A Survey of the Use and Documentation of Architecture Design Rationale

Antony Tang[1] Muhammad Ali Babar[2] Ian Gorton[2] Jun Han[1]

[1]Faculty of ICT
Swinburne University of Technology
Australia
{atang,jhan}@ict.swin.edu.au

[2]National ICT Australia Ltd. and
University of NSW
Australia
{malibaba, ian.gorton}@nicta.com.au

Abstract

Many claims have been made about the problems caused by not documenting design rationale. The general perception is that designers and architects usually do not fully understand the critical role of systematic use and capture of design rationale. However, there is to date little empirical evidence available on what design rationale mean to practitioners, how valuable they consider them, and how they use and document design rationale during the design process. This paper reports an empirical study that surveyed practitioners to probe their perception of the value of design rationale and how they use and document background knowledge related to their design decisions. Based on eighty-one valid responses, this study has discovered that practitioners recognize the importance of documenting design rationale and frequently use them to reason about their design choices. However, they have indicated barriers to the use and documentation of design rationale. Based on the findings, we conclude that much research is needed to develop methodology and tool support for design rationale capture and usage. Furthermore, we put forward some research questions that would benefit from further investigation into design rationale in order to support practice in industry.

1. Introduction

Design rationale (DR) captures the knowledge and reasoning justifying the resulting design. This includes how a design satisfies functional and quality requirements, why certain designs are selected over alternatives and what type of system behavior is expected under different environmental conditions [10, 16]. Despite the growing recognition of the need for documenting and using architecture design rationale by researchers and practitioners [2, 3, 7], there is a lack of appropriate support mechanisms or guidelines on what are the essential elements of DR, and how to reason with and document DR for architecture design decisions. Recently adopted IEEE standards (1471-2000) for describing architecture [1] and architecture documentation methods like Views & Beyond (V&B) [5] raise the awareness about and provide some guidance on documenting design rationale; however, for reasons mentioned in section 2, each has its limitations.

This paper describes our initial investigations on the use and documentation of DR. In the long term, we aim to develop a conceptual framework and associated tools to facilitate the capture and use of DR. We believe that understanding the current industry practice of DR is one of the most important steps towards that goal. However, there is little empirical research that studies what practitioners think about DR, how they reason with DR and document DR, and what factors prevent them from documenting DR.

There are published claims of a lack of capturing and using DR [3, 31], which result in a common perception that architects generally do not realize the critical role of explicitly documenting the contextual knowledge about their design decisions. A lack of empirical evidence makes it difficult to support or refute these claims. Hence we set out to gather evidence from those who design architectures on a regular basis, in order to examine the attitudes of practitioners who have the most impact to the immediate and future use of DR approaches.

The purpose of this paper is to report the results of an empirical study that surveyed practitioners who had experience in architecture design. The results of this survey shed light on how design rationale are used and

Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)
0-7695-2548-2/05 $20.00 © 2005 IEEE

documented, and on the perceptions of those who make design decisions. As such, the objectives of the work described in this paper are:

- To understand architects' perceptions about architecture design rationale and the importance of the different elements of design rationale (such as design constraints, design strengths and weaknesses).
- To determine the frequency of reasoning with and documenting different elements of design rationale, the main reasons for not documenting design rationale, and the common methods, techniques, and tools used to document design rationale.
- To identify the potential challenges and opportunities for improving the use and documentation of design rationale in practice.

We have encountered several interesting findings which enabled us to identify a set of research questions that need to be explored in this line of research. Since a theory explaining the attitude and behaviour toward the use of design rationale does not exist, this study employs an inductive approach (i.e., using facts to develop general conclusions) as an attempt to move toward such a theory.

The paper makes three significant contributions to the Software Architecture (SA) discipline:

- It presents the design and results of the first survey-based empirical study in architecture design rationale practices.
- It provides information about how practitioners think about, reason about, document and use design rationale.
- It identifies the problems and contradictions of current DR practices. As a result, we propose a research agenda that aims to explore and enhance current architecture design rationale practices.

We discuss current approaches of DR in Section 2. We present our research approach in Section 3. Section 4 presents the results of the survey. A discussion of our findings and their limitations are in Sections 5 and 6 respectively. We identify areas for future work and make some concluding remarks in Section 7.

2. Background

2.1 DR Approaches in Software Engineering

Early work emphasizing the importance of design rationale in software design can be found in [21, 26]. Since then, the software engineering community has experimented with several DR approaches such as Issue Based Information Systems (IBIS) [13], Questions, Options, and Criteria (QOC) [18], Procedural Hierarchy of Issues (PHI) [19], and Design Rationale Language (DRL) [17]. Most of these methods have been adopted or modified to capture rationale for software design decisions [26] and requirements specifications [9, 15, 27]. Other approaches (e.g. [24, 25]) combine rationale and scenarios to elicit and refine requirements. While there are claims of several benefits of using these to capture DR, it is not clear how much or how far these techniques have been adopted by practitioners.

Design rationale have been considered an important part of SA since [23] laid the foundation for the evolving community of software architecture. In the years to follow, researchers have emphasized the need for documenting design rationale to maintain and evolve architectural artifacts and to avoid violating design rules that underpin the original architecture [2, 3]. The growing recognition of the vital role of documenting and maintaining rationale for architectural decisions has resulted in several efforts to provide guidance for capturing and using DR such as the IEEE 1471-2000 standard [1] and the Views and Beyond (V&B) approach to document SA [5].

However, both of these are deficient in several ways. For example, the former provides a definition of design rationale without further elaboration, while the latter provides a list of elements that comprise rationale without justifying why these elements are important and how the information captured is beneficial in different contexts. Moreover, it is not clear what types of specific information should be captured as design rationale.

Different approaches tend to characterize DR with different information. For example, Tyree & Akerman [31] provides a template that captures certain types of information as design rationale; the V&B [5] approach considers some other types of information (e.g. information cross-cutting different views) as design rationale; and the Architecture Rationalization Method (ARM) uses qualitative and quantitative rationale in design reasoning [29]. Thus, there is clear need for a common vocabulary or standard guidance so that practitioners understand the issues in reasoning with and documenting DR consistently.

2.2 Generic Design Rationale

According to the Cambridge dictionary, a rationale is a reason or intention for a particular set of thoughts or actions. When architects and designers make design decisions based on their reasoning, what do they consider as a reason or an intention? Is a requirement or a constraint an intention or reason enough for a design? Or is it some generic justification that allows designers to judge that a design is better than its alternatives? In this survey, we listed nine types of generic design rationales selected from various sources to test if and how our respondents perceive and use them. This set of generic rationale characterizes different aspects in which reasons can be portrayed and compared. Their selection is based on templates or methods proposed by researchers to capture design rationale [4] [31] [2] [29]. When selecting the generic DR, we choose those that can be used to reason about and justify decisions in general and excluded those that are specific to project requirements or design. We used common terminologies so that practitioners could relate to them. Since this is an exploratory study, the list is comprehensive but not exhaustive.

1. Design constraints
2. Design assumptions
3. Weakness of a design
4. Benefit of a design
5. Cost of a design
6. Complexity of a design
7. Am I certain that this design would work?
8. Am I certain that I or the team could implement it?
9. Tradeoffs between design alternatives

3. Research Approach

Considering the objectives of our research and available resources, we decided to use a survey research method to understand architects' perceptions and their current practices in architecture design rationale. A survey research method is considered suitable for gathering self-reported quantitative and qualitative data from a large number of respondents. Having reviewed the published literature on design rationale, we developed a survey consisting of 30 questions on design rationale understanding and practices and 10 questions on the demographics of the respondents. Some of the demographic questions were designed to screen the respondents and help identify data sets to be excluded from the final analysis. We ran a formal pilot study to test and refine the survey instrument. Data from the pilot study was not included in the analysis of the main survey. The feedback from the pilot study helped us refine the survey, which received ethics committee's approval.

We used an online web-based tool, Surveyor [20] to implement the survey questionnaire. The target population for the survey consisted of people with three or more years of experience in software development and who work as a software designer or architect. Considering the fact that software designers usually have major time constraints, it was not feasible to attempt random sampling because the response rate could be low. Consequently, we used availability and snowballing sampling techniques. The major drawback of these sampling techniques is that the results are statistically generalizable only to the population with the same characteristics as the samples. Being an exploratory study, we believe our sampling techniques are reasonable.

We invited a pool of designers and architects drawn from the industry contacts of the four investigators, and past and current postgraduate students of Swinburne University of Technology and the University of New South Wales who are working in the software industry. We also requested the invitees to forward the invitation to others who were eligible for participation. For access control and data validation purposes, the URL of the survey website was sent in an email.

4. Survey Findings

The survey questionnaire was divided into seven main parts. The perception of the importance of DR, the use of DR and the documentation of DR are discussed and analyzed in this paper together with the profile of the respondents. Architecture evaluation in organizations, architecture enhancements and risk undertakings in architecture design are the other three parts which will be reported separately. Readers who are interested in the statistics and the questionnaire are referred to [30].

4.1. Demographic Data

We directly sent survey invitations to 171 practitioners. Our invitation was forwarded to 376 more people by the original invitees, meaning 547 invitations were sent. We received a total of 127 responses, which corresponds to 23% response rate. Lack of resources and anonymity did not allow us to contact non-respondents. Out of the total responses, we decided to exclude 46 responses from the analysis as they were incomplete or the respondents did not meet the work experience criteria (minimum 3 years software development experience).

In summary, 80.2% of our respondents were male and 19.8% are female. 67.9% of respondents live in Australasia, 28.4% reside in Asia and 3.7% did not specify the region of their residence. The respondents' experience in the information technology industry varies between 4 years and 37 years with an average of 17.12 years. On average, they have worked as a designer or architect for 9.75 years. The average length of working with one organization (current or previous) is 7.65 years and the average number of co-workers on the current (or last) project is 25 people. 85.2% of the respondents have received an IT related tertiary qualification.

This demographic gives us confidence that we have gathered data from practitioners who are experienced in software architecture and design. Despite not being able to apply systematic random sampling because of the reasons described in section 3, the results are representative of designers with similar characteristics.

4.2. Job Nature of Architects / Designers

In the survey, we asked respondents to tell us the primary tasks they perform as an architect or application designer. A primary task is a task in which they spend at least 10% of their time on. The objective is to find out the scope of their role. A summary of the percentages of respondents who perform those primary tasks are listed below:

- overall system design (86.4%)
- requirements or tender analysis (81.5%)
- non-functional requirements design (64.2%)
- software design and specification (58%)
- project management tasks (50.6%)
- IT planning and proposal preparation (49.4%).
- data modeling (44.4%)
- implementation design (42%)
- program design and specification (35.8%),
- test planning and design (29.6%)
- training (19.8%)

Our typical respondent's main efforts are spent in the early project phases including high level design, requirements and tender analysis, overall design, non-functional design and software design. Most of them also have management responsibilities such as project management and IT planning. To a lesser extent, they perform detailed design and implementation activities.

4.3. Designer's Perception of the Importance of Design Rationale

As there is little empirical evidence on how important DR is considered by designers, we posed a number of questions to this end. Respondents were asked to indicate how often they reason about their design choices and whether they think that design rationale are important to justify their design choices.

The responses to those questions revealed (Table 1 and 2) that the majority of designers frequently apply reasoning to justify their architectural choices and they also consider that DR are important to justify their design choices.

	Never		to		Always
	1	2	3	4	5
No of Respn (%)	0 (0%)	1 (1.2%)	8 (9.9%)	34 (42%)	38 (46.9%)

Table 1: Frequency of Reasoning about Design Choices

	Not Important		to		Very Important
	1	2	3	4	5
No of Respn (%)	0 (0%)	1 (1.2%)	11 (13.6%)	30 (37%)	39 (48.1%)

Table 2: Importance of DR in Justification

We also asked the respondents about the frequency of considering alternative architecture designs (explanation for alternative architecture designs was provided) during their design process, as this is another indicator of the awareness of reasoning about design choices and the rigor that needs to be employed during this process. The responses to this question are provided in Table 3. The result indicates that the majority of respondents compare between alternative designs before selecting a particular architectural design among available alternatives.

	Never		to		Always
	1	2	3	4	5
No of Respn (%)	0 (0%)	1 (1.2%)	15 (18.5%)	31 (38.3%)	34 (42%)

Table 3: Frequency of Considering Alternative Designs

We asked the respondents to rank the importance of each of the nine generic DRs listed in the survey. This ranking reflects the perception of respondents towards how useful a given DR is in design. Since decision making is something our respondents do on a regular basis, their perception of DR's importance should reflect the reasoning process that is usually done intuitively. Table 4 presents the responses to this question. The majority of respondents considered that all nine DR are important.

The responses for all rationales are skewed towards the *very important* end. *Benefits of design, design constraints* and *certainty of design* receive the highest support with combined level *4* and *5* percentages of 90.12%, 87.65% and 85.19% respectively. All other rationales are also considered important with the majority of respondents selecting level *4* or *5*. This shows that most designers perceived that these rationales are important in reasoning about design decisions.

(Results in %)	Not Important 1	2	to 3	4	Very Important 5
Design Constraints	0.0	1.2	11.1	38.3	49.4
Design Assumptions	3.7	7.4	14.8	44.4	29.6
Weakness	2.5	7.4	28.4	43.2	18.5
Costs	0.0	7.4	14.8	43.2	34.6
Benefits	1.2	1.2	7.4	54.3	35.8
Complexity	0.0	2.5	25.9	46.9	24.7
Certainty of Design	0.0	3.7	11.1	29.6	55.6
Certainty of Implementation	2.5	4.9	16.1	32.1	44.4
Tradeoffs	0.0	4.9	30.9	44.4	19.8

Table 4: Importance of Each Generic Rationale

Apart from the above-mentioned nine generic rationales, we also asked the respondents to add other rationale that they use for making architectural design choices. A significant number of the respondents (twenty eight), mentioned additional types of factors that influence their design choices. We have classified those factors into three broad categories. These are:

Business Goals Oriented

1. Enterprise strategies, technical directions and organizational standards
2. Management preferences and acceptance
3. Adherence to industry standards
4. Vendors relationship

Requirements Oriented (functional/non-functional)

5. Fulfill functional and non-functional requirements
6. Satisfy client business motivations
7. Buy vs. build decisions
8. Maintenance and expected life-cycle of products

Constraints and Concerns

9. Viability of solutions
10. Consider existing architecture constraints
11. Current IT architecture and capabilities
12. Compatibility with existing systems
13. Has the design been used before and is it successful
14. Technology and tools availability
15. Prototype and staged delivery
16. Time to market
17. Time availability
18. Risk

These rationales show a variety of common factors that are used in reasoning during the design processes. We considered that these concrete types of rationales are specific to a need of a project or an organization. It is worth mentioning that the main difference between these rationales and the nine generic rationales are in the nature of the reasoning involved. Generic rationale allows designers to compare and judge between alternative designs by using the same generic criteria whereas the concrete rationales are specific reasons that motivate decisions to be made. Both of the lists are not definitive, rather extendable according to needs. A more detailed discussion of their differences is in section 5.

4.4. Using Design Rationale

Another important area of the survey was how frequently DR are used. An aim of the study is to discover whether respondents' perceptions of the importance of DR and their behavior (i.e. what they do) are consistent. Therefore, the same set of DR we presented and discussed in the previous sections were used to query our respondents. In this section, we present the results of a multi-item question on how often they use the generic rationales to reason about architectural decisions. Most respondents say that they frequently or always use the nine generic DR listed in the questionnaire. Table 5 summarizes the frequency of using the different types of rationales.

The results show that *Design constraint* rationale is used most frequently. The reason for the high usage of this could be that designers are usually expected to explore the solution space within certain business and technical constraints. These constraints are consequently prominent in their minds and must be taken into account from the beginning of a project.

(Results in %)	Never 1	2	to 3	4	Always 5
Design Constraints	0.0	0.0	12.3	42.0	45.7
Design Assumptions	2.5	2.5	30.9	33.3	30.8
Weakness	1.2	8.6	34.6	37.0	18.6
Costs	1.2	9.9	19.8	38.3	30.8
Benefits	1.2	1.2	12.3	49.4	35.9
Complexity	0.0	2.5	27.2	34.6	35.7
Certainty of Design	2.5	1.2	11.1	32.1	53.1
Certainty of Implementation	3.7	3.7	16.0	33.3	43.3
Tradeoffs	0.0	6.2	29.6	42.0	22.2

Table 5 - Design Rationale Frequency of Use

Other more frequently used rationales are *benefits of design, certainty of design* and *certainty of implementation.* The combined usage frequencies (level *4* and *5*) for these rationales are 85.3%, 85.2% and 76.6% respectively. We suspect that designers frequently use these types of rationales as they have to make a business case for their architectural choices to the management and justify their design choices using technical arguments to architecture reviewers and technical stakeholders such as programmers, implementers and maintainers. That is why they use rationales more often that can help them to justify their architectural decisions.

On the other hand, respondents are less likely to use those rationales that can highlight the weaknesses of their design decision. That is why the combined usage frequencies (level *4* and *5*) reported by respondents are: *weakness* of a design (55.6%), *costs* (69.1%) and *complexity* (70.3%). This tendency of designers to pay relatively less attention to the weaknesses of their design decisions can also be explained by the Lassing et.al.'s warning against gathering scenarios to evaluate an architecture by the designers themselves, as it is highly likely they would come up with the scenarios that have already been addressed by the proposed architecture [14]. Thus, we hypothesize that designers unknowingly look for the positive rationales to support a design and pay less attention to the negative rationales.

4.5. Documenting Design Rationale

Several arguments have been made about the importance of documenting key architecture decisions along with the contextual information [22, 31]. It is important that DR are documented to a sufficient extent in order to support the subsequent implementation and maintenance of systems. With regards to DR documentation attitude and practice, we paid special attention to the frequency of documenting discarded design decisions, frequency of documenting each of the generic rationales, the reasons for not documenting design decisions (barriers to DR documentation), and method and tools used for documenting DR. Table 6 presents the breakdown of the responses to the question on documenting discarded design decision.

	Never		**to**		**Always**
	1	**2**	**3**	**4**	**5**
No of Respn (%)	11 (13.6)	18 (22.2)	17 (21)	19 (23.5)	16 (19.8)

Table 6: Frequency of Documenting Discarded Decisions

44% of the respondents document discarded decision *very often*. 36% of the respondents do not document discarded decisions. This is likely because designers are under pressure to produce design specifications on schedule. At this stage, we are not aware of any software development or project management methodology that mandate the documentation of discarded decisions or methodically schedule time for such activities to take place. However, documenting the discarded decisions can help newcomers to the project understand the reasons for discarding design alternatives and expedite that understanding during the maintenance phase of the project.

	Never		**to**		**Always**
(Results in %)	**1**	**2**	**3**	**4**	**5**
Design Constraints	1.2	2.5	13.6	19.7	63.0
Design Assumptions	3.7	3.7	13.6	25.9	53.1
Weakness	3.7	23.5	37.0	14.8	21.0
Costs	7.4	16.0	30.9	21.0	24.7
Benefits	2.5	9.9	18.5	32.1	37.0
Complexity	3.7	9.9	35.8	30.9	19.7
Certainty of Design	18.5	14.8	19.8	24.7	22.2
Certainty of Implementation	18.5	17.3	24.7	22.2	17.3
Tradeoffs	6.2	18.5	25.9	32.1	17.3

Table 7: Frequency of Documenting Generic DR

Respondents were also asked to indicate the overall frequency of documenting DR. 62.9% of the respondents replied that they completely document DR, which is an encouraging finding considering the common perception of design rationale not being widely documented.

We also investigated the frequency of documenting each of the generic rationale. Table 7 summarizes the frequency of documentation for each of the nine generic DR used in this research. The results show that *design constraints* and *design assumptions* are documented very frequently but the level of documentation is relatively lower for other types of rationale. 27.2% of the respondents replied that they never or seldom document *design weakness*. Similarly, 33.3% of respondents said they never or seldom document *certainty of design*. 35.8% of them said they never or seldom document *certainty of implementation*. These findings appear to agree with our previous assertion that negative rationales receive relatively less attention.

Based on these results, it appears that design rationales are commonly documented by software designers and architects. However, it also appears that the reasons about why a design is chosen and why it is better than alternative designs are usually not documented. We do not have any theoretical grounds for explaining this phenomenon.

While the level of documentation is relatively high, the survey results give us no insight as to whether the rationales are sufficiently documented so that other designers can understand the architecture design without additional assistance. This raises two issues worthy of further investigation, namely:

(a) identify the rationale documented by architects and evaluate their effectiveness in explaining the designs;

(b) identify how the documented rationale are used in the development life-cycle.

4.5.1 Barriers to Documenting DR

We were also interested in identifying and understanding the reasons for not documenting DR. We believe that it is important to identify those factors that undermine efforts in documenting and maintaining DR. The respondents were given a list of reasons that are common causes of non-documentation in software engineering such as perceived usefulness, project budget and lack of time. The respondents also had a text box to provide other reasons.

Topic of questions	Percent of respondents	Number of Respondents
No standards	42%	34
Not aware of	4.9%	4
Not useful	9.9%	8
No time/budget	60.5%	49
No suitable tool	29.6%	24

Table 8: Reasons for Not Documenting DR

Table 8 summarizes the responses to the reasons for not documenting DR. These results reveal that lack of time/budget (60.5%) is considered the most common cause of not documenting design rationale. There is also a lack of appropriate standards and tools to support the documentation process. Only 4.9% of the respondents were not aware of the need of documenting DR, while 9.9% of the respondents said that documenting DR is not useful. A few respondents also provide several other reasons for not documenting DR. These reasons are:

- Lack of formal review process
- Not required for non-complex solutions
- Afraid of getting into a long cycle of design review
- Not required for low impact solution
- Dynamic nature of technology and solutions make it useless to document DR.
- It is not required for high level decision making

In summary, the reasons for not documenting DR can be classified into these groups: (a) the lack of standards and processes to guide why, how, what and when design rationale should be documented; (b) the time and budget constraints of projects; (c) the question of whether the cost and benefit of rationale documentation can be justified. These reasons are analogous to those concerning requirements traceability documentation in immature software development organizations [28]. Since the sample population is not specific to an industry or capability maturity level, the results may indeed reflect the general architecture design practice.

4.5.2 Methods and Tools for Documenting DR

An important part of any task in the software development lifecycle is the availability of process support and suitable tools to enhance productivity. It is important to identify what type of support is available to designers to improve DR practices. Hence the survey included a question on the methods and tools used for documenting DR. Twenty respondents provided comments to this question. We list the methods and tools used by the respondents to document DR below:

- Apply organization standards and templates to document using Word / Visio / Excel / Powerpoint
- UML tools
- IBM GS Methodology
- Document architecture decisions using formal method and notation
- Internally developed tools
- QMS Design Template document
- Requirements Traceability Matrix
- Architecture tool CORE

Our respondents are using proprietary tools, proprietary templates, the Microsoft Office suite or UML design tools to document DR. As we suggested earlier, there is little awareness about the standards like IEEE 1417-2000 and a methodology such as V&B. DR tools like gIBIS [6] are not used. Although these results are anecdotal evidence, they point to the lack of industry standards as well as proper tools to capture, maintain and trace DR during the development lifecycle.

4.6. Comparing Usage and Documentation of Design Rationale

Given that DR are recognized by our respondents as important, it is revealing to compare the survey results concerning importance, use and documentation of each of the nine generic rationales. Table 9 presents the combined results from the last three sections. The scale is condensed by combining level *4* and level *5 (See the scale in the previous sections to interpret the results)*.

	Level of Importance	Frequency of Use	Frequency to Document
Benefit of Design	90.1%	85.3%	69.1%
Design Constraint	87.6%	87.6%	82.7%
Certainty that design would work	85.2%	85.2%	46.9%
Cost of Design	77.7%	69.1%	45.7%
Certainty that design is implementable	76.5%	76.5%	39.5%
Design Assumption	74.0%	64.1%	79.0%
Complexity of Design	71.6%	70.3%	50.6%
Tradeoffs between alternatives	64.2%	64.2%	49.4%
Weakness of Design	61.7%	55.6%	35.8%

Table 9: Design Rationales Usage

We used Spearman's Rank Order Correlation (rho) to test correlations between the *Level of Importance* and the *Frequency of Use* for the nine generic DR. This revealed that they are all correlated with r values all above 0.5 with the exception of *design complexity*, and all of them tested significant with $p < 0.01$. This indicates that there is a strong relationship between what respondents believe and what they practice. We also observe that across most DR, the usage frequency is less than the perception of importance, and the documentation frequency is less than the usage frequency. This can be considered a strong indicator that our respondents are convinced of the importance of DR and use them more frequently than they document them. Lack of documentation may be caused by the reasons put forwarded by the respondents (section 4.5.1). This may be a reason for the claims of design knowledge vaporization [3, 31].

5. Discussions of Findings

Based on the survey, there is evidence to support that DR are an important part of design, and practitioners believe that DR should be documented. There is also a general perception that methodology and tool support for DR is lacking and there are barriers to DR documentation. These findings lead to a number of areas that require further investigation.

Different Forms of DR: Respondents told us about the different types of rationale they document. These rationales represent the reasons behind the need for a solution. We call them concrete rationales. The generic DR we provided in the questionnaire are reasons to select a design from amongst the alternatives. But there is a difference in nature between the two forms of rationale. As such, we conclude that the generic DR can be used as a function to measure and compare alternative designs using concrete rationale as inputs. The resulting measurements can be a scale (e.g. percentages for measuring risk or value terms for measuring cost) or a rank (e.g. high / low, strong / weak). A *tradeoff* is similar in that it compares alternative designs with their DR justifications as inputs. Examples of generic DR for reasoning are:

- Cost of Design (*functional requirement*, *corporate strategy*, *current IT structure* and others)
- Complexity of Design (*functional requirement*, *non-functional requirement*, *intended design* and so on)
- Tradeoffs (*Design 1 DRs*, *Design 2 DRs*, etc.)

It appears that reasoning with generic DR may often be done intuitively. They may be used but they are seldom documented systematically. The distinction that respondents draw between generic and concrete rationale could potentially provide a structure to explicitly reason about design decisions based on the specific needs that drive a design. Follow-up interview and inspection of specifications must be undertaken to test this hypothesis.

Designers' Attitude: Respondents frequently use DR to justify design choices. When we examine the list of DR they use, it appears that those DR that positively justify the design receive more attention than those negative rationales that explain why the design may have issues. That leads us to suspect that there might be a tendency to present "good news" rather than "bad news" during the design process. An analogous finding [11] may give us some insights to this behavior. In many industry scenarios that we have encountered, some architects have a tendency to promote a design based on the benefits of new technologies. However they often do not explain the potential negative impacts of the new approach. Establishing if such a bias is commonly exhibited in architecture would be useful, because awareness of this phenomenon would help architects to be more objective in the assessment and selection of designs.

Design Rationale Methodology Support: Some of the reasons for not documenting DR are due to budget constraints and lack of methodology. Given that most respondents consider DR important and documentation of DR useful, there needs to be guidelines under which the use and documentation of DR will provide greater benefits than the costs involved. This means that the need for DR documentation should be context dependent. For instance, a non-complex system may require little DR documentation since it can be reconstructed easily [8]. Our literature review shows that there is no comprehensive methodology to guide how we should use rationale-based techniques to design systems. Therefore, further studies of the use and documentation of DR to provide a methodology would be most beneficial.

Design Rationale Tool Support: Tool support for design rationale capture and retrieval is inadequate. The various tools that respondents reported using, including word processors and UML-based tools, do not have traceability features to support systematic DR description and retrieval. Therefore, it is important to understand how to best capture, represent and use DR and then develop such tools to support a DR enabled development environment.

In summary, the survey has gathered invaluable information about how designers use DR. It has confirmed that the use of DR in the architecture design process continues to be challenging for practitioners.

6. Limitations

Our study has a several shortcomings. Like most surveys in software engineering, our study faced reliability and validity threats. Following the guidelines provided in [12], we put certain measures in place to address validity and reliability issues. For example, the research instrument underwent rigorous evaluation by experienced researchers and practitioners, all the questions were tested in a pilot study, and respondents were assured of anonymity and confidentiality. However, completely eliminating the possibility of bias error is difficult.

The results may also suffer from non-response error. If only those with a positive opinion about the DR responded, the results would be biased. However, we are unable to identify non-respondents because the survey was anonymous. Geographical location of the respondents, mainly the Asia Pacific region, is another major limitation as the findings may not be generalized globally.

A further limitation of our study is the non-existence of a proven theory of designers' attitude towards documenting DR to guide our research. Hence we consider this research as an exploratory effort to draw some general conclusions that can help identify future research directions that can develop and validate such a theory.

7. Future Work and Conclusion

Our long-term research objective is to improve the design reasoning process for software architects. We are approaching this by firstly understanding the key elements of the process, and then attempting to develop appropriate support mechanisms and tools to facilitate the design process. In this study, we have gained important insights into the issues of DR use and documentation in the software industry. We found that practitioners view DR as important but there is a lack of methodology and tool support.

To achieve these aims, we need to identify the technical and socio-technical factors that influence those design decisions that have architectural implications. Some of the significant issues that we plan to pursue in our continued research in architecture design rationale are:

- How can DR be explicitly used to objectively measure or quantify the relative merits of a design to improve the decision making process?
- Whether there is a common tendency, intentionally or unintentionally, to focus on positive aspects of design decisions and ignore the negative aspects.
- What are the design or system circumstances that influence the use and documentation of design rationale?
- Under what situations would the use and documentation of DR provide a positive return on investment? Such a mechanism will help make decisions about the level of detail and circumstance under which to document DR.

We plan to design and execute a large scale field study consisting of multiple case studies, as described by Yin [32], and successfully demonstrated by Curtis et. al. [7] in a large scale research into the software design process. Some of the techniques that we plan to use to study practitioners' attitudes towards DR use and documentation are in-depth interviews and examination of design specifications. We expect these experimental techniques will enable us to discover the answers to the questions above. The results will allow us to develop a DR methodology and associated tools to enhance the future use and documentation of DR during software design and maintenance.

8. References

[1] *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*: IEEE Standard No. 1471-2000, Available at http://shop.ieee.org/store/.

[2] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Boston: Addison Wesley, 2003.

[3] J. Bosch, "Software Architecture: The Next Step," European Workshop on Software Architecture, pp, 2004.

[4] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford, *Documenting Software Architectures : Views and Beyond*, Boston ed: Addison Wesley, 2002.
[5] P. Clements, F. Bachmann, L. Bass, D. Garlan, j. Ivers, R. Little, R. Nord, and J. Stafford, *Documenting Software Architectures: Views and Beyond*: Addison-Wesley, 2002.
[6] J. Conklin and M. L. Begeman, "gIBIS: A Hypertext Tool for Exploratory Policy Discussion," *ACM Transactions on Office Information Systems*, vol. 6, pp. 303-331, 1988.
[7] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM*, vol. 31, 1988.
[8] A. Dutoit and B. Peach, "Rationale Management in Software Engineering," in *Handbook of Software Engineering and Knowledge Engineering*, 2000.
[9] Dutoit A.H. and Peach B., "Rationale-Based Use Case Specification," *Requirements Engineering*, vol. 3, pp. 3-19, 2002.
[10] T. R. Gruber and D. M. Russell, "Design Knowledge and Design Rationale: A Framework for Representing, Capture, and Use," Knowledge Systems Laboratory, Standford University, California, USA, California KSL 90-45, 1991.
[11] M. Keil, H. J. Smith, S. Pawlowski, and L. Jin, "'Why didn't somebody tell me?': climate, information asymmetry, and bad news about troubled projects," *SIGMIS Database*, vol. 35, pp. 65-84, 2004.
[12] B. Kitchenham and S. L. Pfleeger, "Principles of Survey Research, Parts 1 to 6," *Software Engineering Notes*, 2001-2002.
[13] W. Kunz and H. W. J. Rittel, "Issues As Elements of Information Systems." Institute of Urban & Regional Development, University of California, Berkeley, 1970.
[14] N. Lassing, P. Bengtsson, J. Bosch, and H. V. Vliet, "Experience with ALMA: Architecture-Level Modifiability Analysis," *Journal of Systems and Software*, vol. 61, pp. 47-57, 2002.
[15] J. Lee, "Extending the Potts and Bruns Model for Recording Design Rationale," 13th International Conference on Software Engineering, pp 114-125, 1991.
[16] J. Lee, "Design Rationale Systems: Understanding the Issues," *IEEE Expert*, vol. 12, pp. 78-85, 1997.
[17] J. Lee and K.-Y. Lai, "What's in Design Rationale?," *Human-Computer Interaction*, 1991.
[18] A. MacLean, R. M. Young, V. M. E. Bellotti, and T. P. Moran, "Questions, Options, and Criteria: Elements of Design Space Analysis," *Human-Computer Interaction*, vol. 6, pp. 201-250, 1991.
[19] R. McCall, "PHIBIS: Procedural Hierarchical Issue-Based Information Systems," Proc. Int'l. Congress on Planning and Design Theory, pp, 1987.
[20] ObjectPlanet Inc., "Surveyor: Web-based Survey Application," 2002, http://www.sharewareconnection.com/surveyor.htm.
[21] D. Parnas and P. Clements, "A rationale design process: How and why to fake it," *IEEE Transactions on Software Engineering*, vol. SE-12, pp. 251-257, 1985.
[22] D. Parnas and P. Clements, "A Rationale Design Process: How and Why to Fake It," *IEEE Transactions of Software Engineering*, vol. 12, pp. 251-257, 1986.
[23] D. E. Perry and A. L. Wolf, "Foundations for the Study of Software Architecture," *ACM SIGSOFT, Software Engineering Notes*, vol. 17, October 1992.
[24] C. Potts, "ScenIC: A Strategy for Inquiry-Driven Requirements Determination," Proc. Int'l. Symposium on requirements engineering, pp 58-65, 1999.
[25] C. Potts, K. Takahashi, and A. I. Anton, "Inquiry-Based Requirements Analysis," *IEEE Software*, vol. 11, pp. 21-32, 1994.
[26] Potts C. and Burns G., "Recording the Reasons for Design Decisions," 10th International Conference on Software Engineering, pp 418-427, 1988.
[27] B. Ramesh and B. Dhar, "Supporting Systems Development by Capturing Deliberations During Requirements Engineering," *IEEE Transactions on Software Engineering*, vol. 18, pp. 498-510, 1992.
[28] B. Ramesh and M. Jarke, "Towards Reference Models for Requirements Traceability," *IEEE Transactions on Software Engineering*, vol. 27, pp. 58-93, 2001.
[29] A. Tang and J. Han, "Architecture Rationalization: a Methodology for Architecture Verifiability, Traceability and Completeness," 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems ECBS 2005, U.S.A., pp 135-144, 2005.
[30] A. Tang, M.A. Barbar, I. Gorton, and J. Han, "A Survey of Architecture Design Rationale," Swinburne University of Technology SUTICT-TR2005.02, 2005.
[31] J. Tyree and A. Akerman, "Architecture Decisions: Demystifying Architecture," *IEEE SOFTWARE*, vol. 22, pp. 19-27, 2005.
[32] R. K. Yin, *Case Study Research: Design and Methods*, vol. 5, 3rd ed: Sage Publications, 2002.