude against any one of its number to detect that member's confidential data. Any member of the group could bias the results by misadjusting the values of partial results and n. Everyone in the *network* had to *trust* its members to be accurate and not collude. It was clear that control over the network membership and the initiation of a calculation round (a *query*) would still require one member of the network to act as a controller to insure members were not misrepresenting themselves or their data, and that the