

**AN EXPLORATION OF THE APPLICATION OF
SOFTWARE ARCHITECTURE EVALUATION TECHNIQUES
TO THE DOMAIN OF SERVICE DESIGN**

SIMON FIELD

A submission presented in partial fulfilment of the
requirements of the University of South Wales/Prifysgol De Cymru
for the degree of Doctor of Business Administration

This research programme was carried out in collaboration with
the UK Border Agency (now UK Visas and Immigration)
with the financial support of Public Service Management Wales
and the Office for National Statistics

October 2016

Abstract

This thesis presents an exploration of the potential application of software architecture evaluation techniques to the domain of service design. The domain of software architecture and the evolution of software architecture evaluation techniques are explored, and compared with the emerging domain of service design. Similarities between the two domains lead to the notion of 'Service Architecture', and a novel method for conducting ex-ante evaluations of competing service designs, the Service Architecture Review Method is presented. It is derived from Architecture Trade-off Analysis Method from the Software Engineering Institute, with a quality model that has been adapted to describe service quality, and incorporating a stakeholder model to cater for the varied stakeholder perspectives often involved in services. A software tool to support the participants in the method's evaluation workshops is described.

A case study, representing the first use of this proposed method in a service design project at the UK Border Agency, is presented. Participants in the method's workshops were subsequently interviewed, and learnings from the case study are presented and discussed. The experience of the case study led to some improvements to the method, which is described in the form of a 'User Guide' in this thesis, as one of this research project's significant contributions to practice. The thesis presents a number of contributions to theory in addition to the above-mentioned concept of 'Service Architecture'. These include the use of a service quality model and a stakeholder model in the evaluation method, and the use of this resulting method for evaluation in a service design project, and as an enhanced method for evaluating competing software architectures.

Future research avenues are proposed, addressing some of the limitations of the research presented in this thesis related to the wider applicability of the method beyond the case study presented here, and possible further refinement of the method itself. Potential to transfer other learnings from the domain of software architecture to service design is also discussed and the thesis concludes with a discussion of the significance of these contributions to the maturing domain of service design.

Table of Contents

Chapter 1: Introduction.....	9
1.1 Introduction.....	10
1.2 Mass collaboration and service design.....	13
1.3 The research issue to be addressed.....	14
1.4 Aim and objectives.....	15
1.5 Accelerated learning.....	17
1.6 Thesis overview.....	19
Chapter 2: Background and Literature.....	22
2.1 Introduction.....	23
2.2 Software architecture and architecture evaluation.....	24
2.3 The role of the architecture.....	28
2.4 Software quality.....	31
2.5 Software evaluation.....	34
2.6 Is it really architecture?.....	40
2.7 What are services?.....	43
2.8 Service Design.....	46
2.9 Service evaluation.....	52
2.10 Understanding stakeholders.....	53
2.11 Service quality model.....	57
2.12 Service Architecture and design reviews.....	64
Chapter 3: Approaches and Methods.....	66
3.1 Introduction.....	67
3.2 Developing the theory.....	69
3.3 Exploring the domain of software architecture.....	71
3.4 Understanding service design.....	72
3.5 Why software?.....	74
3.6 Discussion of the approach and methods.....	76
3.7 Adopting a Case Study approach.....	77
3.8 A first use of SARM.....	79
3.9 Maximising the research value.....	80
3.10 Adopting a single case design.....	82
3.11 Selection criteria for the case study.....	83
3.12 Choosing the case study.....	84
3.13 Collecting evidence at other levels.....	86
3.14 Analysing the evidence.....	87
3.15 Studying design decisions.....	88
3.16 Ethical issues.....	89
3.17 Other potential issues.....	91
3.18 Conclusions.....	93

Chapter 4: Creation and Development of the Service Architecture Review Method (SARM).....	94
4.1 Introduction.....	95
4.2 Adopting a source method.....	95
4.3 Adding a quality model.....	97
4.4 Adding stakeholder perspectives.....	99
4.5 Developing a software tool.....	101
4.6 Subsequent enhancements.....	102
4.7 Adding 'Stakeholder Views'.....	103
4.8 Inverting the likelihood.....	105
4.9 Separating risk input from display.....	106
4.10 Adding the 'Charts' worksheets.....	107
4.11 Adopting a stakeholder model.....	109
4.12 Future enhancements and changes.....	110
 Chapter 5: Service Architecture Review Method.....	 112
5.1 Introduction.....	113
5.2 Service Architecture Review Method.....	113
5.3 Definitions.....	114
5.4 Why should we evaluate competing designs?.....	116
5.5 Key concepts.....	118
5.6 The SARM Process – an overview.....	128
5.7 Introducing a simple service design example.....	129
5.8 Part 1 – creating the context.....	130
5.9 Using the SARM tool.....	132
5.10 Example stakeholders.....	133
5.11 Example scenarios.....	137
5.12 Creation of competing solution options.....	144
5.13 Part 2 – trade-off analysis.....	146
5.14 Quality characteristic burden.....	151
5.15 The Charts worksheet.....	153
5.16 Aggregate Risk Burden.....	154
5.17 Stakeholder burden.....	155
5.18 Reaching a conclusion.....	157
 Chapter 6: Service Design Case Study.....	 158
6.1 Introduction.....	159
6.2 Collecting evidence at the workshops.....	159
6.3 The first workshop.....	161
6.4 Understanding Stakeholder Interest.....	174
6.5 Designing Solution Options.....	175
6.6 The trade-off analysis workshop.....	177
 Chapter 7: Analysis.....	 190
7.1 Introduction.....	191
7.2 Thematic analysis and the case study primary data.....	191

7.3 Theme 1 - Stakeholder considerations.....	194
7.4 Theme 2 - Evidence-based decision making.....	199
7.5 Theme 3 - Service Architectures.....	201
7.6 Theme 4 - Service design and software development.....	204
7.7 Theme 5 - Model-based evaluation in service design.....	206
7.8 Limitations of this research and future opportunities.....	211
7.9 Adopting SARM for software architecture.....	214
7.10 SARM in other domains?.....	216
 Chapter 8: Conclusion.....	 217
8.1 Introduction.....	218
8.2 Key insights from the research - the end of one journey.....	218
8.3 Contribution to theory.....	221
8.4 Contribution to practice.....	223
8.5 Limitations and future work - the beginning of other journeys.....	225
8.6 A personal journey.....	226
8.7 A journey in time.....	227
 References.....	 229

List of Tables

Table 1 ISO 9126-1 Software Quality Model.....	32
Table 2 ONS Risk Assessment Measure of Impact.....	39
Table 3 ONS Risk Assessment Measure of Likelihood.....	39
Table 4 ONS Risk Assessment Measure of Exposure.....	39
Table 5 Servqual Dimensions and the original ten Service Quality Dimensions adapted from (Zeithaml, Parasuraman and Barry, 1990, p. 25).....	59
Table 6 Software Quality Model compared with the original dimensions of Service Quality.....	62
Table 7 The SARM Quality Model.....	122
Table 8 Risk Likelihood.....	127
Table 9 Risk Impact.....	127
Table 10 Scenario impact classification.....	139
Table 11 Inverting the Likelihood.....	151

List of Figures

Figure 1 The relationship of services to the public.....	54
Figure 2 Service Evaluation Research Matrix (Maffei, Villari and Foglieni, 2013).....	57
Figure 3 Mitchell, Agle and Wood's Stakeholder Typology, 1997.....	120
Figure 4 Risk Exposure.....	127
Figure 5 The SARM Process.....	128
Figure 6 Stakeholder Analysis Worksheet.....	132
Figure 7 Example Stakeholder Analysis worksheet (top).....	135
Figure 8 Example Stakeholder Analysis worksheet (bottom).....	135
Figure 9 Scenarios & Stakeholders worksheet.....	140
Figure 10 Completed Scenarios & Stakeholders worksheet.....	141
Figure 11 Stakeholder Views worksheet.....	142
Figure 12 Solution – Scenario Risks worksheet.....	148
Figure 13 Tradeoff Analysis worksheet (top).....	150
Figure 14 Tradeoff Analysis worksheet (bottom).....	150
Figure 15 Average Risk Score Colour Coding.....	153
Figure 16 Charts worksheet.....	154

List of Acronyms

ACM	Association for Computing Machinery
ALPSM	Architecture Level Prediction of Software Maintenance
ATAM	Architecture Trade-off Analysis Method
BCS	British Computer Society
CBAM	Cost Benefit Analysis Method
CMT	Common Measurements Tool
CORBA	Common Object Request Broker Architecture
DSDM	Dynamic Systems Development Method
GE	General Electric
IBM	International Business Machines
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IT	Information Technology
NATO	North Atlantic Treaty Organization
OHSAS	Occupational Health and Safety Assessment Series
ONS	Office for National Statistics
PERT	Program Evaluation and Review Technique
RAG	Red, Amber, Green
RATER	Reliability, Assurance, Tangibles, Empathy and Responsiveness
SAAM	Scenario-Based Architecture Analysis Method
SAEM	A Software Architecture Evaluation Method
SARM	Service Architecture Review Method
SATURN	SEI Architecture Technology User Network
SEI	Software Engineering Institute
SSME	Service Science, Management and Engineering
UKBA	United Kingdom Border Agency
UNHCR	United Nations High Commissioner for Refugees
WRC	Welsh Refugee Council

Chapter 1: Introduction

1.1 Introduction

Interest in services, and the differences between provision and consumption of services on the one hand, and products or goods on the other, dates back to the late 18th century if not further (Smith, 1776). More than two centuries later, and experts are still discussing definitions and distinctions (Rathmell, 1966). The role of services in our economy has broadened, and its significance to almost all aspects of society is now widely recognised (Maciosek, 1995). Even producers of physical goods, such as cars, aircraft, engineering products, from manufacturers such as Renault, Nissan, Tesla, Airbus, Roll Royce, Boeing, GE, see their services as a key competitive battleground, and it is increasingly hard to disentangle the provision of the physical product from the delivery and consumption of services around it. These include the continuous delivery of replacement batteries for electric cars, the continuous monitoring and optimisation of complex engineering components such as engines and turbines, the provision of financial services to facilitate the purchase and maintenance of products, and in the case of Tesla, even the delivery of new product features and capabilities of a car via the Internet.

The role of the state in providing services to its citizens and businesses has also expanded substantially since economists began to discuss services in distinct terms. This provision of public services has, in the case of most countries, moved from that of a basic 'safety net' to aid those who have not means to help themselves with the most fundamental basic needs in areas such as healthcare, shelter and food, to the provision of a much wider range of services that reach almost all citizens and business across all sectors of society and the economy.

In the private sector, competition between service providers has led to growing interest in understanding customer satisfaction, and in evaluating the quality of services from a number of different perspectives (Berry, 1991). The leisure industry, particularly hotels and airlines, were among the first to develop and

adopt formal methods to measure customer satisfaction and service quality (Parasuraman, Zeithaml and Berry, 1985). Banking and financial service providers were quick to follow, and the recognition that services can be designed and that design choices can be made to influence the nature and quality of the resulting service emerged in the 1980s and 1990s (Shostack, 1982). The domain of service design, and with it a new profession of service designers, had arrived.

Services provided by the public sector were less subject to competitive pressures, but the growth of service consumption from the private sector, and the consequent increase in service quality, led to growing recognition of a gulf in quality between services provided by the private and public sectors. As the web site of the Design Council states: "People expect public services to be as good as those they receive from the private sector, but our public services were designed for another time." (Design Council, 2016a). In the 21st century, the term "citizen-centric services" emerged (King and Cotterill, 2007, p. 342), and the disciplines of measuring customer satisfaction and service quality, and of designing services to satisfy the needs and desires of customers as well as service providers, were adopted by the public sector from the private sector (Wisniewski and Donnelly, 1996). But many public services are rather more complex than their private sector counterparts. Public services often have a wider range of stakeholders to be satisfied, and their demands are sometimes conflicting: just maximising the satisfaction of the service recipient may not produce an optimal public service (Vedung, 1999). The job of a designer of public services is complicated by the difficulty in measuring their success – what makes a 'good' service design, and how does one measure the quality of one possible service design over another?

This question of how to measure the impact of different service designs was the trigger for what became the research project described in this thesis. It was posed by a professor during a discussion about the author's research proposal to explore the potential impact of new ways to design public services. The question triggered further investigation, which revealed to the author the great

difficulty in measuring public services. They are multi-dimensional, they have many stakeholders, they sometimes do not even have a clearly identifiable customer. How does one measure a public service?

The author began to explore existing literature on the topic of service evaluation, and following discussions with a number of authorities on the topic, the answer that emerged was 'with considerable difficulty'. The problems described above seemed familiar to the author, especially in the context of an evaluation of competing designs. As a software design professional, he had nearly thirty years' experience in software development, design and architecture. Software solutions are also similarly multi-dimensional, and what is positive for one stakeholder is not necessarily positive for another. And whilst evaluation of software solution designs is recognised as a difficult problem, methods to address this problem emerged in the 1990s (Kazman et al., 1994; Bengtsson and Bosch, 1998), and have continued to evolve since then (Kazman, Klein and Clements, 2000). They explore an abstracted view of competing solutions by looking at their underlying architectures, and examine the trade-offs involved in each design as they inevitably favour some quality characteristics over others. This 'trade-off' analysis approach allows a group of stakeholders to better understand the strengths and weaknesses inherent in each proposed architecture, enabling them to consider how they would play in their particular context, and how they might mitigate any residual risks. The result is an approach that does not resolve the trade-offs, but it does allow the team to take a well-informed decision based on a deep understanding of each of the competing designs.

The author's familiarity with these methods, as applied to software architectures, prompted the transfer of learning that underpins this thesis: might these methods be applicable to the world of service design, given the apparent similarities between these two domains? To begin with the idea was presented in the context of the original research proposal to explore the potential impact of mass collaboration on the design and delivery of public services.

1.2 Mass collaboration and service design

In “An approach to investigating the potential impact of mass collaboration on the design and delivery of public services” the author describes the trend towards a citizen-centric approach to designing and delivering public services, and proposes the analysis of mass collaboration and its potential application as a vehicle for achieving that citizen-centric approach (Field, 2009a, p. 7). In addition to examining the characteristics of mass collaboration, the paper explores the term “impact”, highlights the multi-dimensional nature of service characteristics and the wide range of interested stakeholders, each of whom may have a distinct perspective on the ideal set of characteristics for a given service, and concludes that a multi-dimensional framework for analysing service designs is a pre-requisite for understanding the potential impact of different approaches to service design and delivery, including those that might involve mass collaboration.

The paper's focus on *potential* impact as opposed to *actual* impact implies a desire to predict impact in advance of the deployment of a new service or application of mass collaboration. What is being suggested here is therefore a form of ex-ante evaluation, that can exert an influence over the design, development and deployment of the service and raise the likelihood of its success, rather than the more typical ex-post evaluation that looks back on the actual impact following the deployment (though it does not rule out the possibility of doing both).

This distinguishes it from ex-post evaluation which is the focus of most literature on the broader topic of public service evaluation. Indeed Vedung's definition of 'evaluation' points only to ex-post evaluation, with its insistence that evaluation is “retrospective” (Vedung, 1999, p. 7). Ling, from the National Audit Office in the UK, is a rare example of someone proposing the adoption of ex-ante evaluation for public services (Ling, 2003), in this case the adoption of Scenario Planning methods to explore potential value for money in different future scenarios.

1.3 The research issue to be addressed

“An approach to investigating the potential impact of mass collaboration on the design and delivery of public services” (Field, 2009a) describes at a high level the application of attribute-based methods for evaluating software architectures, and asks the question:

“might it be possible to develop an 'attribute-based methodology', similar to that used to evaluate software architectures, with which to assess the potential impact of mass collaboration on the design and delivery of public services?” (Field, 2009a, p. 5)

The research issue addressed by this thesis is derived from that question. It is:

An exploration of whether an attribute-based evaluation method, derived from a software architecture evaluation method, can be usefully applied to the evaluation of competing service designs.

A research project to conduct this exploration was initiated, and the original research into the potential impact of mass collaboration on the design and delivery of public services was suspended. This thesis presents the results of that research project, and it should be noted that in formulating the research issue to be addressed, the original question has been broadened in two respects:

- a) Firstly, it considers the potential of an attribute-based methodology to be used to evaluate competing service architectures irrespective of whether they involve the application of mass collaboration. The need for a formal method of ex-ante evaluation is not specific to applications of mass collaboration and can be broadened to consider competing service designs more generally. This mirrors the use of attribute-based methods to review competing software architectures, since those methods also do not constrain the types of software or design that can be considered.
- b) Secondly, just as the software architecture review methods are not tied to

a particular subject matter domain, so the research has explored the broader domain of all services, from both the public and private sectors, and their evaluation, while also considering any unique characteristics of public services, and whether these might affect any proposed method of evaluation.

As the project was setting out to explore the application of an evaluation method from one domain (software architecture) to another (service design), it follows that there was first a need to investigate and understand these two domains in more detail. How have the software architecture evaluation methods evolved, and what are the characteristics of software and software architecture that might lead to a successful application of one of these methods to the evaluation of competing service designs?

1.4 Aim and objectives

The aim of the research was to achieve both a theoretical exploration of the research issue and, if possible, a practical exploration by applying an attribute-based evaluation method to address the needs of a real service design project. The objectives were therefore as follows:

1. Explore the similarities between the domains of software architecture evaluation and service design.
2. Explore the differences between the domains of software architecture evaluation and service design.
3. Develop a new evaluation method, derived from those applied to software architecture evaluation, suitable for application to the evaluation of competing service designs, reflecting the similarities and differences uncovered in the earlier objectives.
4. Identify a suitable service design project that could benefit from the first application of this new evaluation method, and that could contribute to a greater understanding of the practical consequences of applying the

method in service design projects.

5. Further refine the new evaluation method in the light of this first practical experience.
6. Identify additional avenues of research and investigation that are beyond the scope of this research, but could make a further valuable contribution to the domains of software architecture evaluation and service design.

The first objectives were to understand both domains (software architecture evaluation and service design) in greater detail, exploring similarities and differences between them, leading to an assessment of the potential to apply an evaluation method developed for use in one domain (software architecture) in the other (service design). Software architecture evaluation methods focus on the differing impacts of alternative architectures on the quality characteristics of the resulting systems. The research therefore needed to understand how the quality of software is described, and how this compares with the ways in which the quality of services are described. The intangible nature of both software and services suggests the likelihood of uncovering synergies, and these needed to be explored in some detail.

The project also needed to explore differences, and determine whether they might be sufficiently significant to demand a change to the evaluation methods before they could be applied to the new domain of service design. The results of these early objectives could then be applied to the next objective, the development of a new evaluation method, derived from those applied to software architecture, but tailored to the new domain of service design.

The practical objectives involved the identification of a suitable candidate service design project, capable and willing to make use of the proposed new evaluation method. This “first use” of the method would inevitably result in further refinements of the method, and the identification of further research questions that would be beyond the scope of the current research, but would have the potential to shed further light on the issue being addressed for the first

time in this research.

1.5 Accelerated learning

This research project, and the resulting thesis, were prompted by a combination of the author's experience of applying and refining software systems architecture evaluation techniques, and his interest in the emerging field of service design. The result is a novel application of a method developed for one domain, software systems, applied to another, service design. One might ask the question "why has a similar evaluation method not already been developed for the service design domain?". Service design is a relatively young domain, and as the review of literature that follows below will reveal, it has been evolving only since emerging as a field in its own right in the early 1980s. Software design began with the birth of the modern computer, in the 1940s, and it took thirty years before the concept of software architecture began to emerge, and a further twenty years before architecture evaluation methods were first published.

By drawing attention to the potential to apply a technique from one domain to that of a newer, much less mature, domain, service design, it is hoped that this thesis can contribute to efforts to accelerate the evolution of the domain of service design. Is this worthwhile? In the domain of software design, architecture evaluation methods that are now regularly applied by software architecture professionals, provide a degree of assurance that the design chosen for a given software solution is that most likely to best satisfy the complex and often conflicting set of requirements. Without such methods, discovery that the resulting software system is not 'fit for purpose' only after considerable time, effort and expense have been consumed in creating it, would be a regular occurrence.

We live in a world dominated by services, delivered by both the public and private sector. Many product producers are recognising that the services they wrap around their products are their key differentiators, so even the traditional, tangible, world of products is becoming transformed into a world of services that

encompass products within them. Car manufacturers, for example, now talk about delivering mobility services, and we are seeing a gradual move away from product ownership to consumption of services. It took the software domain fifty years to reach the level of maturity where the need for formal design evaluation methods was recognised and addressed. There is no need for the service design domain to take such a long time to reach the same conclusion, and this thesis represents the first steps towards the creation of service design evaluation methods that will help designers choose the right design before implementation, improving the quality of services, reducing the cost of their development and delivery, and avoiding what might be very costly mistakes.

This thesis builds on the work of others who have highlighted the links between the domains of software development and service design. It adds to these links by demonstrating how a particular software evaluation technique developed to consider competing software architectures can be applied to service design, and how a quality model developed for assessing the quality of software systems brings new perspectives to the approaches that have been adopted to date when assessing the quality of services. The novel introduction of a stakeholder model to the trade-off analysis approach reflects the significance of stakeholder perspectives in service design.

A case study, applying the proposed service architecture evaluation method to a project designing an important part of the services of the UK Border Agency in dealing with asylum seekers, demonstrates the practical application of the method. The tools and models that were developed to support this case study, and subsequently refined in the light of that experience, have been generalised and documented so that the method can be more widely adopted in future service design projects. Further research is needed to understand the limits and potential of the method's applicability to different kinds of services, from both the public and private sectors. And the case study raises a number of questions about particular aspects of the method that can benefit from closer study, which should result in further refinement of the method and greater understanding of how it can be used.

Stakeholder perspectives are introduced into a method that is already in widespread use for the evaluation of software architecture as a key change to adapt the method to the ex-ante evaluation of service designs. We will see that an unanticipated beneficiary of this extension is the software architecture community. The value of this new approach to the community that first developed architecture evaluation methods has been acknowledged with the publication and presentation of papers at several conferences and the method's adoption by IT architecture professionals at a number of major corporations.

1.6 Thesis overview

Chapter 2 of this thesis, Background and Literature, first explores the domain of software architecture, with a focus on software quality and the evolution of architecture evaluation methods. It then considers the nature of services, the relationship between services and their many stakeholders, the emerging discipline of service design and the role of quality models in the practice of service evaluation. Drawing together its review of software architecture evaluation methods, the similarities and differences between software architecture and services, and the more recent emergence of service design, it concludes that a method for conducting ex-ante evaluation of competing service designs could be derived from existing software architecture evaluation methods, with the incorporation of stakeholder perspectives to take account of their consideration in the domain of service design.

Chapter 3, Approaches and Methods, looks more closely at the research question that lies behind this thesis, and sets out the proposed method to address it. In so doing, it explores in some detail the two phases of the project, the first theoretical phase examining the two domains under consideration, and the second phase consisting of the practical application of a proposed service evaluation method that takes shape as a consequence of the first phase. The chapter explores different possible research approaches and concludes that a case study approach is the most suitable one for this project. Adoption of a single case study is proposed and justified, and the criteria for selecting that

case study are set out. The adoption of a case study at the UK Border Agency is described, and approaches to collect and analyse information at multiple levels are discussed. The chapter concludes with a discussion of potential ethical and other issues associated with the adopted approach, and how these might be dealt with.

Chapter 4, Creation and Development of the Service Architecture Review Method (SARM), describes the transition between the theoretical phase of the research project and the practical phase. This involves the development of an evaluation method suitable for application in a service design case study. The chapter looks at the selection of a suitable source method from among the software architecture evaluation methods surveyed in Chapter 2. It then describes how the project leader adapted the method to suit the particular requirements of service design, in line with the conclusions reached at the end of the theoretical phase of the research project, as described in Chapter 2. The development of a software tool to support the new Service Architecture Review Method is described, and the chapter concludes with a summary of the further enhancements to that method that were developed during and after its first use in the case study with the UK Border Agency.

Chapter 5, Service Architecture Review Method, represents a presentation of the Service Architecture Review Method in full. It takes the form of a practitioner's user guide, explaining how to apply the method and use the accompanying software tool. The version of the method described here is the refined method, incorporating the further enhancements that were applied during and after the completion of the UK Border Agency case study. Readers interested in understanding the earlier version of the method, developed for the commencement of the case study, can deduce the details of it by reading Chapter 5 in conjunction with the latter part of Chapter 4, which articulates those elements of the method and accompanying software tool that were enhanced as a consequence of the case study experience.

Chapter 6, Service Design Case Study, describes the first use of the Service

Architecture Review Method (SARM) in a case study service design project at the UK Border Agency. The chapter makes extensive use of the words of the participants to reflect the impact the method had on their project, the organisation, and the participating individuals themselves. These were captured in interviews that took place shortly after the completion of the service design project.

Chapter 7, Analysis, examines the primary data that were collected during the case study in more detail. These include observations, formal meeting notes, data and decisions captured in the software tool developed to support SARM, and the transcripts of eight interviews with the UK Border Agency participants that constituted the service design evaluation team. The chapter identifies and discusses five themes that characterise the findings of the research, linking together the relevant literature reviewed in Chapter 2 with the results that emerged from the case study described in Chapter 6.

And finally, Chapter 8, Conclusion, brings together the outcomes of the research described in this thesis. Contributions to both theory and practice are highlighted, limitations of this research, and new opportunities for further research are discussed, and the thesis concludes by putting this research into the broader context of transferring knowledge between the domains of software architecture and service design.

Chapter 2: Background and Literature

2.1 Introduction

Chapter 1, Introduction, has described the origins of this research, and the manner in which the underlying research question came about. As set out in that chapter, the objectives of this research involve an exploration of similarities and differences between the domains of software architecture evaluation and service design with the aim of exploring the potential application of an attribute-based method to a set of competing service designs. Key themes that emerge from this chapter are:

1. The concept of software architecture;
2. The development of attribute-based methods for conducting ex-ante evaluations of software architectures;
3. The intangible nature of both software and services;
4. The development of customer-centric models of service quality;
5. The emergence of service design as a discipline;
6. The importance of stakeholder perspectives in service evaluation;
7. The applicability of the concept of architecture to services, and the potential to apply software architecture evaluation methods to service designs.

This chapter begins by looking at the emergence of software architecture as a concept, and the evolution of architecture evaluation techniques in the 1990s. Of particular interest to this research is the concept of software quality. It is widely used as a context for ex-ante architecture evaluation, and for this research, it also provides a basis for comparison between the domains of software architecture and service designs. The history of various evaluation methods is explored, leading to the dominance of one particular method, the Architecture Trade-off Analysis Method, together with a number of derivatives.

The distinctive characteristics of services, as opposed to tangible products and goods, are discussed, followed by an exploration of the history of service design, from its origins in the 1980s. The importance of stakeholders to services, and the complex relationship between many stakeholders and public services, are discussed, before models of service quality are considered. The chapter looks at ways in which the quality of both software and services have been modelled and described, and identifies a significant difference in scope, with the software domain attempting to consider multiple perspectives while the focus of service quality is concentrated on customer satisfaction.

The chapter also examines the nature of software, and the use of the term 'software engineering', and relates this to the nature of services and the discipline of service design. This addresses the following questions, which are likely to occur to the reader:

Why consider applying software architecture evaluation techniques to service design, when software architecture has itself drawn upon an analogy with the much more mature discipline of architecture of buildings?

Why not, therefore, look to the evaluation of building and engineering architectures when seeking methods that might be applicable to service design evaluation?

Given the similarities in nature between software and services, the chapter concludes that the use of the 'architecture' analogy in the software domain has similar potential for the service domain, and therefore that the application of software architecture evaluation techniques to the domain of service design is worthy of investigation.

2.2 Software architecture and architecture evaluation

Software architecture has emerged as a substantial field for both academic study and practical use in the 1990s, but its origins lie in the work of Edsger

Dijkstra in the 1970s, in which he sought to apply formal verification to software as part of the process of construction. Prior to this, the conventional approach was to first construct the software, then apply formal mathematical techniques to verify the correctness of the software. Dijkstra proposed a more structured approach to software development, enabling it to be “correct by construction” (Dijkstra and Hoare, 1974, p. 5). As will be seen later in this section, a description of the structural components of software, and the relationships between those components, is fundamental to the notion of software architecture and its articulation.

Giving the keynote address at the ACM Symposium on Applied Computing in San Antonio on 1 March 1999, he said:

“After more than 45 years in the field, I am still convinced that in computing, elegance is not a dispensable luxury but a quality that decides between success and failure; in this connection I gratefully quote from The Concise Oxford Dictionary a definition of 'elegant', viz. 'ingeniously simple and effective'.” (Dijkstra, 1999, p. 21)

This notion of 'elegance', taken together with his particular interest in a structural approach to software development, suggests the concept of software architecture which, as shall be seen in subsequent definitions, has come to be associated with form, design and abstraction. This is despite the fact that he has not here directly used the term 'architecture', and he very rarely referred to the term “architecture” in his speeches and writings throughout his long and distinguished career (E.W. Dijkstra Archive, 2010). Interestingly, the connection between beauty and architecture was made explicitly in an article by Grady Booch in IEEE Software when, referring to software-intensive systems, he enthused about “studying their architectural patterns and thus exposing these systems' inner beauty” (Booch, 2006, p. 17). This reinforces the analogy being made between the design of software and the design of buildings in the use of the term architecture. The inclusion of elegance and beauty as essential ingredients in architecture goes back to the very origins of western architecture,

with Vitruvius including Beauty as one of his three “architectural universals” along with Utility and Firmness (Marcus Vitruvius Pollio, 25AD).

One of the first explicit references to architecture in the context of software engineering appears in “The Mythical Man-Month” by Fred Brooks (Brooks, 1975, p. 45). Brooks refers to architecture as “the complete and detailed specification of the user interface”, and suggests that “the architect of a system, like the architect of a building, is the user's agent”. It took some ten to fifteen years for the notion that software systems have an architecture to become more widely accepted, and over that period, a broader definition of software architecture evolved.

In “The Golden Age of Software Architecture” (Shaw and Clements, 2006), a look back at the period when software architecture emerged as a distinct topic, the authors relate the evolution of software architecture to the typical phases of maturation identified by Redwine and Riddle (Redwine and Riddle, 1985). They classify the period from 1985 until 1993 as “Basic research”, and indeed the first papers attempting to define “Software Architecture” appear at the end of this period (Shaw and Clements, 2006, p. 32–33).

“Foundations for the study of Software Architecture”, which was published in 1992, described software architecture as being “concerned with the selection of architectural elements, their interactions, and the constraints on those elements and their interactions necessary to provide a framework in which to satisfy the requirements and serve as a basis for the design” (Perry and Wolf, 1992, p. 43). This description is aimed at distinguishing architecture from requirements, design and implementation. The authors highlight their choice of the term “architecture” (and distinguish it from the term “design”) in order “to evoke notions of codification, of abstraction, of standards, of formal training (of software architects), and of style.” (Perry and Wolf, 1992, p. 40)

The second, “Concept formalization” phase, took place, according to Shaw and Clements, from 1992 to 1996. This is the period when definitions and classifications became well established. In 1994, Hayes-Roth described

software architecture as “an abstract system specification consisting primarily of functional components described in terms of their behaviors and interfaces and component-component interconnections.” (*Software Engineering Institute, 2009a*)

The April 1995 edition of IEEE Transactions on Software Engineering was, for the first time, devoted to the subject of software architecture, and the guest editors, David Garlan and Dewayne Perry, proposed the following definition for software architecture:

“The structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time.” (Perry and Garlan, 1995, p. 269)

Perhaps the best-known, and most widely respected, book on the topic is “Software Architecture in Practice” by Len Bass, Paul Clements and Rick Kazman (Bass, Clements and Kazman, 2003), the first edition of which was published in 1998. It contains the following definition for Software Architecture:

“The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationship among them.” (Bass, Clements and Kazman, 2003, p. 21)

Common to all three of these definitions are the concepts of components, interfaces and relationships, together with the implication of abstraction that follows from this level of description. And these aspects are reflected in the formal definition adopted by the IEEE Standard 1471-2000 “Recommended practice for architectural description of software-intensive systems” (which is also an ISO standard, ISO/IEC 42010:2007):

“The fundamental organization of a system, embodied in its components, their relationships to each other and the environment,

and the principles governing its design and evolution.” (International Organization for Standardization, 2007, p. 14)

The Software Engineering Institute at Carnegie Mellon University (SEI) has collected over sixty definitions of “Software Architecture”, publishing them under “Modern”, “Classic”, “Bibliographic” and “Community” categories on its web site (Software Engineering Institute, 2009b). The majority of these definitions comes from this period of the late 1990s.

Whilst there is clearly not a single form of words that is universally accepted, the Bass, Clements and Kazman definition is widely quoted in associated literature, and lies towards the centre of the set of definitions collected by the SEI.

The authors highlight a number of implications of this definition. They argue that an architecture is an abstraction, defining a set of elements, but suppressing those details of the elements that do not relate to how they use, are used by, or interact with, other elements. The definition, they claim, also implies that systems comprise more than one structure with critical relations between them. Following on from this, they assert the implication that all systems have an architecture (even though it may not always be explicitly described), and that the external behaviour of the elements that comprise a system are part of that system's architecture.

2.3 The role of the architecture

The growth in complexity of software systems over time has led to recognition of the importance of abstraction as an aid to communication among the stakeholders of a system, especially during its specification and construction. A representation of the architecture becomes a lingua franca that all stakeholders can speak and understand, as well as serving as a technical “blueprint” for the system that is to be built, modified or analysed.

A somewhat broader definition of Software Architecture was proposed by Professor Barry Boehm and his students at the First International Workshop on

Architectures for Software Systems:

“A software system architecture comprises:

- A collection of software system components, connections and constraints.
- A collection of system stakeholders-need statements.
- A rationale which demonstrates that the components, connections, and constraints define a system that, if implemented, would satisfy the collection of system stakeholder need statements.” (Gacek et al., 1995, p. 2)

This definition goes beyond most commonly agreed definitions, in taking in the stakeholders and their needs, but it does clearly highlight a primary purpose of the architecture; namely to facilitate the satisfaction of stakeholder needs.

There would appear to be broad agreement that the primary purpose of the architecture is to act as a central reference point, drawing together stakeholders, requirements, developers and the design. For example, Perry and Wolf in 1992 suggested that the architecture is “to provide a framework in which to satisfy the requirements and serve as a basis for the design” (Perry and Wolf, 1992, p. 43).

The IEEE recommended practice for architectural description of software intensive systems goes further, highlighting thirteen uses of architectural descriptions (and suggesting that there are other uses beyond these listed):

- a) Analysis of alternative architectures
- b) Review, analysis, and evaluation of the system across the life cycle
- c) Business planning for transition from a legacy architecture to a new architecture
- d) Communications among organizations involved in the development, production, fielding, operation and maintenance of a system
- e) Communications between acquirers and developers as a part of contract negotiations

- f) Criteria for certifying conformance of implementations to the architecture
 - g) Development and maintenance documentation, including material for reuse repositories and training materials
 - h) Input to subsequent system design and development activities
 - i) Input to system generation and analysis tools
 - j) Operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, subsystems, and components
 - k) Planning and budget support
 - l) Preparation of acquisition documents (e.g., requests for proposal and statements of work)
 - m) Specification for a group of systems sharing a common set of features (e.g., product lines)
- (International Organization for Standardization, 2007, p. 8)

The use of the architectural description as a communications vehicle is prominent among these listed uses.

This growth in the complexity of systems has also led to increased interest in the relationship between the architecture of a system and the behaviour of that system. Most of the definitions of software architecture include direct or indirect references to behaviour and the satisfaction of requirements. For example, the Perry and Wolf definition refers to satisfying requirements (Perry and Wolf, 1992), Hayes-Roth to the behaviour of functional components (Software Engineering Institute, 2009a) and Bass, Clements and Kazman to “the external visible properties” of software components (Bass, Clements and Kazman, 2003). Since the architecture encapsulates the external behaviour of the components, it is a combination of those behaviours that will make up the external behaviour of the resulting system. It follows that different architectures can result in different behaviours that might be closer to, or further away from, the set of behaviours most desired by the stakeholders. It is also quite possible that different stakeholders will have different views on the behaviours that are most important or most desired.

2.4 Software quality

Analysis of the behaviour of software has long been studied and discussed under the topic title of “software quality”. The IEEE 1061 Standard for a Software Quality Metrics Methodology describes software quality as “the degree to which software possesses a desired combination of attributes” (Software Engineering Standards Committee of the IEEE Computer Society, 1998, p. 3). The term “quality attribute” has become a standard element in the language of the software architect, though among practising architects there is not yet a common set of attribute names. Wikipedia, for example, contains a catalogue of over sixty differently named quality attributes. Given that Wikipedia is itself maintained by the community of practitioners, this diversity of language is evidence that professional practices among software architects are emergent, and have not yet reached the stability that might be expected in a more mature professional community.

Whilst there is not yet a common set of attribute names among practising architects, there have been attempts to use international standards to describe more clearly the universe of software quality and avoid the degree of overlap and duplication that is evident in, for example, the wikipedia catalogue.

ISO 9126 (International Organization for Standardization, 1991) is an international standard for evaluating software quality. Part one of the standard (ISO 9126-1) sets out a quality model which classifies software quality in a structured set of characteristics and sub-characteristics (see Table 1).

Characteristic	Sub-characteristic
Functionality	Suitability Accuracy Interoperability Security Functionality Compliance
Reliability	Maturity Fault Tolerance Recoverability Reliability Compliance
Usability	Understandability Learnability Operability Attractiveness Usability Compliance
Efficiency	Time Behaviour Resource Utilisation Efficiency Compliance
Maintainability	Analysability Changeability Stability Testability Maintainability Compliance
Portability	Adaptability Installability Co-existence Replaceability Portability Compliance

Table 1 ISO 9126-1 Software Quality Model

The ISO standard makes a clear distinction between the term “attribute” and the term “characteristic”. Within this standard, an attribute is a measurable value that sits within, and contributes to, a quality sub-characteristic, which in turn sits within a quality characteristic.

“In this model the totality of software product quality attributes are classified in a hierarchical tree structure of characteristics and subcharacteristics.” (International Organization for Standardization, 1991, p. 14).

However, it acknowledges that the tree structure is not perfect, and that some quality attributes may contribute to more than one sub-characteristic or

characteristic.

In making this distinction between “attribute” and “characteristic”, the ISO standard is somewhat out of step with other standards, and with common usage of the terms. This is confirmed within the document itself, where it acknowledges that “the term 'attribute' [in ISO/IEC 14598] is used with the same meaning as the term 'characteristic' used in 4.1.1” (International Organization for Standardization, 1991, p. 20; International Organization for Standardization, 1999). The IEEE standard implies that it considers the terms “attribute” and “characteristic” to be interchangeable, defining a Quality Attribute as “a characteristic of software” (Software Engineering Standards Committee of the IEEE Computer Society, 1998, p. 3), and it is clear that the term “attribute” in IEEE 1061 and ISO/IEC 14598 represents the same concept as the term “characteristic” in ISO/IEC 9126.

In proposing its concise model of just six characteristics, broken down further into twenty seven sub-characteristics, the authoring technical committee responsible for ISO/IEC 9126 was satisfying its stated requirement to seek “not more than six to eight characteristics for reasons of clarity and handling” while also aiming “to cover together all aspects of software quality” and “to describe the product quality with a minimum of overlap” (International Organization for Standardization, 1991, p. 24).

The first entry in the list of uses of architectural descriptions in the IEEE Recommended Practice is “analysis of alternative architectures”, and in 1995 the Software Engineering Institute suggested the use of quality attributes to facilitate the evaluation of alternative software architectures. Quality Attributes, a technical report from the Software Engineering Institute at Carnegie Mellon University, introduces a generic taxonomy of software quality attributes, and proposes “an attribute-based methodology for evaluating software architectures” which involves analysing the trade-off between quality attributes offered by different possible software architectures (Barbacci et al., 1995, p. 45–46).

2.5 Software evaluation

This proposal led to the development of Software Architecture Analysis Method (SAAM) (sometimes referred to as Scenario-Based Architecture Analysis Method) at the Software Engineering Institute, which was “the first documented, widely promulgated architecture analysis method” and was “originally created to analyze an architecture for modifiability” (Clements, Kazman and Klein, 2001, p. 211).

The authors had observed that practitioners regularly made claims about their software architectures that were effectively untestable. These claims typically referred to quality attributes, but in a rather general fashion, e.g. “Using CORBA will make our system easy to modify and upgrade”. SAAM was developed to provide a framework in which such claims could be tested.

The authors contend that “software architectures are neither intrinsically good nor intrinsically bad; they can only be evaluated with respect to the needs and goals of the organizations which use them.” (Kazman et al., 1994, p. 7). They also highlight the abstract nature of quality attributes, and suggest that competing architectures can best be evaluated in the context of the ramifications of the quality attributes that are critical to the system under consideration. The method focuses on the Modifiability quality attribute (equivalent to the Changeability sub-characteristic in the ISO 9126-1 Software Quality Model), and so requires the creation of a set of scenarios that illustrate what sorts of modifications to a software system are likely or representative of a given domain.

The method then involves the evaluation of each candidate architecture according to how well it supports the set of likely modifications, or benchmark tasks. It is further suggested that “this set of benchmark modifications can often be given a sample distribution, for the purposes of ranking the individual evaluations” (Kazman et al., 1994, p. 7). In their concluding remarks, the authors suggest that, whilst SAAM is concerned with evaluating architectures against a single quality attribute, Modifiability, they believe “SAAM will work for

other attributes” (Kazman et al., 1994, p. 9).

SAAM spawned a number of extensions that explored this potential to evaluate architectures against multiple quality attributes. Scenario-Based Architecture Reengineering (SBAR) was described by Bengtsson and Bosch in 1998 (Bengtsson and Bosch, 1998). Like SAAM, it proposes the use of scenarios to concretise the actual meaning of quality attributes. It introduces three additional techniques beyond scenarios for assessing the quality attributes – simulation, mathematical modelling and experience-based reasoning.

A year later, the same authors proposed Architecture Level Prediction of Software Maintenance (ALPSM) (Bengtsson and Bosch, 1999) which, like SAAM, from which it was derived, focuses specifically on Modifiability. 1998 also saw the appearance of A Software Architecture Evaluation Method (SAEM) (Duenas, Oliveira and Puente, 1998). SAEM evaluates software architectures against a specific quality model using Goal Question Metric Approach (Basili, Caldiera and Rombach, 2002).

The Software Engineering Institute took up their own proposal to extend SAAM to consider multiple quality attributes, and the resulting Architecture Trade-Off Analysis Method (ATAM) (Kazman, Klein and Clements, 2000) has become the most widely used, and most mature of the many software architecture evaluation methods that have been adopted (Ali Babar, Zhu and Jeffrey, 2004, p. 8). Application of this method involves the development of a set of scenarios drawn from the set of relevant quality attributes, and analysing the different trade-offs between these offered by competing alternative architectures.

The team that originally developed ATAM have also added an extension called Cost Benefit Analysis Method (CBAM) which takes the output of an ATAM evaluation and enables the review team to model costs and benefits from an architectural perspective (Kazman, Asundi and Klein, 2001).

A survey of methods published in IEEE Transactions on Software Engineering in 2002 concluded that ATAM is the most suitable software architecture

evaluation method, based on criteria proposed by the authors (Dobrica and Niemela, 2002, p. 650). A framework for comparing software architecture evaluation methods, published in 2004, also concluded that “only one method, ATAM, provides comprehensive process support” (Ali Babar, Zhu and Jeffrey, 2004, p. 9).

One of the major benefits of assessing different architectures before any have been implemented lies in the high cost of changing a system once it has been developed. If a system fails to exhibit appropriate behaviour in a key quality attribute it is likely that an inappropriate software architecture has been selected. Correcting such a fault is often difficult, involving fundamental re-engineering of the solution.

Modern software development methods, such as Rational Unified Process (Jacobson, Booch and Rumbaugh, 1999) and DSDM Atern (DSDM Consortium, 2008), place the development of the architecture of a system relatively early in the life-cycle, prior to the more labour intensive activity of developing the software. By conducting an evaluation of possible architectures at this early stage, a project is able to ensure that the most suitable architecture is chosen before most of the resources are committed to the project, increasing the chances of a successful project, and maximising the likelihood that the system will satisfy the wishes of the system's stakeholders.

ATAM involves an intensive review by a team of stakeholders, which might include its developers, the system's owner, some users, and those who will have responsibility for running, operating and maintaining the system. The review team collectively develop a set of scenarios that are placed in the context of the key quality attributes by creating a utility tree (where each leaf node is a scenario, and the branch nodes are quality attributes). The tree is then analysed for each architectural approach that is under consideration, uncovering risks, sensitivity points and trade-off points in the tree and deriving a view of the overall utility of each approach. A major benefit of this evaluation method is that the assessment of different architectural approaches is directly

related to those quality attributes that are most important in the eyes of the stakeholders for the system under consideration. It reflects the “multi-attribute” nature of a system's behaviour and exposes the inevitable trade-offs between the different attributes for any one solution.

The Office for National Statistics (ONS) has developed a variation of this method to conduct its own software architecture reviews (Field, 2009b). There are a number of key changes to the ATAM method upon which the ONS method is based. Early attempts at using ATAM at ONS revealed problems in relating the relatively abstract concept of quality attributes to the real needs of a business system, especially on the part of business stakeholders, who were not typically IT professionals, and thus had no previous exposure to the concept of quality attributes.

As a consequence, the ONS method involves an initial assessment of documented project requirements against a list of quality attributes. The result is a matrix that identifies the relationship between requirements and quality attributes. The matrix reveals quality attributes that do not have any corresponding requirements, raising the possibility that either the quality attributes are not as significant as first thought, or that some essential business requirements have yet to be identified and documented. Discussing and populating this matrix has proved much easier than completing the corresponding activity in ATAM, which is the development of the tree of quality attributes and scenarios.

A second innovation introduced by ONS occurs during the trade-off analysis. The trade-off analysis of ATAM deals with utility, showing how utility attached to scenarios varies across different solution approaches. ONS found that stakeholders lacked a common understanding of utility, and agreement on its quantification proved troublesome. The organisation was, however, very familiar with a common corporate approach to risk estimation, with risks being broken down into two factors: Measure of Likelihood and Measure of Impact.

This approach to the measurement of risk is in widespread use. The international standard for occupational health and safety management systems, OHSAS 18001, defines risk as the:

“combination of the likelihood of an occurrence of a hazardous event or exposure(s) and the severity of injury or ill health that can be caused by the event or exposure(s)” (British Standards Institute, 2007, p. 4)

Put more generally, this is the combination of the likelihood of an event occurring and the impact of that event, should it occur.

This is commonly expressed in a quantifiable form by expressing both the likelihood and impact measurements numerically, and calculating the product of these to generate a numeric expression of the overall risk. One of the earliest uses of this simple formula was in the Delta Works project in the Netherlands in 1953. Advice from Dutch mathematician, David van Dantzig, led the project to use this approach to assess flood risk. The approach is now commonly used in fields such as nuclear power, aerospace and chemical manufacturing (Wolman, 2008).

For the corporate risk model used by ONS, each factor has five possible categories bearing a description, summarised with a simple one or two word descriptor and a corresponding numeric level (see Table 2 and Table 3 below for details).

A measure of exposure for each risk is derived by calculating the product of the measure of likelihood and the measure of impact:

$$\text{Impact} \times \text{Likelihood} = \text{Exposure}$$

The resulting exposure measurement is then categorised with a descriptor and colour coded as shown in Table 4 below.

Level	Descriptor	Description
6	Disastrous	Key ONS business objectives not met Significant embarrassment is caused to the ONS Compromises ONS's reputation as a quality organisation
4	Major	Minor embarrassment to ONS Non-key ONS business objectives not met Major disruption to specified work programme
3	Moderate	Insignificant disruption to specified work programme Moderate disruption to specified project
2	Minor	Minor disruption to specified project
1	Insignificant	Of no consequence

Table 2 ONS Risk Assessment Measure of Impact

Level	Descriptor	Description
5	Almost Certain	Is expected to occur
4	Likely	Very good chance that event will occur
3	Possible	Equal chance that event will occur
2	Unlikely	Could occur at some time
1	Rare	May occur only in exceptional circumstances

Table 3 ONS Risk Assessment Measure of Likelihood

Exposure Score	Descriptor
1 – 4	Low Risk
5 - 12	Moderate Risk
13 - 20	High Risk
24 - 30	Corporate Risk

Table 4 ONS Risk Assessment Measure of Exposure

As this risk model was already widely understood among staff at ONS, it was decided to adopt this same model for its software architecture reviews, with the risk of failure to achieve requirements being “traded off” between solution approaches in place of utility. Whereas during an ATAM review, stakeholders must discuss and agree levels of utility for each scenario / solution combination, the ONS review method requires stakeholders to agree an “impact in the event of failure” level for each requirement, and a “likelihood of failure” level for each requirement / solution combination. The result is an analysis matrix that reveals how different solution approaches trade-off the risk of not achieving the set of

requirements against each other in different ways. A preferred solution approach can then be selected, based on agreement among the stakeholders regarding which risks can be more easily borne by the project.

All of the software architecture evaluation methods described above can be characterised as ex-ante “pre-implementation” evaluations. They evaluate a design before that design has been implemented, and are used to select the most suitable design with the aim of reducing the risk that the eventual implementation fails to satisfy the expectations of its stakeholders.

2.6 Is it really architecture?

The adoption of terminology such as “architecture” and “architect” by the software development community has become widespread. As we have seen, the “architecture” label is relatively recent, with the first conferences devoted to the topic taking place in the 1990s. Whilst architecture is a relatively late addition to the lexicon of the software development community, it builds on language that had already fallen into widespread use during the 1960s and 1970s (which is only around thirty years into the very existence of computers and computer software). For example, it is common to hear about the discipline of software “engineering”. Developers refer to the act of “building” or “constructing” software.

There are claims that the term “software engineering” first came into use in the 1950s but these have proved hard to verify. The first formal conference to use the term was organised by the NATO Science Committee and took place in Garmisch, Germany from 7th to 11th October 1968. It was entitled “Working Conference on Software Engineering” (Randell, 1996). This was the first of a pair of conferences on the subject, the follow-up conference taking place in Rome from 27th to 31st October 1969. These two conferences are seen by many in the software industry as representing the official start of the software engineering profession.

The use of such language suggests a similarity between the discipline of

constructing software and other physical engineering disciplines. In “Prospects for an Engineering Discipline of Software” (Shaw, 1990), Mary Shaw explores the evolution of an engineering discipline, suggesting a maturation process from craft to commercial with the introduction of production skills, and thence to professional engineering with the further introduction of scientific knowledge.

She goes on to plot the course of civil engineering, proposing the first century Roman period as the point at which the craft evolved into commercial production, and some time between 1750 and 1850 as the point at which the appreciation of scientific knowledge led to the formation of a professional civil engineering discipline. A similar analysis is given for the evolution of chemical engineering. However, when it comes to considering software engineering, the transition from craft to commercial is identified as taking place in the 1980s with the introduction of software development methodologies, but she acknowledges doubt as to whether there has been sufficient introduction of scientific knowledge to establish the transition from production to professional engineering.

There are those, perhaps a minority, that doubt whether such a transition will ever take place. It seems that, as early as that second NATO Conference in Rome in 1969, there were some present who had serious misgivings about the application of the term “engineering” to the practice of developing software. Brian Randell, who co-authored the published report of the conference, has described how the proposal to establish a NATO-sponsored International Software Engineering Institute was greeted with “considerable scepticism” instead of the anticipated “strong and extensive support” (Randell, 1996).

One of the participants, Tom Simpson of IBM, wrote a satirical article ridiculing the proposal. In it, he imagines a “Working Conference on Masterpiece Engineering” taking place in 1500 at which Leonardo daVinci and other great artists meet to discuss how to automate the production of masterpieces in response to increased demand. His point is clear – that software development is not a scientific discipline, that the analogy with engineering is inappropriate,

and that it would be more appropriate if software development were compared with the creative arts. This view has not prevailed (at least within the software development community), but in 1969 his article was suppressed by the conference organisers who refused a request that it should be included with the official report on the conference, and it was only published in 1996 (Randell, 1996).

The term “architecture” is also applied to the design of computer hardware, which itself is a branch of electrical engineering. This seems altogether more appropriate, in that computer hardware design is founded on the scientific knowledge of the physical properties of the electrical components that make up a computer. Today, new computers, chips, circuit boards and processors, are routinely designed, developed and produced. Architects and designers are able to predict the behaviour of computer hardware with a high degree of accuracy, based on their scientific knowledge of the physical properties of the constituent components. This is something that computer hardware shares with other forms of physical and chemical engineering, from the design and construction of buildings and aircraft, to the development of drugs and industrial chemicals.

The development of computer software is clearly not an engineering discipline in that sense. In 1990, Mary Shaw pointed at just three topics (algorithms, data structures and compiler construction) that might equate to scientific knowledge, and acknowledged that these were “isolated examples”. Today, the degree of certainty surrounding one of these, data structures, has probably diminished. The adoption of xml as a data modelling language and not just as a data transport mechanism, has spawned competitive thinking regarding data modelling principles. Most recently, notions of using existing internet protocols to create a global web of linked data (Berners-Lee, 2006) raise further questions about conventional wisdom on data structures. There is not, yet, a solid foundation of scientific knowledge underpinning software systems, and perhaps there will never be one. Computer software, by its nature, is intangible and comes into existence in a virtual, not physical world, governed by time but not space.

Perhaps, forty years after Tom Simpson expressed his doubts about “software engineering” in such an amusing way, we may be able to acknowledge that he was right.

Doubts about the validity of the term “engineering” in the context of software development help to highlight the particular characteristics of software that affect the complexity involved in its design. This does not, in any way, diminish the value and validity of the design evaluation methods discussed earlier in this chapter (though it does call into question the validity of their use of the term “engineering”, and perhaps by association, the term “architecture”). We will return to the intangibility of software when we consider the distinguishing characteristics of services and the similarity of services to software.

2.7 What are services?

Services have long been identified as having distinct characteristics that mark them out from more tangible products. In “The Wealth of Nations”, published in 1776, Adam Smith distinguished between goods that could be stored after production, and thus retain value, produced by what he called “productive” labour, and the results of “unproductive” labour that failed to create wealth because, in his view, they created services that “perished” at the time of production (Smith, 1776, p. 271). The French economist, Jean-Baptiste Say, described services as “immaterial products”, arguing that production and consumption are inseparable in services (Say, 1803, p. XIII).

These historical views of services have not endured. For many services, production and consumption are clearly separable (e.g. weather forecasting), and not all services are perishable (e.g. recordings of concerts). Most importantly, many services produce enduring value for their consumers (e.g. education). More recent definitions have focused on the distinction between ownership (in the case of products) and non-ownership (in the case of services).

For example, Lovelock and Gummesson suggest that services involve some

form of rental, whether it be of objects, labour and expertise, or facilities and networks (or some combination of these) (Lovelock and Gummesson, 2004). Rathmell considers marketed services to consist of transactions “where the object is other than the transfer of ownership of a tangible commodity” (Rathmell, 1966, p. 33). In distinguishing between goods and services, he also proposes a simple test:

“One implicit distinction is to consider a good to be a noun and a service to be a verb – a good is a thing and a service is an act. The former is an object, an article, a device or a material . . . whereas the latter is a deed, a performance or an effort.” (Rathmell, 1966, p. 33)

A more comprehensive definition of services is proposed by Lovelock and Wirtz:

“Services are economic activities offered by one party to another, most commonly employing time-based performances to bring about desired results in recipients themselves or in objects or other assets for which purchasers have responsibility.

In exchange for their money, time and effort, service customers expect to obtain value from access to goods, labour, professional skills, facilities, networks, and systems; but they do not normally take ownership of any of the physical elements involved” (Lovelock and Wirtz, 2007, p. 15)

This definition explicitly defines services as being an economic activity. In an attempt to be specific, it may have excluded at least some members of a major class of services: Public Services. Consider the public service of benefit distribution. Receipt of the service is not directly in exchange for payment by the customer, who may not even be a tax payer. Payment for public services is often indirect (via the public finances), rather than being a direct economic transaction between the individual customer who pays for the service and is the direct beneficiary and recipient of that service. The cross-subsidies of public finances, and the targeting of many public services such as benefit payments

towards customers who need the service but cannot, or have not, paid for that service, break the economic connection between the individual service customer and the service provider. In a more general sense, a whole community, such as a village, town, county or country, can be seen collectively as choosing to pay for, and receive a service.

Another group of service providers that may feel excluded by this definition is the Third Sector. The UK Government defines the Third Sector as “non-governmental organisations that are value driven and which principally reinvest their surpluses to further social, environmental or cultural objectives” (Department of Communities and Local Government, 2010). Included within that definition are voluntary and community organisations, charities, social enterprises, cooperatives, mutuals and housing associations. Organisations of this type would argue that they do provide services even though, in many cases, the recipient is not the one who pays for the service.

Considering public, private and third sector services, the simpler definition of “an act, deed, performance or effort” from Rathmell might be a better fit. However, this definition might be too simple, since it does not make any provision for providers and customers. Consider the example of an individual playing the piano for their own satisfaction. Is this performance the provision of a service? It seems more likely that it only become a service when there is an audience (at least one customer) to receive the service and derive value from it.

A satisfactory service definition therefore needs to accommodate the following characteristics that have been identified above:

1. Services involve customers and providers
2. Production and consumption of services are not necessarily simultaneous
3. Services create value for their customers

4. A service is an act (and not a thing)
5. Services do not always involve direct payment from the customer to the provider

IBM, on the web site that introduces the concept of “Services Sciences”, defines a service simply as:

“a provider/client interaction that creates value” (IBM Research, 2010)

This definition can be seen as something of a combination of the Lovelock and Wirtz definition with that from Rathmell. An act, deed, performance or effort is represented in the “interaction”, yet the definition adds the notion of a relationship between a provider and a client, and brings back from the Lovelock and Wirtz definition a part of the economic aspect; the creation of value. Yet it has withheld the direct payment aspect from the Lovelock and Wirtz definition that causes such difficulty in the cases of public services and the third sector. Of the various definitions discussed above, this is the only one that satisfactorily accommodates the five characteristics listed above.

2.8 Service Design

G. Lynn Shostack was among the first to articulate the distinction between a product and a service in the context of service design. Her seminal paper, “How to Design a Service”, published in 1982, opens with the following statement:

“The difference between products and services is more than semantic. Products are tangible objects that exist in both time and space; services consist solely of acts or process(es), and exist in time only. The basic distinction between “things” and “processes” is the starting point for a focused investigation of services. Services are rendered; products are possessed. Services cannot be possessed; they can only be experienced, created or participated in.”

(Shostack, 1982, p. 49)

The paper goes on to apply the analogy of molecular modelling to describing complex entities that may be made up of multiple services and products. Services are described as consisting of process components connected together. The concept of a Service Blueprint is introduced, essentially a process model representing the service to be delivered.

This idea may seem obvious today, as it has been adopted widely by the recent explosion of interest in business process modelling. Parallels also exist with object oriented programming, where a Class definition is a form of blueprint, and an Object of that Class is an actual instantiation of the blueprint, just as an instance of the service is an instantiation of the service blueprint. However, in 1982, this was new thinking. In addition to drawing upon molecular modelling as a useful analogy with which to describe the breakdown of services into process components, the author uses electricity, and its existence in both potential and actual forms to underline the distinction between a blueprint and an actual instance of the service.

In developing the idea of service blueprints, Ms Shostack acknowledges the strong connection between the nature of services and the nature of computer software systems:

“Since a service is basically a process, service blueprinting rests, as it must, on systems that have been developed to deal with processes, acts and flows. Three systems are relevant: time/motion methods engineering; PERT project programming; and computer systems and software design.” (Shostack, 1982, p. 56)

Service blueprinting is thus identified as a “close cousin” of systems and software design. The reason for this is made clear a little later in the paper:

“what happens in a computer is often analogous to what must happen in order for a service to be successfully rendered.”

(Shostack, 1982, p. 57)

A computer is a process-performing machine (software forming the processes that are performed by the computer). The software program can be seen as the equivalent of a blueprint that only becomes an instance when the program is loaded into memory and run by the computer's processor. A service is also a collection of processes. Therefore, in seeking methods for describing service designs, blueprints and instances, Ms Shostack has turned to the discipline of software program design, recognising these similarities.

Interestingly, Ms Shostack does not use the phrase “software engineering”, though it was coming into widespread use by the 1980s, as we have seen. She focuses on the word “design”, using it again in an article title in 1984, “Designing services that deliver” (Shostack, 1984), and she is seen today as the founder of service design as a distinct discipline. Having highlighted the clear distinction between the physical world of products and the intangible nature of services (Shostack, 1984, p. 133), it is perhaps understandable that Ms Shostack was not tempted to coin the phrase “Service Engineering”.

In highlighting the intangible nature of services compared with products, Ms Shostack has identified a clear characteristic that services have in common with software systems. This intangibility marks out both services and software systems from most other forms of design and engineering. These other forms tend to belong in the physical world.

There is not, as yet, agreement on the definition of 'Service Design', and many consider it to be an interdisciplinary approach combining methods and tools from a variety of disciplines, rather than a stand-alone academic discipline (Stickdorn and Schneider, 2014, p. 29; Moritz, 2005, p. 7). Some definitions focus on a human-centric view of the service and its design. For example, Mat Hunter, Chief Design Officer at the UK Design Council defines it as “the shaping of service experiences so that they really work for people” (Design Council, 2016b). Prof. Birgit Mager takes a wider view, for example including the service's effectiveness and efficiency as well as human aspects among service

design's aims (Koivisto and Miettinen, 2009, p. 34). Others focus more on the process of service design, with the model proposed by Moritz showing strong similarities to a typical software development life-cycle (Moritz, 2005, p. 158–9).

Service design practitioners have adopted further tools and methods from other aspects of software systems design, including user interface design and requirements analysis (Morelli, 2002; Tassi, 2009). Use case modelling was first developed to support systems design in 1986 by Ivar Jacobson (Jacobson, Booch and Rumbaugh, 1999), but it is now in widespread use among service design practitioners. Agile development is another method developed for software development that has been adopted in Service Design (Stickdorn and Schneider, 2014, p. 196–197). Agile development techniques emerged in software systems design and development in the 1990s, with Rapid Application Development (Martin, 1991), and a number of extensions and variants followed, including Extreme Programming (Beck, 1999), DSDM Atern (DSDM Consortium, 2008), and Scrum (Sutherland and Schwaber, 1995).

“Design Management” by Kathryn Best (Best, 2015) and “Designing Services with Innovative Methods” (Koivisto and Miettinen, 2009) both highlight the importance of prototyping and iteration in service design. These are important elements of agile software development methods, along with co-creation and the involvement of users in the design process, which is also emphasised in the UK Government's “Service Design Manual” (Government Digital Service, 2016a).

The discipline of software engineering has also learnt from the real-world example of services. Software components are designed to reflect their architectural component counterparts, and the terminology of services is adopted to describe these components and their behaviour towards “client” components that call upon their functionality. This “service oriented” approach to software design, which is being widely adopted by systems developers, further illustrates the similarities between the design disciplines applicable to both services and software.

The successful transfer of concepts between the domains of software and services is not, therefore, surprising given the similarities that have been identified. It is suggested that among the concepts worthy of consideration for the services domain, is that of architecture as applied to software. Although the term “Service Architecture” has yet to appear in service design literature, IBM has recently introduced the terms “Service Science” and “Service Science, Management and Engineering” (SSME) (Spohrer et al., 2007). They define SSME as:

“the application of scientific, management, and engineering disciplines to tasks that one organization (service provider) beneficially performs for and with another (service client).” (Spohrer et al., 2007, p. 71)

It would seem that the authors have in mind doing for services what the term “engineering” has done for software, a relationship they call particular attention to: “we see a strong relationship between the study of service systems and the more established study of computational systems”. Louise Downe, from the UK Government Digital Service, describes the hardest part of role of services designer in government as

“stitching them [the constituent transactions needed to form a given service] together into a coherent service that a user can use unaided.” (Government Digital Service, 2016d)

It is in this context that the concept of “Service Architecture” is suggested in this thesis, equivalent to the architectural discipline applied in software design.

In the light of this proposed adoption of the concept of “Service Architecture”, it is worth re-examining the definition of software architecture to see whether it might be amended to apply equally to services. Taking the IEEE definition cited earlier, just substituting the word “service” for the word “system” produces the following definition:

“The fundamental organization of a **service**, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.”

The Bass, Clements and Kazman definition can be similarly transformed, this time by substituting the words *service* or *process* for the words *software*, *program* and *computing system*:

“The architecture of a **service** is the structure or structures of the **service**, which comprise **process** components, the externally visible properties of those components, and the relationship among them.”

The implications highlighted by the Software Engineering Institute in the context of software architecture would appear to be equally valid in the context of Service Architecture: the architecture is an abstraction defining a set of elements; services comprise more than one element with critical relations between them. These are in essence the conclusions that have already been reached by Shostack, Morelli and other contributors to the field of Service Design such as Roberta Tassi (Tassi, 2009).

In the same way that well articulated systems architectures perform a valuable role in the development and maintenance life-cycles of software systems, so Service Architectures can fulfil the same role for services.

Application of the architecture concept to services could bring about benefits similar to those the concept has already brought to the software domain. The ISO list of thirteen uses of software architectures cited earlier could equally apply to uses of Service Architectures. And first among these is “analysis of alternative architectures”. Whilst the remainder of this chapter will focus on this particular application of Service Architectures, it is worth noting that some of the language normally associated with architectural practices is beginning to enter the vocabulary of service design. For example, design patterns were first described by Christopher Alexander in the 1970s, as applied to the design of public spaces, parks and recreational areas (Alexander, 1978). They were

adapted to the world of software in the 1980s (Beck and Cunningham, 1987; Gamma et al., 1994), and the language of 'patterns' is now emerging in the domain of service design, along with other architectural terms such as 'templates', 'components' and 'standards' (Government Digital Service, 2016c).

2.9 Service evaluation

The complexity of evaluating services has been the subject of much study and it is interesting to note that a common theme that emerges from much of the literature is the complexity that arises from the multi-dimensional impact of changing, or creating new, services. This section will focus mainly on the work of those exploring public service design and reform, but it is suggested that many of the issues highlighted in the public service arena are more widely applicable to other service industries.

In “Evaluating public management reforms” (Boyne et al., 2003), the authors highlight the difficulties of evaluating public services and the effects of change. These difficulties relate to the multi-dimensional nature of the criteria, and the authors point out that this is a problem shared with, and stemming from, the difficulty of evaluating organisational performance (Connolly, Conlon and Deutsch, 1980). This is further complicated by the perspectives that different stakeholders will bring.

The authors go on to consider two criteria that have been proposed by proponents of public choice reform: “efficiency” and “responsiveness”. These are each shown to be multi-dimensional with, for example, cost, quantity and quality all being potential dimensions of “efficiency”, and “responsiveness” being measurable from a variety of stakeholder perspectives (Boyne et al., 2003, p. 15–23). They identify a third multi-dimensional criterion for evaluating public service reforms, “equity”.

In *Excellence and Fairness* (Cabinet Office Strategy Unit, 2008), a paper that sets out the UK Government's approach to improving public services, four criteria are proposed for evaluating the extent to which public services can be

considered “world class”: “delivering excellent outcomes”, “offering personalised approaches”, “being fair and equitable” and “offering good value for money”. No formal definition is offered, and whilst these criteria are expressed differently from the three mentioned in the previous paragraph, the examples given in the paper show that they are similar multi-dimensional criteria, and that they share many of the same constituent dimensions.

Evert Vedung (Vedung, 1999) proposes a taxonomy of evaluation models that clearly illustrates the multi-dimensional nature of the problem facing service evaluators. Among its eleven evaluation models are models for goals, side-effects, stakeholder concerns, client concerns, productivity and efficiency measures.

2.10 Understanding stakeholders

In addition to the multi-dimensional criteria that might be used to judge a service reform, the service will have multiple stakeholders, each of whom may have a distinct perspective and different objectives for that service.

An example public service stakeholder model can be found in *Beyond Boundaries: Citizen-Centred Local Services for Wales* (Welsh Assembly Government, 2006) (commonly known as “the Beecham Review”), which examines the Citizen Model advocated in the Welsh Assembly Government's vision for public services, *Making the Connections* (Welsh Assembly Government, 2004). It includes the diagram given in Figure 1 below, representing the relationship of services to the public (Welsh Assembly Government, 2006, p. 4).

Taken together with those involved in commissioning, designing and delivering services (only some of which are reflected in the diagram), this is a starting point for creating a more complete and commonly accepted view of public service stakeholders. It is essentially the same set considered in “Evaluating public management reforms” (Boyne et al., 2003).



Figure 1 The relationship of services to the public

Interest in analysing and understanding stakeholders in a wide range of business contexts was boosted with the publication of “Strategic Management: A Stakeholder Approach” in 1984 (Freeman, 1984). Freeman defines a stakeholder as

“any group or individual who can affect, or is affected by, the achievement of a corporation's purpose” (Freeman, 2010, p. vi)

A range of different modelling approaches have been proposed to facilitate the analysis of stakeholders, their attitudes to a corporation, strategy or project, and to plan their involvement in that endeavour, or communication with them. Two of the most widely adopted models take the form of squares divided into four sections. Two attributes, each with binary values, determine which section a given stakeholder falls into.

For example, Gardner has proposed a Power / Dynamism model (Gardner, Rachlin and Sweeney, 1986). Each stakeholder classified according to their possession of the Power attribute (high or low) and the Dynamism attribute (high or low). This analysis can lead to greater understanding of the likely attitude of the stakeholder to a project or initiative. A stakeholder with high dynamism but low power might be considered to be unpredictable, but

manageable, given their limited influence over the project.

A slightly different model, though taking a similar form, was proposed by Mendelow (Mendelow, 1991). This looks at power or influence as one of the dimensions, and interest (in the endeavour) for the other. Stakeholder analysis has become a standard practice among project managers (Cleland, 1986; Achterkamp and Vos, 2008), and although the simple two-dimensional analysis models are most commonly used, some alternative approaches have emerged.

Alexander and Robertson propose an “onion model” on which to map stakeholders and their relationships (Alexander and Robertson, 2004). This gives a visual sense of the distance of each stakeholder from the 'core' of the endeavour, and allows connections between stakeholders to be represented. Mitchell, Agle and Wood retain two of the most common attributes, Power and Interest (which they call 'Urgency'), and add a third attribute, Legitimacy (Mitchell, Agle and Wood, 1997). This new dimension, which explores the cultural and political alignment of the stakeholder with the aims of the endeavour, provides an interesting new perspective. It complicates the model, as the addition of a third binary attribute results in seven classifications, compared with the four that are possible with two binary attributes. This added complexity allows for a new consideration to be explored: salience. This can be seen as somewhat similar to the proximity to the centre or core that is represented in the 'onion model' described earlier. The authors define salience as “the degree to which managers give priority to competing stakeholder claims” (Mitchell, Agle and Wood, 1997, p. 854). Three classes of salience are defined: definitive stakeholders are those that possess all three attributes, expectant stakeholders possess any two of the three attributes, and latent stakeholders possess just one of the three attributes. Figure 3 shows this model, and its associated classes and levels of salience.

In “Stakeholder Mapping as an Assessment Framework for Policy Implementation” (Mehrizi, Ghasemzadeh and Molas-Gallart, 2009), the authors extend the traditional approach to stakeholder analysis by relating the different

stakeholders to a set of policy goals in a Stakeholder-Goal Matrix. This is used to form the basis of an evaluation prior to the implementation of a policy (ex-ante policy evaluation).

Public services would appear to have a broader range of distinct stakeholders in comparison with those delivered by the private sector. For example, those who are accountable for the delivery of the service may be distinct from those who are responsible for its delivery, and those paying for the service (the source is often ultimately taxpayers' funds) are not necessarily the same as those who use or receive the service. However, in today's world of partnerships and outsourcing, many of these complexities are to be found in both the public and private sector, and so it is suggested that evaluations of all types of service need to consider multiple dimensional impacts from the perspectives of multiple stakeholders.

In the context of services, the term "evaluation" is somewhat controversial when applied to an examination of a service before its implementation. In *Public Policy and Program Evaluation*, the author insists that "evaluation is retrospective", and that "prospective appraisals (i.e., scrutinies of courses of action considered but not yet adopted even as prototypes), are not included in my definition" (Vedung, 1999, p. 7). He argues that to include prospective (ex-ante) assessments in evaluation would be to allow the concept of evaluation to "become too diluted".

However, he does acknowledge that some leading theorists argue that prospective assessment (ex-ante assessment) does belong to evaluation (Vedung, 1999, p. 7). Among them, Rossi, Lipsey and Freeman adopt a broader definition of "program evaluation". They include the design of the programme among the five domains that may be assessed as part of an evaluation (Rossi, Lipsey and Freeman, 2003).

More recently, Maffei et al. have provided a more comprehensive survey of the domain of service evaluation, using the life-cycle model from Moritz (Moritz, 2005) to produce an "Evaluation Research Matrix" that provides the basis for

examining the role of evaluation across the life-cycle (Maffei, Villari and Foglieni, 2013, p. 9). They go on to classify a number of techniques, such as Cost-Benefit Analysis and Customer Satisfaction Surveys, in the relevant cells of their matrix.

		DISCOVER ANALYSIS PHASE	DEFINE DESIGN PHASE	DEVELOP IMPLEMENTATION PHASE	DELIVER EXECUTION PHASE	CONSOLIDATE ADOPTION PHASE
PROVIDER	EVALUATOR					
	PURPOSE	1	2	3	4	5
	CRITERIA					
DESIGNER	EVALUATOR					
	PURPOSE	6	7	8	9	10
	CRITERIA					
USER	EVALUATOR					
	PURPOSE	11	12	13	14	15
	CRITERIA					

Figure 2 Service Evaluation Research Matrix (Maffei, Villari and Foglieni, 2013)

The type of ex-ante evaluation discussed in this thesis would fall into cells 2, 7 and 12 of the Service Evaluation Research Matrix. The authors conclude by describing their work on the evaluation research matrix as an “open issue”, and suggest that “the field of services is still a fragmented and controversial topic” (Maffei, Villari and Foglieni, 2013, p. 12).

2.11 Service quality model

We have seen how the use of quality models has been central to the adoption of ex-ante evaluation methods in the field of software systems. This section explores the adoption of quality models in the field of Services, and the extent to which these have contributed to service evaluation methods.

One service quality model that has been clearly defined and is in widespread use is *Servqual*. It was developed during the 1980s by Zeithaml, Parasuraman and Barry (Zeithaml, Parasuraman and Barry, 1990) and is sometimes referred to as RATER, an acronym formed from the model's five dimension; Reliability, Assurance, Tangibles, Empathy and Responsiveness.

Servqual's conceptual model highlights the identification and measurement of five gaps:

1. Not knowing what customers expect (the gap between management's perceptions of customer expectations and actual customer expectations)
2. The wrong service (the gap between management's perceptions of customer expectations and the service quality specifications of the service provider)
3. The service performance gap (the gap between the specifications and actual service delivery)
4. When promises do not match delivery (the gap between service delivery and communications from the provider to the customer)
5. Customers' assessment of service quality (the gap between customer expectations and their perception of what they actually receive)

The developers of Servqual conducted a series of focus-group interviews, and from the information gathered, identified "ten general criteria or dimensions" (Zeithaml, Parasuraman and Barry, 1990, p. 20). As the authors admit, "the ten dimensions....are not necessarily independent of one another. For instance, facets of credibility and security may indeed overlap somewhat." This exploratory and qualitative analysis was backed up with a subsequent quantitative phase of research. This resulted in a refinement of the dimensional model, collapsing the ten original dimensions into the five final dimensions of Servqual. Table 5 below shows the mapping between the original ten dimensions for evaluating service quality and the final five Servqual dimensions.

Original Dimensions for Evaluating Service Quality	Servqual Dimensions
Tangibles	Tangibles
Reliability	Reliability
Responsiveness	Responsiveness
Competence Courtesy Credibility Security	Assurance
Access Communication Understanding the Customer	Empathy

Table 5 Servqual Dimensions and the original ten Service Quality Dimensions adapted from (Zeithaml, Parasuraman and Barry, 1990, p. 25)

The focus of Servqual is on customer expectations and the ability of the service provider to meet them. Its authors define service quality as “a comparison between expectations and performance” by the customer (Parasuraman, Zeithaml and Berry, 1985, p. 42). This view that service quality is to be determined from the customers' perspective is not limited to the Servqual model. Robert Lewis and Bernard Booms, for example, define service quality as “a measure of how well the service level delivered matches customer expectations” (Lewis and Booms, 1983, p. 99–107). Lovelock and Wirtz state that quality “should be defined from a customer perspective” (Lovelock and Wirtz, 2007, p. 25). And, as Robert Johnston points out, many practitioners have taken customer satisfaction as a direct measure of service quality (Johnston, 1995).

Whilst Servqual was developed from research in private sector services, and has been widely used by commercial service providers, such as credit card providers, telephone companies, repair and maintenance providers and banks (Zeithaml, Parasuraman and Barry, 1990, p. 27), its potential application for

evaluating public services has also been recognised.

The Common Measurements Tool (CMT) was developed for the Canadian Government as a means of surveying client satisfaction with public services (Schmidt and Strickland, 1998). It builds on the work of the developers of Servqual, but adds some emphasis to the identification of what can, in the view of customers, be done to improve the service, and how these improvements should be prioritised. Like Servqual, CMT contains five dimensions, though they are labelled “dimensions of service delivery” not service quality, and the focus of CMT is explicitly on “client satisfaction”. CMT's five dimensions are responsiveness, reliability, access and facilities, communications and cost (where applicable). Although their emphasis is slightly different, the similarity of CMT's dimensions to those of Servqual is clear.

In the United Kingdom, a review conducted by MORI on behalf of the Office of Public Services Reform describes customer satisfaction as the key to service quality, and describes “quality factors” solely from the customers' perspective (MORI Social Research Institute, 2002, p. 23–4). The review recommends the use of CMT, which it says “appears to provide the most useful approach to identifying service factors that most need to be improved” (MORI Social Research Institute, 2002, p. 43).

This strong linkage between customer satisfaction and service quality goes some way to explain why so much of the literature and practice of service evaluation is focused on ex-post evaluation. It is, of course, not possible to establish customer satisfaction before implementation of a service (at least partial implementation, such as via a pilot), and if that measure is seen as being central to the definition of quality, then the focus of service evaluation will inevitably be on post implementation evaluation.

The roots of this linkage lie in the definitions of services, and the distinctions between services and products. As described above, many of the definitions of service reference the creation of value in the eyes of the customer. This implies a degree of co-creation or at least interaction between the service provider and

the service consumer, leading to the possibility that the same service may result in different levels of value among customers.

Yet this focus on a customer-centric view of services seems to exclude the presence and preferences of other stakeholders. As seen earlier, services, and especially public services, often have many stakeholders, and it is quite conceivable that the importance of quality characteristics may vary among the different stakeholders involved in a single service.

Furthermore, it is likely that the characteristics considered important by customers, and assessed as part of a customer satisfaction survey, will be a subset of the total range of quality characteristics that could be used to describe all aspects of the service. Table 6 below illustrates this by relating the ten dimensions of service quality originally identified by the Servqual authors (subsequently collapsed into five Servqual dimensions) to the twenty seven sub-characteristics of the ISO 9126-1 Software Quality Model shown earlier in Table 1.

As can be seen, there are substantial areas of the software quality model that are not covered by the Servqual model, and these predominantly relate to aspects of the service that will be considered important to stakeholders other than the service customer.

It is interesting to note that those sub-characteristics that are not represented among the Servqual dimensions are still highly relevant to a more holistic assessment of the quality of a service (i.e. they are not specific to software). For example, the testability of a new or changed service is of as much importance to the operator of a service, as the testability of a new or changed software application is to the operator of that system.

Table 6 only compares the ISO 9126 model with the dimensions of Servqual. Other customer satisfaction-centric models of service quality (Schmidt and Strickland, 1998; Johnston, 1995) focus on essentially the same characteristics, and therefore similarly lack the more comprehensive view of quality provided by

the ISO 9126 model.

ISO 9126-1 Quality Model		Servqual
Characteristic	Sub-characteristic	Dimension of Service Quality
Functionality	Suitability Accuracy Interoperability Security Functionality Compliance	Competence / Credibility / Access Reliability Security
Reliability	Maturity Fault Tolerance Recoverability Reliability Compliance	Reliability Reliability Reliability / Responsiveness
Usability	Understandability Learnability Operability Attractiveness Usability Compliance	Communication / Understanding Communication Competence Tangibles / Courtesy
Efficiency	Time Behaviour Resource Utilisation Efficiency Compliance	Responsiveness / Access
Maintainability	Analysability Changeability Stability Testability Maintainability Compliance	
Portability	Adaptability Installability Co-existence Replaceability Portability Compliance	

Table 6 Software Quality Model compared with the original dimensions of Service Quality

In a more general way, the studies of evaluating public services and public service reforms cited earlier achieve a broader coverage that is better aligned across the territory covered by the ISO 9126-1 Quality Model. “Efficiency”, “Responsiveness” and “Equity” (Boyne et al., 2003) are acknowledged by the authors to be multi-dimensional composites, measurable from a variety of stakeholder perspectives, and their decomposition has the potential to reach into the “Efficiency”, “Maintainability” and “Portability” characteristics of the ISO model that are so poorly represented in Servqual.

“Equity” is perhaps more difficult to classify. The authors conclude that it, too, is a composite attribute, citing the conceptual framework of LeGrand (Le Grand, 1982) as a means of breaking it down to its constituent attributes. If equity is a desirable characteristic in a given service, it is suggested that it, or its components, should be considered to be attributes of the “Suitability” sub-characteristic, since equity would be a functional requirement of the service.

The concept of equity as a quality attribute may be a key feature that distinguishes public services from those delivered by the private sector. However, consumers have become more aware of environmental and ethical issues surrounding the products and services they purchase in recent years, and so equity may increasingly feature as a desirable attribute of the Suitability sub-characteristic in both public and private sector services.

The ISO specification includes a brief definition for each characteristic and sub-characteristic (International Organization for Standardization, 1991, p. 7–11). Chapter 5 below reproduces these definitions with changes to the text so that the definitions apply to services instead of software. It is notable that all of these definitions, without exception, are relevant to the world of services, and that the vast majority of the words (83.5%) of the original definitions that were applied to software systems have remained in this new version of the definitions applying to services. Indeed, all but a few of the changes merely involved exchanging the word “service” for “software”, and “participants” for “users”.

It is therefore suggested that adoption of this model, of which the Servqual and similar customer satisfaction dimensions are just a subset, for the analysis and evaluation of services would enable this activity to treat customer satisfaction among the dimensions of service quality, but not as the sole determinant of it. Use of this model in service evaluation also opens up the possibility to consider service quality from the perspective of stakeholders other than the customer. In contrast, the Servqual dimensions, and those used by other service evaluation methods, may seem to be quite comprehensive from the customers’ perspective, but they cannot be viewed as a complete model of service quality.

2.12 Service Architecture and design reviews

It has already been suggested, in Section 2.8 above, that the concept of Service Architecture can fulfil a similar role for services that Software Architecture does for software. Furthermore, a primary function of architecture in this context is to facilitate the “analysis of alternative architectures” (International Organization for Standardization, 2007, p. 8) prior to the selection of a preferred architecture for implementation (ex-ante evaluation). We have seen how ATAM and some of the variants derived from it have become the most mature and widely used methods for evaluating software architectures. These methods are “attribute based”, and make use of quality models that describe the universe of software quality in a set of Quality Attributes or Quality Characteristics. Section 2.11 has also shown how an international standard model for software quality (International Organization for Standardization, 1991) has the potential for adoption as a Service Quality Model, and how its set of Quality Characteristics and Sub-characteristics are a superset of those commonly used to measure service quality.

Section 2.10 above has highlighted the complexity that arises in services, and especially public services, from the range of stakeholders involved in funding, creating, delivering and receiving those services. The value of Stakeholder Mapping (Mehrizi, Ghasemzadeh and Molas-Gallart, 2009) as a tool for ex-ante policy evaluation was explored, and this clearly has similar potential application for ex-ante service evaluation.

Bringing all of these elements together, it is suggested that we can answer the question put in Section 1.3 of Chapter 1 in the affirmative:

“It is possible to develop an 'attribute-based methodology', similar to that used to evaluate software architectures, with which to assess the potential impact of mass collaboration on the design and delivery of public services”

(and which can be used more generally to evaluate competing service

architectures in both public and private sectors).

Chapter 4 of this thesis describes the development of the Service Architecture Review Method (SARM) that grew out of this positive conclusion, and a full description of SARM is given in the user guide for the method, Chapter 5. This method is for use prior to the deployment of a service or change in service, and it assumes that for any given service more than one design, based on competing alternative service architectures, can be described and considered using the method. In this way, the method can be seen as complementary to most of the service evaluation methods discussed earlier in this chapter, which focus on evaluating services after they have been implemented (ex-post evaluation).

By introducing this form of ex-ante evaluation, it is possible that it will have an influence that reaches beyond service implementation. The analysis and evaluation of architectural approaches to delivering a new service prior to its implementation enables stakeholders to assess alternative service solutions in terms of the quality characteristics and outcomes they most highly value. The use of such an evaluation approach ex-ante will create a benchmark of expectations that can form the basis for any post implementation evaluation, while also improving the alignment of the service design to the intended programme outcomes (hopefully increasing the chances of a successful programme).

Chapter 3: Approaches and Methods

3.1 Introduction

This chapter examines the approach and methods applicable to this research project which explores the potential application of software architecture evaluation techniques to the domain of service design. Underpinning this research question are a number of implications:

- a) that the domains of software architecture and service design share enough characteristics between them to make the application of techniques in one domain applicable to the other;
- b) that this application, should it be achieved successfully, would be worthwhile, and that it would bring some real value to the domain of service design through the application of these techniques, and that similar or equivalent techniques are not already in use in that domain.

Though not necessarily implied by the research question, the value of the proposed research would be enhanced were it possible to demonstrate that the domain of software architecture has a closer affinity with the domain of service design than other domains that may also have evaluation techniques of potential applicability.

The approach and methods adopted to address this research question must therefore support the formulation of a theory that techniques frequently applied in the one domain can successfully be applied in the other.

The first part of the research project has therefore been necessarily theoretical, examining the two domains, establishing the applicability of the techniques from one to the other, and explicitly describing how these techniques would be applied in the new domain. In doing so, it became evident that some differences in the new domain needed to be reflected in some proposed changes to the established techniques from the originating domain.

It would have been somewhat less than satisfactory to finish the research

having only conducted theoretical work, without any practical demonstration of the applicability that is claimed in the theory. The value of some initial practical demonstration is not to attempt a statistical proof of the theory, but rather to gain new insights and knowledge into the effects of applying the techniques in a new domain for the first time.

Both domains involved in the research question, software architecture and service design, have people at their heart. And the techniques associated with evaluation exist to support design decisions made by people. A purely theoretical approach to the research would therefore have failed to explore the impact on the very people for whom the techniques exist.

In consequence, the second part of the research has been an initial study of the impact of applying these evaluation techniques in the new domain of service design. As already indicated, the approach adopted was not seeking to support a statistical proof of the theory, but rather to guide this initial practical application to draw out valuable insights into the impact of the techniques on the people involved in evaluating service designs. Such insights can provide fruitful ground for further future research, perhaps eventually leading to a wider appreciation and understanding of both the limitations and potential for applying software architecture evaluation techniques to the broad domain of service design. As will be seen in later chapters of this thesis, not only did the practical case study open up some future research directions, it also led directly to some further refinements and enhancements to the method.

The remainder of this chapter reflects this two-stage approach to addressing the research question. First we explore the methods applied to develop the theoretical aspect of the research, and later we look at the methods adopted for the practical application of the techniques in the new domain. There follows a more detailed examination of the approach and methods, including consideration of alternative approaches, explaining how the research approach finally adopted was arrived at. We then consider some of the issues and limitations that arise from this research approach, and the chapter concludes

with a look at the further research opportunities that may be spawned from this initial research.

3.2 Developing the theory

The idea of applying architecture review methods to service designs was first raised in “An approach to investigating the potential impact of mass collaboration on the design and delivery of public services” (Field, 2009a). The paper set out a proposal to explore mass collaboration as applied to public services, but identified a serious difficulty that would first need to be overcome: *how to measure the impact?* It cited the work of (Boyne et al., 2003), (Connolly, Conlon and Deutsch, 1980) and official documents from the UK Government (Cabinet Office Strategy Unit, 2008) to illustrate the multi-dimensional nature of services. The quality of a service cannot be measured by just a single value, such as cost or customer satisfaction, but needs to take account of many different dimensions, the relative importance of which may vary from service to service.

The paper goes on to add a further complexity to the problem of evaluating the impact of a particular approach to designing and delivering public services: *from whose perspective should the impact be measured?* Using publications from the Welsh Assembly Government as illustrations of the problem (Welsh Assembly Government, 2006; Welsh Assembly Government, 2004), the paper highlights the range of stakeholders who all have an interest in public services, and who may each have a somewhat different view of impact, success and failure.

The paper asks a simple key question:

“If the application of mass collaboration to the design or delivery of a public service can be viewed as an alternative service architecture, might it be possible to develop an “attribute-based methodology”, similar to that used to evaluate software architectures, with which to assess the potential impact of mass collaboration on the design and

delivery of public services?” (Field, 2009a, p. 5)

The application of such an “attribute-based methodology” to the domain of public services therefore becomes, in the view of the author, a pre-requisite to being able to properly investigate the impact of mass collaboration on the design and delivery of public services.

Further investigation of the two domains (software architecture and public services) is proposed (Field, 2009a, p. 15), though no estimate of the effort involved in this investigation is given. The paper acknowledges the likely need to adapt software architecture review methods, such as Architecture Trade-off Analysis Method (ATAM) (Kazman, Klein and Clements, 2000) for use to evaluate the impact of mass collaboration on the design and delivery of public services. It also goes on to suggest that:

“it [i.e. the adapted method] may be applicable to evaluating any set of possible service designs, irrespective of whether mass collaboration is involved”. (Field, 2009a, p. 16)

Further work to flesh out the methods needed to successfully deliver this proposed research project led to the realisation that the question considering the applicability of an attribute-based methodology to evaluate service designs was itself a substantial research question, worthy of detailed investigation. Indeed, the author would argue that it is not possible to satisfactorily explore the impact of mass collaboration without first establishing a sound method of evaluating service designs, taking into account their multi-dimensional, multi-stakeholder nature. The focus of the author's research therefore was redirected to answering this specific question:

Can software architecture review methods apply to service design?

The question suggests investigating the application of a method from one domain to another. The starting point for the investigation into this question is a more detailed exploration of each domain.

3.3 Exploring the domain of software architecture

An exploration of the domain of software architecture needs to address the following questions:

1. What is software architecture?
2. What are software architecture review methods?
3. How have they evolved?
4. What do the methods involve?
5. What are the characteristics of software architecture that make those methods particularly applicable to that domain?

Chapter 2 of this thesis has addressed these questions, establishing an understanding of the review methods under consideration, and the context in which they originated. It presents a detailed investigation of the history of software architecture, its origins and the development over time of software architecture review methods.

Computer software is itself a relatively recent concept, electronic computers only having existed since the 1940s. The history of computer software is therefore well documented, and a literature review facilitates a good understanding of how the concepts of software architecture, which has its roots in the 1970s, only became fully articulated in the 1990s. Chapter 2 has examined this in some detail, describing the evolution of software architecture to the point where various aspects of software architecture have been enshrined in international standards (International Organization for Standardization, 1991; International Organization for Standardization, 1999; International Organization for Standardization, 2007).

The review of the relevant literature from this domain set out in Chapter 2 therefore achieves the first step towards developing a theory regarding the

potential application of software architecture review methods to the different domain of service design.

3.4 Understanding service design

Having understood the domain of software architecture, the next step is to understand the world of service design, and explore the similarities and differences between this domain and that of software architecture. This can be broken down into the following additional questions:

1. What are services?
2. What is service design?
3. How has service design evolved?
4. What are the similarities and differences between the domains of service design and software architecture?
5. What ex-ante evaluation methods are applied to services and service design?

Once again a review of relevant literature can help address these questions, and this has been explored in detail in the second part of Chapter 2 of this thesis. The findings may be summarised as follows: whilst the distinct characteristics of services (as opposed to physical products) were being articulated as early as the 18th century (Smith, 1776), definitions of “a service” are still being hotly debated in the 21st century e.g. (Lovelock and Gummesson, 2004; Lovelock and Wirtz, 2007; IBM Research, 2010). Service design, mocked at the time of its first introduction as an academic field in the early 1990s (Erlhoff and Marshall, 2007, p. 354), “is still a very young discipline that contains many exciting, undiscovered lines of research” (Erlhoff and Marshall, 2007, p. 357).

The fact that this field is still in an early evolutionary stage means that the available literature base is inevitably more sparse, and new methods and techniques continue to emerge in both academic and commercial environments.

“Can software architecture review methods apply to service design?” (Field, 2010a) attempted to capture the state of service design at the time of its publication.

By understanding the relationship between the two domains, and their respective ex-ante evaluation methods, the project was able to develop a good understanding of the potential application of software architecture review methods to the field of service design. An analysis of this information has allowed the most suitable method from among the different software architecture review methods to be selected. ATAM has become a dominant formal method for evaluating software architectures, spawning a number of variants and enhancements with the growth of practical experience. A particular variant of ATAM is highlighted in “Can software architecture review methods apply to service design?” (Field, 2010a, p. 114) as being suited for adaptation to services, and especially to public services, on account of its focus on risk (as opposed to utility).

Chapter 2 has therefore seen the development of a theory based on an analysis of the literature relating to two domains: software architecture and service design. It is that the review methods that have evolved to evaluate software architectures prior to their implementations might be applicable to service designs. An insight into the potential benefits of such an application was also gained, following a comparison of the characteristics of software architecture review methods with service design evaluation methods.

Chapter 2, with its detailed comparison of the two domains, and the evaluation techniques applied to both, has led to the selection of a preferred software architecture review method to serve as the starting point for the development of a method suited to ex-ante evaluation of service designs. The knowledge gained from this analysis has also led to a proposed enhancement to the method prior to its application in the domain of service design. The consideration of multiple stakeholder perspectives is a key feature of services, especially public services. This is an aspect that has been absent from

software architecture review methods, but it cannot be ignored if one of these methods is to be adapted for evaluation of service designs. An approach to capturing the relative importance of quality characteristics to a service's various stakeholders is proposed that can then be used to create a set of stakeholder specific trade-off analyses when considering competing service designs (Field, 2010a, p. 120).

Chapter 2 also examines the history of software quality models, leading to the development of international standards and, by drawing on the close comparison of the domains of software and services, shows how the adoption of the ISO 9126-1 Quality Model (International Organization for Standardization, 1991) is not only appropriate for evaluating service designs, but brings more comprehensive coverage than models that have historically be used for evaluating services (Field, 2010a, p. 119). This is the quality model that has been incorporated within the proposed service architecture review method, as described in Chapter 5.

3.5 Why software?

There is one further question that this project attempts to address:

Why choose the domain of software architecture from which to select a review method, and not some other “architectural” discipline?

To answer this question in its entirety it would be necessary to explore all such “architectural” disciplines and compare them with the domain of service design. It may even be necessary to conduct practical experiments on all of them. This is clearly beyond the scope of a single research project. Indeed, it is not strictly necessary to address this question to justify the exploration of methods from just one domain, software architecture, and their applicability to another, service design.

However, it is clearly an interesting and relevant question, and whilst it may not be possible for one project to come to a comprehensive answer, the literature

analysis of the two domains can provide insights and pointers towards a possible answer. The question is particularly interesting for this project given the adoption of the term “architecture” by the software domain from its more engineering-oriented historical origins. Why not choose an evaluation method from structural engineering, a discipline that has a history stretching back thousands of years, and whose methods are likely therefore to have much greater maturity than those of computer software development, which can only boast 70 years of evolution?

The association of the words “software” and “engineering”, and therefore by implication the association of software development to the disciplines of the physical sciences, is somewhat controversial, though the combined term “software engineering” has fallen into common usage since its formal introduction in 1968 (Randell, 1996). The literature analysis in Chapter 2 highlights this controversy, and once again questions the validity of the term “engineering” in the context of computer software design and development. The review of the history of service design in Chapter 2 highlights the intangible nature of services, a characteristic that the services domain shares with the software domain, but distinguishes it from other scientific and engineering domains. This was not lost on one of founders of the discipline of service design, G. Lynn Shostack (Shostack, 1982; Shostack, 1984), nor on more recent academics and practitioners e.g. (Morelli, 2002; Tassi, 2009; IBM Research, 2010).

Whilst it is beyond the scope of this project to manage a comprehensive analysis of all “architectural” domains in comparison with that of service design, the analysis of literature from the domains of software and services, as set out in Chapter 2 of this thesis, does seem to support the suggestion that these domains share some key characteristics that are not present in other scientific and engineering domains.

We have seen how the study of literature in Chapter 2, and its analysis of published methods, approaches and models, has led to the formulation of a

proposition that a method developed and used in one domain can be applied in another, and that this application may bring particular benefits. This analysis has also helped specify adaptations of that method so that it is better suited to application in the new domain. Such a proposition remains entirely theoretical until it is applied in the real world, and in this particular project, this means taking a real service design situation and adopting and applying the proposed service architecture review method. The next section describes the methodological approach taken by this research project to do just that.

3.6 Discussion of the approach and methods

The introduction to this chapter draws attention to the two-stage nature of this research project: the theoretical stage, in which a theoretical proposition is developed concerning the application of software architecture evaluation techniques to the domain of service design; and the practical stage, in which an adaptation from a leading software architecture evaluation technique is to be applied to a real service design situation with a view to studying the consequential effects.

As described earlier in this chapter, the idea for this research sprang from the development of another research question. Turning this idea into a more concrete research question has demanded a study of the history of the two domains in question, and this study has in turn helped to shape the research question, and inform the development of a theoretical proposition. To give that theoretical proposition a more thorough grounding, a proposed method, the Service Architecture Review Method (SARM), has been developed by adapting the Architecture Trade-off Analysis Method, which has become the dominant evaluation technique in the software architecture domain. The development of SARM, and the design choices made to make it suitable for the evaluation of service designs, are described in more detail in Chapter 4, Creation and Development of the Service Architecture Review Method (SARM).

The adaptations themselves have been grounded in the results of a comparison between the two domains, software architecture and service design. So whilst

the original idea grew out of a thought that occurred during consideration of another research question, the formulation of this research question, and its accompanying theoretical proposition, has a solid grounding in a study of the relevant literature. The resulting theoretical proposition involves, by definition, novelty. It proposes the application of an existing technique (software architecture evaluation) in a context (service design) that is alien to that technique. The combination of technique and context is novel, and therefore, prior to this research project, untested.

The exploratory nature of the research question, implying as it does a desire to test the theoretical proposition, might be suggestive of an experimental or quasi-experimental research approach. This is, in effect, an exploratory “what” question (Hedrick, Bickman and Rog, 1993). However, the effect that is being explored is the impact of a method on people, a project and an organisation. An experimental approach demands control or manipulation of behaviours in a direct and systematic manner. In this way the experiment can focus on just one or two isolated variables at a time. Such an approach is not suitable for this research, where behaviour cannot be controlled, nor could it be simulated with the adoption of a quasi-behavioural approach, since there are too many unique features associated with the individual context of a project and its participants to enable the isolation of small numbers of variables across multiple observations (Campbell and Stanley, 1966).

3.7 Adopting a Case Study approach

Yin suggests considering a variety of research methods, and offers some guidance regarding method selection on the basis of three key conditions: the form of the research question, the extent of control over behavioural events, and the degree of focus on contemporary as opposed to historical events (Yin, 2008, p. 8). He suggests considering Experiment, Survey, Archival Analysis, History and Case Study. As already discussed, Experiment is not considered suitable on account of the inability of this research to control behavioural events. Archival Analysis and History can also be ruled out, given that the

project sets out to study the application of a method for the first time in an unfamiliar domain. There cannot therefore be existing archival or historical documentation regarding something that has yet to be tried.

This leaves just two potentially viable methods: Survey and Case Study. Whilst one could conceive a survey of service designers, or organisations that design or deliver services, their ability to respond to questions about the possible application of evaluation techniques with which they are likely to be, as yet, unfamiliar, will be severely constrained. A survey could, at best, only scratch the surface regarding the possible need for improved ex-ante evaluation techniques, and would be unable to provide much valuable insight into the potential impact of applying software architecture evaluation techniques in the service design domain.

This leaves a Case Study approach. But it would be somewhat unsatisfactory if the research method adopted was to be chosen by default, as a result of being “the last man standing”. We should therefore look at the “logic of design” (Platt, 1992) that would lead this research naturally to adopt a case study approach. Yin proposes a twofold, technical definition of case studies:

1. A case study is an empirical enquiry that
 - investigates a contemporary phenomenon in depth and within its real-life context, especially when
 - the boundaries between the phenomenon and context are not clearly evident.
2. The case study inquiry
 - copes with the technically distinctive situation in which there will be many more variables of interest than data points, and as one result
 - relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result
 - benefits from the prior development of theoretical propositions to guide data collection and analysis.

(Yin, 2008, p. 18)

Here we can see the very strong fit between the research involved in this project and the case study research method. With a first use of the proposed new evaluation method, SARM, we have the possibility of an in-depth investigation of “a contemporary phenomenon...within its real-life context”. Practical use of SARM will also enable the research to draw upon multiple sources of evidence, including documentation, interviews, direct observation and possibly video or voice recordings. The first part of this two-part research project has consisted of the development of a theoretical proposition that can be tested in the second part. There is therefore a clear and positive reason for choosing a case study approach, in addition to the reasons given earlier for rejecting other approaches.

3.8 A first use of SARM

The Service Architecture Review Method (SARM), as described in Chapter 5 of this thesis, was derived from a variant of the Architecture Trade-off Analysis Method, but in arriving at a method suited to the evaluation of service designs, a significant addition to the method has been introduced, in the form of service stakeholder perspectives. Its development is described in Chapter 4, Creation and Development of the Service Architecture Review Method (SARM), and the final version of the method that emerged at the conclusion of this research is described in full in the user guide in Chapter 5, Service Architecture Review Method.

A first use of this new evaluation method, described in Chapter 6, has provided an opportunity to test out the method in a real service design situation. And, as will be seen, the understanding that was derived from its use has led to further improvements in the method. But the application of the method in a real situation is also an opportunity to conduct further research. As indicated in the introduction to this chapter, service design is a human activity, and the service design decisions are made by people. The first application of the method represents not just an opportunity to explore a single service design, but an opportunity to explore the key design decisions and the effects of the method

from the perspectives of the different members of the design team. Are their experiences and stories similar or different?

3.9 Maximising the research value

As already indicated, a live trial of the method, which is described in detail in Chapter 5, offers an attractive opportunity to study the effects of the method. But on what? Opportunities exist at a number of different levels.

At the organisational level, one can seek to learn about the effect on organisational decision making, on organisational culture, and how the introduction of a new method disrupts or is accommodated by the organisation's existing processes.

At the project level, how does the method cope with the specific service design task in question? How does the project cope with the introduction of a new ex-ante evaluation method?

And at the individual team member or stakeholder level, how does participation in the method's workshops affect their design decisions or recommendations? Does it have an influence on their attitude to the project? Does it affect their understanding of the project, or of their fellow participants and stakeholders?

A first, single, live trial of a new method is clearly not able to yield generalisable results. It can only provide insights into the effects on the individual and specific case that is selected. But it can highlight aspects and effects in that one project that might be worthy of wider study in a set of subsequent applications of the method, in more projects, or across multiple organisations. And at the individual team member or stakeholder level, the one project will yield multiple units of analysis, in the form of the different experiences of the individuals involved. Their individual experiences are worthy of study, and an analysis of what they have in common or experience differently has provided a pointer to potential future research with other types of project, project teams or organisations.

As described above, the research acts on a number of different levels: individual participants, the service design project and the organisation. The single case therefore has multiple units of analysis: the participating individuals plus the project plus the organisation that owns the service design project. The project's case study design therefore corresponds to an embedded, single-case design according to Yin's classification of case study designs (Yin, 2008, p. 46–60). This allows the project to explore more completely the impact of applying SARM for the first time in a service design project from a variety of perspectives.

As is common in case study designs, interviews are among the most important sources of information for this project. In interviewing different participants who all took part in the same workshop or workshops, a focused interview approach was adopted (Merton, Fiske and Kendall, 1990). Whilst the interviews were open-ended, and conversational in style, they followed a certain set of key questions. This common pattern has facilitated some comparison of the experiences of the different individuals who all participated in common and shared activities.

As the lead researcher conducted the interviews, it is inevitable that the resulting interviews were as much a reflection of the interaction between interviewer and interviewee as a record of the views and experiences of the interviewee, especially since both were involved in the SARM process and its associated workshops. Whilst some might see this as a source of unacceptable bias in the research, it is suggested that this should instead be seen as a strength. All interviews are inevitably coloured by the nature of the interaction between interviewer and interviewee (Hammersley and Atkinson, 2007, p. 101), and the fact that both interviewer and interviewee experienced attendance at the same events should give rise to a more in-depth conversation, with consequently greater insights. The resulting interviews can therefore be seen as “active interviews” as defined by Holstein and Gubrium (Holstein and Gubrium, 1995, p. 16).

3.10 Adopting a single case design

The aim of the research has been to begin to understand the implications of applying an evaluation method drawn from the software architecture domain to the service domain - a domain that has not adopted similar methods to date. Is a single case satisfactory? Each service design project has its own, unique set of characteristics. It belongs to a particular organisational culture, involves a specific set of stakeholders, and happens in a social, cultural and legal environment that is distinct. Other services, delivered by other organisations, are inevitably very different.

The first adoption of the method in a service design context can clearly provide considerable insight into the impact of the new method on the organisation, the project, and the project team members. The information generated as a consequence of collecting and analysing evidence collected from documents, observations, interviews and other artefacts created during the project, allows conclusions to be drawn about the successes, failures, disruptions and new knowledge and experiences gained at each of these levels. These insights are also likely to help identify potential improvements to the new method that might subsequently be introduced before the method is applied to another project, either in the same, or a different, organisation.

The single case study will not directly inform the extent to which the method can produce the same impacts, benefits, disruptions and the like, in another project, or another organisation. It is just the “first toe in the water”, and whilst the experiences gained from it may encourage others to attempt similar trials or adoptions of the method, the inevitable differences associated with their people, their projects, their organisation and its culture, and the distinct nature of their business, in contrast with those involved in this initial research project, means that their experiences will inevitably be different.

This does not mean that the single case studied in this research cannot inform the theoretical proposition or propositions that were developed in the first part of the research, and that initiated the second part. The case should not be

confused with a sampled population of one, in the statistical sense, and any generalisation relates to the theoretical proposition, and not to populations or universes. This form of generalisation has been referred to as “analytic generalisation” to distinguish it clearly from “statistical generalisation” (Yin, 2008). As Lipset, Trow and Coleman showed in 1956, it is possible to conduct a generalising, as opposed to particularising, analysis of a single case study (Platt, 1992).

Further, given the novelty involved in this research, applying a new method for the first time in a given domain, it is highly likely that a range of further research possibilities and avenues of enquiry will be spawned by this first case study. Simply pushing ahead with a second, third or more cases, without first taking stock of the breadth of knowledge gained from its first application, would risk failing to maximise the value of the evidence gathered from this initial research.

It is therefore suggested that completion of the first adoption of the method, and a full analysis of the evidence gained from that experience at a number of different levels, is a suitable point at which to draw the initial research project to a close. An anticipated outcome of this research has been to point the way for future research projects to expand upon this initial case, perhaps replicating or contradicting the results obtained, and thereby leading to an even greater understanding of the application of software architecture evaluation techniques to the domain of service design.

3.11 Selection criteria for the case study

Earlier parts of this chapter have set out the research design for this project, dealing with the four key problems: what questions to study, what data are relevant, what data to collect, and how to analyse the results (Philliber, Schwab and Sloss, 1980). A single case study approach has been proposed, and we now turn to the criteria for selecting a suitable service design project for that case study, which will involve the first use of SARM.

“Can Software Architecture Review Methods Apply to Service Design?”

highlights the need for SARM to take account of the range of stakeholders involved in a service (Field, 2010a, p. 120), and this is reflected in the introduction of an innovative extension to the ATAM approach that is being adapted from the software architecture domain. The paper also suggests that “public services would appear to have a broader range of distinct stakeholders in comparison with those delivered by the private sector” (Field, 2010a, p. 116). Although the analysis that has led to the development of SARM has not been constrained to the sub-domain of public services, selection of a public service design project would enable this new extension to ATAM to be more fully tested.

In addition to this preference for a public service design project, it is also desirable that the service involved is not heavily dependent on information technology (IT). With the widespread creation of digital services, there is a substantial overlap between IT projects and service design projects, especially where the services can be delivered digitally, for example via a web site or mobile phone application. The aim of this research is to explore the transfer of an evaluation method from the domain of software architecture to the domain of service design. This transfer will be more clearly tested if the service design project is not, at its heart, essentially a software development project. In such a case, it might be argued that the method is continuing to be applied in its originating domain of software architecture. Adoption of a design project for a service that is delivered through human, rather than computer interaction, would therefore more clearly test the method's applicability to service design projects.

3.12 Choosing the case study

The leadership of this research from within a public sector organisation led to a number of candidate public sector projects, not just from within that one organisation, the Office for National Statistics (Office for National Statistics, 2010), but also from other central and local government organisations.

Adoption of a project at the Office for National Statistics (ONS) would seem to be the logical choice, given the familiarity of that organisation with the ATAM approach for software architecture reviews. However, that familiarity is itself a

potential issue. If some of the stakeholders involved in the service design have previously encountered ATAM reviews for software projects, their previous experience of ATAM may colour their approach and attitude to SARM, which carries so many similar features. This issue could be further heightened by the involvement of the author, who is leading the research, since in his capacity as the ONS Chief Technology Officer, he has been regularly involved in the organisation's software architecture reviews.

And whilst this potential issue may itself be worthy of study, it is suggested that the first use of SARM is not the appropriate place for such a study, and given the researcher's involvement in the organisation, and its use of ATAM, this could not be considered a "typical" instance of the issue.

Another reason for not selecting an ONS project is the nature of services at ONS. Whilst ONS can be seen as delivering services to its customers as part of the wider public sector in the UK, the nature of those services is rather less than typical of either public services in particular, or services more generally. The ONS mission is "to improve understanding of life in the United Kingdom and enable informed decisions through trusted, relevant, and independent statistics and analysis", and its main way of delivering that service is through the free publication of data and accompanying analysis.

This activity does fit within the definition of a service preferred in Chapter 2, namely "a provider/client interaction that creates value" (IBM Research, 2010). However, a large part of the service consists of publication activities, and the degree of "provider/client interaction" involved in the act of publishing on the part of the service provider, and consuming publications on the part of the service consumer, is minimal.

The UK Border Agency (UKBA) "is responsible for securing the UK border and controlling migration in the UK" (UK Border Agency, 2010b). Many of the services it provides necessarily involve "provider/client interaction", and one such is the consideration of applications for asylum. Asylum is "protection given by a country to someone who is fleeing persecution in their own country. It is

given under the 1951 United Nations Convention Relating to the Status of Refugees” (UK Border Agency, 2010a).

UKBA aims to conclude all new asylum applications within six months, and to help achieve this, it embarked in 2010 on a number of service design projects to re-examine different parts of the overall process. One of these involved decisions surrounding cases that go to appeal. The process of dealing with appeals in court can be lengthy, so the project to re-design the appeals process part of the service was seen as key to helping the Agency achieve its “six months aim”. This was the service design project that was chosen to be the first one to apply SARM, an account of which is given in detail in Chapter 6.

3.13 Collecting evidence at other levels

The case study provided a clear opportunity to gain evidence as the project progressed, at the individual project level, and how this was done is described in the account of the case study given in Chapter 6. As identified earlier, this research has the potential to gather information beyond the project level, by understanding the effect of the method on individual participants, and also at the level of the organisation (the UK Border Agency).

To achieve this, the lead researcher followed up the application of SARM with interviews of each of the participants. Eight interviews were conducted, yielding a substantial quantity of evidence at the individual level for analysis. These interviews took place within a month of the concluding workshop to ensure that the experiences remained fresh in the minds of the interviewees.

Each interview lasted up to an hour, and was conducted in private by the lead researcher with each individual in a private meeting room at the Cardiff offices of UK Border Agency. An audio recording was made, following the granting of explicit permission by each interviewee, allowing the interviewer to focus on interacting with the interviewee. The recordings were subsequently transcribed in full by the lead researcher, facilitating their analysis using a qualitative analysis software tool.

3.14 Analysing the evidence

As already indicated, this research has been able to draw on a variety of sources from the single application of SARM at the UK Border Agency. The interviews of participants provide a rich source of semi-structured information for analysis using a thematic analysis approach (Braun and Clarke, 2006). A narrative element is revealed by these interviews, in the form of the personal journey each participant undertook as a consequence of participating in the two workshops, and in some cases, working on new service designs. It became apparent that the application of the method itself helped to inform participants and affected their understanding of the project and the design decisions they took.

This information has been supplemented by the observation evidence from the two workshops, which may inform both the personal experiences of the participants, and also the collective experience of the project team as a whole. This observation-based evidence is itself supported by the documentation produced during and in between the workshops. These document the collective views, judgements and expert opinions of the participants, and represent the outcomes of the deliberations that were recorded in observational reports following the workshops.

The analysis phase of the research has remained open to exploring a variety of effects: the effect of the method on individual participants, the effect across the whole team and on the project as a whole, and the effect on the organisation.

As this was the first use of the method in the domain of service design, the evidence gathered has also been able to identify some ways in which the domain has affected the method (compared with that normally experienced when architecture reviews are conducted on software architectures). This resulted in some further recommended amendments to SARM. These are summarised in Chapter 4, Section 4.6, Subsequent enhancements.

3.15 Studying design decisions

The trade-off analysis conducted in the second workshop as part of the SARM process involves the creation of a number of matrices that explore risk exposure, and how it varies by scenario, quality characteristic and stakeholder across different solution options. These were described in some detail in “Can software architecture review methods apply to service design?” (Field, 2010a), and in Chapter 5 of this thesis. It is easy to be drawn into thinking that these matrices, and therefore the method, aim to identify “the right decision”. But as that earlier paper clearly points out:

“The analysis helps the stakeholders understand how the balance of risks lie across the scenario / solution landscape. The purpose of the matrix is not to inform the review team which solution approach they should adopt, but to provide them with better information upon which they can base their decision. This is a decision support, not a decision making, tool.” (Field, 2010a, p. 122)

The focus of the method is on improving understanding, both individual and collective, so that service design decisions can be made using a rich and broad base of information, and taking into account a wide range of perspectives. The research is therefore focused on exploring whether the method does indeed achieve its aims of generating that wider knowledge and understanding, and if so, how that is used in the context of a service design project. Neither the method, nor the research, are directly involved in assessing in any way the “quality” of design decisions that are made.

Indeed the nature of the method itself highlights the fact that there is not, typically, a single “best” design. “Best” will tend to depend on an individual perspective, whether that is from a particular stakeholder, quality characteristic or scenario. A change in the perspective will often lead to a change in preferred design, and whilst the method can help the design team understand and manage the risks associated with these changing trade-offs, a final design decision will always need to be taken by humans exercising their judgements. It

is therefore the effect of the method on the humans that take design decisions that this research must study, and not the decisions themselves.

3.16 Ethical issues

There are potential ethical issues that arise from the choice of case study organisation, and its services. The asylum system in the UK has come under considerable pressure from the large numbers of applications in recent years. The process has become a sensitive political issue, and it deals with vulnerable people and sensitive and personal information. This could raise a number of ethical issues that may inhibit the research and the ability of the fruits of the research to be openly published.

Fortunately, the research plan does not need to examine individual asylum cases, and there is no requirement for details of individual cases, or indeed the members of the design team, to be revealed or exposed during the research. The focus is on the service, how it functions, and its accompanying business processes, and not on individual cases. In the language of service design, the project is exploring how an organisation develops its service blueprints, and not examining individual service instances.

Nevertheless, the details of the process, the decisions surrounding appeals, and the individual civil servants involved, all demand confidential treatment, and the project's ethical statement aims to ensure that sensitive information that needs to remain confidential does indeed remain confidential without compromising the research value of the project.

“All material relating to interviews conducted during the course of this research (recordings, transcripts and notes) will be held securely under the control of the researcher, and will only be used for the purposes of research or the reporting of that research. All interviewees are offered the right to request access to the recordings, transcripts and notes relating to their individual contribution to the research. This source material will otherwise only be accessed by

the researcher and those assessing the research. Interview respondents will be anonymised in the event that any interview content is included in papers, presentations or reports intended for publication unless permission to identify the interviewee has been given explicitly by that interviewee.” (Project Ethical Statement)

This statement was strictly adhered to. As will be described in Chapter 6, none of the project meetings were recorded (neither audio nor video), and no individual asylum cases were referenced. Each member of the UK Border Agency team that participated in the case study was subsequently interviewed, and all interviews were audio recorded, to enable an accurate transcript of each interview to be created. Permission to record the interviews was sought and obtained from each participant, and it was agreed that further permission would be sought in the event that the lead researcher should wish to publish an attributable quotation.

The audio recordings of the interviews were only accessed by the lead researcher, who undertook the task of transcribing the interviews personally. This helped to ensure confidentiality of data, while also keeping control over the quality of the transcripts. A number of journal and conference papers were published by the lead researcher, in addition to this thesis, but none of these involved named reference to any of the individuals involved in the case study other than the lead researcher himself.

Interviewees have been identified in this thesis with the letter 'I' followed by a number. This allows the reader to track the various quotations of each individual while maintaining their anonymity. The only exceptions to this are quotations from the Project Leader, where these quotations clearly identify the speaker in that role. So the Project Leader's quotations are sometimes referenced 'Project Leader', and sometimes as an anonymous numbered interviewee.

3.17 Other potential issues

A number of other potential issues have been discussed throughout this chapter, and this section aims to draw them together and highlight the way in which they can be resolved.

One of the most obvious potential issues that runs throughout the project is the heavy involvement of the lead researcher (the author) in the activities that are being studied. Given the novelty of the research, it is inevitable that the creator of the intellectual property that is being explored must be involved in its exploration, and in its study and reporting. This does indeed introduce a necessary bias, but this is not statistical analysis (despite the organisation that is co-sponsoring the research!), and as a study of the very first use of a new technique, the nature of generalisation that might be possible in relation to the theoretical proposition is extremely limited. It will be for others in future studies to build on this work and gather a broader evidence base from which more worthwhile generalisations can be drawn, if they consider it a potentially fruitful avenue worth pursuing. This “bias” is therefore one from which the research should profit, as it will deliver greater insights, drawing as it does on the observations and insights of the creator of the intellectual property that forms the foundations of this research.

It is possible for the lead researcher to become so involved in the activities that his role as observer may become compromised, endangering the quality of the evidence being collected. The project has mitigated this risk with the involvement of a second expert to facilitate the case study workshops, freeing the lead researcher to adopt a primary role of observer.

The concern that the decision to focus on a single case study limits the ability of the research results to be generalised has been discussed at length, and the advantages of taking stock after the first application of the method to a service design project have been presented. It is suggested that a number of new research questions may arise from this first application, and it is already possible to envisage some of them in relation to generalisation.

For example, the project has selected an adaptation of the ATAM approach as the method that will be further adapted for use to evaluate service designs. The reasons for choosing ATAM have been articulated, but what would have been the results had a different software architecture evaluation technique been chosen?

A project from the UK Border Agency was selected. How would the results have differed had a different UKBA project been chosen? Or a different public service? Or a commercial service from the private sector? These questions go far beyond the scope of this research project, but they illustrate the different directions that a search for generalisation might take in the future.

The choice of the UK Border Agency does help to mitigate another risk: that participants that have previously encountered ATAM might bring with them an IT software-coloured view of the process that then gets in the way of the project's goal, which is to take the technique from the domain of software architecture, but to apply it specifically to service design (shaking off entirely its IT software origins).

None of those involved in the selected service design project at UKBA has had experience of ATAM, and the selected service design project does not contain any IT elements - it is very much the provision of a service as a business process that is people- and document-driven, and not IT-driven. This allowed the service design participants to come to this process in a completely fresh frame of mind - SARM was simply seen as a new way of exploring service design, and its antecedence from the world of software architecture should have had little or no impact on the team. As pointed out elsewhere in this chapter, this might not have been the case had the selected service design project been one owned by ONS.

Finally, there is a substantial area of potential research around the way in which teams reach a consensus view regarding risk, and the ways in which individuals perceive risk. SARM adopts an established method of quantifying risk that involves the division of risk into two constituent components: impact and

likelihood. Both of these are quantified numerically, with the product representing overall risk exposure. But do different individuals see risk in a similar way? How consistently do individuals respond to demands to quantify these two risk components? Consistency within a single evaluation will tend to emerge from the group discussion, and the generation of all the risk estimates by the team together in the review workshops. But this question of consistency will become a more serious issue if one were to explore the relationships between more than one evaluation. There are some interesting avenues of research here that are clearly beyond the scope of this initial research project, but this illustrates the wide and rich range of future research that can follow published accounts of the first application of SARM to a service design project.

3.18 Conclusions

This chapter has described a research project that aimed to explore the application of software architecture techniques to the domain of service design. In particular, the chapter has explored the research approach that was adopted, and examined in some detail the two stages to the project: the first one being the development of a theoretical proposition, and the second being the adoption of a case study approach to gathering and analysing evidence from multiple sources to explore that proposition in a real-life context.

The chapter has sought to establish a firm basis for the research approach and methods being adopted, examining alternatives, identifying risks and issues and articulating the ways in which those risks may be mitigated, and issues dealt with. It is inevitable that some issues have remained outstanding, and these present opportunities for further research that may be initiated following the completion of this project, and the chapter has begun to illuminate the future research landscape that is likely to emerge following the successful completion of this project. These are explored in more detail in the concluding chapter of this thesis.

Chapter 4: Creation and Development of the Service Architecture Review Method (SARM)

4.1 Introduction

This chapter describes the development of the Service Architecture Review Method (SARM). Chapter 2 has explored the relationship between the domains of software architecture and service design, and tracked the evolution of ex-ante evaluation methods used to review competing software architectures, concluding that similar methods might usefully be applied to competing service designs.

Chapter 3 sets out the potential value of creating an attribute-based method for evaluating service designs, derived from those used to evaluate software architectures, and proposes the adoption of a case study approach to exploring the impact of its first use in this domain.

This chapter describes the development of that method, from the adoption of a preferred source method from the domain of software architecture evaluation, through its adaptation for use in the domain of service design, to the enhancements that were adopted following the experience of its first use in the service design case study at the UK Border Agency.

4.2 Adopting a source method

As described in Chapter 2, methods for evaluating software architectures began to emerge in the late 1990s, with the second-generation Architecture Trade-off Analysis Method (ATAM), a refinement of SAAM, becoming the dominant method. Its maturity and dominance are highlighted in a survey of methods published in IEEE Transactions on Software Engineering in 2002 (Dobrica and Niemela, 2002), and again in a framework comparing software architecture evaluation methods presented at the Australian Software Engineering Conference in 2004 (Ali Babar, Zhu and Jeffrey, 2004).

Around 2006, the Office for National Statistics (ONS), of which the author was at the time the Chief Technology Officer, chose to adopt ATAM as its standard

for evaluating software architectures. The method, which calls for the creation of utility trees to examine the trade-offs involved between quality attributes, led to difficulties in fully engaging the relevant business stakeholders. Utility trees proved quite complex, and to users, who are more comfortable discussing specific details of their business requirements, quality attributes seemed abstract and remote.

A decision was taken to adapt ATAM to ensure that business stakeholders could fully participate in reviews. The adaptations were described in a paper presented at the International Meeting on the Management of Statistical Systems in April 2009, which was jointly organised by the United Nations, the European Commission and the OECD (Field, 2009b). Two key changes to ATAM stand out. The emphasis in the trade-off analysis was switched to the business requirements, but only after each high-level business requirement had been mapped to the set of desired quality attributes, which were selected as being appropriate to the solution under consideration. The abstract nature of 'utility' was replaced with the better understood concept of 'risk'. This last change was perhaps the more significant, and has endured through a number of iterations of the method, and remains a key feature of SARM.

Section 2.5 of Chapter 2 describes in some detail the risk model that was used as a corporate standard at ONS. This had the benefit of being widely understood by both business and IT staff at ONS, with a clear language and published guidance to aid its consistent interpretation by staff when measuring risks.

Its incorporation into the ONS variant of ATAM, combined with the greater focus on business requirements, led to a much greater level of engagement with business stakeholders during architecture reviews. Trade-offs were now discussed, not in terms of utility, as originally outlined in ATAM, but in terms of risk, a business language well understood by stakeholders. The two components of risk, 'impact' and 'likelihood', fitted neatly into different phases of the method, combining in the end to reveal the key trade-offs involved in each

potential architecture under consideration.

The 'impact' element could be assigned to each scenario, representing the impact on the business in the event that the adopted solution should fail to satisfactorily achieve the given scenario. The numeric value of the impact would remain consistent for all solution options under consideration, since the impact of failure relates to the relationship between the scenario and the business, and does not involve the solution options.

The 'likelihood' element of the risk relates directly to each solution option, and this becomes the key question that the review team need to address for each scenario / solution option under consideration: “what is the likelihood that this solution option will fail to satisfactorily achieve this scenario?”. Once this question has been address for each scenario / solution combination, the set of likelihood values can be recombined with the impact values assigned earlier to produce a set of risk exposure scores that reveal the trade-offs among the solution options. A spreadsheet tool to support the evaluation process, and to automate the calculation of risk scores, was developed and used at ONS.

This ONS variant of ATAM was the software architecture review method that was adopted as the source method for the development of SARM. It had the benefit of having been tried and tested on real IT projects, was well documented, and supported with a spreadsheet tool. In addition, the lead researcher had led the effort to refine ATAM, and so was intimately familiar with the method. The arguments for adopting a risk model approach to the trade-off analysis (replacing the use of the more abstract concept of 'utility') seemed, to the lead researcher, as likely to apply for the stakeholders of a service design project as they did for the stakeholders of a software solution at ONS.

4.3 Adding a quality model

The study of literature reported in Chapter 2 of this thesis drew the attention of the lead researcher to the existence of ISO 9126, the international standard for

software engineering product quality (International Organization for Standardization, 1991). Formal guidance relating to the use of ATAM does not specify any source model of quality upon which to draw when selecting the desired quality attributes that will be considered in an architecture evaluation. Practice at ONS had therefore relied on the judgement of the architecture team to select the most suitable quality attributes when preparing for a review. However, this meant that each review might consider a totally different set of attributes, and this variety made comparisons between reviews difficult, and participants and stakeholders faced an ever-changing vocabulary over time.

The discovery of a standardised quality model in part one of ISO 9126, that captured “all aspects of software quality” across just six characteristics, broken down further into twenty seven sub-characteristics, led the lead researcher to consider the potential benefits of its adoption for software architecture reviews at ONS. It could provide a consistent language with which participants and stakeholders, over time, could become increasingly familiar. And the consistency with which requirements would be classified would open up new possibilities to compare the outcomes of different architecture reviews.

At the same time, the lead researcher was exploring similarities and differences between the domains of software architecture and service design for this research project. The standardised quality model of ISO 9126 provided a consistent language with which to compare models of service quality. The results of this analysis are presented in Section 2.11 of Chapter 2, with a particular focus on the comparison of the Servqual dimensions of service quality with the characteristics and sub-characteristics of the ISO 9126-1 quality model.

The quality model from ISO 9126 was therefore adopted for use in SARM, enabling scenarios to be classified according to this standardised model that appeared to be as comprehensive in describing service quality as it was in describing software quality. The sixth characteristic, 'Portability' was renamed 'Adaptability'. There was no intention of changing its meaning, but the word 'Portability' seemed strongly associated with software. In preparing the

guidance, the language used to describe 'Portability' in the original ISO standard remained largely unchanged. The change in the name of this characteristic necessitated a further change to one of its constituent sub-characteristics. In ISO 9126, the first sub-characteristic of 'Portability' is 'Adaptability'. Having chosen this word for the new name of the characteristic, a different but similar meaning word had to be found to take its place. The word 'Variability' was chosen, and again the definition in the SARM guide is almost unchanged from the original ISO 9126 definition of the sub-characteristic 'Adaptability'. The resulting SARM service quality model is given in full in Chapter 5, Service Architecture Review Method, Section 5.5.2.

4.4 Adding stakeholder perspectives

The comparison of the domains of software architecture and service design, described in Chapter 2 of this thesis, highlights the importance of stakeholder perspectives in the evaluation of service designs. ATAM does not explicitly include stakeholder perspectives in the evaluation approach, but its developers do emphasise the importance of involving stakeholders in the development of the scenarios and throughout the trade-off analysis process (Kazman, Klein and Clements, 2000).

For the creation of SARM, the conclusion of the theoretical analysis strongly suggested a more explicit inclusion of stakeholder perspectives in the method, that might sit alongside the primary role of quality attributes in facilitating the examination of trade-offs among solution options. In the same way that aggregation of risk among scenarios that shared a common quality characteristic could highlight the level of risk associated with that quality characteristic for a given solution option, a method of aggregating risk according to the interests of different stakeholders was sought. This would allow the evaluation team to see each solution option 'through the eyes' of each stakeholder, potentially highlighting that some solution options might favour certain stakeholders to the detriment of others, just as the quality model is used to highlight the differences between solution options from a quality characteristic

perspective.

Achieving this desirable outcome would necessitate additional effort on the part of the evaluation team. In order to aggregate risks associated with each stakeholder, it would first be necessary to identify the stakeholders concerned, and then associate each stakeholder with those scenarios that are of significant interest to them. In this way, the overall risk landscape could be varied for each stakeholder. For example, a scenario that shows a high risk of failure with a particular solution option may be of significant importance to some stakeholders, but not to others.

A weighted model was proposed that would represent the importance of each scenario to each identified stakeholder. These weightings could be incorporated into the spreadsheet tool, which could then apply the weightings to the final risk landscape for all solution options, revealing the burden of risk borne by each stakeholder. The level of interest that each stakeholder has in each scenario would be classified as 'Strong Interest', 'Interest' or 'None'. These would be translated into weights of 2.0, 1.0 and 0.0 respectively, and used with the following formula to calculate the burden of risk borne by each stakeholder in a similar way to that used to calculate the burden of risk associated with each quality characteristic:

$$Burden_{jk} = \frac{\sum_{i=1}^n Risk_{ij} Weight_{ik}}{\sum_{i=1}^n Weight_{ik}}$$

where:

- i = a scenario
- j = a solution
- k = a stakeholder

The effect is that of a filter, revealing the level of risk associated only with those scenarios in which the stakeholder has an interest, duly weighted. A more sophisticated weighting model, with a more finely grained classification of

interest, would certainly be possible, but it was felt that agreement and consistency among the evaluation team could be more easily achieved with this simple 3-level model. The weights were separately parameterized in the spreadsheet implementation, so that changes to the weighting model could be made without difficulty.

4.5 Developing a software tool

A software tool to support the SARM process was developed, using Microsoft Excel spreadsheet software. The first worksheet was used to capture the scenarios and classify them according to the SARM quality model, assigning each scenario to a corresponding sub-characteristic. Each scenario occupied a row in the spreadsheet, and columns were allocated to the scenario description, the associated sub-characteristic and the risk impact component, representing the impact on the business should the service fail to satisfactorily achieve the given scenario.

The remaining columns of the worksheet were used to document the stakeholders, and the resulting matrix documented the level of interest of each stakeholder in each scenario. This worksheet thus corresponded to the creation of the context for the evaluation, with the set of scenarios duly classified according to both the risk and quality models, and associated to the set of stakeholders. The worksheet did not contain any information relating to the possible service designs or solution options that would be considered in the trade-off workshop. This information was the subject of the second worksheet.

The top part of the second worksheet replicated the rows of the first worksheet, with a row dedicated to each scenario. The columns were used to represent the solution options that would be evaluated. Up to six options were catered for, but only four solution options were considered by the evaluation team at UK Border Agency in the case study project. The matrix formed by the scenarios and the solution options was used to document the likelihood element of the risk. The likelihood here represented the likelihood of that particular solution

option failing to satisfactorily achieve the given scenario. As the corresponding impact level was already known for each scenario, the overall risk exposure score could be calculated automatically as soon as the likelihood value was entered. A neighbouring cell displayed the risk exposure, both as a numeric value, and with a red, amber or green background according to the severity of the risk.

Further down the worksheet, the risk burdens associated with each quality characteristic for each solution option were displayed, and below these, the corresponding risk burdens associated with each stakeholder for each solution option.

This worksheet thus allowed the evaluation team to explore risk trade-offs between the competing solution options from the perspectives of stakeholders and quality characteristics, and if required, examine the risk exposures at the level of individual solution / scenario combinations.

This was the version of SARM and its accompanying software tool that were presented at the First International Conference on Exploring Services Sciences in February 2010 (Field, 2010a) and used at the beginning of the case study. However, one major enhancement was made in the course of the case study to address an issue that emerged during the first workshop. And further enhancements were developed after the case study in the light of the experience of that first use of SARM. These enhancements are described in the remaining sections of this chapter, together resulting in the refined version of SARM, the user guide of which is presented in Chapter 5.

4.6 Subsequent enhancements

The first use of SARM at the UK Border Agency drew attention to a number of opportunities to make improvements to the method. The circumstances in which these opportunities arose, and the detail of how these improvements were developed and incorporated into an enhanced version of SARM, are

described in detail in Chapter 6, Service Design Case Study. Chapter 5, Service Architecture Review Method, presents the refined version of SARM in full. A description is given below of the enhancements that were made both during and after the completion of the case study, providing clarification of the distinction between the initial version of SARM that was adopted for the case study, and the refined version of SARM that is presented in detail in Chapter 5.

4.7 Adding 'Stakeholder Views'

As described in Section 4.4 above, the addition of stakeholder perspectives necessitates the completion of a substantial matrix of data to identify the level of interest each stakeholder has in each scenario. The software tool catered for this, and as a consequence of defining 21 scenarios and 12 stakeholder groups, the UK Border Agency evaluation team duly specified a total of 252 levels of interest. What had not been anticipated was the difficulty with which this substantial array of data could be validated. The team wanted to be sure that what they had agreed 'looked and felt right'. But re-examining 252 cells in a 21 x 12 spreadsheet did not help to give the team the confidence it sought.

The lead researcher undertook to develop a summary view that might help the team to verify that the information they had provided was both complete and coherent. Within a few days a visual 'heat map' had been developed in a separate worksheet entitled 'Stakeholder Views'. It was recognised that the need to summarise the data demanded a method for condensing the quantity of data. This problem centred around the quantity of scenarios; there were 21 in the case study, but future projects could well produce even more. The SARM quality model was recognised to be the key to the solution. Each scenario had been associated by the evaluation team with one of the model's 27 sub-characteristics, and each of these in turn belongs to one of six quality characteristics.

The scenario dimension of the matrix could therefore be reduced to just six by aggregating the data according to the characteristic with which each scenario

had been associated. The weights that had been applied to the levels of interest, as described in Section 4.4 above, could then be used to indicate the strength of association between each quality characteristic and each stakeholder. The new spreadsheet table therefore consisted of six columns (one for each quality characteristic), and one row for each stakeholder (twelve for this case study project). The following formula was then applied to each cell in the resulting matrix, to indicate the strength of association between each stakeholder and characteristic:

$$Interest_{jk} = \frac{\sum_{i=1}^{n_j} Weight_{ijk}}{2n_j}$$

where:

- i = a scenario
- j = a characteristic
- k = a stakeholder

If a stakeholder has 'Strong Interest' in all of the scenarios that are associated with that particular characteristic, the formula will produce a score of 1.0. If the stakeholder has just 'Interest' in all of the scenarios, a score of 0.5 will result, while 'No Interest' in all of the relevant scenarios would produce a score of 0.0.

An overall level of interest was also calculated by taking an average of the weights for a stakeholder across all scenarios, again placed on a scale of 0.0 to 1.0. This is calculated as follows:

$$Interest_k = \frac{\sum_{i=1}^n Weight_{ik}}{2n}$$

where:

- i = a scenario
- k = a stakeholder

A ranking of the stakeholders according to this overall level of interest allows a project team to see which stakeholders appear to have the greatest interest across the set of scenarios, and which have the least. The weighted scores, presented as a matrix with stakeholders occupying rows and the quality

characteristics on the columns, can be read in either direction. Reading across a row allows the user to see which characteristics are of most, and least, interest to that particular stakeholder, while reading downwards allows the user to see which stakeholders are most, and least, interested in that particular characteristic.

The visual appearance, and its understandability, were enhanced with the use of colour. Consultation with a data visualisation expert led to the choice of a neutral colour (blue) to indicate the strength of association represented by the scores. This aims to avoid a user's typical interpretation of the colour red indicating 'bad', or green indicating 'good'. A strong or weak association between a stakeholder and a quality characteristic is neither good nor bad, and there was no desire to lead the user of the tool towards a particular interpretation. A version of this table was sent to the Project Leader a week after the conclusion of the workshop along with a summary of the workshop outcome. It was well received, and used to help validate the stakeholder analysis that had taken place in the workshop.

4.8 Inverting the likelihood

The two dimensions of the risk model used in SARM required the evaluation team to assess the impact on the business in the event that the new service would fail to satisfactorily achieve a scenario, and the likelihood of that failure occurring for a given solution option. Both of these assessments demanded that the team consider the risk of failure to achieve the scenario, and this negative consideration caused a degree of confusion during the trade-off analysis that was not anticipated by the lead researcher.

The same risk model had been successfully used for many software architecture reviews at the Office for National Statistics without any similar confusion among review participants. But for the service evaluation participants in the trade-off analysis workshop at the UK Border Agency, the need to assess the likelihood of failure proved much more difficult. A likely source of this

confusion was the way in which the scenarios had been phrased. For a software architecture, the description will almost always be a positive one, expressing the need for a particular feature or function. For the UK Border Agency's service design, a number of the scenarios were described in negative terms, indicating that the new service design *must not* exhibit a particular feature or characteristic.

The confusion, described in more detail in Section 6.6 of Chapter 6, led the lead researcher to reconsider the way in which the risk model is applied to SARM. The solution was to invert the likelihood question that is asked of the team. Instead of requiring the team to indicate the likelihood of the solution *failing* to satisfactorily achieve the scenario, the system could ask the team to indicate the likelihood of *success*, and then invert the probability in order to combine the likelihood of failure with the impact for the generation of the overall risk exposure.

This new approach was duly implemented in the refined version of the SARM spreadsheet tool, and is described in detail in Chapter 5, Section 5.13.

4.9 Separating risk input from display

Another change directly triggered by the experience of the case study, and subsequently implemented, was to separate the input of risk likelihoods in the software tool from the display of the overall risk exposures. As described in Section 4.5 above, the original version of the tool had a single worksheet devoted to the trade-off analysis workshop. As each risk likelihood was entered, so the resulting risk exposure (calculated by multiplying the likelihood by its corresponding impact) was immediately displayed, both numerically and graphically (with a red, amber or green background to the cell to indicate a high, medium or low level of risk).

This step-by-step gradual revelation of the risk landscape attracted the attention of the workshop participants, and was quite entertaining. Four of the

participants commented directly on it in the subsequent interviews, one highlighting that you “could look at where it was going with the colours”, while another indicated that “it was evident sort of half way through where we were going with it”. Although one added that “it was quite nice that the results confirmed the way we were thinking”, reflection by the lead researcher after the completion of the project led to the conclusion that there was a high risk with the current solution that the project team might be unduly influenced in their later judgements by the visual signs revealed as a consequence of their earlier ones.

As a result, it was decided to separate the data entry of the risk likelihoods (of which there are a considerable number, being the product of the number of scenarios and the number of solution options) from the visual display of the resulting trade-off analysis. A separate worksheet, named 'Solution – Scenario Risks' was created for this purpose. It also had the effect of simplifying, somewhat, the subsequent 'Trade-off Analysis' worksheet which now focused entirely on risk exposure scores from a range of different perspectives.

The separation additionally facilitated a change to the presentation order of information on the 'Trade-off Analysis' worksheet. Formerly, the individual scenarios were presented in the top half of the worksheet in order that the risk likelihoods could be entered in sequence. Now there was no need for data entry on this worksheet, and so the presentation of information was reversed, with the summary, aggregated information moved to the top half, and the scenario-level detail moved to the bottom. This representation encourages the evaluation team to review the more architecturally significant aspects first, as revealed by the comparison of risks by quality characteristic and stakeholder.

4.10 Adding the 'Charts' worksheets

This enhancement was not prompted directly by anything that occurred during the case study project, but resulted from a more general reflection on the way in which evidence is presented to the evaluation team in the 'Trade-off Analysis' worksheet. The risk burden measures associated with each quality

characteristic help the team to see the overall trade-offs that are inherent among the competing solution options. They draw together the risk measures from the scenarios that share a common quality characteristic, so that the team can see whether some solutions might favour some characteristics at the expense of others.

This aggregation of risk is achieved by calculating an average across the set of associated risk measures. The nature of an average means that extreme individual scores are hidden, and sharper differences that may be clearly apparent at the individual score level will tend to diminish when brought together in an average value. Two similar average measures may therefore be composed quite differently: one being made up of similar levels close to the average, while the other might be made up of more extreme levels, perhaps an equal number of very high and very low levels of risk.

These differences might be highly significant to the evaluation. It could be, for example, that one of the constituent very high risks cannot easily be mitigated. Yet its significance would be masked if the evaluation team only viewed the resulting average measures at the level of quality characteristic. The 'Charts' worksheet was created to help draw the attention of the evaluation team to these situations. Using stacked bar charts, it displays graphically the distribution of constituent risks for each solution option by quality characteristic. If two similar levels of average risk, at the level of the quality characteristic, are composed of very different levels of underlying risk, this will be visually apparent.

The 'Charts' worksheet was placed immediately following the 'Trade-off Analysis' worksheet, making it easy for an evaluation team to swap between the two worksheets as they explore the information that has been calculated for use in the trade-off analysis workshop.

4.11 Adopting a stakeholder model

The most visually noticeable change between the first version of the software tool that was used for the case study, and the more recent refined version described in Chapter 5 is the incorporation of a stakeholder model. Like the addition of the 'Charts' worksheet, this change was not prompted specifically by the case study, but it was inspired by further research in recognition of the significance of stakeholders to both the case study project, and more generally to service design projects.

It will be recalled that SARM requires the identification of stakeholders or stakeholder groups, and their association with the various scenarios that represent the high-level business requirements of the service. This information then facilitates the exploration of the risk landscape of each solution option from the different perspectives of the individual stakeholders in the trade-off analysis, and the data to support this is duly calculated and displayed in the 'Trade-off Analysis' worksheet of the software tool.

In the same way that the use of a quality model provides a valuable abstraction from the lower-level detail of individual scenarios, so it was recognised that SARM could benefit from a similar abstraction of the stakeholder dimension. Stakeholders identified in the first workshop could be classified according to the model, and the classes or categories of stakeholder represented in the model could then be used to aggregate and display risk, facilitating a new level of trade-off analysis – by stakeholder class.

Sadly, this recognition arrived too late to be introduced into SARM before the completion of the case study project, but it was added to both SARM and the software tool subsequently. Section 2.10 of Chapter 2 discusses a number of models that are used to classify stakeholders of a project or business. Having recognised the potential benefit of incorporating a stakeholder model, the lead researcher was then faced with the task of selecting a suitable model from among a wide range of available models. As described in Chapter 2, the model

proposed by Mitchell, Agle and Wood combines elements of the commonly used two-dimensional models with aspects of the 'onion model'. The richness offered by this model persuaded the lead researcher to adopt it for the refined version of SARM, as it enable two different levels of abstraction, Stakeholder Saliency and Stakeholder Class (Mitchell, Agle and Wood, 1997).

A stakeholder is classified by the model according to their possession of up to three attributes: Power, Legitimacy and Urgency. As identification of relevant stakeholders is the first step in the SARM process, a new worksheet was placed at the beginning of the software tool named 'Stakeholder Analysis'. This is used to capture up to fifteen different stakeholders or stakeholder groups, and classify them according to the Mitchell, Agle and Wood model. A graphical representation of the model is provided in the original paper, and a version of this has been incorporated into the 'Stakeholder Analysis' worksheet, allowing the user to present the stakeholder analysis in a graphical as well as textual form. The information captured and calculated in this first worksheet is then used to pre-populate the remaining worksheets where appropriate.

The 'Stakeholder Burden' part of the 'Trade-off Analysis' worksheet was extended to include aggregated stakeholder burden risk measures by Stakeholder Saliency and Stakeholder Class, in addition to the existing risk measures by individual stakeholder or stakeholder group.

4.12 Future enhancements and changes

This chapter has documented the choices that were made in transforming a version of ATAM used for software architecture reviews into the first version of the Service Architecture Review Method. It has described the development of a tool to support the new method, and the enhancements to both method and tool that were made during and following the first use of SARM in a service design project at the UK Border Agency.

The reasons for adopting a derivation of ISO 9126 for SARM's service quality

model, the ONS risk model, and later, the Mitchell, Agle and Wood stakeholder typology, have been explained. But it is clear that different decisions could have been made, and future researchers or practitioners may have good reason to wish to adopt alternative models in place of the three that were adopted for these initial versions of SARM. The overall structure of SARM need not be changed to accommodate different models for any of these three perspectives. The software tool has been implemented with the possibility of change in mind. A separate worksheet, 'References', holds the details of each of these models, along with the weighting values that are assigned to different levels of stakeholder interest in scenarios. Most of the work to change any one of the models would involve amending parameters on this worksheet, though the visual representations on other worksheets may also require some modification.

One practical example of such a change has recently been completed. In 2011, a new standard for systems and software quality, ISO 25010, was published, replacing ISO 9126 (International Organization for Standardization, 2011). An alternative version of the software tool, incorporating the quality model of this new standard, has already been created in anticipation of future practical use.

The 'refined' version of SARM presented in the following chapter cannot be considered a final version of SARM. It reflects the refinements adopted after just a single use of the method, and more changes will surely follow in the light of greater experience.

Chapter 5: Service Architecture Review Method

5.1 Introduction

This chapter presents the refined version of the Service Architecture Review Method (SARM) in the form of a guide for practitioners. It represents a key deliverable of the research project, and a contribution to practice that is directly and immediately usable by practitioners. It should be noted that the method presented in this guide incorporates a number of enhancements that were added during and after the conduct of the case study with the UK Border Agency. A description of how the original version of SARM was developed, and the enhancements that were subsequently adopted to form this refined version of the method has been given in Chapter 4.

5.2 Service Architecture Review Method

Welcome to this practitioners' guide to the application and use of Service Architecture Review Method (SARM). SARM is an evaluation method, intended for use to evaluate and compare competing service designs *prior* to the implementation of a preferred design. Of course it may also be used to evaluate potential changes to an existing service, in which case the existing *As-Is* service will be one of the competing service designs to be evaluated alongside some new alternatives. This guide explains how to apply SARM – that is, how to evaluate competing service designs.

It does not explain how to design those competing services. Indeed it can be seen as a strength of the method that it is completely independent of the means by which service designs are derived. They may be alternative designs created within the service delivery organisation, or they may be designs proposed by external design companies that have been invited to respond to an 'invitation to tender'. The methods used to design the proposed services remain outside the scope of this guide, and do not affect the validity of the Service Architecture Review Method. All that is required is that each design that is to be submitted to the evaluation process can be described in sufficient detail that its *architecture* can be understood and assessed by an evaluation team against a

set of high level requirements.

As we shall see, SARM is divided into two parts – Part 1 involves the creation of a context in which the evaluation will take place. This includes the identification and classification of the service's stakeholders, and the description of what is expected of the service. Part 2 involves the evaluation of competing service architectures and the selection of a preferred service architecture. There may be a gap in time between the conclusion of the activities involved in Part 1 and the commencement of Part 2 while solution options are developed or commissioned.

5.3 Definitions

This guide uses terms that have a specific meaning in the context of SARM, with which some readers may not be familiar. We shall start with a definition of these terms so that the reader can understand what is meant by the use of these terms within this guide. There is no desire or need to enforce the use of these terms on organisations that adopt SARM, and where similar roles or functions carry different labels, it is strongly suggested that the terms used here are changed to suit the prevailing language and culture of the adopting organisation.

Architecture The architecture of a service is the structure or structures of the service, which comprise process components, the externally visible properties of those components, and the relationship among them. (*Field, 2010a, p. 118*)

Business Analyst A liaison among stakeholders in order to understand the structure, policies, and operations of an organisation, and to recommend solutions that enable the organisation to achieve its goals. (Brennan, 2009)

Facilitator	An individual who enables groups and organisations to work more effectively; to collaborate and achieve synergy. She or he is a “content-neutral” party who by not taking sides or expressing or advocating a point of view during the meeting, can advocate for fair, open, and inclusive procedures to accomplish the group's work. (Kaner et al., 2007, p. xiii)
Project	A temporary endeavour undertaken to create a unique product or service. (Project Management Institute Standards Committee, 1996)
Risk	The combination of the likelihood of an occurrence of a hazardous event or exposure(s) and the severity of impact that can be caused by the event or exposure(s). <i>adapted from (British Standards Institute, 2007, p. 4)</i>
Service	A provider/client interaction that creates value. (IBM Research, 2010)
Sponsor	The Project Sponsor is responsible for identifying the business need, problem or opportunity. The sponsor ensures the project remains a viable proposition and that benefits are realised. (Association for Project Management, 2006)
Stakeholder	Any group or individual who can affect or is affected by the Service that also has at least one of the attributes Power, Legitimacy or Urgency. <i>adapted from (Freeman, 2010, p. 46)</i> and (Mitchell, Agle and Wood, 1997)

5.4 Why should we evaluate competing designs?

Before proceeding to a detailed description of the SARM approach to evaluating service designs, it is worth first considering why one would wish to undertake such an exercise, given that it involves considerable time and effort.

SARM provides a structured approach to examining the way in which competing service designs, or solution options, satisfy a set of business requirements from an architectural perspective.

It is proposed that any service design has an underlying architecture or structure that consists of the service design's components, their outward properties and the relationships among those components. The architecture can be thought of as an abstraction of the detailed design revealing the fundamental structure of the design, having stripped away the 'cosmetic' details of the design. This is not to suggest that those 'cosmetic' details are unimportant – they are often critical in defining the quality of the service in question. But by focusing on the architecture of service designs, it becomes easier to identify and explore the relative strengths and weaknesses of competing designs. It is also typically the case that an architectural flaw in a service design is difficult to correct, whereas a cosmetic detail can often be changed without much disruption to the service. For example, a change to a customer service representative's script can often be made without difficulty, while changing the location of a particular customer interaction might involve substantial re-modelling of a service, perhaps even changing the layout of a building to suit a changed customer journey.

It is an underlying assumption here that, for any given set of business requirements, there is more than one possible service design that can provide a satisfactory solution. It is commonly the case that a set of business requirements contains conflicts that have to be confronted by any one proposed design, and that complete satisfaction of some of the requirements will inevitably result in, at best, partial satisfaction of others.

An example can illustrate this point. Among the business requirements for a service might be one that requires keeping information secure, and ensuring that customer information is kept confidential. Another requirement might be that the customer experience is non-intrusive, and customer touch-points are dealt with quickly and efficiently. The first relates to the Security of the service, whilst the second can be considered an aspect of its Attractiveness. A design that wishes to address the Security requirement might involve assuring the identity of the customer before disclosing any customer information. Varying degrees of assurance are possible, which might include biometric verification of the customer's identity. However, implementation of a biometric solution, whilst providing a highly accurate means of addressing the Security requirement, is likely to result in an expensive service that is relatively intrusive and potentially time-consuming from the customer's perspective. The Attractiveness requirement has been compromised to a certain extent in a bid to address the Security requirement. An alternative solution may find a different 'trade-off' between these two requirements, perhaps reducing the degree of assurance regarding the customer's identity, but also reducing the degree of intrusion and inconvenience to the customer.

There is no 'correct' solution here – either design might be preferred, depending on the balance of risks involved in considering the two designs against the two requirements. Much will depend on the nature of the service and information concerned, and the importance of the two quality characteristics involved in this simple example, Security and Attractiveness.

This illustration considers just two simple requirements. In a complex service there may be many requirements, and numerous conflicts that may only become apparent when considering them in the context of proposed service designs.

SARM allows a project team to first articulate the service's stakeholders and requirements, then using this as a context for evaluation, to examine competing service designs to understand each design's trade-offs among the

requirements. A decision can then be taken on which design to implement, in the knowledge of the inevitable compromises that the preferred design represents.

By focusing on the architecture of each potential design, and relating these to the requirements and stakeholders, it is possible to understand these trade-offs before implementing the designs, and without incurring the cost of completing each design in full detail. All that is required of each potential design is an understanding of its architecture – the structure and properties of its components and their relationships to each other – and how that architecture relates to the service requirements.

It is strongly suggested that a collaborative approach should be adopted when applying SARM to a project. Much of the effort involved in articulating the requirements and evaluating the solution options in a trade-off analysis takes place in collaborative workshops, involving a representative group of stakeholders. This is a time-consuming approach, and may seem unduly expensive, especially as it takes place before any effort has been expended implementing the service.

However, experience has shown that involving stakeholders in the early stages of design improves collective understanding of all the requirements, and reaching an early and collective decision on the preferred design results in a much smoother implementation. Changing an unsatisfactory service after it has been implemented is often much more costly and disruptive than spending time considering competing architectures and selecting one that best satisfies the set of requirements.

5.5 Key concepts

Before moving on to a detailed description of the method, SARM makes use of three key concepts that we will introduce here, so that they will not require detailed explanation when they appear during the description of the SARM process. They are Stakeholder Analysis, the SARM Quality Model and the

SARM Risk Model.

5.5.1 Stakeholder Analysis

The inclusion of stakeholder analysis in business change projects has long been considered a recommended practice (Littau, Jujagiri and Adlbrecht, 2010), ensuring that the varying perspectives of all those affected by the project are taken into account. There is no reason for projects dedicated to service design to deviate from this practice, and the stakeholder analysis conducted at the beginning of the SARM process facilitates the evaluation of competing solution options from the perspectives of different stakeholder groups and classes.

There are a number of popular approaches to classifying stakeholders when conducting a stakeholder analysis. SARM uses the model first introduced by Mitchell, Agle and Wood (Mitchell, Agle and Wood, 1997) which classifies stakeholders using three attributes: Power, Legitimacy and Urgency, as illustrated in Figure 3 below.

A stakeholder has Power if they have the ability to get the project or service to do something that it would otherwise not have done. A stakeholder has Legitimacy if their actions are considered to be desirable, proper and appropriate in the context of the project or service. And a stakeholder has Urgency if their claims call for immediate action.

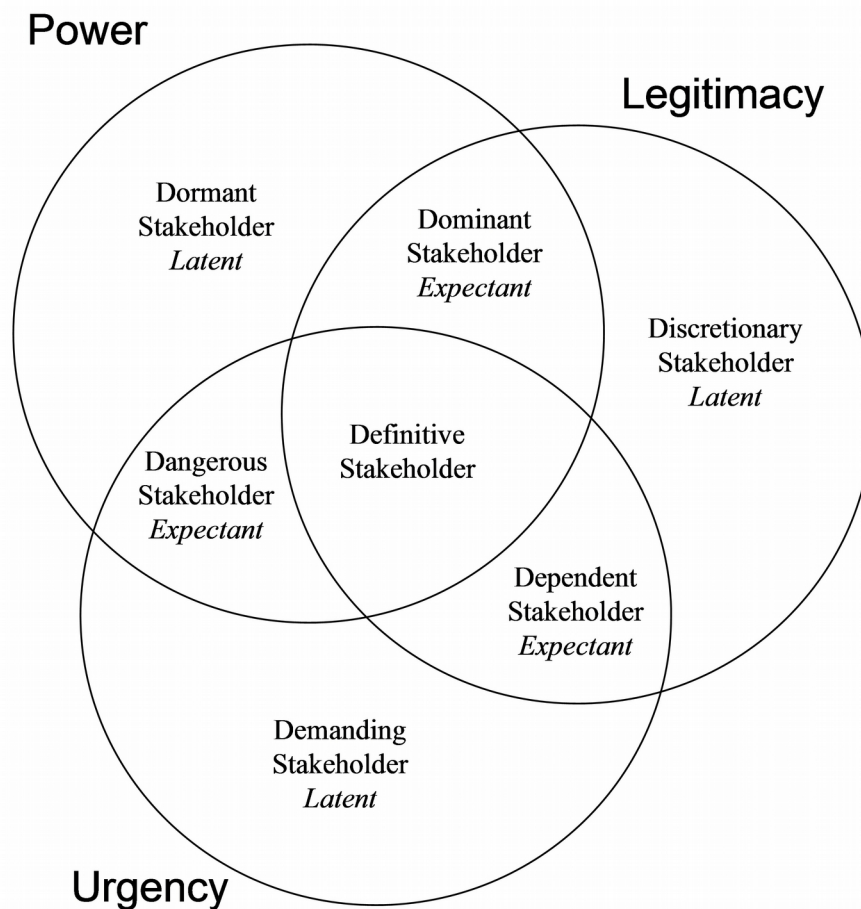


Figure 3 Mitchell, Agle and Wood's Stakeholder Typology, 1997

Possession of one, two or all three of these attributes will result in a stakeholder falling into one of seven classes, as illustrated in above. These classes have been further grouped according to their Salience, with three classes forming the Latent Stakeholders (possessing just one of the attributes), three forming the Expectant Stakeholders (possessing any two of the attributes), and one forming the Definitive Stakeholders, who have all three attributes. The short descriptions given below, adapted from Mitchell, Agle and Wood's original paper, will help the reader to understand the characteristics of each stakeholder class:

Latent Stakeholders

Dormant stakeholders possess power to impose their will on a project or solution, but by not having a legitimate relationship or an urgent claim, their

power remains unused (though they may have potential to acquire legitimacy or urgency).

Discretionary stakeholders are typically beneficiaries of a solution that have made no claims for involvement (lacking power and urgency). There is often no pressure on managers to engage in an active relationship with them.

Demanding stakeholders, those with urgent claims but having neither power nor legitimacy, are the “mosquitoes buzzing in the ears” of managers: irksome but not dangerous, bothersome but not warranting more than passing management attention, if any at all.

Expectant Stakeholders

Dominant stakeholders will typically have some formal mechanism in place that acknowledges the importance of their relationship with the project or solution. They form the “dominant coalition” in the enterprise and have substantial influence.

Dependent stakeholders depend upon others for the power necessary to carry out their will. Power, in their case, is governed through the advocacy of guardianship of other stakeholders, or through the guidance of internal management values.

Dangerous stakeholders can be coercive, even violent, due to their lack of legitimacy, and so are literally “dangerous” to the project or solution.

Definitive Stakeholders

Definitive stakeholders are, by exhibiting both power and legitimacy, already a member of the dominant coalition. With the addition of urgency, managers have a clear and immediate mandate to attend to, and give priority to, these stakeholders’ claims. Any of the expectant stakeholders can become definitive by acquiring the missing attribute, and thus moving to the heart of the project or solution.

5.5.2 The SARM Quality Model

As described earlier, SARM helps project teams understand the trade-offs and compromise inevitable in any realisation of a given set of requirements. There are typically too many requirements for the trade-off analysis to be conducted at the level of the individual requirement. So a Quality Model is used to classify requirements, allowing those requirements that share a common characteristic to be grouped together. The trade-off analysis can then be conducted initially at the more manageable level of the Quality Characteristics, with the ability to 'drill down' into the constituent requirements for each characteristic whenever a more detailed exploration of a given trade-off is required.

Characteristic	Sub-characteristic
Functionality	Suitability Accuracy Interoperability Security Functionality Compliance
Reliability	Maturity Fault Tolerance Recoverability Reliability Compliance
Usability	Understandability Learnability Operability Attractiveness Usability Compliance
Efficiency	Time Behaviour Resource Utilisation Efficiency Compliance
Maintainability	Analysability Changeability Stability Testability Maintainability Compliance
Adaptability	Variability Installability Co-existence Replaceability Adaptability Compliance

Table 7 The SARM Quality Model

The model, which has been adapted from ISO 9126-1 (International Organization for Standardization, 1991) contains just six characteristics, further broken down into a total of 27 sub-characteristics (see Table 7 above). Any number of scenarios can be classified according to this model by allocating each scenario to the sub-characteristic that best characterises the nature of the requirement. In this way, all scenarios can be grouped into just six characteristics that together cover a 360 degree view of the requirements landscape.

A brief description of each characteristic and sub-characteristic is given below. It has been adapted from the original ISO 9126-1 standard.

Functionality

The capability of the service to provide functions which meet stated and implied needs when the service is performed under specified conditions.

Suitability - The capability of the service to provide an appropriate set of offerings for specified tasks and user objectives.

Accuracy - The capability of the service to provide the right or agreed outcomes with the needed degree of precision.

Interoperability - The capability of the service to interact with one or more specified services.

Security - The capability of the service to protect assets (people, things, information) so that unauthorised persons or systems cannot access them and authorised persons or systems are not denied access to them.

Functionality Compliance - The capability of the service to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality.

Reliability

The capability of the service to maintain a specified level of performance when performed under specified conditions.

Maturity - The capability of the service to avoid failure as a result of faults in the service.

Fault Tolerance - The capability of the service to maintain a specified level of performance in cases of service component faults.

Recoverability - The capability of the service to re-establish a specified level of performance and recover any assets (e.g. information) affected in the case of a failure.

Reliability Compliance - The capability of the service to adhere to standards, conventions or regulations relating to reliability.

Usability

The capability of the service to be understood, learned, used and attractive to the service participants, when used under specified conditions.

Understandability - The capability of the service to enable service participants to understand whether the service is suitable, and how it can be used for particular tasks and conditions of use.

Learnability - The capability of the service to enable service participants (customers, providers) to learn its application.

Operability - The capability of the service to enable its participants to operate and control it.

Attractiveness - The capability of the service to be attractive to its participants.

Usability Compliance - The capability of the service to adhere to standards, conventions, style guides or regulations relating to usability.

Efficiency

The capability to provide appropriate service levels, relative to the amount of resources used, under stated conditions.

Time Behaviour - The capability of the service to provide appropriate response, speed and throughput rates when being performed, under stated conditions.

Resource Utilisation - The capability of the service to use appropriate amounts and types of resources when the service is performed under stated conditions.

Efficiency Compliance - The capability of the service to adhere to standards or conventions relating to efficiency.

Maintainability

The capability of the service to be modified. Modifications may include corrections, improvements or adaptation of the service to changes in environment, and in requirements and functional specifications.

Analysability - The capability of the service to be diagnosed for deficiencies or causes of failure, or for the service elements to be modified to be identified.

Changeability - The capability of the service to enable a specified modification to be implemented.

Stability - The capability of the service to avoid unexpected effects from modifications of the service.

Testability - The capability of the service to enable service modifications to be validated.

Maintainability Compliance - The capability of the service to adhere to standards or conventions relating to maintainability.

Adaptability

The capability of the service to adapt to a changed environment, or be deployed in a different environment.

Variability - The capability of the service to be varied for different specified environments without applying actions or means other than those provided for this purpose for the service considered.

Installability - The capability of the service to be installed in a specified environment.

Coexistence - The capability of the service to co-exist with other independent services in a common environment sharing common resources.

Replaceability - The capability of the service to be used in place of another specified service for the same purpose in the same environment.

Adaptability Compliance - The capability of the service to adhere to standards or conventions relating to adaptability.

5.5.3 The SARM Risk Model

Management of Risk: Guidance for Practitioners defines risk as “an uncertain event or set of events which, should it occur, will have an effect on the achievement of objectives” ((Murray-Webster and Office of Government Commerce, 2010, p. 1)). According to the Office of Government Commerce a risk consists of a combination of the probability of a perceived threat occurring and the magnitude of its impact on objectives. Many organisations have adopted an approach to quantifying risks that involves separately assessing the likelihood and impact of a given risk using numeric scales, and estimating the

overall exposure to that risk by calculating the product of the two assessments. SARM follows this approach, which has become increasingly widespread among risk management professionals since its early use in 1953 by Dutch mathematician and engineer, David van Dantzig, to assess flood risk.

Table 8 and Table 9 below show the scales adopted by the SARM Risk Model for risk likelihood and risk impact respectively:

Level	Description
1	Rare
2	Unlikely
3	Possible
4	Likely
5	Almost Certain

Table 8 Risk Likelihood

Level	Description
1	Negligible
2	Minor
3	Moderate
5	Major
6	Catastrophic

Table 9 Risk Impact

For any given risk, the overall exposure can be calculated from the product of the risk's likelihood and impact levels. Figure 4 below shows this, together with colour coding that illustrates the overall level of exposure:

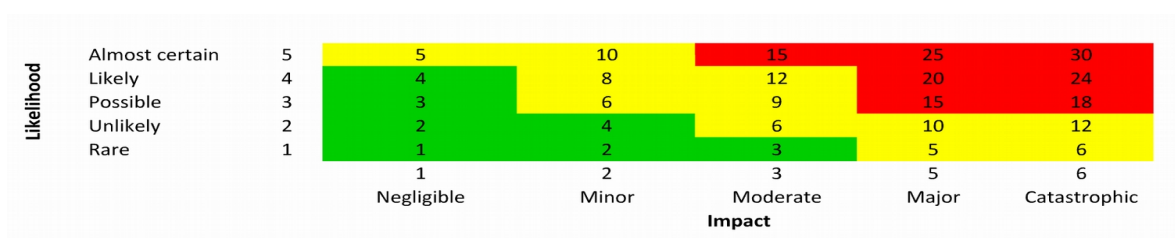


Figure 4 Risk Exposure

Now that we have established why evaluation of competing service designs is worthwhile, and introduced the key concepts that are used within SARM, we can describe the method itself.

5.6 The SARM Process – an overview

We will describe SARM via its process, following a simple example that can be used to illustrate the method from start to finish. As SARM involves a trade-off analysis that requires quite complex calculations of risk scores, a tool implemented with a spreadsheet software application supports the process. This practitioners' guide also serves as a user's guide to the tool, and includes screen-shots of the tool as we run through the process with the example.

Figure 5 below illustrates the SARM process, each part of which will be subsequently described and illustrated in detail:

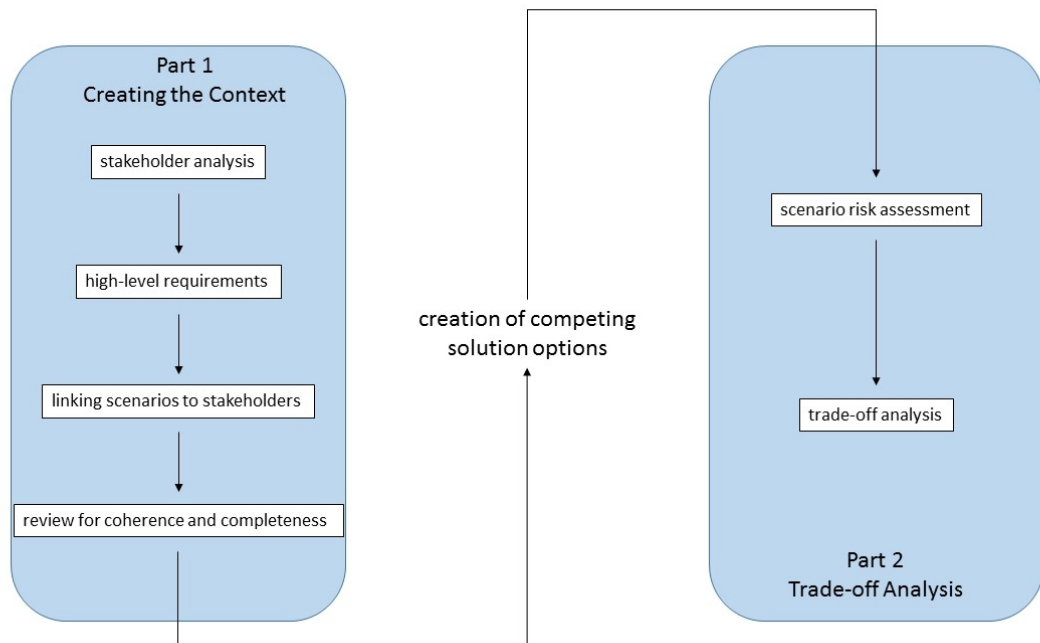


Figure 5 The SARM Process

The nature of different organisations, and different service design projects, will lead to different ways of organising a service design project. Indeed, some organisations will conduct service design without calling it that, and without any

formal project structure. Others will see service design as a major business change activity, and treat it as a formally managed project.

For the purposes of this guide, we will assume two key roles:

1. Project Sponsor – the person responsible for identifying the business need relating to the service, and typically the owner of the service, and budget-holder responsible for paying for the new service design.
2. Architect – a person capable of leading the architectural evaluation of competing service designs, also possessing the skills of a Business Analyst and a Facilitator.

The roles and labels used in the guide will not necessarily correspond directly to those used in various organisations, but the skills and functions should be recognisable and may be divided among a wider group of project participants. It is the skills and activities that are important, not the number of people or their job titles, which may vary considerably from one organisation to another.

5.7 Introducing a simple service design example

This guide will use a simple example throughout to illustrate the method. It has been chosen for its simplicity rather than its realism, so that the example uses of the tool, screen-shots of which are included, can be easily followed by the reader.

The example involves the question of which mode of transport to utilise to travel to and from work each day. We will keep the number of stakeholders and requirements, as described in scenarios, to a minimum, to make the example easy to follow, and as the application of the method involves a numerical expression of risk, the mathematics that will be shown with the tool can be easily verified by the reader with a calculator or pen and paper.

5.8 Part 1 – creating the context

The first part of SARM involves creating the context in which solution options can be evaluated. This can be further divided into four steps:

- a) identifying and mapping the stakeholders;
- b) defining the high-level requirements through a set of scenarios, and classifying them;
- c) linking the scenarios to the stakeholders;
- d) reviewing the information for coherence and completeness.

Each of these steps is a separate activity, and is likely to involve different people.

1a - Identifying and mapping the stakeholders

It is recommended that the stakeholder analysis is conducted in a meeting involving the Project Sponsor and the Architect. Others involved in the service concerned, or with relevant skills (e.g. a business analyst), may also attend. The initial analysis can be typically completed in a single meeting of about 90 minutes' duration.

The aim of the meeting is to identify and classify up to fifteen stakeholders. A stakeholder is defined as:

any group or individual who can affect or is affected by the Service
(Freeman, 1984)

It is suggested that the process of identification begins with a 'brainstorming' approach, without consideration of the limit of fifteen that will eventually be imposed. Names of individuals or groups should be posted to a board, and discussed freely among the attendees to ensure that the proposed stakeholders fit within the above definition. It can sometimes be useful to split the discussion

into consideration first of stakeholders within the organisation ('internal' stakeholders), and then move on to stakeholders that are 'external' to the organisation (e.g. customers, suppliers, partners).

If, at the end of this process, there are fewer than fifteen stakeholders, the meeting can proceed to the classification stage. If there are more, then some of the names should be grouped together to form clusters of stakeholders who share a common perspective on the service. This process should continue until there are fifteen or fewer stakeholder groups, and where clusters have been formed, a collective name for the cluster should be agreed, and the composition of each cluster should be documented.

It should be borne in mind that the stakeholder analysis need not only take place once during the project. There may be good reason to revisit the stakeholder analysis following articulation of high-level requirements.

Once fifteen or fewer stakeholders (or stakeholder clusters) have been identified, the next activity is to classify them according to the stakeholder model described earlier in this guide.

To classify the stakeholders that have been identified, it is suggested that they are considered in turn, and their possession of each of the attributes (Power, Legitimacy and Urgency) is discussed and agreed. These decisions should be documented, and each stakeholder can then be assigned to its corresponding Class in the Topology. The descriptions of the classes can be used to help verify the correct classification of each stakeholder.

Plotting the stakeholders graphically on the typology can also help the meeting participants confirm that they have identified all the relevant stakeholders. Whilst you should not necessarily expect to find stakeholders in each class (most if not all stakeholders will have Legitimacy, so Demanding, Dormant and Dangerous stakeholders are less common), it is worth taking a little time to review any empty classes to confirm that there are not stakeholders belonging to them for this project.

5.9 Using the SARM tool

The SARM tool supports each step of the SARM process, and the first worksheet in the spreadsheet is dedicated to the Stakeholder Analysis. It allows the user to document the stakeholders, record their possession of the attributes, and it automatically assigns their corresponding Salience and Class.

However, it is strongly suggested that the tool should not be used from the very beginning of the stakeholder analysis meeting. The meeting should encourage a flow of discussion and debate, and this can often be inhibited by group completion of a form in a software tool, which can minimise discussion and focus too much on just ‘getting the form filled’. Most of the analysis should therefore be completed on a whiteboard or flip chart, and the tool can then be used when the discussion is complete, to record the decision and classify the stakeholders.

Figure 6 shows how the stakeholder analysis worksheet looks when first opened.

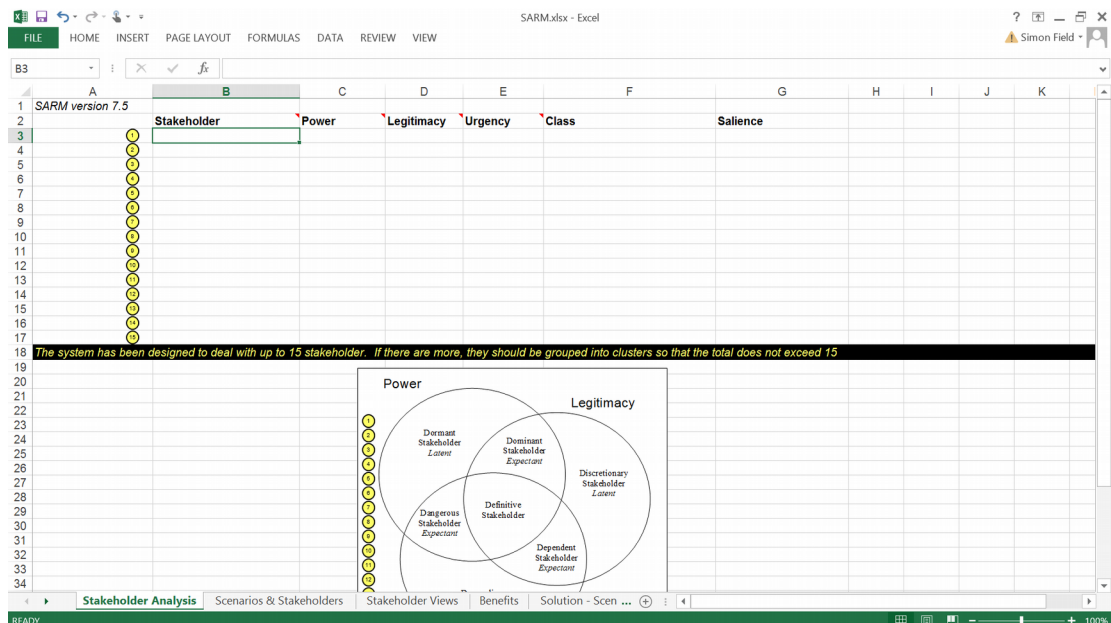


Figure 6 Stakeholder Analysis Worksheet

Stakeholder names are entered in column B, and columns C, D and E record

whether or not each stakeholder possesses Power, Legitimacy or Urgency. The Class and Salience for each stakeholder are populated automatically in columns F and G respectively.

The graphic in the lower half of the worksheet is provided to allow the user to drag the numbered marker representing each stakeholder into the relevant class on the Topology, providing a useful graphical representation of all the stakeholders.

The spreadsheet can now be saved, and further information will be added to the other worksheets as the project progresses through the SARM process.

5.10 Example stakeholders

Having described the process and introduced the tool, we now return to the example that we will be following throughout this guide – the question of which service to adopt for travelling to and from work. At this stage we are not concerned with solutions to the problem; we are beginning to create the context in which solution options will be evaluated. The start of this process is to identify the stakeholders for this by considering who can affect, or is affected by, a person's journey to and from work.

The individual who is actually making the journey (the commuter) is perhaps the most obvious stakeholder. His or her work colleagues are also affected – especially if the individual regularly arrives late, or in an unfit state! And the individual's immediate family can be likewise affected if the individual returns home late or in an unfit state. Whilst the family can probably remain a single group of stakeholders, who share a common perspective on the individual, his or her work colleagues might be a large group perhaps worth breaking down into sub-groups that have distinct perspectives. For the purposes of this simple example we will break this into two stakeholders, the individual's peer colleagues, and his or her boss.

A more comprehensive stakeholder list might also include transport providers,

local authorities and others involved in the supply of transport services. But for the purposes of this simple example, we will keep to the four stakeholders identified above:

1. The commuter
2. Work colleagues
3. Boss
4. Family

Now we have identified the stakeholder groups, the next task is to place them on the Stakeholder Typology. This involves allocating each group to one of the seven classes described earlier. Perhaps the easiest way to do this is to go through each group and consider whether they possess each of the three attributes, Power, Legitimacy and Urgency.

The commuter has all three attributes, and is thus a Definitive Stakeholder. The work colleagues are, perhaps, the least able to exert influence over the solution, and do not have particular urgency in seeking a solution, but they have legitimacy, and so fall into the Discretionary Stakeholder class. The boss clearly has both power and legitimacy, but like the work colleagues, no urgency, and so falls within the Dominant Stakeholder class, while the family has all three attributes, and so falls into the Definitive Stakeholder class alongside the commuter.

The Stakeholder Analysis worksheet of the spreadsheet makes it easy to classify each stakeholder group. Enter the stakeholder group names in column B, and indicate possession of the three attributes in columns C to E by selecting TRUE or FALSE from the drop-down menus. The class and salience classifications are automatically determined, based on the possession of the three attributes.

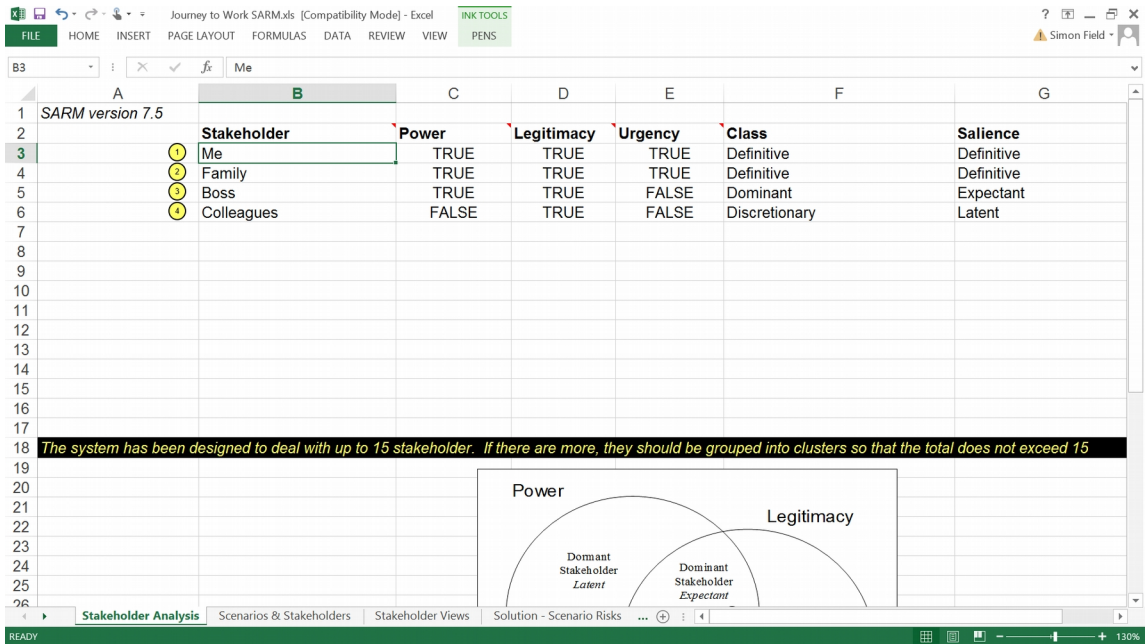


Figure 7 Example Stakeholder Analysis worksheet (top)

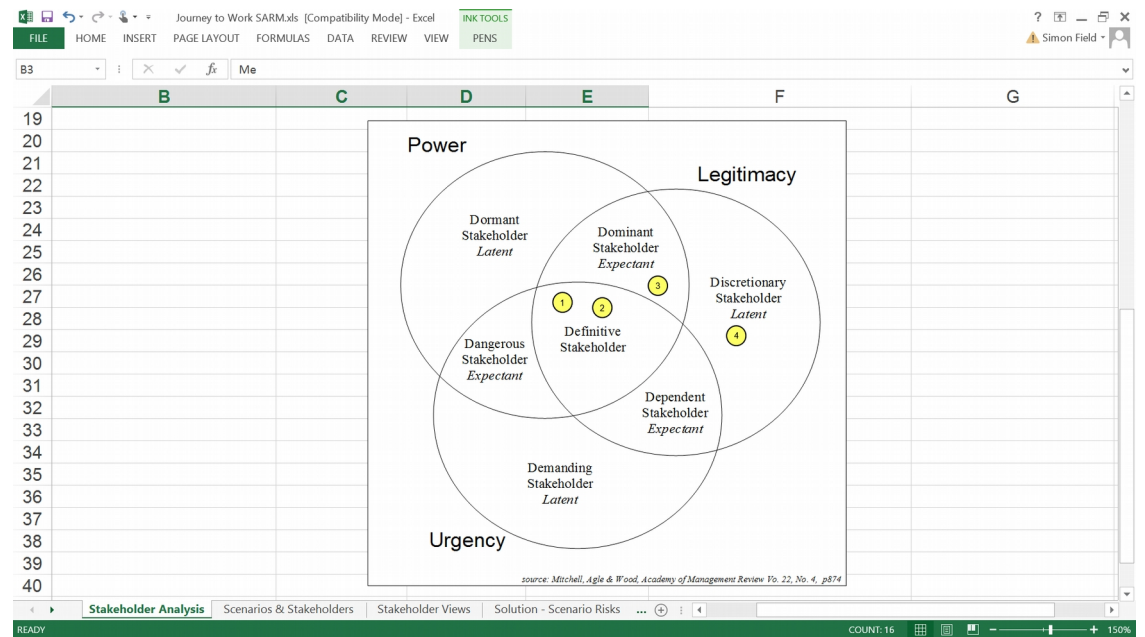


Figure 8 Example Stakeholder Analysis worksheet (bottom)

The lower portion of the worksheet contains a copy of the graphical typology, with a set of numbered discs that correspond to the stakeholders that have been entered. This allows the user to move each disc to the relevant section of the typology, providing a visual ‘map’ of the stakeholders, which may be of use when discussing the outcome of this analysis with the project sponsor or other stakeholders.

1b - Identifying the scenarios

The next step in creating the context for the trade-off analysis is the identification of the scenarios that collectively describe the high level requirements. This activity is best conducted as a workshop, and it is recommended that a representative group of stakeholders should participate in this. This ensures that scenarios are captured that represent the requirements from the perspectives of different stakeholders. If it is not possible for all stakeholders to be present, it is a good idea to have some of those present in the workshop 'put themselves in the shoes' of absent stakeholders, and where appropriate, suggest requirements 'on behalf of' those absent stakeholders.

Depending on the size and scope of the service under consideration, this workshop can take at least half a day, and it may be worth considering splitting the workshop into different sessions across more than one day, to ensure that energy levels in the room remain high.

As with the Stakeholder Analysis, it is strongly recommended to make use of a whiteboard or flipchart while capturing information from participants, rather than focusing too much attention on the software tool. Having a 'scribe' re-enter in the tool what is written on the flipchart can save time later on, but this activity should not become a distraction for participants.

The aim of the first part of this workshop is to obtain a set of scenarios that collectively describe the high level requirements of the service, and to classify these scenarios according to the SARM Quality Model.

The aim is to arrive at a set of scenarios that together encompass the high level requirements of the service. There is no definitive answer to the question 'how many scenarios do we need?', but they need to cover all perspectives and aspects of quality without descending into low-level detail. The quantity will vary from one service to another, but it is suggested that it should be possible to achieve good coverage with less than 100 scenarios, and the tool will not permit the creation of more than 100 scenarios.

The SARM Quality Model is a very useful aid to achieving the right level of scenario coverage. Each scenario should be associated with a quality sub-characteristic from the model, and absence of scenarios associated with some sub-characteristics can be an indication that there are gaps in the set of scenarios.

A recommended way of using the model is to leave it out of discussions at the start of the workshop. Enthusiastic workshop participants will be full of ideas about what the planned service should achieve, and how it should work. These can be documented by the facilitator in an initial 'brain-storming' session. Once the ideas begin to run dry, the group can look at the Quality Model, and revisit the scenarios that have been identified, allocating each scenario to the sub-characteristic in the model that is most strongly represented by the scenario.

It is common that many of the initial scenarios captured focus on the most visible aspects of the service. So the allocated sub-characteristics will tend to be drawn from the Functionality and Usability characteristics.

The facilitator can then invite the team to consider all of the other sub-characteristics and determine whether they are not relevant to the service in question, or whether there are some significant requirements that could be described in one or more scenarios, but that were not previously considered. The model has thus become a prompt for some important, but less obvious, requirements. After working through the Quality Model and considering again the set of scenarios, it is not uncommon for the number of scenarios documented to have doubled from that captured during the initial brain-storm.

5.11 Example scenarios

Returning to our 'Journey to work' example, the initial brain-storm might identify the following scenarios:

1. Quick journey – using the minimum amount of time to accomplish the journey in each direction.

2. Cost – minimising the overall cost of the journey.
3. Comfort – an enjoyable and comfortable experience during the journey.
4. Car sharing – the family has only one car, which ideally can be used by more than one member of the family.
5. Minimise delays – a predictable journey is highly desirable, with disruptions and delays being a rarity, rather than a frequent occurrence.

A review of the Quality Model would likely allocate these five scenarios to the following sub-characteristics respectively: Time Behaviour, Resource Utilisation, Attractiveness, Suitability, Fault Tolerance.

Further examination of the Quality Model might also draw out some additional scenarios, associated with quality sub-characteristics that are significant, but were not among those uppermost in the minds of the participants during the initial brain-storm:

6. Avoid Accidents – minimise the likelihood of being involved in an accident, associated with the Security sub-characteristic.
7. Flexibility – being able to cater for changed requirements due to a sudden change in circumstances, associated with the Changeability sub-characteristic.

This worked example has been purposely limited to less than ten scenarios, for the sake of simplicity. Description of a real service might result in 40 or more scenarios being documented and classified.

The final aspect of scenario classification relates to the 'impact' of each scenario. It is recommended that the team agree on the list of all scenarios first before going back over each scenario to discuss, and agree, on each one's level of impact. The 'impact' under discussion for each scenario is the impact on the resulting service if the solution deployed fails to satisfactorily achieve that

scenario. The team should discuss then agree an impact classification for each scenario, using the scale shown in Table 9 above, from the SARM Risk Model.

Returning to our example, Table 10 shows the classification of impact for each of the seven scenarios identified:

Scenario	Impact
Quick Journey	Major
Comfortable Journey	Moderate
Minimise Cost	Minor
Ability to share car with family	Moderate
Minimise delays	Moderate
Avoid accidents	Catastrophic
Flexibility	Moderate

Table 10 Scenario impact classification

The 'Scenarios and Stakeholders' worksheet of the SARM Tool facilitates the documentation of the scenarios and their associated classifications according to the SARM Quality and Risk Models. When entering scenarios in the tool, you should consider whether some scenarios are really composites that should be further broken down. In our example above, some of the scenarios relate to characteristics of the journey in each direction. If some of the solution options propose different modes of transport for the journeys to and from work, then these scenarios should be split to allow for the possibility that some scenarios may satisfy those characteristics differently for both journeys. To allow for this possibility, we have split the 'Minimise delays' and 'Comfortable Journey' scenarios into 'to work' and 'from work' scenarios, resulting in a total of nine scenarios.

Figure 9 shows how they appear once they have been documented in the SARM tool:

	A	B	C	D	E	F	G	H
				Me	Family	Boss	Colleagues	
4	Time Behaviour	Quick Journey	Major					
5	Fault Tolerance	Predictable journey to work - minimal delays	Moderate					
6	Attractiveness	Comfortable journey to work	Moderate					
7	Resource Utilisation	Minimise cost	Minor					
8	Security	No accidents	Catastrophic					
9	Changeability	Flexibility	Moderate					
10	Suitability	Ability to share car with family	Moderate					
11	Fault Tolerance	Predictable journey home - minimal delays	Moderate					
12	Attractiveness	Comfortable journey home	Moderate					

Figure 9 Scenarios & Stakeholders worksheet

1c – Linking the scenarios to the stakeholders

It is highly likely that some of the scenarios that have been identified will be more important to some of the stakeholders than to others. This suggests that when the inevitable compromises involved in a real implementation are considered, some stakeholders may be less satisfied with the result, if a scenario that they are particularly interested in is not satisfactorily achieved.

SARM allows us to consider such possibilities during the trade-off analysis when considering competing solution designs. But to be able to do this, we need to understand the level of interest that each stakeholder group has in each scenario. Identifying this is the next step of the SARM process.

Each scenario should be considered from the perspective of each stakeholder group, and the level of interest that stakeholder group has in the scenario under consideration should be determined. Three levels of interest are used in SARM – 'None', 'Interest' and 'Strong Interest'. These will be used later by the tool to calculate the burden of risk that each solution design represents to that stakeholder group – where there is a high risk that a solution might fail to achieve a given scenario, this will be of greater significance to the stakeholder

group when they have a stronger interest in that scenario.

The team that developed the scenarios can together reach consensus on the appropriate levels of interest of each stakeholder group in each scenario. Alternatively, if the project team has access to representatives of each stakeholder group, they can be asked individually to assess their level of interest in each scenario, and the results can be collated and documented in the tool.

Each stakeholder group has a dedicated column in the “Scenarios & Stakeholders” worksheet, allowing the user to select the chosen level of interest for each documented scenario in the column. Figure 10 shows this completed for our example:

Sub-characteristics	Scenarios	Impact	Me	Family	Boss	Colleagues
Time Behaviour	Quick Journey	Major	Strong Interest	Strong Interest		
Fault Tolerance	Predictable journey to work - minimal delays	Moderate	Strong Interest		Strong Interest	Interest
Attractiveness	Comfortable journey to work	Moderate	Strong Interest	Interest		Strong Interest
Resource Utilisation	Minimise cost	Minor	Strong Interest	Strong Interest		
Security	No accidents	Catastrophic	Strong Interest	Strong Interest	Interest	Interest
Changeability	Flexibility	Moderate	Strong Interest	Interest	Interest	
Suitability	Ability to share car with family	Moderate	Interest	Strong Interest		
Fault Tolerance	Predictable journey home - minimal delays	Moderate	Strong Interest	Strong Interest		
Attractiveness	Comfortable journey home	Moderate	Strong Interest	Strong Interest		

Figure 10 Completed Scenarios & Stakeholders worksheet

For example, the project team here have concluded that whilst the commuter and his or her family have a strong interest in a quick journey, maximising the time they can spend together, the work colleagues and boss have no particular interest in this scenario, though they do share an interest in the commuter arriving at work on time.

1d – Review for coherence and completeness

This simple example is easy enough to review for coherence and completeness – we have just nine scenarios and four stakeholder groups, and even the resulting matrix, consisting of thirty-six cells, can be viewed on a single page as a whole. But what if we are dealing with a much more complex service, with perhaps up to a hundred scenarios, and as many as fifteen stakeholder groups? The resulting matrix would contain 1,500 cells. How does one answer the question “does this look right?”?

As we have classified each scenario according to the SARM Quality Model, we are able to draw together the stated levels of interest of each stakeholder in the scenarios that share the same quality characteristic. Since there are only six quality characteristics in the model, this information can be presented in a visual form that is much easier to interpret than a huge matrix. The next worksheet in the tool, entitled “Stakeholder Views” does just that. Figure 11 shows this for our worked example:

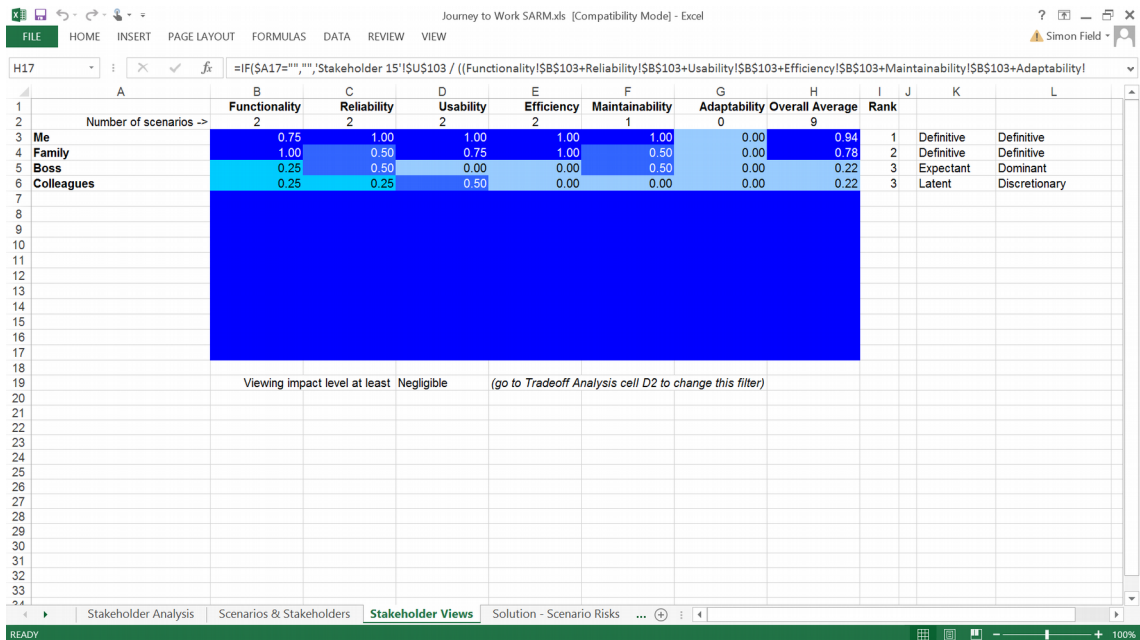


Figure 11 Stakeholder Views worksheet

Row 2 indicates the number of scenarios that have been defined for each of the quality characteristics in the Quality Model. We can clearly see in this example that the Adaptability characteristic is of no importance to this project.

The remaining cells in columns B to G represent the levels of interest that each stakeholder group has in each of the model's quality characteristics, on a scale of 0.0 to 1.0. A weighted model is used, where “No Interest” in a particular scenario carries a weight of 0.0, “Interest” carries a weight of 1.0, and “Strong Interest” carries a weight of 2.0. An average score for each quality characteristic is calculated on a scale of 0.0 to 1.0, using the following formula:

$$Interest_{jk} = \frac{\sum_{i=1}^{n_j} Weight_{ijk}}{2n_j}$$

where:

- i = a scenario
- j = a characteristic
- k = a stakeholder

So for the aggregate interest to have a score of 1.0 in a cell, that stakeholder group must have a 'Strong Interest' in each of the scenarios associated with that quality characteristic. Each cell therefore represents the relative interest that the stakeholder group has in each quality characteristic. Colour coding helps the user interpret this information, with the lightest shade of blue being associated with the lowest level of interest, the darkest, strongest blue with the highest level of interest.

Reading the table by rows, the user can see which characteristics are of greatest importance to each stakeholder group – the commuter's family, in this example, being most interested in Functionality and Efficiency, while the commuter's boss is most interested in Reliability and Maintainability. This information can be used by the project team to verify their understanding of each of the stakeholders and their likely priorities.

Viewed by columns, the table reveals which stakeholders have the greatest and least interest in each of the Quality Characteristics. This view, again, can help the project team verify that the information they have collected as a whole reflects their knowledge of the stakeholders and their likely interests in the service.

Column G contains an overall 'level of interest' score. Using the same weighted approach, each cell represents the overall level of interest that stakeholder has across the whole set of scenarios. The formula for its calculation is:

$$Interest_k = \frac{\sum_{i=1}^n Weight_{ik}}{2n}$$

where:

i = a scenario

k = a stakeholder

Again, a score of 1.0 here would indicate that the stakeholder group has a strong interest in every scenario, while a score of 0.0 means that the stakeholder group has no interest in any of the scenarios. The same colour coding for the cells is applied to ease visual interpretation.

Column I shows the ranking of the stakeholder groups by overall average interest, and the remaining two columns re-state the stakeholder salience and class respectively from the earlier stakeholder analysis. Reading these columns together can further help the project team to verify the information that has been captured on the 'Scenarios & Stakeholders' worksheet. Anomalies should be investigated and a satisfactory explanation sought, as they may indicate that some important scenarios have not been captured, or that the stakeholders' interests have been incorrectly classified.

For example, it is suggested that there is likely to be a correlation between the stakeholder salience and the ranking in Column I. One would generally expect Definitive stakeholders to have the greatest level of interest across the set of scenarios (and therefore be among the highest ranked), and latent stakeholders to be among the lowest ranked.

5.12 Creation of competing solution options

We have now completed Part 1 of the SARM process, and the context in which the later trade-off analysis will take place is now complete. In addition to gathering information that will be used in the trade-off analysis, the Part 1 steps

have helped the project team to gain a complete understanding of the service's high level requirements, and the differing perspectives of the service's stakeholders. If the project team will be involved in developing one or more solution option, then this information should greatly assist them in creating good designs. If, on the other hand, the team plans to outsource the design task to one or more design agency, the information gathered can usefully contribute to the creation of a high quality Request for Proposal (RfP).

As stated in the introduction to the method, the actual creation or commissioning of solution options falls outside the scope of SARM. The next part of SARM requires there to exist one or more solution option that can be subjected to the trade-off analysis using the context that was created in Part 1. It is therefore likely that some time will elapse between the completion of Part 1 and the commencement of Part 2, as solution options are developed to sufficient detail to participate in the trade-off analysis.

It is important to recognise that this does not require a highly detailed service design for each option. What is being evaluated is the architecture of each option and the degree to which it is likely to satisfy the quality characteristics that are important for this particular service. As long as the overall service architecture is understood and can be described, then no more detail needs to be developed until a preferred solution option has been chosen.

In our example, four solution options are proposed:

1. Car – the commuter can drive the family car directly from home to work, parking the car at his or her place of work, then drive home in the evening.
2. Bus / Metro – a bus service exists departing near the commuter's home that will take him or her from home to a metro station, and a ride on the metro completes the journey to work. The same process in reverse makes up the return journey home after work.

3. Car / Metro - the Metro station also offers free parking, so the commuter could drive from home to the Metro station instead of taking the bus, then complete the journey to work with the Metro.
4. Lift / Metro / Bus – a neighbour of the commuter works near the Metro station and drives there every day, and has offered the commuter a lift. This avoids the need to take the bus in the morning, but as both parties often finish work at different times, the commuter will have to rely on the Metro and the bus to return home in the evening.

5.13 Part 2 – trade-off analysis

Part 2 of SARM involves two steps: assessing the risks associated with each solution option, and the trade-off analysis itself. It is suggested that both of these steps can be completed in a workshop involving the same group of stakeholders that came together to create the context in Part 1. The workshop duration will vary substantially depending on the size and complexity of the service, and the number of solution options under consideration (the tool allows for a maximum of six solution options). It may be necessary to split the workshop across more than one day.

As with any workshop involving many participants, it can be valuable for this workshop to be run by a trained facilitator. This will help ensure that each participant makes the most valuable contribution possible, and ensures that the workshop remains on track to achieving its objectives.

Experience has also shown that completion of the SARM trade-off analysis should not be used to stifle discussion during the workshop. Rather, the trade-off analysis should be used to stimulate discussion and foster understanding. Such discussion might produce unexpected but important results – such as the discovery of missing requirements, or the generation of a new, perhaps hybrid, solution option that might be more attractive than any of those presented at the outset. New risks, or means of mitigating risks, might also emerge. It is important that the facilitator and fellow participants are open to such valuable

outcomes, and are ready to capture these should they occur.

The workshop should open with the presentation of each of the solution options. The aim is to ensure that each participant understands the architecture of each solution design sufficiently to be able to judge its ability to satisfactorily achieve each scenario. Details of each design may have been circulated to each participant ahead of the workshop.

2a - Scenario risk assessment

Once the competing solution options have been described, the team should now consider the ability of each solution option to satisfactorily achieve each scenario. As this information forms an essential part of the risk calculations that inform the trade-off analysis, it is important that these assessments are captured directly in the tool, using the “Solution – Scenario Risks” worksheet.

The names of up to six solution options can be entered in row 2 of the worksheet, columns B to G. The team are now asked to reach a consensus on the following question for each solution option / scenario combination:

What is the likelihood that this solution option will satisfactorily achieve this scenario?

The five possible answers are drawn from the Risk Likelihood descriptions in the SARM Risk Model (see Table 8).

It is recognised that this question will be asked, discussed, then answered, many times (the number of scenarios multiplied by the number of solution options, to be precise). It is only by going to this level of detail that the particular strengths and weaknesses of each solution option will be uncovered, and then revealed in the overall context of the service in the subsequent trade-off analysis.

Regarding the sequence in which these questions should be addressed, it has generally been found easier to work across each row before moving on to the

next row (as opposed to completing one column before moving on to the next). This is because it helps the team to focus on the relative differences between the competing solution options. Much valuable discussion often occurs as each question is considered, sometimes shedding new light on various aspects of the service under consideration.

Scenarios	Car Likelihood	Bus / Metro Likelihood	Car / Metro Likelihood	Lift / Metro / Bus Likelihood
Quick Journey	Almost certain	Unlikely	Likely	Possible
Predictable journey to work - minimal delays	Likely	Likely	Almost certain	Almost certain
Comfortable journey to work	Almost certain	Rare	Possible	Unlikely
Minimise cost	Unlikely	Likely	Possible	Almost certain
No accidents	Likely	Almost certain	Almost certain	Almost certain
Flexibility	Almost certain	Possible	Likely	Possible
Ability to share car with family	Rare	Almost certain	Possible	Almost certain
Predictable journey home - minimal delays	Likely	Likely	Almost certain	Likely
Comfortable journey home	Almost certain	Rare	Possible	Rare
	Possible	Possible	Possible	Possible
	Possible	Possible	Possible	Possible
	Possible	Possible	Possible	Possible
	Possible	Possible	Possible	Possible
	Possible	Possible	Possible	Possible

Figure 12 Solution – Scenario Risks worksheet

Figure 12 shows the completed “Solution – Scenario Risks” worksheet for our example. It can be seen, for example, that the Car option is “Almost Certain” to achieve a quick journey, but will rarely facilitate the “Ability to share car with family” (because during the day, the car will be parked at the commuter's office). Whereas the Car / Metro option is less certain to produce a quick journey (though still 'Likely'), but since the car will be parked at the Metro station, just a bus ride from home, family sharing of the car becomes 'Possible'. Inevitably, much debate is often needed before a consensus among participants emerges to be able to answer the key question “how likely is it that this solution will satisfactorily achieve this scenario?”.

One might ask whether consensus is really necessary. Might it be possible to allow each participant an independent answer to these questions, and then

merge their answers using some mathematical formula? The theoretical answer to this question is, of course, 'yes'. A scheme could be devised to cater for differences of opinion. However, a major purpose of the workshop is to foster broader mutual understanding; of the different solution options, and of the various stakeholder perspectives of the required service and the competing solution options. Rather than allow for the persistence of alternative views, it is suggested that a valuable contribution of the SARM approach is to drive the project team and representative stakeholders towards greater mutual understanding and consensus. Experience has shown that, provided the architecture of each solution option can be clearly articulated, it is not difficult to arrive at a consensus for this critical question for each scenario / solution option combination.

2b - trade-off analysis

Completion of the 'Solution – Scenario Risks' worksheet allows us to move on to the core output of the SARM process, the trade-off analysis. In this, the tool takes centre stage, as it has automatically calculated a risk level for each scenario / solution option combination, and aggregated these to present an analysis from a number of perspectives: the quality characteristics from the SARM Quality Model, the Stakeholder Salience and Classes from the Stakeholder Model and each of the individual stakeholder groups. The 'Tradeoff Analysis' worksheet therefore needs to be visible to every workshop participant.

It is important that a SARM practitioner fully understands how each number in the SARM tool has been calculated. Risk exposure values that have been calculated for individual scenario / solution options have been aggregated, and the method of aggregation has inevitably involved choices that affect the nature of the outcome. This next section will therefore discuss in detail the way in which the tool has been programmed.

We will start with two screen-shots (Figure 13 and Figure 14), which between them show the whole of the 'Tradeoff Analysis' worksheet for our example:

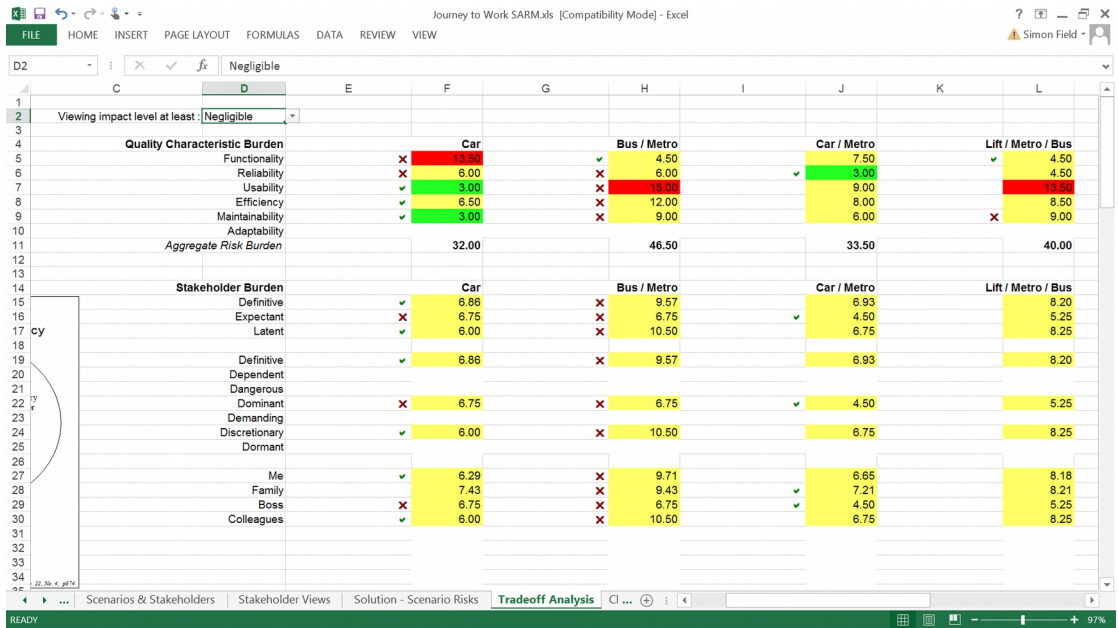


Figure 13 Tradeoff Analysis worksheet (top)

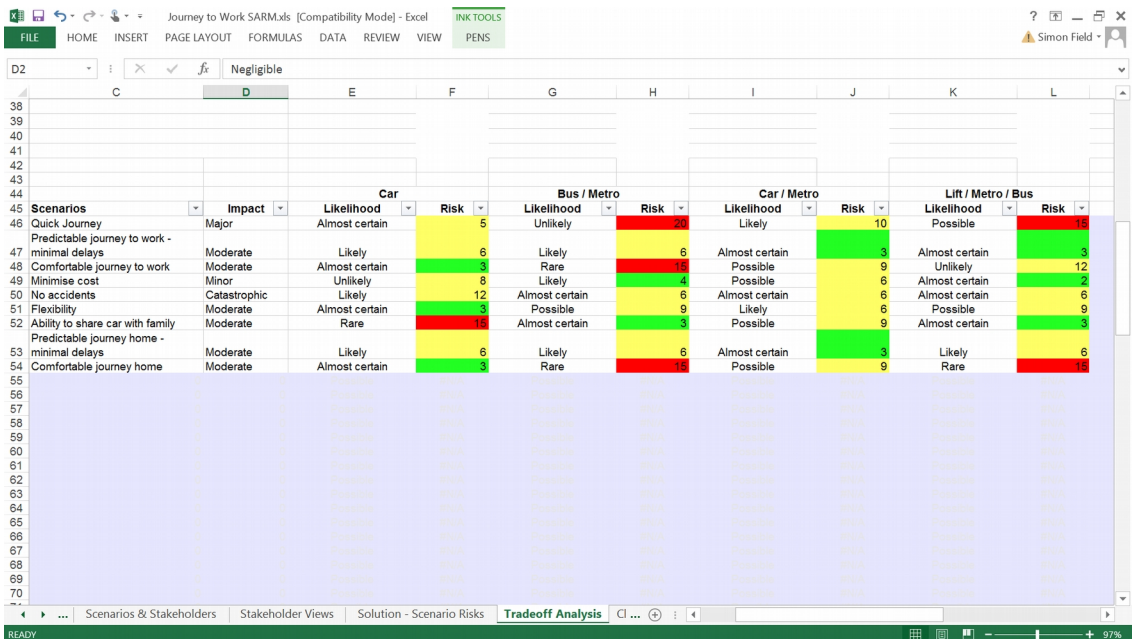


Figure 14 Tradeoff Analysis worksheet (bottom)

The lower journey portion of the worksheet, shown in Figure 14, shows the risk exposure score for each scenario / solution option combination. Each score, shown in the 'Risk' columns for each solution option, has been calculated according to the SARM Risk Model described earlier (as illustrated in Figure 4 above). The most observant reader may notice that the Likelihood question that was asked in order to complete the Scenario – Risk Assessment on the

previous worksheet was not about the likelihood of the risk occurring (i.e. the likelihood of the solution failing to achieve the scenario), but about the likelihood of the risk *not occurring* (i.e. the likelihood of the solution achieving the scenario). This is the value (the likelihood of success) that is shown in the 'Likelihood' column for each solution option. The risk likelihood is therefore the inverse of this answer, so the team's answers have been converted to risk likelihoods using the table below (Table 11), and then the risk exposure has been calculated and colour coded using the SARM Risk Model as shown in Figure 4 above. Readers should remember that a higher number means a higher level of risk that the scenario will not be achieved.

Likelihood of Success	Risk Likelihood
Almost Certain	Rare
Likely	Unlikely
Possible	Possible
Unlikely	Likely
Rare	Almost Certain

Table 11 Inverting the Likelihood

We will now explain the aggregated risk information contained in the first screen-shot above. Row 2 contains a user-controlled filter that allows the user to select a minimum risk impact level. By default, the worksheet will show data relating to all scenarios with assigned impact levels of any value (from 'Negligible' to 'Catastrophic'). However, it can sometimes be useful for the project team to consider the data relating to only those scenarios with a higher assigned level of impact. The cell 'D2' allows the user to apply a filter that will remove data from the worksheet relating to all impact levels below that specified.

5.14 Quality characteristic burden

Rows 4 to 10 of this worksheet show scenario risk data aggregated by quality characteristic. The numeric values in the columns for each solution option are the average of the risk exposure scores for those scenarios that have been

allocated to sub-characteristics that belong to each of the six quality characteristics in the SARM Quality Model (Table 7 above).

So in our example, the 'Car' scenario has an average Risk Exposure Score of 13.5 for the 'Functionality' quality characteristic, and an average Risk Exposure Score of 3.0 for 'Usability'. In contrast, the 'Bus / Metro' scenario has an average score of 4.5 for 'Functionality', but 15.0 for 'Usability'. These illustrate a clear trade-off in terms of quality between the two solutions – the 'Car' scenario sacrifices Functionality but achieves low risk in Usability, while the 'Bus / Metro' scenario achieves the opposite effect. Which of the two is the most suitable service design depends on the relative importance of the two quality characteristics, the details of other trade-offs that may exist within the solutions, and the ease with which some of the higher risks could be mitigated.

It is important to note that the tool, and the overall SARM process, is not advising the project team which solution option is 'best'. Indeed, there is no clear definition of 'best', given the multi-dimensional nature of service quality and the many service stakeholders. But the tool is drawing the attention of the project team to the differing areas of risk that will be borne if a particular solution option is chosen. The aim of this worksheet is to foster discussion about the many trade-offs that exist, among quality characteristics, among stakeholders and among individual scenarios, for each of the solution options. It is then up to the project team to choose a preferred solution option by determining which solution option represents the most acceptable compromise.

The risk burden of a quality characteristic for a given solution option is therefore calculated as follows:

$$Burden_{hj} = \frac{\sum_{i=1}^{n_h} Risk_{hij}}{n_h}$$

where:

- h = a characteristic
- i = a scenario
- j = a solution

It should be noted that this is an unweighted average, and so it treats each scenario as having equal importance to other scenarios that share the same parent characteristic.

As with the individual scenario risk exposure scores, the average risk exposure scores are also colour coded to aid visual interpretation. Average risk scores might not be integer values, and so the colour coding shown for individual risks in Figure 4 above has been adapted as follows for all of the aggregate risk scores in the worksheet as shown in Figure 15:

Average Risk Score
< 4.5
>= 4.5 and < 12.5
>= 12.5

Figure 15 Average Risk Score Colour Coding

To the left of each average risk score cell there may appear a green tick or a red cross. The green tick indicates that the neighbouring score has the lowest risk score among the solution options for that particular quality characteristic, while the red cross indicates the highest risk score. Returning to our example, we can see that the 'Car / Metro' option carries the lowest average level of risk for achieving the 'Reliability' scenarios, while the 'Car' and 'Bus / Metro' solutions share the highest level of risk for 'Reliability'.

5.15 The Charts worksheet

Whilst the upper part of the 'Tradeoff Analysis' worksheet gives a valuable 'snapshot' view, allowing the project team to compare the solution options in terms of their ability to satisfy the six Quality Characteristics, the nature of an average means that extreme individual scores are hidden, and sharper differences that may be clearly apparent at the individual score level will tend to diminish when brought together in an average value. The neighbouring 'Charts' worksheet in the tool is designed to help project teams explore the composition

of the average risk scores by quality characteristic in more detail. It shows the distribution of scenario risk scores by colour coding so that, for example, an average score that is composed of extreme values can be distinguished from one composed of moderate values even if they share similar overall values.

Figure 16, taken from our example, illustrates this:

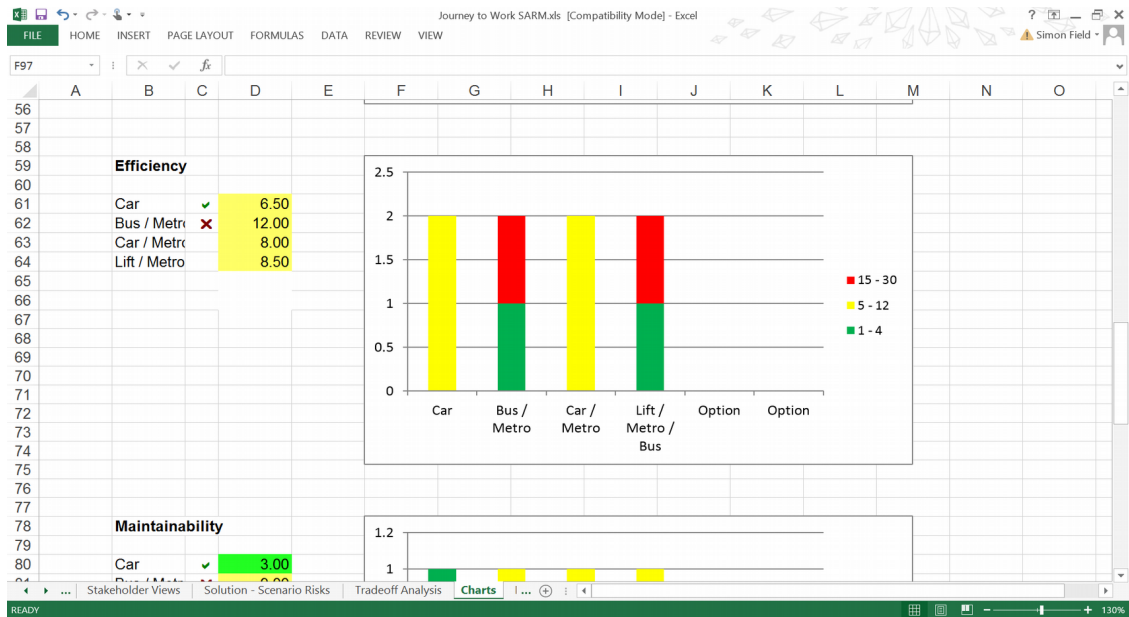


Figure 16 Charts worksheet

We can see that for 'Efficiency' the 'Car / Metro' and 'Lift / Metro / Bus' solutions have very similar average risk scores (8.0 and 8.5 respectively). However, their compositions are very different, with the 'Car / Metro' being composed of two moderate risks (both coloured Amber), and the 'Lift / Metro / Bus' containing one low risk score (Green) and one high risk score (Red). Use of this worksheet can help the workshop participants to 'dig deeper' to understand what has triggered a particular level of risk. In this case, they might return to the lower portion of the 'Tradeoff Analysis' worksheet to look more closely at the individual scenarios concerned to consider how these risks might be mitigated.

5.16 Aggregate Risk Burden

Returning to our examination of the upper portion of the 'Tradeoff Analysis'

worksheet, row 11 contains a total score for each solution option entitled 'Aggregate Risk Burden'. This is a simple summation of the average risk scores of each Quality Characteristic. Whilst it might be considered to represent an 'overall score', with a lower score representing a lower overall level of risk, it should be treated with some considerable degree of caution by the workshop participants. It facilitates an overall comparison of the solution options on the basis of treating each quality characteristic as being of equal importance to the others. There is no weighting provided for individual quality characteristics – each effectively has an equal weight, so for example, Functionality is considered to be of equal importance as Efficiency. This is the case even if the Functionality score consists of many scenarios and Efficiency has just one scenario. The value and applicability of this information will vary from one application of SARM to another, and should be discussed and considered by the workshop participants.

5.17 Stakeholder burden

The remaining section of the 'Tradeoff Analysis' worksheet relates to the distribution of risk by stakeholder salience, class and individual stakeholder groups. In the same way that average risk scores have been calculated for each quality characteristic, so average risk scores can be calculated for stakeholder groups, and for their groupings according to the Stakeholder Typology (see Figure 3).

We will consider first the average risk burden for individual stakeholder groups (shown in rows 27 to 30 in our worked example). The calculation of an average by stakeholder group has to reflect the level of interest each stakeholder has in the various scenarios. A solution that contains a high risk of failure for a particular scenario will be of concern only to those stakeholders that have an interest in that particular scenario. It will be remembered that stakeholder interest in each scenario was classified using the 'Scenarios & Stakeholders' worksheet, and verified with the 'Stakeholder Views' worksheet. One of three different levels of interest were recorded for each stakeholder group in respect

of each scenario: 'No Interest', 'Interest' or 'Strong Interest'. In the 'Stakeholder Views' table, weights of 0.0, 1.0 and 2.0 were applied respectively to these levels of interest. The same weights are now applied to produce a weighted average risk score (known as the 'Stakeholder Burden') for each solution option for each stakeholder group.

The formal definition of this score is given below:

$$Burden_{jk} = \frac{\sum_{i=1}^n Risk_{ij} Weight_{ik}}{\sum_{i=1}^n Weight_{ik}}$$

where:

- i = a scenario
- j = a solution
- k = a stakeholder

Rows 15 to 17 of the worksheet show the average of these scores for all stakeholders that share the same level of stakeholder salience according to the stakeholder analysis conducted during Phase 1 of SARM, while rows 19 to 25 show the average of these scores by stakeholder class.

In our worked example, which only contains four stakeholder groups, these groupings of stakeholder groups are of limited value, but in a project where there is a larger number of stakeholder groups, these higher level views may be of significant value.

Whilst the stakeholder views may not be the primary determinant of the preferred solution option, they do draw attention to which stakeholder groups are most likely to benefit from, or suffer disadvantage from, the various solution options. It is interesting to note in our worked example that the commuter and his or her work colleagues would appear to favour the 'Car' commuting option, while the commuter's family and boss both favour the 'Car / Metro' option. Exploration of the individual scenario risks can help uncover the detailed reasons for this, and perhaps be used to consider how the highest levels of risk

might be mitigated.

Whatever solution is adopted, it is likely that the benefits for some stakeholders will be put at risk as a consequence of the inevitable compromises that make up that particular solution. This part of the 'Tradeoff Analysis' worksheet can be used to anticipate future stakeholder concerns as the new service is deployed, enabling the project team to consider in advance how it will manage stakeholder relations further into the project.

5.18 Reaching a conclusion

The trade-off analysis step in Phase 2 of the SARM process brings the overall SARM process to a conclusion. It may seem strange to the reader that this guide does not contain clear instructions on how to arrive at a recommended solution design. Each service design project is different, but all have many different dimensions and perspectives. Selection of a preferred design cannot be automated, though the tool, and especially its 'Trade-off Analysis' worksheet and accompanying 'Charts' worksheet, can shed light on all corners of the problem space and the extent to which each competing solution option fills it, or leaves gaps.

The latter part of the workshop will inevitably follow an individual course, depending on the questions that are raised as participants explore the risks and the various trade-offs that appear among the solutions, stakeholders, quality characteristics and scenarios. Only humans can decide which residual risks are acceptable, and can be borne or mitigated in some way.

In our experience, there is a story that underpins every solution option. The numbers and colours of the trade-off analysis in themselves represent dry facts, but they are there to reveal the highlights of the story. It is the job of the workshop participants to uncover the stories that lie behind the numbers for each solution option, bring those stories to life, then determine together which of those stories is most appealing in the context of the service under consideration.

Chapter 6: Service Design Case Study

6.1 Introduction

Chapter 3 has described the circumstances in which the UK Border Agency (UKBA) was chosen as the source of the case study for this research project. At the time of this research project, the UKBA was reviewing the design of its asylum service, with particular emphasis on the part of the service that deals with appeals. It was the review of this part of the service that was chosen for the case study, including the evaluation of new candidate service designs that might represent an improvement over the present service design. This chapter describes how the case study was conducted. The content of the eight interviews conducted with the participants from UKBA will be woven into the narration, providing a commentary on the effects of each phase of SARM as it was applied at UKBA. The interviewees will be referenced with 'I' followed by a number, so that they can be distinguished from each other while maintaining their anonymity. The only exception to this is where the Project Leader's comments clearly identify himself or herself as the project leader. So the Project Leader's quotations are sometimes referenced 'Project Leader', and sometimes as an anonymous numbered interviewee.

As described in Chapter 4, SARM did not “spring fully formed” as it is presented in Chapter 5. The final form presented there was the result of the initial thinking presented in Chapter 2, with further refinements that took place during and following the case study. Each refinement will be described in the appropriate section of this chapter, along with the 'trigger idea' or thought that led to the change.

6.2 Collecting evidence at the workshops

The project leader at UKBA was briefed ahead of the commencement of the project, so that he fully understood SARM, the requirements the method placed on the project, its participants and stakeholders. And of course, he was able to assure himself that the approach would help him achieve his aims for the project. Once that had been accomplished, the first workshop was planned,

with the project leader determining the workshop participants in consultation with the lead researcher.

The first workshop in SARM, as described in Chapter 5, generates a number of tangible assets; the stakeholder analysis, the list and classification of the scenarios, the team's impact assessment of each scenario and their view on the importance of each scenario to each stakeholder. In the software tool, these correspond to the first three spreadsheet tabs: 'Stakeholder Analysis', 'Scenarios & Stakeholders' and 'Stakeholder Views', as described in Chapter 5. There is also the opportunity to observe the workshop, capturing the interaction among the participants, the levels of agreement and disagreement, and the different levels of engagement of the participants.

From a research perspective, it might therefore be an attractive proposition to film or record the workshop. This evidence could then subsequently be reviewed in great detail, reducing the burden on the researcher during the actual workshop. However, there are risks associated with such recording. It would have to be done with the consent of the participants, and as has been seen earlier, the method seeks to draw upon a range of participants in various positions within an organisation's hierarchy. Might the capture of a 'permanent' record of the conversations and behaviours of participants inhibit their participation? The subject matter itself, dealing with the way in which asylum seekers and their cases are managed, is a highly sensitive topic. The confidentiality of plans to change such processes must be respected prior to their publication through appropriate channels following official approval. In consideration of these matters, the project leader and his management decided that no recording, either video or audio, would take place during the workshops.

This placed a greater burden on observation during the workshops, and raised a further issue to be addressed before commencing the project. Who would facilitate the workshops?

The workshops need to be facilitated by someone who is familiar with SARM, and has a deep understanding of the concepts that underpin the method. As

this was the first application of the method, the number of candidates for this role was severely limited, and the most obvious candidate was the lead researcher (the author) who developed SARM. However, he was also the most obvious candidate to lead the observation of the workshop, and there are clear disadvantages in being forced to adopt a “participant-observation” approach throughout the workshops.

Fortunately the project was able to draw on someone with considerable experience of facilitating ATAM reviews (the method from which SARM has been derived), a colleague of the lead researcher at ONS, who had also been close to the development of the extensions to ATAM described in Chapter 2 that have been adopted with SARM. His introduction to the project as facilitator allowed the lead researcher to take the role of passive observer for most of the duration of the workshops.

6.3 The first workshop

The first workshop duly took place on the morning of Thursday, 6th May 2010. In addition to the lead researcher (the author), and the facilitator, both of whom were from ONS, the five other attendees were all from UK Border Agency. They consisted of the Project Leader, who was also the manager with overall responsibility for the service in question, two managers of teams that were directly responsible for dealing with asylum cases, one 'Case Owner' (a member of such a team), and a Presenting Officer. This last role involves presenting cases in court on behalf of UK Border Agency.

During the planning of this workshop, the Project Leader had indicated that it was rare, and indeed outside their experience, for a service design activity such as this to involve representation of a range of stakeholders involved in the delivery of the service. Much more common was the design of new processes by management, and subsequent consultation with staff once the new process had been approved. However, the Project Leader had obtained very strong support from higher management for this more open and collaborative approach to service design.

The workshop commenced with a short introduction from each participant, and an outline of the overall project's goals given by the Project Leader. The Project Leader explained the decision to adopt an experimental approach to Service Design for this project, and introduced the Lead Researcher and his colleague from ONS, the facilitator.

They then gave an overview of the research project and sought, and obtained, permission to interview each participant about their experiences in the project some time after the second workshop (once the project had selected a preferred service design).

The Service Quality Model was introduced, with distribution of a laminated copy of the model on a two-sided A4 sheet (see Table 7 The SARM Quality Model and its accompanying description in Chapter 5). The Lead Researcher then described the overall process, covering both workshops and demonstrating the spreadsheet tool that would be used throughout. This was illustrated with a simple worked example process, the topic being 'Planning a day trip to Barry Island'. The example is intended to be simple enough for the application of the quality and risk models to be easily understood, and remote enough from the project in hand to allow participants to concentrate on understanding the concepts and design evaluation method (rather than worrying about the validity of the example service designs). The competing designs involve different means of travelling to and from a day at the beach in Wales. The example presented was similar in character, though different in detail, to the "travel to work" example presented in Chapter 5.

Inevitably, many questions were asked and answered, and a typical comment at the conclusion of this part of the workshop was "I get the idea, but I'm not quite sure how this will work precisely".

"I liked the way you started it off with that trip to Barry Island, that put it all into..., well it helped to understand where you were going, and why you were saying what you were saying." (I3)

All participants expressed enthusiasm to get started. The workshop had consumed approximately 45 minutes up to this point, and the Facilitator now took over the conduct of the remainder of the meeting.

For this next part of the workshop, the Facilitator led a 'brain storming' session to capture the main stakeholders. One of the participants took the role of 'scribe' on the flip chart, and it was agreed to separate consideration of stakeholders that fell within UKBA, and those that could be considered to lie outside (labelled 'Internal' and 'External' respectively). Separate sheets were used for each group, and all members of the team were actively engaged in proposing and discussing stakeholders. There was no disagreement among the team, who collectively identified thirteen internal stakeholders, and fifteen external stakeholders. There was some discussion at the large number, which were identified very quickly and occasional discussion of the definitions of "internal" and "external". The possibility of consolidating the list was discussed, but it was agreed that this would be reconsidered in the context of the scenarios (which had yet to be created) at the end of the meeting.

The facilitator turned the team's attention to the need to identify scenarios, and led an initial 'brainstorming' session to capture scenarios that reflected important quality characteristics. There was a rapid and steady flow of ideas, but the content was closer to an expression of issues, problems and symptoms associated with the current service than descriptions of desired "to be" scenarios. About 30 minutes was spent capturing and discussing 18 "issues".

"We quickly got into producing a list of issues. I think that went very well, quite similarly to the stakeholder analysis.... Although it wasn't exactly what the model required, it was useful in informing the development of the scenarios." (I1)

"Yes - it was a bit of a wish list, I think. If I remember rightly, a bit of a wish list of what you'd really like to get from the actual outcome." (I2)

As I1 correctly identified, identification of issues is not exactly what is envisaged

in the method, which seeks to identify scenarios that describe the target service. It was clear that participants were initially uncomfortable imagining a 'to-be' service, but given that the target was an improvement upon the existing service with which they were very familiar, they could all readily discuss shortcomings and issues with that existing service:

“I think it was at that point that I wasn't entirely certain if we were going in the right direction in terms of what the model required.” (I1)

So it became necessary to start by documenting issues, and to use these to generate scenarios for the 'to-be' service. To begin the process of turning the issues into scenarios, the team worked through the quality model sub-characteristics, identifying those that are most relevant to the service in question, and linking issues earlier identified to individual sub-characteristics. Now the team felt able to create a set of scenarios, and a total of 21 scenarios were documented, describing 15 sub-characteristics. All six quality characteristics were represented.

“There was a lot of discussion went on about what we wanted to get out of it, what the scenarios were, but I think it was right to do that at that time as well, because when it was all plotted out on the spreadsheets, it just seemed that it was worth taking the time doing that initial thing. Otherwise you could have got lost - indeed we did get a little bit lost where maybe the scenario wasn't quite worded correctly and people were losing it a little bit then again afterwards. So, probably the amount of time we spent in the initial stages gave benefits a bit later on.” (I2)

It was recognised that having a range of stakeholders contribute to the scenarios led to a deeper understanding of what was required than might have been the case had only one or two individuals analysed the problem:

“What was quite interesting there is, even just in that room, although we're all case owners or involved in the asylum process itself, just

coming from a slightly different angle that we all were, everyone had a little bit different to say. So it wasn't just a case of you had, say, six, the top six where what everyone was moaning about; it turned out to be eighteen, because you had that only slightly different mix of people - all doing basically the same job, but slightly differently. And you obviously got eighteen out of it. So there is obviously, there is room for improvement or change of policy or whatever.” (I3)

“It was definitely inclusive. I think everybody felt that they could make their opinions known.” (I5)

The quality model was new to the participants, and it did take a bit of time for them to become familiar with it. However, it was also felt that use of the model helped to develop a deeper understanding of the requirements:

“It's quite a different sort of way of looking at things. You normally look at what you want to do, and look at issues you think might arise, but then to have to align those issues with the characteristics which you wouldn't sort of really think about – subliminally probably you are, but you don't actually label them – that was quite difficult to do, and I think that was probably reflected in the amount of time it took for us to do that part of the process.” (I5)

“And the initial bit, where we had to come up with the scenarios for the characteristics - that was quite difficult but once we started to get the drift with that, that was very [pause] it's just a different way of looking at things. And once you get more used to it, that started to flow more quickly.” (I5)

“Yes, at this point, I think I, and I think we, had a much better understanding of where we were going by this point in time. It made sense to us, I think, once we had done one or two of these, its fairly easy then to getting into doing the others. I seem to recall that part of the meeting being very productive.” (I1)

“Well, it needed to be done. [creating consensus around the construction of scenarios] That's the thing, it had to be, because as you can see, it needed to be brought down into what you wanted it to be brought into these characteristics, otherwise we would have had however many is up there, floating around. So it was very helpful, because also as well you had certain things that all fall under the one characteristic, which helps you to understand how they are all linked together.” (I3)

“I suppose without following the model all the way through, then the initial advantages that I spoke about – i.e. looking at the issues you've got in a lot more detail and then looking at the solutions you've got in a lot more detail, you can't really have one without the other. So the model in that respect was good. I don't think we thought, when we were looking at the scenario, that there was going to be quite as many things to take into account. Maybe, because you've run it on probably much more complicated scenarios, this didn't seem a very big issue for us – we were just looking at something that on paper would look quite a simple scenario – I was quite surprised at how much, when we broke it down, there was to think about. Which probably tells me that generally we don't look at everything we should be looking at when we are trying to design a solution for anything.” (I5)

The model also highlighted the subtle difference between a service provided by a government organisation, and one delivered by a services industry:

“The service quality model is perhaps more designed to systems whereby organisations are providing a certain level of customer service in a more service orientated environment than the system which we operate, although obviously we do have customers within our process. It is not a bank, for example, providing customer service to customers, although like I say we have got customers it's

slightly different in terms of our objective – it may ultimately be that we want to remove our customer from the country and not give them the best service in terms of, for example, a bank the best bank account for that customer.” (I8)

However the same participant acknowledged that this was not a barrier to using the model for this type of service:

“No, I think it worked.” (I8)

The impact level was agreed for each scenario, with consensus quickly established. At most, there was some discussion occasionally about moving up or down one level, but wider disagreement was absent.

“I remember we had some constructive discussion around the level of some of the impacts, but not many, I think we were fairly unanimous about most.” (I1)

“That was straightforward once we had the scenarios against the characteristics – it was quite easy to see whether it was going to impact or not, so that was fine.” (I5)

During the subsequent interviews, one participant (I6) wondered whether selecting one of five impact levels for each scenario was unnecessarily complex. Might three levels (*high, medium, low*) be sufficient? An investigation into this question is beyond the scope of this project, but could be a worthwhile subject for future research:

“So it might be that a level of detail slightly less than this, as you've said already, the more familiar categorisation of risk, might have been more appropriate. And I'm sure it wouldn't have skewed the answers in any meaningful way. We'd still have come to the same solution. Would it have skewed it at all, do you think? Taking it down to just three?” (I6)

The team now returned to the list of stakeholders, and grouped some of them together. A group of stakeholders would be identified if the team felt that several stakeholders would share a common view of the set of scenarios that had been identified. For example, the applicant and their representing solicitor were considered to view the scenarios from a similar perspective, and so these two were brought together in a single group. Out of the 28 stakeholders identified earlier in the workshop, there emerged twelve stakeholder groups, five internal and seven external.

Some of the participants found the stakeholder analysis enlightening:

“I was quite surprised how many there were. Even though most of them we come into contact with on a daily basis, you don't register really how many people you are actually trying to keep happy all the time, so that was kind of the biggest thing.” (I5)

“I think again you realise how many people you are actually dealing with, and what a complex process it is. When you deal with all these people on a daily basis, you don't really think about how many stakeholders are actually involved.” (I2)

All participants found the process of identifying and grouping the stakeholders straight-forward:

“I think that went very well. We quite quickly came to a very comprehensive list of our stakeholders and we just had to do a bit of refinement in grouping them together because there was some overlap and it wouldn't have been possible for us to analyse that many individual stakeholders. I thought that part of the process went very well - everyone contributed well, we covered the full range there, and we got it down to a manageable level quite quickly.” (I1)

“This work seemed to go fine. Obviously we're all dealing with the same sort of stakeholders, so we knew what stakeholders were

around. It was just a case of splitting into internal and external, which was done easily...it gave us good setting up for where we were going to go in the next couple of meetings. I think the first bit was productive, and it put us on the right steady path to go forward with it." (I3)

At this stage in the evolution of SARM, the method did not incorporate the use of a standard stakeholder model. The model shown in Figure 3, and described in some detail in Chapter 5, was introduced after the case study, when the principal researcher recognised that the use of a stakeholder model would mirror the use of the quality model.

The final activity of the first workshop was to assess the level of interest each stakeholder group has in each scenario. With 12 stakeholder groups and 21 scenarios, this would involve a total of 252 decisions. Even though each one only involves determining whether the stakeholder has strong interest, some interest or no interest in a given scenario, and so is a relatively straight-forward decision, the volume of decisions demands considerable time. So this activity was deferred to a separate meeting to be arranged the following week, and the team agreed that it would not be necessary for all members of the team to be present.

This follow-up meeting duly took place on 17th May 2010, with the same facilitator and with the Lead Researcher and just four of the original UKBA team present. The two meeting objectives were to confirm and name the stakeholder groups that had been identified in the first meeting, and to assign levels of interest to each stakeholder group for each scenario, thus completing the activities for the first workshop in SARM.

The stakeholder groups proposed at the previous meeting were quickly confirmed without further change, and each was given an appropriate name. The spreadsheet tool was used to document the stakeholder interest in each scenario, and the team completed this activity in just 40 minutes, with occasional debate, but no difficulty in reaching a decision.

“That was straight forward, yes. Because everyone was going towards the same sort of goals.” (I3)

“That bit was fairly straightforward as well. I mean there were a few where we had a bit of a discussion about, but I seem to remember that that bit was quite quick.” (I5)

“I don't recall there being any difficulty, ...we rattled through it quite quickly.” (I8)

One participant raised an interesting concern: is there a risk that, by examining and discussing the level of interest each stakeholder has in each scenario, the importance of stakeholders might become distorted?

“Maybe there was a potential over-egging of the impact that some of these people have on us... if you have a microscope and you put everything underneath it and move it around, then everything becomes important.” (I4)

These connections are used later in the trade-off analysis step to relate risks associated with each solution option to the particular interests of each stakeholder. It might be possible to avoid the low-level of detail (which participant *I4* suggests might anyhow be somewhat exaggerated) by asking participants just to relate stakeholders to quality characteristics. However, this would make an assumption that a stakeholder that is interested in a particular quality characteristic would have equal interest in all of the scenarios that are aligned to that characteristic. Would such an approach be more, or less accurate than the one adopted in this case study? It would certainly reduce a time-consuming activity, and so it might be worthy of future study to understand the impact of each approach.

At the conclusion of this activity, it was apparent that the exercise had captured a rich view of the stakeholder landscape as it related to the service in question.

However, it was very difficult to “see the wood for the trees”, as this landscape was composed of 252 individual spreadsheet cells. It was agreed that a visual overview of this information could be of value to the project, and would also make it easier to validate the information that had been captured in the workshop. This was the origin of what eventually became the Stakeholder Views worksheet in the SARM tool (see Figure 11 and the accompanying description in Chapter 5).

A prototype version of this worksheet was developed a few days after the workshop, and the output was sent to the project leader. A data visualisation expert was consulted. He advised against a Red, Amber, Green colour-coding scheme, which can appear so colourfully busy that it confuses rather than informs. Instead, a 'heat-map' approach was proposed, with the strength of the colour growing in line with the strength of the relationship between the quality characteristic and the stakeholder. As the strength of the relationship does not signify something positive or negative, a neutral colour was recommended (as opposed to red or green). Blue was chosen, though this choice was mildly criticised by one participant:

“I think the colour scheme is slightly counter-intuitive as a heat map, but I understand why you've chosen to do that because red and so on are not very readable. I'm not sure there is a better way of doing it, to be honest, or what the answer is or what the colour scheme would be but I personally tend to associate blue with cold and a lack of interest. That's a simple thing just to think about in my own mind, isn't it, the stronger the colour, the greater the interest.” (I1)

Overall, participants found this chart helpful in validating the decisions that were made regarding stakeholder interests. For this case study, unfortunately, the chart was not available at the end of the workshop, but by building it into the spreadsheet tool, future users of SARM will be able to review the chart immediately, and if necessary, revisit their earlier decisions.

“This is useful, yes. Its a lot more accessible than the table.” (I1)

“Yes. It's highlighted that efficiency and reliability were key - that's what we were looking for, which is probably right. That's what we want in the process - we want it to be efficient and we want it to be reliable.” (I2)

“So its just an easy way of quickly looking at something and picking out specifics, as opposed to going...if you had to keep looking for 1s all the time, like you said, you become blind to it after a while. You start missing things out - the colours - I think that's a really good way of doing it.” (I3)

“You can see what it's for, but the ranking was the key thing for me. And it reflected really what I expected it to reflect, which was encouraging... And the decision owners, the asylum decision owners, at number one, and the target drivers at number three – I would – it's expected.” (I5)

“It was useful to be able to see the wood for the trees and have it visually put in front of us. Obviously in colours, it is something that people are quite used to seeing in terms of... obviously this is in blue but often you see the traffic light type system and it does show bulks of the darker blue are the areas where there'd be most impact and so on. So it was certainly a useful tool.” (I8)

Concluding the workshop, the Lead Researcher outlined the requirements of the next workshop, and the activities that had to be undertaken in the meantime. The project team agreed to develop between 3 and 5 service design options, and prepare presentations for each of them. The meeting concluded with a discussion about who should participate in the Trade-off Analysis Workshop that would follow.

It will be recalled that a key feature of SARM is the participation of a range of stakeholders in the trade-off analysis that lies at the heart of SARM. For some of this project's stakeholders, it would not be practical to expect attendance at a

design workshop (e.g. UNHCR, HM Government Ministers). For others, the project sponsor would have to decide whether external stakeholders, that might include representative asylum seekers, their legal representatives, charities and politicians, could be allowed to participate in the design of a service for which the department alone has responsibility.

Some time after this first workshop, a decision regarding stakeholder participation was reached by senior management at UKBA. Given the confidential nature of the project, which was planning potential changes to a highly sensitive government process, and the fact that none of the information and potential service designs likely to be discussed would have been cleared for external publication, it was concluded that the workshop participants would be drawn only from civil servants from the UK Border Agency. However, in order to enable the different perspectives of other stakeholders to be reflected in the trade-off analysis, and to ensure that these views could influence the final decisions being taken, the project team proposed to nominate different members of their team to represent each of the key stakeholder groups in addition to representing their own views.

Whilst this would clearly not have the same effect as having stakeholder representatives directly involved, it did represent an imaginative proposal from the project team to enter into the spirit of stakeholder consideration in the process of service design. And it could be considered a considerable step away from the more typical 'top down' design by management approach to service design found in UKBA and other government organisations.

To summarise the assets created directly in the first workshop, a set of internal and external stakeholders were identified, and subsequently clustered into stakeholder groups. A set of scenarios describing the desired new service was defined, each classified according to the Service Quality Model, and the likely impact of their failure was agreed and documented. And each scenario was linked to each stakeholder in terms of the level of interest that stakeholder would have in the scenario.

This set of information was captured in a spreadsheet application (an early prototype of what became the SARM tool, screenshots of which are shown in Chapter 5), and collectively can be considered as the context in which competing service designs can be created and compared.

6.4 Understanding Stakeholder Interest

One unanticipated outcome of this first SARM workshop was the desire and need to visualise the stakeholder interest in the scenarios. The spreadsheet used to capture this information for this particular instance is a 21 by 12 matrix. The larger dimension, representing the set of scenarios, can be reduced by grouping them according to the Quality Model by which they have already been classified. So in this instance, the 21 scenarios can be aggregated by Quality Sub-Characteristics, and further aggregated by Characteristic, reducing 21 items to just six.

To create a visual 'heat map', the weightings of 0, 1 and 2 that are applied in SARM to 'No Interest', 'Interest' and 'Strong Interest' have been used to calculate the weighted aggregate level of interest of a stakeholder in a given characteristic, scaled to the range 0.0 to 1.0.

The level of interest in each cell of the table is therefore calculated as follows:

$$Interest_{jk} = \frac{\sum_{i=1}^{n_j} Weight_{ijk}}{2n_j}$$

where:

- i = a scenario
- j = a characteristic
- k = a stakeholder

It can be seen that a score of 1.0 will be derived if a particular stakeholder has 'Strong Interest' in all of the scenarios that are associated with that particular characteristic. If the stakeholder had just 'Interest' in all of the scenarios, a score of 0.5 will result, while 'No Interest' in all of the relevant scenarios will produce a score of 0.0.

It is also possible to calculate an overall level of interest by taking an average of the weights for a stakeholder across all scenarios, again placed on a scale of 0.0 to 1.0. This is calculated as follows:

$$Interest_k = \frac{\sum_{i=1}^n Weight_{ik}}{2n}$$

where:

i = a scenario

k = a stakeholder

A ranking of the stakeholders according to this overall level of interest allows a project team to see which stakeholders appear to have the greatest interest across the set of scenarios, and which have the least. The weighted scores, presented as a matrix with stakeholders occupying rows and the quality characteristics on the columns, can be read in either direction. Reading across a row allows the user to see which characteristics are of most, and least, interest to that particular stakeholder, while reading downwards allows the user to see which stakeholders are most, and least, interested in that particular characteristic.

The visual appearance, and its understandability, can be enhanced with the use of colour. The scores, which range from 0.0 (not at all interested), to 1.0 (completely interested), can be associated in bands with colours, or shades of a colour, resulting in a heat map representing increasing levels of interest that can be interpreted at a glance. A version of this table was sent to the Project Leader a week after the conclusion of the workshop along with a summary of the workshop outcome. It was well received, and used to help validate the stakeholder analysis that took place in the workshop. It was also incorporated in later versions of the SARM tool.

6.5 Designing Solution Options

A gap of about one month occurred between the conclusion of the first workshop and the scheduling of the second, trade-off analysis workshop. This

allowed the project team the time to develop a number of competing service design options for consideration and evaluation in that second workshop. Whilst this design activity lies outside the scope of the research project (which is not seeking to explore how service designs are created), its completion was essential to the successful completion of the research project. Without a number of competing service design options, there would be no content for the second workshop to evaluate.

The design activities were conducted by different members of the project team, who were given freedom and time to develop their own ideas on how to change and improve the service in question. The best of these were refined by the team, and presentations were developed for four options that could be evaluated in the trade-off analysis.

“The team leaders took them forward but we did have a discussion as a group as to what we wanted to look at, so we discussed them at a high level as a group and then they went away and developed the detailed analysis of it and the presentations.” (I1)

It is an interesting question as to whether participation in the workshops of Part 1 of SARM had an influence on the solution options that the team subsequently developed. Two of the participants considered this possibility during the later interviews, and both felt that there was little or no connection, or at least that there was no need for service designers to have been involved in Part 1:

“That’s an interesting question, because it obviously gets people thinking about the model, but I don’t think that an understanding of the model is necessarily required to do the work we did developing the solutions. You could almost have gone out to someone who had not been involved and said “Can you please come up with an idea to solve this problem” - that wouldn’t stop you then applying the model you’ve created to test it out. So, in that sense, you could almost cast the net wider in terms of getting input from people, than just those who attend the meetings.” (I1)

“I didn't think about solutions with that [Part 1 of SARM] in mind, only really from the perspective that I figured that the model would do that for us really. Because once you'd designed the solutions, all the criteria that we'd discussed were going to be applied to them anyway... So I didn't attempt to pre-empt that, if you know what I mean. It was just looking at what potentially we thought the best solutions might be. Yes, so I didn't look at it from that whole perspective because I kind of figured that's what the model was going to do.” (I5)

However, it is the case that those who participated in the workshops of Part 1 of SARM took place in the subsequent development of the solutions options that were considered in Part 2. It is not possible to prove the extent of the influence that participation had on the solution designs, but given that Part 1 involved the articulation of high level requirements, in the form of scenarios, it is highly likely that those who participated in Part 1 would have brought that experience, consciously or unconsciously, to bear on their contributions to the activity of solution design.

6.6 The trade-off analysis workshop

This workshop took place on 22nd June 2010, and was attended by eight members of the project team plus the facilitator and the Lead Researcher. The first part of the workshop consisted of the presentation of the service design options that were to be evaluated. Each service design option had a primary advocate from among the project team, who had prepared a powerpoint presentation with which to describe their design and convey its main characteristics and benefits.

“Once I'd worked out what the process was, and had a rough idea of how we'd reached the point that we were at, it all kind of made sense. It all fitted together pretty neatly. I liked the idea of having the four presentations regardless of their respective value, but those presentations were there because they were options which we had to

consider. And then look at, on the basis of the analysis which had been done. So, it made sense. It all made sense.” (I6)

“It was good to have a presentation on each different solution, and after those and working through the service quality model we could quickly see where the two most beneficial options were.” (I8)

As mentioned previously, some of those attending the workshop were also tasked with representing the likely views of external stakeholders, and each solution presentation was accompanied by lively discussion. The aim at this stage was to ensure that participants all had a good understanding of each option.

The main part of the workshop was taken up with completing the risk assessment for each solution option. This involved reading out each scenario, and then considering and discussing that scenario for each solution option in turn. For each scenario / solution option combination, the team were asked to determine the likelihood of that solution failing to deliver the scenario in a satisfactory manner. They were asked to agree a level of risk on a five-point scale: “Unlikely”, “Possible, but not likely”, “Strong possibility”, “Highly likely” and “Almost certain”.

This generated considerable discussion, and the language of the model (which calls for the team to consider the risk of failure) caused a certain amount of confusion. This was overcome with assistance from the facilitator, but it was clear that asking the team to conceive the likelihood of failure was more difficult than had the question been about the likelihood of success. This was especially the case where the scenario itself contained a negative in its language (e.g. avoiding something). In this situation, the team had to assess the likelihood of failing to not do something, and the 'double-negative' involved somewhat hindered the discussion.

“That was where I struggled a bit with the phrasing - the likelihood of not doing it...We did struggle on a few, didn't we.” (I1)

“The only thing that hindered it a little bit is the confusion over whether its a double negative, or a treble negative.” (I2)

“This was when we all got very confused...But it was a case of the wording, like you said you've already tried to do, I'd reword it. And make it just read as it's meant to read, as opposed to having a double-negative into it. Because you could tell people were confused, and they were giving an answer and then realising it had to be changed over.” (I3)

“Of the upside down way of thinking about it? Which I thought confused matters. With that being clarified I think we probably would have gone a long way a lot quicker if you see what I mean.” (I4)

“That was tough. Because you were looking at it in a negative and that wasn't easy. So I think that probably took longer than we thought it might – just from the perspective that it was hard to align what we were saying “yes” or “no” to, if you like. It was quite difficult trying to get your train of thought into “will this have...” ... even now I can't remember exactly how we had to word it! ...Yes, if it was a negative thing, then it was a negative and a negative and ... yes it was difficult trying to be sure you were saying the right thing.” (I5)

“It felt a little bit counter-intuitive, having that additional...well there seemed to be about four negatives in the sentence.” (I6)

“I found it very confusing, to be perfectly honest. When people started talking about “well, you've got to give the answer as a double-negative”, “put a negative in front of the question”.” (I7)

It was clear that this confusion was not present for every scenario:

“I can't remember exactly which ones were difficult, but I do remember that during the meeting we had to rewrite a couple of them

to make sure that they made sense to the people that were analysing them.” (I1)

As suggested from some of the above quotations, the confusion arose when the scenario itself contained a negative – for example *“the solution should not...”*. In this situation, the team were being asked to assess the likelihood of each solution option *“failing to not...”*. Hence the team's references to 'double negatives'. Where a scenario was expressed in positive language (e.g. *“the solution should...”*), there was little difficulty for the team to agree on the likelihood of a given solution option *“failing to...”*. The confusion, even though it was limited to a certain number of scenarios, certainly slowed down the progress of the workshop, and caused an unnecessary degree of frustration for the participants.

Could it have been avoided? This situation would seem more likely to occur when the service design project is developing an improvement or a change to an existing service. In these circumstances, it is much more likely that a requirement will be expressed as a negative comment about the existing service. When a project is creating a completely new service, stakeholders are much less likely to describe their needs negatively (*“the ideal solution should not...”*). One way to limit the risk of falling into this trap would be to take more care of the wording of each scenario when they are created in the Part 1 workshop. However, a constraint on the language of the scenario might limit the ability of the stakeholders to properly express their needs.

One positive aspect of this confusion was that it required the team to help each other clearly understand what the risk options actually meant, so when consensus was reached, it was with a strong degree of certainty.

“I don't know why there was a tendency not to disagree more, but we did have a good debate about a number of them.” (I1)

Note that this was a key difference from the approach eventually adopted in SARM (and which is described in Chapter 5) which requires the team to assess

the likelihood of success (as opposed to the likelihood of failure). This change in approach was introduced as a direct consequence of the project team's experience, and difficulty. Later versions of the tool adopted this 'reverse interpretation', inviting the user to assess the likelihood of success, and 'reversing it back behind the scenes' to calculate the risk exposure (see Table 11 Inverting the Likelihood and its accompanying explanation in Chapter 5). This approach allows the scenarios to be worded positively or negatively, and the question that has to be assessed by the team is relatively simple in either case:

“What is the likelihood of this happening with this solution option?”

Agreed risk ratings were captured in the spreadsheet tool, and by the time this exercise was completed, the team had reach 84 risk decisions (21 scenarios and four solution options). It would be wrong to characterise the workshop as consisting of continuous confusion and debate. The participants could see that there was some logic to the questions being asked, and over time, the team made more rapid progress:

“I thought it was logical. I thought it was easy, so even somebody who has not been involved in the process, as we've proved, it was easy for them to understand it, and it was easy for them to put a rating on it. I thought the discussion was good.” (I2)

“Once everyone got their heads around it, it all seemed to flow a bit then. So that was useful.” (I3)

“But it's like anything else, once you started to get into it, it didn't take so long. So that was ok, it was ok. But it was probably the most difficult part of the process.” (I5)

“I enjoyed it. Certainly it was interesting, and like you say, once we got past that initial confusion, and we were all singing from the same hymn sheet and able to apply it, it was quite clear.” (I8)

This risk information is combined by the software with the impact information that had been captured in the first workshop, resulting in a risk rating in accordance with SARM's risk model for every scenario / solution combination.

During the subsequent interviews, participants were asked whether they thought it appropriate to classify risks according to five categories, or whether, perhaps, three categories (*Low, Medium, High*) might have sufficed. Whilst there was some agreement that fewer categories might have led to quicker results, participants preferred the richer language of five categories:

“Yes, I thought in terms of risk they mapped out the spectrum that exists in risk, certainly.” (14)

“Yes that was fine. [having five categories] It was all or nothing really - a lot of it. It was either not going to impact, or it was going to impact heavily.” (15)

“Maybe we'd spend a bit too long debating whether something was Almost Certain or Possible, but not Likely so I think there perhaps would be some merit in just having Low, Medium and High, in terms of the risk. But again it's always useful to have the different levels and the different impact at each stage, because Low, Medium and High is a little bit ambiguous really. It doesn't quite get down to the nitty gritty of it in terms of the difference between something being Almost Certain and Highly Likely. I would certainly prefer to have the five options there.” (18)

The risk model adopted for SARM has five options for each dimension (Impact and Likelihood). Although outside the scope of this research project, it would be interesting to explore the effects of an alternative 'three category' model (“Low”, “Medium” and “High” for each dimension). Does the richer language of the chosen model make a difference? Does it slow down the process, and if so, does greater deliberation lead to a more satisfactory conclusion? This is a potential avenue of future research.

The team were now able to review this information from a number of different perspectives, using the spreadsheet tool as a visual aid. Each solution option could be viewed from the perspective of each Quality Characteristic, and also from the perspective of each Stakeholder Group. The red, amber, green colouring associated with the risk model helped the team identify areas of particularly high or low risk, and the risks associated with each individual scenario could be explored when greater detail or an explanation for a particular aggregate value was sought.

The “RAG” colour-coding was recognised as being easy to interpret, providing a rapid guide to the areas that needed further discussion:

“It's a clear pictorial, it's there, you can see it straight away.” (I2)

“I thought this was a really good way of doing it as well. Again you could see the colours, because obviously red strikes danger in everyone's mind really, doesn't it.” (I3)

“And it's quite useful to look at where some of the failings are just on a colour basis...The kind of RAG rating is very useful for quick impressions.” (I4)

“You kind of could look at where it was going with the colours, which was fine.” (I5)

Trade-offs, whereby competing solutions 'trade' strengths and weaknesses, were identified, and discussed. Two solution options were quickly ruled out as being unsatisfactory, and the discussion turned to the relative merits of the two remaining solutions.

“The other two, it appears to be just out of the question and that was a clear pictorial, you could see that straight away.” (I2)

In fact the two 'weaker' solutions were perhaps ruled out even earlier, as the version of the tool being used at this time showed the risk values and

corresponding colours as the team were adding their judgements regarding risk likelihood for each solution / scenario combination. The result was that the overall picture gradually emerged as the spreadsheet was being completed. Although this appealed to one of the participants:

“And I think it was evident sort of half way through where we were going with it, or where we appeared to be going with it. So it was quite nice that the results confirmed the way we were thinking.” (I5)

there was clearly a risk with this approach that later risk decisions would become influenced by the results that were already showing on the screen, a risk that was recognised by two of the participants during the subsequent interview:

“There's always an element of risk that people will then start to differ the answers they give – perhaps choose different options to add support to what appear to be the more preferable options which are coming out of the process. So whether it's perhaps worth doing that afterwards rather than...I think we could see the colours emerging as we went through.” (I8)

“And so seeing it being coloured in as we went along, it did...I don't want to say it turned it into it's own self-fulfilling prophecy, but it did validate it the way that we, well, I personally, expected it would.” (I6)

In response to this concern, the tool was subsequently changed so that the trade-off information, with its risk scores and colours, now appears on a separate tab that should be viewed after completion of the Solution / Scenario risk likelihood assessments. A participant, I3, rather dramatically described this improvement:

“Just for sheer dramatic value, if we finished the entire exercise and you hit a button and it appears on the screen, that would be quite impressive, and clearly if it was a game show on ITV3, that's

probably the way that they'd do it. With an advertising break in between completing the exercise and the results coming up on the screen.” (I6)

A number of the participants drew attention during the interviews to the value of exploring the solution options from the perspectives of different stakeholders:

“And so it is quite interesting to look at it with these other people in mind as to 'ok, if we do this what is the consequence of doing this for these key people'. And although you'd kind of do that in any process mapping, you don't really pay that much attention to who are the key people, and what is it...it doesn't matter, we need to make these decisions for productivity purposes and statistical purposes often and we don't really think about what's the impact of that on this spectrum of people. So I found that quite useful, to be sat there thinking 'OK - if we did this, what would be the impact on WRC [Welsh Refugee Council], for instance.’” (I4)

“I think the first table made it obvious then it just obviously then broke it down further and we could see the benefits of each, and the options which were more risky would have more of an impact on the different stakeholders.” (I8)

Although two of the options were clear 'leaders', there was still a desire among the team to use the tool to dig a bit deeper and understand why that was the case:

“So, it was quite interesting seeing it against the respective stakeholders and, as I say, the characteristics - a lot of reading across from one to the other, and then working it out why that is the case.” (I6)

An unexpected consequence of this discussion was the development, during the workshop, of a new, fifth, design that combined the characteristics of the two

preferred solution options. This outcome was a surprise to the project leader, who thought that they had already identified the strongest solution option going into the workshop:

“For me the output of this process is to give me strong ammunition for a business case for the option I already had in mind as being a strong option, but also giving me a 'left field' idea that, as it happens, is much stronger than I'd ever thought before, and is absolutely worth implementing.” (Project Leader)

It is as if the process of exploring a number of solution options from every possible angle, and comparing them from a variety of perspectives, opened minds to the possibility of an innovative solution option that had not been considered in the course of the earlier solution design activities. As the project's leader acknowledged in the subsequent interview:

“Inevitably people go into this sort of thing with preconceptions, and this is the whole purpose of the model really, isn't it, to say well our instinct might be to do X, but in reality we might find that Y is a better fit for the stakeholders and the wider requirements that we model, so in that sense I think it is worth going down the blind alleys. That's where the major value was for me in the process - having a thorough analysis of a range of options that perhaps we hadn't fully considered or wouldn't ordinarily fully consider.” (Project Leader)

The workshop concluded with a decision to further develop the new 'fifth option', and explore the potential to add this to the spreadsheet tool for a formal evaluation at a later date. In the event, time did not permit this, as the team were under pressure to reach a decision and implement the selected solution, which was created during the workshop as the 'fifth option'.

A number of participants highlighted during the interviews the benefit of having the evidence of a formal evaluation, in the form of the output of the tool, that can be used to justify a recommended change to an existing service:

“So although the original one agreed with my thinking, and it could be argued then that I could have proceeded without the model, to have that scientific basis for doing it, especially when I'm in a position where to do it I would have to bid for some funding or staffing, and I would have to sell the idea to other managers, to have the scientific analysis that says 'look, we've considered this from the point of view of the characteristics and the stakeholder interest, and the analysis really is telling us that this is the way to go', then that allows me to build a much stronger case for doing that.” (Project Leader)

“And I think to ensure that that works, we are talking about this [solution option], but we're getting a lot of resistance from people who weren't even at our meeting, so we've said that we've done this, a totally independent process.” (I2)

“But if we were challenged about, and we haven't been, but if we were challenged about the decision to, not disband the team, but to dismantle the team and reassemble it as part of three other teams, then we have some useful analysis which shows that the impacts are almost entirely positive, or at least, the risks associated with it are minimal. And this is one of the few occasions where we could say 'this is how we reached the decision', as opposed to 'well, it seemed like the best thing to do'.” (I6)

In summary, the trade-off analysis workshop built on the assets that were created in the first workshop by articulating the risk of failure to achieve each scenario for each solution option, and additional assets were generated for analysis (the quality characteristic perspectives, and the stakeholder perspectives of the trade-off analysis). The workshop also brought in an additional set of assets generated by the design team(s) in between the two workshops: the proposed service designs (or solution options, as they have also been referred to).

Each interviewee was asked to summarise their experience across the whole

process.

“I think it was interesting, I think it was useful and rewarding, but I think it was, at times, confusing.” (I1)

“I thought it was very good. I thought it was scientific, which I think is needed quite a lot, and I think we could use it a lot.” (I2)

“Useful. Informative. It was good to get people around a table and to be able to throw ideas into it. It was good for people to be able to argue back to get the precise information that we could actually use to take forward. This was good, this was a good tool, I liked your tool... it was a very good tool to actually group everything that we do as a business into six easy characteristics and then just quickly look at and think 'ok, that's easy to understand'. It was just a very well thought out process that you put together which hopefully...well, as you can see something is actually taking place... so you clearly had a lasting effect on the people who matter in this organisation, for them to actually take something forward. And it got people thinking of how we could take what we are doing now to the next level. So it seems to have been a very useful process which is bearing fruits, hopefully.” (I3)

“Enlightening, enjoyable and practical.” (I4)

“Ooh! Interesting...different, actually - it was a very different way of taking an opportunity or a problem whichever way you want to look at it, and breaking it down into component parts and looking at how solutions are going to affect everyone involved, and I think if nothing else, that was, it was quite enlightening in that respect because you kind of think, you know, we could have sat down in a room and come up with these solutions but we wouldn't have considered the impact on...I don't know... half the people that we did.” (I5)

“It was actually really fun. It was nice to see some of my colleagues being put on the spot – possibly defending decisions which they... or recommendations which they didn't necessarily agree with, but still recommending them. It was good fun to have an opportunity to challenge people in a safe environment, and I genuinely enjoyed it. It was...yes, fun is the best word I can use to describe it. If we'd had to have a meeting or a series of working groups to reach the decision that we did, I'm reasonably confident that it would have been as dull as dishwater. But this was... it did make a potentially excruciating process reasonably entertaining. It wasn't dry, it wasn't dull, as you say, the opportunity to challenge, to see the presentations, to challenge them in, not necessarily your role, or your capacity, but in a role that you assumed for the day, that was good. What was nice was everybody participated as well.” (16)

“I enjoyed it. It did my head in on occasions, I have to say. I went back thinking 'oh no!'. [laughter] What am I in here, what have I said? ...But I did enjoy the day. I did. I thought it was good to hear everyone's points of view, be able to give a point of view, yes it was enjoyable in that way. Yes, very much.” (17)

“I'd say enlightening, interesting, useful. It's good now to see my managers ...using this now to actually bring about change in the office. I certainly feel confident in the changes which are being made, knowing that we've gone through the service quality model and I know myself, perhaps not everybody in the office does, but I certainly know that these are certainly the more favourable outcomes as we move forward always looking at trying to improve our process.” (18)

Chapter 7: Analysis

7.1 Introduction

This chapter describes the thematic analysis approach that draws together the theoretical and practical parts of this research (described in Chapter 2 and Chapter 6 respectively), leading to the identification of five themes that characterise the findings of the research. The five themes are stakeholder considerations, evidence-based decision making, the concept of service architecture, the relationship between the disciplines of service design and software development, and the value of model-based evaluation as a tool for use in service design activities. These five themes will be considered in greater detail following a more detailed discussion of the thematic analysis, and the use of both deductive and inductive approaches to the identification of themes.

7.2 Thematic analysis and the case study primary data

Immediately following the completion of the project, the eight members of the UK Border Agency evaluation team were interviewed by the lead researcher. The interviews were recorded, with the explicit permission of each interviewee, and subsequently transcribed by the lead researcher. Each interview lasted up to one hour, and all followed the same semi-structured pattern which mirrored the SARM process, with the addition of a final question to capture an overall impression.

Analysis with a qualitative analysis software package followed, and interviewee comments were tagged to associate them with each step of the SARM Process, and further tagged to indicate a “positive” or “negative” comment. To explain these basic labels further, a “positive” comment was one where the interviewee expressed a positive sentiment, that might reflect personal enjoyment or satisfaction, or an indication that the process worked well, or the team reached agreement. A “negative” label was applied where an interviewee expressed difficulty, a lack of understanding, suggested an improvement or alternative, or indicated that the process was not helping the team progress towards a successful conclusion. Both “positive” and “negative” comments have been

widely used in the previous chapter to describe the views and feelings of the participants throughout the case study, and the “negative” comments made a significant contribution to two of the subsequent enhancements to SARM (see Sections 4.8 and 4.9 for details).

The process of identifying themes from the available information involved iteration between the theoretical part of the project, comparing and analysing the literature from the domains of software architecture and service design, and the practical part of the project, analysing the material generated by the case study, especially the interview transcripts. Five themes were finally chosen out of this process, involving a mixture of theoretical and inductive thematic analysis (Braun and Clarke, 2006, p. 83–84). The significance of stakeholders to service design, especially involving public services, was highlighted in the review of literature (see Chapter 2), and was a key design requirement in the initial development of SARM. It lies at the heart of two of the steps in Part 1 of SARM ('stakeholder analysis' and 'linking scenarios to stakeholders'), and stakeholder considerations led to a further extension of SARM during the project (see Section 4.7 Adding 'Stakeholder Views' for details). The interview transcripts were additionally tagged with “stakeholder considerations” to facilitate an exploration of the extent to which interview data supported this topic as a significant theme. Seven of the eight interviewees commented specifically on the topic.

Another theme that was strongly suggested by the literature review and subsequent initial development of SARM was the novel use of models in the evaluation of competing service designs. Again, the interview transcripts were tagged with this topic, which attracted eleven significant comments from six of the interviewees.

Recognition of the similarities between software development and service design was the original trigger for this research project, and this topic is another underlying theme that permeates the analysis of the literature of both domains. Whilst the relationship between these domains was not explicitly referenced

during the case study or the subsequent interviews, it can be recognised as a latent theme (Braun and Clarke, 2006, p. 84) in the comments of five of the interviewees who drew attention, for example, to the complexity of the service and the value of a formal process to evaluate alternatives.

One theme was strongly suggested by the theoretical analysis but not significantly referenced during the interviews: the concept of service architecture. This was retained as a theme despite the lack of support for it in the interviews, as its absence was not surprising. It is the most conceptual and theoretical of the themes, and apart from a passing comment from the Project Leader, the interviews with the participants focused on the practical experiences of those involved in the case study, and the resulting outcomes.

The last theme to be identified was strongly suggested by the interview with the Project Leader, with supporting evidence from two other interviewees. The topic, “evidence-based decision making”, was not identified during the analysis of the literature, but its significance was highlighted by the participants during the case study and emphasised by the Project Leader during the subsequent interview. The importance of articulating the underlying reasons for making a recommendation was emphasised by several participants, who saw the documented outcome of SARM, in the form of the completed spreadsheet with its collection of collective judgements and decisions, as a valuable piece of evidence that could be used to explain and justify the final selection of a preferred service design.

As a result of this iterative analysis process, linking together the theoretical analysis of the literature from the two domains, and the artefacts generated by the case study at the UK Border Agency, five themes were identified:

1. Stakeholder considerations;
2. Evidence-based decision making;
3. Service Architectures;

4. Service design and software development;
5. Model-based evaluation in service design.

These five themes are explored in more detail individually in the following sections.

7.3 Theme 1 - Stakeholder considerations

The review of literature presented in Chapter 2 shows the multi-dimensional nature of services, with different stakeholders often among those dimensions (Boyne et al., 2003). This complicates the process of evaluation, as different stakeholders can often point to different, and potentially competing, goals (Connolly, Conlon and Deutsch, 1980). Vedung's taxonomy recognises stakeholder concerns as one of these dimensions (Vedung, 1999). This dimension is itself potentially multi-dimensional, with public services often needing to satisfy the differing requirements of a range of different stakeholders. The Beyond Boundaries report identifies six distinct stakeholder groups (Welsh Assembly Government, 2006, p. 4), while Mehrizi et al. propose a stakeholder mapping approach to address the "increasingly complex network of actors" involved in public service delivery (Mehrizi, Ghasemzadeh and Molas-Gallart, 2009, p. 427).

However, it is striking that the wider literature on service quality, which extends beyond consideration of public services, focuses very heavily (indeed, almost exclusively) on the customer's perspective. Lewis and Booms define service quality as "a measure of how well the service level delivered matches customer expectations" (Lewis and Booms, 1983, p. 99–107), and Johnston views service quality solely from the customer's perspective (Johnston, 1995). While Parasuraman et al. look at the service from both consumer and marketer perspectives, it is noticeable that the marketer perspective is driven by the marketer's view of the consumer's expectations (Parasuraman, Zeithaml and Berry, 1985). Lovelock and Wirtz see a service as a bi-directional transaction between a provider and a paying consumer (Lovelock and Wirtz, 2007). None

of these consider situations, such as can occur in public services, where it can be difficult to define precisely who 'the customer' is.

Although this research presents a single case study, and one which resides firmly in the sub-domain of the design of public services, the case study and the interview evidence obtained clearly recognise the importance of considering the service design from the perspectives of different stakeholders. Two of the participants (I2 and I5) expressed surprise at the number of stakeholder groups that were identified in the first workshop, acknowledging that the exercise of identifying and grouping them drew attention to their significance that might otherwise not be recognised:

“I was quite surprised how many there were. Even though most of them we come into contact with on a daily basis, you don't register really how many people you are actually trying to keep happy all the time, so that was kind of the biggest thing.” (I5)

“I think again you realise how many people you are actually dealing with, and what a complex process it is. When you deal with all these people on a daily basis, you don't really think about how many stakeholders are actually involved.” (I2)

It was the recognition of the significance of stakeholders, and their differing perspectives on the requirements and desired outcomes for a service, that led to further development of the stakeholder perspectives, and the incorporation of a stakeholder model in the evaluation process. This introduction was inspired by the review of relevant literature, and its value has been acknowledged by the participants in the case study. By combining a stakeholder model with a service design quality model in a single unified evaluation process, this research has developed an approach that allows service designers to view quality from different stakeholders' perspectives. Trade-off analysis methods that have been widely adopted for software architecture evaluation, such as ATAM, do not explicitly consider stakeholder perspectives, concentrating on an exploration of trade-offs only from the perspective of requirements, as characterised by the

quality model (Kazman, Klein and Clements, 2000).

The approach proposed in this research, and adopted for the case study, combining a stakeholder model with a service quality model, inevitably introduces a complexity to the evaluation process that was not present in the simpler prevailing customer-centric view of service design and service quality. The benefit of being able to analyse trade-offs between stakeholder groups and stakeholder classes in the later trade-off analysis workshop has to be weighed against the additional cost in time and complexity, since it depends on having articulated the level of interest that each stakeholder group has in each scenario in the first workshop. The participants of this case study found that this process was both quick and straight-forward:

“That was straight forward, yes.” (I3)

“That bit was fairly straightforward as well.” (I5)

“I don't recall there being any difficulty, ...we rattled through it quite quickly.” (I8)

The practical experience of the case study did lead to the identification of a further difficulty, caused by the combination of many scenarios and many stakeholders. The process of agreeing the level of interest of each stakeholder group in each scenario demands that the workshop participants consider each combination in turn. For this case study, this meant reaching 252 separate judgements. Whilst the process itself was considered “straightforward”, as indicated above, it proved difficult to validate the correctness of the outcome.

The desire to validate these 252 judgements, by way of some form of summary, led to an innovative refinement of the method, involving the development of a “heat map” that showed the relationship between the stakeholder groups and the quality characteristics. By drawing together the judgements for all scenarios that belong to the same quality characteristics, the 252 judgements could be summarised in a 12 by 6 matrix, with an additional column providing an overall

view that ranked the stakeholders. Additional clarity was provided by converting the numerical content of each cell into a colour, resulting in a heat map that was easy to interpret (see Figure 5 “review for coherence and completeness” and Figure 11).

This matrix can be read both vertically and horizontally. A vertical reading allows the viewer to consider each quality characteristic in turn, and see the relative interest in that characteristic of each stakeholder group. This view helps to validate the earlier exercise of agreeing the level of interest each stakeholder group has in each scenario.

Reading the matrix horizontally allows the viewer to consider each stakeholder group in turn, examining the relative interest of each quality characteristic for that stakeholder group. This view can help to identify any missing scenarios which might cause a stakeholder group to have an unexpectedly low level of interest for a particular quality characteristic.

Participants confirmed the value of this heat map:

“This is useful, yes. Its a lot more accessible than the table.” (I1)

“Yes. It's highlighted that efficiency and reliability were key - that's what we were looking for, which is probably right.” (I2)

“So its just an easy way of quickly looking at something and picking out specifics ... I think that's a really good way of doing it.” (I3)

“You can see what it's for, but the ranking was the key thing for me. And it reflected really what I expected it to reflect, which was encouraging...” (I5)

“It was useful to be able to see the wood for the trees and have it visually put in front of us. ... So it was certainly a useful tool.” (I8)

The final trade-off analysis (see Figure 5 “trade-off analysis” and Figure 13)

enables the workshop participants to consider risk trade-offs from a number of perspectives, including those of the stakeholder classes (see Figure 3) and, where necessary, the individual stakeholder groups. This allows the project team to understand the likely impact of each solution option on the different stakeholders, enabling them to determine whether a given solution can be accommodated satisfactorily, perhaps by mitigating some of the residual risks for affected individual stakeholders or classes. This ability to explore options from different stakeholder perspectives in the trade-off analysis workshop was highlighted as a major benefit of the method by a number of the case study participants:

“So, it was quite interesting seeing it against the respective stakeholders” (I6)

“We could see the benefits of each, and the options which were more risky would have more of an impact on the different stakeholders.” (I8)

“So I found that quite useful, to be sat there thinking 'OK - if we did this, what would be the impact on WRC [Welsh Refugee Council], for instance.’” (I4)

This research has highlighted the potential value of exploring stakeholder perspectives when evaluating service designs. Whilst the Power / Legitimacy / Urgency model (Mitchell, Agle and Wood, 1997) was adopted for SARM, the concept of combining a stakeholder model with a quality model to bring stakeholder perspectives into the trade-off analysis is not dependent on this particular model. It would, for example, be a simple matter to incorporate an alternative model, such as Gardner's Power / Dynamism model (Gardner, Rachlin and Sweeney, 1986), or Mendelow's Power / Interest Grid (Mendelow, 1991). The extent to which the combined stakeholder / service quality approach applies to all service design initiatives beyond this single case study remains unexplored. This is clearly an interesting avenue for future research, and the Service Architecture Review Method presented in Chapter 5 has been

generalised so that is not specific to the case study, or any one particular service. Indeed, it is intended for use in evaluating both public and private sector service designs.

7.4 Theme 2 - Evidence-based decision making

We have seen in Chapter 2 how policy making in government, in the UK and elsewhere, has been influenced by a desire to see decisions backed up by evidence. This has influenced the way in which decisions are made in government, and there is some evidence in this project that the analytical approach of SARM has been perceived by participants as a means of creating an evidence base with which to justify a service design decision.

During the interview, the Project Leader makes the point that a recommendation to adopt a new service design is likely to be accompanied by a bid for new staffing or funding. He describes the need to “sell the idea to other managers”, and suggests that the analytical approach of SARM “allows me to build a much stronger case for doing that”. He suggests that SARM gives him a “scientific basis” for the recommended solution, with “scientific analysis” that “really is telling us that this is the way to go”. For him, the role of SARM is “to give me strong ammunition for a business case”. It is clear that he recognises the importance of bringing something like evidence to justify his recommendation.

A number of the participants used the term “evidence” in describing the SARM approach:

“it’s a clear pictorial, it’s there, you can see it straight away, so I think we should be showing people that the evidence is there.” (I2)

“My only regret is that we didn’t use this for a significant business change where there’s an opportunity to say to a director ‘look at this! For once, look at this. This shows objectively what we should be doing, and this is the proposal that we’re putting forward, and here’s the evidence to support it.’” (I6)

The word 'evidence' is perhaps open to more than one interpretation here. Rycroft-Malone et al. highlight the broad range of evidence types that can contribute to an 'evidence-based practice' (Rycroft-Malone et al., 2004). In the context of service design, where one is considering the relative merits of competing service designs *before any service design has been implemented* (i.e. ex-ante evaluation), the evidence that can be collected cannot be evidence of the actual performance of the service under different designs. That would only be possible if competing designs were to be implemented in the form of pilot implementations. But the thorough examination of each alternative design, using a rigorous method, and the application of models (such as the stakeholder, service quality and risk models used in SARM) to ensure that designs are looked at 'in the round' can be considered an approach to obtain evidence in support of a decision to adopt a particular design. This is exactly what SARM achieves, and this is recognised by the case study participants through their use of words such as 'scientific', 'analytical', 'evidence' and even 'ammunition'. According to some of the participants in the case study from UK Border Agency, this approach has brought a degree of rigour regarding the evaluation of alternatives that was not present before:

“This actually provided, not exactly a scientific explanation, but it was a rational explanation of why the solution which we came to was the most effective solution that we could have reached.” (I6)

Or in the words of the Project Leader:

“a thorough analysis of a range of options that perhaps we hadn't fully considered or wouldn't ordinarily fully consider”

A decision on which of several possible service designs to adopt is not quite the same as a policy decision. But there are strong similarities, and in the case of a public service, the service design that is chosen for implementation becomes the embodiment of a policy decision that was previously made. Policy makers also have to consider alternatives before choosing a preferred policy for implementation. Jesse Grimes has highlighted how service design techniques

are already being applied to policy (Grimes, 2016), and Rachel Glennerster, Executive Director of the Abdul Latif Jameel Poverty Action Laboratory at MIT, has acknowledged the need for a rigorous evaluation, stating in an address to the Institute for Government in 2012:

“Rigour matters. There’s what you think you know, and when you go and look at something carefully and rigorously sometime we learn that our gut reactions and conventional wisdom are wrong.” (Rutter, 2012, p. 6)

The 'Evidence and evaluation in policy making' report suggest that “Well-designed evaluations can then inform the future evidence base.” (Rutter, 2012, p. 8). Whilst SARM will never generate the kind of evidence that will come from a live trial of a new service, it does bring rigour to the earlier phase of the design of new services, helping to lay down a trail of evidence regarding the choices made by the design team and the information that informed them, and ensuring that they consider competing service designs from a wide range of perspectives.

7.5 Theme 3 - Service Architectures

In “How to design a service”, Lynn Shostack was among the first to identify the connection between service design and “computer systems and software design” (Shostack, 1982, p. 57). These parallels are mirrored in the methods that have been adopted to design services, which follow closely approaches developed to design and implement software systems. Design Management, by Kathryn Best (Best, 2015), and “Designing Services with Innovative Methods” (Koivisto and Miettinen, 2009) both highlight the importance of prototyping and iteration in service design. Co-creation is recognised as a key element to the process of service design in “This is service design thinking” (Stickdorn and Schneider, 2014), and in the UK Government's “Service Design Manual” (Government Digital Service, 2016a). These approaches emerged in software system design in the 1990s, with Rapid Application Development (Martin, 1991) and the subsequent rise of agile software development

techniques (Beck et al., 2001) such as Extreme Programming (Beck, 1999), Atern (DSDM Consortium, 2008) and Scrum (Sutherland and Schwaber, 1995).

We have also seen a range of tools and techniques being adapted and adopted from software systems design to the realm of service design (Tassi, 2009). A key finding of this research is that these parallels are worth exploring, and that lessons learned in the field of software systems might be applicable to that of service design. It has been shown how, in the 1980s and 1990s, computer scientists concluded that the complexities of software design can be better managed through their abstraction in considering a system's underlying architecture (Dijkstra and Hoare, 1974; Perry and Wolf, 1992). The similarities between services and software systems would suggest that services too have an underlying architecture, and that this architecture can be the subject of study and analysis, in much the same way that architectural methods and practices have evolved as important elements of the development, implementation and management of software systems. Much of the service design literature focuses on the range of activities and processes involved in designing services (Stickdorn and Schneider, 2014) rather than on what actually gets created. However, attention is turning towards the outcome rather than the process:

“You’d be forgiven for thinking that it [service design] was about the process of design, rather than changing outcomes for users.” Louise Downe, Government Digital Service in “What we mean by service design” (Government Digital Service, 2016d)

Ms Downe goes on to describe the job of a service designer in government as:

“90% archeology - finding out which transactions are involved in which user need, what their original purpose was and whether they’re still effective at doing that. The next 10% is a lot harder - stitching them together into a coherent service that a user can use unaided.” (Government Digital Service, 2016d)

Although Ms Downe does not use the term “architecture”, the above description

resonates with the definitions of software architecture cited in Chapter 2 (Perry and Garlan, 1995; Bass, Clements and Kazman, 2003; International Organization for Standardization, 2007), all of which highlight the connectivity between components.

This thesis, and its description of a service design project that adopted an architectural approach to the evaluation of competing designs, adds weight to the evidence that it can be valuable to think architecturally when designing services. Its focus has been on the application of evaluation techniques originally developed to consider competing software system architectures to the consideration of competing service designs. In this particular case study, it was the architectural trade-off analysis process that directly led to the development of the service design that was finally adopted – a hybrid solution that emerged from discussions during the trade-off analysis workshop. This was described by the project leader as “a 'left field' idea that, as it happens, is much stronger than I'd ever thought before, and is absolutely worth implementing.” (Project Leader)

Implicit in the evaluation approach adopted in this case study is the suggestion that, like software systems, services have architectures that too can be evaluated using similar techniques. If this is correct, it would follow that other architectural disciplines that have been successfully applied to the design, implementation and management of software systems could be similarly applied to that of services. These might include, for example, the adoption of design patterns. Christopher Alexander's work on architectural patterns in the 1970s (Alexander, 1978) was adapted to the world of software in the 1980s (Beck and Cunningham, 1987; Gamma et al., 1994) and is now widely recognised as an important element of software systems design. We are beginning to see the term “Service Pattern” emerge in the field of service design, with the UK Government Digital Service describing the concept of a Service Pattern with its adoption of templates, components and standards (Government Digital Service, 2015).

It is beyond the scope of this project to fully explore the implications that flow

from the recognition of the architectural properties of services, and this research has begun this process by highlighting the potential, and by beginning the exploration of one of the key architectural techniques employed in software design, evaluation, to the field of service design. The term itself, “Service Architecture” is not in common usage, and it is to be hoped that its novel introduction in this thesis will encourage the wider consideration of services from an architectural perspective.

In examining the parallels between the design of software systems and services, Chapter 2 has highlighted the pioneering work of Lynn Shostack, who was among the first to articulate service design as a distinct discipline, highlighting the parallels to the more mature discipline of software systems design. A number of techniques that were first established in the field of software systems design and development have been adopted by the service design professional community. These include blueprinting (Shostack, 1982), use case modelling (Morelli, 2002), agile development (Stickdorn and Schneider, 2014, p. 197–8), prototyping (Stickdorn and Schneider, 2014, p. 192–3) and design patterns (Government Digital Service, 2015). This project has pursued a similar path, but in a new direction, exploring the applicability of software architecture evaluation methods to service design. The evidence from the case study presented in Chapter 6 suggests that such methods would appear to be worthy of further investigation in a broader range of service design projects.

7.6 Theme 4 - Service design and software development

In introducing ATAM for software architecture analysis, the authors suggest that since “a software architecture is a key business asset for an organization, then architectural analysis must also be a key practice for that organization.”. They go on to propose that:

“architectures are complex and involve many design tradeoffs. Without undertaking a formal analysis process, the organization cannot ensure that the architectural decisions made ... are advisable

ones that appropriately mitigate risks.” (Kazman, Klein and Clements, 2000, p. vii)

This research project, and its case study, support the findings of Kazman et al. Trade-off analysis has proved to be a valuable vehicle to stimulate consideration of competing architectures. However, this project has made the substantial leap from ATAM's purpose of evaluating competing software architectures to SARM's purpose of evaluating competing service architectures.

SARM has retained ATAM's core notion of exploring the trade-off between solutions in terms of quality characteristics. Indeed SARM has adopted a software industry standard quality model, ISO 9126-1 (International Organization for Standardization, 1991), with very few changes in terminology. However, in recognition of the often competing demands of many service stakeholders, SARM has also introduced the notion of trade-off between solutions in terms of stakeholder interests.

This method, derived from architecture evaluation techniques widely used for software architecture evaluation, has in turn been applied in practice in a real service design project for the UK Border Agency, as presented in Chapter 6. The subsequent interviews with the participants of that service design project included the following comments:

“I was quite surprised how many there were. Even though most of them we come into contact with on a daily basis, you don't register really how many people you are actually trying to keep happy all the time, so that was kind of the biggest thing.” (I5)

“I think again you realise how many people you are actually dealing with, and what a complex process it is.” (I2)

“It was useful to be able to see the wood for the trees and have it visually put in front of us.” (I8)

“That's where the major value was for me in the process - having a thorough analysis of a range of options that perhaps we hadn't fully considered or wouldn't ordinarily fully consider.” (Project Leader)

“Enlightening, enjoyable and practical.” (I4)

“I'd say enlightening, interesting, useful.” (I8)

A majority of the interviewees here have highlighted, either directly or indirectly (for example by using the term 'enlightening') the complexity of the service under consideration, and the value of the method in shedding light on that complexity to help them select a preferred service design. The statement quoted earlier from the introduction to ATAM could just as easily be applied to the domain of service design.

In addition to making the case that software architecture evaluation methods are applicable to service design, this research strengthens the overall case that similarities between software development and service design can justify exploration of the potential to transfer methods and practices adopted in the one domain to the other.

7.7 Theme 5 - Model-based evaluation in service design

At the heart of this research has been consideration of the potential applicability of software architecture evaluation techniques to the domain of service design. Lynn Shostack was among the first to describe the discipline of service design and highlight the similarities between computer software and services:

“what happens in a computer is often analogous to what must happen in order for a service to be successfully rendered.”
(Shostack, 1982, p. 57)

This has provided encouragement for this research project to explore a technique for evaluating competing software architectures and develop a version suitable for evaluating competing service architectures. Central to the

evaluation of software architecture is the notion of quality characteristics, which are used to explore the differing strengths and weaknesses of competing software architectures in the context of a particular set of solution requirements. The widespread use of common language to describe software quality characteristics has resulted in the development, publication and adoption of an industry standard software quality model, ISO 9126-1 (International Organization for Standardization, 1991), and its more recent iteration, ISO 25010 (International Organization for Standardization, 2011).

The services world has also explored concepts similar to software quality characteristics. Servqual is a quality model that is widely used to evaluate services (Zeithaml, Parasuraman and Barry, 1990). Chapter 2 compares these two models, and highlights many similarities between them, adding further support to the proposition of Ms Shostack and others that the worlds of service design and software design are closely related. The comparison also reveals that the software quality model covers a rather broader range, reflecting the success of its designers in meeting their first requirement to “cover together all aspects of software quality” (International Organization for Standardization, 1991, p. 24).

This thesis has proposed a model-based method for evaluating service designs called the Service Architecture Review Method. It is derived from the dominant software architecture evaluation technique (ATAM) from the Software Engineering Institute at Carnegie Mellon University (Kazman, Klein and Clements, 2000) and in its final form, presented in Chapter 5, it incorporates three models:

1. A service quality model, derived from the ISO 9126-1 software quality model (International Organization for Standardization, 1991), allowing high level service requirements to be classified and grouped, based on their quality characteristics and sub-characteristics;
2. A risk model derived from Management of Risk: Guidance for Practitioners (Murray-Webster and Office of Government Commerce,

2010), used to quantify, discuss and evaluate risks inherent in competing service designs.

3. The stakeholder model, proposed by Mitchell, Agle and Wood (Mitchell, Agle and Wood, 1997), that allows a service's stakeholders to be classified according to their possession of three key attributes;

Chapter 6 presents a case study where an early version of this method is adopted for the evaluation of competing designs of one of the services relating to the handling of asylum seekers in the UK by the UK Border Agency. A number of further refinements to the method were identified during this case study, one being adopted "in-flight" to help the project team visualise the high level requirements and their relationship to the service's stakeholders. Others, relating to visualisation of the trade-off analysis within individual quality characteristics, representation of the 'likelihood' element of risk, and the adoption of a model for the classification of stakeholders, were incorporated within SARM as a consequence of examining lessons learned after the conclusion of the case study.

The use of a Quality Model to explore and classify the set of high level requirements was new to the participants of the case study at UK Border Agency. In the subsequent interviews, they acknowledge the value it brought to the process:

"So it was very helpful, because also as well you had certain things that all fall under the one characteristic, which helps you to understand how they are all linked together." (I3)

"I was quite surprised at how much, when we broke it down [using the Quality Model], there was to think about. Which probably tells me that generally we don't look at everything we should be looking at when we are trying to design a solution for anything." (I5)

"Working through the service quality model we could quickly see

where the two most beneficial options were.” (18)

“It was a very good tool to actually group everything that we do as a business into six easy characteristics and then just quickly look at and think 'ok, that's easy to understand'.” (13)

“I certainly feel confident in the changes which are being made, knowing that we've gone through the service quality model and I know myself ... that these are certainly the more favourable outcomes as we move forward always looking at trying to improve our process.” (18)

The workshop participants are here describing how easily they were able to pick out the different strengths and weaknesses of the competing service designs, and how useful it was that they were forced to take a '360 degree' view of each solution. Both of these benefits derive from the adoption of the Service Quality Model in SARM. In this case study it has performed the same role of abstraction and simplification as it does in a software architecture trade-off analysis, helping the participants understand the significant trade-offs involved when comparing competing solutions. Without the use of the model, the participants would be faced with trying to compare four solution options across a set of 21 scenarios, a total of 84 different cells. The use of the quality model has reduced the 21 scenarios to six standard quality characteristics, enabling participants to view a summary of the trade-offs at a single glance, as can be seen in the example shown in Figure 13.

The model also played an important role in the first workshop, during the definition of the scenarios, although the benefits only became apparent in the trade-off analysis workshop. As recognised by several of the above participants, the model helped to ensure a '360 degree' view of the requirements by acting as a kind of 'checklist' during the creation of the scenarios. This gives the participants some assurance that the solution options have been considered and evaluated 'in the round', and as participant 15 acknowledged, “generally we don't look at everything we should be looking at

when we are trying to design a solution for anything”.

Although the way in which the risk model was adopted for the case study introduced a degree of confusion in the trade-off analysis workshop (as described in detail in Chapter 6), its incorporation was key to giving participants an easy, graphical view of the risk trade-offs involved among the competing solutions. The value of this visualisation, underpinned by the risk model, was recognised by participants:

“It's a clear pictorial, it's there, you can see it straight away.” (I2)

“I thought this was a really good way of doing it as well. Again you could see the colours, because obviously red strikes danger in everyone's mind really, doesn't it.” (I3)

“And it's quite useful to look at where some of the failings are just on a colour basis...The kind of RAG rating is very useful for quick impressions.” (I4)

“You kind of could look at where it was going with the colours, which was fine.” (I5)

“And so seeing it being coloured in as we went along, it did ... validate it the way that we, well, I personally, expected it would.” (I6)

The use of models in service design is not new. A high level framework model has been proposed to “better understand which can be the role of evaluation in service design practices and define – in future stages – a unique blend of service evaluation techniques supporting service design, development, delivery and consolidation phases, even in an adoption perspective.” (Maffei, Villari and Foglieni, 2013, p. 12) In addition to that used in the Servqual method referenced earlier, many service design practitioners advocate the use of stakeholder maps and models (Stickdorn and Schneider, 2014, p. 150–151; Segelström, 2013) and the UK Government Service Design Manual places a

strong emphasis on an analysis of risks in the Alpha Phase of the service design process (Government Digital Service, 2016b). However, it is believed that SARM is the first model-based evaluation method that has been applied to service design, and the first published description of the adoption of a software architecture review method to the domain of service design (Field, 2013a).

SARM fits within cells 2, 7 and 12 of the Service Evaluation Research Matrix proposed by Maffei, Villari and Foglieni (Maffei, Villari and Foglieni, 2013, p. 9), representing a new contribution to the domain of evaluation in the field of services, described as an “open issue” and a “fragmented and controversial topic” (Maffei, Villari and Foglieni, 2013, p. 12).

As the findings of this study have shown, the case study participants found the method, and its use of the risk and quality models (the stakeholder model was introduced to the method after completion of the case study), valuable in helping the re-design of a significant part of the asylum appeals process in the UK Border Agency. Indeed, the preferred service design was developed as a direct consequence of the trade-off analysis workshop that is the concluding element of the SARM process.

The incorporation of the quality, stakeholder and risk models give SARM a consistent language that can be applied across different service design projects, and by playing the dual roles of abstraction and simplification, they make it much easier for project participants to see the essential differences between competing designs for what might be a highly complex service, with many requirements aimed at satisfying the conflicting demands of numerous stakeholders. In a nutshell, the models enable the team to “see the wood for the trees”.

7.8 Limitations of this research and future opportunities

This study has provided further evidence of the strong connection between software systems design and service design, as first suggested by Lynn Shostack. It has proposed the adoption of an ex-ante evaluation method,

derived from a software architecture evaluation method, for use in evaluating competing service designs, and its use in a case study has shown it to be an effective evaluation tool and method in the project to re-design a service relating to asylum seekers for the UK Border Agency. There are some inevitable limitations to the approach adopted, which this section seeks to highlight, recognising the opportunities for future research to revisit the limitations and obtain further evidence and experience where appropriate.

At the heart of SARM, as described in Chapter 5, lies three interconnected models: the quality model, the risk model and the stakeholder model. The quality model chosen was adapted from a software quality model that was a current ISO standard at the time the project began (International Organization for Standardization, 1991). This model has now been succeeded by another ISO Standard, ISO 25010 (International Organization for Standardization, 2011). Might the case study have achieved a different outcome if the evaluation had used the newer standard? Whilst the reason for adapting and adopting a standard originally designed for software quality has been articulated clearly in Chapter 2, the effect of adopting completely different service quality models would be an interesting area of study.

The same question can be asked of the decision to adopt the risk model that was chosen. The overall structure of the model adopted, with its separation of risk into impact and likelihood components, proved to be highly suited to adoption in SARM. It has allowed the impact element to be attached to each scenario, while likelihood is assessed by the participants in the trade-off analysis workshop for each scenario / solution combination. However, the model adopted had five categories of impact, and five of likelihood, with textual labels being associated with a numeric value on each scale. The adopted risk model is in effect a 5 x 5 matrix. Would a different number of categories on either or both scales have made a difference? Might, for example, participants have a better, and more consistent, understanding of “Low”, “Medium” and “High”, for either Impact or Likelihood, instead of the more complex “Insignificant”, “Minor”, “Moderate”, “Major” and “Disastrous” categories for

impact, and “Rare”, “Unlikely”, “Possible”, “Likely” and “Almost Certain” for likelihood? Further research is needed to identify categories that are least likely to be misunderstood, and that are consistently interpreted by different participants in a workshop.

The adoption of a stakeholder model in SARM was introduced after the case study. It introduces a degree of abstraction on the stakeholder dimension that mirrors the abstraction on the requirements dimension achieved by the quality model. Might a different model be more suitable? Which, of various possible stakeholder models, would give participants in a trade-off analysis workshop the greatest insights? As with the risk and quality models, there is opportunity for significant research to explore the choice of model and its possible impact on decisions made by participants of a service design trade-off analysis workshop.

The combination of risk, quality and stakeholder models produces an analytical approach to ex-ante evaluation of service designs. The details contained within this analysis include an estimate of the trade-offs, strengths and weaknesses, involved in each candidate design. If one of these is chosen for implementation, the analysis has potential to form the basis for developing mitigation plans for those scenarios that are less likely to be satisfactorily achieved. What is the best way of doing this? How might the outcome of the trade-off analysis be used in an ex-post evaluation of the service following the adoption and implementation of the preferred candidate design? The structure provided by SARM with its three models has potential application beyond the evaluation of competing designs, and is another area for future research.

This thesis presents a single case study that adopted SARM to evaluate competing designs for a key part of the services provided by the UK government to deal with asylum seekers. SARM has been documented as a method for evaluating services more generally, yet it has only been tested with this single example of a public service. Chapter 2 highlights some of the distinct characteristics of public services. The suitability of SARM for other kinds of service remains untested, and the extent to which SARM can be applied to a

wider range of services remains an unexplored area ripe for further research. One might go further, and pose the question as to whether SARM might be applicable to fields beyond service design? This is far beyond the scope of this thesis, yet during the course of this research project, two fields have been proposed: software architecture and policy. These are discussed in a little more detail later in this chapter.

This thesis, by including both a theoretical presentation of a method, and a case study examining its first use on a real service design project, has taken the important first step from its contribution to theory to its first use in practice, applying the ideas to a real service design case, and using the outcome of the project to identify further questions worthy of investigation. These further questions go beyond the possible scope of a single research project, and a valuable outcome of this research project is the identification of a number of avenues of exploration, suitable for their own dedicated research efforts, that can extend the field of knowledge around the important topic of evaluating service designs, and exploring the potential of examining services from an architectural perspective.

7.9 Adopting SARM for software architecture

The aim of this research project has been to explore the application of software architecture evaluation techniques to the domain of service design. Whilst some adaptation of these techniques to fit them more appropriately to the different domain of service design might be reasonably expected, it was not anticipated that these refinements might themselves be suitable, and beneficial, for re-adoption in the domain of software architecture evaluation. The key innovation that has triggered this has been the introduction of the stakeholder perspectives to the architecture evaluation method. Whilst this was initiated on consideration of the many stakeholders involved in services, and especially public services (as described in Chapter 2), it has been recognised that stakeholder perspectives are of similar significance to software solutions. Indeed, this is well documented in the project management literature (Littau,

Jujagiri and Adlbrecht, 2010) and also that relating to software architecture (Rozanski and Woods, 2011). The Software Engineering Institute, when discussing the importance of software architecture evaluation, warns that:

“if the architecture's stakeholders are not involved, the comprehensive goals and requirements for the architecture (against which it must be evaluated) will not emerge.” (Software Engineering Institute, 2016)

However, none of the major software architecture evaluation methods formally incorporates stakeholder perspectives. The inclusion of this aspect in the Service Architecture Review Method described in Chapter 5, therefore presents an opportunity to introduce this back into the software architecture evaluation methods from which it was derived.

The author was privileged to have taken up the role of Chief IT Architect at Emirates Airlines in 2011, and the success of the case study documented in this thesis, and particularly the impact of stakeholder perspectives on the participants of the trade-off analysis workshop, had a strong influence on the way in which the author introduced a software architecture review process at Emirates Group. The opportunity was taken, and rather than follow an industry standard approach for software architecture reviews, such as ATAM (Kazman, Klein and Clements, 2000), the author chose to apply SARM as the standard method for reviewing software architectures at Emirates Group. It was adopted exactly as described in Chapter 5, substituting the word “Solution” for “Service” in the method title and in the descriptions of the quality characteristics and sub-characteristics. It remains in use at Emirates Group at the time of writing.

This approach and the benefits that have resulted have been presented at international conferences focused on software architecture (Field, 2012) and computer science (Field, 2013c) and a number of other organisations have begun to adopt the method. Published documentation of these further experiences has yet to be forthcoming, and so there remains opportunity for

further research to more formally examine the consequences of adopting this approach in the 'home domain' of software architecture.

7.10 SARM in other domains?

Another domain to which this research may be applicable is that of policy. The relationship between SARM and evidence-based policy has already been explored earlier in this chapter. It is interesting to note that a number of service design practitioners have been promoting the concept of applying service design methods to the domain of policy. In (Grimes, 2016, p. 3) we read that the success of service design “has seen its application broadening, to include policy-making”. And in (Thévenet and Vincent, 2013) the authors discuss the involvement of service design in government policy-making in the UK and Denmark. Whether this application of service design to government policy extends to the application of SARM to policy decisions is untested. Such an exploration is beyond the scope of this project, and it would appear to be a possibility worthy of future study, given the broader interest and growing experience in applying service design techniques to policy-making.

Chapter 8: Conclusion

8.1 Introduction

This concluding chapter brings together the outcomes of the research described in this thesis. It begins, with Section 8.2, by drawing to a close the journey that has been the research project and its associated case study. The aim and objectives are reviewed in retrospect, and Sections 8.3 and 8.4 highlight the contributions to theory and practice respectively. Section 8.5 points the way to future research; new journeys, the possibilities of which have opened up as a result of the research, in some cases as a consequence of the research's limitations, in others due to the opportunities identified among the project's insights and contributions. These lie outside the scope of the original research project, but offer interesting new avenues of exploration for the future. Section 8.6 reflects on the personal journey undertaken by the author and lead researcher. In addition to the more objective pursuit of the project's research plan, its execution has led the author into unexpected areas of knowledge and insight that leave a permanent impression and will have effects far beyond the narrow boundaries of the research's focus of attention. The chapter, and the thesis, conclude, in Section 8.7, by putting this research into a broader context, offering the hope that it can help accelerate the acquisition of insights and ideas in the emerging domain of service design through the study and practical application of experience gained in the slightly older domain of software architecture.

8.2 Key insights from the research - the end of one journey

Economists have recognised and discussed the distinction between services and physical goods for centuries. Yet recognition of the act of designing a service and the concept that a designer has a number of design choices to make when creating and deploying a service are relatively new, with interest growing since the 1980s. Since then, much of the academic focus of attention has understandably been on the act of designing a service. Indeed one of the earliest papers, considered by many to signify the 'birth' of service design, was entitled 'How to Design a Service' (Shostack, 1982). Corporations and public

sector bodies have placed increasing emphasis on the importance of services, and on measuring and improving their quality. Considerable attention has been given to methods for assessing service quality once the service is being delivered and consumed (i.e. ex-post evaluation). Evaluation of a potential service design, or of competing service designs, *prior* to implementation has been a neglected field (Maffei, Villari and Foglieni, 2013), yet one that has some significance given the high cost of creating or changing services, and the potentially high impact of deploying a flawed service design.

This thesis has conducted the reader through the journey of a research project that has explored that neglected field, the ex-ante evaluation of service designs. It has done so through the lens of experience gained in a similar, yet somewhat more mature domain, the evaluation of software solution architectures. As a submission presented in partial fulfilment of the requirements for the degree of Doctor of Business Administration, it has articulated contributions to both theory and practice. It has confirmed the similarities between the two domains that were highlighted by earlier authors, and in considering the application of an architecture evaluation technique, it has recognised the need to address the many perspectives that different stakeholders bring to a service. The introduction of a stakeholder model to a derivative of the Architecture Trade-off Analysis Method (ATAM) from the Software Engineering Institute is a novel enhancement to the method when used to evaluate competing software solutions, though its introduction was inspired by considerations of how to adapt the method for application to service design evaluation.

This thesis has also described how the research project was itself divided into two parts, a theoretical examination of the relevant areas of the domains of software development and service design, and a practical case study involving the development and first use of a proposed method for the evaluation of competing service designs. However, it would be wrong to leap to the conclusion that the theoretical part of the project was solely responsible for the contributions to theory, or that the case study was solely responsible for the contributions to practice. Whilst the initial proposition of a service architecture

review method grew out of the analysis of both domains (software development and service design), and an outline of the concept is presented in Chapter 2, Background and Literature, of this thesis, the proposed method itself was further refined, as described in Chapter 4, in the light of the practical experience that came out of the case study. And further opportunities for study, as highlighted in Chapter 7, Analysis, were identified following the analysis of that experience. And the combined theoretical and practical contributions of this thesis have been widely shared with other service design practitioners at international conferences in Switzerland, Sweden and the UK (Field, 2010a; Field, 2010b; Field, 2013b), and in the leading journal of service design, Touchpoint (Field, 2013a).

The study of the two domains led to the proposed service architecture review method, which was applied for the first time in the case study in partnership with the UK Border Agency. The case study contributed directly to the design decisions that were made at the UK Border Agency, and the project leader and team members acknowledged the valuable contribution that the application of SARM made to their decision process, and their selection of a new service design. The case study has made a wider contribution to practice, in the form of improvements to the method that were identified during its use. These include the visualisation of stakeholder interests (see Figure 11), and the visualisation of risk distribution by quality characteristic (see Figure 16). The most significant improvement to come from the practical experience of the case study was the inversion of risk likelihood relating to each scenario / solution combination. Without this inversion (which is shown in Table 11 Inverting the Likelihood), workshop participants became confused by the negative language of having to assess the likelihood of failure. This confusion was recalled by all participants in their subsequent interviews, as described in Chapter 6, Service Design Case Study. It triggered a re-think on how best to describe the risks associated with each scenario / solution combination, and the result of this has been an improvement that will benefit all subsequent users of SARM.

The unexpected opportunity to contribute some of this research project's

findings back into the world of IT architecture was a consequence of the author's ongoing engagement in the IT profession. During the course of the research, the author was appointed to the position of Chief Architect at Emirates Group IT in Dubai. This senior management position carried with it responsibility to assess all prospective IT solution architectures, and the author recognised the opportunity to introduce the adaptation of ATAM that was SARM.

Given the similarities between the domains of software development and service design, it is perhaps not surprising that the enhancements developed to suit the characteristics of services, as described in this thesis, would be suitable for adoption in IT architecture reviews. The primary enhancement here was the introduction of the stakeholder model and perspectives, and these were warmly welcomed by the project teams at Emirates Group that adopted SARM. The service quality model adopted for SARM, as described in Table 7 The SARM Quality Model, was itself a derivative of an IT Software Quality Model (ISO 9126-1), so reverting to the original model when applying SARM to IT projects was no difficulty. The experience of applying SARM for IT architecture reviews was presented at the leading global IT architecture conference SATURN in 2012 (Field, 2012), and at the BCS International IT Conference in 2013 (Field, 2013c).

Returning to the aim and objectives set out in Section 1.4 of Chapter 1, we can see that the project has accomplished its aim of achieving both a theoretical and a practical exploration of whether an attribute-based evaluation method, derived from a software architecture evaluation method, can be usefully applied to the evaluation of competing service designs. The project unexpectedly shed new light on the domain of software architecture evaluation as a consequence of achieving its six objectives, which sought to transfer knowledge from that domain to the newer domain of service design.

8.3 Contribution to theory

The contributions to theory made by the research presented in this thesis stem from the connections made between two distinct domains: that of software

development, and more specifically, software architecture, and the emerging domain of service design. This research is not the first to highlight a possible connection between software development and service design. Indeed that connection was highlighted by one of the earliest pioneers of service design, Lynn Shostack, and a number of techniques that were first developed in software systems design and development have transferred successfully, and become widely adopted, in the emerging field of service design.

However, the transfer of the concept of service architecture from software architecture is novel, and from this flows the possibility that evaluation methods used to select software systems architectures from among possible candidate designs can be similarly applied to service designs. This research project has taken the first step to realising this possibility, with both the theoretical development of a suitable service architecture review method (SARM) and its first practical application to a real service design project. This new concept of a service architecture opens up potential for further transfers of techniques that have been developed in the domain of software architecture to the domain of service design.

This research has proposed a service quality model that is also derived from a software industry standard. It extends the scope of quality models that have been adopted to date when considering service quality, adding quality characteristics such as efficiency, maintainability and adaptability, to the much narrower set of customer-oriented quality characteristics that have been used to date to evaluate service quality. The result is a more rounded approach to considering service quality, again enabling the domain of service design to benefit from the more mature techniques and approaches that have been developed and applied over the years in the domain of software design and development.

Although the importance of stakeholders is widely recognised in the service design literature, the focus of literature on service quality is firmly placed on customer satisfaction and customer perception. This research has presented a

broader view of service quality, not just in terms of quality characteristics as suggested above, but also in terms of stakeholder perspectives. The need to address many stakeholder perspectives has long been recognised as a difficulty in designing, delivering and evaluating public services. In this research, the introduction of stakeholder perspectives to an existing trade-off analysis method represents an advance not only for the evaluation of service designs, but also for the method's original application to the evaluation of competing software architectures.

The combining of a quality model, a risk model and a stakeholder model in a method with which to conduct architectural evaluations is also novel. This idea of applying an ex-ante evaluation method to service designs adapted from software architecture review methods was first presented by the author publicly in a conference paper in Geneva in February 2010 (Field, 2010a). The introduction of all three models in a software architectural evaluation method was first presented by the author at the SATURN conference in 2012 (Field, 2012). Together, these represent contributions to theory in the fields of both service design and software systems architecture.

8.4 Contribution to practice

The case study presented in this thesis, and its successful outcome, as acknowledged by the participants of the service design project at UK Border Agency, represents a contribution to practice. As we have seen in Chapter 6, the process was described by the participants from UK Border Agency variously as “Informative”, “Enlightening”, “Practical”, “Enjoyable”, “Really fun”. One participant suggested that “I think we could use it a lot”, while another stated that:

“It was just a very well thought out process ... And it got people thinking of how we could take what we are doing now to the next level. So it seems to have been a very useful process which is bearing fruits, hopefully.” (I3)

Chapter 5 of this thesis is also a contribution to practice. It has been written as a tutorial for future SARM practitioners, incorporating the further refinements of the method that were devised during the research project. The software tool that was used to manage the process in the case study has also been developed with reuse in mind. Details specific to the UK Border Agency have been removed, and it can be adopted for any project that seeks to conduct an evaluation using the Service Architecture Review Method. It has been developed using Microsoft Excel to facilitate its widespread adoption, while a version of this tool using an open source alternative, LibreOffice Calc, has also been created to further broaden its appeal.

The author has engaged widely with both academic and commercial practitioners to publicise the concepts, the research, the case study, and the resulting method, SARM. Within the domain of service design, a paper describing the approach was presented at the First International Conference on Exploring Services Science in Geneva in February 2010 (Field, 2010a). The author led a workshop describing SARM and its application in the case study with UK Border Agency at the second ServDes Conference in Linköping, Sweden in December 2010. A further presentation describing SARM was given at the Service Design Network Conference (SDNC 13) in Cardiff in November 2013, and a paper entitled “Introducing the Service Architecture Review Method” was published in Vol. 5 No. 2 of the leading service design journal Touchpoint in September 2013.

An unanticipated outcome of this research, highlighted in Section 7.9 of Chapter 7, has been the incorporation of stakeholder perspectives in a software architecture evaluation method. During the course of this research project, the author was appointed Chief IT Architect at Emirates Group, the global airline group. This opened the opportunity to introduce a software architecture evaluation method based on SARM, and in due course, the method was adopted by Emirates Group IT as its standard architecture evaluation method. To propagate this more widely among the IT architecture professional community, the author presented the approach at SATURN 2012, the annual

software architecture conference organised by the Software Engineering Institute in St. Petersburg, Florida, in May 2012 (Field, 2012). To further publicise the approach, a paper entitled “Solution Architecture Review Method - a formal approach to evaluating solution options” was presented at the BCS International IT Conference in March 2013 in Abu Dhabi (Field, 2013c).

In addition to being adopted as the software architecture review method in use at Emirates Group IT, the software and user guide have been made available on request to the IT department of one of the world's largest gas producers in Qatar, and a leading pharmaceutical company in Switzerland.

8.5 Limitations and future work - the beginning of other journeys

A project, including a research project such as the one described in this thesis, has a distinct beginning and end. Yet this research project should be seen as just stage one of a larger journey that has not yet completed, or perhaps a number of journeys, as it splits off into a number of interesting avenues of exploration. A number of innovations have been introduced in this thesis, and the extent to which each has been thoroughly tested, validated and understood, is inevitably limited. So each of these represents opportunity for the commencement of further 'journeys' of discovery.

Perhaps the most substantial innovation introduced in this thesis is the transfer of learning from one domain, software development, to another, service design. The thesis finishes with conclusions drawn from the first application of a review method that embodies that transfer of learning. The single case study led to further lessons and refinements, but the conclusion of this research at this stage leaves open some interesting further avenues to be explored in future research. These have been discussed in Chapter 7, Analysis, and involve validating the results of this research, and exploring the extent to which the learning from software development can be applied to a range of different types of service. And if methods of reviewing software architectures can be successfully transferred from software development to service design, are there other methods and techniques that are similarly applicable?

The adoption and adaptation of a number of models, to represent quality, risk and stakeholders, suggests another set of future research avenues to explore. Are these the right models? Would outcomes be different with the adoption of alternative models? Does the use of these models provide the basis for comparing the application of the method across different service architecture reviews? Do they contribute to new methods of ex-post service evaluation?

And finally, the behaviour of the participants in the case study described in this thesis has suggested the potential for further research to understand participant behaviours and their impacts on decisions. Do individuals view risk in a similar manner? If a group of individuals are asked collectively to agree a quantitative measure of risk impact or likelihood, how easily are they influenced by one or two strong leaders? Might valuable inputs be lost with such an approach? Do people prefer to quantify risk impacts and likelihoods with words or numbers?

By drawing two domains together, and transferring some knowledge from one to the other, this thesis has perhaps asked rather more questions than it has answered, opening up a number of new avenues for future investigation.

8.6 A personal journey

The project, and the writing of this thesis, has also been something of a personal journey for the author. Although quite experienced in academic writing in the field of computer science, he found the rigorous demands of exploring two distinct domains, one of them dominated by business and social considerations over technical ones, to be a new and enlightening experience. Even in his 'home' territory of computer science, there was new and surprising material waiting to be uncovered, and of particular interest here was the 1969 satirical article written in response to the NATO conference on software engineering, that was suppressed until 1996 (Randell, 1996). This 'corner' of the research prompted profound questions about the nature of a profession, and how a discipline evolves from a craft towards commercial replicability and, eventually, a scientific engineering discipline. Another serendipitous discovery was the Treatise "De Architectura" by Marcus Vitruvius Pollio (Marcus Vitruvius

Pollio, 25AD), thankfully available in English translation, whose statements about elegance and beauty resonated with the author, and are mirrored in the writings of Grady Booch nearly 2,000 years later (Booch, 2006).

At a more practical level, the research gave the author the opportunity to put theory into practice, both in terms of applying SARM as part of real service design project at the UK Border Agency, and later, in further adapting it for application in software systems architecture reviews at Emirates Group and subsequently with a number of other major commercial organisations. The project triggered interaction with communities, in the areas of service design and public policy, that were new to the author. These are links that endure today, thanks to the benefit of social media, and continued membership of a professional body, the Service Design Network (SDN, 2016).

8.7 A journey in time

To conclude this thesis, we return to the dominant aspect of the research described here – the transfer of knowledge from one domain to another. The thesis has highlighted the similarities between the domains of software development and service design, but there is one characteristic that they do not share - 'age'. The world of software development began in the 1940s, with the early development of the electronic computer. Though it may be acknowledged that some of the theory of software was formed as early as the 1840s by Ada Byron, Countess of Lovelace, and expanded in the 1930s by Alan Turing, the practical development of software commenced with the creation of the first electronic computers. It took about 30 years for the early concepts of software architecture to appear, and a further 20 years for methods and practices dealing with software architecture to develop and reach the stage when they were ready for broad adoption by practitioners. Mary Shaw has suggested that the world of software progressed from being a 'craft' to 'production' over this period, moving into 'commercialisation', with some adoption of 'science' (Shaw, 1990). She concludes that it is on the road to become a professional engineering discipline, but has not yet arrived at that point.

Service design is a much younger domain, whose origins as a distinct discipline can be seen emerging in the 1980s. In the short time since it emerged, services have become recognised as critical to almost every aspect of modern life. Services extend far beyond what is loosely termed 'the services industry'. As highlighted elsewhere in this thesis, most producers of physical products now recognise the importance of services that complement, and sometimes envelop, their physical goods. The interaction between citizens and businesses on the one hand, and government on the other, is typically described as the consumption and delivery of 'public services'. Increasing numbers of experts, in both the public and private sectors, are devoted to designing, developing, delivering, measuring and improving, services.

The key contribution of this thesis is the transfer of knowledge from one domain to a somewhat younger one. It represents a rare opportunity to make 'a journey in time', indeed to cheat time, by enabling the younger domain of service design to benefit from the experience of its 'older cousin', software development. As we have seen, it took software development about 50 years for the discipline of architecture evaluation to emerge and mature. Without the benefit of this transfer, it might still be another 20 years before the need for similar methods is recognised and addressed in the domain of service design. The heavy reliance of modern society on services suggests that the domain of service design needs to evolve more quickly than did its 'older cousin', and it is hoped that this thesis represents a significant contribution to this much needed acceleration.

References

- Achterkamp, M. C. and Vos, J. F. J. (2008) Investigating the use of the stakeholder notion in project management literature, a meta-analysis, *International Journal of Project Management*, **26**(7), pp. 749–757.
- Alexander, C. (1978) *A Pattern Language: Towns, Buildings, Construction*, New York, OUP USA.
- Alexander, I. and Robertson, S. (2004) Understanding project sociology by modeling stakeholders, *IEEE Software*, **21**(1), pp. 23–27.
- Ali Babar, M., Zhu, L. and Jeffrey, R. (2004) A Framework For Classifying And Comparing Software Architecture Evaluation Methods, In *Proceedings of the Australian Software Engineering Conference*, IEEE, pp. 309–318.
- Association for Project Management (2006) *APM Body of Knowledge*, 5th edition, Association for Project Management.
- Barbacci, M., Klein, M. H., Longstaff, T. A. and Weinstock, C. B. (1995) *Quality Attributes*, Technical Report, Software Engineering Institute, Carnegie Mellon University.
- Basili, V. R., Caldiera, G. and Rombach, H. D. (2002) *The Goal Question Metric Approach*, University of Maryland, Institute for Advanced Computer Studies.
- Bass, L., Clements, P. and Kazman, R. (2003) *Software Architecture in Practice (2nd Edition)*, 2nd ed, Addison-Wesley Professional.
- Beck, K. (1999) *Extreme Programming Explained: Embrace Change*, 1 edition, Reading, MA, Addison Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001) Manifesto for Agile Software Development, [online] Available from: <http://agilemanifesto.org/> (Accessed 20 August 2016).
- Beck, K. and Cunningham, W. (1987) Using Pattern Languages for Object-Oriented Programs, In Orlando, [online] Available from: <http://c2.com/doc/oopsla87.html> (Accessed 9 July 2016).
- Bengtsson, P. and Bosch, J. (1998) Scenario-based software architecture reengineering, In *Software Reuse, 1998. Proceedings. Fifth International Conference on*, pp. 308–317.
- Bengtsson, P. and Bosch, J. (1999) Architecture level prediction of software maintenance, In *Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on*, pp. 139–147.
- Berners-Lee, T. (2006) Linked Data - Design Issues, [online] Available from: <http://www.w3.org/DesignIssues/LinkedData.html> (Accessed 7 February 2010).

- Berry, L. L. (1991) *Marketing Services: Competing Through Quality*, New York : Toronto : New York, Free Press.
- Best, K. (2015) *Design Management: Managing Design Strategy, Process and Implementation*, London, Fairchild Books.
- Booch, G. (2006) On Architecture, *IEEE Software*, **23**(2), pp. 16–18.
- Boyne, G., Farrell, C., Law, J., Powell, M. and Walker, R. (2003) *Evaluating Public Management Reforms: Principles and Practice*, Open University Press.
- Braun, V. and Clarke, V. (2006) Using thematic analysis in psychology, *Qualitative Research in Psychology*, **3**(2), pp. 77–101.
- Brennan, K. (ed.) (2009) *A Guide to the Business Analysis Body of Knowledge(R) (BABO)*, International Institute of Business Analysis.
- British Standards Institute (2007) *OHSAS 18001 Occupational Health and Safety*.
- Brooks, F. (1975) *The mythical man-month : essays on software engineering*, Reading Mass., Addison-Wesley Pub. Co.
- Cabinet Office Strategy Unit (2008) Excellence and Fairness: Achieving world class public services, [online] Available from: http://www.cabinetoffice.gov.uk/strategy/publications/excellence_and_fairness/report/html.aspx (Accessed 12 January 2009).
- Campbell, D. T. and Stanley, J. C. (1966) *Experimental and Quasi-experimental Designs for Research*, Houghton Mifflin (Academic).
- Cleland, D. I. (1986) Project Stakeholder Management, *Project Management Journal*, **17**(4), pp. 36–44.
- Clements, P., Kazman, R. and Klein, M. (2001) *Evaluating Software Architectures: Methods and Case Studies*, Addison Wesley.
- Connolly, T., Conlon, E. J. and Deutsch, S. J. (1980) Organizational Effectiveness: A Multiple-Constituency Approach., *Academy of Management Review*, **5**(2), pp. 211–218.
- Department of Communities and Local Government (2010) Third sector, community and voluntary groups - Communities and neighbourhoods - Communities and Local Government, [online] Available from: <http://www.communities.gov.uk/communities/thirdsector/> (Accessed 13 February 2010).
- Design Council (2016a) Service transformation, *Design Council*, [online] Available from: <http://www.designcouncil.org.uk/our-services/service-transformation> (Accessed 7 October 2016).
- Design Council (2016b) Video: What is service design?, *Design Council*, [online] Available from: <http://www.designcouncil.org.uk/news-opinion/video-what-service-design> (Accessed 29 July 2016).
- Dijkstra, E. W. (1999) E.W. Dijkstra Archive: Computing Science: Achievements and

- Challenges (EWD1284), [online] Available from:
<http://www.cs.utexas.edu/users/EWD/transcriptions/EWD12xx/EWD1284.html>
 (Accessed 22 December 2009).
- Dijkstra, E. W. and Hoare, C. A. R. (1974) *Structured Programming*, New Ed, Academic Press Inc.
- Dobrica, L. and Niemela, E. (2002) A survey on software architecture analysis methods, *IEEE Transactions on Software Engineering*, **28**(7), pp. 638–653.
- DSDM Consortium (2008) *DSDM Atern The Handbook*, 1st ed, DSDM Consortium.
- Duenas, J. C., Oliveira, W. L. de and Puente, J. A. de la (1998) A Software Architecture Evaluation Model, In *Proceedings of the Second International ESPRIT ARES Workshop*, Las Palmas de Gran Canaria, Springer.
- Erlhoff, M. and Marshall, T. (2007) *Design Dictionary*, Birkhauser Verlag AG.
- E.W. Dijkstra Archive (2010) EWD Archive: Home: architecture, [online] Available from:
<http://ewd.cs.utexas.edu.master.com/texis/master/search/mysite.html?q=architecture>
 (Accessed 23 January 2010).
- Field, S. (2009a) *An approach to investigating the potential impact of mass collaboration on the design and delivery of public services*, 1st Assignment, DBA, Glamorgan Business School, University of Glamorgan.
- Field, S. (2009b) Evaluating architectural options: ONS experience of conducting architecture reviews, In *MSIS 2009*, Oslo, United Nations Economic Commission for Europe.
- Field, S. (2010a) Can Software Architecture Review Methods Apply to Service Design?, In *IESS 2010*, Geneva, Springer-Verlag, pp. 111–124.
- Field, S. (2010b) Service Architecture Review Method, In *Conference Proceedings; ServDes 2010*, Linköping, Linköping University Electronic Press.
- Field, S. (2012) Reflecting Stakeholder Perspectives in Architecture Reviews, In *SATURN 2012*, St. Petersburg, Florida, Software Engineering Institute, Carnegie Mellon University.
- Field, S. (2013a) Introducing the Service Architecture Review Method, *Touchpoint*, **5**(2), pp. 72–75.
- Field, S. (2013b) Service Architecture Review Method, In *Service Design Global Conference*, Cardiff, Service Design Network.
- Field, S. (2013c) Solution Architecture Review Method - a formal approach to evaluating solution options, In *BCS International IT Conference 2013*, Abu Dhabi.
- Freeman, R. E. (1984) *Strategic management: a stakeholder approach*, Boston, Pitman.
- Freeman, R. E. (2010) *Strategic Management: A Stakeholder Approach*, Cambridge

University Press.

Gacek, C., Abd-Allah, A., Clark, B. and Boehm, B. (1995) On the Definition of Software System Architecture, In *Proceedings of the First International Workshop on Architectures for Software Systems*.

Gamma, E., Helm, R., Johnson, R., Vlissides, J. and Booch, G. (1994) *Design Patterns: Elements of Reusable Object-Oriented Software*, 1 edition, Reading, Mass, Addison-Wesley Professional.

Gardner, J. R., Rachlin, R. and Sweeney, H. W. A. (eds.) (1986) *Handbook of Strategic Planning*, 99th edition, New York, John Wiley & Sons.

Government Digital Service (2015) Better services with patterns and standards | Government Digital Service, [online] Available from: <https://gds.blog.gov.uk/2015/08/06/better-services-with-patterns-and-standards/> (Accessed 20 August 2016).

Government Digital Service (2016a) Government Service Design Manual, [online] Available from: <https://www.gov.uk/service-manual> (Accessed 20 August 2016).

Government Digital Service (2016b) How the alpha phase works - Digital Service Manual - GOV.UK, [online] Available from: <https://www.gov.uk/service-manual/agile-delivery/how-the-alpha-phase-works> (Accessed 20 August 2016).

Government Digital Service (2016c) The shotgun and the service designer | Government as a Platform, [online] Available from: <https://governmentasaplatform.blog.gov.uk/2016/01/18/shotgun-and-service-designer/> (Accessed 20 August 2016).

Government Digital Service (2016d) What we mean by service design | Government Digital Service, [online] Available from: <https://gds.blog.gov.uk/2016/04/18/what-we-mean-by-service-design/> (Accessed 20 August 2016).

Grimes (2016) From the Editors: Service Design Policy, *Touchpoint*, 7(1), p. 3.

Hammersley, M. and Atkinson, P. (2007) *Ethnography: Principles in Practice*, 3rd Revised edition, Routledge.

Hedrick, T. E., Bickman, M. L. and Rog, M. D. (1993) *Applied Research Design: A Practical Guide: 032*, illustrated edition, Sage Publications, Inc.

Holstein, J. A. and Gubrium, J. F. (1995) *The Active Interview*, Sage Publications, Inc.

IBM Research (2010) IBM Research: Services Sciences, Management and Engineering | Services definition, [online] Available from: <http://www.research.ibm.com/ssme/services.shtml> (Accessed 13 February 2010).

International Organization for Standardization (1991) *ISO/IEC 9126 - Software engineering product quality*, International Organization for Standardization.

International Organization for Standardization (1999) *ISO/IEC 14598 Information technology. Software production evaluation.*, International Organization for

Standardization.

International Organization for Standardization (2007) *ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems*, International Organization for Standardization.

International Organization for Standardization (2011) *ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*, International Organization for Standardization.

Jacobson, I., Booch, G. and Rumbaugh, J. (1999) *The Unified Software Development Process*, Addison Wesley.

Johnston, R. (1995) The determinants of service quality: satisfiers and dissatisfiers ., *International Journal of Service Industry Management*, **6**(5), p. 53.

Kaner, S., Lind, L., Toldi, C., Fisk, S. and Berger, D. (2007) *Facilitator's Guide to Participatory Decision-Making*, 2nd Edition, Jossey Bass.

Kazman, R., Abowd, G., Bass, L. and Webb, M. (1994) SAAM: A Method for Analyzing the Properties of Software Architectures, In.

Kazman, R., Asundi, J. and Klein, M. (2001) Quantifying the costs and benefits of architectural decisions, In *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, Ontario, Canada, IEEE Computer Society, pp. 297–306, [online] Available from: <http://portal.acm.org.ergo.glam.ac.uk/citation.cfm?id=381473.381504&coll=ACM&dl=ACM&type=series&idx=SERIES402&part=series&WantType=Proceedings&title=ICSE&CFID=69202465&CFTOKEN=49512941> (Accessed 21 December 2009).

Kazman, R., Klein, M. and Clements, P. (2000) *ATAM: Method for Architecture Evaluation*, Technical Report, Software Engineering Institute, Carnegie Mellon University.

King, S. and Cotterill, S. (2007) Transformational Government? The role of information technology in delivering citizen-centric local public services, *Local Government Studies*, **33**(3), pp. 333–354.

Koivisto, M. and Miettinen, S. (2009) *Designing Services With Innovative Methods*, University of Art and Design Helsinki, [online] Available from: <http://www.ellibs.com/book/9789525018431/designing-services-with-innovative-methods> (Accessed 29 July 2016).

Le Grand, J. (1982) *The Strategy of Equality: Redistribution and the Social Services*, Routledge.

Lewis, R. C. and Booms, B. H. (1983) The Marketing Aspects of Service Quality, *Emerging Perspectives on Services Marketing*, pp. 99–107.

Ling, T. (2003) Ex Ante Evaluation and the Changing Public Audit Function The Scenario Planning Approach, *Evaluation*, **9**(4), pp. 437–452.

- Littau, P., Jujagiri, N. J. and Adlbrecht, G. (2010) 25 years of stakeholder theory in project management literature (1984–2009), *Project Management Journal*, **41**(4), pp. 17–29.
- Lovelock, C. and Gummesson, E. (2004) Whither Services Marketing?: In Search of a New Paradigm and Fresh Perspectives, *Journal of Service Research*, **7**(1), pp. 20–41.
- Lovelock, C. H. and Wirtz, J. (2007) *Services Marketing: People, Technology, Strategy*, 6th ed, Pearson Education.
- Maciosek, M. V. (1995) Behind the growth of services, *Illinois Business Review*, **52**(3), p. 3.
- Maffei, S., Villari, B. and Foglieni, F. (2013) Evaluating services An exploratory approach beyond Service Design, In [online] Available from: https://www.researchgate.net/publication/269996000_Evaluating_services_An_exploratory_approach_beyond_Service_Design (Accessed 20 August 2016).
- Marcus Vitruvius Pollio (25AD) *De Architectura*.
- Martin, J. (1991) *Rapid Application Development*, New York : Toronto : New York, Macmillan USA.
- Mehrizi, M. H. R., Ghasemzadeh, F. and Molas-Gallart, J. (2009) Stakeholder Mapping as an Assessment Framework for Policy Implementation, *Evaluation*, **15**(4), pp. 427–444.
- Mendelow, A. (1991) Environmental scanning: the impact of the stakeholder concept, In Cambridge MA.
- Merton, R. K., Fiske, M. and Kendall, P. L. (1990) *The Focused Interview: A Manual of Problems and Procedures*, 2nd ed, Simon and Schuster.
- Mitchell, R. K., Agle, B. R. and Wood, D. J. (1997) Toward a Theory of Stakeholder Identification and Saliency: Defining the Principle of Who and What Really Counts, *The Academy of Management Review*, **22**(4), pp. 853–886.
- Morelli, N. (2002) Designing Product/Service Systems: A Methodological Exploration., *Design Issues*, **18**(3), pp. 3–17.
- MORI Social Research Institute (2002) *Public Service Reform: Measuring and Understanding Customer Satisfaction*, London, MORI Social Research Institute, p. 46.
- Moritz, S. (2005) *Service Design: Practical access to an evolving field*, Köln International School of Design, [online] Available from: <https://uploads.strikinglycdn.com/files/280585/5847bd6a-e928-4f0f-b677-ed7df26fa1df/Practical%20Access%20to%20Service%20Design.pdf>.
- Murray-Webster, R. and Office of Government Commerce (2010) *Management of risk: guidance for practitioners*, 3rd ed., 2010 edition, London, Stationery Office.
- Office for National Statistics (2010) About ONS, Website facility, [online] Available from: <http://www.ons.gov.uk/about/index.html> (Accessed 28 November 2010).

- Parasuraman, A., Zeithaml, V. A. and Berry, L. L. (1985) A Conceptual Model of Service Quality and Its Implications for Future Research., *Journal of Marketing*, **49**(4), pp. 41–50.
- Perry, D. E. and Wolf, A. L. (1992) Foundations for the study of software architecture, *SIGSOFT Softw. Eng. Notes*, **17**(4), pp. 40–52.
- Perry, D. and Garlan, D. (1995) Editorial, *IEEE Transactions on Software Engineering*, **21**(4), [online] Available from: <http://ieeexplore.ieee.org/ergo.glam.ac.uk/xpl/selected.jsp?imageField.x=88&imageField.y=17&imageField=View+Selected+Items&chklist=385967%40ieejrns> (Accessed 22 December 2009).
- Philliber, S., Schwab, M. and Sloss, S. (1980) *Social Research*, F E Peacock Pub.
- Platt, J. (1992) 'Case Study' in American Methodological Thought, *Current Sociology*, **40**(1), pp. 17–48.
- Project Management Institute Standards Committee (1996) *A Guide to the Project Management Body of Knowledge*, PMIC, U.S.
- Randell, B. (1996) The 1968/69 Nato Software Engineering Reports, *NATO Software Engineering Conference 1968*, [online] Available from: <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOREports/index.html> (Accessed 31 January 2010).
- Rathmell, J. M. (1966) What Is Meant by Services?, *Journal of Marketing*, **30**(4), pp. 32–36.
- Redwine, S. T. and Riddle, W. E. (1985) Software technology maturation, In *Proceedings of the 8th international conference on Software engineering*, London, England, IEEE Computer Society Press, pp. 189–200, [online] Available from: <http://portal.acm.org/ergo.glam.ac.uk/citation.cfm?id=319568.319624&coll=ACM&dl=ACM&CFID=69393422&CFTOKEN=12006899> (Accessed 22 December 2009).
- Rossi, D. P. H., Lipsey, M. W. and Freeman, D. H. E. (2003) *Evaluation: A Systematic Approach*, Seventh Edition, Sage Publications, Inc.
- Rozanski, N. and Woods, E. (2011) *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*, 2 edition, Upper Saddle River, NJ, Addison Wesley.
- Rutter, J. (2012) *Evidence and evaluation in policy making*, Institute for Government, [online] Available from: <http://www.instituteforgovernment.org.uk/publications/evidence-and-evaluation-policy-making> (Accessed 24 September 2016).
- Rycroft-Malone, J., Seers, K., Titchen, A., Harvey, G., Kitson, A. and McCormack, B. (2004) What counts as evidence in evidence-based practice?, *Journal of Advanced Nursing*, **47**(1), pp. 81–90.
- Say, J. B. (1803) *A treatise on political economy; or, The production, distribution, and consumption of wealth*, J.B. Lippincott & Co.

- Schmidt, F. and Strickland, T. (1998) *Client Satisfaction Surveying: Common Measurements Tool*, Canada, Canadian Centre for Management Development, p. 50.
- SDN (2016) SDN | Service Design Network, [online] Available from: <https://www.service-design-network.org/> (Accessed 8 October 2016).
- Segelström, F. (2013) *Stakeholder engagement for service design : how service designers identify and communicate insights /*, Dept. of Computer and Information Science, Linköping University.
- Shaw, M. (1990) Prospects for an engineering discipline of software, *Software, IEEE*, **7**(6), pp. 15–24.
- Shaw, M. and Clements, P. (2006) The golden age of software architecture, *IEEE Software*, **23**(2), pp. 31–39.
- Shostack, G. L. (1982) How to Design a Service, *European Journal of Marketing*, **16**(1), p. 49.
- Shostack, G. L. (1984) Designing services that deliver, *Harvard Business Review*, **62**(1), pp. 133–139.
- Smith, A. (1776) *An inquiry into the nature and causes of the wealth of nations : a selected edition*, Oxford ;;New York, Oxford University Press.
- Software Engineering Institute (2009a) Classic Software Architecture Definitions, [online] Available from: <http://www.sei.cmu.edu/architecture/start/classicdefs.cfm> (Accessed 22 December 2009).
- Software Engineering Institute (2009b) Defining Software Architecture, [online] Available from: <http://www.sei.cmu.edu/architecture/start/definitions.cfm> (Accessed 22 December 2009).
- Software Engineering Institute (2016) A Framework for Software Product Line Practice, Version 5.0, [online] Available from: http://www.sei.cmu.edu/productlines/frame_report/architectureEval.htm (Accessed 21 May 2016).
- Software Engineering Standards Committee of the IEEE Computer Society (1998) *IEEE Standard for a Software Quality Metrics Methodology*, IEEE Standard, IEEE.
- Spohrer, J., Maglio, P., Bailey, J. and Gruhl, D. (2007) Steps Toward a Science of Service Systems, *Computer*, **40**(1), pp. 71–77.
- Stickdorn, M. and Schneider, J. (2014) *This is Service Design Thinking.: Basics - Tools - Cases*, 01 edition, Amsterdam, Bis Publishers.
- Sutherland, J. and Schwaber, K. (1995) Business object design and implementation, In Austin, TX, University of Michigan, p. 118.
- Tassi, R. (2009) Service Design Tools | Communication methods supporting design processes, [online] Available from: <http://www.servicedesigntools.org/> (Accessed 27 July 2009).

Thévenet, R. and Vincent, S. (2013) From Public Service Design to Public Policy Design: Opportunities and challenges of Service design in government, *Touchpoint*, 5(2), pp. 41–45.

UK Border Agency (2010a) Asylum, [online] Available from: <http://www.ukba.homeoffice.gov.uk/asylum/> (Accessed 28 November 2010).

UK Border Agency (2010b) Home Page, [online] Available from: <http://www.ukba.homeoffice.gov.uk/> (Accessed 28 November 2010).

Vedung, E. (1999) *Public Policy and Program Evaluation*, Transaction Publishers.

Welsh Assembly Government (2004) *Making the Connections: Delivering Better Services for Wales*.

Welsh Assembly Government (2006) Beyond Boundaries: Citizen-Centred Local Services for Wales, General, [online] Available from: <http://new.wales.gov.uk/topics/improvingservices/strategy/beyondboundaries/?lang=en> (Accessed 17 December 2008).

Wisniewski, M. and Donnelly, M. (1996) Measuring service quality in the public sector: the potential for SERVQUAL, *Total Quality Management*, 7(4), pp. 357–366.

Wolman, D. (2008) Before the Levees Break: A Plan to Save the Netherlands, [online] Available from: http://www.wired.com/science/planetearth/magazine/17-01/ff_dutch_delta?currentPage=all (Accessed 20 February 2010).

Yin, D. R. K. (2008) *Case Study Research: Design and Methods*, Fourth Edition, Sage Publications, Inc.

Zeithaml, V. A., Parasuraman, A. and Barry, L. L. (1990) *Delivering Quality Service: Balancing Customer Perceptions and Expectations*, illustrated edition, The Free Press.