Applying evaluations while building the artifact - Experiences from the development of process model complexity metrics

Daniel Braunnagel
University of Regensburg
daniel.braunnagel@uni-r.de

Susanne Leist
University of Regensburg
susanne.leist@uni-r.de

Abstract

The Design Science Research method is decisive for the quality of the resulting solution. Thus, many discussions focus the evaluation of the solution at the end of the Design Science cycle. But design, implementation and evaluation of artifacts are laborious and need to be repeated if the artifact does not meet the evaluation criteria. Thus, recent works have proposed to conduct additional evaluations early in the Design Science process to possibly reduce the number of repetitions of the research process. However, such early evaluations may also be an unnecessary burden. Therefore, this work presents a case where these additional evaluations are applied ex-post in a practical research project which developed process model complexity metrics and the outcomes are compared. Once compared, benefits and limitations of early evaluations are discussed.

1. Introduction

The application of Design Science Research (DSR) in Information Systems Research is currently an often discussed topic. Especially researchers are interested in improving the rigorous application of the method (cf. [1–3]). Many publications in DSR focus on the evaluation of the artifact. They define techniques or criteria to prove whether the developed artifact meets defined requirements and can be considered an optimal or adequate solution for the problem. Occasionally, researchers essentially follow a search process to find an effective solution for a problem. This search process forces them to conduct the build and evaluation phases in many iterations as the developed artifact has to be evaluated to identify whether the problem is solved [1]. If the proof fails, a new solution has to be developed, which will be evaluated again to see if it is more satisfying or optimal. This is referred to as the design cycle [1, 4].

For the aim of reducing the number of design cycles, a promising approach could be to focus on the Build phase and identify techniques or criteria to support the development of an optimal or at least satisfying solution. On the other hand, identifying and using these techniques or criteria is time consuming and only few researchers have been dealing with the definition, application or usefulness of such approaches to improve the build phase.

Against this background, the aim of the paper is to investigate in how far such an effort in the build phase can contribute to the development of the artifact and e.g. can reduce the number of design cycles.

We do so at hand of a design science (DS) project, which was previously conducted in the traditional way. For the current work, we repeat the build phase with the aid of evaluation techniques and discuss the impact techniques and criteria to support the development in the build phase have in this case.

In the previous DS project, we developed so called coupling metrics. They are used to assist process modelers with guidance about the quality of the models. The perspective of coupling evaluates the understandability in particular. (cf. [5])

For our evaluation at hand, this particular DS project is especially interesting. First, since the metrics are fully formalized and can be described within a paper, the influence which the different methods have on the artifact can be discovered clearly. Further, the development of the metrics is complex and the current research provides little guidance for design decisions, because of which the results of evaluations are not foreseeable during the build phase. *Sonnenberg and vom Brocke* [3] call this situation the emergent nature of knowledge in DS projects, because of which they propose to evaluate early and often along the DS process (cf. [3]). And since we had recently built the metrics with a traditional build and evaluate approach, we were highly interested if and how the results differ with another evaluation strategy.

The structure of our paper is as follows. Section 2 explains the basics, which comprise the Design Science and coupling metrics. Section 3 presents the methodology which we followed originally, the methodology which includes the additional early evaluation

activities and the corresponding evaluation criteria. In section 4 we explain how the evaluation was conducted in detail, as well as its result. Section 5 discusses the implications of performing additional, early evaluations and section 6 concludes our work.

2. Basics and related work

2.1 Design Science

The design science paradigm seeks for the enhancement of human and organizational capabilities by creating new and innovative artifacts. Emanating from engineering and the sciences of the artificial [6], it is concerned with the design, development, implementation, and use of socio-technical systems in organizational contexts. Design scientists produce and apply the knowledge about tasks or situations in order to create effective artifacts [7]. Thus, DSR is fundamentally a problem solving paradigm. A challenge in design science results from the fact that an artifact's performance depends on the environment in which it is used. An incomplete understanding of the environment can lead to inappropriately designed artifacts [7]. In consequence, the evaluation of the designed artifacts is particularly important.

Currently a variety of different approaches for the conduct of design science research can be found (cf. [8]), which all describe a process organized in the two phases build and evaluate (cf. [3, 9]). As a prominent example, *Peffers et al.* developed an approach which represents the synthesis of design science processes from IS and other disciplines and comprises six steps: (1) identify problem & motivate, (2) define objectives of a solution, (3) design & development, (4) demonstration, (5) evaluation, and (6) communication of the results (see Table 1) [10].

Whereas the first three steps are part of the build phase of the design science research method, the last three steps are assigned to the evaluation phase. The build phase is especially in design science projects of great importance and researchers spend much time on designing and constructing the artifact [3]. Accordingly, many publications on design projects lay their focus on the build phase, while the evaluation phase is often either neglected or only described as an evaluation concept. Only a minority of these publications do in fact evaluate the developed artifact [11]. A possible explanation for the high emphasis on building an artifact could be that it is a less satisfying duty for researchers to check whether all their efforts to strengthen the applicability and usefulness during the construction of the artifact does actually hold truth value during its evaluation [3].

In addition, many publications discuss the use of the design science research method theoretically, mostly without focusing a specific DS project (e.g. [1, 12, 13]. In these publications, the main question is how to conduct the design science research process more rigorously in order to provide guidance for the researchers. In contrast to practical publications of applied design science projects, theoretical works often emphasize the evaluation phase (e.g. [2]) and introduce several methods and techniques for evaluation (cf. [8, 14]). E.g. *Venable et al.* [14] present an evaluation framework, assisting a DS researcher in the selection of methods for ex ante and ex post evaluations. They provide detailed guidance for the evaluations themselves, though without alignment with current DSR processes. (cf. [2]) Also, only few theoretical publications focus on the construction of the artifact (cf. [15]). Since the practical publications are necessarily case specific, only rather rudimental guidance is provided for the Build phase, and almost none of these publications consider the different types of artifacts [11]. The best known publications for the build phase are the following.

- *Vaishnavi and Kuechler* [16] define patterns for techniques that can be applied to support the construction of the artifact. Similarly, *Sonnenberg and vom Brocke* [3] describe patterns which are used to evaluate the results of different steps during the build and evaluation phase of a design science project.
- Some authors define activities or guidelines which describe in more detail tasks to conduct in the build phase (e.g. [1, 9, 17])
- Other authors define requirements an artifact has to meet, which should already be considered during the construction of the artifact (e.g. [12, 18]).
- *Gericke* suggests approaches to support the construction for three artifact types [19].
- *Sein et al* [20] developed an approach which conducts the activities during the Build and Evaluate phase concurrently to immediately reflect the progress achieved and to trigger artifact revisions early within a design process.

2.2 Coupling Metrics

The subject of the DS project upon which we conduct the analysis is the development of so called "Coupling Metrics" which are used to support business process modeling in assessing and managing the quality of process models [21]. Coupling does not only evaluate the quality of single process models, there is a strong focus on interdependencies between models as well. Further, many operationalizations of coupling can be calculated automatically. This is an obligatory

prerequisite for the use in practice, which frequently encompass a very high number of process models [22].

Coupling in process management was preceded by coupling in software engineering. There, it was recognized as an indicator for the complexity of conceptual models. As such, the indicator is used to predict areas of high complexity, since the complexity of a system is known to cause implementation errors. Thus, coupling in conceptual models in software engineering is assessed with the intention to avoid errors in the conceptual stage, prior to their implementation when it is more difficult and expensive to correct them. [21]

In process management, the means to measure coupling in conceptual process models are based on transferring knowledge from software engineering to process modelling (cf. [5, 21, 23, 24]). For example, *Vanderfeesten et al.* [21] introduce the concept as originating in software engineering: "Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections." Thus, coupling is measurable and the measurement uses modules and interconnections as input. Further, the measurement indicates complexity (cf. [21]). As such, the measurement is again conducted with the intention to identify areas of high complexity in conceptual process models. Just as in software engineering, it is expected that highly complex processes lead to errors during their implementation. Further, due to their formalized description, the metrics can be computed automatically without user intervention [5]. They thus extend the currently existing means to measure and control the quality of conceptual models in business process management.

2.3 Development of Coupling Metrics

To make the concept of coupling available for end users, we transferred coupling metrics from the software engineering domain and specified them for the use in process models (cf. [5, 24]). In order to guide the transfer, we followed the activities of the Design Science Research (DSR) method. Therefore, DSR supports the development of our coupling metrics for process models and additionally helps to ensure the applicability and usefulness of the developed metrics.

From a DS perspective, our artifacts are the metrics' implementation in a process modeling environment, where they are supposed to assist process modelers in creating models that are easier to understand for a user. The metrics measure the complexity of process models providing guidance to improve the models, for which, however, the metrics need to be easily accessible and provide useful information. The respective quality of the metrics is regularly gained through an evaluation in a practical setting, e.g. in a case study, and serves to re-design, re-implement and re-evaluate the artifact, which, however, is laborious and expensive. To reduce the number of design cycles, an ex-ante evaluation as proposed by *Sonnenberg and vom Brocke* [3] seems promising to prevent potential design flaws which would otherwise surface either during the construction or, worse, during the practical evaluation.

3. Methodology

3.1 DSR methods

The well-known methods to conduct Design Science Research have general similarities [17]. Table 1 shows a comparison of prominent DSR methods, adapted from *Fischer and Gregor* [17] where they identified similar steps, arranged the methods accordingly and derived an idealized research model for DSR [17]. While we can assume a general sequence of steps in the DSR process, we omitted returns for reasons of clarity. While some of the authors defined returns for their DSR process, the trigger was not always obvious. In this work, however, the focus are returns due to the result of an evaluation and it can be assumed that in every method the result of an evaluation can be a reason to repeat the previous steps in a DSR cycle.

For our purpose, Table 1 highlights the evaluation steps of each method and also includes the DSR method by *Sonnenberg and vom Brocke* [3]. It can clearly be seen that while the currently prominent methods propose to explicitly evaluate the artifact only at the end of one cycle, the latter method proposes to evaluate after each step to avoid causes of repetition beforehand.

Our investigation aims to contribute to the discussion whether early evaluations in the build phase are beneficial or superfluous. To provide practical insights, we repeat a previous DS project in which we followed the Build and Evaluate approach. Now, we perform the Build phase with early evaluation activities in addition and compare results. This allows us to show whether in this case the gains, both in the quality of the solution and in the reduced DS cycles outweighed the additional evaluation effort, or not.

3.2 Previous procedure

The focus of our paper is not the evaluation of approaches for the Build phase in general. Instead, we

want to show the usefulness of evaluations in the Build phase to reduce the overall effort. Thus, we present the following discussion on the basis of the DS methodology by *Sonnenberg and vom Brocke* [3], which is applicable for the construction of our coupling metrics.

So far, we pursued the research of coupling metrics in a traditional Build-Evaluate approach. Table 1 visualizes our approach as an adaption of the method by *Peffers et al.* [10].

(1) First, as part of our research on process model understandability, we discovered that the currently available means to control process model understandability were insufficient, while e.g. software engineering successfully used metrics to analyze the complexity of conceptual models.

(2) In the second step, we defined the objectives for the metrics in more detail. E.g. we decided to limit our work on metrics which can be applied in the design phase of business process management, as such metrics would serve to avoid issues prior to their implementation.

(3) Third, we designed the actual metrics. To do so, we conducted a literature review with the objective of discovering existing coupling metrics for conceptual models in e.g. software engineering. The review was aligned to the review method by *Cooper [28]*. The well-known literature databases Google Scholar, Computer.org (IEEE Computer Society), AISeL and Emerald Insight, that offer a wide range of different electronic sources were queried using the term pair '"coupling metrics" "business process model"' as well as "coupling metrics". We then transferred the metrics from their original domain to process modeling. For the transfer itself, we distinguished between the cause of the complexity impairing the users' understanding and the artifacts upon which the cause was calculated by each metric. (cf. [29]) E.g. one metric would compute upon the number of connections between two randomly chosen artifacts in the model (cf. [30]), whereas another one would focus the hierarchical decomposition of the modelled artifacts (cf. [31]). This allowed us to search for artifacts within process modelling fulfilling an equivalent role regarding e.g. the number of steps in the control flow for any nodes of the control flow or the hierarchical decomposition of nodes in a process model hierarchy. Then, with these artifact candidates we were able to redefine the metrics by replacing the original artifacts with those from the domain of process modeling. At this stage, the coupling metrics were defined, based on the theoretical description of the respective process modelling language. At this stage, we had a theoretical definition for each metric, the artifacts' design. To make this design accessible for the end-user, we still needed to actually construct the metrics, as so far we had only implemented the metrics prototypically in a process modeling. A more detailed presentation of our research work can be found in [5, 24]. The resulting metrics are presented in sect. 3.5.

Originally, the subsequent activities were the (4) demonstration and (5) evaluation of the artifact, which in case of the metrics would be twofold. First, we planned to conduct a laboratory experiment to verify that after the adaption the metrics still do indicate complexity and a degraded understandability. Second, the usability of an actual implementation needs to be evaluated in an organization, e.g. by means of a case study. Both evaluations are laborious and would have to be repeated if they uncovered reasons to alter the design of the metrics.

For later reference, it is important to point out that this method does not refer to intermediate evaluations.

Table 1: Comparison of different DS approaches (cf. [17])

Idealized Research Model [17]	Nunamaker et al. [25]	Hevner et al. [1], also Peffers et al. [10]	Peffers et al. [10]	Takeda et al. [26], Kuechler & Vaishnavi [27]	Sonnnenberg & vom Brocke [3]
(1) Novelty/ Anomaly	(1) Construct a conceptual framework	(1) Important and relevant problems	(1) Identify problem and motivate (2) Define objectives of a solution	(1) Awareness of problem	(1) Identify problem (2) Eval1
(2) Generation of conjectures	(1) Construct a conceptual framework (2) Develop a system architecture (3) Analyze and design the system	(2) Iterative search process	(2) Define objectives of a solution (3) Design and development	(2) Suggestion	(3) Design (4) Eval2
(3) Generation of hypotheses	(3) Analyze and design the system (4) Build the prototype system	(3) Evaluate	(3) Design and development (4) Demonstration	(3) Development	(5) Construct (6) Eval3
(4) Empirical testing of the hypotheses	(5) Observe and evaluate the system	(4) Communicate	(5) Evaluation (6) Communication	(4)Evaluation (5) Conclusion	(7) Use (8) Eval4

The first systematically manifested feedback on the artifact is expected after the "Design & Development" and "Demonstration" steps. In the preceding steps, reasons to reconsider the design of the artifact in special or the method in general surface only by coincidence and not due to a systematic assessment.

3.3 Applying the DSR Evaluation pattern

In order to assist researchers in their DS projects, *Sonnenberg and vom Brocke* [3] propose the so called "General DSR Evaluation Pattern". The pattern is a high level description of a DS process, which takes into account the emergent nature of DS artifacts by introducing additional evaluation steps. In detail, the described process consists of the four DS activities "Identify problem", "Design", "Construct", and "Use" each of which is followed by an evaluation activity "Eval1" to "Eval4". Depending on their position in the pattern relative to the construction activity, they are considered Ex-Ante or Ex-Post evaluations. (cf. [3])

The evaluation activities of the general DSR Evaluation Pattern are described in more detail by separate patterns. There, Eval1, which follows the identification of the problem, is termed Justify. It serves to show that the current DS project is a meaningful one. Next to its objective, the description of the evaluation activity also shows possible methods to do so, e.g. an assertion, a literature review or a review of practitioner initiatives. (cf. [3])

The second evaluation activity of the process, Eval2, follows the design activity. It is meant to evaluate the design and to show that the design can bear a possible solution of the DS problem. Possible methods of this activity are an assertion, a mathematical proof, or logical reasoning, etc. (cf. [3])

The Ex-Post evaluation activities follow the construction of the artifact as well as its use. Eval3, performed after the construction, is meant to initially demonstrate the artifact, e.g. by a prototype demonstration or experiment. Eval4 evaluates the artifact in its environment to show that it is practically useful, e.g. with a case study or a field experiment. (cf. [3])

Comparing our previous procedure and the Evaluation Pattern by *Sonnenberg and vom Brocke* [3], Table 1 shows which additional steps we performed ex-post when we repeatedly developed the metrics with the new method.

(1) We started with the identification of the problem, which led us to the same problem statement as the previous DS project had.

(2) Second, the pattern suggests to evaluate the identified problem, by means of e.g. a literature review. To do so, we use the review method by *Cooper* [28], proposing five steps to specify a systematic conduct of the review. Starting with the problem statement and the search criteria, the method lead us to the same review we had conducted for the previous DS project. This review gave us feedback on a potential solution, we consider this early evaluation done. In fact, the literature we found underlined the relevance of the problem we identified once again and also provided feedback about the objectives because of which we focused on fully automatized metrics.

(3) Third, the pattern suggests to design the artifact. Since, up to this point, we had no additional knowledge with respect to design decisions, we came up with the same metrics as in our previous DS project.

(4) Fourth, after the design, the pattern proposes another evaluation, Eval2, by means of e.g. an assertion. However, originally we did not evaluate the design of our artifact (Eval2) before we implemented a prototype of the metrics. Thus, we do so in the following.

(5)-(8) Following the Eval2, the pattern suggests to construct the artifact. To do so, it is planned to implement the metrics in a process modelling software where they are accessible for modelers in their daily work. This implementation does then require an additional evaluation (6), before the use phase (7) and its evaluation (8), which provides feedback for further problems. These steps are subject for future work, which depends on the results of the early evaluations (2) and (4).

To analyze the benefit of early evaluations in the build phase, we instantiated the general DSR evaluation pattern. Thus, we performed the additional Eval2. It is to our benefit that the experiences from a long history of research on complexity metrics for conceptual models is documented in software engineering by *Weyuker* [33]. We could thus base our informed argument (cf. [3]) on extensive previous knowledge and benefit from the rigorous documentation of prescriptive knowledge on previous instantiations of the DSS methodology.

3.4 Evaluation criteria

To evaluate the design of our artifact (Eval2), we adapted the criteria by *Weyuker* [33] for process models, a set of desirable properties of complexity metrics in relation to their calculation:

P1: A metric should not rate all models equally complex, regardless of differences in their content.

The notion behind this property is that if one metric assigns the same level of complexity to each and every process model, it has no value to a user. Further, such a metric would contradict both our intuition and our empirical knowledge that differences in the complexity of process models do exist.

P2: A metric should not divide all models in only a few complexity classes.

This property extends **P1**. If a metric assigns e.g. only two or three different classes of complexity to process models of each and every size and shape, this is only of limited value to a user. Such metrics would not be suitable to prioritize a large set of models to be reworked, as many models of highly different complexity would be assigned the same class.

P3: A metric should allow for different models with equal complexity.

To explain this property, let us imagine a metric violating this property. It would assign a unique degree of complexity to each model, leading to an order of absolutely all possible process models. Since different processes lead to different models, this would imply that most processes cannot be modelled in a simple fashion. This is unrealistic. A sufficiently high degree of abstraction would lead to a simple model, which would still differ albeit only by the names of the nodes. Here, we preclude the case of infinite decimal places, as in practical settings users would most probably ignore very small differences anyway and thus assume different models to have equal complexity.

P4: A metric should allow for different models with the same semantic with different complexity.

The same process can be displayed in models which differ, despite having an equal level of detail and same information. Such a case can be devised by decomposing process models or aggregating functions differently. Different decompositions of the same process can lead to different degrees of complexity as is shown in empirical work (cf. [34])

P5: A metric should be monotone, thus the complexity of two concatenated models cannot be lower than the complexity of either of the two individually.

Complexity, as a property of the artifact, can further be disaggregated into the number of elements and their connections. Thus, if the number of elements increases, the complexity of a model will increase as well. Further, experience has shown that if one combines two previously separated processes into one common model, especially when done poorly, a reader will be even more overwhelmed.

P6: A metric should account for that two models with the same complexity may interact with a third model in different ways and thus have different complexities if concatenated.

Again, mind the decomposition of process models. If the concatenation is performed upon sub-models of a process model, it makes a difference if, otherwise identical, models are e.g. either attached to the end of the parent model or if the concatenation extends an already complex branching structure.

P7: A metric should account that permutations of one model may lead to different complexities.

This property reflects the motivation to decompose process models in the first place. If a process model is both very detailed with an extensive branching structure and very large, its complexity may challenge a reader. Therefore, modelers may break the model into parts with different decomposition. Overall, this will not change the model, but only permute its nodes over different parts. The resulting degree of complexity, however, depends on the actual decomposition. It is not hard to imagine, that a decomposition that tears apart closely related parts of a process model will increase the overall complexity (cf. [35])

P8: A metric should result in the same complexity if models are renamed.

We cannot make up a case where the naming of a process model increases or decreases its complexity.

P9: A metric should allow that the complexity of two models united is more than the sum of the individual models.

Imagine that processes with previously separated resources are joined into one common process. This will introduce interaction between the processes due to shared resources which did not exist before. Thus, the coupling encompasses not only the sum of the two models but also the newly introduced connections.

P10: A metric should not allow for the complexity of two united models to be lower than the sum of individual models.

This property extends **P5**, complexity is considered the result of elements and their connections. Assuming that by uniting two models the resulting number of nodes equals at least the sum of the individual numbers, the complexity of two models united cannot be lower than the sum of the individual ones.

We use these criteria to conduct the evaluation of our metrics' design, which was not part of our previous methodology.

3.5 Selected metrics

A short description of the metrics upon which the evaluation is done can be found in the following.

Coupling of a module This metric uses information theory to quantify the amount of information in the model graph within a sub model. An exceedingly high amount of information is expected to correlate with a decreased understanding and indicate complexity. [5, 24, 36] **Intramodule Coupling of a module** This metric quantifies the amount of information as well, but does so upon the graph of arcs which connect sub models instead of the model graph itself. [5, 24, 36] **CBO** The argumentation behind the CBO

Table 2: Measures and properties

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Coupling of a module [5, 24, 36]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Intramodule Coupling of a module [5, 24, 36]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
RFC [5, 24, 30]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CBO [5, 24, 30]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Direct Coupling [5, 24, 31]	Y	Y	Y	Y	N	Y	Y	Y	N	N
Indirect Coupling [5, 24, 31]	Y	Y	Y	Y	N	Y	Y	Y	N	N
Total Coupling [5, 24, 31]	Y	Y	Y	Y	N	Y	Y	Y	N	N
Process Coupling [5, 24, 37]	Y	Y	Y	Y	N	Y	Y	Y	N	N

metric is that sub models with a high number of connections are suspect of more external influence with unpredictable behaviour and are thus more difficult to understand. Therefore, the metric counts the connections a model has with other models to assess this form of complexity. [5, 24, 30] **RFC** The RFC metric extends the CBO metric. Here, the metric further asses the size of a model by the number of functions. Braunnagel 2013 #2855}[5, 30] **Direct Coupling** For the Direct Coupling metric, complexity is the result of connections between models in relation to the number of functions. [5, 24, 31] **Indirect Coupling** The Indirect Coupling metric extends Direct Coupling over transitive connections. Thus, the strength of the connections between two randomly chosen submodels is calculated. [5, 24, 31] **Total Coupling** The Total Coupling metric aggregates the prior over all sub models to indicate the overall complexity of a set of models. [5, 24, 31] **Process Coupling** Its objective is the delineation of functions that are executed in one block. Since overly large work units render processes inflexible and overly small work units increase the number of handovers, making processes failure-prone, the balanced delineation of functions in a workflow is a means for its improvement. For this metric, a function is large if it refers to many information elements and functions are coupled if they share a common information element. [5, 24, 37]

4. Evaluation

In the following, for the purpose of illustration, we discuss the properties with one of the metrics, and Table 2 shows the result for the remaining metrics. For more information readers may refer to [5] to verify our application of the properties onto all metrics.

4.1 Exemplary application

The **Process Coupling** metric was originally invented by [37]. It compares different process designs regarding the alignment of tasks in functions in a process. As the dependence between two functions due to shared information increases the number of handovers and possible failures, the metric calculates the fraction of functions which depend on the same information. We adapted the artifacts to the information elements and the functions in a business process. Thus, it is calculated as follows:

$$k = \begin{cases} \frac{\sum_{f_x, f_y \in F} conn\,(f_x, f_y)}{|F| * (|F| - 1)}, |F| > 1 \\ 0, \quad else \end{cases}$$

If the set of functions (F) in a process model is greater than 1, then the degree of coupling (k) is calculated by the quotient of the sum of connected function pairs (f_x, f_y) to all possible function pairs (|F|*(|F|-1)).

$$conn(f_x, f_y) = \begin{cases} 1, if\ (f_x, i_i) \in A \wedge (f_y, i_i) \in A \wedge (f_x \neq f_y) \\ 0, else \end{cases}$$

A pair of functions (fx, fy) is connected, if there is an arc (fx,ii) between the function fx and the information element ii and an arc (fy, ii) between the function fy and the information element ii in the set of arcs A. Originally, the authors argue that with a high degree of shared information elements, a workflow becomes less flexible. We argue, that this principle applies to business process modelling as well. Evaluating our design against the previous ten properties might present further insight into whether the implementation of the metric is a worthy pursuit.

Regarding **P1** and **P2**, it can be easily seen that the ratio of connected to all function pairs is different for models with either a different number of functions, connected functions, or both. Also, since there is no syntactical limit to the number of functions in a process model, the number of possible coupling degrees is unlimited within the range of [0, 1] as well. Thus, the properties **P1** and **P2** are fulfilled by the metric.

P3, different models with equal complexity, is fulfilled as well. Since the metric refers only to the number of functions and information elements, but not to their semantic, one may easily replace the actual elements, and thus create new models with the same complexity. Also, one may alter further nodes or the size of a model. As long as the ratio of connected and unconnected function pairs remains constant, all different models have the same degree of complexity.

The metric was originally created to assist companies in creating flexible processes by aligning tasks

differently. Aggregating tasks into functions in such a fashion that collects those tasks which require the same information elements leads to processes with a low degree of **Process Coupling**, notwithstanding that another composition of the same process might lead to another degree of coupling. Thus, **P4** is fulfilled.

P5, two models together may not be less complex than any of the individual ones, is violated due to the scaling. For two models, with the one model having coupled function pairs, and the second one not having coupled functions, the resulting **Process Coupling** Degree will be lower than that of the first model. In fact, the current construction of the metric tempts a user to create larger process models (i.e. include more functions) to reduce the degree of **Process Coupling**. Thus, the alignment of functions and information elements remains unchanged, even though it was identified as a source of inflexibility in the first place. The solution therefore is to omit the scaling by |F|*(|F|-1). This alternation would not violate the previous conditions, for the same reasons as before.

Regardless of whether we omit the scaling or not, **P6** is fulfilled by the metric. If a model is joined with one of two other models, it may or may not happen that functions from both models share information elements and that thus the number of paired functions increase for more than the additional model. Therefore, further interaction may affect the metric value, depending on whether this scenario happens or not.

The original intention of the metric was to point out the alignment of functions and information elements leading to the lowest sharing of information elements, thus encouraging process designer to permute functions and information elements in such a way as to reduce **Process Coupling**. Thus, **P7**, a metric should account that permutations lead to different complexities, is fulfilled.

The same can be said for **P8**. Since the name of the process is not considered in the calculation, it does not change the coupling value.

The **Process Coupling** metric allows for the complexity of two models united to exceed the sum of the individual models (**P9**) if the scaling is omitted. This can be shown by example, when two models are joined which share common information elements, the number of function pairs may rise beyond the sum of the pairs. If the scaling is still done, the calculated value will either decrease or remain unchanged, which appears counter-intuitive to us.

As a consequence of the scaling, the metric as originally presented systematically violates **P10**. If two models are united, the degree of **Process Coupling** will either decrease or remain the same. Again, as a solution, one may omit the scaling in (2).

4.2 Result

The above discussion has shown that our current design of the **Process Coupling** metric cannot fulfill three of the desirable properties. They are all related to the scaling of the original design which causes a side effect that we did not anticipate. Originally, the metric was supposed to aid a practitioner in evaluating his process design, regarding dependencies among functions which result from shared resources. A lower metric value is supposed to indicate a better design regarding the resource coupling of functions which indicates a higher flexibility in the process' execution since fewer functions depend on each other due to shared information. However, due to the scaling, a lower metric value can also be achieved by e.g. merging different models or otherwise by introducing additional functions without any link to a resource. As a result, the number of coupled functions would not decrease and the actual flexibility of a process would not improve. Instead, the model would either be filled with irrelevant information or merged unnecessarily. In any case, its understandability would be degraded. To avoid this side effect, we alter the metric by omitting the scaling.

Table 2 shows the performance of the metrics in our current design regarding the desirable properties. Our adaption of the Process Coupling metric violates the desirable properties **P5**, **P9** and **P10** due to a scaling function and so do the metrics **Direct Coupling, Indirect Coupling, Total Coupling** and **Conceptual Coupling**.

The evaluation framework by *Venable et al.* [14] distinguishes between naturalistic evaluations (e.g. action research) and formalistic evaluations (e.g. criteria based). The latter are generally less costly. Following the method of *Sonnenberg and vom Brocke* [3], we performed a formal (criteria based) evaluation after the design step, prior to the more costly evaluations, and identified potential design issues.

In our original research method, we had planned to evaluate this design in a laboratory experiment and a practical setting. We suspect that the design issue would have surfaced in the latter evaluation, too but would have triggered another design cycle in addition. Thus, the altered design would have required another costly evaluation, both in a laboratory and a practical setting.

Of course the early evaluation cannot guarantee that every design flaw was uncovered, and thus no further cycles are necessary. However, it will ease the identification of the causes to fail the practical evaluation, since less issues will cause less confusing interplay.

5. Discussion

The DSR method of *Sonnenberg and vom Brocke* suggests two additional evaluation activities for each phase of the general Build and Evaluation phases [3]. Especially the two evaluations in the Build phase (Eval1 and Eval2) seem to be an interesting and novel recommendation. The aim of our paper was to investigate in how far the evaluation of the artifact during the Build phase can contribute to the quality of the research by reducing the number of cycles in the whole DS project. On the one hand, the conduct of additional evaluation activities during the Build phase should generally contribute to the quality of the artifact. On the other hand, additional evaluation activities are time consuming and the required effort must be adequate regarding the gains in quality.

Sonnenberg and vom Brocke suggest a first evaluation after the problem identification which is meant to ensure that a meaningful design science research problem and a meaningful statement is formulated [3]. In our research project, Eval1 was based on a literature review. The review served not only as proof of relevance for the problem, but also for the refinement of the problem definition. We identified that coupling is especially relevant for process architectures because automatically computable coupling metrics are of great help for the design and development of process models in a process architecture.

The second evaluation activity (Eval2) serves to show that an artifact design provides the solution to the stated problem as well as to ensure the solution's quality. The object of this evaluation is the artifact as a concept and not the finished solution. We did not evaluate the concept of our artifact in our primary investigation (see section 3.2). Therefore the emphasis in our investigation was on examining the contribution of Eval2. To perform our evaluation, we instantiated the "Assertion" pattern and presented an informed argument with all metrics as concepts and criteria in the form of desirable properties, which we found in literature. As a result, we identified four metrics which were not able to meet all desirable properties. We could show that the original design which neglects the missing properties would have misled practitioners. The metrics indicated improvements which actually degraded the models. Originally, the metrics gave e.g. a better rating to a model if it was inflated with unnecessary information and the source of harmful coupling remained unaltered. Such a model would be more difficult to read, implement or maintain and generally more difficult to use. To avoid this effect, we altered the metric by omitting the scaling. Therefore, the benefit of conducting Eval2 was twofold: we did not implement misleading metrics and thus saved time otherwise spent on needless implementation efforts. In addition we could further improve these four metrics in an early phase and eliminate their defects.

All in all, we could demonstrate with our investigation that the application of Eval2 reduced the cycle time of our research project, and we improved the quality of the artifact at an early stage during the design science project. Further, we are certain that in our case the additional evaluation was more efficient in comparison to additional evaluation and implementation cycles. However, this was much due to the easily available evaluation criteria in our project. If *Weyuker* [33] had not documented the experiences from decades of metric development, the efforts to find applicable evaluation criteria or apply another evaluation technique would have been larger and they might possibly even have outweighed the additional implementation cycles. In summary, despite our promising results, we cannot declare the early evaluations in the Build phase to be generally reasonable for all DS projects, but we do argue that early evaluations of the concept are generally a worthwhile consideration. As we have shown in our case, they can spare DS cycles and improve the concept and thus the solution, as well. Also, as experience from software engineering shows, resolving issues in an early phase of the SE cycle (e.g. during the analysis phase) is less time-consuming and cost-intensive than later e.g. during implementation. (cf. [3]).

6. Conclusion

Our paper deals with the application of the DS methodology by *Sonnenberg and vom Brocke* [3], resp. with the application of early evaluations in the build phase. In comparison to traditional DS approaches, this puts particular emphasis on the evaluation after each step of the DS method. The additional evaluations can be either superfluous or helpful to uncover pitfalls which otherwise enforce additional DS cycles. Thus, we compare the procedure of one of our research projects where we follow a traditional DS approach with a procedure including early evaluations in the build phase.

In our case, the additional evaluation uncovered, and also helped to mitigate, design flaws that would have forced us to repeat the laborious evaluation in the organization, had they remained undiscovered. The effort of the additional evaluation profits from the availability of evaluation criteria. We benefited from ready-to-use criteria from complexity metrics development. Thus, we encourage the research community to document prescriptive knowledge (cf. [38]) as support for DS projects.

As stated, the evaluation of the developed artifact is missing in many publications about DS projects (cf. [11]). This has been discussed in literature, too (cf. [3]). Apart from the different possible explanations to this observation, this underlines the necessity to apply the evaluation patterns for the Build phase, as they provide the possibility to assert the artifact's concept and to show its usefulness and superiority.

7. References

[1] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," MISQ, vol. 28, no. 1, pp. 75–105, 2004.
[2] J. Pries-Heje, R. L. Baskerville, and J. R. Venable, "Strategies for Design Science Research Evaluation," in Proc. o. 16th ECIS, 2008, pp. 255–266.
[3] C. Sonnenberg and J. Vom Brocke, "Evaluation Patterns for Design Science Research Artefacts," *CCIS 286* Springer, 2012, pp. 71–83.
[4] A. R. Hevner and S. Chatterjee, "Design Science Research Frameworks," ISIS 22, Springer, 2010, pp. 23–31.
[5] D. Braunnagel, F. Johannsen, and S. Leist, "Coupling and process modelling," in Modellierung 201, pp. 121–137.
[6] H. A. Simon, The sciences of the artificial, 3rd ed. Cambridge, Mass: MIT Press, 2008.
[7] S. T. March and G. F. Smith, "Design and natural science research on information technology," Decision Support Systems, vol. 15, no. 4, pp. 251–266, 1995.
[8] E. Klecun and T. Cornford, "A critical approach to evaluation," Eur J Inf Syst, vol. 14, no. 3, pp. 229–243, 2005.
[9] P. Offermann, O. Levina, M. Schönherr, and U. Bub, "Outline of a design science research process," in Proc. o. 4th DESRIST, New York: ACM Press, 2009, p. 7.
[10] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," JMIS, vol. 24, no. 3, pp. 45–77, 2007.
[11] P. Griesberger, "Developing the Evaluation of a Pattern-Based Approach for Business Process Improvement," Proc. o. DESRIST 2014, Springer, pp. 225–240.
[12] S. Gregor and D. Jones, "The Anatomy of a Design Theory," JAIS, vol. 8, no. 5, p. 2, 2007.
[13] J. Pries-Heje and R. Baskerville, "The Design Theory Nexus," MISQ, vol. 32, no. 4, pp. 731–755, 2008.
[14] J. R. Venable, J. Pries-Heje, and R. L. Baskerville, "A Comprehensive Framework for Evaluation in Design Science Research," in Proc. of DESRIST 2012, pp. 423–438.
[15] J. Pries-Heje, R. Baskerville, and J. Venable, "Soft Design Science Research," in DESRIST 2007, pp. 18–38.
[16] V. Vaishnavi and W. Kuechler, Design science research methods and patterns. Boca Raton: Auerbach, 2007.
[17] C. Fischer and S. D. Gregor, "Forms of Reasoning in the Design Science Research Process," in Proc. o. DESRIST 2011, pp. 17–31.
[18] J. Iivari, "A Paradigmatic Analysis of Information Systems As a Design Science," SJIS, vol. 19, no. 2, p. 5, 2007.
[19] A. Gericke, "Konstruktion situativer Artefakte," Dissertation, Universität St. Gallen, 2009.
[20] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, "Action Design Research," MISQ, vol. 35, no. 1, pp. 37–56, 2011.
[21] I. T. Vanderfeesten, J. Cardoso, J. Mendling, H. A. Reijers, and W. M. P. van der Aalst, "Quality Metrics for Business Process Models," *BPM and workflow handbook*, L. Fischer, Ed, USA: Future Strategies, 2007, pp. 179–191.
[22] M. Malinova, H. Leopold, and J. Mendling, "An Empirical Investigation on the Design of Process Architectures," in Int. Conf. on Wirtschaftsinformatik, 2013.
[23] W. Khlif, L. Makni, N. Zaaboub, and H. Ben-Abdalla, "Quality metrics for business process modeling," in Proc. o. 9th WSEAS Int. Conf. on ACS, 2009, pp. 195–200.
[24] D. Braunnagel and F. Johannsen, "Coupling Metrics for EPC Models," in Int. Conf. on Wirtschaftsinformatik: 2013, pp. 1797–1811.
[25] Nunamaker, Jay, F, Jr, "Systems developement in information systems research," JMIS, vol. 7, no. 3, pp. 89–106, 1990.
[26] H. Takeda, P. Veerkamp, T. Tomiyama, and H. Yoshikawa, "Modeling Design Process," AI Magazine, vol. 11, no. 4, pp. 37–48, 1990.
[27] B. Kuechler and V. Vaishnavi, "On theory development in design science research," Eur J Inf Syst, vol. 17, no. 5, pp. 489–504, 2008.
[28] H. Cooper, Synthesizing research, SAGE, 2006.
[29] R. L. Flood and E. R. Carson, Dealing with complexity, 2nd ed. New York: Plenum Press, 1993.
[30] G. Gui and P. D. Scott, "New Coupling and Cohesion Metrics for Evaluation of Software Component Reusability," in 9th Int. Conf. for Young Computer Scientists, Los Alamitos, USA: IEEE, 2008, pp. 1181–1186.
[31] S. R. Chidamber and C. F. Kemerer, "A Metrics Suite for Object Oriented Design," IEEE Trans. Software Eng, vol. 20, no. 6, pp. 476–493, 1994.
[33] E. J. Weyuker, "Evaluating software complexity measures," IEEE Trans. Software Eng, vol. 14, no. 9, pp. 1357–1365, 1988.
[34] F. Johannsen, S. Leist, and D. Braunnagel, "Testing the Impact of Wand and Weber's Decomposition Model on Process Model Understandability,", ICIS, 2014.
[35] S. Zugal, "Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models," Dissertation, University of Innsbruck, 2013.
[36] E. B. Allen, T. M. Khoshgoftaar, and Y. Chen, "Measuring Coupling and Cohesion of Software Modules," in 7th Int. Software Metrics Symposium. 2001.
[37] I. T. Vanderfeesten, H. A. Reijers, and W. M. P. van der Aalst, "Evaluating workflow process designs using cohesion and coupling metrics," Computers in industry, vol. 59, no. 5, pp. 420–437, 2008.
[38] C. Sonnenberg and J. Vom Brocke, "Evaluations in the Science of the Artificial," in Proc. of DESRIST 2012, pp. 381–397.