Faculty of Information and Communications Technologies
Centre for Component Software and Enterprise Systems

A Survey of Architecture Design Rationale

(A Joint Research Project by Swinburne University of Technology and NICTA)

Technical Report: SUTICT-TR2005.02
CeCSES Centre Report: SUT.CeCSES-TR008

Antony Tang, Swinburne University of Technology
Muhammad Ali Babar, National ICT Australia Ltd. and University of NSW
Ian Gorton, National ICT Australia Ltd. and University of NSW
Jun Han, Swinburne University of Technology
31st May 2005

SWINBURNE UNIVERSITY OF TECHNOLOGY

Table of Contents

Swinburne University of Technology and NICTA

A Survey of Architecture Design Rationale

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 [1, 2]. Despite the growing recognition of the need for documenting and using architecture design rationale by researchers and practitioners [3-5], 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 [6] and architecture documentation methods like Views & Beyond (V&B) [7] 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, 8], 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 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.

The survey investigated a number of related areas surrounding design rationale. In addition to the work reported in this paper, we also studied design rationale and its use in relation to system maintenance in a separate paper. We have gathered information on how respondents assess risk in architecture design and it will be studied separately.

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. Appendix A lists the questions used in the questionnaire and Appendix B provides a summary of the statistics collected in the survey.

2. Background

2.1 DR Approaches in Software Engineering

Early work emphasizing the importance of design rationale in software design can be found in [9, 10]. Since then, the software engineering community has experimented with several DR approaches such as Issue Based Information Systems (IBIS) [11], Questions, Options, and Criteria (QOC) [12], Procedural Hierarchy of Issues (PHI) [13], and Design Rationale Language (DRL) [14]. Most of these methods have been adopted or modified to capture rationale for software design decisions [10] and requirements specifications [15-17]. Other approaches (e.g. [18, 19]) 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 [20] 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 [3, 5]. 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 [6] and the Views and Beyond (V&B) approach to document SA [7].

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 [8] provides a template that captures certain types of information as design rationale; the V&B [7] 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 [21]. 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 [22] [8] [5] [21]. 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 [23] 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. 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 [24].

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. However, we believe that non-respondents did not cause any systemic bias in the collected data. 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.

	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

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.

	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 for 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.

	Not Important		to	Very Important	
(Results in %)	1	2	3	4	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.

	Never		to		Always
(Results in %)	1	2	3	4	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 [25]. 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 [8, 26]. 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 [27].

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 [28] are not used. These results 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 the reason for the claims of design knowledge vaporization [3, 8].

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 [29] 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 not require DR documentation. 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 [30], 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 [31], and successfully demonstrated by Curtis et. al. [4] 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. Gruber, T.R. and D.M. Russell, *Design Knowledge and Design Rationale: A Framework for Representing, Capture, and Use*. 1991, Knowledge Systems Laboratory, Standford University, California, USA: California.
2. Lee, J., *Design Rationale Systems: Understanding the Issues.* IEEE Expert, 1997. **12**(3): p. 78-85.
3. Bosch, J. *Software Architecture: The Next Step*. in *European Workshop on Software Architecture*. 2004.
4. Curtis, B., H. Krasner, and N. Iscoe, *A Field Study of the Software Design Process for Large Systems.* Communications of the ACM, 1988. **31**(11).
5. Bass, L., P. Clements, and R. Kazman, *Software Architecture in Practice*. 2003, Boston: Addison Wesley.
6. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*: IEEE Standard No. 1471-2000, Available at http://shop.ieee.org/store/.
7. Clements, P., et al., *Documenting Software Architectures: Views and Beyond*. 2002: Addison-Wesley.
8. Tyree, J. and A. Akerman, *Architecture Decisions: Demystifying Architecture.* IEEE SOFTWARE, 2005. **22**(2): p. 19-27.
9. Parnas, D. and P. Clements, *A rationale design process: How and why to fake it.* IEEE Transactions on Software Engineering, 1985. **SE-12**: p. 251-257.
10. Potts C. and Burns G. *Recording the Reasons for Design Decisions*. in *10th International Conference on Software Engineering*. 1988.
11. Kunz, W. and H.W.J. Rittel, *Issues As Elements of Information Systems*. 1970: Institute of Urban & Regional Development, University of California, Berkeley.
12. MacLean, A., et al., *Questions, Options, and Criteria: Elements of Design Space Analysis.* Human-Computer Interaction, 1991. **6**: p. 201-250.
13. McCall, R. *PHIBIS: Procedural Hierarchical Issue-Based Information Systems*. in *Proc. Int'l. Congress on Planning and Design Theory*. 1987: American Society of Mechanical Engineers.
14. Lee, J. and K.-Y. Lai, *What's in Design Rationale?* Human-Computer Interaction, 1991(Special issue on Design Rationale).

15. Lee, J. *Extending the Potts and Bruns Model for Recording Design Rationale*. in *13th International Conference on Software Engineering*. 1991.
16. Dutoit A.H. and Peach B., *Rationale-Based Use Case Specification.* Requirements Engineering, 2002. **3**(1): p. 3-19.
17. Ramesh, B. and B. Dhar, *Supporting Systems Development by Capturing Deliberations During Requirements Engineering.* IEEE Transactions on Software Engineering, 1992. **18**(6): p. 498-510.
18. Potts, C., K. Takahashi, and A.I. Anton, *Inquiry-Based Requirements Analysis.* IEEE Software, 1994. **11**(2): p. 21-32.
19. Potts, C. *ScenIC: A Strategy for Inquiry-Driven Requirements Determination*. in *Proc. Int'l. Symposium on requirements engineering*. 1999.
20. Perry, D.E. and A.L. Wolf, *Foundations for the Study of Software Architecture.* ACM SIGSOFT, Software Engineering Notes, October 1992. **17**(4).
21. Tang, A. and J. Han. *Architecture Rationalization: a Methodology for Architecture Verifiability, Traceability and Completeness*. in *12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems ECBS 2005*. 2005. U.S.A.: IEEE.
22. Clements, P., et al., *Documenting Software Architectures : Views and Beyond*. Boston ed. 2002: Addison Wesley. 512.
23. ObjectPlanet Inc., *Surveyor: Web-based Survey Application*. 2002.
24. Tang, A., et al., *A Survey of Architecture Design Rationale*, in *Swinburne FICT Technical Report*. 2005, Swinburne University of Technology.
25. Lassing, N., et al., *Experience with ALMA: Architecture-Level Modifiability Analysis.* Journal of Systems and Software, 2002. **61**(1): p. 47-57.
26. Parnas, D. and P. Clements, *A Rationale Design Process: How and Why to Fake It.* IEEE Transactions of Software Engineering, 1986. **12**(2): p. 251-257.
27. Ramesh, B. and M. Jarke, *Towards Reference Models for Requirements Traceability.* IEEE Transactions on Software Engineering, 2001. **27**(1): p. 58-93.
28. Conklin, J. and M.L. Begeman, *gIBIS: A Hypertext Tool for Exploratory Policy Discussion.* ACM Transactions on Office Information Systems, 1988. **6**(4): p. 303-331.
29. Keil, M., et al., *'Why didn't somebody tell me?': climate, information asymmetry, and bad news about troubled projects.* SIGMIS Database, 2004. **35**(2): p. 65-84.
30. Kitchenham, B. and S.L. Pfleeger, *Principles of Survey Research, Parts 1 to 6.* Software Engineering Notes, 2001-2002.
31. Yin, R.K., *Case Study Research: Design and Methods*. 3rd ed. Vol. 5. 2002: Sage Publications.

Appendix A. Survey Questionnaire

An Investigation into Architecture Design Rationale

(A Joint Research Project by Swinburne University of Technology and NICTA)

Q1. I agree to participate in this activity, realising that I may withdraw at any time. I agree that the research data collected for this study may be published or used by the investigators for research purposes.

Q2. I would like to receive a copy of the research report when it becomes available.

Q3. As a designer/architect, the following are my job's primary tasks. (Tick any task if you spend at least 10% of your time on that task in a project)

a) Project Management Tasks
b) IT Planning or Proposal Preparation
c) Requirements Analysis or Tender Analysis
d) Overall Design of System
e) Software Design and Specification
f) Data Modelling
g) Program Design and Specification
h) Test Planning and Design
i) Design of Non-functional Requirements (security, performance, interoperability, flexibility, standards, usability etc.)
j) Implementation Design (capacity planning, system environment, platforms etc.)
k) Training

Q4. The role of software architect is formally recognised in my organisation for: (Please tick the appropriate choice)

a) All projects across the organisation
b) Some projects only
c) Not at all

Q5. If your answer to the last question is for **some projects only**, the criteria that dictate whether the project needs an architect are: (Please tick all appropriate choices)

a) New systems
b) Mission or business critical systems
c) Systems which are considered high-risk
d) Systems which are over certain budget
e) Other criteria, please specify: ______________(text 256 char)

Q6. The organisation that I work with carries out software architecture reviews by architects external to the project for : (Please tick the appropriate choice)

a) All projects across the organisation
b) Some projects only
c) Not at all

Q7. If your answer to the last question is for **some projects only**, the criteria that dictate whether the project requires external architect review are: (Please tick all appropriate choices)

a) New systems
b) Mission or business critical systems
c) Systems which are considered high-risk
d) Systems which are over certain budget
e) Other criteria, please specify: _____________(text 256 char)

Q8. I consider the appropriateness of alternative architecture designs during the design process before I make a decision (Note: an alternative design is a design that you have considered.): (Frequency of occurrence)

Q9. I document discarded alternative designs : (Frequency of occurrence)

Q10. When making architecture design decisions, the importance of each of the following design rationales play in my decision making process is : (Note: design rationales are reasons to justify the design.) (Level of Importance)

a) Design constraints
b) Design assumptions
c) Weakness of a design
d) Cost of a design
e) Benefit of a design
f) Complexity of a design
g) Am I certain that this design would work
h) Am I certain that I or the team could implement it
i) Tradeoffs between design alternatives

Q11. This is an optional question. The other design rationales I also consider but are not listed above are ________________________ text(256)

Q12. I use the following design rationales to reason about my architecture design: (Frequency of occurrence)

a) Design constraints
b) Design Assumptions
c) Weakness of design
d) Cost of design
e) Benefit of design
f) Complexity of design
g) Certainty that design would work
h) Certainty that you could implement it
i) Tradeoffs between alternatives

Q13. I document these types of architecture design rationales: (Frequency of occurrence)

a) Design constraints
b) Design Assumptions
c) Weakness of design
d) Cost of design
e) Benefit of design
f) Complexity of design
g) Certainty that design would work
h) Certainty that you could implement it
i) Tradeoffs between alternatives

Q14. On an overall scale, the level of documentation of architecture design rationales that I do is: (Level of Documentation)

Q15. If I do not document architecture design rationale, the reasons are as follows. (Please tick applicable reasons.)

a) There are no standards or requirements in the project or organisation to do so
b) I am not aware of the need to document it
c) Documenting design rationale is not useful
d) Time / budget constraints
e) Absence of appropriate tools for documenting design rationale
f) Other reasons, please specify : _______(text 256 chars)

Q16. This is an optional question. If and when I document architecture design rationale, these are the tools, procedures or methods that I use ___________________________(text 256 chars)

Q17. When I design a system or software architecture, I reason about why I make certain design choices. (Frequency of occurrence)

Q18. I think it is important to use design rationales to justify design choices.. (Level of agreement)

Q19. In my education or my professional training, I have been trained to make use of design rationales explicitly to justify design choices. (Level of Training)

Q20. I sometimes design architectures to enhance existing systems. Yes / No (if No, go to question 28)

Q21. I revisit architecture design documents and design specifications to help me understand the design of the system for making system enhancements. (Frequency of occurrence)

Q22. Design rationales of existing systems are important to help me understand previous designs and assess my options in system enhancements and integration. (Level of Agreement)

Q23. I forget the reasons that justify my designs after a period of time. (Level of forgetfulness)

Q24. If I am not the designer of an architecture, I may not know why existing designs are created in a certain way without documented design rationale or someone who can explain the design. (Level of Agreement)

Q25. I do architectural impact analysis during system enhancements and integration to assess how new changes might affect the existing system. (Note: architectural impact analysis is used to analyse the extent and impact of changes to the structure of the system.) (Frequency of occurrence)

Q26. The following items are important when I carry out architectural impact analysis. (Level of Agreement)

a) Analyse and Trace Requirements
b) Analyse Specifications of Previous Design
c) Analyse Design Rationale of Previous Design
d) Analyse Feasibility of Implementation
e) Analyse Violation of Constraints or Assumptions of Previous Design
f) Analyse Scenarios
g) Analyse Cost of Implementation
h) Analyse Risk of Implementation

Q27. This is an optional input. Other additional steps that I will take when carrying out impact analysis are ________________________ text(256)

Q28. When I design, I am relatively certain (i.e. I consider the risk factor) that the resulting design will work and I or my team are capable of implementing it. (Level of Certainty)

Q29. I explicitly quantify the risk of implementation when I design. (Frequency of occurrence)

Q30. Different potential architecture designs have different degrees of uncertainty, or risk, to achieve the desired business outcomes. (Level of Agreement)

Q31. There might be different degrees of uncertainty, or risk, in implementation depending on the capability of the design/development team. (Note: The capability refers to the experience and knowledge of particular technology used in the implementation.) (Level of Agreement)

Q32. The level of risk of an architecture design¸ i.e. before detailed design and implementation, that I consider acceptable for an important project is: (Level of Risk Scale)

Q33. My E-mail address: ___________________________(text 128)

Q34. Sex: Male (M) / Female (F)

Q35. City of residence: _____________________________(text 40)

Q36. No of years I have been in the IT industry: XX

Q37. No of years I have been a designer / architect: XX

Q38. No of years I have been with the current (or last) organisation: XX

Q39. My current (or last) job title : _____________________________ (text 40)

Q40. I have used at least one Software Development Methodology or Standard in a project in the past. Yes / No.

Q41. The number of co-workers in my current or last project team, including project managers, architects, designers, programmers and testers, is: XXXX

Q42. I have obtained a tertiary qualification in an IT related field such as Computer Science, Information Technology etc. Yes / No

Appendix B. A Summary of Survey Results

B.1. Response Rate

- There are 171 direct invitations and an estimated 376 indirect invitations. A total of 547 people have received the invitation.
- There are 127 responses, the breakdown of the responses are
  - 81 valid responses
  - 5 respondents have under the required minimum experience
  - 41 incomplete responses

B.2. Characteristics of Respondents

- Q34 – Gender – 80.2% male / 19.8% female
- Q35 – Area of residence is coded into:
  - 1) Australia / New Zealand – 67.9%
  - 2) Asia – 28.4%
  - 3) Other – 0%
  - 4) Unknown – 3.7%
- Q36 - number of years in the IT industry:
  - Minimum 4 yrs
  - Maximum 37 yrs
  - Mean 17.12 yrs
  - Median 15 yrs
- Q37 – No of years as a designer / architect:
  - Minimum 1 yrs
  - Maximum 35 yrs
  - Mean 9.75 yrs
  - Median 8 yrs
- Q38 – No of years I have been with the current or last organisation:
  - Minimum 1 yrs
  - Maximum 37 yrs
  - Mean 7.65 yrs
  - Median 6 yrs
- Q39 - Job Title Listings (Distinct Titles)
  - System Engineer
  - Professor
  - Senior Computer Officer
  - Systems Architect
  - Region Consulting Partner
  - Principal Consultant
  - PS Manager
  - Chief Engineer
  - Project manager
  - Business Consultant
  - Director
  - Technical Chief
  - Web Developer
  - Web Applications Developer
  - Industry Expert
  - Engineer
  - Technical Consultant
  - Technical Operations Manager
  - Consultant
  - Solution Architect
  - Analyst Programmer

- Researcher
- Senior Analyst Programmer
- Research Developer
- Senior engineer
- IT Architect
- Senior IT Architect
- Systems & Network Architect
- Architect
- Technical Solution Architect
- Security Architect
- Oracle DBA
- IT Postdoctoral Researcher
- Systems Architect
- Consulting IT Architect
- Application architect
- Associate IT Architect
- Principal Programme Director
- IT Manager
- Manager (Computer Projects)
- Senior Research Scientist
- MIS Manager
- Architect Advisor
- Software Development Consultant
- Professional Services Consultant

- Q40 – Use of at least one software development methodology
  - Yes – 95.1 %
  - No – 4.9%
- Q41 – Number of co-workers in current or last project team:
  - Minimum 3 people
  - Maximum 200 people
  - Mean 25 people
  - Median 15 people
- Q42 – Tertiary education in an IT related field
  - Yes – 85.2%
  - No – 14.8%

B.3. Roles Designers / Architects Play

Q3. The primary tasks of designer / architect

The following are the percentages of designers who perform the tasks:

- Project Management – 50.6%
- IT Planning or proposal preparation – 49.4%
- Requirements Analysis or tender analysis – 81.5%
- Overall design of system – 86.4%
- Software design and specification – 58%
- Data Modelling – 44.4%
- Program design and specification – 35.8%
- Test panning and design – 29.6%
- Design of Non-functional requirements – 64.2%
- Implementation Design – 42%
- Training – 19.8%

The following is the distribution of designers / architects who perform between one and eleven tasks.

Recode - no of job responsibilities

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.2	1.2
	2	5	6.2	6.2	7.4
	3	4	4.9	4.9	12.3
	4	13	16.0	16.0	28.4
	5	14	17.3	17.3	45.7
	6	17	21.0	21.0	66.7
	7	15	18.5	18.5	85.2
	8	8	9.9	9.9	95.1
	9	2	2.5	2.5	97.5
	10	1	1.2	1.2	98.8
	11	1	1.2	1.2	100.0
	Total	81	100.0	100.0	

- We find no relationship using Spearman's rho between no of job responsibilities and no of years in the IT industry and no relationship between number of years as designer and job relationship

B.4. Architecture Evaluation

Q4. About the role of SA

- All projects use SA – 43.2%
- Some projects use SA – 48.1%
- Not at all – 8.6%

Q5. For some projects, criteria for using SA are

- New system – 23.55%
- Mission critical – 35.8%
- High Risk – 27.2%
- High cost – 18.5%

Q5e. Other criteria includes

- System that has many independent modules and the integration of them to make the system operable is critical.
- Depending on complexity/budget of projects, sometimes several roles like PM/Architect etc. are clubbed into one.
- by type and complexity of project
- Basically any piece of work initiated by the Business. This includes all of the above choices.
- This role is just part of the job, but not a specific position.
- Changes that impact other systems. Or Other systems impacting architects areas of responsibility.
- Systems which have a wide impact across many parts of the organisation.

Q6. Projects that are reviewed by SA externally

- all projects – 16%
- some projects – 49.4%
- not at all – 34.6%

Q7. If only some projects are reviewed externally, criteria of reviews are

- 17.3% new system
- 27.2% mission critical
- 27.2% high risk
- 21% high budget

Q7. Other criteria for external reviews

- For projects which have framework changes or additions which would impact other projects in the future.
- Time permitting
- Systems which are large and complex
- Systems which are paid by govement

ng
reviews are
y involve
es to other
e performed.

Q8 Freq Alt Solutions

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	1.2	1.2	1.2
	3	15	18.5	18.5	19.8
	4	31	38.3	38.3	58.0
	5	34	42.0	42.0	100.0
	Total	81	100.0	100.0	

B.5. Architecture Rationale - Perception, Use and Documentation

Q8. Consider appropriateness of alternative architecture designs

Q8 Freq Alt Solutions

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	1.2	1.2	1.2
	3	15	18.5	18.5	19.8
	4	31	38.3	38.3	58.0
	5	34	42.0	42.0	100.0
	Total	81	100.0	100.0	

Q9. Document discarded alternative designs

Q9 Freq Document Alt Design

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	11	13.6	13.6	13.6
	2	18	22.2	22.2	35.8
	3	17	21.0	21.0	56.8
	4	19	23.5	23.5	80.2
	5	16	19.8	19.8	100.0
	Total	81	100.0	100.0	

Q10. Level of importance of 9 different design rationales when making architecture decisions

Q10a LI - constraint

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	1.2	1.2	1.2
	3	9	11.1	11.1	12.3
	4	31	38.3	38.3	50.6
	5	40	49.4	49.4	100.0
	Total	81	100.0	100.0	

Q10b LI - assumption

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	3.7	3.7	3.7
	2	6	7.4	7.4	11.1
	3	12	14.8	14.8	25.9
	4	36	44.4	44.4	70.4
	5	24	29.6	29.6	100.0
	Total	81	100.0	100.0	

Q10c LI - weakness

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	2.5	2.5	2.5
	2	6	7.4	7.4	9.9
	3	23	28.4	28.4	38.3
	4	35	43.2	43.2	81.5
	5	15	18.5	18.5	100.0
	Total	81	100.0	100.0	

Q10d LI - cost

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	6	7.4	7.4	7.4
	3	12	14.8	14.8	22.2
	4	35	43.2	43.2	65.4
	5	28	34.6	34.6	100.0
	Total	81	100.0	100.0	

Q10e LI - benefit

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.2	1.2
	2	1	1.2	1.2	2.5
	3	6	7.4	7.4	9.9
	4	44	54.3	54.3	64.2
	5	29	35.8	35.8	100.0
	Total	81	100.0	100.0	

Q10f LI - complexity

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	2	2.5	2.5	2.5
	3	21	25.9	25.9	28.4
	4	38	46.9	46.9	75.3
	5	20	24.7	24.7	100.0
	Total	81	100.0	100.0	

Q10g LI - certainty would work

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	3	3.7	3.7	3.7
	3	9	11.1	11.1	14.8
	4	24	29.6	29.6	44.4
	5	45	55.6	55.6	100.0
	Total	81	100.0	100.0	

Q10h LI - certainty implementable

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	2.5	2.5	2.5
	2	4	4.9	4.9	7.4
	3	13	16.0	16.0	23.5
	4	26	32.1	32.1	55.6
	5	36	44.4	44.4	100.0
	Total	81	100.0	100.0	

Q10i LI - tradeoffs

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	4	4.9	4.9	4.9
	3	25	30.9	30.9	35.8
	4	36	44.4	44.4	80.2
	5	16	19.8	19.8	100.0
	Total	81	100.0	100.0	

Q11. Other design rationales that are considered but not listed are (comments by respondents):

- Ease of post-adoption/implementation maintenance including system upgrades vis-a-vis availability of new technology in all components (hardware, software, telco, security, etc)

- Is the design applying standard pattern(s)Proportion of configuration vs customization vs buildNumber of technologies employedNumber of vendors employedPreferred product and tool setsNumber of new vs existing technologiesLevel of reuseFit with existing systems/technologiesAbility to iteratively design, implement and deploy
- Risk, time-to-market, enterprise technology policy, divisional/domain strategy
- Time availability ; 80/20 rule ; some designs are already pre- defined within the architecture and cannot be easily altered
- Related to the opposite of complexity but not exactly so: Design Elegance.
- Design is something that happens during the coding and yet all the questions so far indicate a bias towards doing design before coding. This is not the correct way to write software. I also think that the terms used in this are not clear - they should be explained.
- Overall compatibility with other systems in hetrogeneous organization rates a 4 on the above scale
- Refactoring of existing systems
- Reliability -- will the solution have a reasonable mean time to failure.Avialbility -- can it meet the customer avialability requirments ie 24X7 or never down for more than 30 minutesServiceability -- can it be maintained, upgraded, and defects isolated and resolved easily are their tools to assist this built into the design
- Major reason for architecture decisions is the understanding of the clients business motivations.
- 'Design Constraints' is a vague broad category within it I would use:Fit of design to business requirements, both functional and non functional;Performance requirements;Ongoing maintainance;
- adherence to requirements (functional & non-functionsl), Risk
- testability of the solution, maintainability/longevity, reuse and buy vs. build decisions
- Has this design been used before and was it successful. Does the design follow a standard or repeatable approach.
- Security policy compliance
- Availability of tools to create the design.Any requirements to integrate with existing systems.
- Performance of design (latency)
- Performance
- Will the organisation accept the design.
- Flexibility in accommodating future changes
- The expected lifetime of the software to be used
- Meeting the criteria of the requirements in a user friendly way. The deadline of the project.The skills of the project team.
- Alignment to the organisations technology direction, current IT architecture and IT capability.
- Standard operating environmentsIndustry, de facto and emerging standardsCurrent Expertise of teams providing detailed design and implementation
- Availability of prototype or staged delivery of a design
- As I am working in far future (2012+) I look at whether the technology will be available for the design.
- organisational politics - preference for SOE & adopted methods, internal & external standards, never use release 1.0, always search for prior intellectual capital (reuse opportunity)
- management preference/IT strategy and directions, viability of vendor providing solution, vendor relationship, existing skills and experience of staff, speed to market, lower risk options, novelness of technology components, Gartner/analyst recommendations

Q12. What reasons architects use to rationalise architecture design

Q12a FO - constraint

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	10	12.3	12.3	12.3
	4	34	42.0	42.0	54.3
	5	37	45.7	45.7	100.0
	Total	81	100.0	100.0	

Q12b FO - assumptions

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	2.5	2.5	2.5
	2	2	2.5	2.5	4.9
	3	25	30.9	30.9	35.8
	4	27	33.3	33.3	69.1
	5	25	30.9	30.9	100.0
	Total	81	100.0	100.0	

Q12c FO - weakness

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.2	1.2
	2	7	8.6	8.6	9.9
	3	28	34.6	34.6	44.4
	4	30	37.0	37.0	81.5
	5	15	18.5	18.5	100.0
	Total	81	100.0	100.0	

Q12d FO - cost

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.2	1.2
	2	8	9.9	9.9	11.1
	3	16	19.8	19.8	30.9
	4	31	38.3	38.3	69.1
	5	25	30.9	30.9	100.0
	Total	81	100.0	100.0	

Q12e FO - benefit

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.2	1.2
	2	1	1.2	1.2	2.5
	3	10	12.3	12.3	14.8
	4	40	49.4	49.4	64.2
	5	29	35.8	35.8	100.0
	Total	81	100.0	100.0	

Q12f FO - complexity

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	2	2.5	2.5	2.5
	3	22	27.2	27.2	29.6
	4	28	34.6	34.6	64.2
	5	29	35.8	35.8	100.0
	Total	81	100.0	100.0	

Q12g FO - certainty would work

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	2.5	2.5	2.5
	2	1	1.2	1.2	3.7
	3	9	11.1	11.1	14.8
	4	26	32.1	32.1	46.9
	5	43	53.1	53.1	100.0
	Total	81	100.0	100.0	

Q12h FO - certainty implementable

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	3.7	3.7	3.7
	2	3	3.7	3.7	7.4
	3	13	16.0	16.0	23.5
	4	27	33.3	33.3	56.8
	5	35	43.2	43.2	100.0
	Total	81	100.0	100.0	

Q12i FO - tradeoffs

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	5	6.2	6.2	6.2
	3	24	29.6	29.6	35.8
	4	34	42.0	42.0	77.8
	5	18	22.2	22.2	100.0
	Total	81	100.0	100.0	

- comparing the list in Q12 to the importance list in Q10, we get a correlation for each rationale to test consistency of responses using Spearman Rank Order Correlation (rho) in a 2-tailed test. The results are summarised below.
  - Constraint – [$r(81) = +.539$, $p<0.001$]
  - Assumption - [$r(81) = +.592$, $p<0.001$]
  - Weakness of design - [$r(81) = +.553$, $p<0.001$]
  - Cost of design - [$r(81) = +.616$, $p<0.0005$]
  - Benefit of design - [$r(81) = +.596$, $p<0.001$]
  - Complexity of design - [$r(81) = +.458$, $p<0.001$]
  - Certainty design would work - [$r(81) = +.715$, $p<0.001$]
  - Certainty design Implementable - [$r(81) = +.78$, $p<0.001$]
  - Tradeoffs between alternatives - [$r(81) = +.528$, $p<0.001$]
- We test the correlation between using tradeoffs and no of years in IT industry
  - [$r(81) = +.099$, $p<0.379$]
- We test the correlation between using tradeoffs and no of years doing design and found that there is a weak correlation
  - [$r(81) = +.239$, $p<0.05$]

Correlations

Correlations

			Q12i FO - tradeoffs	Q37 - Years as Designer
Spearman's rho	Q12i FO - tradeoffs	Correlation Coefficient	1.000	.239*
		Sig. (2-tailed)	.	.033
		N	81	80
	Q37 - Years as Designer	Correlation Coefficient	.239*	1.000
		Sig. (2-tailed)	.033	.
		N	80	80

*. Correlation is significant at the 0.05 level (2-tailed).

' - Years)esigner
.239* .033 80
1.000 . 80

Q13. Level of documentation of each design rationale

Q13a FO - constraints

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.2	1.2
	2	2	2.5	2.5	3.7
	3	11	13.6	13.6	17.3
	4	16	19.8	19.8	37.0
	5	51	63.0	63.0	100.0
	Total	81	100.0	100.0	

Q13b FO - assumption

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	3.7	3.7	3.7
	2	3	3.7	3.7	7.4
	3	11	13.6	13.6	21.0
	4	21	25.9	25.9	46.9
	5	43	53.1	53.1	100.0
	Total	81	100.0	100.0	

Q13c FO - weakness

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	3.7	3.7	3.7
	2	19	23.5	23.5	27.2
	3	30	37.0	37.0	64.2
	4	12	14.8	14.8	79.0
	5	17	21.0	21.0	100.0
	Total	81	100.0	100.0	

Q13d FO - cost

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	6	7.4	7.4	7.4
	2	13	16.0	16.0	23.5
	3	25	30.9	30.9	54.3
	4	17	21.0	21.0	75.3
	5	20	24.7	24.7	100.0
	Total	81	100.0	100.0	

Q13e FO - benefit

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	2.5	2.5	2.5
	2	8	9.9	9.9	12.3
	3	15	18.5	18.5	30.9
	4	26	32.1	32.1	63.0
	5	30	37.0	37.0	100.0
	Total	81	100.0	100.0	

Q13f FO - complexity

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	3.7	3.7	3.7
	2	8	9.9	9.9	13.6
	3	29	35.8	35.8	49.4
	4	25	30.9	30.9	80.2
	5	16	19.8	19.8	100.0
	Total	81	100.0	100.0	

Q13g FO - certainty would work

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	15	18.5	18.5	18.5
	2	12	14.8	14.8	33.3
	3	16	19.8	19.8	53.1
	4	20	24.7	24.7	77.8
	5	18	22.2	22.2	100.0
	Total	81	100.0	100.0	

Q13h FO - certainty implementable

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	15	18.5	18.5	18.5
	2	14	17.3	17.3	35.8
	3	20	24.7	24.7	60.5
	4	18	22.2	22.2	82.7
	5	14	17.3	17.3	100.0
	Total	81	100.0	100.0	

Q13i FO - tradeoffs

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	5	6.2	6.2	6.2
	2	15	18.5	18.5	24.7
	3	21	25.9	25.9	50.6
	4	26	32.1	32.1	82.7
	5	14	17.3	17.3	100.0
	Total	81	100.0	100.0	

Q14. Overall scale of documentation

Overall Documentation of DR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	10	12.3	12.3	12.3
	3	20	24.7	24.7	37.0
	4	44	54.3	54.3	91.4
	5	7	8.6	8.6	100.0
	Total	81	100.0	100.0	

Q15. If I don't document, the reasons are

Q15a - No Stds

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No Standards	34	42.0	100.0	100.0
Missing	System	47	58.0		
Total		81	100.0		

Q15b - Not aware

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not aware	4	4.9	100.0	100.0
Missing	System	77	95.1		
Total		81	100.0		

Q15c - Not useful

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not useful	8	9.9	100.0	100.0
Missing	System	73	90.1		
Total		81	100.0		

Q15d - Time/budget

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Time/budget	49	60.5	100.0	100.0
Missing	System	32	39.5		
Total		81	100.0		

Q15e - No Tool

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No Tool	24	29.6	100.0	100.0
Missing	System	57	70.4		
Total		81	100.0		

- Other reasons stated by respondents
  - Lack of formal review/justification by external (outside project) processes
  - I suggest it is not worthy to spend much time on it, because we always choose appropriateness, not a perfect solution. Perfect solution cost more time and cost, is it?
  - Not complex
  - Do not want to get into a design-review loop that never ends.
  - The impact of the solution is low.
  - Only required to document the final design. Technology is evolving quickly, a better architecture solution will soon be available.
  - I will only not document the design rationale if the design decision is of little or no impact to the solution.
  - They are not required for the higher level decision making process.

Q16. Tools & methods used in documentation

- WordVisioStructure diagramsExcel models of complexity level
- Defined and managed via templatised document. Entered into a managed register. Socialisation and review within design team and project management. Ratified via project design authority. Presented as key collateral to enterprise architecture review.
- Not all of them can be found. However, I sometime change my idea according to the tools...
- The portion of design rationales that I document is based its presence in the design document template that we use.
- WordVisioEnterprise Architect
- These seem like strange questions. The code is the documentation of the architecture and design of a piece of software. How is it possible to develop software without documenting the design if the code is your documentation?
- Model MakerUML
- Guided by the IBM Method that has been tailored for the engagement so that the appropriate work products are produced.
- IBM internal design method (GSMETHOD)
- Word (sometimes with prescribed template)
- No tools used. Results recorded in a word document.
- IBM GS Methodology
- Architectire Decision Document
- Design Template in Microsoft Word.
- They are documented as architectural decisions following a formal method and notation
- UML, Text description
- Usually define the architecture choices at a high level and the reasons for discarding those that were not selected. Also, capture why the preferred design was chosen.
- internally developed tools
- Generally, I use a cost-benefit analysis, ie what are the expected deliverables and their benefits, and what is the expected cost.
- Word Processor
- IBM GSMethods
- Word or Rational as part of the solution.
- Use Architectural Decisions template provided by Global Services Method.
- QMS Design Template docuement. Developer and Manager Reviews.
- A methodology. Requirements Traceability Matrix.
- If I cannot capture it in the architecture tool (CORE) it is not captured.
- company method & rpcoedures(aligned with CMMI 5) & client method & procedures(if relevant); tools - Word, Excel
- Microsoft word and powerpoint slides, especially powerpoint, so to better facilitate real communication and socialisation of architecture principle and designs.

Q17. Reason about why I make decision (level of frequency)

Q17 FO - Reason about choices

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	1.2	1.2	1.2
	3	8	9.9	9.9	11.1
	4	34	42.0	42.0	53.1
	5	38	46.9	46.9	100.0
	Total	81	100.0	100.0	

Q18. Important to justify (level of agreement)

Q18 LA - Importance of DR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	1.2	1.2	1.2
	3	11	13.6	13.6	14.8
	4	30	37.0	37.0	51.9
	5	39	48.1	48.1	100.0
	Total	81	100.0	100.0	

Q19. Level of training

Q19 Level of Training

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	10	12.3	12.3	12.3
	2	16	19.8	19.8	32.1
	3	21	25.9	25.9	58.0
	4	18	22.2	22.2	80.2
	5	16	19.8	19.8	100.0
	Total	81	100.0	100.0	

B.6. Architecture Rationale, System Enhancements and Impact Analysis

Q20. System enhancements

Q20 Design for Enhancement

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Yes	69	85.2	85.2	85.2
	No	12	14.8	14.8	100.0
	Total	81	100.0	100.0	

Q21. Revisit architecture design documents

Q21 FO - Revisit doc to understand

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	8	9.9	11.9	11.9
	3	9	11.1	13.4	25.4
	4	20	24.7	29.9	55.2
	5	30	37.0	44.8	100.0
	Total	67	82.7	100.0	
Missing	NA	14	17.3		
Total		81	100.0		

Q22. DR helps understand previous design to assess options

Q22 LA - DR Help Understanding

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	3	3.7	4.5	4.5
	3	10	12.3	14.9	19.4
	4	21	25.9	31.3	50.7
	5	33	40.7	49.3	100.0
	Total	67	82.7	100.0	
Missing	NA	14	17.3		
Total		81	100.0		

Q23. Level of forgetfulness after a while

Q23 LF - Forget Reasons

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	2	2.5	3.0	3.0
	2	15	18.5	22.7	25.8
	3	22	27.2	33.3	59.1
	4	22	27.2	33.3	92.4
	5	5	6.2	7.6	100.0
	Total	66	81.5	100.0	
Missing	NA	15	18.5		
Total		81	100.0		

Q24. May not know why existing design is why it is (Level of agreement)

Q24 LA - Not knowing without DR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	2	3	3.7	4.6	6.2
	3	9	11.1	13.8	20.0
	4	23	28.4	35.4	55.4
	5	29	35.8	44.6	100.0
	Total	65	80.2	100.0	
Missing	NA	16	19.8		
Total		81	100.0		

Q25. Do impact analysis to assess how changes may affect existing system

Q25 DO - Do Impact Analysis

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	4	4.9	6.2	6.2
	3	13	16.0	20.0	26.2
	4	29	35.8	44.6	70.8
	5	19	23.5	29.2	100.0
	Total	65	80.2	100.0	
Missing	NA	16	19.8		
Total		81	100.0		

Q26. Level of importance in impact analysis

Q26a LA - Req

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	3	7	8.6	10.3	11.8
	4	27	33.3	39.7	51.5
	5	33	40.7	48.5	100.0
	Total	68	84.0	100.0	
Missing	System	13	16.0		
Total		81	100.0		

Q26b LA - Previous Spec

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	2	7	8.6	10.3	11.8
	3	13	16.0	19.1	30.9
	4	24	29.6	35.3	66.2
	5	23	28.4	33.8	100.0
	Total	68	84.0	100.0	
Missing	System	13	16.0		
Total		81	100.0		

Q26c LA - DR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	2	8	9.9	11.9	13.4
	3	22	27.2	32.8	46.3
	4	22	27.2	32.8	79.1
	5	14	17.3	20.9	100.0
	Total	67	82.7	100.0	
Missing	System	14	17.3		
Total		81	100.0		

Q26d LA - Feasibility

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	2	1	1.2	1.5	2.9
	3	6	7.4	8.8	11.8
	4	28	34.6	41.2	52.9
	5	32	39.5	47.1	100.0
	Total	68	84.0	100.0	
Missing	System	13	16.0		
Total		81	100.0		

Q26e LA - Assumptions

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	2	6	7.4	8.8	10.3
	3	17	21.0	25.0	35.3
	4	28	34.6	41.2	76.5
	5	16	19.8	23.5	100.0
	Total	68	84.0	100.0	
Missing	System	13	16.0		
Total		81	100.0		

Q26f LA - Scenarios

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	1	1.2	1.5	1.5
	2	5	6.2	7.6	9.1
	3	11	13.6	16.7	25.8
	4	24	29.6	36.4	62.1
	5	25	30.9	37.9	100.0
	Total	66	81.5	100.0	
Missing	System	15	18.5		
Total		81	100.0		

Q26g LA - Cost of Impl

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	7	8.6	10.3	10.3
	3	9	11.1	13.2	23.5
	4	21	25.9	30.9	54.4
	5	31	38.3	45.6	100.0
	Total	68	84.0	100.0	
Missing	System	13	16.0		
Total		81	100.0		

Q26h LA - Risk of Impl

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	3	3.7	4.4	4.4
	3	4	4.9	5.9	10.3
	4	22	27.2	32.4	42.6
	5	39	48.1	57.4	100.0
	Total	68	84.0	100.0	
Missing	System	13	16.0		
Total		81	100.0		

Q27. Additional impact analysis steps are:

- Changes to business processes or organisational changes.
- analyse performance impact of altering or adding addional load / transactions
- Talking to the client, understanding other systems (computer and manual) they may have and possible impact (direct and indirect) and relating these back to the functional and non functional requirements and using this information to look for gaps. Also an architect needs to be involved in the whole client change management process along with the business analyst. Managing client expectations allows you to manage scope which helps in managing risk. It is no use building what the client asked for if won't solve the actual problem.
- rerun tests or examine test results for the previous system.Talk to current users of the system
- What new skills will be required in the team to implement the design.
- Performance
- Any ISO documents to complete.
- Working with the current maintainers/operators of the system, as design documents are not always upto date and operational issues aren't always clearly documented.
- Ensure full regression testing is possible before beginning.
- Comment: As I have often experinced that documentation is not updated in step with system changes I tend to distrust it - thus my lack of enthusiasm for past rationales etc.
- client context, needs & expectations - need to understand these to frame the effort (particularly as usually working under cost & time constraints)

B.7. Using Risks to Rationalise Architecture Design

Q28. Level of certainly in implementing the design

Q28 LC - Certainty design work

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	2	1	1.2	1.2	1.2
	3	6	7.4	7.4	8.6
	4	34	42.0	42.0	50.6
	5	40	49.4	49.4	100.0
	Total	81	100.0	100.0	

Q29. Explicitly quantify the risk of implementation

Q29 FO - Quantify Risk of Impl

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	5	6.2	6.2	6.2
	2	6	7.4	7.4	13.6
	3	21	25.9	25.9	39.5
	4	29	35.8	35.8	75.3
	5	20	24.7	24.7	100.0
	Total	81	100.0	100.0	

Q30. Different designs have different risk level in achieving business outcomes

Q30 LA - Diff Arch dff risk

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	1	1.2	1.2	1.2
	4	34	42.0	42.0	43.2
	5	46	56.8	56.8	100.0
	Total	81	100.0	100.0	

Q31. Different degree of risk depending on the capability of team

Q31 LA - Diff risk bc of Team

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	4	4.9	4.9	4.9
	4	35	43.2	43.2	48.1
	5	42	51.9	51.9	100.0
	Total	81	100.0	100.0	

Q32. Level of acceptable risk in architecture design

Q32 - Acceptable Risk

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	3	3.7	3.7	3.7
	2	9	11.1	11.1	14.8
	3	19	23.5	23.5	38.3
	4	15	18.5	18.5	56.8
	5	4	4.9	4.9	61.7
	6	3	3.7	3.7	65.4
	7	12	14.8	14.8	80.2
	8	11	13.6	13.6	93.8
	9	4	4.9	4.9	98.8
	10	1	1.2	1.2	100.0
	Total	81	100.0	100.0	