Evaluating Design Rationale in Architecture

Pim de Jong
Sogeti Netherlands B.V.
Nieuwegein, The Netherlands
pim.de.jong@sogeti.com

Jan Martijn E. M. van der Werf
Utrecht University
Utrecht, the Netherlands
j.m.e.m.vanderwerf@uu.nl

Marlies van Steenbergen
Sogeti Netherlands B.V.
Nieuwegein, The Netherlands
marlies.van.steenbergen@sogeti.com

Floris Bex
Utrecht University
Utrecht, the Netherlands
f.j.bex@uu.nl

Matthieu Brinkhuis
Utrecht University
Utrecht, the Netherlands
m.j.s.brinkhuis@uu.nl

***Abstract*—Although architecture is often seen as the culmination of design decisions, design rationale is a supposititious child in architecture documentation. Many architecture frameworks and standards, like TOGAF and ISO/IEC 42010, recognize the importance, but do not offer any support in the rationale process. Recent initiatives have shown that simple means help in providing more rationale. However, there are very few studies that give evidence whether more rationale indeed leads to better quality.**

In this paper, we propose a non-invasive method, the Rationale Capture Cycle, that supports architects in capturing rationale during the design process. Through a controlled experiment with 10 experienced architects, we observe the effectiveness of the method in terms of design quality through different measures. The results of our experiments show that: (1) better rationale is strongly correlated with high quality, and (2) the test group with our proposed method outperforms the control group.

***Index Terms*—design reasoning, design rationale, enterprise architecture, architecture documentation**

I. Introduction

Architectures, or more specifically architectural descriptions, refer to the fundamental elements of a system and how they relate to comprise a perspective of the system in its context [1]. In the case of Enterprise Architecture the architectural description hence concerns the organization or enterprise, its IT systems, (functional) units, business processes and external contacts such as customers and suppliers.

Architecture is often regarded as the culmination of design decisions [2], [3]. Such design decisions are made during the design reasoning process [2], in which we have to identify the context and requirements of the enterprise, formulate and structure the design problem, think about possible solution options and make trade-offs between different options. Designing complex architectures for larger enterprises can be considered a *wicked problem* [4]: the design problem is ill-defined; there is no stopping rule to tell when an acceptable solution has been reached; it is difficult to tell what is true or false in a design (at least not before implementation); resolving one design issue can give rise to other related design issues; and understanding the problem depends on how the designer wants to solve it.

In the case of wicked design problems it is important to capture the *design rationales*, the reasons underlying design decisions. Properly documenting architectural design decisions and their rationale is essential to any architectural description, as it explains the reasoning behind why the architecture is as it is [5]. Without proper explicit rationale, the architecture degradates [6], and architectures become increasingly brittle over time if documentation of decisions is not kept. For example, architects may leave the organization, switch roles or switch projects. Additionaly, architectural documentation with rationale makes it easier for architects to understand a design, especially if the architects were not involved in the original design process: Tang et al. [5] found that architects strongly agree on the premise that they cannot understand a design effectively without its rationale.

In architecture design[1], many methods and techniques for reasoning about architecture designs have been proposed. Examples of specific techniques are solution option generation, constraint analysis, risk analysis, assumption analysis, and trade-off analysis. The research and presentation of these techniques, however, is fragmented, and a single method that encompasses all of them is lacking. Frameworks like IBIS (Issue-Based Information Systems) [7] and QOC (Questions-Options-Criteria) [8] combine different concepts such as options, issues and constraints and provide a language and accompanying modelling tool to map design rationale. A major problem of this focus on "rationale mapping" is that it interferes with the design reasoning process: there is a cognitive overload that results from having to learn and use such tools at the same time as having to discuss and think about complex architectural designs [5]. Although widely available, these frameworks are therefore still underutilized in industry [5], [9], [10].

Architecture development methods such as the Togaf Architecture Development Method [11] also recognise the importance of careful design reasoning and include separate steps for, for example, requirements analysis and cost-benefit analysis. Furthermore, architectural standards such as SysADL [12], GRASP [6] and ISO/IEC 42010 [1] mention the importance of the architecture rationale, that is, the reasons underlying the important architecture design decisions. However, the focus of these methods and standards lies firmly on the description of the enterprise or system itself, and not on the reasoning

[1]Note that here we look at both Enterprise Architecture and Software Architecture, as many of the general frameworks, standards and techniques for each of the architectural fields are very similar.

process or the product of this process, the design rationales, which are only mentioned superficially. In other words, it is not exactly elaborate *how* designers should reason and capture rationale.

Recently, initiatives have been started to develop and evaluate normative methods that encourage and support the architecture design reasoning process and associated rationale capture. For example, Razavian et al. [13] employ reflective questions by an external observer, Schriek et al. [14] present a card game, and Keil et al. [15] use checklists for decision making. All studies show that with simple means, the rationale process can be improved.

On the other hand, only very few studies study the benefit of improving the rationale process. For example, Tang et. al. [16] show that architects produce, on average, better architectures by explicitly reasoning about their design decisions. Their study shows that design reasoning principles are applied more effectively when provided with a systematic approach.

Under practicioners, there is the general perception that, although design rationale should be captured in documentation, a proper methodology is lacking [5]. This observation is the starting point of our paper: *"What is an effective method to support the provision of design rationale in the architecture design process?"*. To answer this question, we study the existing body of knowledge in enterprise architecture and design reasoning, to develop a new method: the Rationale Capture Cycle. To study the effect of the method, we perform a controlled experiment with 10 experienced enterprise architects. We define effectiveness as the perceived quality of the solutions delivered by the architect, and measure this in three ways: (1) through peer review by the participating architects, (2) by analyzing the documentation on rationale usage, and (3) by a ranking of an external architect. Our results are twofold: (1) better rationale is strongly correlated with high quality, and (2) the test group outperforms the control group.

The remainder of this paper is structured as follows. First, we discuss in Section II the setup of our research. Section III provides a background in design rationale, and introduces our treatment method, the RCC. The results of the experiment are presented in Section IV and discussed in Section V. Section VI concludes the paper.

II. Research Design

The ultimate objective of this study is to provide a means by which architecture designers can systematically employ design reasoning. To this end, we apply design science [17]. First we construct a method through method engineering. Through a controlled experiment we evaluate the effect of the method: the method is given as treatment to the test group, whereas the contol group works without the method. For the experiment, we followed the guidelines by Wohlin et al. [18].

A. Method Design

To construct a method that supports the reasoning process, we followed the method association approach [19]. The final method has been developed through two iterative cycles. In the first cycle, we studied existing rationale methods and techniques and architectural frameworks and languages. This resulted in a set of requirements, both on the steps in the method, and which rationale techniques could be employed in each step. Next, the method has been validated and discussed through semi-structured interviews with 10 architects. The feedback we received from these interviews were then used to improve and finetune the resulting method.

B. Experiment Design

To evaluate the effect of the method, we conduct a controlled experiment with two groups: a test group that receives the method as treatment, and a control group who works without [18]. We applied stratified sampling for assigning the participants to the groups, controlling the factors experience (m=30yr, SD=8), age (m=56yr, SD=7), and familiarity.

The experiments consists of two stages. In the first stage, each participant has to individually solve a case where they are asked to design in two hours a complete architecture, and provide textual rationale for their reasoning process. First, the case is introduced by one of the authors to all the participants in a group session of about 30 minutes. Next, the participants in the control group leave the session and start on their solution. In the mean time, the participants in the test group are asked to use the treatment before they start to work on their assignment. At the end of the design session, each participant fills in a questionnaire about the reasoning process, the case and the method. This stage results in 10 solutions to the case: five for each group.

In the next stage, each participant ranks three solutions by dividing 100 points to the different solutions, without creating a tie. The design of the ranking satisfies the following criteria:

- Each participant reviews at least a solution from both groups, not being their own;
- Each combination should be unique, i.e., no given ranking occurs more than once;
- Each solution is ranked an equal number of times, equally distributed over the two groups;
- In the ranking no cluster of solutions is disconnected from the others, i.e., a total order can be induced from the partial orders formed by the individual rankings;

To observe the effect of the treatment, we use four measures:

1) The total points given by the ranking of the architects, to provide the relative design quality of the solutions;
2) The documentation delivered by the participants is counted, coded and analysed;
3) A full ranking of all solutions created by an external enterprise architect;
4) The results from the questionaire;

The coding schema used for the second measure is based on the available literature on rationale techniques gathered during the method design. Based on these measures, we evaluate whether the proposed method is effective for capturing rationale, and for delivering better quality.

Table I
ARCHITECTURE RATIONALE TECHNIQUES, THEIR DEFINITION AS USED DURING THE CODING STAGE, AND ITS SOURCE

Rationale type	Definition	Source
Design decision	Final decision made during the reasoning process.	[20]
Problem analysis	Key issues and requirements in a design.	
Constraint analysis	Limitations on the design, how they relate and materialize.	[5]
Assumption analysis	Unknown factors affecting the design, and their consequences.	
Option analysis	Alternatives addressing the same design problem.	[20]
Benefit analysis	Benefits a design option can deliver to satisfy the requirements.	[5]
Weakness analysis	Weaknesses a design option have and their effect to the design.	
Trade-off analysis	Trade-offs to compare alternatives and its supporting rationales.	[21]
Risk analysis	Unknown factors with negative implications, on a design option and how they impact the design.	[20]
Evaluation	Feedback loop to rethink critical elements in the rationale.	[22]
Reflection	Verification of the architect's own reasoning process.	

C. Case Design

Participants are presented a fictional case of a Dutch bank. The case describes a bank that started as an exclusive bank serving generations of prosperous families, but now wants to expand its clientage. In this process, the bank is facing a surge of new clients that bring about new wishes and needs in terms of the service. As the services provided by the bank are outdated, the bank hires architects to provide goal architecture models to sketch the ideal system landscape for them. The architects need to provide an architecture description in which their rationale must be documented.

The case has been designed such that it is simple and clear enough to understand, and feasible to complete within 2 hours, yet be intricate enough, so rationale can be present. A pilot test with 4 master students from Utrecht University is held to validate the case and eliminate any ambiguous elements. The pilot showed that the case is challenging, especially given the limited amount of time. As architects have more experience, and constantly need to make decisions on what to pursue, we decided to keep the length and level of difficulty.

III. CAPTURING DESIGN RATIONALE

Design reasoning refers to the process of reasoning on design issues and coming to a design decision [5]. It relies on critical and reflective thinking, evaluation, discussion and logical inferencing. Key principles include problem structuring, assumption recognition, and trade-off analysis. The product of design reasoning is design rationale, which can be defined as the justification for a design decision. This includes the entirety of explicit reasoning that was required to make a decision, i.e. the explicit culmination of argumentative knowledge to weigh and elect options.

A. Architecture Rationale

The Oxford Dictionary defines rationale as a "reason or intention for a specific action". In the context of architecture design, rationale is considered to be "the reasons behind a design decision, the justifications for it, the alternatives considered, trade-offs evaluated and argumentation that led to the decision" [23]. An overview of the most frequently used rationale techniques is presented in Table I.

Design rationale can occur in various forms and types, depending on their use and current need [24]:

- *Argumentation based:* the design rationale presents arguments that define the design. These arguments present pros and cons for each alternative, option and issue.
- *History based:* the design rationale represents the design history of a system. This includes the dimension of time, i.e., design decisions are included chronologically.
- *Device based:* the design rationale is based on a model of the device or system itself. The model can simulate the behaviour of the system, from which the design rationale can be deduced.
- *Process based:* the design rationale is completely interwoven with the design process itself and therefore guides the format of the rationale.
- *Active document based:* the design rationale is predefined and already saved in the system. When an architect designs an architecture, the design rationale system automatically generates rationale for the elements designed by the architect.

These types are by no means exhaustive and may simultaneously be true. For example, architecture documentation can contain a full history of all options with their pros and cons, and which options have been chosen. Which type of rationale is used, also depends on how rationale is captured. Lee [23] identified five major methods to do so:

- *Reconstruction* constitutes the production of rationale by reasoning from existing knowledge (introspection). Reconstruction is always post-mortem, i.e., completely separated from the design process. As such is it non-intrusive to the design process itself.
- *Record and Replay* captures rationale during the design process. Design problems are identified, alternatives are considered and criteria and claims are defined whilst design is taking place. This can occur through digital means (forums, videoconferencing) or through regular face to face meetings.
- *Methodological By-product:* in this approach the rationale is produced as a logical by-product of following a method in the architecture design process. This method constitutes capturing the rationale. The idea is that the

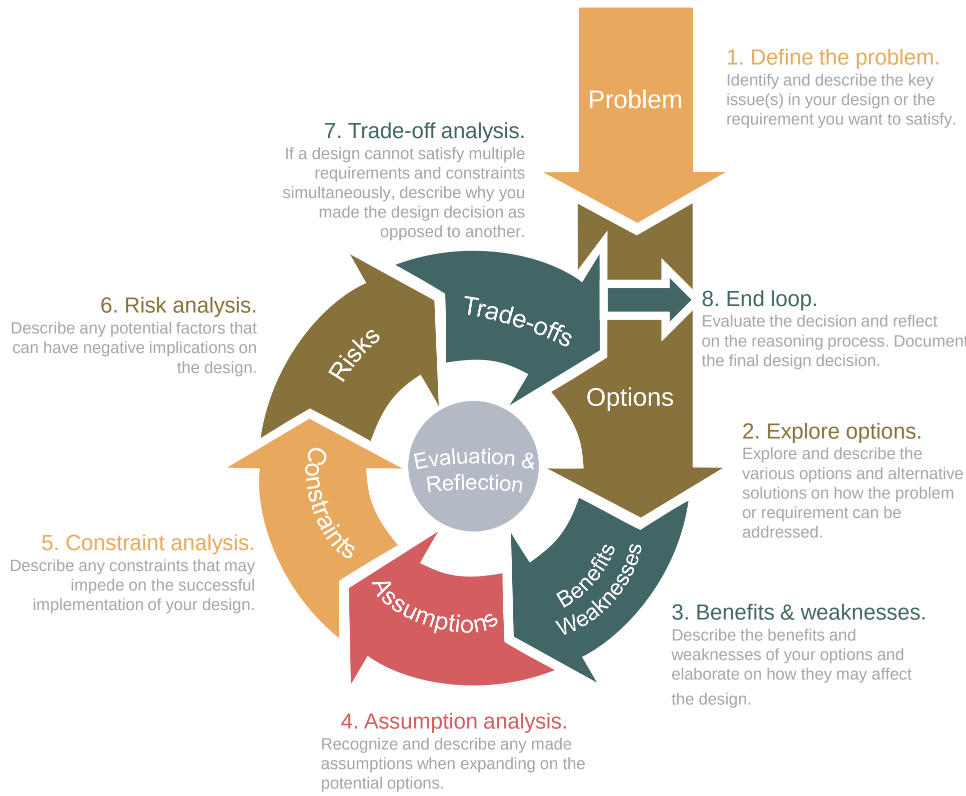

Figure 1. The Rationale Capture Cycle

architect follows a method that supports generating rationale during the design process.

- *Apprentice:* the apprentice approach consists of the interaction between a designer and a computer system. The system verifies design decisions made by the designer and asks questions whenever an action is made which it does not understand.
- *Automatic Generation:* constitutes a system that produces rationale from an existing rationale base. The system analyses a complete history of designs and defines the how's and why's of the performed actions.

The main design philosophy revolves around creating a method that captures rationale whilst least interfering with regular design activities. Therefore, the most successful rationale capture approach is likely to be a methodological by-product approach [23], combined with process- and argumentation-based rationale. Such combination should result in a feasible and approachable method [24].

B. The Rationale Capture Cycle

Previously proposed methods for rationale capture were ineffective due to a variety of reasons [5]. Methods are either (1) too limited in depth: offering no real handson support, (2) too complex: overly daunting for an average user, or (3) too specific: often inapplicable in various domains or projects. These factors all contribute to the lack of industry standards [5]. The Rationale Capture Cycle (RCC) is created around countering these three factors.

The sequence of the RCC is based on the Planning and Problem-Solution Co-Evolution theory [20]. As early decisions heavily influence the process in which design activities are carried out [20], designers should consider a high-level design plan first, i.e., one should identify the main requirements and design issues first.

The rationale process is triggered through the identification of a problem, such as the satisfaction of a requirement. For this solution, many different options can be generated and explored, which all need to be analysed. For each option, the architect should analyze the possible benefits and weaknesses, which assumptions are foundational to the option, which constraints are either being put by the option, or hamper the option, the risks involved and a final weighted decision based on a proper trade-off analysis. For all activities, the architect should evaluate and reflect to what extent the process has been followed, and critically assess what could be improved to the rationale, and the process itself. The RCC captures each of these activities in a cyclic manner.

IV. Results

In the controlled experiment, we observed three measures to define a ranking: the points awarded by the participants, the coding of the documentation, and the full ranking by the external architect (Table II).

Table II
RANKINGS BY THE DIFFERENT MEASURES: POSITION, POINTS, DOCUMENTATION AND EXTERNAL RANKING. ELEMENTS MARKED WITH * OR ** ARE RANDOMLY ASSIGNED, AS THEY HAVE THE SAME AMOUNT OF POINTS.

Pos.	Points	Doc.	Ext.	Pos.	Points	Doc.	Ext.
1.	T4	T4	T4	6.	C1 *	C2	C1
2.	T2	T2	T2	7.	T5 **	T5	T5
3.	T3	C5	T3	8.	C2**	C1	C2
4.	C5	T1	C5	9.	C3	C4	C4
5.	T1 *	T3	T1	10.	C4	C3	C3

Table III
SOLUTIONS RANKED BASED ON THE ASSIGNED POINTS. ELEMENTS MARKED WITH * OR ** ARE RANDOMLY ASSIGNED, AS THEY HAVE THE SAME AMOUNT OF POINTS.

Rank	Solution	Points	Rank	Solution	Points
1	T4	160	6	C1*	95
2	T2	130	7	T5**	90
3	T4	115	8	C2**	90
4	C5	110	9	C4	60
5	T1*	95	10	C3	55

A. Design Quality

The first measure to observe the design quality is the relative ranking provided by the participants themselves. Recall that each architect ranked three solutions by dividing 100 points to the solutions, and that ties were not allowed. Each participant received at least one solution from both groups. By summing the points per solution, we obtain the relative ranking by the participants.

None of the participants gave a solution 0 points. Half of them used a 50-30-20 point scale. The other distributions were 45-35-20 (two times), 75-20-5 and 65-20-15 (both one time). The solutions of T1 and C1, and of T5 and C2 received an equal amount of points. To maintain a full ranking, these were randomly assigned to their respective positions. This results in the ranking of Table III. A first observation is that three solutions out of the top 4 are from the test group.

B. Rationale Documentation

The participants had to explicate their entire reasoning process in the documentation, including the justifications for their design decisions. As a next step, we coded the architecture documentation of each participant to distinguish the individual rationale types in each document. The coding was performed by one of the authors, after which the results were validated and discussed with two of the other authors to reach consensus. The coding schema used is given in Table I. During the coding, we made an explicit distinction between identifying a rationale technique, and describing it. For example, one can identify a risk, but not describe why it is a risk, or how the architecture is affected. Additionally, we counted the number of words in the documentation.

The coding shows that the participants of the test group used 122 rationale elements, whereas the control group used 69 elements. These were individual elements of rationale, elaborating on why they made certain design decisions. Interestingly, there are three elements that have no occurrence in the documents across the various participants: evaluation, reflection, and assumption definition. Important to realize is that this does not imply that these techniques have not been used, solely that in the documentation these were not demonstrated explicitly.

C. Feedback by the Architects

At the end of the session, the architects of both groups were asked to fill in a questionaire to give feedback and additional thoughts on the RCC, the case, and the overall experiment. Most questions regarding the RCC concerned added value, effectiveness, readability, usability, intuitiveness, and ease of use.

a) Added value & effectiveness: There were mixed feelings about the value and usefulness of the architecture: two architects found the case too small for using something like the RCC, whereas three other architects claimed that the RCC helped them to capture rationale more easily, especially since the RCC made them capture assumptions that they otherwise would not have made explicit. The RCC gave them structure to the process which is “desperately needed”.

b) Readability & usability: Most architects found the RCC easy to read and understand. One architect mentioned the cycle was confusing, as there is no real end to the process.

c) Intuitiveness & ease: Most architects mentioned that the RCC did not intrude on the design process, and that it was easy to use. One architect found that the RCC asks for a too high investment of time, which is only realistic if the case would offer choice moments of larger size.

d) : The participants were also asked to share their thoughts regarding the case. Out of all 10 architects, 4 mention that the case did not accurately mimic a real life scenario, as the architect normally has a chance to ask and research the scenario, which is inherent to the type of case provided. Half of the architects mentioned the time limit: time was too short to create a complete architecture for the case, uttering that their models and documentation were unfinished by the time the session ended. One architect rather liked the case, as it demonstrated the realistic scenario of a board that does not accurately know what it wants. Overall the architects found the RCC helpful in better structuring the architecture documentation and the reasoning process.

V. ANALYSIS AND DISCUSSION

Answering whether the RCC is an effective method to stimulate the rationale process, boils down into two questions we need to analyze: ““are the different rankings in agreement about the quality of the architectures?”, and “perform the participants in the test group better than the control group? ”. In the next subsections, we analyze the results to answer these questions.

A. Agreement Between the Rankings

A scatter plot of the rankings is depicted in Figure 2. The plot shows that each solution is either in the lower-left or in

Table IV
ANALYSIS OF THE DIFFERENT SOLUTIONS, BASED ON THE CODING SCHEMA IN TABLE I

		C1	**C2**	**C3**	**C4**	**C5**	**Total**	**T1**	**T2**	**T3**	**T4**	**T5**	**Total**
Word count		311	251	114	317	605	**1598**	289	511	284	490	472	**2046**
Rationale type													
Design decision		2	3	2	2		**9**	4	8	3	1	4	**20**
Problem	Ident.		1		3	4	**8**	3	4	3	4	2	**16**
analysis	Descr.				2	4	**6**	2		4		1	**7**
Constraint	Ident.										5		**5**
analysis	Descr.										1		**1**
Asumption	Ident.										5		**5**
analysis	Descr.												**0**
Option	Ident.	2	2	2	2	6	**14**	3	10	3	3	1	**20**
analysis	Descr.	1	1	4	2	5	**13**	6	2	1	1	2	**12**
Benefit	Ident.	5	6	3		4	**18**	1	6	3	7	1	**18**
analysis	Descr.	1					**1**				3	1	**4**
Weakness	Ident.										4		**4**
analysis	Descr.										4		**4**
Trade-off	Ident.										1		**1**
analysis	Descr.										1		**1**
Risk	Ident.										1		**1**
analysis	Descr.										1		**1**
Evaluation													
Reflection													
Total		11	13	11	11	23	**69**	19	32	17	42	12	**122**

the upper-right quadrant. Furthermore, the grayscale of the solutions shows that solutions in the upper right quadrant are higher ranked by the external architect as well.

The scatter plot indicates a high corration between the different rankings. To confirm this, we calculated Spearman's rank correlations pairwise (Table V). As the results shown, all three rankings have a very high, significant correlation. From these correlations we may conclude that **more rationale is positively correlated with better quality**. To exclude the possibility that word count influences the rankings as well, we used Spearman's rank correlation to evaluate whether word count has an influence fairly high correlation, At the table shows, the correlations with word count are not significant.

B. *Effectiveness of the RCC*

Each of the rankings have the same solutions in the top 5, although the exact order slightly differs between each of them. Another observation is that in this top 5, four solutions come from participants of the test group. This is a clear indicator that the participants in the test group perform better than the control group. To test this hypothesis, we used Wilcoxon's signed rank exact test on each of the rankings to evaluate whether the test group performs better by comparing the median rank with the control group. As Table VI shows, the significance of each of the rankings is smaller than 0.05. We thus may conclude that **participants in the test group create solutions of better quality than the control group**.

Another aspect of effectiveness of the method concerns the time dimension. Many practitioners consider lack of time and/or budget (60.5%) as the most common cause of not documenting design rationale [5]. In this study, all participants had two hours to fulfill the exercise. Although on average, the participants of the test group took more time (m=130), than the control group (m=107.2), the difference was not significant (U=5.500, p=.138). In other words, our proposed method is non-invasive, and supports the process well, which was confirmed in the feedback by the participants. Hence, we may conclude that **participants in the test group do not take more time than participants in the control group to come with solutions of better quality**.

C. *Influence of Rationale Techniques*

As a next step, we compared the rationale techniques present in the solutions with their ranking. For the total number of rationale techniques applied, and for each individual rationale technique, we created two hypotheses:

- H_0^T There is no difference in the frequency of technique T between the groups.
- H_1^T There is a difference in the frequency of technique T between the groups.

After analysis, it turns out that for none of the hypotheses the difference is significant. Hence, the individual techniques used do not account for the observed phenomena. Interestingly, test groups use more rationale techniques: the frequency is increased with nearly 77%. Also, the spread of which rationale types occur are more comprehensive in the test group. The participants of the test group used almost all rationale types (16 / 19 were present), whereas the control group omitted 12 rationale types (7/19 were present). It is interesting to note that the increase in coded rationale types seem to be widespread, and not due to isolated elements. Comparing the rationale in the documentation of the top 5 solutions with the other solutions, reveals that all solutions in the top 5 contain more rationale techniques than the others.

D. *Threats to Validity*

As for all experiments, there are threats to validity that need to be addressed [18].

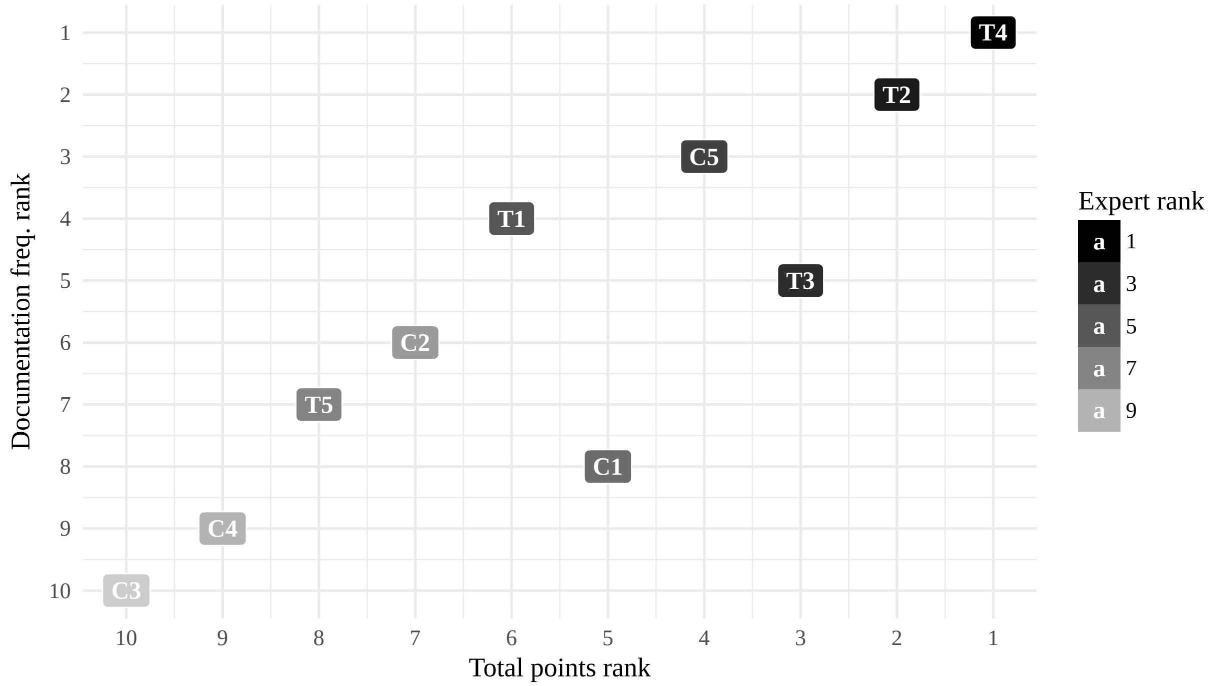

Figure 2. The rankings of Table II visualised, where the axes represent the total number of points attributed by the participants (horizontal) and the document frequency rank (vertical). The external expert rank of is represented by the grayscale, where darker (black) is a lower rank, i.e., a higher perceived quality. An approximate monotonic relationship can be observed between all three variables.

Table V
SPEARMAN'S RANK CORRELATION FOR THE RANKINGS. VALUES MARKED WITH + HAVE SIGNIFICANCE $p < 0,05$, VALUES MARKED WITH * HAVE SIGNIFICANCE $p < 0.001$, UNMARKED VALUES HAVE SIGNIFICANCE $p > 0,05$.

Ranking	Points	Documentation	External	Word count
Points	1	ρ = 0.879 +	ρ = 0.976 *	ρ= -0.534
Documentation	ρ= 0.879 +	1	ρ = 0.915 *	ρ= -0.612
External	ρ= 0.976 *	ρ = 0.915 *	1	ρ= -0.588
Word count	ρ=-0.534	ρ =-0.612	ρ =-0.588	1

a) Internal Validity: In this experiment, the participants form the largest treat to internal validity. Familiarity is a potential influencing factor: some of the participants are more familiar with the research than others. All of the participating architects were interviewed in an earlier stage of the research on how they document design rationale. The experiment was split in several sessions, as not all architects could participate at the same time. Therefore, all participants were given the same elaborate introduction to ensure all participants have equal knowledge before going into the experiment. Still, each architect is different and has different levels of knowledge about the architecture tool used in the experiment. Therefore, we controlled as many factors as possible to create equal groups.

Table VI
WILCOXON SIGNED RANK TEST. ELEMENTS MARKED WITH ** HAVE SIGNIFICANCE $p < 0.001$.

Variable	Test statistic Z	Exact p-value
Points	2.505	0.006**
Documentation	2.555	0.004**
External	2.505	0.006**

An influencing factor with respect to the quality of the rankings is the familiarity of the participants with each other. As all architects have the same affiliation, it can well be possible that architects know each others' way of writing, style and output habits of the participants they have to rank. As there is no way to control this, each architect had to rank 3 solutions. Additionally, the ranking and integrity of the architects was constantly emphasized: architects are supposed to rank solutions rather than individuals. This premise was emphasized by the researchers both during the experiment and in the instructions for the ranking.

b) External Validity: All architects that participated in this experiment work at the same organization. Furthermore, a population of only 10 architects is limited. For that reason, generalizing the results is difficult. Still, our findings point in the same direction as other researchers found in the domain of Software Architecture [13]–[15], [25], that all pointed out that some external stimulus such as checklists or prompting can help designers to reason more, and that more reasoning can have a positive impact on the quality [16].

c) Reliability: Rankings created by people are subjective by nature. Therefore, we studied and compared multiple rankings, and to which extent these rankings agree. The high correlations (¿.9) between the three independent variables points, external expert ranking and documentation frequency show reliability to be quite high.

VI. Conclusions

Many practicioners acknowledge the need for documenting design rationale, yet experience a lack of tool support [5]. Recent studies show that simple means can help in providing more rationale, such as a cardgame [14], checklists [15], or an external observer [13]. However, architecture frameworks and language have little to no support for capturing design rationale.

In this paper, we studied the effect of design rationale to enterprise architecture documentation on design quality. We propose the Rationale Capture Cycle to support architects in their reasoning process. The idea of the method is that it is non-intrusive, i.e., that it does not hamper architects in their design process.

To measure the effectiveness of the proposed method, we performed a controlled experiment. Ten experienced architects participated by creating an architecture for a specifically designed case. Half of the architects received the method as treatment, the other half did not. Through three different measures we evaluated the design quality: partial rankings based on points by the architects themselves, by coding the rationale the architects documented, and by an external architect.

The results of our experiment show a large agreement between the different rankings. From the results, we may conclude that **solutions with more rationale are significantly higher ranked in all three rankings**, and that **architects using the method perform significantly better**, confirming our initial research question that the RCC is an effective method. At the same time, the experiment shows that **the participants do not take significantly more time to produce the architecture**, countering the general believe that documenting rationale is too expensive in time and/or budget [5].

Although the results look promising, the experiment shows that not all architects fully utilized the RCC. As a consequence, the results are smaller than they could have been. Only 2 architects made full use of the cycle, causing a large difference in the comparison. Further research into how such a model can be fully implemented during architecture design is needed. We believe that with more research into actively embedding the reasoning model in architecture frameworks, further improvements can be made.

References

[1] ISO/IEC/IEEE, "Standard 42010 - Systems and software engineering — Architecture description," ISO/IEC/IEEE, Tech. Rep., 2011. [Online]. Available: http://www.iso-architecture.org/ieee-1471/

[2] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions," in *Working IEEE/IFIP Conference on Software Architecture*. IEEE, 2005, pp. 109–120.

[3] E. Poort, "RCDA: Risk- and cost-driven architecture: A solution architect's handbook," CGI Netherlands B.V., Tech. Rep., 2014.

[4] H. W. J. Rittel and M. M. Webber, "Dilemnas in a general theory of planning," *Policy Sci.*, vol. 4, pp. 155–168, 1973.

[5] A. Tang, M. A. Babar, I. Gorton, and J. Han, "A survey of architecture design rationale," *Journal of Systems and Software*, vol. 79, no. 12, pp. 1792–1804, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2006.04.029

[6] D. Perry and A. Wolf, "Foundations for the study of software architecture," *ACM SIGSOFT Sofware Engineering Notes*, vol. 17, no. 4, pp. 40–52, 1992.

[7] J. Conklin and M. Begeman, "gIBIS: a hypertext tool for exploratory policy discussion," in *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*. ACM, 1988, pp. 140–152.

[8] A. MacLean, R. Young, V. Bellotti, and T. Moran, "Questions, options, and criteria: Elements of design space analysis," *Human-Computer Interaction*, vol. 6, no. 3-4, pp. 201–250, 1991.

[9] W. Regli, X. Hu, M. Atwood, and W. Sun, "A survey of design rationale systems: Approaches, representations, capture and retrieval," *Engineering with Computers*, vol. 16, no. 3-4, pp. 209–235, 2000.

[10] J. Verries, A. Sahraoui, and M. Paludetto, "Design rationale in system design," in *International Conference on Systems Engineering*, 2008, pp. 380–385.

[11] T. O. Group, *he Open Group Architecture Framework Version 9.1*. The Open Group, 2011.

[12] J. Leite, F. Oquendo, and T. Batista, "Sysadl: A sysml profile for software architecture description," in *European Conference on Software Architecture*, ser. LNCS, vol. 7957. Springer, Berlin, 2013.

[13] M. Razavian, A. Tang, R. Capilla, and P. Lago, "In two minds: how reflections influence software design thinking," *Journal of Software: Evolution and Process*, vol. 28, pp. 394–426, 2016.

[14] C. Schriek, J. M. E. M. van der Werf, A. Tang, and F. Bex, "Software architecture design reasoning: A card game to help novice designers," in *European Conference on Software Architecture*, ser. LNCS, vol. 9839. Springer, Berlin, 2016, pp. 22–38.

[15] M. Keil, L. Li, L. Mathiassen, and G. Zheng, "The influence of checklists and roles on software practitioner risk perception and decision-making," *Journal of Systems and Software*, vol. 81, pp. 908–919, 2008.

[16] A. Tang, M. Tran, J. Han, and H. Vliet, "Design reasoning improves software design quality," in *Quality of Software Architectures*, ser. LNCS, vol. 5581. Springer, Berlin, 2008, pp. 28–42.

[17] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[18] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnel, and A. Wesslén, *Experimentation in Software Engineering*. Springer, Berlin, 2012.

[19] R. Deneckère, C. Hug, J. Onderstal, and S. Brinkkemper, "Method association approach: situational construction and evaluation of an implementation method for software products," in *Research challenges in information science*. IEEE, 2015, pp. 274–285.

[20] A. Tang, "Software designers, are you biased?" in *Workshop on SHAring and Reusing Architectural Knowledge*. ACM, 2011, pp. 1–8.

[21] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Addison Wesley, Reading, MA, USA, 2012.

[22] R. McCall, *Critical Conversations: Feedback as a Stimulus to Creativity in Software Design*. Springer, Berlin, 2013, pp. 11–40.

[23] J. Lee, "Design rationale systems: Understanding the issues," *IEEE Expert. Syst. their Appl.*, vol. 12, pp. 78—-85, 1997.

[24] J. Burge and D. Brown, *Design Rationale Types and Tools*. AI in Design Group, 1998.

[25] U. Heesch, P. Avgeriou, and A. Tang, "Does decision documentation help junior designers rationalize their decisions? a comparative multiple-case study," *Journal of Systems and Software*, vol. 86, pp. 1545—1565, 2013.