School of Design, Engineering and Computing

Software Systems Research Centre

The derivation of a pragmatic requirements framework for web development

by

Sheridan Paula Jeary

In partial fulfilment of the award of Doctor of Philosophy

May 2010

The derivation of a pragmatic requirements framework for web development

Sheridan Paula Jeary

Web-based development is a relatively immature area of Software Engineering, producing often complex applications to many different types of end user and stakeholders. Web Engineering as a research area, was created to introduce processes that enable web based development to be repeatable and to avoid potential failure in the fast changing landscape that is the current ubiquitous Internet. A survey of existing perspectives from the literature highlights a number of points. Firstly, that web development has a number of subtle differences to Software Engineering and that many web development methods are not used. Further, that there has been little work done on what should be in a web development method. A full survey of 50 web development methods finds that they do not give enough detail to be used in their entirety; they are difficult for a non-computer scientist to understand in the techniques they use and most do not cover the lifecycle, particularly in the area of requirements, implementation and testing.

This thesis introduces a requirements framework for novice web developers. It is created following an in-depth case study carried out over two years that investigates the use of web development methods by novice developers. The study finds that web development methods are not easy to understand, there is a lack of explanation as to how to use the techniques within the method and the language used is too complex. A high level method is derived with an iterative process and with the requirements phase in the form of a framework; it addresses the problems that are discussed and provides excellent support for a novice web developer in the requirements phase of the lifecycle. An evaluation of the method using a group of novice developers who reflect on the method and a group who use it for development finds that the method is both easy to understand and use.

Contents

List of Figures

List of Tables

Acknowledgement

I could never have completed this work without the encouragement and support I have received from Keith Phalp and the unswerving belief, love and support of Georgette. I thank you both.

Authors declaration

Some of the material presented in Chapters 2, 3, 5 and 7 has appeared in papers as listed below.

Jeary, S., Phalp, K, & Vincent, J., (2009), An evaluation of the utility of web development methods, Software Quality Journal, Vol. 17, 2, pages 125-150, June 2009.

Jeary, S., Phalp, K. T. and Vincent, J., 2007. Are Web Development Methods Too Difficult To Use? In: 15th International Software Quality Management Conference/INSPIRE conference, 2-4 April 2007, Stafford, England

Jeary, S., Phalp, K. T. and Vincent, J., 2007. Web Development Methods and Techniques – A Survey. In: 15th International Software Quality Management Conference/INSPIRE conference, 2-4 April 2007, Stafford, England.

Jeary, S., Phalp, K., Xu,L., deVrieze, P. , 2010. A requirements framework for novice web developers. In 1st Workshop on The Web and Requirements Engineering, 28 September 2010, Sydney, Australia.

To my late mother

Chapter 1

Introduction

1.1 Motivation

Web application development is a relatively new field of software engineering. It has been described as ad hoc-based, quick and dirty development of small sets of web pages mainly used for toy purposes, information publishing or advertising (Cartensen and Vogelsang, 2001). There is concern that this approach to Web development will lead to problems in the successful development, deployment and maintenance of Web systems in the long term (Ginige and Murugesan, 2001b). This has given rise to the term 'tangled web' implying that the web is becoming "a morass of poorly developed web applications that have too high a probability of failure"(Pressman, 2000). The term 'web crisis' has been discussed (Murugesan, Deshpande, Hansen and Ginige, 2001, Pressman, 2000, Ginige and Murugesan, 2001b) with the view that the software development process has not been adapted to the new challenges of the internet.

These challenges have given rise to the need for an engineering approach to web development such that development is disciplined and systematic (Lowe and Hall, 1999) and the field of research that addresses this is called Web Engineering. Web Engineering requires methods, techniques and tools that will ensure a cost effective and quantifiable approach to web development. A number of web development

methods have been available since 1995, which aim to provide this type of approach but do not appear to have resolved these issues. Hence this work is timely in its exploration of the inadequacies of the field of web development methods.

1.2 Problem and Research Aims

Much of the literature on web application development has argued that it is different from traditional software application development and has cited numerous reasons. These range from the more philosophical viewpoint that the web creates a greater mesh between art and science than general software development and that it focuses on visual creativity (Murugesan, Deshpande, Hansen and Ginige, 2001). Other examples include, the fact that web applications tend to have shorter timeframes with smaller budgets (Lowe, 2003) implying that traditional methods appear to be too time consuming to apply. In addition, web technology and web development tools are evolving very fast; meaning that the need to use a systematic design approach has not been recognised by web developers (Balasubramanian and Bashian, 1998). In addition, many web developers are immature in their experience of the web and have no knowledge of computer science or development methods, thus they may have little understanding of, nor consider use of, a development method (Powell, Jones and Cutts, 1999, Vora, 1998, Murugesan, Deshpande, Hansen and Ginige, 2001). The premise of web development methods, for example (Gnaho, 2000, De Troyer, 2001, Schwabe and Rossi, 1998, Fraternali and Paolini, 2000, Escalona and Aragon, 2008), is that to build successful, 'good quality', complex web based systems, web

developers need to adopt a disciplined and systematic approach (Murugesan and Ginige, 2005).

There has been a great deal of work in software engineering to ensure the adoption of systematic requirements gathering and an engineering based approach to analysis and modelling (Nuseibeh and Easterbrook, 2000). However, it is evident that many web development methods are weak in the area of requirements; assuming that these will already be in the possession of the developer before development begins. In addition, a number of web development methods have been proposed and used that would assist in a more repeatable and methodical approach to software development; such as Atern (DSDM Consortium, 2009) and the Unified Process (IBM, 2009) however, these are commercial development methods that are proprietary and have to be subscribed to, or paid for, to get full details.

There have been three previous surveys on the use of development methods specifically for the web; however, these conclude that web development methods are rarely used. One, is a survey by researchers in the hypermedia community (Barry and Lang, 2001b) to businesses in Ireland using a questionnaire for their hypermedia developers. The results indicate that web development methods are not being used and that many companies use their own in-house methods. Taylor (2002) published a study with similar findings. Lang and Fitzgerald (2005) found in their survey of web developers that only two percent of respondents had ever used

a web development method. There is therefore a need to understand why developers do not use web development methods.

A number of reasons have been put forward as to why web development methods are not used, such as the short time scales of web development and the fact that many developers do not have a software engineering background. Another reason put forward for the non-use, is that they are too difficult to use (Barry and Lang, 2001b). This is confirmed by anecdotal evidence obtained from a number of web developers and university lecturers at a conference where the author was presenting this work; and which also highlighted the difficulty of finding a suitable web development method to teach to undergraduates who are prospective web developers. There has been no published work in this area.

The aims of this research are therefore to investigate and identify:

1 Whether web development methods are used?

2 Which web development methods are available? What is their scope of coverage? What guidance is available for their use?

3 Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?

4 What components, techniques and tools should constitute a web development method?

5 Whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?

6 Whether it is possible to take the findings of this work and use them to inform the design of a web development method that is suitable for novice web developers?

1.3. Key contributions

The main achievements of the work are:

1. A survey of 52 web development methods which categorises the methods from a practitioner's point of view. It shows the scope of the methods across the Systems Development Life Cycle. This highlights a problem of many web development methods in that they do not cover the early or later stages of the lifecycle and particularly concentrate on modelling content, navigation and presentation. It shows how the sources are from a broad cross section of the research community and published in a similarly broad cross section of sources.
2. The first case study shows how 23 students, who may be classed as novice developers, approached using a web development method and the issues that arose. The findings highlighted the inaccessibility of current web development methods when all but one of the student developers abandoned the use of the method. A number of factors appear to have contributed to this abandonment; such as the methods were too complicated, required too many products to be created and lacked guidance in terms of the applicability of the method to their specific project. There was little guidance to aid understanding and the language was considered academic and 'intellectual'. The scope did not cover

the lifecycle and the requirements phase was particularly commented upon, as being missing.

3. The work is of importance to the Software Engineering community because if components of methods or techniques can be identified as complex or difficult to use, then web development method designers can ensure that the component is either changed or adequate explanation is given for the use of that component within a web development method.
4. Finally, an overview web development method called the PECS method (Pragmatic, Effective, Common–sense Simple) is created from the collected novice developer views which focuses particularly on the requirements phase of the development life cycle.
5. The evaluation of the method shows that the method is easy to use and understand.

In summary, the novelty and contribution will thus be to have furthered the understanding of the use of web development methods; and the design and production of a web development method which reflects the views of novice web developers.

1.4. Thesis structure

Chapter 2, *Perspectives on Web Development,* presents a number of areas which are pertinent. It starts with a thorough literature review illustrating how web development is different. It highlights three viewpoints of the difference; the user domain, the developer and development team and their domain, and finally environment issues. This is followed by a discussion on the use and non-use of web

development methods. Finally it reviews the contents of a web development method as discussed by a variety of authors from the literature.

Chapter 3, *Survey of Web Development Methods,* reviews the literature to allow the reader to become acquainted with the field of web development methods. The literature is categorised to show the variety of sources of web development methods and the different communities where they originate. The survey itself is categorised in pragmatic terms that will allow a practitioner to select a method according to specific criteria such as scope. Finally a discussion is presented which explores the notion of difficulty of use.

Chapter 4, *Research Approach,* describes the selection of Creswell's conceptual framework which is used to inform the research approach. Following analysis of the different approaches the research is presented as qualitative and using an inductive approach to theory building. The data collection and analysis processes which inform the research take place within a case study and use both documents and semi-structured interviews. The inductive approach is used within template analysis which informs both the method creation and its later evaluation.

Chapter 5, *Case Studies,* presents the case study design, the data collection and analysis for four case studies. The first study involves twenty three students over two years. This shows that web development methods are not easy to use and that there are issues with their use. These findings are added to a template alongside the findings of the second study which outlines the components that the students, as

novice developers, feel should be in a web development method. The findings of a third study which relate to an in-depth study by three students are also presented to further inform the case study. Finally, four industry interviews are reported which were conducted to test the relevance of the information presented by the student novice developers.

Chapter 6, *Creation of the Requirements Framework,* reviews the requirements literature relating specifically to web development and highlights the mismatch between their importance and their absence in web development methods. It then presents the creation of the PECS (Pragmatic, Effective, Common–sense Simple) web development method which focuses specifically on the requirements component.

Chapter 7, *Evaluation,* details the evaluation of the PECS web development method using an adaptation of Sol's generalised measurement tool. Sixty three students evaluate the method by creating their own list of features and comparing them to the method; 569 comments were received which are analysed inductively. The findings are presented in the categories of the method and discussion follows which details those findings which result in amendment to the method and which findings need further work.

Chapter 8, *Conclusions and Further Work*, concludes the thesis. It summaries how the research questions have been answered and how the work adds to the body of knowledge before discussing areas for further work. This includes completing the

method to include using components such as testing and maintenance; and investigation of using a problem frame approach to specific techniques within the PECS method.

Appendix A, B, D and F show sample mind maps used as part of the work in Chapters 5 and 7; Appendix C is a copy of the assignment used by students to evaluate the PECS method. Appendix E details the questions used in interviews with web developers in industry and Appendix G details the full method.

It should be noted that the sequencing of Chapters 5, 6, and 7, and the studies within them, follows the research approach as identified in Chapter 4; it is important to recognise that the outcomes of each of the studies informed the design and conduct of the next study or phase, and that this is part of the method.

Chapter 2

Perspectives on Web Development

2.1 Introduction

Web development and its associated methods are a relatively immature area of research in comparison to the field of Software Engineering. However, it has been recognised that Web-based development delivers a large number of complex applications to many different types of end users and that development practices may have not improved since the early days of the Internet (Ginige and Murugesan, 2001a).

This chapter explores the perspectives that surround web development and aims to show the diversity of areas that have contributed to the literature on web development and its methods. It begins by exploring whether web development is different to software engineering, before looking at the way different researchers have classified web sites. The following section examines the terminology that the method authors use to describe their methods. Finally the work explores contents of a web development method, and in the absence of any discussion in the literature, uses work that has detailed web development method comparisons to identify what different authors consider should constitute a web development method.

2.2. Is web development different?

There have been a number of discussions outlined in the literature concerning why web development may be different from other forms of software engineering. The differences may be divided into three main areas: the user domain, the developer domain and the environment which are summarised in Table 1.

2.2.1. User Domain Issues

The web at the interface is of primary importance to the user and is an area where there are a number of differences. The user may potentially enter any web site at any point and not as traditionally expected at the start of an application. They will also leave at any point (Barry and Lang, 2001b). In addition, the technology is more visible to the user and they will blame any delay in the internet technology transmission on the site they are visiting (Lowe, 2003, Powell, Jones and Cutts, 1999).

When creating a new web application for a client the web developer rarely has any contact with any expected user of the site, except through email and survey, and thus the user is often misrepresented in a development (Holck and Clemmensen, 2001). In addition, if they do not like the results of the development they are only a mouse click away from going to the competition (Glass, 2001).

2.2.2. Developer and Development Team Domain Issues

The developer domain is important when considering web development methods, and exhibits a number of differences from the software engineering development or developer's domain. One of the major differences is in the demographic of the team. For example, web development teams typically contain graphic designers and marketers, and the developer has a lack of experience and qualifications (Powell, Jones and Cutts, 1999, Vora, 1998, Holck, 2003, Overmyer, 2000). This can be an important factor in the adoption of a development method. The web developer is unlikely to have formal computing qualifications and knowledge. Barry and Lang (2002) believe that it is not only difficulty in using or understanding formalised methods that is the inhibiting factor against their use, but the fact that they are also too cumbersome. However, this point is conjecture and there is no evidence to support it. Finally, the technology and development tools in web development are rapidly changing, which means that the web developer needs to be continually updating their skills (Cartensen and Vogelsang, 2001, Lowe, 2003).

2.2.3. Environment Issues

The environmental differences consist particularly of the linkages between the business and its web site and the way that the web is 'hype driven' (Powell, Jones and Cutts, 1999). This creates shorter life cycles and shorter time frames for initial delivery (Overmyer, 2000). If, in addition, the link between the business architecture and the technical design is tighter than conventional software systems; and that the web development may change the way the business operates (Lowe,

2003), there is a considerable difference to the traditional development model. Finally, the evolution and maintenance of the site is fine grained and continually changing, which Lowe (1998) likens to gardening.

Domain	Difference	Source
User domain	Most web-based systems need to cater to users with a diverse range of skills and capability	(Murugesan, Deshpande, Hansen and Ginige, 2001)
	The user is divorced from the development and may not be engaged with the process	(Holck and Clemmensen, 2001)
	There is no traditional entry or exit point from the web pages	(Barry and Lang, 2001b)
	The technology is more visible to the user	(Lowe, 2003)
	There is a high reliance on the user interface	(Lowe, 2003, Murugesan, Deshpande, Hansen and Ginige, 2001)
	The non-functional requirements may be primary	(Holck and Clemmensen, 2001)
	There is volatility in the user requirements	(Lowe, 2003)
	Unpredictable publishing environment where users blame the site	(Powell, Jones and Cutts, 1999, Murugesan, Deshpande, Hansen and Ginige, 2001)
Developer	Development teams contain graphic designers, marketers etc. with different disciplinary emphasis	(Overmyer, 2000, Holck and Clemmensen, 2001, Murugesan, Deshpande, Hansen and Ginige, 2001)
	Aesthetic and cognition	(Nanard and Nanard, 1995, Murugesan, Deshpande, Hansen and Ginige, 2001)
	Developers suffer from cognitive overload	(Lowe and Hall, 1999)
	Developer immaturity in experience and as a resource to use in estimation	(Powell, Jones and Cutts, 1999, Vora, 1998)
	Uncertainty in the developer domain	(Lowe, 2003)
	Rapidly changing technology and tools	(Cartensen and Vogelsang, 2001, Lowe, 2003)
	Lack of useful methods	(Cartensen and Vogelsang, 2001)
	Methods are written for computer scientists	(Fernandez, Florescu, Levy and Suciu, 2000)
Environment	The linkage between the business architecture and the technical design is tighter than conventional software systems	(Lowe, 2003)
	Web sites are like magazines, design of information structure	(Overmyer, 2000)
	Shorter life cycles, aggressive release demands, shorter time frames for initial delivery	(Overmyer, 2000, Murugesan, Deshpande, Hansen and Ginige, 2001)
	Hype driven	(Powell, Jones and Cutts, 1999)
	Immaturity of web development	(Powell, Jones and Cutts,

		1999)
	The development changes the business model	(Lowe, 2003)
	The development is of a sophisticated business model, a complex and a component based information architecture	(Lowe, 2003)
	Highly competitive, market environment	(Lowe, 2003)
	Fine grained evolution and maintenance	(Lowe, 2003)

Table 1: Three perspectives on the different nature of web development

2.3. Classification of web sites

When reference is made to web development, the application that the developer builds can fit anywhere on a broad spectrum; from a web site of linked pages in HTML to a site based on the semantic web. This section looks at the work that has been done on web site classification to build a picture of the kinds of development that a web development method would need to support.

Holck and Clemmensen (2001) attempt to classify web sites by discussing them in terms of genre; including Contents, Transactions, Branding, Service and Entertainment, with the addition of message delivery and bidirectional communication. This categorisation is similar to that of Ginige and Murugesan (2001b) who use the categories of Informational, Interactive, Transactional, Workflow, Collaborative Work Environments, On-line Communities and Marketplaces and Web Portals. Chiu (2002) believes that Web Information Systems are either re-engineered Information Systems using Web technology or purpose built. They include decision support, executive information, finance, data mining, on-line analytical processing, knowledge management and digital libraries. Vidgen (1998) introduces maturity with his classification using a marketing perspective. He classifies web sites into low, medium and high maturity. The early web sites at low

maturity consist of many pages of text, often linked only by page and reflecting the business goals. They appear as an online company brochure which is later enhanced with product and sales promotion information. From this a searchable database can be implemented and web page content can be individually customised based on customer defined criteria. From here medium maturity is the establishment of intranets and ordering and shopping cart/transaction processing systems extending to offering new product combinations and the definition of business processes. High maturity would come from strategic alliances between organisations and being able to deliver new products to new markets. With the advent of Web 2.0 additional classifications could be added to include perhaps Enterprise 2.0, folksonomies, social networks, blogs and wikis, mashups and virtual worlds.

Differing communities have their own perspectives on web development. The database community views the Web as a collection of databases and views the programs that extract and process Web data automatically as database applications (Arocena and Mendelzon, 1998). The hypermedia community view hypermedia as creating value-added support for the functionality of web sites, making applications more effective (Chiu, Bieber and Lu, 2002) Web architects view the layers or tiers as the important categorisation and are increasingly using the separation of concerns as an overriding principle (Eichinger, 2006). Powell (1999) classifies web sites on a matrix between simple and complex and document centred and application centred and this useful categorisation can be updated to include

more modern web applications such as portals, the semantic web and the ubiquitous web (Kappel, Proll, Reich and Retschitzegger, 2006) as shown in Figure 1. A document centric web site consists of a selection of web pages usually handcrafted by an individual in HTML with basic linking of the contents. Whereas an interactive site is a Web Information System which dynamically generates its contents (Chiu, Bieber and Lu, 2002) .

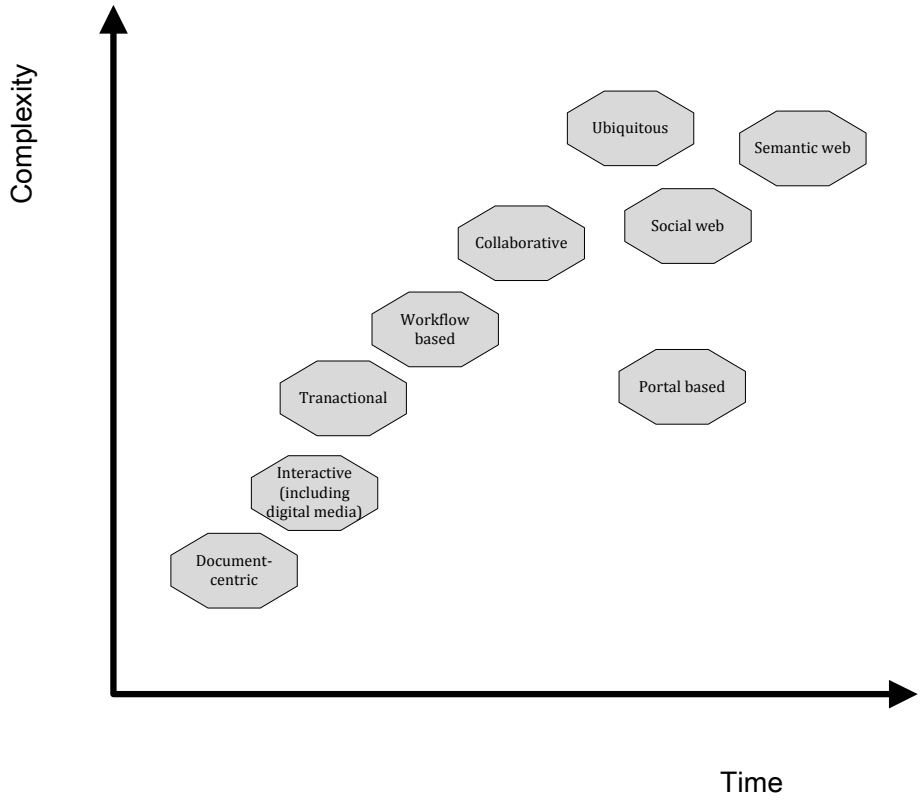


Figure 1: A classification of web sites (adapted from (Kappel et al., 2006))

2.4. Terminology

There has been much discussion about Web Systems using a multitude of different names such as Web Information Systems and Web-based systems (Holck, 2003), but alluding to a similar concept. Holck (2003) believes that the different views of

different authors can be accounted for by the authors different perspectives of the field and this is also true for other terms used by authors researched in this work. It is evident that many of the authors of web development 'methods' papers are unaware of any categorisation of web development methods, models, techniques and tools. This has meant that the area has much ambiguity and this research has found that many things are termed methods when they are often techniques or tools, or they only cover some phases or sub-phases of the Systems Development Life Cycle yet label themselves as Web Engineering methods implying the full cycle. Thus, the terms have been found to be used interchangeably.

Discussion in the Information Systems community about the term 'methodology' and 'method' continues and is summarised by Fitzgerald et al. (2002) but the definition put forward by Avison and Fitzgerald (2006) can be slightly amended to specify a web development method and is the definition that will be used in this work.

> '*A collection of techniques, tools and documentation aids which will help* ***web*** *developers in their efforts to implement a new* ***web*** *information system. A method will consist of phases, themselves consisting of sub-phases, which will guide the systems developers in their choice of techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate* ***web*** *information systems projects'*.

A model is a visual representation of some aspect of the real world (Lieberman, 2003) and the web development community uses modelling languages and

notations, often formally described, or modelling techniques to design the information, navigation and presentation aspects of a web application. Many of the modelling techniques described in the literature are in fact modelling methods and vice versa.

There is a more general consensus of the term technique which Fitzgerald et al. (2002) explain as the 'how to do' to a method's 'what to do'. A technique will explain how to carry out an activity, but a particular method may recommend specific techniques to carry out the development activities (Avison and Fitzgerald, 2006). A tool is an artefact involved in the development process either to enable a method or a technique (Avison and Fitzgerald, 2006).

2.5. The use and non-use of Web development methods

This section of the literature review explores whether web development methods are used in response to Research Question 1 from Section 1.2. There have been four surveys of web development practitioners, two conducted as questionnaires (Barry and Lang, 2001b, Lang and Fitzgerald, 2005) and two using interviews (McDonald and Welland, 2001a, Taylor, McWilliam, Forsyth and Wade, 2002). In addition there have been a number of in depth interviews conducted (Britton, Jones, Myers and Sharif, 1997, Whitley, 1998, Cartensen and Vogelsang, 2001) although only one study is recent (Lang and Fitzgerald, 2007). From this work it is apparent that web development methods are not used or the level of usage is 'negligible' (Lang and

Fitzgerald, 2007). However, there has been no research into the reasons for the non-use of the web development methods although there has been considerable supposition. Reasons put forward include that web development methods are too cumbersome or are too difficult to use (Barry and Lang, 2001a) that developers are inexperienced (Powell, Jones and Cutts, 1999) and do not know of the existence of methods as they are not software or systems engineers (Fernandez, Florescu, Levy and Suciu, 2000).

Taylor et al. interviewed 25 developers in a variety of organisations (Taylor, England and Gresty, 2001, Taylor, McWilliam, Forsyth and Wade, 2002) who were all using advertising and informational web sites. They found that development mostly occurred in an 'ad-hoc manner at the discretion of the individual developer' (Taylor, McWilliam, Forsyth and Wade, 2002 p.386) and that 68% used no design techniques. The design techniques that were used were informal hierarchy charts, page layout and storyboards. This appears to be despite the organisation using more formalised methods for more traditional development. No reasons were put forward for the non-use.

Barry and Lang (2001b) conducted a questionnaire of multimedia developers in general industry and multimedia companies in Ireland, and their results contrast with Taylor's in that only 25.6% of the respondents did not use a method at all. The study showed that developers find web development difficult and complex and that even advanced Rapid Application Development processes are not robust enough to assist in development practices. Most developers had not heard of any

web development methods, and Barry and Lang (2001b) comment that even HDM (Garzotto, Paolini and Schwabe, 1993) as a 'reference' web development method has never been used. They wonder if it is because it is too difficult to apply (Barry and Lang, 2001b).

A survey by Lang and Fitzgerald (2005) suggest that 16% of organisations have no process in place for web development and most find this a problem. However, they themselves comment that the section in their survey asking for details of hypermedia development methods that the developers had used was poorly answered. They received many ambiguous replies, however less that 2% had ever used a web development method, and less than 5% had ever heard of them. They believe that the answer lies in the fact that understandability and ease of use are major issues in method selection.

McDonald and Welland (2001a) interviewed fifteen people in seven organisations. Seven of the interviewees had a development process in place for web development, however only two of them were found to have an industry standard development method in place and they both worked for the same organisation. The rest had an in-house process which appeared to consist of a scoping document that covered requirements and design, and a second deliverable of the application itself.

Other work, in the area of multimedia is all from within the UK and Ireland. Britton et.al. (1997) interviewed multimedia developers who were developing instructional software and, although a little dated, found that there was little use of formal

modelling techniques and wondered whether it was an issue of training. There have been two interpretive studies of web development. One is the work of Cartensen and Vogelsang in Denmark (2001) who conduct a field study in a web development department of a software development company where they found that implementation of a web development method had problems in implementation. They argue that web development methods should not be too complex or require specialised skills in modelling and specification. The second work is that of Lang and Fitzgerald (2007) who interview respondents to their survey and found that the use of web development methods was negligible.

Explanations given for the non-use of web development methods include that developers do not know that the methods exist because they are unaware of the academic literature (Barry and Lang, 2001b, Taylor, McWilliam, Forsyth and Wade, 2002) and there are problems in adapting traditional approaches to web development (Cartensen and Vogelsang, 2001). The average web developer is inexperienced (Powell, Jones and Cutts, 1999) and does not come from a computer science background and in contrast writers of web development methods are not web developers but software engineers interested in semantics and data analysis (Fernandez, Florescu, Levy and Suciu, 2000). This mismatch of method user and method creator is likely to create a chasm between what may, or may not, be achievable by a web developer. Whitley (1998) discusses the use of a method which may not be used as the author intended and may be part of the reason for the mis-match. Many web developers use the Graphical User Interfaces (GUI's) of Rapid

Application Development (RAD) tools to facilitate the development process (Fernandez, Florescu, Levy and Suciu, 2000) and know nothing about the lifecycle and web development processes and techniques. Other reasons suggested are that web development methods are not universally applicable and have not been sufficiently tested in live situations (Barry and Lang, 2001b), and that the technology and tools are immature and that web development reflects an immature discipline (Cartensen and Vogelsang, 2001).

The term 'web developer' could cover a variety of different backgrounds and responsibilities. Only 33% of web developers in Lang and Fitzgerald's survey (2007) are from a software developer background whilst another 26% are from graphic design. The rest are from a category of such titles as 'information architects', 'web designers' (Lang and Fitzgerald, 2005) and 'new media managers'. These web developers are likely to have a mix of backgrounds and thus their understanding of the development process is likely to be mixed. These findings are corroborated by McDonald and Welland (2001a) who have similar findings. Academic supposition states that many developers find web development methods cumbersome (Barry and Lang, 2001b) and that for method selection a web development method needs to be to be understandable and easy to use (Barry and Lang, 2001b, Lang and Fitzgerald, 2005). It is also expected that it will not be too complex or require specialised skills in modelling and specification (Cartensen and Vogelsang, 2001). It is possible that these reasons all taken together may account for the non-use of web

development methods, however there has been no work done to date to explore this.

Kushawa (2006) highlights the complexity of web development and suggests that there are a number of factors which lead to it requiring greater cognitive skills than traditional software development. These factors are all related to the reasons that are given for why web development is different to software development, as described in Table 1. Any development which is, in itself complex, will need methods that will assist with managing that complexity. In addition if, as reported, a large number of web developers are not computer scientists or hypermedia specialists (Holck and Clemmensen, 2001, Overmyer, 2000, Lang and Fitzgerald, 2005), then any notation or method needs to be clearly defined and simple to use, and thus needs to be learnable by its intended audience.

In summary, it would appear from the literature that web development methods are not used and, in addition, many developers had never heard of any web development method. This directly answers Research Question 1 as to whether web development methods are used.

2.6. The lack of use of development methods

It should be commented upon here that there has been considerable debate about the use and non-use of development methods for the analysis and design of software in general. This is usefully summarised by Avison and Fitzgerald (2003b)

who class the current time as the post-methodology era where organisations are turning to newer methods or abandoning methods completely. They discuss the reasons behind this citing lack of developer productivity, the methods being overly complex leading to requirements bloat, with costly and difficult to use tools and techniques (Avison and Fitzgerald, 2006). In addition the method often assumes that all projects have the same method requirements and thus can be inflexible, not allowing changes in requirements, inhibiting creative thinking and are often difficult to adopt because of developer resistance (Avison and Fitzgerald, 2003b).

There is a 'dearth' of work on the determinants of methodology acceptance and little research into the determinants of individual software developers intentions to use or not use methodologies in an organisational context (Riemenschneider, Hardgrave and Davis, 2002). Much of the work relates specifically to 'tool' as opposed to methodology introduction or transferring the use of tool acceptance models to methodologies.

2.7. Contents of a web development method

This section explores the background in the literature to Research Question 4: what components, techniques and tools should constitute a web development method. There has been very little research done on the requirements for the contents of a web development method. To investigate what should be in a method it has been necessary to look at the work researchers have done in method comparison. By detailing the taxonomy for what should be in a method, allowing a user to compare it to another, it may be possible to see what the contents of a method should be. A

number of academics from differing communities have completed a method comparison between a number of methods and have used a comparison framework. Whilst these comparison frameworks are not as necessarily detailed as Avison and Fitzgerald's, (2003a) for example not including ontological or epistemological comparisons, some could be considered useful as a starting point for the contents of a method. It should be noted here that the authors are often highlighting their own method, thus they will select features that offer a good comparison and thus such as list may be, to some extent, flawed.

2.7.1. From the database community

Garzotto et al (1995) compare applications using content, structure, presentation, dynamics and interaction, but focus on structure and dynamics. Their evaluation criteria are richness, ease of use, consistency, self-evidence, predictability, readability and reuse. They consider this to be a useful starting point for comparing methods.

Fraternali (1999) whilst describing tools and approaches to data-intensive web development uses lifecycle coverage, process automation, modelling abstractions, reuse and components, default architecture and support to usability as categories. With each tool that he surveys, he breaks it down to show the way that it fits into these categories. He terms these as perspectives. Whilst his survey is aimed at tools he does compare five database research projects (Araneus (Mecca et al., 1998), Autoweb (Fraternali and Paolini, 2000), Strudel (Fernandez et al., 1997),

WebArchitect (Takahashi and Liang, 1997) and W3I3 (Ceri, Fraternali, Paraboschi and Pozzi, 1998) and he also selects two web development methods (RMM (Isakowitz, Stohr and Balasubramanian, 1995) and OOHDM (Schwabe and Rossi, 1995)). The criterion that is used for the selection of the projects and methods is not apparent.

2.7.2. From the hypermedia community

Christodoulou et al. (1998) evaluate Hypermedia Application Development and Management Systems looking to include a development and management methodology and its environment in their evaluation. The purpose is to study a development method, its environment and its system in relation to specific application requirements. These requirements are for a hypermedia system that provides an extensible abstract data model that is easily mapped or converted into a static or dynamic application. In addition it needs to support the integration and reuse of content in other hypermedia systems. Their framework places methods into six main approaches and for each approach they take one method as atypical. Their approaches are Object-Oriented, Entity-Relationship, component-based, hybrid–heterogeneous sources, open hypermedia and other miscellaneous. They then create forty six evaluation criteria. The criteria are detailed, covering from structural constraints, to node and link features to version control. They then compare the approaches to the criteria. With so many evaluation criteria it is perhaps not surprising that the study concludes that no method efficiently covers them. In a second paper (2000) they refine the criteria and believe that the basic requirements for a web development framework are that it is modular allowing

abstract components or modules to operate through their interfaces and technology independent. The rest of the paper deals with the requirements for a development framework of an automated tool or environment and has thus been discounted in this part of the survey.

2.7.3. From the modelling community

Retschitzegger and Schwinger (2000) look at the requirements for a modelling method for web development. They believe that it should be described in three dimensions. The first of these dimensions is 'levels' (presentation, hypertext and content). They believe that there should be clear separation between the levels yet flexible mapping and design should be either bottom up or top down. The second dimension, aspects, looks at the structure and behaviour of the three levels. They discuss the requirement for a modelling formalism and the use of patterns to support reuse. The third dimension is the modelling phases from analysis to implementation and support from a development process. They use this framework to compare eight methods. They conclude that there are a number of shortcomings in the methods as most do not conform to the requirements they have listed.

Koch (1999) uses a list from Henderson-Sellers (1995) who is discussing the requirements for methods in general and not web development specifically. She compares eight methods. The nine requirements are a full life cycle process, a full set of concepts and models that are internally consistent, a collection of rules and guidelines, a full description of deliverables, a workable notation, a set of metrics

with advice on quality, standards and test strategies, guidelines for project management, advice for library management and reuse and identification of organisational rules. Koch (1999) concedes that most of the methods developed for hypermedia systems only partially cover the life cycle and focus on the design. She concludes that research is needed to improve and test methods that cover the complete life cycle.

2.7.4. From the multimedia community

Woukeu (2003) looks at web design models using Hall and Lowe's framework (1999) to compare a number of methods and identify a shortfall in the way methods deal with the content.

2.7.5. From the web engineering community

Montero et al.(2002) take a number of software engineering requirements and hypermedia requirements from both the literature and personal practice to present a framework for comparing methods. These include describing a formal process, being model-based, providing products to specify functional, non-functional and usability requirements and validation rules for each design product from the software engineering domain. From the hypermedia domain these include describing the problem domain in terms of hypermedia components, allowing the modelling of the navigation structure, organisation and harmonisation of multimedia contents and providing conceptual tools to formalise security policies. They use this framework to compare seven methods.

Lee divides methods into formal and informal methods. The formal methods are split into Entity-Relationship model based and Object-Oriented Model based. The methods are then compared according to method phases, from customer analysis to implementation and maintenance (Lee, Suh and Lee, 2004).

Other work that was considered was Russo and Graham (1998) who produced a 'learning by doing' draft method. The criteria that the team had for the development could have been useful; however the team were from a structured background and considered the elements in a traditional life cycle manner.

The issues addressed have been collated together and are shown at Table 2.

	Papers						
Comparison attribute/ functional area in development	**Montero (Montero, Diaz and Aedo, 2002)**	**Koch (Koch, 1999)**	**Retschitzegger and Schwinger (Retschitzegger and Schwinger, 2000)**	**Lee (Lee, Suh and Lee, 2004)**	**Woukeu (Woukeu, Carr, Wills and Hall, 2003)**	**Christodoulou (Christodoulou, Styliaras and Papatheodourou, 1998)**	**Fraternali (Fraternali, 1999)**
Process	Describe a formal process to guide the development of a software application	Full life-cycle process A collection of rules and guidelines	Modelling phases From analysis to implementation. Process support	Phases	Is the full development lifecycle supported? Is process management supported?		Lifecycle coverage
Deliverables		A full description of deliverables					
Automation	Software support tools to support the process						Process automation
Approach				Approach: Formal or informal ER or OO		Approach: OO ER Component-based Hybrid Open systems Other	
Models	Contain a model to describe the real world and transfer it to a physical system	A full set of concepts and models that are internally consistent	Levels: Presentation Hypertext Content Clear separation between levels Aspects: Structure and behaviour of the three levels	Source of navigation design	Separation between levels	Conceptual data model design.	Modelling abstractions
Notation	Include validation rules for each design product	A workable notation	Need for modelling formalism				
Mappings	Maintain integrity relationships among phases		Flexible mapping between levels				
Reuse	Design reuse	Advice for library management and reuse	Use of patterns to support reuse		How is reuse addressed?		Reuse and components
Hypermedia	Allow the problem domain to be described				How are issues of link and content validity	Abstract navigational model	

	in terms of hypermedia components: nodes, contents, links, anchors Allow the description of interactive behaviour in hypermedia systems				addressed? How is effective navigation and browsing addressed?	design User interface and run time behaviour design	
Multimedia	Organise and harmonise multimedia contents				How is the use of different media managed?		
Requirements and usability	Provide the designer with products to specify functional, non-functional and usability requirements			System requirement analysis		Usability	Support for usability
User analysis	Model the different types of users			Customer Analysis			
Security	Provide conceptual tools to formalise security policies						
Bottom up design	Allow a bottom up design		Design either bottom up or top down				
Evaluation	Make possible the evaluation of the system utility	A set of metrics with advice on quality, standards and test strategies					
Project Management		Guidelines for project management			How is the issue of cognitive management during development addressed? Support for enhancing developer productivity?		
Maintenance and testing					How is the issue of application maintenance addressed?	Support for implementation, testing and maintenance	

Table 2: A list of possible contents for a web development method

By synthesising these criteria, a more definitive list of possible contents of a web development method can be found. These are:

- **Process:**

  The use of a formal process to guide the development of the software application (Montero, Diaz and Aedo, 2002) this extends to coverage of the whole lifecycle (Fraternali, 1999), with process support (Retschitzegger and Schwinger, 2000), a full set of guidelines and rules and description of deliverables (Koch, 1999).

- **Requirements:**

  Provide the designer with products to specify functional, non-functional (Lee, Suh and Lee, 2004) and usability requirements.(Fraternali, 1999, Christodoulou, Styliaras and Papatheodourou, 1998, Montero, Diaz and Aedo, 2002)

- **Models:**

  Contain a full set of models (Koch, 1999) to describe the real world and transfer it to a physical system (Montero, Diaz and Aedo, 2002) and provide modelling abstractions (Fraternali, 1999). The models should clearly separate the three levels of presentation, navigation and content, (Woukeu, Carr, Wills and Hall, 2003, Christodoulou, Styliaras and Papatheodourou, 1998), they should be internally consistent (Koch, 1999) and provide for the flexible mapping between levels.(Retschitzegger and Schwinger, 2000) The behaviour (Montero, Diaz and Aedo, 2002) and structure of the three levels should be considered, as should the need for modelling formalisms. (Retschitzegger and Schwinger, 2000). Each product should include validation rules. (Montero, Diaz and Aedo, 2002)

- **Hypermedia**

  Allow the problem domain to be described in terms of hypermedia components (Montero, Diaz and Aedo, 2002) and ensuring link and content validity and allowing the organisation and harmonisation of multimedia contents. (Woukeu, Carr, Wills and Hall, 2003).

- **Reuse:**

  Provide advice for library management (Koch, 1999) and reuse, (Koch, 1999, Montero, Diaz and Aedo, 2002, Retschitzegger and Schwinger, 2000) using patterns (Retschitzegger and Schwinger, 2000) and components (Fraternali, 1999)

- **Other**

  There are a number of issues that are only described by one or two authors such as security behaviour, bottom up design, user modelling, project management and the issue of maintenance and quality.

However, it is doubtful that any one single method would fit all these requirements together because the list is quite extensive and is of necessity at a high level (Avison and Fitzgerald, 2006).

2.8. Summary

This chapter has explored the background to web development and its methods, from a number of different perspectives. It investigated the difference between web development and software engineering before looking at web site classification and the terminology surrounding web development. The background to Research Question 2 (whether web development methods are used?) was explored when the use and non-use

of web development methods was presented prior to the lack of use of development methods, in general. The contents of a web development method were researched and having found no research on the topic, the criteria used for web development method comparison were explored before being moulded into a single set of high level requirements for a web development method in response to Research Question 4 (What components, techniques and tools should constitute a web development method?)

Having detailed the background and outlined some of the major perspectives that should be considered in relation to web development and its methods, the next chapter presents a survey of web development methods, classified pragmatically, as well as the tabularising of the sources and the communities from whence they come.

Chapter 3

Survey of Web Development Methods

3.1 Introduction

If the premise that web development methods are not used is true then the first step in any exploration would be to find out what methods exist and how long they have been available. This Chapter further explores the literature in response to the following research questions taken from Section 1.2.

Research Question 2: Which web development methods are available? What is their scope of coverage? What guidance is available for their use?

Research Question 3: Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?

Research Question 4: What components, techniques and tools should constitute a web development method?

The research has found that it could be considered that the web development method field is maturing as there are 52 methods available and work has been taking place in this area since the early 1990's. However, this survey will show that the method sources are diverse, with contributions from different academic fields published in numerous different journals and conferences. Further, it finds that there is no accepted taxonomy, the language used to describe phases within the methods differs from method to method

and the notation is often proprietary. There seems to be no work on the suitability of different methods for different web development tasks.

3.2. Survey design

3.2.1. Survey goals

This survey was completed to provide a review of the field of web development methods that will allow the reader to acquire knowledge in the field and is in response to the first of the research aims. The goals of the survey were:

1. To allow an appraisal of the field as there has been no full review of the methods, their development, nor the use of methods in the development process,
2. To identify some of the gaps in current knowledge.
3. To investigate the scope of coverage of individual development methods
4. To investigate the instructions for the use of individual methods.

3.2.2. Survey method

A review should describe the information sources searched and the inclusion criteria used in selecting the cited papers (Weed, 1997). The sources for web development methods were believed to be very diverse in that the methods came from a number of different areas, such as the Human Computer Interaction (HCI) and Usability community, Hypermedia and Hypertext, and the Information Systems and Database communities.

The method used to obtain the data was to select a number of key words:

Web Development Methods
Hypermedia Development Methods
Web Engineering
Web Design

And variation of those words such as:

Web Database Development
Web Usability Methods

A number of different sources were visited including ISI Web of Knowledge, Scopus and Google Scholar. From a variety of conferences, workshops and journals a number of papers were found. With each paper that was collected the references at the end of the paper were checked with the list of discovered methods, to see if further methods could be included. The journal, conference or book that the paper came from was hand-checked to see if any further papers relevant to web development methods were included. If so this new web development was added to the list. The risk attached to this method is that a web development method that is not listed using the selected words may have been missed. Hand searching also introduces human error.

The criteria for inclusion of a method in the review were that the paper either had to be clear that it contained a web development method, or:

- it had to refer to some part of the Systems Development Life Cycle (SDLC) and contain a framework or technique that would enable some part (or all) of web development to take place
- or, it referred to a web modelling language or method

- or, it contained details of a web development tool that was part of a described development method.

As Lowe (2003) points out, the web development method research literature is extremely fragmented, with few attempts to draw the work together into a cohesive picture. Hence, the quality of the journal or conference was not considered as part of the inclusion criteria because, despite the possible risk of including methods of differing academic quality, it was felt that it was more important to provide as complete a picture as possible. For this reason in particular, this survey would not be considered a Systematic Review. Systematic reviews are considered an essential part of Evidence-Based Software Engineering (Kitchenham, Dyba and Jorgensen, 2004) and are used to provide a rigorous review of the literature, taking care to create a sound protocol beforehand and to only use work of the best quality; thus providing evidence based guidelines as support for software engineers (Kitchenham et al., 2009). Each phase is checked by another researcher and the protocol is tightly controlled. This is in contrast to what Kitchenham terms an adhoc literature selection (2009)which has been used here to give a broader coverage of findings but not necessarily the highest quality.

3.3. Method sources

The sources for the literature on web development methods are very diverse. There are 17 different conference and workshop tracks and 10 different journals from a variety of different fields as can be seen in Table 3.

Conferences	**Methods**
Web Engineering	WISD (Gnaho, 2000),WebComposition (Gaedke and Graf, 2000), RSD (Janssen and Steen, 2001), CFEP

	(Norton, 2001), FPM (Olsina, Lafuente and Rossi, 2000), Christodoulou (Christodoulou, Zafiris and Papatheodorou, 2000), HERA (Houben, Barna, Frasincar and Vdovjak, 2003), AWE (McDonald and Welland, 2001a), NDT (Escalona and Aragon, 2008)
World Wide Web Conference	WSDM (De Troyer and Leune, 1998), Takahashi (Takahashi and Liang, 1997)
International Conference on Database Technology	HDM-Lite (Fraternali and Paolini, 1998)
VLDB Conference	Araneus (Atzeni, Mecca and Merialdo, 1997b)
European Conference on Information Systems	SHDT (Bichler and Nusser, 1996b), ICDM (Standing, 2001)
Hypertext and Hypermedia	FPM (Olsina, 1997), RNA (Yoo and Bieber, 2000)
IEEE Conference on Multimedia Computing and Systems	JWeb (Bochicchio, Paiano and Paolini, 1999), AWARE (Bolchini and Paolini, 2004)
SIGMOD	Strudel (Fernandez et al., 1997)
Hawaii International Conference on System Science	EORM (Lange, 1994), SOHDM (Lee, Lee and Yoo, 1998)
Workshop on Software Specification and Design	Coda (Coda, Ghezzi, Vigna and Garzotto, 1998)
Human Computer Interaction (INTERACT)	Turbo-prototyping (Ghosh, 1999)
Software Engineering and Knowledge Engineering	WebML+ (Tongrungrojana and Lowe, 2003), SWIM (Griffiths, Hebbron, Lockyer and Oates, 2002)
Workshop on Web-Oriented Software Technology	UWE (Koch and Kraus, 2002), OOWS (Pastor, Fons and Pelechano, 2003)
E-Commerce and Web Technologies	RDF/WE (Kalpsing and Neumann, 2000)
DEXA	ARIADNE (Diaz, Aedo and Montero, 2001)
ACM Symposium on Applied Computing	WARP (Bochicchio and Fiore, 2004)
International Conference on Computer Documentation	UCDM (Fuccella, 1997)
Journals	
ACM Transactions on Information Systems	AutoWeb (Fraternali and Paolini, 2000), HDM (Garzotto, Paolini and Schwabe, 1993)
Journal Of Database Management	VHDM (Lee, Kim, Kim and Cho, 1999)
Communications of the ACM	RMM (Isakowitz, Stohr and Balasubramanian, 1995), OOHDM (Schwabe and Rossi, 1995), Nanard & Nanard (Nanard and Nanard, 1995), UML+ (Conallen, 1999b)
IEEE Multimedia	OO-H (Gomez, Cachero and Pastor, 2001), DPWA (Uden, 2002), JESSICA (Goeschka and Schranz, 2001)
ACM Transactions on Internet Technology	ARANEUS (Merialdo, Atzeni and Mecca, 2003)
Internet Research: Electronic Networking Applications and Policy	FECWAD (Lu and Yeung, 1998)
European Journal of Information Systems	HADT (Hatzopoulos, Vazirgiannis and Rizos, 1993)
Information and Software Technology	SOHDM (Lee, Suh and Lee, 2004)
Information Systems Journal	WISDM (Vigden, 2002)
IEEE Internet Computing	WebComposition (Gellerson and Gaedke, 1999)
Books	W3DT (Scharl, 2000), UML+ (Conallen, 1998), WebML (Ceri et al., 2003), HMT (Zoller, 2001), UCWD (McCracken and Wolfe, 2004)
PhD Thesis	UWE (Koch, 2000)

Table 3: Methods and their publications

3.4. Sources of method by community

There have been no comprehensive surveys of web development methods, although a number of academics from differing communities have carried out a comparison of some of the methods, see for example Koch (1999), Montero (2002), Gu (2002) and Escalona and Koch (2004). It should be noted that none of the comparisons give any criteria for the selection of the methods that were used for the comparison and that all comparisons were from the modelling community. The sources of the 52 methods that were found by the survey consisted of conferences, journals, books and a PhD thesis and are summarised in Table 3. Each method is categorised by the academic community it came from in an attempt to identify the motivation behind the methods (see Table 4). It is accepted that the categorisation may be flawed in some areas where there is no clarity of background in the individual papers, and some methods might fit equally well into more than one community.

Academic Community	Methods
Database	HDM Lite (Fraternali and Paolini, 1998), STRUDEL (Fernandez et al., 1997), WARP (Bochicchio and Fiore, 2004), Autoweb (Fraternali and Paolini, 2000), ARANEUS (Merialdo, Atzeni and Mecca, 2003), WebML (Ceri et al., 2003), Jessica (Goeschka and Schranz, 2001)
Hypertext and Hypermedia	SHDT (Bichler and Nusser, 1996b), W3DT (Scharl, 2000), VHDM (Lee, Kim, Kim and Cho, 1999), RMM (Isakowitz, Stohr and Balasubramanian, 1995), HDM (Garzotto, Paolini and Schwabe, 1993), OOHDM (Schwabe and Rossi, 1995), RNA (Yoo and Bieber, 2000), ADWIS (Takahashi and Liang, 1997), EORM (Lange, 1994), HADT (Hatzopoulos, Vazirgiannis and Rizos, 1993), ARIADNE (Diaz, Aedo and Montero, 2001), SOHDM (Lee, Lee and Yoo, 1998), HERA (Houben, Barna, Frasincar and Vdovjak, 2003), SchemaText (Kuhnke, Schneeberger and Turk, 2000), WSDM (De Troyer and Leune, 1998)
Object Oriented Analysis and Design	OO-H (Gomez, Cachero and Pastor, 2001), UML+ (Conallen, 1999b), WebComposition (Gellerson and Gaedke, 1999), OOWS (Pelechano, Fons, Manoli and Pastor, 2003)
Modelling/Notation	WISD (Gnaho, 2000), RSD (Janssen and Steen, 2001), UWE (Koch, 2000), FECWAD (Lu and Yeung, 1998), HMT (Zoller, 2001), NDT (Escalona and Aragon, 2008)
Multimedia	JWeb (Bochicchio, Paiano and Paolini, 1999), MATILDA (Lowe, Ginige, Sifer and Potter, 1996)
Information Systems/Software Engineering Methods	WISDM (Vigden, 2002), AWE (McDonald and Welland, 2001b). Turbo-prototyping (Ghosh, 1999), ICDM (Standing, 2001)
Human Computer Interaction	Macweb (Nanard and Nanard, 1995), UCDM (Fuccella, 1997), URMDP (Alaa and Fitzgerald, 2004), UCWD (McCracken and Wolfe, 2004), DPWA (Uden, 2002),
Other	CFEP (Norton, 2001),WOOM (Coda, Ghezzi, Vigna and Garzotto, 1998), RDF/WE (Kalpsing and Neumann, 2000), Christodoulou (Christodoulou, Zafiris and Papatheodorou, 2000), SWIM (Griffiths, Hebbron, Lockyer and Oates, 2002), WebML+ (Tongrungrojana and Lowe, 2003), AWARE(Bolchini and Paolini, 2004) ,QEM (Olsina, Lafuente and Rossi, 2000), FPM (Olsina, 1997),

Table 4 : Web development methods by academic community

3.5. Survey

The practitioner requires methods suitable to use for their specific developments, and needs to assess methods according to criteria such as scope or focus. Hence, the following section attempts to categorise methods in a manner that is likely to aid their selection or rejection for development, rather than simply categorise them by a theoretical dimension for purely academic purposes. Any attempt at method categorisation is recognized as a difficult problem (Glass, 2004) and comparison of methods is also a complex issue, which has been the subject of much research over a number of years within the Information Systems community, as summarised by Avison and Fitzgerald (2003a) and discussed in Section 2.6. This review examines methods by adopting a variant of the approach used by Lee et al. (1998) reflecting a practice-based approach which identifies the underlying modelling technique and the primary focus of the method.

3.5.1. Scope

One difficulty with disparate strands of research, from differing academic communities, is that different terminology is often used. Thus, it is not always easy to understand to which development phase an author may be referring. In addition, the majority of methods adopt different notations. These two factors contribute to making web development methods a particularly complex area to navigate. In addition, even the notion of a method has different connotations and meanings. Many methods such as ARANEUS (Mecca et al., 1998) and WebComposition (Gellerson and Gaedke, 1999) concentrate on the modelling aspects alone, and whilst they may mention other parts of

the lifecycle, they do not provide enough detail to allow implementation of those stages. The Modelling/Notation community believes that conceptual, logical and physical models are the route from requirements to implementation, and the greater the level of model detail, the closer the method is to implementation. Further, a web application development is characterised by three major design dimensions, which the Web Engineering community has recognised should be kept separate: structure, (describing the organisation of the information managed by the application.), navigation (concerning the ability to access the information) and presentation (allowing the content to be produced for the user) (Fraternali, 1999).

However, the majority of web development projects will be implemented using some variant of the Systems Development Life Cycle (SDLC). The SDLC is alternatively known as the waterfall model and has been in existence since the 1970's; and indicates a staged process (Avison and Fitzgerald, 2006). Although having a number of variants, the basic structure includes feasibility, system investigation (including requirements elicitation and recording) system analysis, design, implementation, review and maintenance (Avison and Fitzgerald, 2006). The methods have been categorised according to their scope of the Systems Development Life Cycle as follows:

3.5.1.1. Those methods that do not prescribe a particular methodological approach

For example, RSD (Janssen and Steen, 2001) offers an integrated framework on two dimensions – business and service oriented models – but does not specify a particular methodological approach nor associated techniques. WebComposition (Gellerson, Wicke and Gaedke, 1997) specifies a component model based on the design,

implementation and maintenance of a web application. Although it discusses a number of approaches it has an open process model focussing on reuse. WISDM (Vigden, 2002) covers requirements and the software model, but does not prescribe an approach, nor an implementation. In QEM (Olsina, Lafuente and Rossi, 2000) the user of the method specifies which part of the product lifecycle they intend to investigate.

3.5.1.2. Methods covering the full systems development life cycle.

This section outlines those methods that, by covering most of the SDLC, enable a web development project to be completed using a lifecycle approach. WOOM (Coda, Ghezzi, Vigna and Garzotto, 1998), WSDM (De Troyer, 2001), UWE (Koch, 2000) use an iterative approach, and WebML (Ceri et al., 2003), cover the full lifecycle from requirements to implementation, focussing on a modelling framework. OOHDM describes domain analysis, navigational design, abstract interface design and implementation. It has since evolved to encompass the full lifecycle (Guell, Schwabe and Vilain, 2000). JWeb starts at the definition of a Hypermedia Design Method schema (Garzotto, Paolini and Schwabe, 1993) and provides an environment to assist in the whole process, including implementation (Bochicchio, Paiano and Paolini, 1999). DPWA (Uden, 2002) focuses on requirements (using Applied Cognitive Task Analysis (Militello and Hutton, 1998)), usability requirements and Relational Navigation Analysis however, it does cover the whole lifecycle including maintenance. OOWS (Pelechano, Fons, Manoli and Pastor, 2003) covers requirements (use cases and scenarios) taken from the OO-Method (Pastor, Pelechano, Insfran and Gomez, 1998). It also covers conceptual modelling, navigational and presentation modelling, architecture design and implementation, adding patterns for presentation and services for architecture.

3.5.1.3. Methods discussing a lifecycle approach with missing aspects

A number of methods, such as HERA (Houben, Barna, Frasincar and Vdovjak, 2003), XWMF (Kalpsing and Neumann, 2000) and VHDM (Lee, Kim, Kim and Cho, 1999), discuss a full lifecycle approach but do not explicitly cover feasibility, requirements, implementation or maintenance. ADWIS (Lu and Yeung, 1998) does not explicitly cover them either, but adds Scenario Analysis. Similarly, SHDT/W3DT (Bichler and Nusser, 1996b), does mention doing requirements analysis and completing a requirements document but does not give any further details. SOHDM (Lee, Lee and Yoo, 1998) covers construction but not feasibility, requirements gathering or maintenance. HFPM (Olsina, 1997) describes the process of a hypermedia design method, functionally, using the concept of views. It outlines a full lifecycle approach but provides few details except in the main area of focus. Christodoulou *et. al.* (2000) do not cover feasibility nor requirements, but do cover development and maintenance. SchemaText (Kuhnke, Schneeberger and Turk, 2000) advocates analysis, design, implementation, test and maintenance but only provides an overview of document engineering techniques and navigation structures. SWIM (Griffiths, Hebbron, Lockyer and Oates, 2002) covers the whole lifecycle but not in great depth, concentrating on an Integrated Project Support Environment for teaching.

3.5.1.4. Methods not discussing a lifecycle approach and covering only part of the lifecycle.

Three methods focus on the requirements aspect of the SDLC. Firstly, RNA (Yoo and Bieber, 2000) describes a process for finding and modelling the links between information domains. Secondly, CFEP (Norton, 2001)describes a current practitioner

approach to web development and focuses on the requirements from a user and product viewpoint using Joint Application Development (Soltys and Crawford, Undated). Finally, AWARE (Bolchini and Paolini, 2004) uses goals to assist in the identification of requirements and helps with content, interaction, navigation and presentation. UCDM covers audience definition and content identification and validation, highlighting usability aspects (Fuccella, 1997).

Whilst UML+ (Conallen, 1998) provides extensions to the UML notation to take account of different web page requirements on both client and server, the method used is the Unified Process (Kruchten, 2000). OO-H (Gomez, Cachero and Pastor, 2001) gives details of requirements gathering using use case diagrams and a business class diagram. The navigation requirements are modelled from a class diagram and the top level Navigation Access Diagram is automatically generated. The scope of this method extends from problem space to solution space, but with an emphasis on design and implementation. WISD (Gnaho, 2000) concentrates on user modelling and navigational modelling. ARANEUS (Atzeni, Mecca and Merialdo, 1998) starts at the database conceptual schema design and continues to page generation. HMT (Zoller, 2001) covers design and does mention requirements. EORM describes an iterative design method (Lange, 1994). Nanard and Nanard (1995) cover the design process as does ADM (Diaz, Aedo and Montero, 2001). Finally, STRUDEL (Fernandez, Florescu, Levy and Suciu, 2000) is a web site implementation tool focussing on the management of different types of data from differing sources.

The scope of the methods illustrated by Table 5 emphasises the mixed scope of the methods researched for this survey and the fact that many methods do not cover the SDLC from end to end. Of those that do, the majority such as AWE and HERA only mention many of the SDLC phases and do not cover them. OOHDM started out with coverage of the design phases only and as papers were written additional phases were added to the original design phase. This evolved to coverage of the full lifecycle; however OOHDM is not supported as a method and would appear to have been a research vehicle. Further investigation of OOHDM is done as part of this work in Section 5.4

The key to Table 5:

m	Mentions the phase but gives no details
x	Covers the phase
e	Evolved into the full lifecycle

	Pre -Req	Requirements	Analysis		Design		Implementation	Test/Maintenance	Comment
				Conceptual	Navigation	Presentation			
ADM				x	x	x			
ADWIS		m	x	x	x	x	m	m	
ARANEUS				x	x	x	x		
ARIADNE				x	x	x	m	m	
AUTOWEB				m	m	m			An application generator
AWARE		x							
AWE		m	m	m	m	m	m	m	
CFEM		x							
CHRISTODOULOU							x	x	
CODA		m		x	x	x			Uses HDM
DPWA		x	x	x	x	x	x	x	
EORM				x	x	x	x		
FECWAD	x								
HADT									
HDM				x	x	x			
HDM-LITE				x	x	x			An evolution of HDM
HERA		m	x	x	x	x	m	m	
HMT		m		x	x	x			
ICDM									
HPFM		x	x	x	x	x	x	x	
JWEB				x	x	x	x		
NANARD &NANARD				x	x	x	x	x	
NDT		x		x	x	x	x		Model driven Web development
OO-H		x		x	x	x	x		
OOHDM	e	e	e	x	x	x	e	e	e - evolved into full life cycle
OOWS		x	x	x	x	x	x		
QEM									Quality evaluation method
RDF/XML FRMWRK				x	x	x	x		
RDF/WE									Describes a vocabulary
RMM		m	m	x	x	x	x	m	
RNA		x					x		
RSD									Process approach

	Pre -Req	Requirements	Analysis		Design		Implementation	Test/Maintenance	Comment
				Conceptual	Navigation	Presentation			
SCHEMATEXT			m	m	x	m	m	m	
SHDT/W3DT		m	x	x	x	x	m	m	
SOHDM		x	x	x	x	x	x		
STRUDEL									Site implementation tool
SWIM		m	m	m	m	m	m	m	
TURBO		x	x			x			
UCDM			x	x			m		
UCWD		x	x			x	x	x	
UML+									Uses Unified Process
UWE		x	x	x	x	x	x	x	
VHDM		m	x	x	x	x	m	m	
WARP		m	m	m	m	m	m		On –line design environment
WEBCOMPOSITION					x		x	x	
WEBML		x	x	x	x	x	x	x	
WEBML+	x								
WISD					x				
WISDM	x	x		x	x	x			
WOOM		x	x	x	x	x	x	x	
WSDM		m	x	x	x	x	m	m	
XWMF		m	x	x	x	x	m	m	

Table 5: Scope of web development methods

3.5.2. *Approach*

In this section, web development methods are classified according to the underlying modelling concept.

3.5.2.1. Methods taking an Entity Relation (ER) approach

A number of methods have been developed based on the Entity-Relationship (ER) model. Since ER models are widely used and understood, these methods have the advantage that a new data modelling language does not need to be learnt (Ceri et al., 2003), in addition to the associated other techniques. Although HDM is included, it differs from the ER approach, but is considered to be a mix between the ER model and the Dexter model (Garzotto, Paolini and Schwabe, 1993).

HDM (Garzotto, Paolini and Schwabe, 1993)
VHDM (Lee, Kim, Kim and Cho, 1999)
RMM (Isakowitz, Stohr and Balasubramanian, 1995)
HMT (Zoller, 2001)
ARANEUS (Atzeni, Mecca and Merialdo, 1998)
WebML (Ceri et al., 2003)
Hera (Houben, Barna, Frasincar and Vdovjak, 2003)
ADWIS (Lu and Yeung, 1998)
ADWIS (Lu and Yeung, 1998)
HMT (Zoller, 2001)
JESSICA (Goeschka and Schranz, 2001)
SHDT and W3DT (Bichler and Nusser, 1996b)

3.5.2.2. Methods taking an Object-Oriented approach

Many of the methods surveyed adopt an object-oriented approach. Although some methods are based directly on the Unified Modelling Language (UML), others use either a proprietary notation or bespoke extensions to the UML.

WISD (Gnaho, 2000)
WSDM (De Troyer and Leune, 1998, De Troyer, 2001)
HFPM (Olsina, 1997)
OO-H (Gomez, Cachero and Pastor, 2001)
Nanard and Nanard (Nanard and Nanard, 1995)
OOWS (Pelechano, Fons, Manoli and Pastor, 2003)

OOHDM (Schwabe and Rossi, 1995)
UML+ (Conallen, 1998)
WebComposition (Gellerson, Wicke and Gaedke, 1997)
WOOM (Coda, Ghezzi, Vigna and Garzotto, 1998)
XWMF (Kalpsing and Neumann, 2000)
Partly Schematext (Kuhnke, Schneeberger and Turk, 2000)

DPWA (Uden, 2002)
UWE* (Koch, 2000)
JESSICA (Goeschka and Schranz, 2001)
EORM (Lange, 1994)
SOHDM (Lee, Lee and Yoo, 1998)
NDT(Escalona and Aragon, 2008)

3.5.2.3. Methods based on neither the Entity Relationship nor Object-Oriented approaches

A number of methods do not advocate a specific approach to modelling. For some, this is due to their focus on early stages of the lifecycle, which does not necessitate a modelling approach. RNA (Yoo and Bieber, 2000) only considers the analysis of data relationships and hence is not based on a data model. FECWAD (Lu and Yeung, 1998) provides a framework for feasibility. AWARE is a framework for requirements identification (Bolchini and Paolini, 2004). QEM (Olsina, Lafuente and Rossi, 2000) looks at the quality of the artefact produced in different phases of the lifecycle.

eW3DT (Scharl, 2000) is based on a visualisation approach which complements either OO or ER approaches. Neither RSD (Janssen and Steen, 2001), CFEP (Norton, 2001) nor ADM (Diaz, Aedo and Montero, 2001) presribe any specific technique. SWIM (Griffiths, Hebbron, Lockyer and Oates, 2002) is based on a process of stages and deliverables but does not prescribe a modelling approach. SchemaText (Kuhnke, Schneeberger and Turk, 2000) is based on a hypertext approach, UCDW

(McCracken and Wolfe, 2004) is based on an HCI approach and UCDM is based on a usability approach, which deals with requirements and audience definition (Fuccella, 1997).

STRUDEL (Fernandez et al., 1997) provides a data management system, whilst UWE (Koch, 2000) uses the Unified Process as a basis, as does Connallen (1999b), and Christodoulou et al. (2000) provide a generic theoretical framework that can be applied to any development.

3.5.3. Focus

Many of the surveyed development methods are focussed on a specific area, as their authors investigate phenomena of interest to them. In this section, methods are classified according to focus.

3.5.3.1. Focus on pre-requirements

FECWAD (Lu and Yeung, 1998) is a framework for assessing the feasibility or merit of a project before development begins. WebML+ (Tongrungrojana and Lowe, 2003) is a modelling language for forming a bridge between web business and information modelling. WISDM (Vigden, 2002) could be partly considered as covering pre-requirements however, as it considers organisational analysis early in the development.

FECWAD (Lu and Yeung, 1998)	WebML+ (Tongrungrojana and Lowe, 2003)	WISDM (Vigden, 2002) (partly)
ICDM (Standing, 2001)		

3.5.3.2. Focus on user modelling/requirements

Some methods have requirements and user modelling as their stated focus, such as UCDM (Fuccella, 1997), WSDM (De Troyer and Leune, 1998), TURBO (Ghosh, 1999) and AWARE (Bolchini and Paolini, 2004). Other methods have been placed in this category as it was felt that they have some specific and useful focus in this area, such as DPWA (Uden, 2002) for its Applied Cognitive Task Analysis, NDT (Escalona and Aragon, 2008)for its use of patterns for requirements and SOHDM (Lee, Suh and Lee, 2004) for its customer analysis.

ADM (Diaz, Aedo and Montero, 2001)
AWARE (Bolchini and Paolini, 2004)
CFE (Norton, 2001)
DPWA (Uden, 2002)
NDT (Escalona and Aragon, 2008)
SOHDM (Lee, Lee and Yoo, 1998) – customer analysis
TURBO (Ghosh, 1999)
UCDM (Fuccella, 1997)
UCDW (McCracken and Wolfe, 2004)
WISD (Gnaho, 2000)
WSDM (De Troyer and Leune, 1998)

3.5.3.3. Focus on conceptual models/design models

A large number of methods focus on the modelling stages, with particular interest in the separation into the data layer, navigation layer and the presentation layer. Of particular concern to the authors has been the means of transformation of the models between the layers, and ensuring that modelling information is retained in such transformations.

ADM (Diaz, Aedo and Montero, 2001)
ADWIS (Takahashi and Liang, 1997)
Araneus (Atzeni, Mecca and Merialdo, 1998)
EORM (Lange, 1994)
HDM
HDM-Lite (Fraternali and Paolini, 1998)
HMT (Zoller, 2001)
JESSICA (Goeschka and Schranz, 2001)
Jweb (Bochicchio, Paiano and Paolini, 1999)
MacWeb (Nanard and Nanard, 1995)
OOHDM (Schwabe and Rossi, 1995)
OOWS (Pelechano, Fons, Manoli and Pastor, 2003)
SHDT/W3DT/eW3DT (Bichler and Nusser, 1996b)
UML+ (Conallen, 1998)
UWE (Koch, 2000)
WSDM (De Troyer and Leune, 1998)

3.5.3.4. Focus on aiming to automate or be part automated

Some methods specifically aim to automate the process, or parts of it, such as STRUDEL (Fernandez, Florescu, Levy and Suciu, 2000), which provides a tool that can manage disparate data sources. Others, such as WSDM (De Troyer, 2001), aim to turn the user (audience) requirements into a high level formal description in the conceptual design, which can later be used to automatically or semi-automatically generate effective web sites. It concentrates on the information design and the navigation structure, using a proprietary notation. In OO-H (Gomez, Cachero and Pastor, 2001), from a UML compliant class diagram, personalised navigation access diagrams are produced for each user type. The default interface is then generated, and improved using a pattern catalogue. JWeb (Bochicchio, Paiano and Paolini, 1999) is a development environment which assists, from the development of the definition of the conceptual schema using HDM constructs, to implementation of an application, whereas OOWS (Pelechano, Fons, Manoli and Pastor, 2003) is devised to use the OO-Method and is aiming for a fully automated environment. These methods are summarised in Table 6.

Method	Tool name	Notes
ADM (Diaz, Aedo and Montero, 2001)	Ariadne Tool	
ADWIS (Takahashi and Liang, 1997)	Pilot Boat & Web Architect	Not available
Araneus (Atzeni, Mecca and Merialdo, 1997a)	ULIXES, PENELOPE	
Christodoulou (Christodoulou, Zafiris and Papatheodorou, 2000)		Apply tool support using RDF/XML.
EORM (Lange, 1994)		
HDM Lite (Fraternali and Paolini, 1998)	Autoweb	
HERA (Houben, Barna, Frasincar and Vdovjak, 2003)		
HMT (Zoller, 2001)	WebCon	
JESSICA (Goeschka and Schranz, 2001)		
JWeb (Bochicchio, Paiano and Paolini, 1999)		
MacWeb (Nanard and Nanard, 1995)		
NDT (Escalona and Aragon, 2008)	NDT-Suite	
OOHDM (Schwabe and Rossi, 1998)	OOHDM-Web	
QEM (Olsina, Lafuente and Rossi, 2000)	WebQEM_Tool	
Schematext (Kuhnke, Schneeberger and Turk, 2000)		
SHDT/W3DT (Bichler and Nusser, 1996b)	Web Designer	no longer available
STRUDEL (Fernandez et al., 1997)		
SWIM (Griffiths, Hebbron, Lockyer and Oates, 2002)		
UWE (Koch and Kraus, 2002)	ArgoUML/MagicUWE	
WebComposition (Gellerson, Wicke and Gaedke, 1997)		
WebML (Ceri et al., 2003)	WebRatio	
WOOM (Coda, Ghezzi, Vigna and Garzotto, 1998)		
XWMF (Kalpsing and Neumann, 2000)	GRAMTOR, RDF-Handle, WebObjectBrowser	

Table 6: Methods aiming to automate the web design process

3.6. Difficulty of use

The majority of the web development methods in the survey, as shown in Table 5, do not give enough detail to enable them to be used in their entirety. Even when

methods are covered in their entirety, they are rarely covered in enough depth to allow their use for development.

In addition, the explanations of how to use a number of methods may be difficult for the non-computer scientist to understand. For example, in Araneus (Atzeni, Mecca and Merialdo, 1997b) the developer needs an understanding of both hypertext and database theory; and similarly with Strudel (Fernandez, Florescu, Levy and Suciu, 2000), which adds differing data types and sources to the method, and has not been designed with the practitioner in mind. Other methods that also make assumptions on user expertise and understanding are OO-H (Gomez, Cachero and Pastor, 2001) and Jessica (Goeschka and Schranz, 2001). DPWA (Uden, 2002) covers Applied Cognitive Task Analysis (Militello and Hutton, 1998), which was found to be cumbersome by Uden's Masters students. WISD (Gnaho, 2000) is clearly to be used by developers with knowledge and understanding of fuzzy logic, which is important in defining the user model. These methods are unlikely to be taken up by the majority of web developers. The explanations of the techniques are overly technical and complex and with web development time scales being shorter than traditional development cycles with aggressive release demands (Overmyer, 2000) there is little time to be spent on learning complex techniques.

A number of the methods surveyed have explained some of their phases or techniques in enough detail to possibly make them teachable or learnable. For example: FECWAD (Lu and Yeung, 1998), WSDM (De Troyer and Leune, 1998),

WebML (Ceri et al., 2003), OOHDM (Schwabe and Rossi, 1995) SHDT/W3DT (Bichler and Nusser, 1996b), eW3DT (Scharl, 2000) and RNA (Yoo and Bieber, 2000). UCDM (Fuccella, 1997) is described as having 'quick and dirty' approaches to getting user feedback for usability processes and TURBO (Ghosh, 1999) is a practitioner based method.

If, as has been suggested by Barry and Lang (2001b) and Lang and Fitzgerald (2005), web development methods are not used because they are too difficult, then the notion 'difficult to use' needs to be explored. There has been no research investigating the ease of use of web development methods, nor what the difficulty is. It is possible that it is purely a mis-match between the method author's perception of the knowledge and background of a method user and the relative experience or otherwise of the average developer.

Further, it is difficult to comprehend the degree of understanding of a 'web developer' in general and to create a method that is sufficient for their purpose. As far back as 1994, Vessey and Glass (1994) were exploring the use of method in software development and believed that software development itself was complex and was increasing in complexity. This complexity means that a successful systems developer has to master large amounts of knowledge both in the area of the problem domain and the software construction area. Thus it is perhaps arguable whether a simple method could aid something as complex as a successful software development. There has been little exploration of this area although Cockburn has

made some attempts at fitting his methods to different application domain problems (Cockburn, 2000).

Vessey and Glass (1994) put forward the concept of technique - based approaches, where current methods or approaches are 'dis-aggregated' and the resulting techniques are identified as to where they would be most useful or least useful. The approach from a research perspective has been less than successful, with Glass commenting ten years later that there has been no research into method taxonomies and problem domain taxonomies nor anything which maps the method to the problem domain (Glass, 2004). However, web developers have identified a pragmatic approach to the complexity and technique problem. Much as a doctor will dispense medicine from a bag of possible medicines a developer will utilise any technique that they believe useful in a specific situation. This has been validated to some extent by this research, where students, despite abandoning suggested methods, have carried out successful web developments by either creating their own custom made method or utilising a selection of techniques around the Systems Development Life Cycle. The students were final year BSc Business Information Technology students who had completed a one year industrial placement and were completing their final year project. The study is fully detailed in Section 5.2.

Research Question 3 asks whether the uptake of web development methods is affected by the *difficulty of using* them in guiding web developers through a web development? To further explore '*difficulty of use*' the literature on complexity and

cognition will be examined to see what may be considered useful to the development process for novice developers.

3.6.1. The difficulty of the process and notation from a complexity perspective

The IEEE has a definition of complexity which is useful as it directly refers to the notion of being difficult to use:

Complexity is "the degree to which a system or component has a design or implementation that is difficult to understand and verify" (IEEE, 1990).

In contrast, Richardson et al. (2001) define a complex system as one that comprises or consists of a large number of entities that display a high level of interaction. It will consist of interconnected parts that are interwoven, intricate and complicated. In addition, a method is considered as an approach to perform a systems development, if it is based on a specific way of thinking, consisting of direction and rules, structured in a systematic way in development activities with corresponding development products (Rossi and Brinkkemper, 1996). In tandem during development, various structures, processes, links and transformations have to be modelled and this requires a multitude of modelling techniques (Siau and Rossi, 1998). Thus, it can be considered that the development process and the method that supports it could have a degree of complexity.

There have also been a number of approaches to studying complexity, however those relating to Natural Computing and Organisational Science have been considered to be outside the scope of this work. Siau and Cao (2003) provide an overview of some of the complexity measures in Software Engineering believing that a complexity measure rests on its explanatory power and applicability. They qualify this by explaining that the measure needs to be able to explain the interrelationships among complexity, quality and other programming and design parameters. In addition, the measure should be applied to improve the quality of work during the design, coding and testing stages. They classify complexity measures into four categories. Firstly, Lines of Code (LOC) refer to the number of instruction statements in a module of code. The LOC represents the program's size and complexity. The work in this area refers mainly to defect detection and the idea that the more LOC, the larger and more complex the program and the more defects as the size and complexity increase beyond the programmers control.

The second complexity measure category refers to Halstead's Software Science that developed a quantitative measure of complexity which were a set of equations based as a function on the number of variables and constants along with operators and keywords present in the code (Halstead, 1987). These are the most well known and most comprehensively studied of all complexity measures (Cardoso, Mendling, Neumann and Riejers, 2006).

The third category is McCabe's Cyclomatic Complexity metrics (McCabe, 1976) which are designed to measure a programs testability and understandability.

Finally, all the measures so far refer specifically to program modules and assume that each module is a separate entity. Following considerable discussion about the usefulness of the program metrics relating solely to individual modules, Structure Metrics were suggested. Rossi and Brinkemper introduced a structure metric for measuring complexity based on seventeen metrics for systems development methods and techniques (1996). These metrics are based on the number of object types, relationships and properties and their connections.

There appears to be no work on the complexity of the constructs in web development methods in particular, although some work has been done on assessing elements of the Unified Modelling Language (UML) (Siau and Cao, 2003). The measures are considered important because UML is larger and more complex than other techniques and, therefore, more difficult to learn and use (Erickson and Siau, 2007). The reason that the validity of these complexity techniques has been discredited, to some extent, is the fact that it reduces complexity to a single figure. The reality of a complexity measure being 26.4 (the complexity measure of a class diagram) relates little meaning in itself and even compared to another technique with a complexity of 11.18 (the complexity measure of an activity diagram) it is difficult to understand the complexity issues (Erickson and Siau, 2007). In practice, complexity relates to numerous variables including the problem domain, the

techniques being used, the timescale etc therefore use of a single figure can be misleading. In addition, none of these measures take into account the 'soft' element of complexity such as the human element and Erickson and Siau's work in particular is done with a set of experts, thus making it difficult to consider a set of novices.

3.6.2. The difficulty of the notation and process from a cognitive perspective

Siau and Wang (2006) explore a number of modelling techniques from the Unified Modelling Language using a cognitive evaluation process. However, similar to their program complexity work, the techniques are examined in isolation and are not looking at the structure of the techniques which make up a development method. In terms of the accessibility of methods, approaches such as Cognitive Dimensions (Green, 1989a) appear to offer more potential, and have, for example, been applied to the assessment of elements of the Unified Modelling Language (Cox, 2000) and the user interface of Model Driven Architecture tools (Kanyaru et al., 2008).

Green (1989a) believes that the relationship between notation and the environment that supports it is vital, and the Cognitive Dimensions Framework has been formulated as a discussion tool to take advantage of this. It is particularly useful to users who are not Human Computer Interaction (HCI) specialists (Green and Petre, 1996). Blackwell describes a notational system as consisting of marks made on some medium and using the example of a computer screen there may be multiple

notations such as the main notation and the notation in surrounding menu dialogues (Blackwell and Green, 2003). It is a framework that allows developers to think about the nature of the notational system, the way that people interact with it and provides a structure within which to understand the vocabulary (Blackwell et al., 2001). The Cognitive Dimensions Framework is described in (Green and Petre, 1996, Green, 1989b, Blackwell et al., 2001, Blackwell, Whitley, Good and Petre, 2001) and may be summarised as:

- Viscosity: resistance to change
  A viscous system needs many user actions to accomplish one goal. How many user actions are necessary to make one change?
- Visibility: ability to view components easily
  Systems that have low visibility bury information in encapsulations. Is every part of the notation visible or is it at least possible to juxtapose two parts?
- Premature commitment: constraints on the order of doing things
  Does the user have to make decisions before having the information they need?
- Hidden dependencies: important links between entities are not visible
  Is every dependency clearly indicated in both directions? Is the indication perceptual or symbolic?
- Role-expressiveness: the purpose of an entity is readily inferred
  Can the user see how each of the components relates to the whole?
- Error proneness: the notation invites mistakes and the system gives little precaution
- Abstraction gradient: Types and availability of abstraction mechanisms
- Secondary notation: extra information in means other than formal syntax
  Can users use layout, colour and other cues to convey extra meaning, beyond the semantics of the notation or language?
- Closeness of mapping: closeness of representation to domain.
- Consistency: similar semantics are expressed in similar syntactic forms

When some of the notation or language has been learned, can the rest be inferred?

- Diffuseness: verbosity of language
  How many symbols or graphic entities are required to express a meaning
- Hard mental operations: high demand on cognitive resources
  Are there places where the user needs pencil and paper to track what is happening?
- Provisionality: degree of commitment to actions or marks
- Progressive evaluation: work to date can be checked at any time
  Can a partially completed operation be executed to obtain feedback?

The cognitive dimensions framework will allow discussion of the effects on cognition of different design decisions, and as such these will always involve trade-offs and these can be illustrated in Figure 2. Blackwell (2001) gives the example that changing the structure of a notation to reduce viscosity is likely to affect other dimensions such as introducing hidden dependencies or increasing the abstraction.

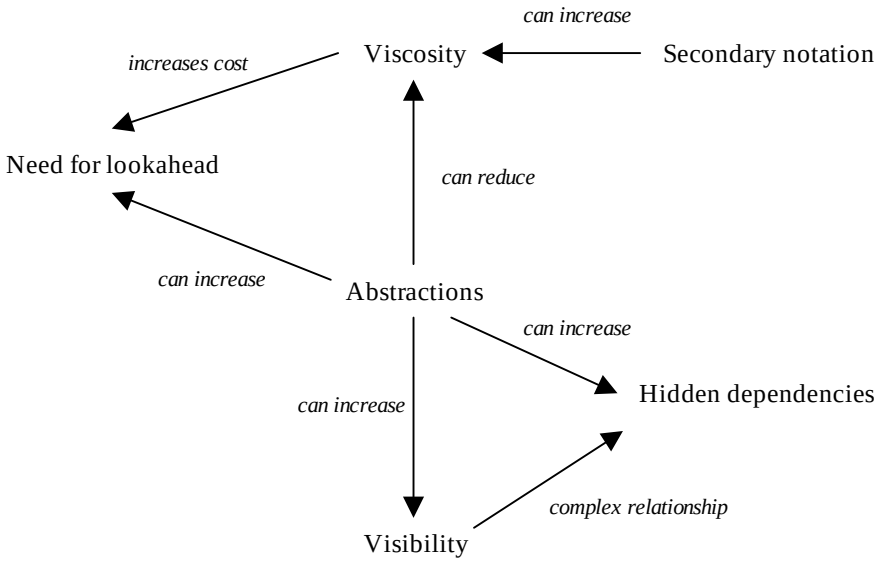


Figure 2: Cognitive dimensions trade-off (Blackwell and Green, 2003)

The Cognitive Dimensions Framework will be used as part of this research to see if it enables discussion to take place about the use of different methods by web

developers and thus assist in the data collection and analysis process. (See Section 4.5.2 and Section 5.4). It will also address Research Question 5 which asks whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?

3.7. Summary

This chapter reviewed the literature to allow the reader to become acquainted with the field of web development methods. In response to Research Question 2, the literature was categorised to show the variety of sources of web development methods and the different communities from where they originate. A survey was conducted which categorised web development methods in pragmatic terms to allow a practitioner to select a method according to specific criteria such as scope (Jeary, Phalp and Vincent, 2007b, Jeary, Phalp and Vincent, 2009). In response to Research Question 3 a discussion was presented which explores the notion of difficulty of use and investigated complexity metrics before highlighting Cognitive Dimensions as a possible useful approach to allow reflection and discussion on design decisions.

Having thus completed a review of the literature in the area surrounding web development methods, Chapter 4 will outline the research approach for this work using Creswell's (2003) conceptual framework before detailing the specific techniques that will be used for data collection and analysis.

Chapter 4

Research Approach

4.1. Introduction

The research process begins when a researcher with a certain 'world view' that is formed by their class, gender and background approaches research with a specific theory that details a set of questions that are then examined in specific ways (Denzin and Lincoln, 2005b).

This chapter examines approaches to research starting with the research aims and the shaping of the 'world view' of the researcher and the different theories that help to produce the research design. It then details the identification and selection of a research design that highlights the nature of this research before outlining the research strategy and the techniques that will be used for data analysis and collection.

4.2. Research Aims

The research from the literature surrounding web development methods in Section 2.5 showed that web development methods are generally not used and this has answered the first research aims. There are a number of reasons given for their non-use but Lang and Fitzgerald (2005) believe that they are too difficult to use. There is

no published work on the difficulty of use of web development methods and techniques and no work appears to have been done in the area. The rest of this work therefore needs to focus on answering the specific research questions 2 to 5.

The aims of this research are to investigate and identify:

1. Whether web development methods are used?
2. Which web development methods are available? What is their scope of coverage? What guidance is available for their use?
3. Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?
4. What components, techniques and tools should constitute a web development method?
5. Whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?
6. Whether it is possible to take the findings of this work and use them to inform the design of a web development method that is suitable for novice web developers?

4.3. The notion of 'world view' or Elements of Inquiry

The area of knowledge, its acquisition and the way an individual views the world, (the researcher notion of 'world view') along with the accompanying philosophical background can be complex because of the terminology which is used and the

different uses to which the same phrases are put by different researchers (Crotty, 2004). For example, a researcher coming to a project will make certain assumptions, which Creswell (2003) terms knowledge claims about what they will learn from their work and how they will learn it. A knowledge claim is a philosophical stance which informs the research according to Creswell (2003) but Crotty (2004) uses the term theoretical perspective to define the assumption about reality which we bring to our work. Denzin and Lincoln (2005a) describe the researchers methodological, ontological, and epistemological beliefs as paradigms. Guba and Lincoln (1994) have the same view and discuss paradigm as having three strands

- What is the form and nature of reality (the ontological question)?
- What is the relationship between the researcher and what can be known (the epistemological question)?
- How does the researcher find out whatever they believe can be known (the methodological question)?

The answers to these questions form the beliefs that shape the world as the researcher both sees it and acts in it (Denzin and Lincoln, 2005b). For the purpose of this thesis, however, those elements that make up the researchers 'world view' as described by Creswells research design framework have been adopted and are shown in Figure 3.

The framework describes how alternative knowledge claims, strategies of inquiry and methods are conceptualised by the researcher who then reflects this in their approach to completing a piece of research. Finally these approaches are translated

into practice by such things as the research questions, data collection and analysis and lastly validation (Creswell, 2003).

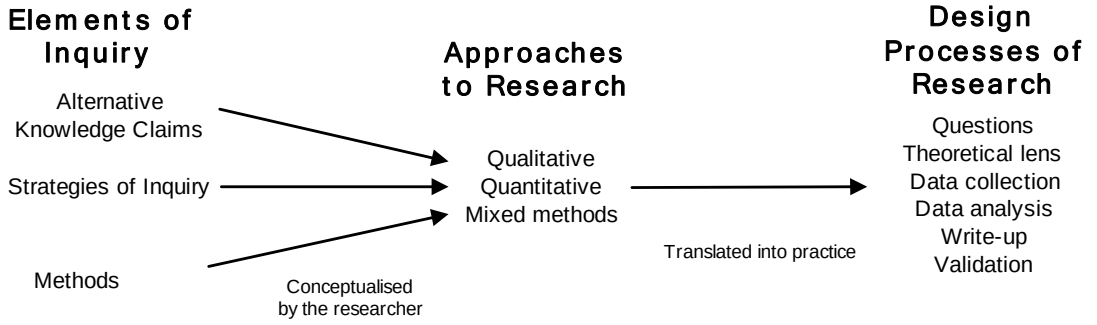


Figure 3: The conceptual research framework (Creswell, 2003)

4.3.1. Alternative knowledge claims

Creswell (2003) describes alternative knowledge claims, where each alternative view differs by the way the individual lives and reacts with the world around them. This section outlines three different approaches: positivism and post-positivism; interpretivism and constructivism; and finally, pragmatism.

4.3.1.1. Positivism and Post-positivism

A traditional theory of knowledge used in Software Engineering is that of positivism or post positivism; it has also been called empirical science and quantitative research (Creswell, 2003). The positivist approach is based on the assumption of objective reality; all knowledge can be gained by creating testable hypotheses and proving them by using measurable empirical and statistical analysis (Straub, Gefen and Boudreau, 2004). Facts and values are distinct and scientific knowledge consists only of facts (Walsham, 2002). In a positivist approach validity and reliability are at the core of all work (Silverman, 2006), and theory can

be tested and thus all of science should be value free and objective (Flick, 2006). It is most often used where results are clearly measurable and where variables can be controlled. More recently, many of the assumptions of the positivists have been discredited as it has become accepted that all measurement is based on theory and therefore an objective reality is not possible. This has lead to a post positive, or after positive position, which challenges the absolute truth of knowledge and recognising that when studying humans we cannot necessarily be 'positive'. Creswell (2003) describes the use of 'scientific method' or doing 'science' research as a common description of post positivism. The causes need to be examined to determine the effects or the outcomes and we therefore need to reduce ideas to small sets of ideas to test. This is the principle of reductionism (Creswell, 2003). Measurements for validity and testing will be reduced to the numeric and the null hypothesis is a central tenet (Straub, Gefen and Boudreau, 2004). Denzin and Lincoln (2005b) describe the criteria for evaluation as internal and external validity, reliability and objectivity; and the form of narration is a scientific report.

4.3.1.2. Intrepretivism or constructivism

In contrast, the constructionist or interpretive view is that meaning is constructed by individuals as they live and work in the world. As the individual tries to make sense of the world they construct multiple meanings and the researcher aims to look for the complexity of views (Creswell, 2003). Crotty (2004) believes that we interact with the world and make sense of it based on our historical and social perspective, that our culture influences the way we understand. By seeking to understand, the researcher will visit the context and gather the information

personally. From this, they will make an interpretation which is based on the researchers own cultural and experiential background. The goal is to rely on the individuals' view of the situation being studied, to ask questions that are 'broad and general' which will allow the participant to make sense of the situation (Creswell, 2003). Denzin and Lincoln (2005b) describe the criteria for evaluation as trustworthiness, credibility, transferability and confirmability; and the form of narration as interpretive case studies or ethnographic fiction.

4.3.1.3. Pragmatism

A third knowledge claim is that of pragmatism for which there are many forms, because pragmatism is not committed to any one philosophy (Creswell, 2003). Research in the pragmatic paradigm looks to the consequences of actions, knowledge claims arise out of actions, situations and consequences and not what has gone before (Cherryholmes, 1992). Cherryholmes (1992) also believes that pragmatic choices about what to research and how to go about it are conditioned by the sense of 'where we want to go'. This equates to an effects or outcome oriented position in which the researcher thinks about what will happen if they do X, practical experiences such as the researcher thinking about what will happen in their experience if they do X, and experiments by trying out X and observing the consequences (Johnson and Onwuegbuzie, 2004). Pragmatism realises that both the natural world and the social world have their place and that knowledge of the world is both constructed and based on the reality we experience. A pragmatist will prefer action to philosophising and endorses practical theory (Johnson and Onwuegbuzie, 2004). Pragmatists consider the research questions to be more

important than either the method they use or the 'world view' which underlies it (Tashakkori and Teddle, 1998). Pragmatists are concerned with real world practice and are not committed to any one system of philosophy; they are free to choose the methods, techniques and procedures of research that best meet their needs (Creswell, 2003).

4.3.2. Approaches to Research

The three different approaches to research that reflect the alternative knowledge claims and the strategies of inquiry are quantitative, qualitative and mixed method approaches.

4.3.2.1. Quantitative approach

A Quantitative approach shows that the world has an objective reality. This reality is reflected scientifically (often in terms of numbers) by carefully selecting hypotheses and testing them, looking to generate data and measurements that may be analysed typically in statistical form; and be repeatable. The data needs to be generalisable to other situations. To ensure this, sampling strategies need to be selected with care and the researcher has to ensure that they do not contaminate the data. The researcher has to be objective (Glesne and Peshkin, 1992) and is seldom able to reflect the subject's perspective because they use remote inferential methods and materials (Denzin and Lincoln, 2005b).

A deductive approach, is typically used in quantitative research and places the theory at the beginning or early in the process and uses it deductively in order to

test it (Creswell, 2003). Qualitative research can also make use of theory early in the research process when for example it provides an explanation for behaviour and attitudes or when researchers are using a theoretical lens to guide their studies (Creswell, 2003).

Straub, Gefen et al. (2004) believe that deductive theory building involves testing for internal validity and ensuring that the basics of the theory are testable. The theory will need to show that it advances knowledge from existing theory and has greater empirical grounding. It should allow empirical testing which aims to falsify the predictions of the theory. A researcher in the deductive process thus uses their research to test a theory by creating a hypothesis or propositions and defining variables to examine, then test, as shown in Figure 4.

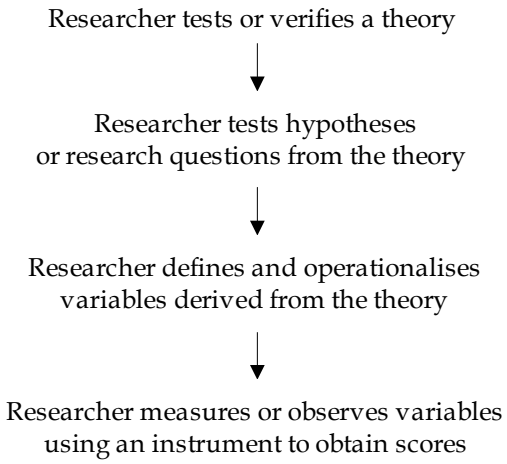


Figure 4: The deductive approach to theory generation (Creswell, 2003)

4.3.2.2. Qualitative approach

The qualitative approach emphasises that reality is socially constructed and involves an interpretive, naturalistic approach to the world. The researcher studies subjects and phenomena in their natural settings attempting to make sense of the meanings that the subjects give them (Denzin and Lincoln, 2005b). The researcher can never be separated from the process and recognising that their interpretation is based on personal experience they place themselves within the work (Creswell, 2003).

This is the alternative approach to deductive analysis and occurs when the theory or other broad explanation becomes the end point of the study. It is an inductive process of building from the data via themes to a generalised model or theory as illustrated in Figure 5.

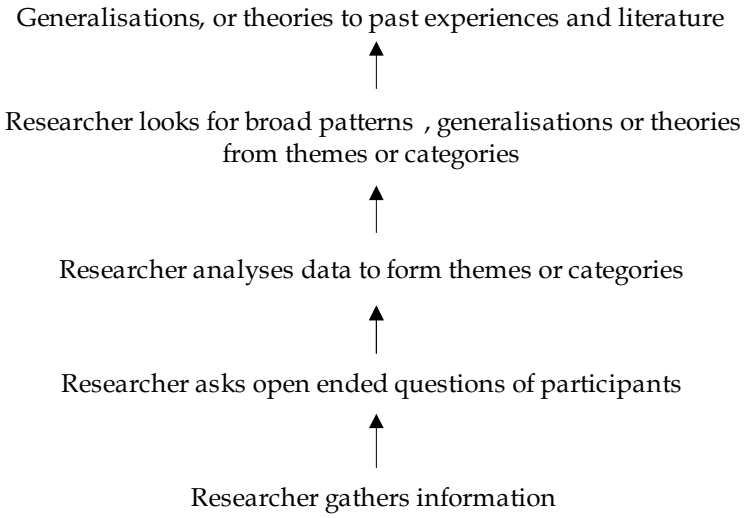


Figure 5: The inductive approach to theory generation (Creswell, 2003)

4.3.2.3. Mixed Method approach

The mixed method approach combines the qualitative and quantitative approaches of research into a single study. They are described by Tashakkori and Teddle (1998) as being used in sequential, parallel, equivalent status and multi- level approaches, all based on triangulation of results.

4.3.3. Strategies of Inquiry

At this point the researcher, examines the research questions and the purpose of the study to ascertain the strategy that is best used to obtain it (Denzin and Lincoln, 2005b). Wolcott (2001) gives nineteen different research strategies for qualitative research and Creswell (2003) creates five broad categories, whilst Galliers and Land (2002) identify seven for quantitative research. The major strategies from each of the paradigms or knowledge claims have been placed in Table 7:

Quantitative/ Positivist	**Qualitative/ Interpretive**	**Mixed Method/ Pragmatic**
Experiment	Grounded theory	Sequential order
Case study	Case study	Concurrent order
Survey	Ethnographic approaches	Theoretical lens

Table 7: A selection of major research strategies adapted from Galliers and Land (2002)

Case study research is an accepted research strategy in the social science, humanities and anthropology areas (Denzin and Lincoln, 2005b) and in Information Systems (Cavaye, 1996) where it is useful in studying the use of Information Technology in its organisational context (Darke, Shanks and Broadbent, 1998). From

Table 7 it can be seen that at the strategic level a case study can be used for both qualitative and quantitative approaches. A case study has a range of dimensions based on the number of cases and the amount of information collected (Gomm, Hammersley and Foster, 2007b). If a comparison is made between case study research and an experimental strategic approach or a survey approach using a positivist stance then the number of cases studied and the amount of information collected about each case differs on those dimensions. A survey will collect relatively little data about individual cases in contrast to a single case where the depth of data and information obtained across a number of different methods of collection is paramount. This dimension of case study research can also apply to a relatively small number of cases studied, often just one, in considerable depth and using a qualitative, interpretive approach (Gomm, Hammersley and Foster, 2007b). Another difference that a case study strategy has over, for example, an experimental strategy is the amount of control of the various variables that are likely to affect the different cases. Experiments aim to have full control of the variables whereas case study research is much more useful to explore cases in their real world setting when there is very little control of the variables (Yin, 2003). The use of case study research should help people toward further understanding of a subject (Stake, 2007).

Further strategies using a qualitative approach include grounded theory which is a method of qualitative data analysis where the theory is allowed to emerge from the data without the use of any pre-existing framework. It is interpretive and the data is

gathered through theoretical sampling guided by the concepts that are resultant from the analysis (Strauss and Corbin, 1998). The grounded theory strategy allows the researcher to embark on a study with no theoretical position and allows the researcher to actively construct the data into multiple layers of meaning. The ethnographic approaches as highlighted in Table 7 refer to cultural studies which are interested in discovering the relationship between culture and behaviour. They are mostly referring to observational studies of 'sites' as opposed to individuals and are thus looking for as many informants as possible (Gray, 2009).

The pragmatic, mixed method approach allows for the research to have a mix of the two paradigms, both qualitative and quantitative and is increasingly being accepted as a third major paradigm (Johnson and Onwuegbuzie, 2004). The reference to sequential and concurrent order in Table 7 refers to the mix of paradigms that is used and the order that they are used in. Thus sequential allows for a qualitative study to be followed by a quantitative study, and the results from the first used in the second; or vice versa. Whilst concurrent order refers to a qualitative and a quantitative study that are independent and are run in parallel. A theoretical lens or perspective approach uses theories such as gender, lifestyle perspectives or such things as cultural, racial or ethnic perspectives and produces emancipatory or anti-discriminatory theories.

4.3.4. Methods

Method is defined by Creswell as being the specific methods of data collection and analysis. Wolcott (2001) believes that method is more than fieldwork techniques,

and that readers of any research should be provided with sufficient detail about how the data was obtained, and more importantly how it was analysed. Methods of data collection enables the collection of data using such things as observations, text and documents, interviews and audio and video recordings (Silverman, 2006).

4.4. Personal epistemology

Qualitative research brings to prominence the role of the researcher. The researcher's values, and previous knowledge and experiences have significant effect on the researcher's interpretation of the data and the environment (Trauth, 2001). The researcher despite being an outside observer interprets the data and as such introduces their own subjectivity (Walsham, 2002). In addition, the Information Systems discipline has had much discussion on the emergent socio-technical field, at the intersection of technology and the social environment into which it is placed (see for example (Lee, 2001) for a summary discussion).

Personal examination by the researcher for this study finds that their 'world view' is that of a pragmatist and this is reflected in the formulation of the research aims. The pragmatist will look to see how it is possible to get the best results for the study (Creswell, 2003). Thus following the pragmatic philosophy outlined by Tashakkori and Teddle (1998), where the research questions are addressed by any methodological tool available 'that works'. An additional perspective to philosophy that also needs to be considered in the terms of systems development method is the personal philosophical viewpoint in terms of systems development. The

importance of the interpretive perspective and the link between this and the socio-technical viewpoint; understanding how the people and procedures are affected by the technology and vice-versa are part of this researchers personal perspective and are also reflected in this study and the design of it. The positivistic, technical and scientific viewpoint is not evident.

4.5. Research Strategy for this thesis

4.5.1. Introduction

The previous sections have outlined the philosophical background to the study and the role of the researcher. This section looks at the research questions and the selected strategy for investigating those questions. The questions are repeated here for convenience:

1. Whether web development methods are used?
2. Which web development methods are available? What is their scope of coverage? What guidance is available for their use?
3. Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?
4. What components, techniques and tools should constitute a web development method?
5. Whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?

6. Whether it is possible to take the findings of this work and use them to inform the design of a web development method that is suitable for novice web developers?

4.5.2. Data Collection

This section will outline the data collection methods and techniques that will be used for answering the questions that are outlined in the previous section.

4.5.2.1. Case study approach

Case study research is a relatively recent form of formal research method and was originally considered as an exploratory stage of some other research method (Yin, 2009). Much of the Software Engineering literature is based on the use of the case study in a positivistic quantitative paradigm and thus is looking at the comparison of case studies with scientific experiments and surveys (Kitchenham, Pickard and Pfleeger, 1995, Kitchenham, 1996). The importance of social issues (in terms of people and procedures) in relation to computer-based systems and the resultant research focussing on interpretation of the situation and the search for meaning has given rise to the use of in-depth case studies in information systems research (Walsham, 2002).

Because this research is focussed on the use of a 'procedure' by 'people' in relation to computer based systems it is therefore suitable for case study research. In addition, it is investigating a contemporary set of events in their normal setting and thus would appear suitable using Yin's (2009) criteria. The study is designed after

Yin (2009) who believes that case studies are one of the most challenging research methods and shows that case study research designs have not been codified. This has resulted in case study research being criticised for a lack of rigour. However, he identifies a number of different styles of case study, based on single cases and multiple cases. A single case study is suitable in circumstances when there is a critical test of existing theory, a rare or unique circumstance, a representative or typical case; or where the case serves a revelatory or longitudinal purpose (Yin, 2009). This study is revelatory in nature, where the research is initially trying to identify the ease of use of methods and latterly the components that should be included and therefore fits with Yin's designs.

The study must show that it is following rigorous methodological guidelines but the use of a single case study has dangers. These dangers are related to when no sub units can be identified or when the case study relates to relevant theory which subsumes the case study; thus meaning the study becomes abstract and the results are difficult to measure (Yin, 2009). Using an embedded design can help this situation, where a sub unit is identified and can thus be used because it will give significant opportunities for analysis and provide insights into the single case (Yin, 2009). In this case the unit is the individual developer or student and their interpretation of the procedures they are presented with.

Wolcott (2001) however believes that a case study is a reporting mechanism as opposed to a method that allows the detail of specific research techniques to be presented. This opinion, in some respects, results in some of the criticisms of case

studies. The major criticism is, particularly when case studies are compared to the use of surveys, that the findings of the case study are not necessarily generalisable (Gomm, Hammersley and Foster, 2007a). This criticism relates particularly to the single case study where Stake (2007) and Lincoln and Guba (2007) argue that thick description and naturalistic generalisation may not be useful in scientific discourse of the type that is rationalistic and law-like, but is more intuitive and based on personal direct experience. Any generalisation that is made has to be made by the user or reader of the study and is not made by the researcher (Gomm, Hammersley and Foster, 2007a). This research design will thus be based on Yin (2009) and not on the work of Stake (2007) or Lincoln and Guba (2007); the method of selection of the 'sub units' is defined so that readers can see how the case study is constructed and how the data is collected.

Other issues that Yin (2009) identifies is the length of time a case study can take and the amount of documentation that they generate. However, the use of narratives to describe the findings of the research gives a richness to the results that allows the reader a real insight into the case that is being explored, something that a summation or distillation into a theory will lose (Flyvbjerg, 2006).

4.5.2.2. Interview

The objective of this research is to understand how the subjects come to see the research questions from their perspective and thus the interview can be considered a useful technique. McCracken (1988) believes that the interview is one of the most revealing data gathering techniques and allows insights into the mental world of

the individual and their unique world view, known as thick description; and the examination of feelings or attitudes is particularly suited to using interviews (Gray, 2009). There are a number of different types of interview such as structured, semi structured, non-directive, focused and informal conversational, each is characterised by the amount of active listening that is required (Silverman, 2006, Gray, 2009). Structured interviews require very little active listening as the interaction is pre-decided, however in open ended interviews the listening is active to enable the rich data of interaction to be obtained and the interviewee is allowed freedom to explore meanings and understanding (Silverman, 2006).

Rubin and Rubin (2005) however, use a different taxonomy and believe that the differences should be considered in terms of the breadth of interviewers questions. Initial exploratory questions, will of necessity be wide ranging, but other interviews which focus on a single core idea are likely to be much narrower in focus. Either way, qualitative interviews are considered to be extended conversations. (Rubin and Rubin, 2005, Patton, 2002, Gray, 2009).

The interview design in this research is based on the model of interviewing identified by Rubin and Rubin (2005) as responsive interviewing. Responsive interviewing is based on the earlier outlined interpretive constructionist philosophy and highlights the role of the interviewer and interviewee who together form a relationship during the interview that generates ethical obligations for the interviewer. The interview is about obtaining the views, experiences and

understanding of the interviewee of the world in which they live and work. The interviewer is not likely to remain neutral; Rubin (2005) describes this neutral position as being an automaton. The interaction between interviewer and interviewee is precisely that, an interaction. This is affected by the personality of either role, the questions that are being asked and the environment.

Interviewers should therefore understand that they need to retain an awareness of their position throughout the interview and be aware of how they have affected the session. The researcher should continually examine their own understandings and biases. In addition Gray (2009) suggests bias may occur when interviewers depart from the research design or interview instructions and do not manage to either create or maintain a rapport with the interviewee. The interviewer could alter planned questions or rephrase attitude questions in addition to being careless about prompting responses; they could ask biased probes along with asking questions out of sequence. Active involvement in an interview can create problems as researcher emotions, prejudices and biases can affect both the questions asked and how the interviewee responds. The generation of information that may be personal and private should be treated with care; the situation means that the interviewer has obligations to protect the interviewee. This should be followed automatically for each interview, although the researcher should be aware of interviewees who wish their story to be told and wish to be identified.

This research uses interviews in two specific areas; firstly, as part of the initial case study. Here, the motivation and understanding of the student web developers is explored. This work is done using semi structured interviews where the direction of conversation throughout is previously planned and the interview follows this planned sequence. If an item is particularly of interest it can be followed up within the bounds of the interview plan. The second set of interviews will be part of an industrial exploratory study, investigating the work of web developers in industry. This work is primarily exploratory and as such, with the use of interviews as probably the best approach they will, in the main be unstructured (Gray, 2009).

4.5.2.3. Documents

Documents are one of the most frequently used and unobtrusive measures of research. Organisational and institutional documents are most often explored and the data is often compared to that from other sources. In organisational archives there may be issues of selective deposit and selective survival (Hakim, 2000) where the policy of the organisation may to some extent affect the selection and survival of documentary records. Whilst the documents used in this thesis are organisational documents they are student dissertations, and thus are personal in that they reflect the views, knowledge and experience of the writers. In this case they will have the same issues as personal records and they may be inaccurate, incomplete or contain distortions (Gray, 2009) and should therefore be treated with caution.

4.5.3. Data analysis

Template analysis (King, 2004) can be used as a vehicle for analysing interpretive data. King describes how it does not prescribe a single method of progression, but allows for the thematic organisation and analysis of information; and it is therefore useful for those researchers who require a more flexible approach (King, 2004) than the set process that must be followed by grounded theory for data collection and analysis (Strauss and Corbin, 1998) as it may be tailored to match the requirements of specific researchers.

The template is normally constructed prior to analysis with the aim of aiding the initial analysis. To do this a number of pre-defined codes are constructed, where a code is a label that can be attached to a piece of text. Once the initial template is constructed then the researcher goes through the full set of transcripts, identifying sections of text that are suitable for the project aims and marking them with label(s) from the initial template. It may be necessary to add additional codes, to delete existing codes or to change the scope of previously defined codes (King, 2004).

4.5.4. Validation

The best form of validation in a case study is to give the text to the informants to read and comment upon (Yin, 2009). In addition, by creating a sub-unit within the study of a novice developer it is possible to use them for comparison and validation of the results (Yin, 2009). Finally, the results of the research may be given to a group similar to the informants to critique and thus give very good validation of the findings.

4.6. Summary

This section examined the philosophical background to research in general and this research in particular, focussing initially on the epistemological and ontological areas and basing the examination on the conceptual framework defined by Creswell (2003) before looking at the researchers personal perspective to the research. This was followed by a detailed discussion of the strategic method of enquiry and by the use of the case study as a method. The use of techniques for data collection such as interviews and documents are then outlined, followed by template analysis as the means of data analysis. Finally the issue of validation is explored.

The next chapter outlines the data collection and analysis element of the research and the design of the case study that is used to collect the data.

Chapter 5

Case Studies

5.1. Introduction

As discussed in section 2.5 the investigation of the literature on web development methods suggests that web development methods are not used (Barry and Lang, 2001b, Lang and Fitzgerald, 2005, Taylor, McWilliam, Forsyth and Wade, 2002, Lang and Fitzgerald, 2007). However, there have been no studies and there is no evidence that would point to the reasons why they are not used. A number of suppositions have been made (Powell, Jones and Cutts, 1999, Barry and Lang, 2001b, Barry and Lang, 2002) such as that web development methods are too cumbersome, that developers are too in-experienced or that the methods are too difficult to use.

To further explore these suppositions and to find out what should constitute a web development method (in addition to how successful cognitive dimensions are in describing web development methods) four studies were designed, within the initial case study, to collect the data.

The first study which is outlined in section 5.2 details a study that took place over two years using 23 BSc students at Bournemouth University completing their

dissertation projects, all of whom can be classed as novice developers. Their use of, and comments on, a number of web development methods are analysed (Jeary, Phalp and Vincent, 2007a, Jeary, Phalp and Vincent, 2009).

The second study that is detailed in section 5.3 reports on the findings of the 23 novice developers when they were asked to identify what should be in a web development method.

The third study, which follows in section 5.4., is an in depth exploration of three well known web development methods and their use by novice developers in building a simple web content management system. Their comments and their use of the methods are analysed along with their description relating to the cognitive dimensions.

Finally Section 5.5 describes a case study that details the results of interviews with four web developers in industry. These were carried out to further inform the findings of the earlier studies.

5.2. Initial study

5.2.1. Introduction

Research should be designed to ensure that the researcher collects data that is both relevant and answers the initial research questions in a way that is both unbiased

and logical (Yin, 2009). In order to explore and understand the difficulty in using web development methods and investigate possible reasons for their non-use the initial study was designed as a qualitative case study after Yin (2009). Yin suggests that at the outset of a case study the research questions need to be identified and thus set the initial boundaries for the case study. In this study these reflect the issues extracted from the literature review in Chapter 2. They are:

> Research Question 3: Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?

The rest of the case study was then designed using Yin's (2009) framework.

5.2.2. Design

Yin's (2009) framework identifies the unit of analysis and the propositions as the next steps in further defining the boundaries and scope of the case study. The unit of analysis ensures that the researcher explores the problem of outlining the case for the study. The unit of analysis becomes apparent when the research question that needs to be answered has been accurately identified (Yin, 2009). The unit of analysis for this study will be that of a simple web development (such as a content management system) for a client using a web development method undertaken by a single developer. By creating questions in proposition format Yin's framework forces the researcher to identify what should be studied (2009). The propositions in this study need to identify whether web development methods are difficult to use

for a group of inexperienced web developers and whether this may explain their relative lack of use. They are:

- Web development methods are not easy to use
- There are issues with using web development methods
- Web development methods and their techniques are not easy to learn
- Students as novice developers expect a web development method to give them clear instructions to complete the development process
- Web development methods lack scope when used in practice
- Web development methods are complicated

5.2.3. Data analysis

As outlined in Section 4.5 the data will be analysed using template analysis. Early in a study an initial template may be considered and a few pre-defined codes can be outlined which will give a guide to data analysis (King, 2004). Whilst King (2004) suggests creating an initial template from high level analysis of collected data, the method allows enough flexibility for an initial template to be created using other means.

The initial template was created by taking the macro level discussion topics that were used between the author and students in web development methods lectures and seminars. These topics were initially defined to facilitate the discussion and relate to components from the System's Development Life Cycle introducing areas

of information and navigational analysis which were discussed as necessary to a web development method in HDM (Garzotto, Paolini and Schwabe, 1993); and are thus a good starting point for a template. This can be seen in Figure 6.

1. Requirements
2. Analysis
    - Information Analysis
    - Navigation
3. Design
4. Implementation
5. Testing

Figure 6: Initial template

5.2.4. Conduct of the Case Study

The study ran over two years, using fourth year students on a BSc (Honours) Business Information Technology degree at Bournemouth University in the UK. The students spend the third year of their degree on placement within the computing industry. The degree is at the technical end of the Business IT spectrum.

This student group was selected because it was considered that they had a similar experience level to novice web developers. Because of their background they had some knowledge of the development environment, but were not computer scientists. They all had the knowledge they had received from the course, but had never put it into practice. None of them had completed any significant web application. All students on the course studied programming for two years (mainly Java and some PHP/MySQL), and database design and performance for three years, in addition to marketing, accounting and business systems. For their final

year project the students had to complete a significant, individual piece of development for an independent client in a timescale of less than six months. The final year project was one third of the students' final year marks and thus an important element of their study. The short time scale and the pressurised development environment would thus tend to mirror that of a web development in industry. Prior to the study, none of the students had heard of any web development methods before, but they were aware of the existence of 'formalised methods' as defined by Fitzgerald et al. (2002).

Fourteen students voluntarily completed a Belbin (1981) test and none were found to score highly as Completer Finisher, thus indicating that they were unlikely to have a tendency to use a method and complete it because of character traits. However, eight of these fourteen students scored highly as Resource Investigators, which could correlate to a character type with a tendency toward Systems Analysis (Gifford, Henry and Schoenhoff, 2003).

The participating students were an average set of students shown by the marks awarded for the projects. These were double blind marked and eight students achieved over 70%, five achieved between 60% and 70%, six achieved between 50% and 60%, three between 40% and 50% and one failed.

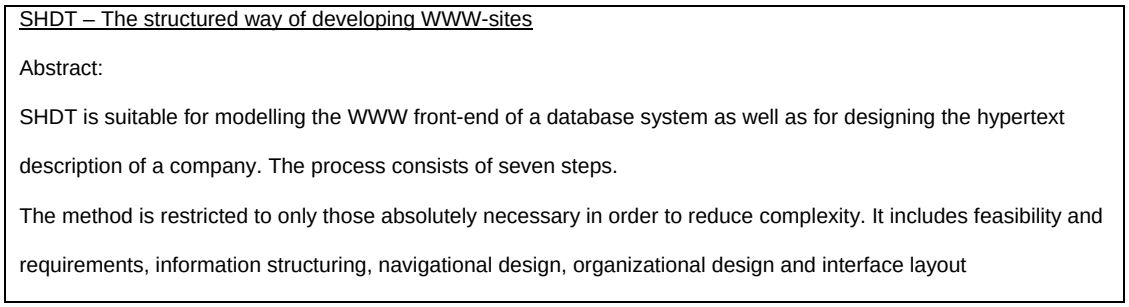
Each student was self selecting in that they requested the author as a project supervisor and were undertaking a web development project. There was no

particular difference in the background of the students that did the study to those that did not. Web development methods were not a final year option and the selected students did a mix of final year options. Each student was interviewed and subsequently given details of three web development methods randomly drawn by the student from those in Table 8. They could use them for the development project if they wished. They were given an overview of each of the three methods (for an example see Figure 7) along with relevant journal papers for them. These web development methods were chosen by the author as, first, being within the scope of the students understanding, and, second, having enough information available that some sense could be made of the contents.

No	**Name/Description**
1	WSDM: A user centered design method for web sites (De Troyer and Leune, 1998)
2	RMM: A methodology for structured hypermedia design (Isakowitz, Stohr and Balasubramanian, 1995)
3	ADWIS: The analysis and design of Web-based Information Systems (Takahashi and Liang, 1997)
4	FECWAD: A framework for effective commercial web application development (Lu and Yeung, 1998)
5	Connallen: Modelling web application design with UML (Conallen, 1999a)
6	DPWA: A design process for web applications (Uden, 2002)
7	First year of study: SHDT/W3DT The Structured Way of Developing WWW sites (Bichler and Nusser, 1996b, Bichler and Nusser, 1996a) Second year of study : SOHDM: Scenario-based Object-oriented Methodology for Developing Hypermedia Systems (Lee, Lee and Yoo, 1998)

Table 8: List of web development methods

The students made available their dissertations which included both evaluation and reflective sections for this research. The student's identities were removed from the work and they were labelled and identified only by numbers. In addition, three students completed follow up interviews. Thirteen students took part during the first year of the study, and ten in the second year. None were offered any reward for their participation.



SHDT – The structured way of developing WWW-sites

Abstract:

SHDT is suitable for modelling the WWW front-end of a database system as well as for designing the hypertext description of a company. The process consists of seven steps.

The method is restricted to only those absolutely necessary in order to reduce complexity. It includes feasibility and requirements, information structuring, navigational design, organizational design and interface layout

Figure 7: Example of a method overview given to students

5.2.5. Case study process

The author read through the dissertations for a first understanding of the work. The researcher first approaches the work being open to the data (Strauss and Corbin, 1998). This approach is common in qualitative analysis when trying to identify themes in the data. This was then followed by a repeat reading, and any sections which appeared relevant to the research questions were marked. The text was then inductively analysed and recurring themes were noted in a mind map. A copy of a section of the mind map is at Appendix A. This same process was followed for the transcriptions of the interviews held with the students.

A template was then created of the themes that were evident. This template was then applied to the dissertations and interviews to add any themes or topics that had been missed.

5.2.6. Initial study findings overview

Of the twenty three students over both years, thirteen decided not to use a web development method at all, and used either a traditional method or the Systems Development Life Cycle. Eight students used part of a method; for example, two used RMM slice diagrams and another used scenario analysis, before abandoning the use of the method.

Only one student used a web development method (W3DT (Bichler and Nusser, 1996a)) for the whole process. He used a reference web site that was available at the time, and spoke to Bichler via email who referred him to a book by Scharl (2000). He comments that *"...the documentation was not freely available...which made the methodology more difficult to understand. Most literature on the methodology was in complex academic language which did not help the developer who was using the methodology for the first time...There was not enough emphasis on the collection of requirements"*.

Of the thirteen students that did not use a web development method, most described the criteria of what they were looking for in a method. They found that the web development methods given to them did not match the criteria they

considered necessary for a web development method. Interestingly, different students considered different criteria as important. For example six students considered coverage of the complete Systems Development Life Cycle important, and six considered user involvement to be paramount.

5.2.7. Detailed study findings

The findings have been categorised to answer each of the initial propositions.

5.2.7.1. How easy are web development methods to use?

All the students reported difficulties in some areas of the methods. As inexperienced developers the students were expecting more guidance, one student considered they needed more assistance "*in deciding which method to use*" and another found that their method selection meant it "*may not have provided the level of support I expected*" whilst a third student "*may not have made the best choice of methodology*". One student believed that reviewing the three methods given them was difficult "…as *reviewing the information…made each method more difficult to understand*" and another thought the three methods "*..lacked a systematic approach…..*". These findings are perhaps to be expected as the students were inexperienced and therefore not confident.

5.2.7.2. Are there any issues with using web development methods?

Four students reported that the method and project were ill-matched. They were creating relatively small developments and felt that the method was designed for a much more complex, large-scale project. They commented that "*the project wasn't*

large enough on the scale that ...method projects are made for..." and that the method was "*overly complicated for the solution being built*"; the use of the method created work in that "*…it was an arduous task selecting the appropriate techniques to use for a small scale project*".

Eight students reported problems with the method stages and their applicability and found there was no guidance as to which parts of the method had to be done and which were "*red tape*" and which could be missed. In addition guidance was missing on "*which parts of the method suited the [type of] project*" the student was doing the web development for. Conversely, some suggested that there was an over-abundance of advice when "*different papers said different things*" and "*the literature based around them is not always consistent*".

The number of models and techniques that a method used, was also an issue for the student in that the "*method was too demanding in its suggested products*". Many development methods required a number of documents and models to be produced and the students were overwhelmed with product that they did not understand and did not consider useful. For example "*…and many documents were discarded as maintaining them required more effort than carrying out the task*" whilst "*…documents created to support the project slowed progression by moving attention away from the overall objective*". Indeed one student had to "*revise the project plan to fit the framework [of project documentation]*".

5.2.7.3. Are the methods and their techniques easy to learn?

Lack of instruction in how to apply techniques was reported by seven students, as were lack of explanations as to why products were necessary and these reasons were given for students abandoning the method …" *because it wasn't helping.. I couldn't see from the model I'd got at the last stage I did how was it going to make me a web site… it doesn't fit. I've got this model now how does that web site fit....*"

Timing issues were discussed and the time taken to understand some of the concepts was an issue " …*the first section went through quite quickly but the next few sections got very confusing and took quite a few hours"* and the "*design method was very confusing and would take time and experience to implement successfully*".

Others may have abandoned the method, but they still recognised some of the techniques they used as useful. For example SOHDM (Lee, Lee and Yoo, 1998) uses Scenario Analysis which it is possible to research and find 'how to' instructions in other sources and one student using it found "….*was pretty good because that in turn went to be the testing for it".*

Finally a number of students added prototyping and iterations as techniques to their development and one commented "*they were good because I could go and do a bit and take it to (the client) and they say oh we like that bit we don't like that bit, rather than going at the end agh we don't like that bit*".

5.2.7.4. Are there missing instructions in web development methods?

Some models or documents were to be reused later in the method to create another deliverable. One student pointed out there was no explanation given for retaining the resources or deliverables from various components within the method which caused a problem. "*How do you know that by not doing a particular component there is not some point further in the development when the deliverable from that component is necessary?"* One student found that "*the ability to know exactly which parts of each phase acted as resources for the next was a problematic factor for me".*

The information given in the method was often not clear enough for a novice developer, "*when decisions about exactly what should be used were not clear*" and "there *were no sets of instructions to do this at this stage*" and the information about "*what is expected at each stage is simply not available"*. Others found that there was no information at a vital stage "*… tried to stick to following one paper but then you get to something you didn't know...they'd come out with this model...and then they'd show you the next model...but they didn't tell you how they produced it*".

5.2.7.5. Do students as novice developers expect a web development method to be useful?

The students abandoned the method for a variety of reasons one of which was their doubt over the value gained from using the method. One student commented that " *none of the methods appeared to offer anything of extra value to the development process beyond that of the Systems Development Life Cycle implemented in an iterative manner*"

whilst others were a little more blunt "*…there was nothing useful in the whole thing..*" and *"...was not as useful as expected. The outcome produced is basically common sense and could have been achieved without the aid of a navigation model….".*

Some students recognised the importance of the method to their development and the confidence it gave to their development journey *"…dismissed the value of methodologies in the early stages of development…, It soon became apparent that the guidance and direction …was going to be a highly necessary requirement".*

5.2.7.6. Do web development methods have enough scope when used in practice?

Seven of the methods included in the study covered much of the Systems Development Lifecycle, However, only two of the methods covered requirements and both assumed that requirements had already been obtained and there was no guidance about the different requirements that were necessary for web development. Similarly four methods only mentioned that requirements were necessary but gave no further instructions. Further, the latter stages of the SDLC were rarely mentioned; in fact implementation was beyond the completion stage of all the methods, therefore testing, maintenance and evolution were not covered. The students commented "*…it didn't cover testing or anything like that…it seemed to stop suddenly*" and that " *it just sort of stops on….design stages it didn't go further".*

5.2.7.7. Are web development methods too complicated?

Whilst some students pointed out that that the methods were too complex for the solution being built and not necessarily adding value to the project (see Section 5.2.7.2); a number believed the methods were too complicated in a number of different ways. Firstly, advice and guidelines given made the method "*...too complex to understand in areas*…." and the academic language was an issue in that " *..the methods were written in intellectual language and [were] relatively difficult to follow*…". Indeed another student found one method " *..very confusing*" and believed "*it would take time and experience to implement successfully*". This adds weight to conjecture that the investment in time must be perceived to add value. If the investment is not worthwhile then the developer will not use the method. In addition one student believed that the developer needed more academic qualifications or experience than they had ".. *[it] was also confusing and possibly difficult to implement without the correct knowledge*..". There were some specific areas where the techniques were too complicated for the student such as attribute design and architecture design "*..so I didn't really understand it …and I didn't get what they were trying to say…I didn't really understand…I mean I understand what attributes are but in the way they were describing them I didn't understand what particularly I was going to get from it that I hadn't got from ER analysis… I had my attributes then so I left it out*….". Once again the student had the perception that if the technique did not add value then it would be dropped.

5.2.7.8. Summary

These initial case study findings should then be linked to the initial propositions to show how the data obtained relates to the questions the case was exploring (Yin, 2009). The propositions are detailed below with the linked findings.

- Web development methods are not easy to use

The proposition was found to be true; all of the students reported problems with using the methods and all but one abandoned their use of their chosen method.

- There are issues with using web development methods.

The proposition was found to be true; there were a number of issues reported such as matching the method to the project, which parts of the method could be missed and inconsistent advice given about using the method. In addition, there appeared to be a necessity to produce too many documents and models that the students did not understand either the requirement for or the use of.

- Web development methods and their techniques are not easy to learn

The proposition was found to be true. There was a lack of instruction about how to apply the techniques and why the techniques were necessary. These reasons contributed to students abandoning the method.

- There are missing or confusing instructions in web development methods

The proposition was found to be true. No explanation was given about retaining deliverables for use later in the method. Information given in methods

was not clear enough; there was no information at points the students considered vital.

- Students as novice developers will find a web development method useful

This proposition had mixed results. Some students found the web development method useful despite dismissing their value in the early stages. Other students found little of value in a web development method.

- Web development methods lack scope when used in practice

The proposition was found to be true. The methods used covered most of the lifecycle but the requirements were always assumed to have already been obtained and the latter stages such as testing were also missing.

- Web development methods are complicated

The proposition was found to be true. Methods were found to be complex, written in intellectual language and difficult to follow. The students found they didn't have the correct knowledge and didn't understand some of the concepts.

5.2.8. First iteration of the template

By taking the findings and abstracting them to the template it is possible to start to build a picture of what a novice considers should be possible components in a web development method using the initial template described in Section 5.2.2. The resultant template is shown at Figure 8. A category of environment/method was added to include the issues brought up and highlighted in the previous sections, such as freely available documentation, simple language and a systematic approach. Other points that were highlighted included the necessity to define

requirements collection (elicitation) in the requirements section. The results also specified that a method should cover the whole life cycle and include prototyping and iteration. In order to separate these elements from method design a category of operations was included as part of the template.

1. Requirements
    - Include requirements collection
2. Analysis
    - Information
    - Navigation
3. Design
4. Implementation
5. Testing
6. Operations
    - Cover whole life cycle
    - Prototyping
    - Iteration
7. Environment/method design
    - Freely available documentation
    - Simple language
    - Guidance on method selection
    - A systematic approach
    - Guidance on what is essential and what is optional
    - Technique instruction
    - Explanation of necessity for products
    - Continuity of resource

Figure 8: Template with initial study findings added

5.3. Second Study

The second study needed to explore which components the novice group believed should be in a method. The unit of analysis was still the same as in the initial study however, the proposition is:

1. There are a number of components that must be in a web development method for a novice.

The criteria that a developer will name as useful requirements for a web development method for their personal use are likely to reflect the background and experience of the developer at that point in their developer career. It is likely that exposure to different development environments will raise the profile of differing issues to the developer. Whilst all students had completed an industrial placement, the students could all be considered novice developers with little exposure to web development in an industrial setting, and few had much exposure to development and development methods in general. Interestingly, whilst all had received elementary instruction on development methods, few had used one whilst on their placement indeed many companies did not appear to use one in practice. Those that had some exposure to method often detailed the exposure and showed the method was relating to programming practice. The words used to describe what they thought of development methods in general were "*difficult*", "*complicated*" and "*frightening*".

A number of the students specifically detailed the criteria they expected of a web development method; however the detail and understanding was very dependent

upon the student. Nine students did not refer to any requirements for a web development method, nor criteria that they expected.

The analysis was completed using the student dissertation extracts and interviews and using an inductive process the criteria were collated into themed areas. A mind map was developed that reflected the criteria that the students considered important. This is shown in Figure 9 and includes the number of students that proposed each of the criteria.

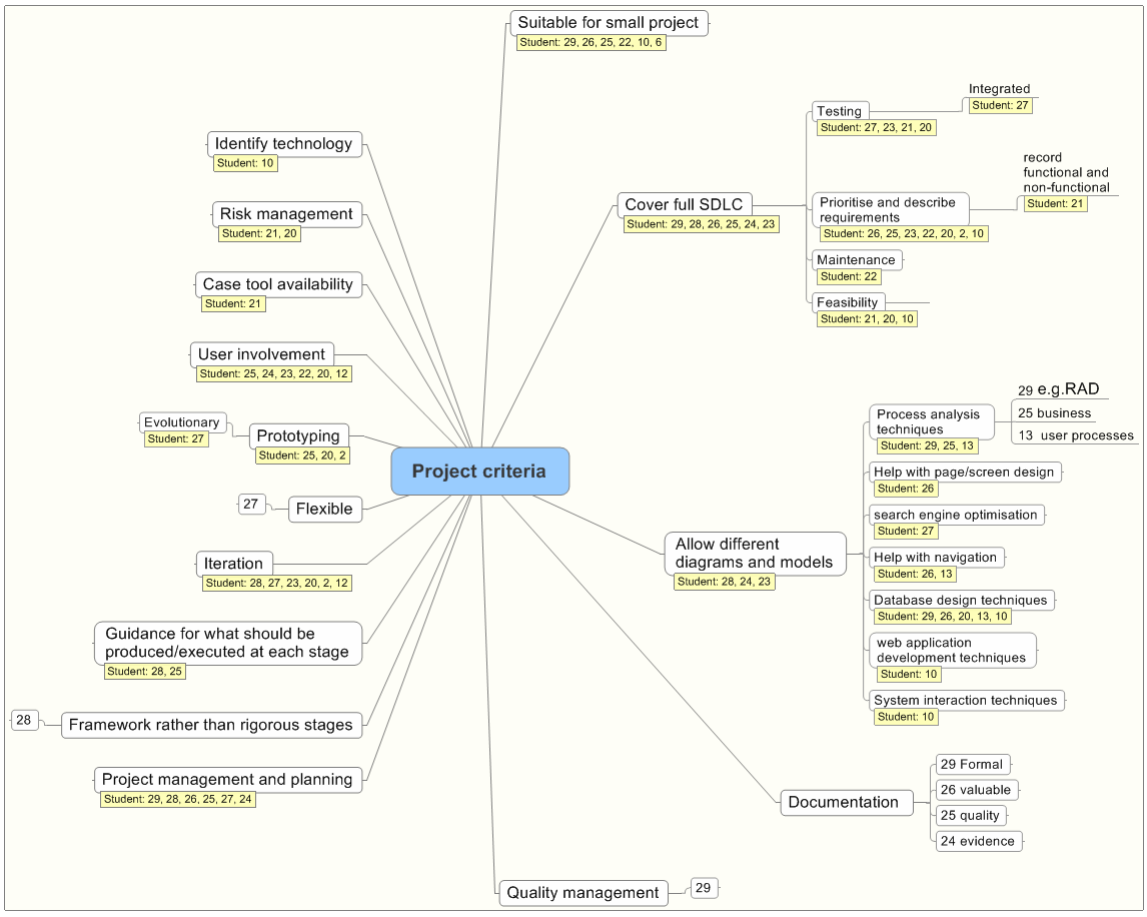


Figure 9: Mind map of student requirements for a web development method

Interestingly the criteria most often referred to were related to the scope of the Systems Development Life Cycle; six students felt it should cover the whole life cycle and seven students listed criteria which were to prioritise and describe requirements and five students included testing.

As can be seen six students considered iteration to be an important criterion and one of them, along with three others, found that prototyping would be necessary. Six students also considered project management and planning to be an important criterion for a method and a similar number believed that user involvement was important.

Suitability for a small project was also included as a category, but students were looking for a method that they could use specifically for their own projects and this was therefore removed as a category in its own right before being added to the template as 'guidance on method selection'.

Aside from database design techniques, which five students suggested, most other modelling or diagramming techniques were suggested by one or two students only. This may reflect the issues students had with trying to implement the web development methods they were given, many of which had a number of models.

5.3.1. Second iteration of template

The findings as summarised above were taken and added to the template described in Figure 8 as possible components of the method. The topics that were identified

consisted of issues concerning the operational side of the method such as scope and method stages, or development techniques. Only those components which three or more students identified were added to the template. There were also a large number of criteria which were placed in a category termed quality attributes. The template was split at this point into two separate sections. The first section, as shown in Figure 10, refers specifically to the components of the method, such as requirements. The rest of the criteria, which refer specifically to method design are in a separate method design template which is illustrated in Figure 11.

1. .Feasibility
2. Requirements
    a. Include requirements collection
    b. Prioritise
    c. Describe
3. Analysis
    a. Process Analysis models
        i. Role Activity Diagrams
        ii. Business
        iii. User processes
    b. Information
    c. Navigation
4. Design
    a. Navigation
    b. Database design techniques
5. Implementation
    a. Functionality
    b. Content input
6. Testing
7. Documentation
8. Operations
    a. Cover whole life cycle
    b. Prototyping
    c. Iteration

Figure 10: Second iteration of template –Part A- Method

1. Method design
   a. Framework rather than rigorous stages
   b. Guidance for outputs at each stage
   c. Allow different diagrams and models
   d. Flexible
   e. Simple
   f. Full and cut down version (Novice version too)
   g. Simple language
   h. Guidance on method selection
   i. A systematic approach
   j. Guidance on what is essential and what is optional
   k. Technique instruction
   l. Explanation of necessity for products
   m. Continuity of resource

Figure 11: Second iteration of template – Part B - Quality attribute criteria

5.4. Third study - focussed in depth study

5.4.1. Introduction

The first phase of this research found that web development methods are indeed difficult to use and understand; they make assumptions about complex computing techniques and use academic language which developers find difficult to understand. In addition they rarely cover the full lifecycle and do not provide enough guidance about how the method and its constituent techniques should be applied. Unsurprisingly if the method or its techniques are too difficult or do not have perceived value they may be abandoned.

In classroom tasks with students that had not taken part in this study it appeared that WSDM (De Troyer and Leune, 1998, De Troyer, 2001) had some techniques such as audience identification that students had found understandable and had

attempted to implement. It was therefore possible that WSDM could be implemented in full by a novice developer. If the novice reflected on the process and could describe any issues then the feedback could be useful in exploring any successes or difficulties in using the method and the full range of techniques that the method contained. This could provide valuable insight into method content and use. Similarly, students in classroom tasks had had some success in understanding the design phase involving structure and site view design in WebML (Ceri, Fraternali and Matera, 2001, Ceri, Fraternali and Matera, 2002). There is considerable information available, published in numerous widely available papers and journal articles which can be accessed via the method web site (WebML, 2009); in addition the authors have collaborated to produce a book which covers the whole development life cycle (Ceri et al., 2003). Finally, OOHDM (Schwabe and Rossi, 1995, Schwabe, Rossi and Barbosa, 1996, Schwabe and Rossi, 1998) whilst no longer being actively supported is one of the most often cited methods with three other methods directly based on it (OOH (Gomez, Cachero and Pastor, 2001), HFPM (Olsina, 1997) and UWE (Koch, 2000)), and with several papers available via a web site (OOHDM, 2009). There appeared to be enough information available to support a novice developer. These three methods were therefore selected for use by novice web developers doing a web development.

This study relates directly to Research Questions 3 and 5

Research Question 3: Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?

Research Question 5: Whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?

5.4.2. Design

To define the boundary and scope of the study the unit of analysis and the propositions need to be identified (Yin, 2009). Similar to the Initial Case Study in Section 5.2.2 the unit of analysis is apparent when the research question is identified (Yin, 2009). The unit of analysis for this study will be that of a simple web development (such as a content management system) for a client using a web development method undertaken by a single developer. By creating propositions the researcher is able to identify what should be studied. The propositions in this study need to identify whether specific web development methods are difficult to use for a group of inexperienced developers and whether the Cognitive Dimensions Framework can provide any assistance in describing the use of web development methods. They are:

- OOHDM, WebML and WSDM are easy to use
- There are issues with using OOHDM, WebML and WSDM at differing stages of the SDLC.
- OOHDM, WebML and WSDM are difficult to use.
- OOHDM, WebML and WSDM are well explained.

- The Cognitive Dimensions Framework allows discussion of the methods and their implementation.

5.4.3. Data Analysis

The data will be analysed as described in Section's 4.5.3 and 5.2.3 using template analysis. The initial template was created by taking the phases of the Systems Development Life Cycle and depicting them in a Mind Map. The individual student work was inductively analysed and mapped to the phases. See Appendix B.

5.4.4. Conduct of the Case Study

Three students in their fourth and final year of either a BSc (Hons) in Business Information Technology or a BSc (Hons) in Software Engineering degree at Bournemouth University completed their final year projects; one using WebML, one using OOHDM and the third using WSDM. The students were self selecting in that they asked the author for a project idea and the author gave each student a web development method and acted as the client for the development of a prototype web application. Each student kept a reflective log as a deliverable. In addition to using the method, each student was asked to evaluate their work using the Cognitive Dimensions Framework.

Whilst there are a number of issues with completing a study of this kind it is expected that any results will highlight some of the capabilities of the method. The first issue is that it is recognised that individual students have different capabilities and strengths, and that any results from this exercise could reflect as much about

the individual as the method. However, it was expected that the individual results when accompanied by the content of the reflective journals and the use of the Cognitive Dimensions Framework would illustrate the issues with the methods. A second issue is that the students are not web developers and are not working in an industrial environment. However, they had each spent a year in an industrial placement and none had been exposed to web development in any depth. Two students had developed web components in the second year but all three students had not done any development whilst on industrial placements and had all been in junior IT management roles. They all had knowledge of programming, databases, systems analysis and project management and could be considered novice web developers.

5.4.5. Case Study Process

The author read through the student dissertations and reflective logs for a first understanding of the work. The process was the same as that carried out for the first case study discussed in Section 5.2.5. This was then followed by a repeat reading and relevant sections were marked. The text was then analysed inductively and the findings were stored using mind maps before relating the findings to themes within the template. A sample mind map is shown at Appendix B. The findings are discussed in Section 5.4.6. This same process was followed for each student and a separate Mind Map created for the Cognitive Dimensions findings. A sample is included at Appendix B and a discussion on the findings follow in Section 5.4.7. Finally, a summary discussion is given in Section 5.4.8.

5.4.6. In depth study findings –part one

All three students produced prototype implementations which were very different in approach; however, all were adequate for the task they had been set, to create an informational student recruitment web site for the course that the student was on. The discussion in the following sections highlights the individual method performance in the different phases of the lifecycle. Actions are recorded at the end of each section. These are points from each section which are noted as being important for any method which is created.

5.4.6.1. Requirements

Whilst requirements specification is integral to any development, web or otherwise, the original version of OOHDM (Schwabe and Rossi, 1995) did not specify explicitly any special techniques for elicitation or for recording requirements. However, subsequently Guell (2000) suggests using user scenarios, use cases and a new notation, User Interaction Diagrams (UID) for OOHDM. The scenario specification proved to be complex and lengthy taking the student over 9 hours to complete and had to be done twice to produce a good output. Whilst the explanations for UID were clear, (in fact the student found them the best explained notation in the whole of OOHDM), the requirements took 50 hours to complete for a simple web site and a significant amount of the time was spent in understanding the techniques.

WSDM does not cover requirements, and it is assumed that requirements are collected before starting the method; the method gives no advice how they should

be elicited or collated which the student believed would have been beneficial in the early stages. The Mission Statement, a key aspect of the method, was considered relatively simple and the need was explained very clearly in the publication. The student found it [the mission statement] "*became a focus point throughout the project and was a constant reminder of the objective of the web site…it not only goes on to help the audience modelling but also helps with evaluation of the project at its completion*". The audience modelling was also considered simple and clearly explained, and the student found the focus on the website user was beneficial; it helped in deciding how the information was going to be displayed and how the navigation should work within the site.

With WebML, the student found that elicitation techniques for requirements appeared to be limited to interviewing the '*main players*' (key stakeholders) and reviewing available documentation. They felt that neither would be suitable in the scenario of their project as the period of access to potential users would not be long enough to interview them. The student added "*The guidance was not very helpful as the authors assume the analyst is already quite knowledgeable…this means inexperienced analysts may struggle with the minimal guidance in what many deem a critical task*". The use case specification sheets were considered useful because a lot of information could be expressed in them, but the use case diagrams were not used as it was felt that they served no purpose. The data dictionary examples were quite hard to understand which meant the student found their application quite difficult. There were a number of method aspects which appeared to relate to personalisation and because the student was not creating a personalised web site these aspects of the

method were missed. Interestingly, the student took 50 hours to complete the requirements phase which is the same timescale quoted by the OOHDM student. (They were in different cohorts and on different courses and unaware of the existence of each other so it must be assumed to be a coincidence). The requirements specification was 27 pages long for a very simple web site. Nearly all the elements of the specification overlapped, leading to large amounts of information being repeated. The student wondered about the size of a requirements specification for a corporate web site using this method.

Action: Specify techniques for eliciting and recording requirements. Ensure simplicity of specification techniques

5.4.6.2. Conceptual design

The conceptual design in OOHDM was a '*challenge*' for the student, the requirements had focussed on the user and their tasks and the student was already building a mental picture of the navigational hierarchy but this stage did not include work on navigation. Modelling was completed, although time consuming, as the authors gave '*vague descriptions*'. This stage highlighted issues with the models created in the requirements stage which the student had created at too low a level and therefore had missed some classes.

This shows that author expectations of model levels may have a number of implications. The conceptual schema created in this stage of the method was relied upon in the subsequent design phase and the implementation, and highlights that

the importance of the schema as any anomalies and assumptions are carried through to the implementation.

The conceptual design in WSDM had some issues, in that the information modelling stage used Object Relational (OR) modelling. The student found OR modelling as described in the method difficult to follow and the method used confusing notation making the step slow and arduous. They reported that the time taken to learn OR modelling, as described, meant that the benefits were limited. Another issue was that the instructions stated that the steps in the conceptual design could be followed in any order. The student having had difficulties in the information modelling stage abandoned it for some time in order to commence the navigational design. However, the navigation could not be completed without the information chunks from the abandoned stage. The student reported that it would give developers cause to lose faith in the method and their own work. Indeed the student considered abandoning the method at this stage; (and may well have done so had it not been the major part of their project). Therefore, there needs to be a description of the dependencies among activities even if there is no distinct sequencing.

There were no WebML comments in this section because WebMl does not have a specific conceptual design phase.

Action: Modelling techniques should be explained clearly and levels of abstraction should be highlighted. Product information should be given showing which

products are required later and a full description given of any dependencies between products.

5.4.6.3. Navigation design

The navigation design in OOHDM was again complex and lengthy to implement, and was confusing in that three papers gave different, sometimes conflicting, instructions and thus created confusion about how to create the models. The technique of creating specification cards for the student web development was a lengthy process and cards only differed in some small aspect, thus creating a large cost for seemingly little benefit. It was realised at the end of the project that the cards were never referred to again.

In contrast, the navigation design in WSDM however, was considered simple by the student and the notation relatively easy to understand once audience classes had been identified.

The student did not make any WebML comments in this section.

Action: Explore the navigation techniques used in WSDM for suitability.

5.4.6.4. Abstract Interface Design

The abstract interface design in OOHDM was the section with the weakest documentation within the method, providing very little information on producing the required models; further, the information given was complex and guidelines for

model completion were not found. The student eventually completed the models from analysis of given examples. The interface design in WSDM was also was poorly explained in all publications and the student was never aware whether they had completed this section correctly. They reported this section as the least useful as it did not cover things like 'look and feel' of the site which the student considered important. It also led to changing the structure of the navigation design, which the student had reported was easy to understand and had originally completed correctly.

The design stage in WebML also caused problems for the student. The core objects list was part of this stage, but had not featured in the specification so it was unclear from where it should come. The guidance given for this whole section was confusing to the student; they had issues understanding the data schema during the earlier stages; they found the hypertext fragment specification unclear and this gave rise to misinterpretation and uncertainty; the student was confused about the differences between core entities, site views and areas. In addition the student struggled to understand the specialised language used in the book (Ceri et al., 2003). For example, the explanation of the sub-schema definition was complicated and the accompanying examples did not reinforce each point well enough; therefore the student used them incorrectly. The guidelines and examples were inconsistent in many places and it was difficult to determine what each stage entailed. The design stage was supposed to outline the navigational and content structure and the student agreed that it had worked "*to a certain extent*". But the design spanned 24 pages which meant that any person viewing it would have to

memorise pages at a time and there was no overview. This stage took 115 hours which was over twice the time reported for the other methods, and the student reported that having spent this time carefully following the guidance that the result would be "*clear, concise and near perfect*" but the "*designs looked as if they could have been developed in a day and most do not have any meaning in the context of the design process*".

There were no WSDM comments for this section.

Action: Nothing additional at this stage

5.4.6.5. Implementation

The implementation using OOHDM created a number of problems when the student went to use the completed models. Issues arose when the student was deciding how to implement the system; trying to work out how to utilise the information that had been gathered and used to produce the models. The student was unable to determine which models contained the most important information and which they should therefore use. They thought there were too many models, and eventually the student only referred to a small number during the build.

The build used the navigational structures which had been both complex and lengthy to create; and revealed a number of anomalies such as duplicate menu items and the location of items in the wrong places. The hardest part of the implementation was found to be incorporating the content which the modelling had forced to be placed on separate pages which pragmatically would have been

combined. This caused difficulties linking navigational structures and sections together resulting in numerous menu items pointing to the same page and many pages with little information on them. This was a major issue for the student and the implementation meant that users often visited pages that contained little information. Interestingly, the Project Portinari website created using the method by its authors has similar issues (Portinari, 2003).

Neither did the final system take account of accessibility or availability, which the student acknowledged as their oversight; however they felt there should be guidelines and standards in the method as to how interfaces should be designed to take this in to account.

Using WSDM, when it came to implementation the student found that the navigation tracks were discriminative (the student implemented an application which suggested that the user click a section if they were 'disabled' and gave them a separate navigation structure to follow). In addition, there were very long navigation tracks for each class variant which a user had to follow before they obtained any information and were thus not the best way to show information on the site.

There were no WebML comments in this section.

Action: None at this stage

5.4.6.6. Summary

All three methods gave problems to the novice developers at different points in the development process. The requirements process was reported as taking 50 hours in two of the methods. The design process was difficult for all three methods, and two of the students reported their implementation using the method as having flaws. Whilst the study was useful in terms of finding the shortcomings of web development using available methods the actions provided no new information for the templates. The student evaluation of their developments using the Cognitive Dimensions Framework (Green, 1989a) is reported in the next section.

5.4.7. Cognitive Dimensions discussion

The students were told that each method had to be evaluated using the Cognitive Dimensions Framework (CDF) (Green, 1989a) which is designed to enable communication about the design decisions for applications and thus give a clear starting point for discussion. Using the framework the students would be able to highlight specific issues in the use of the methods that they may not have considered.

The undergraduate students used a subset of these dimensions for discussion which they selected individually. The student evaluating WSDM did not use the framework for their discussion which is explained by their struggle to use the method and complete the project in the timescale therefore their evaluation was made without the Cognitive Dimensions Framework. However where a point they raised relates to the framework and is discussed by other students, the discussion

relating to WSDM has been added. The questions that the students asked were outlined in Blackwell (2001) explaining the concept of the Cognitive Dimensions (CD's) to undergraduates and are reproduced for convenience in each section.

5.4.7.1. Viscosity

When you need to make changes to previous work, how easy is it to make the change? Why? Are there particular changes that are especially difficult to make? Which ones?

WebML used abstraction which reduced the viscosity of the designs to a certain extent as it allowed changes to be made in the early stages without causing subsequent problems. The viscosity did increase as the design stages became more detailed. When changes had to be made the student experienced both knock-on and repetitive viscosity; knock on having to move all the boxes around to fit one in and repetitive by having to re-draw the lines. The student reported the final design as extremely viscous because of the document size at 24 pages.

OOHDM was very viscous with making changes being a lengthy and demanding activity requiring all models to be re-visited to determine if changes were required. It was easy to overlook areas that needed updating and the number of models and the relationships among them results in the method being slow to deal with change. When requirements were changed or anomalies were identified in the design, significant work was required to update previously completed models and at the end of the project a number of discrepancies were identified which the student had been unaware of whilst modelling.

The student using WSDM did not discuss viscosity.

5.4.7.2. Visibility

How easy is it to see or find the various parts of the notation while it is being created or changed? Why? What kinds of things are difficult to see or find? If you need to compare or combine different parts, can you see them at the same time? If not, why not?

WebML had problems where it was difficult to view components side by side as the design was spread over a number of pages. The student introduced their own secondary notation to combat this problem by using dotted lines and arrows with descriptions of which components fitted together and how they related as a whole.

The students using OOHDM and WSDM did not discuss visibility.

5.4.7.3. Premature commitment

When you are working with the notation, can you go about the job in any order you like, or does the system force you to think ahead and make certain decisions first? If so, what decisions do you need to make in advance? What sort of problems can this cause in your work?

OOHDM ensures that the steps of the method guide the developer through the development process but the student considered the approach to be overly complex due to the number of prescribed models and the dependencies that existed among them. For example the OOHDM user interaction diagrams were used to create the

OOHDM context diagrams. These are Navigational Context Schemas and are used to create specification cards. The context diagram cannot be created until the navigational nodes have been defined. The previously defined Conceptual schema is evolved into the navigation class diagram. The approach is logical (which is, according to the student, its key strength) but the models have to be completed in a pre-defined order since one model is often the primary input for another subsequent model and this causes the development to be "*too cumbersome*" and "*extremely regimented*".

WSDM allows freedom in the order of production of models at the conceptual design stage, but the student found that one model is required as input for another which ensures that a pre-defined order has to be followed.

The student using WebML did not discuss premature commitment.

5.4.7.4. Role expressiveness

When reading the notation, is it easy to tell what each part is for? Why? Are there some parts that are particularly difficult to interpret? Which ones? Are there parts that you really don't know what they mean, but you put them in just because it's always been that way? What are they?

OOHDM with its large number of models meant that a new user such as the student felt obligated to produce all the models as they were unsure what could be left out. This could be explained as a lack of either experience or guidance as to

which models were necessary. This was exacerbated because the student had difficulties determining the purpose and role of most of the models. At a high level the method was straightforward, but when in depth investigations began, the weaknesses of the documentation were identified. The student felt that far too many models were being created and many were inter-related which added to the earlier problems and the student did not see which added value.

WSDM design stage left the student wondering how the design might be updated. This again is caused by the interrelated models. Foe example the information chunks from the conceptual design are used as part of the page design. The page design also uses a structure which is broken down into audience classes used in the audience modelling stage. The website is uses the navigational model which also highlights access to the page.

The student using WebML did not discuss role expressiveness.

5.4.7.5. Diffuseness/ verbosity of language

Does the notation a) let you say what you want reasonably briefly, or b) is it long-winded? Why? What sorts of things take more space to describe?

WebML gave the student a "*two pronged challenge*" in that they stated that the language was often verbose and with technical jargon. They reported that the roles of components within examples were not immediately obvious. The use of UML comparisons were useful (the student understood UML) but this could not extend

to the textual explanations which were often not well explained. The student in fact did not understand some of the notations and therefore did not use them.

OOHDM was similar in that the student reported finding simple concepts hard to grasp (perhaps the restrictive word counts of journal articles and conference papers and the language used led to this issue). However, the student considered the whole method diffuse in the number of models produced for a straightforward and small scale development. Many of the created models were not used during implementation or merely depicted information from previous models in a different way. The method was far too rigid and documentation-oriented in its approach and appeared to encourage a number of models to be produced to represent information which the student considered to be obvious and, whilst worth defining, did not in their view, merit the time taken in producing the model.

The student using WSDM did not discuss diffuseness or verbosity of language.

5.4.7.6. Hard mental operations

What kind of things require the most mental effort with this notation? Do some things seem especially complex or difficult to work out in your head (e.g. when combining several things)? What are they?

WebML caused the student to use hard mental operations throughout much of the project including the requirements stage (which the student felt confident with) where they needed a lot of effort to understand the reasoning behind an instruction or make a decision about what was required.

OOHDM gave the most problems in this area as the student found the method difficult to use, involving a number of hard mental operations. They undertook research to complete some steps but it did not guarantee that an activity or model would be successful and many activities and models had to be re-started a number of times before an appropriate approach was found. The OOHDM documentation was written at a level of which the authors assume the developer will have certain knowledge and thus provides little guidance and few explanations for activities. The research that was undertaken by the student was time consuming and required careful analysis and few occasions were encountered where the same example explained how the method should be followed. The method focussed on what activities needed undertaking and not how to complete them.

WSDM proved difficult for the student because the limited information coupled with different publications repeating each other increased the difficulty of learning.

5.4.7.7. Progressive evaluation

When reading the notation, is it easy to tell what each part is for? Why? Are there some parts that are particularly difficult to interpret? Which ones? Are there parts that you really don't know what they mean, but you put them in just because it's always been that way? What are they?

OOHDM does not provide definitive guidelines as to how it should be applied and in many instances varying guidelines were provided for the same activity, and the

pre-defined order of activities was not clearly defined, thus making the checking of the project progress, and in some instances the next activity became problematic. The creation either of a prototype or incremental development are not excluded and it is possible for partially completed versions to be demonstrated, but not until the implementation stage of the project. The student was also unsure of the suitability of the models as a communication aid for demonstrating the system to the user.

WSDM proved an issue when trying to use the limited resources to relate parts of the method to the application the student was developing.

The student using WebML did not discuss progressive evaluation.

5.4.7.8. Summary

Cognitive Dimensions enabled interesting discussion about the issues involved in the three methods from a novice developer perspective. A simple summary as shown in Table 9 can be useful in showing where the student developers reported issues.

Dimension	WebML	OOHDM	WSDM	Section
Viscosity	X	X		5.4.7.1
Visibility	X			5.4.7.2
Premature commitment		X	X	5.4.7.3
Role expressiveness		X	X	5.4.7.4
Diffuseness/verbosity of language	X	X		5.4.7.5
Hard mental operations	X	X	X	5.4.7.6
Progressive evaluation		X	X	5.4.7.7

Table 9: Summary of Cognitive Dimensions discussions

Key: X means that students reported issues with the method in this dimension.

Whilst this table shows that OOHDM had the most reported issues these results reflect not only the student understanding of the method, but in addition their understanding of the Cognitive Dimensions Framework. As stated earlier, the student using WSDM did not use the framework. However, the discussion around the framework has proved very enlightening and has shown that in any web development method design all of these factors needs to be considered to enable its use by novices.

5.4.8. Third study summary

The case study findings should be linked to the initial propositions to detail how the data obtained relates to the questions the case was exploring (Yin, 2009). The propositions are detailed below linked to the findings.

- OOHDM, WebML and WSDM are easy to use

The proposition was found to be untrue; each of the methods gave the students problems. The design sections of all three methods caused problems in particular.

- There are issues with using OOHDM, WebML and WSDM at differing stages of the SDLC.

The proposition was found to be true; each method had issues at differing parts of the lifecycle. However, all three methods did not cover requirements elicitation and only OOHDM gave further requirements techniques, these proving difficult to use.

- OOHDM, WebML and WSDM are difficult to use.

The proposition was found to be true; all three students found all three methods difficult to use. The students using WebML and OOHDM found them time consuming and all three students had problems with individual techniques.

- OOHDM, WebML and WSDM are well explained.

The proposition was found to be untrue; all three methods were difficult to understand at times. OOHDM had different instructions in different papers which sometimes contradicted each other, WSDM discussed OR modelling which the student could not follow and the design stage in WebML left the student confused.

- The Cognitive Dimensions Framework allows discussion of the methods and their implementation.

The proposition was found to be true; the discussion relating to the individual methods, particularly their techniques was very useful. The questions in the framework gave the students a clear set of phases to describe.

The third study looked at OOHDM, WebML and WSDM, in depth to explore the difficulty of use of the methods for a particular development. It found that whilst there is a lot of information available for each of the method, they all have issues, are difficult to use and are not well explained for novice developers. The Cognitive dimensions framework allowed a detailed discussion about some of the issues.

5.4.9. Third iteration of the template

There were no further points to add to the method template or the method design template at this stage. According to Strauss and Corbin (1998) once there is no further information being gained from the data, the analysis will be considered complete.

5.5. Fourth study - industry interviews

5.5.1. Introduction

The fourth study needed to explore the use of methods in industry. The non –use of methods has already been discussed in Section 2.5 where the findings from four surveys reported that web development methods are not used or that the level of

use is negligible (Lang and Fitzgerald, 2007). The research question that sets the boundary of the study as described by Yin (2009) reflects the issues discussed in Section 2.5 and is:

Research Question 1: Whether web development methods are used?

5.5.2. Design

To define the boundaries and scope of the study Yin (2009) suggests that the unit of analysis and propositions are the next step. The unit of analysis is of a web manager/developer in industry whilst the type of development is that of content management systems. This differs slightly from the unit of analysis in the previous studies in that the developer is not a novice and the development context is of greater complexity. The propositions in this study need to identify the extent to which web development methods are used in industry and if so in what circumstances. They are:

- Web development methods are rarely used in industry
- Web developers do not report the need for a method
- Web developers follow the process of the Systems Development Lifecycle without considering it as a method

5.5.3. Data Analysis

As with the previous three studies, data obtained from the study will be analysed inductively and presented using template analysis (King, 2004). Findings will be added to the templates that have been completed as part of the previous studies.

5.5.4. Conduct of the Case study

Four interviews were conducted to explore the views of web developers who were conducting web development in industry. The interviews explored the issue of how web development was carried out in an industrial setting and whether any of the comments made by the novice developers had any relevance to, or resonance with, industrial practice.

The developers were selected both on the basis of personal contacts with the author and for their roles within the web development community as either developers and/or managers of development teams working on content management systems. These developments were more complex than those of the students as novice developers, but were selected because the development environment was similar to those that the students could reasonably be expected to work in once they completed their studies as novice developers.

The first interviewee was a web team leader in a publicly funded body who was building web based applications that were a part of the institution's web site or their intranet referred to as Web1. The team leader had a History degree and had done web development as a personal interest for nine years since the age of 15. None of Web1's staff had a computing qualification; they were all encouraged to do a philosophy course, even a one day seminar, for '*the underlying logic structures and thinking methodologies*'. The development team was using Macromedia Cold Fusion.

The second interviewee was a web development manager in a different publicly funded institution who was in charge of the re-build of a complex site and is referred to as Web2. The manager did not have a university education but had a sales and marketing background and had previously worked managing information for an international company. In addition, they had been involved in a large scale web re-design and development which required a large amount of information migration. Web 2's staff did have some computing qualifications; they were working on the project part time and were allocated specific tasks at the manager's direction and had little strategic knowledge. One member was interviewed, see Web 3 below. The development environment was bespoke using a Serena Collage Content Management System (Serena, 2008a) the supply of which was discontinued as of 31 March 2008 (Serena, 2008b).

The third interviewee was a member of Web2's web development team and referred to as Web3. They had a degree in Software Engineering Management and were a Senior Technical Developer working 50% of their time on the web re-design project. They had a background in programming in several different languages.

The fourth interviewee was a self employed web developer who was developing Content Management Systems and local government web sites referred to as Web4. They had three years experience in this role. Web4 worked alone, had a Masters degree in Computing and had spent time lecturing students in the past. The development environment used by the developer was mainly PHP and MySQL.

The qualifications and background of the interviewees are summarised in

Title	Formal Computing Qualifications	Experience in web development	Experience in other areas of software development	Position
Web1	None	Nine years informal development practice starting with HTML and CSS and moving to ColdFusion	Project management	Web team leader
Web2	None	Two years working on web site re-design. No development experience	Website design, information management and migration	New media Manager
Web3	BSc Software Engineering Management	One year working part time on project and informal web site creation for family members	Visual Basic, Java, PHP, C++, C#	Senior Technical Developer
Web4	MSc Computing	Three years PHP/MySQL development of Content Management Systems and local government sites	Lecturer in e-commerce prior to commencing full time web development.	Director

Table 10: Web developer qualifications and experience

5.5.5. Case study process

Each of the interviews was conducted in a private room and was recorded with the permission of the interviewee. The interviews were all unstructured but followed a general set of questions which are reproduced at Appendix C. This approach allowed the interview to be both wide ranging and in –depth in areas that became interesting as the interviews progressed. The interviews were then transcribed and inductively analysed; the results were placed on a mind map where the comments were categorised in the same way as the earlier analyses, using the phases of the

System's Development Life Cycle as a structuring mechanism. A sample mind map is at Appendix D.

5.5.6. Fourth study findings

The findings have been categorised to answer the propositions.

5.5.6.1. Are web development methods used in industry?

Despite both Web3 and 4 having an academic computing qualification in their background, none of the developers had heard of, or used any web development method which therefore agrees with previous survey findings (see (Barry and Lang, 2001b, Taylor, McWilliam, Forsyth and Wade, 2002)). Both Web1 and Web4 referred to Structured Systems Analysis and Design (SSADM) (Goodland and Slater, 1995) as being themethod they knew. Neither used it. Web3 had never used any kind of method "*I am told this is what we want and gone off and done it basically*" although he did class the process as evolutionary, "*there being an end goal in sight and moving along bit by bit*". Web1 classed their companies coding standards adapted from Fusebox as a method and Web2 classed running the project on Prince 2 principles as a method.

5.5.5.2. Do web developers believe there is a need for a method?

Web4 believed that " *experience colours the choice of how you do things I think if I had never done this before I would be lost, I wouldn't have a clue where to start um so something that would help me there would be a benefit*" . Web1 had created a process which suited his team's way of working; this was based on what product would be supplied by when and coding standards the developers needed to program. Web2 as a

marketing professional did not understand the need for a method other than a list of what should be produced and by what date. Web3 was not given the option, although he felt that he needed to produce a quality and robust system and was not *"doing what we were taught in uni to do requirements and then the specification…from my point of view that definitely did not happen."*.

5.5.5.3. Do web developers follow the process of the SDLC without calling it a method?

All the interviewees did some kind of design before implementation, with their approach to requirements and evaluation being much more ad-hoc. The requirements approach was similar across developers in the respect that all four interviewees assumed that they knew what the client wanted and client feedback was considered a supplementary process. Web 1 spoke to people who were producing content and asked them for feedback they may have had about what already existed, whereas Web 4 "*made a list of all the things I want to include on the site*" and then would "*chat to people to see if we've got everything*". Web 1 was more honest stating "*its a dodgy way of putting it but our rule in all initial dealings is that the customer is not right 'cos they do not know what they want*". Web 1 then described a "*standard procedure you turn up with a helluva load of post it notes and some ..marker pens and whatever, a list of things that off the top of your head we've developed before and therefore have live versions on that we can demo or that you feel may fall in line with anything that you've seen elsewhere"*. Using the examples they go through identifying the sort of thing the user wants before starting to work out a task flow using post-it notes. "*I always do a storyboard after an initial meeting…I can go back to him and say here*

you are so that's your first screen, here's an example of a transaction happening through three different images...that will be all paper based because people can actually get a physical look at them". Web3 was given no formal specifications, "*we had screen shots and pdf documents…with 'that is what we want it to look like'….I went off and used some of their code but most of it was rewritten…once we had the basics we'd make changes and go back to the manager [Web2]".* He also added he did not know who the user/client was other than Web2. Web3 and 4 knew about specification but had never used one, nor had they ever used any informal requirements document.

Although called the Project Initiation Document (PID) by Web 1 this was more essentially a specification document which covered such things as the scope of the work, ensured that the individual developers on the team were aware of what needed to be developed and what the web development team had agreed to supply. Web 1 used the PID to document their interaction with the client "*from a technical level lets get some basic flow charting worked out here in terms of where you want to go so we know what we need to be building".* Web 2 didn't use one "*at this stage [there] wasn't actually a project initiation document although the project was running in Prince 2 sort of principles and all those sort of things"*

The aim of content is to get information across to users of the site and "*your users need something consistent across the whole site*". There are issues on large institutional sites because "*we need to make sure we don't say it twice and we don't say it with different things*". Content is easy to control if the only people adding content are the web team, but if the content and the management is devolved then there are issues.

None of the developers did any modelling. One stating "*I don't ever use UML unless I really have to* " and "*I tend to steer clear from the whole lot and I use what I like and what I like is the technical side ….we comment out everything anyway...our code is designed to be so that someone with very little effort can pick it up and do it..".* Web 2 did not have any technical knowledge at all and therefore did not understand the concept of modelling.

Similarly with databases "*I tend to try and avoid like the plague working with or leaving myself to work with databases..."* was the comment from Web 1. However, Web 3 and 4 considered an Entity Relationship diagram as standard.

The design elements could be considered in terms of the information, and the navigation. Web 4 considered that the placing of information was important for usability "*web usability is gonna say things like you shouldn't have more than four links off any page for example, you know just a random figure, well that means then I've got to think about what are my top level sets of information how am I going to have to group my information*". Web 2 discussed the design in terms of 'landing pages' where the "*main sort of erm drivers of the visual impact of the new design is that there are these landing pages*" which when asked about were "*[the web site has] 12 or 13 top levels and each of those has a landing page so you've got the ability for them to raise their own profile and have nice graphics and make the web look attractive to users".* The developer's designs were thus very pragmatically completed and the levels were not

systematically devised. This was corroborated by Web3 who was one of Web2's development team.

All four interviewees discussed producing a quality product, with Web1 discussing clients "buy-in to the development" and usability and objective testing with a client user group. Web2 had not considered evaluation until they were asked the question by the author, and Web3 considered testing and evaluation as giving his work to Web2. Web 4 tested the product before handing over to the client.

5.5.6. Summary

As in the previous studies, the findings are then linked to the propositions to show how the data relates to the questions the case was exploring (Yin, 2009). The propositions are detailed below with the linked findings.

- Web development methods are rarely used in industry

  The proposition was found to be true; none of the interviewees had heard of or used a web development method despite two of them having an academic computing qualification.

- Web developers do not feel the need for a method

  This proposition had mixed results; those developers with development experience created their own process but accepted that novice developers may need some help with what to do to undertake a web development. The novice developer, who knew about method and process, was not given the opportunity to use one. The manager with a marketing background was not aware of, nor had any use for, method.

- Web developers follow the process of the Systems Development Lifecycle without classing it as a method

This proposition had mixed results. All developers follow some kind of implicit process to do their web development. They obtain some kind of need from a client, design and then build a solution. However, interestingly it highlighted the developer viewpoint about obtaining requirements and that the user often did not know what they wanted. The developers assumed they knew what the user wanted. In addition only one developer did anything in the way of evaluating what they had done.

These findings provide more information in response to Research Question 1 and whether web development methods are used. The significance of this case study is that it highlights that experienced developers use a process that is based on their experience, whereas novice developers may need some extra guidance, particularly early in the life cycle.

5.6. Conclusion

This chapter has outlined four different studies which were designed to inform the research by providing findings which would answer the following questions:

- Is the uptake of web development methods affected by the difficulty of using them in guiding web developers through a web development?
- What components and techniques should constitute a web development method?

- Whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?

Section 5.2 describes a case study which was carried out over two years with 23 final year students completing a BSc Business Information Technology degree who completed their final year projects given the option of using one of three web development methods given them. All students reported difficulties with using the methods and all but one abandoned their use.

The chapter went on to discuss the findings of the case study and demonstrated that web development methods were not easy to use and that there are issues with using the methods. In addition, there appeared to be a necessity to produce too many documents and models that the students did not understand. The web development methods and their techniques were not easy to learn and there was a lack of instruction in how to apply the techniques and of explanation as to why the techniques were necessary. These reasons contributed to students abandoning the method. The study showed that there were missing or confusing instructions in the methods and there was no information at points the students considered vital. The web development methods lacked scope when used in practice and the requirements were always assumed to have already been obtained. Finally the students found that web development methods were complicated. They reported them as being complex, written in intellectual language and difficult to follow. The students found they didn't have the correct knowledge and didn't understand some of the concepts.

The chapter went on to show how the findings were added to a template laid out in sections which mapped to the System's Development Lifecycle. The creation of a quality and environment section to hold a number of attributes which were non-operational was also detailed.

A second study in Section 5.3 explored the components that the 23 novice developers believed should be in a web development method. Only 9 students provided any information as to what should be in a method, and interestingly they reflected a pragmatic and practical viewpoint which relates to their need to complete their development. Components that were mentioned included, the coverage of the whole SDLC, testing and the description, along with the prioritisation of requirements.

The third study in Section 5.4 then moved to investigating the use of methods in three in-depth projects. Three methods were selected (WebML, OOHDM and WSDM) that students in the cohort had appeared to have some success in understanding, in classroom tasks, one or two techniques; and for which there were a number of references available. The findings showed that the methods were difficult to use, written in academic language and did not provide enough information for implementation. In addition, the findings were reported using the Cognitive Dimensions Framework which showed the usefulness of Cognitive Dimensions in reporting the problems of web development method usage.

Finally the chapter detailed the conduct and findings of the fourth study highlighting four industry interviews which showed that the student comments had significant relevance and resonance with industry practice. In particular, three of the respondents detailed how the client did not know what they wanted and highlighted practice which took their input for granted.

This completes the data collection and analysis sections of the research and highlights the requirements findings from within this chapter. This will inform the detail of the design of a web development method in the next chapter.

Chapter 6

Creation of the Requirements Framework

6.1. Introduction

A case study into the use of web development methods was detailed in the previous chapter and the completion of a template to cover the findings from the study was discussed. The template set out to capture the method categories that the students suggested were necessary for a web development method along with any useful elements that could be used as part of the design of a method. There were elements from across the scope of the full life cycle and therefore the template illustrates what a method should contain[1] in order to provide comprehensive guidance. Adding the details from Table 2 which showed the desirable contents of a web development from the literature the template would arguably be of such excellence that it might well be unattainable for a method to be adopted in practice. Indeed, it is possible that such a method containing all these components and capabilities would be so large and cumbersome as to be unworkable. The studies have found no evidence that any existing method comes close to fulfilling these expectations.

However, what was most interesting was that the requirements stage of a web development method featured on nearly every subject's list in the studies as being

[1] According to the findings of the case study

essential in some way within a method. Indeed, interviews with web developers in industry showed that elicitation and recording of requirements were considered vital to any development; yet very few method authors until recently have responded to these criteria. It may be that the expectation is that requirements are self explanatory and that it is a case of just finding out what the user wants. However, this work shows that web development could be considered different to software engineering particularly in respect of requirements. Table 1 listed different perspectives on the argument that web development is different and a number of these related to requirements; such as volatility in user requirements, the non-functional requirements may be primary and the user is divorced from the development. In particular, the case study highlighted that the requirements element of the life cycle, including elicitation and specification, is essential in ensuring the completion of any web development project. We will therefore use the template as a basis for creating a method, but will focus on determining a detailed requirements phase first.

This chapter will firstly explore requirements as detailed in the literature before detailing the creation of the web development method using the templates from the initial case study and student input over five years.

6.2. Requirements background from the literature

The requirements phase of traditional software development has been well researched over the last thirty years (van Lamsweerde, 2000). Nuseibeh and

Easterbrook (2000) define requirements as ''the process of discovering that purpose (for which the software was intended) by identifying stakeholders and their needs and documenting them in a form that is amenable to analysis, communication, and subsequent implementation". Requirements engineering is a human–centred activity which requires its practitioners to have skills in, and understanding of, a number of different areas. These relate both to the skills involved in the requirements process itself as well as the ability to understand the domain in which they are working.

The requirements process involves such areas as domain analysis, elicitation of the requirements, negotiation and agreement where alternatives are evaluated and risks ascertained; specification at the interface of the problem and solution domain, specification analysis where the specification is examined for completeness, documentation where rationale is recorded and evolution where new objectives modify requirements (van Lamsweerde, 2000).

Web development has a number of issues that make the requirements process more difficult to apply than in typical application development. Firstly, most web development methods assume that requirements have already been obtained and therefore do not consider there would be any difficulties. Lowe and Eklund (2002) however discuss the issues inherent in obtaining requirements late in the development process, when clients are unable to articulate them until after the initial design has been created. These have been discussed in the review in Chapter 2 but are reiterated here for completeness. The issues that arise for the requirements

stage include shorter development cycle times and uncertainty and volatility in user/client requirements coupled with the user/client only knowing their requirements when presented with a solution (Lowe and Eklund, 2002). This uncertain environment causes developers to commence implementation with very little knowledge of the user /client requirements and these requirements only become evident as the system evolves over its life (Lowe, 2003). Consequently, the requirements phase has a prototypical approach as part of an iterative cycle of requirements elicitation, design and prototyping as the clients needs are better understood and the system evolves (Lowe and Eklund, 2002). The work of Lowe and Eklund (2002) is however, at a high level and it does not cover the detailed requirements that should be considered as part of any web development.

The survey outcomes (undertaken as part of this thesis) which are summarised in Table 5 show clearly that of the 52 web development methods only 14 cover requirements elicitation and their subsequent handling to produce the finished system. This may be because the method authors make assumptions about developer understanding and they expect developers to be able to liaise with clients, stakeholders and users to obtain a complete set of requirements which can subsequently be documented, prioritised, actioned and finally used to test and evaluate the finished system. However, our novice developer group considered the requirements section a vital part of their requirements for a method and expected to be advised on the elements that needed to be considered and how to handle such elements. This observation has been corroborated by our own anecdotal evidence from undergraduate project supervision, which suggests that novice developers are

often aware that they need to find out what the user needs and/or wants of any system, but are unable to work out the scope of the requirements process for a web development. The interviews with developers in industry confirm this view (see Section 5.5). Those interviewed believe that the user does not know what they want and therefore need to be guided into making appropriate decisions. That is, they need to be guided as to what are the important issues that need to be explored and what can usefully be omitted.

The only work in guiding developers about what to consider when approaching requirements for a web development to date has been that of Alaa and Fitzgerald (2004) in the domain of e-commerce projects; where a brainstorming approach is used to identify strategic requirements. Their work highlighted the need for a requirements process that is agile and lightweight because of the prescriptive nature of formal requirements techniques. They create a template of issues to brainstorm. This template may well be useful in a particular e-commerce setting but has considerable redundancy when used across the spectrum of web applications. Furthermore, there are several areas that are not explored at all, such as user identification and requirements, and the objectives of the application.

Other requirements work in the field of web development is once again academic community based, for example the work of Bolchini and Paolini (2004) creates a method for obtaining requirements focusing on a specialisation of the *i** framework to allow the developer to obtain hypermedia requirements. It is, as such, a modelling technique and whilst it allows the developer to obtain details of the user

goals it does not provide a requirements framework assisting the developer in ascertaining the direction elicitation should take. Bolchini and Paolini's work explores the use of a requirements technique from software engineering to deal with web development. Similarly the work of Al-Salem and Samaha (2007) uses the work of Kotonya (1999) to investigate the use of viewpoint-oriented requirements specification for a web development. This is also a modelling approach. Escalona and Koch (2006) take this a step further and relate modelling of requirements processes to create a meta model. This eventually becomes part of a model driven approach to web development using NDT (Escalona and Aragon, 2008).

The modelling fraternity in web development explore the use of various models to define navigation and information for an application and assume that requirements already exist; they have become pervasive in their efforts to extol the virtues of analysis modelling and hardly mention requirements at all. High quality Web Engineering papers discuss such issues such as, '*is it necessary to include the user or client in the treatment of Web requirements*?' (Escalona and Aragon, 2008 p. 379) and decide that where possible at least a group of expert users '*could be very useful'*.

Another means of requirements generation from the Information Systems/Software Engineering field is that of using a Soft Systems approach (Checkland and Scholes, 1990). This is done by Meldrum and Rose (2004) who amalgamate it with a business strategy approach whereby activities are mapped to a matrix of information, communication, distribution or transaction 'virtual spaces'. The authors themselves

admit that their approach appears valid, but are unsure if it has any practical benefits and state it will require further work.

Requirements have traditionally been divided into functional or non-functional requirements (Maciaszek, 2007). The functional requirements are requirements that can be met by appropriate behaviour or functionality on the part of the solution system (Bray, 2002). However, Davis (2005) terms the non-functional requirements as non-behavioural requirements and adds that they will include some requirements that are not easily categorised; such as response time, capacity, degradation, maintainability and adaptability, reliability, tailorability and portability (2005). Bray (2002) is somewhat more pragmatic and terms them as performance requirements using a slimmed down list which includes some of the Davis list and includes usability (Bray, 2002). Web development however, involves all the categories from Davis (2005) and usability as suggested by Bray(2002). Bray (2002 p.17) identifies a third category of requirements which he terms design constraints, 'the true non-functional requirements which identify how the system is built but not what it does'. If, as Lowe (2003) believes, web development means that traditional non-functional requirements become paramount, then arguably the requirements process should have greater significance in the development process, and the traditional requirements approaches need to be re-considered.

In summary, web development is different to software engineering particularly in relation to requirements. However, only 25% per cent of web development methods cover requirements in any depth, despite their importance as highlighted by the

novice developer group. A method that guides a novice as to the elements they need consider when obtaining the requirements and how to approach using those elements is something that is required. This thesis therefore, looks at the requirements section of a web development method with specific focus on the inclusion of non-functional requirements to create an all encompassing framework.

6.3. Requirements Framework

6.3.1. Method

To create a requirements framework the templates from the initial case study were taken and it was found that the data on the components necessary for the method were of a very high level, whilst the quality attributes were particularly useful. Therefore to create a web development method a number of groups of final year BSc Business Information Technology and BSc Computing and Internet Technology (CIT) students who had been on industrial placement were used. Some of the CIT students had come through a more practical route and many had extra exposure to industry and web development in particular, in paid development roles. The framework was completed using discussion (and revision) with the students over five different cohorts and took place over a five year period.

For this study, which focused on the requirements for a requirements framework, an initial high level template was created to encourage discussion and following each session the template was updated to reflect the conversations. After five years,

no new additions were made to the template, which appeared to have become stable, and the template was considered complete.

6.3.2. Findings

The initial requirements focused template was based upon a number of initial findings. Firstly, that although the earlier template had split requirements into functional and non-functional, to ensure the importance of non-functional requirements and to ensure they considered evenly with functional requirements and design constraints, they would not be differentiated at this stage. The initial template was designed as a series of questions that could be discussed and is shown at Figure 12.

1. Do we include planning?
2. What is the site for?
3. Should we consider developer constraints here?
   What about customer constraints?
4. Who are the audience?
5. What is the content?
6. Security?
7. What about technical issues?

Figure 12: Initial requirements template

Surprisingly, though expectations might be that the students would have unclear ideas as novice developer, they had strong opinions about what should and should not be included in the prospective requirements part of the method. A number of them felt that some guidance, particularly for people doing web development early in their careers was essential.

The findings became a series of headings that the students considered should be in the method and a discussion follows with each showing the motivation behind their inclusion. A number of students identified with concepts from some of the existing web development methods that they understood, therefore, the statement of purpose and audience identification for the template were elements of WSDM (De Troyer and Leune, 1998) that the students believed could be used to good effect in any method. The students were exploring some approachable techniques from a number of methods in seminar tasks and WSDM audience identification was one of these techniques.

6.3.2.1. What is the site for?

This element is included, at the students direction; it comes from both WSDM (De Troyer and Leune, 1998) and Turbo prototyping (Ghosh, 1999). De Troyer (2001) believes that not having a mission statement or statement of purpose will mean that goals may never be reached and there is no high level basis for any evaluation into the effectiveness of the site.

It is important for a developer to understand the purpose of the web site and to ensure that the application reflects that purpose. The development of a high standard multi-media site that gives an immersive marketing experience is not necessarily the best environment for a transactional site where customers will not wait for graphics to load. Hence, a developer's role (or certainly the development organisation) should include assisting the client in determining and refining the purpose of the site.

6.3.2.2. Who is the target audience?

This is again an element taken from WSDM that the students considered important to include. The identification of the audience is something that is a complex part of web development. There may be market research data available, in which case this may be used, but without this the site may not cater to the tastes of its audience. The target audience is 'that specific audience who is interested in and will benefit from the site' (De Troyer, 2001). Audience identification is a complex process which many developers do not consider in enough depth; for example with a commercial site as well as customers for example there are likely to be potential customers, suppliers, distributors, potential investors etc. One of the major issues is that users are divorced from the development and may not be able to engage with the process (Holck and Clemmensen, 2001). Users of web sites are often asked to complete market surveys about their usage of the site and it is questionable whether market research surveys offer additional value. User forums are often a source of useful information, customer service blogs and wikis can be helpful. Interestingly, De Troyer (2001) suggests looking at the business activities and decomposing them into target audience classes. (The link between the business processes and the web site are not followed up in her work, and are considered outside the scope of the work here. However, this is something that should be explored in further work, and some authors have suggested using business process models as part of web development methods e.g. Bleistein and others (2006)). The two steps consisting of statement of purpose and identification of the target audience are important at a high level at the beginning of any project and during requirements gathering and

specification will be re-visited and undoubtedly identified in greater detail. Web sites dealing with web design also give guidance such as that described by Leigh (2008) which suggests thinking about the following categories:

- Economic situations
- Computer equipment they will likely have
- Applicable jargon or buzz words that your visitors will likely (or won't) know
- Geographic locations
- Interests and hobbies
- Their needs and reasons for coming to your site.

6.3.2.3. What is the business model?

The business model is introduced at this stage. Osterwalder and Pigneur (2002) summarise five reasons why a business model is essential to a business and three of these can be seen as important in requirements elicitation. These are firstly, identification of elements in a domain and the linkages between them; secondly,a formal e-business model allows useful discussion between managers and stakeholders and finally, the mapping of the model provides a foundation for discussion and facilitates change. This model is likely to become part of the discussion between the developer and the client and stakeholders relating to the statement of purpose and target audience, but is added to ensure that the topic is covered. The site will typically need to be paid for either by the client, sponsor or stakeholder and if the site needs to be self-sustaining the business model becomes

an important issue. Therefore both business and revenue models become important in the setting up and operation of the site.

6.3.2.4. Development Constraints

Originally conceived from a development point of view, constraints could be considered in terms of budget, development environment or programmer availability. However, Chevalier and Bonnardel (2007) believe that constraints can be split into either client or user constraints. Client constraints are those that result from interactions with the client or are inferred from interactions with previous clients, and user constraints which are developed as a result of a web developers own usage of web sites. They may address general aesthetics of web sites or specific aspects relating to usability such as the navigation.

6.3.2.5. High Level Content Analysis

Most web sites have some kind of informational content. This may be reflected in the statement of purpose of the site for example 'to provide information to the local community on the different transport links'. However, e-commerce sites which have catalogues of sales information have to ensure that the information is considered and dealt with correctly. The contents of this section are ensuring that the developer considers the information and the ownership, use and updating of it. This is particularly necessary for the developer in terms of 'how' the information is entered and kept up to date.

6.3.2.6. Security implications

Different business and revenue models will require a developer to consider the amount of security and privacy that are required. For example business applications are expected to be secure so that personal data belonging to users is safe. Providers of services need to ensure that they are not open to third party breaches such that business critical information can be accessed or modified. Much of the information available to developers is in the form of how to and there is little reference to the developer reflecting on the different business and revenue models to decide what security will be needed such as integrity, non-repudiation, confidentiality, authentication etc. (Wimmer, Kemper and Seltzsam, 2006).

6.3.2.7. Timescale

The system will have a variety of timescales for development that will fit with the high level business plans, and at this point in the development cycle the developer should be considering the amount of time that the development should take and whether the planned level of business need can be met in that timescale.

6.3.2.8. Technical issues

This section considers the technical requirements at a high level such as hosting requirements and their suitability for the business need, the hardware and software requirements, along with the possible development environment.

6.3.2.9. Towards specification

Having identified the users of the site at a high level, a more detailed analysis is now conducted to identify the different user groups. The form that the students

liked for this section was where each different class of audience is drawn life-size on a wall chart, and their habits, typical likes and dislikes are added by the development team so that the way they use a site is shown in a visual manner enabling the developer to have greater understanding of the user. It also ensures that individual developers have input into the user classification and gives a greater degree of inclusivity.

Each identified user group would have some kind of information needs and would wish to complete a number of tasks. The students therefore opted to create scenarios for each audience type. For each scenario, they defined their information requirements and the functions they need to complete. It would then become possible to link together similar user types who had either similar information requirements or similar functions that they needed to complete.

6.4. Method completion

At this point it is possible to start to look at the development of the full method which will take account of the input from the templates and the work carried out thus far.

6.4.1. The complete method outline

Method design has been the subject of numerous papers and books since the 1970's, as developers and software engineers started to work to improve the software development process. It has been accepted in the method fraternity that no method will suit every situation. (Cockburn, 2000, Glass, 2004). The method described

within this thesis has been designed to assist the novice developer in completing a web development where the team consists of a single or possibly two developers and the build is to complete an informational, community, e-commerce site or content management system. This means that many of the quality attributes that the students considered essential for a web development method (which they described in terms of their own web development) would suit this kind of method.

Therefore, guidance as to when the method should be used has already been fulfilled. The method should be a framework. A framework in the software engineering world tends to mean that an outline software design is created into which can be plugged a number of different components and code libraries to create a software architecture. However, in this context it means a conceptual framework, where a complex scenario is simplified by creating an approach that can be followed. In this respect it has much more in similarity to Multiview 1 which was described by Avison and Woodharper (1990) as an exploration information systems development. It should produce clear outputs, be flexible and simple in design and language. By creating a framework it is possible to describe a systematic approach and show that it is not a 'hotch potch of techniques and tools'. (Avison and Fitzgerald, 2006 p537). From the original quality list from the novice developers as shown in Figure 11, the quality attributes that will be missed are guidance on what is essential and optional. This has been excluded, because this method is designed for novice developers and they need to be encouraged to make their own decisions about what can be missed out of a method. To do this a developer needs to have an experienced understanding of both development and method. This

could be considered dangerous, in that a novice developer may exclude important elements, however with the framework in the form of high level questions it should be apparent where a section does not apply. Since one of the quality attributes was to allow different techniques and tools, this was felt to be important. Therefore, providing instruction in techniques was excluded from the list.

The final method was created using a mind map and included the expanded areas of requirements gathering and specification. Project planning was included as a heading but was not elaborated on at this stage and a decision was taken to leave it to ensure reflection and discussion at the evaluation stage to ascertain if this was a necessary element of the method. A statement of purpose was included as part of the method and this would be further expanded as part of the requirements framework.

The requirements and specification headings were in full as shown Figure 13. The only other element that was added at this stage was design. Murugesan and Ginige (2005) believe that you need to design for usability with both navigation and the interface, comprehension, responsiveness (and therefore performance), security, evolution, growth and maintainability. The design reflects the elements that should have been linked to discussion in the requirements phase and shows the direct linkage between requirements and design in web development. The final outline of the whole method is shown in Figure 14 where the requirements section is reduced to a single heading. There are two particular attributes missing that were in the

early templates of the method, those of prototyping and iteration but they will be included in due course to demonstrate the process element of the method.

Requirements Gathering

1. What is the site for?
    a. To provide a community
    b. To increase company visibility
    c. To provide a new business model
    d. To provide information
    e. To make money
2. Who are the target audience?
    a. How will you find this out? What problems are there to doing this?
    b. What are the expected types? What are the requirements of each type?
    c. Who are the stakeholder audience?
3. What is the business model?
4. High Level Content Analysis
    a. Who is providing the content?
    b. Marketing analysis and planning
        i. What are the expected visitor numbers and types
        ii. Product releases etc
    c. Who owns it?
    d. What about copyright?
    e. How often does it need updating?
    f. Who will update it?
    g. Consider privacy, accuracy, property, accessibility
5. Security implications?
    a. For different revenue models
    b. For different audience types
    c. Sensitive information
6. Timescale
    a. What is required by when?
7. Technical issues
    a. Hosting requirements
    b. Development environment
    c. Language
    d. Hardware

Specification

1. Identify audience types using:
    a. Statement of purpose
    b. Market research
    c. High level target audience identification
2. Write scenarios for each audience type
    a. From scenarios identify information requirements
    b. From scenarios identify functions they will need to complete
3. Identify audience structure linking similar information and functions

Figure 13: Final requirements template

1. Project planning
    - Is this part of the development method or separate?
2. Statement of purpose
    - High level
    - Who are the target audience?
3. Development Constraints
    - Staffing
    - Availability
    - Skills and experience
    - Both client company and development team
4. Requirements Gathering
5. Specification
6. Design
    - Information
        - Storage medium
        - Structure
    - Navigation
    - Presentation

Figure 14: Final method outline

6.5. Summary

This chapter outlined the literature background to the requirements phase in web development and the work carried out in this area to date. It continued with development of the requirements phase of the method, an area that it was highlighted that was missing in much of the web development method research. Finally, the complete outline of the method was made from the earlier templates which showed the result of five years student discussion in the elements they considered useful to web development, particularly in the requirements phase (Jeary, Phalp, Xu and deVrieze, 2010).

Chapter 7

Evaluation

7.1. Introduction

Chapter 6 outlined how the web development method was created using the input from students as novice developers; focussing particularly on the requirements phase. The elements that the students considered important were included and the method was named the Pragmatic, Effective, Common–sense Simple (PECS) method which highlighted the philosophy behind its design and content. To validate these, the method needed to be evaluated. This chapter, therefore, details a review of various approaches to evaluation of development methods and discusses the design and outcomes of the two evaluation approaches that were used. Finally it highlights the changes that resulted from the feedback.

7.2. Selection of method evaluation techniques

There is a large body of work on method comparison (Wood-Harper and Fitzgerald, 1982, Olle, Sol and Verrijn-Stuart, 1986, Fitzgerald, Russo and Stolterman, 2002, Avison and Fitzgerald, 2006) but little work specifically on evaluation. Sol (1983) describes five different ways of undertaking method comparisons which are all with the final purpose of selecting a method to use. To this end, they can, therefore, be usefully examined to see if they could be adapted to

allow the evaluation of the PECS method. Jayaratna (1994) summarises Sol's five different ways of method evaluation which are listed below for completeness:

- Describe the 'ideal' methodology, then compare with other methodologies
- Construct a 'generalised' measurement tool by selecting appropriate features from a number of existing methodologies
- Test hypothesis about the features based on the study of different methodologies
- Develop a common frame of reference for viewing the different methodologies (thus providing a meta-language for communication)
- Develop a contingency framework to allow the appropriate methodology to be mapped to a particular environment.

The first option, that of creating an ideal method, and comparing with other methods is that used by Avison and Fitzgerald (2006) who define an ideal method with 28 criteria, suggesting that these criteria will not be found in any one method; they suggest that the criteria could form a check list which could be tailored for a particular purpose. This is a possible approach for evaluation in this thesis, although the 28 criteria are of necessity at a high level and to be relevant for all methods.

The second option, of a generalised measurement tool which is created by selecting features from methods that may be appropriate, could also have merit. The word tool is misleading and is not a software tool, but more a 'vehicle' for allowing generalised measurement. This approach has already been attempted in the creation of the method and the discussion in Section 6.1 demonstrated how a list of

requirements for the contents of a web development method were collated from the literature as shown in Table 2. However, it became obvious that (similar to the first option above) no single method would meet all the criteria. The use of these exemplar method criteria are an issue particularly when there is still no definitive link between problem domain, the many different types of development and the methods that are available. Glass (2004) suspects that this is a problem that is too difficult to resolve. It leaves open such questions as to which methods are suitable for which domains and which types of development. However, it may be that the approach could be adapted to allow an evaluation to be made.

Both these first two options from Sol's list are also considered to be subjective; how does a user select the features for the ideal when there are probably many, depending on the situation and the perspective of the user. These issues, in addition to the issues about ideal features are also relevant for the third option on the list, which describe testing hypothesis about the features based on the study of different methods, which is more useful to enable method comparison. For example, which features in which method should be selected to create the hypothesis? Hypothesis testing would also be of issue in a human activity system such as a method; if the same person were used to test methods they would use the experience they gained in testing the first to inform the second, and so on.

Jayaratna (1994) considered the fourth approach, a common frame of reference, successful when he implemented the NIMSAD framework, and evaluates the problem situation, the methodology user and the methodology. He uses the

approach to provide a conceptual understanding of a number of different methods. However, the framework suffers from the problem that whilst discussing the methods the interpretation is all Jayaratna's own. Hence it is in turn subjective. It is possible that elements of the framework could be taken to inform the evaluation of the PECS method, as the framework consists of a series of questions that should be considered. The methods that Jayaratna evaluates are Structured Analysis and Specification (DeMarco, 1978) , ETHICS (Mumford, 1983) and Soft Systems Methodology (Checkland and Scholes, 1990), which he selects as they have different structures, steps and stages. Jayaratna believed that his framework would add to the methodology debate and expected methodology evaluation to be as '*much an intellectual as a practical activity*'; he also accepted that his view was subjective and asked readers to examine the methods from their own experiences.

The fifth approach, the contingency approach, is based on the ideas of Davis (1982) in the area of requirements. By measuring the level of uncertainty in a system, he believed that the correct approach to determining requirements could be made. The measures of uncertainty relate to the complexity or ill-structuredness of the system, the state of flux, the user component i.e. the number of users and the level of skill they have, and finally the skill level of the analysts. This work was furthered by Avison and Taylor (1997) who identified five problem situations and suitable approaches for each. For example in a well structured situation, with a well defined problem and clear requirements a Systems Development Life Cycle approach will be suitable. In a similar situation, but with unclear requirements a data, process modelling or prototyping approach will be suitable. This is an early attempt at

classification as discussed in Glass (2004) and the approach is reflected in work relating to problem frames (Jackson, 2001) . This work is an interesting approach and is something that will be pursued as part of further work.

Having explored the evaluation of methods it would appear that there are some techniques that could prove useful in the evaluation of the PECS method; in particular, comparison with an ideal method and the generalised measurement tool.

7.3. Evaluation study design (Part 1)

The criticisms of the ideal methodology comparison and the generalised measurement tool are particularly related to the concept that the views are of necessity subjective. To counter-act these criticisms the evaluation of the PECS method could be given to a number of different people and thus a generalised view of the results could be obtained. This would be a variation on the outlined approaches.

The evaluation of the PECS method was made by a group of developers of a similar experience level to the developers that informed the creation of the method. Therefore 63 final year students on various Bachelor degree programmes as part of the Software Systems Framework at a British University were given the task of evaluating the method. The degrees were BSc Business Information Technology, BSc Computing, BSc Network and Systems Management, BSc Software Engineering and Software Engineering Management and BSc Web Systems. They were self

selecting in that they opted to complete a Level H unit entitled Web Systems and the evaluation was the assignment which was a 20 credit unit within their relevant degrees. The work constituted 50% of the unit mark.

The students were aware that the method was part of a PhD study and aware that the evaluation was also part. It is accepted that there may be some bias, in that students could be concerned about criticising a lecturer's work; however the students were fully briefed about being critical and their choices.

To create the evaluation the students were given a short case study to give them context; they were then asked, in a classroom situation, to list the broad requirements categories they thought they would need to obtain and to identify any additional items that they believed should be in the set of requirements. This was then collected and later returned to them. The students were then asked to reflect on their choices and in specific sections of the assignment were asked to add or remove categories from their choices. They were then asked to reflect on the PECS method and asked to add or remove categories or entries. Finally, they were asked to consider whether there was a 'superset' of method categories and what should be in them.

The presentation of the PECS method did not include a process and it was expected that the better students would discuss alternatives, possibly suggesting using iteration and prototyping as was considered by their peers earlier. The full

assignment is shown in Appendix C and the Requirements Framework is shown at Appendix G.

7.4. Evaluation study findings

All students were allowed to opt out of the study and could do so if they did not submit their work in electronic format. Therefore of the 63 submissions, 48 students submitted a copy of their work in electronic format allowing analysis using Nvivo8. Of the 48 students three students made no useful contribution. These results therefore detail the results from 45 students. Each of the submissions was read through, and any suggestions for item removal from the method or additions to it were noted along with their context. Any specific positive comments were noted as were any specific negative comments.

Overall, the method was well received with 569 comments being noted which were later analysed using an inductive process. Of the 569 comments, 257 were related directly to the categories already included in the method and 188 comments suggested other categories that could be included, a further 124 comments suggested that either a category was missing or wanted to remove a category that was already in the method and were thus also defined as negative comments. One student had 10 negative comments. (Six students made suggestions that were based on incorrect suppositions and were therefore just wrong.) The comments which related to the method categories were placed in these categories and analysed and the number of comments are summarised in Table 11.

Method category	Sub category	No of students	No of comments
Project planning		14	18
Statement of purpose		12	13
	Who are the target audience?	13	16
	What is the purpose?	2	2
Development constraints		17	22
Requirements gathering		33	54
	High level content analysis	5	6
	Security implications	2	2
	Technical	22	28
	Timescale	9	10
	What is the business model?	6	6
	What is the site for?	3	3
	Who is the target audience?	9	12
Specification		14	18
	Scenario Analysis	4	4
Design		12	13
	Information	8	9
	Navigation	9	12
	Objectives	1	1
	Presentation	7	8
			257

Table 11: Evaluation sources and comments relating to the method categories

7.4.1. PECS Method categories

7.4.1.1. Project planning

The inclusion of project planning within the method was an element that had mixed results. This was because some students felt that the project planning should be at a higher level than the development process and other students found that without planning in the development process it is difficult to produce a quality system. The survey by Lang and Fitzgerald (2007) shows, however, that 94% of web developers when surveyed considered that project planning was very important. Student 5 suggested that "*the timeline that is set out for the project is a very important requirement that should be included in every web development project to ensure that it is executed*

successfully, no matter what methodology or other development techniques are being used."

Other students agreed by discussing the timescale requirements of the project and the various requirements deadlines. Six students felt that project planning should be removed, whilst two considered that something like PRINCE2 should be considered as a project management tool.

Action: Use the concept of a timeline for the project.

7.4.1.2. Statement of purpose

Of the 14 students that discussed the statement of purpose in any depth only four would have removed it, two of these because they felt that it was repeated at the specification stage. However, this was part of the method design and ten students agreed it was important with Student 19 recognising the design decision "*the original intent is referred to by different questioning, three times before the design or aesthetic of the system is considered, thereby re-enforcing its objective to the developer. Once this, as the most important factor, is embedded in the developers mind it is only then that the technical factors take over regarding tools to be used etc. This is in my opinion is a great advantage over other methods that leave the original intent to an initial statement that doesn't seem to get questioned again at any stage of the developmental process.*"

Action: Clarify that the statement of purpose is re-visited deliberately at different stages in the development process.

7.4.1.3. Who is the target audience?

The most common comment here was the suggestion to split the target user and stakeholders (4 students) because their requirements would be different and thus could be separated. Only one student would not have used this as a category of the

method, with eight agreeing as to the usefulness of the approach. Two students recognized the use of the target audience again later in the method and the greater detail that would be added later in the iterations.

Action: Clarify that identification of the target audience is re-visited at different stages in the development process and ensure that different classes of stakeholder and audience are defined separately

7.4.1.4. Development constraints

Of the 22 comments in this section, the most interesting discussed the inclusion of budget and timescale which could both give development constraints. Although one student discussed moving development constraints later in the method process, and another student earlier in the process, Student 43 pointed out that "*development constraints appear in the first part of the hierarchal stage before the timescale is identified in the requirement gathering section. Personally I find this illogical and confusing*." and suggested putting it within the requirements gathering section and making the development constraints part of the requirements gathering process.

Action: Put the development constraints category within the requirements gathering section.

7.4.1.5. Requirements gathering

This was the section that resulted in the most comments. 33 students made specific comments in this section and 121 comments were recorded of which 54 were general in nature. Two students suggested the use of time-boxing and MOSCOW rules for prioritizing requirements and three suggested the method should outline some techniques for telling novices where to find information that provides

guidance as to how to obtain requirements because traditional elicitation techniques were unlikely to be useful. Student 43 suggested that a list of possible techniques should be given so that novices could select a technique. The list they provide shows, for example, 8 different elicitation techniques.

The aim of the framework was not to force users to use specific techniques, but to allow the developer a free choice and provide what Cockburn terms a 'lite' method (Cockburn, 2000). But the idea of providing a 'Body of Knowledge' of possibilities is attractive. Hence this suggestion has not been followed up at this stage. (This will be a significant piece of work and will be used as a follow up study exploring technique choices for novice developers).

There was much discussion about which requirements were useful and which method questions identified 'functional requirements'. These needed to have been articulated more clearly in the method. However, the students were all aware of the term functional requirements and were looking for an instruction which sent them to get them. By setting up scenarios at the specification stage, the functional requirements will be teased out without the novice developer realizing it. Student 54 suggested adding two additional sections:

- **"HOW will the site be created?**
    - Which language will be used? (this may be highly dependant on the abilities of the development team)
    - What hardware will be used?
    - What kind of hosting is required?
    - What kind of security is required?
- **WHEN is the product needed?**
    - When is the final deadline?

- A GANNT chart or equivalent should be created, displaying deadlines for each part of the product".

These categories appear useful and whilst re-articulating information found elsewhere in the method, if the questions are moved into these sections then the contents of the requirements gathering becomes more logical. Once again the subject of timescale/timeline is mentioned. The suggestion has been added to the list of method amendments.

There were a further 67 comments which were categorised into each area of the requirements gathering phase. The category with the most specific comments was the technical section. This is possibly as a result of the technical nature of the degrees that many of these students are following; however some points are felt to be relevant. Of the 28 comments in the technical section there was debate by some students as to whether performance requirements should be included, student 54 believed that "*when designing the site there will be many factors that have to be taken into account, which may not be specified as functional requirements, but rather as performance requirements. These days, however, performance has become a large part of the functionality, so the two overlap"*. This is a well articulated thought and reflects the nature of the method, thus technical requirements will remain. Student 5 agrees and states that "*by defining exactly what technologies that are going to be used the web developers can educate the clients on the technologies and their capabilities – which may have a major impact on other parts of any development method, such as design and navigation"*. There is some evidence that client understanding of technological capabilities is low at the outset of projects (Lowe and Eklund, 2002) and by

explicitly introducing it at the requirements stage it may help the development process.

Action: Add two additional categories to Requirements Gathering. How will the site be created and using which technologies? When is the product needed? Further work will explore the production of a list of techniques and tools that will be useful for novice developers in the different sections.

7.4.1.6. Specification

Only 14 students commented on the specification, most agreeing with the proposed structure, although two clearly did not understand the concept of specification was. 5 students comment on scenario analysis and Student 57 commented "*scenarios should be produced. I have assumed that these scenarios are in textual form from the use of the verb "write". I believe this should be replaced with storyboards and prototypes, rather than written evaluations of scenarios.*" This is a useful observation and one that will be added to the amendments. It is probable that many users will understand the storyboard much more clearly than other analysis models or large amounts of text. However, more formal models or structured text may be required for some web developments when dependencies amongst actions will need to be explored. Therefore the wording is left as a suggestion.

Action: Scenarios should be 'created' not written, and storyboards or prototypes may be considered.

7.4.1.7. Design

Of the 43 comments in this section, many were relating to the specific categories of information, navigation and presentation. There were no disagreements in the

inclusion of these categories although there was more discussion on the inclusion of navigation. One student considered that an element of the requirements process should be included in the design and then the requirements should be revisited *"[presenting the client with their model] of the problem domain and a potential solution would provide the client with a greater understanding and thus enable them to visualise and communicate their requirements more effectively"*. This idea is taken from Lowe and Eklund (2002) which the student felt was a very useful approach. There is an element of this in the PECS method, but by not overtly discussing iteration and prototyping in the student assignment, many students tried to use the method with a waterfall process and did not consider any other form of process hence not realizing the linkages between requirements elicitation and design.

Action: None at this stage

7.4.2. Additional categories

There were 188 comments regarding categories that were not mentioned as part of the method, which the students considered should be included in the method.

7.4.2.1. Analysis and techniques

Most students agreed with the early phases of the life cycle, which is where the PECS method concentrates, however, three students advocated the use of SWOT analysis to assist in the development of the business model and one suggested that knowledge of competitors and their offerings would be useful. Two students suggested that budget considerations should be considered in the method, as technical choices may be dependent on budget constraints and three suggested that

legal issues should be considered, if necessary. A further three students suggested some kind of user needs analysis, which is mentioned as being vital by Lowe and Hall (1999) but is not specifically defined as to how it is to be done. Four students recommend using Joint Application Development (JAD) workshops (Avison and Fitzgerald, 2006) to assist with defining the client and user requirements and this idea can be useful as a possible technique for client requirements however, to manage to get user representatives in a room with developers in the web environment is a difficult proposition. Once again the use of specific techniques has not been included in the method at this stage, however, depending on the type of development different techniques could be useful, and as stated earlier will be part of further work.

Action: Use of SWOT Analysis, User Needs Analysis and JAD should be considered as possible candidate techniques in further work

7.4.2.2. Lifecycle stages

Six students suggested that a specific feasibility study should be considered at the start of the development project and thus give benefits if the developer(s) has a detailed understanding of the environment and area of work that the project is situated in (Pressman, 2000). One student considered that implementation should be more specifically detailed, whilst ten students considered that the method does not specifically consider maintenance or, more particularly, evolution. The maintenance of web applications has been likened by Lowe (1998) to gardening, the fact that the web application is built in response to a business need, and the business needs evolve along with the web application. The code base will grow and

needs to be regularly maintained to ensure that it functions efficiently and continues to respond to business needs. This should be considered as part of any method. The major omission from the method in the student view is the consideration of testing and specifying its position within the method. Fourteen students identified it was extremely important and they proposed various places within the method for its inclusion from within the requirements phase to within or after the implementation. Interestingly, none of the web development methods explicitly discuss testing or implementation of a testing approach, although Test-driven development is currently used in many industrial environments (Beck, 2003), and has been considered by some to be an efficient approach to programming (Erdogmus, Morisio and Torchiano, 2005); it could certainly inform the way the PECS method evolves, as it involves selecting features from the user stories (read scenario's for the PECS method) and writing tests for the features of the story.

Whilst not specifically a lifecycle stage, 16 students considered that security was such an important feature that it should be highlighted in greater depth than just security implications within the requirements gathering phase. Student 22 suggested "*…the consideration of a site security policy which includes Permissions, Access, Data Protection and Critical Area's and …..*

- *Identify Sensitive Information (Data protection)*
- *Identify area's to be secured*
- *Identify suitable security level for revenue*
- *Identify Audit Level required (minimum logging to full path traceability)*

- *Trust and Credibility"*

Action: An evolution/maintenance phase should be added to the method. A specific testing regime should be considered as part of the method. Give more attention to security issues throughout the method. Add an option of a feasibility study.

7.4.2.3. Process

Interestingly, most students considered the process in a waterfall manner which may be a reflection on what they have been taught or have experienced. There were only ten students which included iteration and these and a further three included prototyping. It is considered that requirements are unlikely to be defined in the first cycle of development (Standing, 2001) and will be collected iteratively (Grunbacher, 2006), therefore iteration is an important aspect early in the development cycle. In addition students outlined the importance of design elements within the requirements process allowing the developer to tease out user requirements particularly with respect to the look and feel of the site (see Section 7.4.1.7.) and elements of prototyping and iteration will be useful here.

Action: Add specific iteration and prototyping to the method

7.4.2.4. Human Computer Interaction (HCI)

This section included discussions by students specifically on accessibility and usability and there was little discussion on HCI specifically; again this may be a reflection on what they have been taught or experienced. Whilst accessibility is an extremely important issue in web design the approach to accessibility is

generalisable across all web developments and considered as a standard. There are excellent guidelines available from the World Wide Web Accessibility Group and therefore accessibility issues are considered outside the scope of this work. Usability was mentioned by six students and they mostly considered it a design consideration; that you needed to design for usability. Again this is considered outside the scope of this work.

Action: None at this stage

7.4.2.5. Support

The support element consists of several categories that were placed together for pragmatic reasons. The points that students picked out included the fact that the method did not mention documentation and six students had concerns for this reason. It is certainly a factor that should be considered in any method, and the current agile ethos that considers only producing a document if it adds value to the development would be a premise that would fit with the pragmatic beginnings of the method. The other issue that students considered important in this section was method use, where 9 students had concerns that it wasn't necessarily useful for different kinds of developments and may not be useful in a large scale development.

Action: Add the use of documentation as part of further work

7.4.3. *Overview comments*

A number of students had general comments about the method, both good and bad. The good comments from students were of a similar nature to "*I liked the flexibility and adaptability*" to "*…overall the method flows and it is easy to understand*". The negative comments were such as "*they do not offer enough information to be able to create a whole web system with, and only give basic information as a starting point for the development process*" and "*there is also a lack of practical guidance, that some users would find difficult as the description focuses on the rationale rather than the application*". Other students thought that the method "*would benefit from the inclusion of modeling techniques giving clear indication of what outputs are expected to be produced at each phase*".

Student 51 believed that "*the methodology would be most successfully utilised as a lightweight extension to an existing agile development method such as XP offering developers relevant information regarding the requirements of a website*". This is both an interesting and intuitive observation, as the requirements gathering section of the method will sit very easily as a front – end process on either Feature Driven Development (Cause, 2004, De Luca, 2009) or Test Driven Development (Beck, 2003).

7.4.4. *Method amendments*

The comments discussed within the evaluation were considered and those that showed promise were abstracted for further work and inclusion into the next

version of the method. The major changes have been identified and listed at Table 12.

Addition to method	Discussion see
Use the concept of a timeline for the project	7.4.1.1
Clarify that the statement of purpose and identification of the target audience are re-visited at different stages in the development process.	7.4.1.2 7.4.1.3
Ensure that stakeholders and audience are defined separately	7.4.1.3
Put development constraints category within the requirements gathering section	7.4.1.4
Add two additional categories to Requirements Gathering: How will the site be created? When is the product needed? Further work will explore the production of a list of techniques and tools that will be useful for novice developers in the different sections.	7.4.1.5
Scenarios should be 'created' not written, and storyboards or prototypes should be considered.	7.4.1.6
Use of SWOT Analysis, User Needs Analysis and JAD should be considered as possible candidate techniques in further work	7.4.2.1
An evolution/maintenance phase should be added to the method. A specific testing regime should be considered as part of the method. Consider if security issues are well covered throughout the method. Consider adding a feasibility study.	7.4.2.2
Add specific iteration and prototyping to the method	7.4.2.3
Add the use of documentation as part of further work	7.4.2.5

Table 12: Method amendments after Evaluation Study (Part 1)

7.5. Evaluation study design (Part 2)

A second approach to evaluation is to create a dialogue as discussed by Jayaratna (1994) which will enable shortcomings and issues with the method to be identified. Therefore, the second part of the evaluation involved seven final year Software Systems Framework students on BSc Business Information Technology, Computing and Software Engineering completing their 60 credit final year project (which

makes up half the final year mark) using the PECS method. The students were self selecting in that they were producing a web application and decided to use the method as a development method. The students were split between two different first supervisors, neither of whom was the author. The types of web applications they produced were mostly a variant of some kind of content management system, for example, for a parish council, for an international bank or a photographic studio. The student project reports were made available to the author for analysis, and these were analysed inductively and the results were stored in a series of mind maps. The positive points and the negative points were placed in separate mind maps and categorised into major themes. For a sample, see Appendix .

7.6. Results

The projects resulted in seven different web applications of differing success. This would be expected from a cross section of student projects. Interestingly, nearly all students commented, as a plus point of the method, on its simplicity, flexibility and ease of use; although the same simplicity and flexibility were also picked out as negative points suggesting that the developer needed experience to be able to follow the method. The major categories and comments are discussed below.

7.6.1. Project Planning

Three students commented specifically on the project planning phase all suggesting it needed "*further definition*". This area had been left out of the original method except as a heading as a placeholder to enquire whether it should be part of the method. All students either created their own project planning method using Gantt

charts or timelines or adapting other methods such as Hallows (2005) who has a check list of 15 items. This approach brought the student into conflict with the PECS method, as Hallows list included such things as "align the schedule to customer requirements" and because the project planning was at the start of the PECS method the student had no requirements yet. One student created a " *…Gantt chart [which] was used to timetable phases of development and set deadlines for aspects of the project… but was not referred to again*", whilst the student that used Hallows list had "*phases overlapping and not being completed in time".* This bought into focus that different developers will expect different amounts of guidance, and that a method creator has to take this into account.

Action: add a parallel project management activity and define the steps similar to Hallows (Hallows, 2005) at their respective points in the process.

7.6.2. Requirements

This was the section of the method that received the most positive comments and was "*the most straightforward stage to follow comprehensively*". The simplicity of the language "*that on the whole could be understood by people outside academia and people with little technical development experience"* and meant that the steps "*could be discussed with a client to gather more detailed information regarding what the system needs to incorporate"* were considered important points by the students. Another student used "*the points as the centre for brainstorming sessions helping to note and develop further requirements*". One student believed it "*simple to apply as a result of the clear communication of its key principles, and the amount of information it puts across. This makes it easy to think of the requirements, making this a good way to document*

requirements for projects like this one". Although another student felt that they had too much information and created "*a disorganised requirements document"* and felt it "*could be a reflection of the method, ….or the developer….but a technique to ..fathom….documentation may have been useful here".* Inclusion of several steps received positive comments, including statements of purpose which "*gave a high level overview of the aspirations of the website and was good for remaining focussed during the development*" and user identification when comments such as a "*…clearer understanding of what is needed from the website is generated*" were made.

Three students felt that there needed to be some guidance on how to collect the requirements, one commenting that the lack of guidance "*adds flexibility to the person using the method but may make it harder to follow, and restrict [its] uptake*" and another stating it could "*leave people lost as to what they should do with the suggestions the method provides*".

Action: the addition of a selection of requirements gathering techniques is seen as outside the scope of this work, but will be considered as part of further work.

7.6.3. Specification

The specification phase did not give explicit guidance as to what should be included or documented, and this was selected as a problem by three students. Two students found it a "*time-consuming process*" and one found it "*the largest phase in the development other than the implementation*". Because the method did not recommend any tools or techniques then one student thought that "*they may have difficulty in completing this section as they would not know how to specify the web site*". Two students

felt that it did not add any value to their development, one stating "*it did not provide useful output*" and the other found it "*not very useful…in comparison to how much time it took to put together*". These comments both relate to a one person, small scale development where arguably a specification is less of an issue. However, we believe that the specification is a valuable stage in the development process and gives both contractual and developmental information that is necessary in any project that has more than a single developer.

Action: Consider the use of a specification template and investigate the use of specification in the method with more than one developer as further work. In addition the earlier identified as further work in Section 7.1 in terms of problem frames could have merit here.

7.6.4. Design

The design phase of the method lists an overview but does not recommend any specific tools or techniques, and this was a factor that was commented on by five students. A variety of comments were made about the section from it "*being insufficient*" to not considering "*issues such as colour, page layout or site hierarchy*" and "*data structures*" thus "*as a result less experienced designers would be more susceptible to designing a substandard system*". Another student believed that "*a suggestion of design techniques would be helpful in informing the designer how to formulate there[sic] designs*", whilst another believed that the "*suggestion of possible design techniques to use for this phase would also strengthen the method*". This criticism is accepted. The method is focussed toward the requirements elicitation and is therefore lacking in detail in this section.

Action: suggest suitable techniques for the design phase of the PECS method as part of further work

7.6.5. Implementation

This section is not detailed in the method and is pointed out by three students as needing more detail. There need to be "*at least some considerations given….or perhaps a way of structuring the development…*", "*PECS does not state what to do for the implementation, let alone how to implement the site*". As novice developers it is possible that some students were looking for coding assistance within the method, and certainly there is some informal evidence that this is the case. Students under-confident in coding find the early stages of development daunting and consider that a method should explain where to start and give a list of what steps to follow.

Action: Further work could consider the role of method at the development stage for novice developers.

7.6.6. Other points

The final issues that were raised by the students have been categorised into a number of different areas:

7.6.6.1. Structure

The students found that the structure was a "*crossover between a method and a framework*", and "*it guides the user along the development lifecycle using 6 steps*". One student thought it versatile, and others that it, *" was very easy to follow and apply"* and *"was not too cumbersome"*. One student thought that *"it gave structure to the project and the development*" and another that it had "*an easily comprehensible*

structured order and is almost a step by step guide to web development"; although one student believed that "*it was broad enough for tailoring, but perhaps had gone too far in the other direction and being so open-ended it doesn't give any value to the developer*".

Action: None

7.6.6.2. Guidance

The design of the method was based upon a series of questions which the developer needs to answer as part of the requirements process and one student comments on this stating that it is "*left up to the user to interpret how the results will be found*". This once again refers to the lack of techniques that the students believed should be provided as extra guidance. Another student felt that there "*were clear steps that were repeatable; however the considerations and the lack of detail leave a lot to interpretation and therefore restrict the validity of its repeatability*." The student was concerned that the method would be interpreted differently by different developers. However, this was an intentional part of the design and, as such, has its own element of risk. Methods are differently interpreted by developers depending on the role of the method, the business development context, the experience of the developer and the type of system being developed (Fitzgerald, Russo and Stolterman, 2002). However, it is accepted that some sections did not give enough guidance, particularly the design section as discussed at Section 7.6.4. One student felt that "*novice developers would be completely lost if they didn't have other knowledge to use in the sections were [sic] not much guidance is given*". However, one student felt that because the method "*never forces the developer to perform a phase in an exact way…[it] allows the project team to decide what techniques to use to perform the phase*" and another felt that developers

"*with different skills and preferences for the coding of a website, to use their preferred techniques*". One student felt that the whole method required some "*prior knowledge and experience in development…..to perform the whole method. As the requirements gathering phase is so detailed and maintains the use of simple language throughout it is the exception to this*". This difference reflects the developer confidence with the development process and the more experienced novice requiring less explicit guidance.

Interestingly, some of the developers when using the method required their supervisor to remind them of the tools and techniques they already knew and suggest they use them. The use of the method meant that they 'forgot' some of their previous knowledge and experience, and did not consider their use within the method. They were expecting to be guided through every step and stage.
Action: As discussed in earlier sections a further study will explore the use of more guidance for complete novices and suggestions as to tools and techniques for specific areas.

7.6.6.3. Iterations

The iterative nature of the method was not made clear enough for the developers. Whilst they agreed that it was "*possible to re-visit each stage and make alterations as appropriate*" and "*it gave the developers some degree of freedom when carrying out the different phases*", they also needed to be more clearly defined. The iterations allowed "*the updating of requirements and design when changes need to be made, thereby allowing faster…response to changing situations*" but developers felt they needed more

guidance as the method did not "*state the recommended number of cycles or iterations needed*" nor "*when the iterations should be made*".

Action: Clearer definition of the iterations should be made in the guidance

7.6.6.4. Prototyping

The prototyping was considered useful by two students. One thought that it was good "for web development where users' needs are less clearly defined as it promotes a continuous dialogue" and the other thought it gave the project "the flexibility to make slight changes to the design throughout the prototyping process".

Action: None at this stage

7.6.6.5. Categories

This section was created by drawing together the categories that the developers felt were missing from the method. These reflect some of the earlier concerns from phase one of the evaluation. The main issues were the lack of a clear test plan and lack of recommended documentation.

Action: A more comprehensive test plan will be explored in further work along with guidance as to when to complete documentation.

7.6.6.6. Method Use

Four students commented on the use of the method, one believing it to be "*very effective for business sites*". One thought it more of a framework which needed to be used alongside other methods and techniques although another considered it "*extremely versatile in that it can be usefully applied to the development of all sorts of*

systems and by different types of developer". From their use of the method, one student produced the scenario where the method would be useful:

- *"The development team has a strong set of skills to apply to help achieve goals of phases.*
- *There is thought to a lot of different sorts of requirements that need to be realised for the system to be a success.*
- *The developers want the method to be quick and easy to follow.*
- *The developer has little prior experience of performing the whole lifecycle of a project.*
- *Communication of plans between developer and client is of high importance."*

Action: More detailed guidance as to where the method may be used should be published in the next iteration of the method.

7.7. Method amendments (Part 2)

As a result of the evaluation studies a number of issues have been identified (see Table 12 and Table 13). The final version of the framework taking into account the suggestions from the evaluation is at Appendix G. The requirements gathering stage, which was the focus of this work, has proved very successful. Students commented on the comprehensiveness of the guidance that is written in plain language which has allowed them to enter into dialogue with the clients and stakeholders. Whilst the rest of the method has some issues in terms of needing more complete guidance (particularly for novices) and suggestions for techniques which may be used, it has not been the focus of this study and is a rich seam of

study for further work. A full list of suggestions for improvements as part of further work is shown at Appendix G.

Addition to method	Discussion see
Add a parallel project management activity and define the steps similar to Hallows (Hallows, 2005) at their respective points in the process. Use the concept of a timeline for the project	7.6.1
The addition of a selection of requirements gathering techniques is seen as outside the scope of this work, but will be considered as part of further work.	7.6.2
Consider the use of a specification template and investigate the use of specification in the method with more than one developer as further work.	7.6.3
Suggest suitable techniques for the design phase of the PECS method as part of further work	7.6.4
Further work could consider the role of method at the development stage for novice developers.	7.6.5
As discussed in earlier sections further work will explore the use of more guidance for complete novices and suggestions as to tools and techniques for specific areas.	7.6.6.2
Clearer definition of the iterations should be made in the guidance	7.6.6.3
A more comprehensive test plan will be explored in further work along with guidance as to when to complete documentation.	7.6.6.5
More detailed guidance as to where the method may be used should be published in the next iteration of the method.	7.6.6.6

Table 13: Method amendments after Evaluation Study (Part 2)

7.8. Summary

This section has explored the issue of method evaluation in the literature and used an adaptation of Sol's (1983) approach. Understanding the criticism about the techniques of creating a generalisable measurement tool, the evaluation sought to negate some of the criticism by asking 64 students to do the evaluation. By inductively analysing the responses of 45 students, a less subjective approach has been created. However, it is recognised there is still an element of subjectivity with the cohort of students that were used.

The evaluation study for this work was divided into two halves. The first study involved 64 students evaluating and critiquing the method. This resulted in 537 comments which were inductively analysed and formed into categories which were then individually discussed. Many of the comments discussed categories of the method that had been omitted mostly by design or occasionally by oversight. The second half of the evaluation involved seven students developing a system as part of their dissertation project using the method, and critically reviewing its use.

The evaluation has been thorough and has found that the requirements gathering element of the method is particularly successful, and whilst the method assisted in the development of the projects, there are some areas that will need further work in the future, particularly in relation to design and testing. A full copy of the method as amended with future work indicated is at Appendix G3.

Whilst the method has, as yet, not had formal validation in industry, two of the students that evaluated the method in Evaluation Study 1, went out into web development and used the method. The first added it to the companies existing process and found it added value where the company's process was poor particularly in the requirements gathering and specification. A second student, who gained a position as a trainee web developer, realised that his company were not using any web development method. He introduced the PECS method by asking the other developers and his superior the questions that were set out in the method. As a result the method was put into use and the ex-student promoted to Lead Developer.

Chapter 8

Conclusions and further work

8.1. Introduction

This chapter will detail the research objectives before summarising the work that has been carried out. It will highlight the findings of the research and discuss how and where they have been fulfilled. In the process it will examine the novelty of the work and highlight where it has added to the body of knowledge.

8.2. Research objectives

The initial research objectives were to investigate and identify:

1. Whether web development methods are used?
2. Which web development methods are available? What is their scope of coverage? What guidance is available for their use?
3. Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?
4. What components, techniques and tools should constitute a web development method?
5. Whether the Cognitive Dimensions Framework as outlined by Green (1989a) can provide an insight into the assessment of web development methods?

6. Whether it is possible to take the findings of this work and use them to inform the design of a web development method that is suitable for novice web developers?

8.3. Summary of findings

The findings are summarised in response to the Research Questions.

8.3.1. Whether web development methods are used?

By showing that there were plenty of methods available their non-use was not down to non availability. Interestingly, none of the novice developers had heard of any web development methods and had not used one during their industrial placement; and no web developers who were interviewed as part of the fourth study used or had used a development method.

8.3.2. Which web development methods are available? What is their scope of coverage? What guidance is available for their use?

The review of the literature highlighted concern that web development methods are not used. To further explore this issue, a survey of web development methods was carried out, to investigate methods were available. The survey found that there were 52 methods available and they were categorised according to a number of different attributes that would be important to practitioners. These attributes related to scope and coverage of the lifecycle, the type of approach and the focus of the method. The survey was timely, in that there had been no survey done of this

kind and there was little awareness of the number of methods available (Jeary, Phalp and Vincent, 2007b).

8.3.3. Whether the uptake of web development methods is affected by the difficulty of using them in guiding web developers through a web development?

The survey found that the majority of the methods did not cover the whole of the life cycle and a number contained academic techniques that were unlikely to have been known by the average web developer, such as fuzzy logic and Applied Cognitive Task Analysis which was found cumbersome by Masters students given the method by its' author. A case study was designed to investigate web development method utility for a novice developer. Twenty three students from two separate cohorts were given a random selection of three development methods to use in their final year project. The final year project replicated the tight time scale of the web development environment in industry. The students could also, following industrial placements, be classified as novice web developers.

All but one student abandoned the methods they were given and found that they were difficult to use and understand. The methods made assumptions about complex computing techniques and used academic language that novice developers found difficult to understand. In addition to rarely covering the whole lifecycle, the methods did not provide enough guidance about how the method and its techniques should be applied. If the method or its techniques are too difficult or do

not have perceived value then they are likely be abandoned. (It is recognised that in an industrial setting abandoning the method may not be an option).

A further in-depth study was then conducted using three web development methods which appeared in classroom tasks to have some techniques that students found understandable, and that had a large amount of explanatory documentation available. Three students built a simple recruitment web site using one of the methods. Similarly, the students found the methods difficult to use and all spent a long time producing models for which they did not understand the rationale. The documentation was, in all cases, written assuming a level of knowledge none of the students had, and providing little guidance and explanation for activities. One method did not provide definitive guidelines and different guidance was given for the same activity in different papers; in addition the language used in all methods meant that simple concepts were hard to grasp.

8.3.4. What components, techniques and tools should constitute a web development method?

Taking the work from the initial study, where the students selected criteria for what they believed should be in a web development method, two passes were made across a template using template analysis. This resulted in a two section list. The first section listed the 'operational' criteria and the second section listed quality criteria for a web development method. This list was comprehensive. The scope of the method was then adjusted to look specifically at what should be contained within the requirements element of the method. Using classroom discussion over

five years, the contents of the requirements gathering phase of a web development method were refined.

8.3.5. Whether the Cognitive Dimensions Framework provides an insight into the assessment of web development methods?

The Cognitive Dimensions framework was used by students in the evaluation of WSDM, OOHDM and WebML. The students that used the framework selected relevant sections of it to assess their use of the method. Those that used it were able to describe a number of concepts successfully and it was successful as a communication aid. However, of the seven students creating systems in the evaluation phase, it was suggested as a tool to three. None of them used it. In later discussion, one said that it was "*too difficult to use in the timescale*" and a second one used it to describe the system they had built instead of the method. The use of the framework was not successful for describing the PECS method, simply because the method was a framework itself and had no prescribed modelling techniques. Using the PECS method with individual modelling techniques will allow description using the Cognitive Dimensions framework and this could be useful in further work.

8.3.6. Whether it is possible to take the findings of this work and use them to inform the design of a web development method that is suitable for novice web developers?

The Pragmatic, Effective, Common-sense, Simple (PECS) method was created using the results of the four studies that were included in the initial case study and classroom discussions; the focus for this work has been on the requirements framework. This was then evaluated by 45 students who had suggestions as to what should be added and removed from the method. Seven students then used the method to create a web-based system. The overall opinion was that it was a success. It was flexible, easy to use, and that the requirements gathering element was comprehensive and complete with easy to understand guidance that aided communication with clients and stakeholders. This was particularly useful for novice developers who had little experience in this area. There is no framework generally available that will allow a novice developer to consider the elements necessary to be included as part of the requirements for a web development, and the requirements element of the PECS method is both novel and innovative.

8.4. Conduct of the research

8.4.1. Research strategy

The research strategy consisted of using an interpretive, qualitative approach within a case study. The data was collected using documentary evidence and

interviews, and analysed using an analytic inductive process to create a template. Analytic induction was used as an approach in the evaluation process.

8.4.2. Issues with the case study

The case study had few issues. The research design and strategy was clear at the beginning and had few changes. The case study was very suitable for this type of study, which had exploratory elements followed by construction elements and finally an evaluation phase. This could all be conducted within the study and used similar sub-units, the novice developer.

The amount of documentary evidence that was collected was large, and could have been problematical as outlined by Yin(2009). The suggestion by Yin (2009) to create a database of evidence could have been useful however, a methodical approach to filing and the use of mind maps was an approach that suited the author's way of working and proved successful.

8.4.3. Issues with the template analysis

The template analysis was the most successful part of the study. Giving more flexibility than grounded theory to the process, it was particularly useful. A traditional grounded theory approach to this study would have been difficult in that little of the literature should have been studied in advance and the data should be allowed to inform the study. However, to ask the correct questions and to proceed with this study a comprehensive literature review was necessary in

advance and it was felt that template analysis suited this process nicely. The completed template formed the basis of the final PECS method.

8.4.4. Evaluation of the method

The evaluation of development methods is a particularly complex area which has been the subject of much study in the last twenty years. Therefore to ask novice developers (students) to give their opinions about the method was considered both a novel and a useful approach as it combined both ideal method comparison and a generalised measurement tool. This was borne out by the results. The 45 students all made comments, both positive and negative and a number had some very thoughtful insights into the method. The in-depth study by seven students added extra value and showed some interesting results. The requirements section was particularly successful and found useful by all of the students.

8.5. Key Contributions

This work has made an important contribution to the web development community, particularly in the area of requirements engineering for web development. Firstly, the survey categorises 52 web development methods from a practitioners point of view highlighting the problem that many do not cover the early or later stages of the Systems Development Life Cycle (Jeary, Phalp and Vincent, 2007b, Jeary, Phalp and Vincent, 2009)

Secondly, a series of four case studies show how 23 students, who may be classed as novice developers, approached using a web development method and the issues

that arose. The findings highlighted the inaccessibility of current web development methods when all but one of the student developers abandoned the use of the method. A number of factors appear to have contributed to this abandonment; such as the methods were too complicated, required too many products to be created and lacked guidance in terms of the applicability of the method to their specific project. There was little guidance to aid understanding and the language was considered academic and 'intellectual'. The scope did not cover the lifecycle and the requirements phase was particularly commented upon, as being missing. (Jeary, Phalp and Vincent, 2007a, Jeary, Phalp and Vincent, 2009). The fourth case study details the interviews with four web developers in industry and showed that the requirements element of web development has issues. The consensus among the developers was that the customer did not know what they wanted, but the developer did. There was also recognition that the novice developer may need some kind of method to assist them with how to proceed in a web development.

An overview web development method called the PECS (Pragmatic, Effective, Common–sense Simple) method was created from the collected novice developer views which focused particularly on the requirements phase of the development life cycle and resulted in a requirements framework for novice developers (Jeary, Phalp, Xu and deVrieze, 2010). The evaluation of the method shows that the method is easy to use and understand.

The work is of importance to the Software Engineering community because if components of methods or techniques can be identified as difficult to use, then web

development method designers can ensure that the component is either changed or adequate explanation is given for the use of that component within a web development method. It is also important that recognition be given to novice developers and some of the issues they encounter when undertaking their first few developments.

In summary, the novelty and contribution will thus be to have furthered the understanding of the use of web development methods; and the design and production of a web development method and requirements framework which reflects the views of novice web developers. There are no web development methods available which take account of the views and needs of novice web developers and therefore this method is useful as both an academic tool and as an aid to novice developers in industry.

8.6. Further work

The final evaluation showed that the requirements section of the method was easy to use, comprehensive and gave students a communication aid to talk to clients and stakeholders. The structure and other elements of the method, if given the same amount of attention could prove as successful. Further work relates to improving the rest of the method and providing more guidance and documentation to the user.

The use of problem frames and patterns may be useful in helping to make the framework accessible to novices in particular and this is a further element that can be explored.

Chapter 9

References

AL-SALEM, L. S. & SAMAHA, A. A. (2007) Eliciting Web application requirements - an industrial case study. *The Journal of Systems and Software,* 80**,** 294-313.

ALAA, G. & FITZGERALD, G. (2004) A Proposal of an Upfront Requirements Modelling and Design Practice for e-Commerce Projects. *37th Hawaii International Conference on System Sciences.* Los Alamitos, California, IEEE Computer Society.

AROCENA, G. O. & MENDELZON, A. O. (1998) Viewing WISs as Database Applications. *Communications of the ACM,* 41**,** 101-102.

ATZENI, P., MECCA, G. & MERIALDO, P. (1997a) Design and Maintenance of Data-Intensive Web Sites. Roma, Universita degli Studi di Roma Tre.

ATZENI, P., MECCA, G. & MERIALDO, P. (1997b) To Weave the Web. *International Conference on Very Large Databases.* Athens, Greece.

ATZENI, P., MECCA, G. & MERIALDO, P. (1998) Design and Maintenance of Data-Intensive Web Sites. *Conference on Extending Database Technology.* Valencia, Spain.

AVISON, D. & FITZGERALD, G. (2003a) *Information Systems Development: Methodologies, Techniques and Tools,* London, McGraw-Hill.

AVISON, D. & FITZGERALD, G. (2003b) Where Now for Development Methodologies? *Communications of the ACM,* Vol 46**,** 79-81.

AVISON, D. & FITZGERALD, G. (2006) *Information Systems Development: Methodologies, Techniques and Tools,* London, McGraw-Hill.

AVISON, D. & TAYLOR, V. (1997) Information Systems Development Methodologies: A classification according to problem situations. *Journal of Information Technology,* 12**,** 73-81.

AVISON, D. & WOOD-HARPER, A. T. (1990) *Multiview: an exploration in information systems development,* Oxford, Blackwell Scientific Publication.

BALASUBRAMANIAN, V. & BASHIAN, A. (1998) Document Management and Web Technologies: Alice marries the Mad Hatter. *Communications of the ACM,* 41**,** 107-115.

BARRY, C. & LANG, M. (2001a) Contrasting approaches - a study of traditional and multimedia information systems development in Ireland. *Irish Academy of Management.* Derry, Ireland.

BARRY, C. & LANG, M. (2001b) A Survey of Multimedia and Web Development Techniques and Methodology Usage. *IEEE Multimedia,* 8**,** 52-59.

BARRY, C. & LANG, M. (2002) A comparison of 'traditional' and multimedia information systems development practices. *Information and Software Technology,* 45**,** 217-227.

BECK, K. (2003) *Test Driven Development By Example,* Boston, Addison Wesley.

BELBIN, M. R. (1981) *Management Teams,* New York, Addison Wesley.

BICHLER, M. & NUSSER, S. (1996a) Developing Structured WWW-Sites with W3DT. *WebNet 96.* San Francisco, CA.

BICHLER, M. & NUSSER, S. (1996b) SHDT - The structured way of developing WWW-sites. IN COELHO, J. D., KONIG, W., KREMAR, H., O'CALLAGHAN, R. & SAARKSJARVI, M. (Eds.) *Proceedings of the 4th European Conference on Information Systems.* Lisbon, Portugal.

BLACKWELL, A. F., (2001). Human Computer Interaction Lecture Notes: Advanced Graphics and HCI, [online]. Available from: http://www.cl.cam.ac.uk/teaching/2000/AGraphHCI/HCI/hcinotes.html [Accessed: 23 July 2009].

BLACKWELL, A. F., BRITTON, C., COX, A., GREEN, T. R. G., GURR, C., KADODA, G., KUTAR, M. S., LOOMES, M., NEHANIV, C. L., PETRE, M., ROAST, C., ROES, C., WONG, A. & YOUNG, R. M. (2001) Cognitive Dimensions of Notations:Design Tools for Cognitive Technology. *Cognitve Technology 2001.* Springer-Verlag.

BLACKWELL, A. F. & GREEN, T. R. G. (2003) Notational Systems - The Cognitive Dimensions Framework. IN CARROLL, J. (Ed.) *HCI Models, Theories and Frameworks: Toward a multidisciplinary science.* San Fransisco, USA, Morgan Kaufmann.

BLACKWELL, A. F., WHITLEY, K. N., GOOD, J. & PETRE, M. (2001) Cognitive Factors in Programming with Diagrams. *Artificial Intelligence Review,* 15**,** 95-114.

BLEISTEIN, S., COX, K., VERNER, J. & PHALP, K. (2006) Requirements Engineering for e-Business Advantage. *Requirements Engineering Journal,* 11**,** 4-16.

BOCHICCHIO, M. & FIORE, N. (2004) WARP: Web Application Rapid Prototyping. *2004 ACM Symposium on Applied Computing.* Nicosia, Cyprus.

BOCHICCHIO, M. A., PAIANO, R. & PAOLINI, P. (1999) JWeb: an HDM environment for fast development of web applications. *IEEE Conference on Multimedia Computing and Systems.* Florence, Italy, IEEE Computer Society, Los Alamitos, CA.

BOLCHINI, D. & PAOLINI, P. (2004) Goal-driven requirements analysis for hypermedia-intensive Web applications. *Requirements Engineering,* 9**,** 85-103.

BRAY, I. (2002) *An Introduction to Requirements Engineering*, Pearson Education.

BRITTON, C., JONES, S., MYERS, M. & SHARIF, M. (1997) A survey of current practice in the development of multimedia systems. *Information and Software technology,* 39**,** 695-705.

CARDOSO, J., MENDLING, J., NEUMANN, G. & RIEJERS, H. A. (2006) A Discourse on Complexity of Process Models (Survey Paper). IN EDER, J. & DUSTDAR, S. (Eds.) *Business Process Modelling (BPM) 2006 Workshops* Springer - Verlag

CARTENSEN, P. H. & VOGELSANG, L. (2001) Design of Web-based Information systems - New challenges for Systems Development? *9th International Conference on Information Systems.* Bled, Slovenia.

CAUSE, G. (2004) Delivering Real Business Value using FDD. *Methods and Tools.*

CAVAYE, A. L. M. (1996) Case study research: a multi-facted research approach for IS. *Information Systems Journal,* 6**,** 227-242.

CERI, S., FRATERNALI, P., BONGIO, A., BRAMBILLA, M., COMAI, S. & MATERA, M. (2003) *Designing Data-Intensive Web Applications,* Amsterdam, Morgan Kaufmann Publishers.

CERI, S., FRATERNALI, P. & MATERA, M. (2001) WebML Application Frameworks: a Conceptual Tool for Enhancing Design Reuse. *WWW10 International Workshop on Web Engineering*. Hong Kong.

CERI, S., FRATERNALI, P. & MATERA, M. (2002) Conceptual modelling of data-intensive Web applications. *IEEE Internet Computing,* 6**,** 20-30.

CERI, S., FRATERNALI, P., PARABOSCHI, S. & POZZI, G. (1998) Consolidated specification of WWWW intelligent information infrastructure (W3I3). Milano, Politecnico di Milano.

CHECKLAND, P. & SCHOLES, J. (1990) *Soft Systems Methodology in Action,* Chichester, J Wiley and Sons.

CHERRYHOLMES, C. H. (1992) Notes on Pragmatism and Scientific Realism. *Educational Researcher,* 21**,** 13-17.

CHEVALIER, A. & BONNARDEL, N. (2007) Articulation of web site design constraints: Effects of the task and designers' expertise. *Computers in Human Behavior* 23**,** 2455-2472.

CHIU, C.-M., BIEBER, M. & LU, Q. (2002) Towards Integrating Hypermedia on the Web. *35th Hawaii International Conference on System Sciences.* IEEE Computer Society.

CHRISTODOULOU, S., STYLIARAS, G. & PAPATHEODOUROU, T. (1998) Evaluation of hypermedia application development and management systems. *ACM Hypertext 98.* Pittsburgh.

CHRISTODOULOU, S. P., ZAFIRIS, P. A. & PAPATHEODOROU, T. S. (2000) Web Engineering: The Developers View and a Practioners Approach. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering 2000.* Springer-Verlag, Berlin Heidleberg.

COCKBURN, A. (2000) Selecting a Project's Methodology. *IEEE Software,* 17**,** 64-71.

CODA, F., GHEZZI, C., VIGNA, G. & GARZOTTO, F. (1998) Towards a Software Engineering Approach to Web Site Development. *9th International Workshop on Software Specification and Design*. California, IEEE Computer Society.

CONALLEN, J., (1998). Modelling Web Application Design with UML, [online]. Available from: http://www.rational.com/products/whitepapers/100462.jsp [Accessed: 27 June 2003].

CONALLEN, J. (1999a) *Building Web Applications with UML,* Boston, Addison-Wesley.

CONALLEN, J. (1999b) Modelling Web Application Architectures with UML. *Communications of the ACM,* 42**,** 63-70.

COX, K. (2000) Cognitive Dimensions of Use Cases- feedback from a student questionnaire. IN BLACKWELL, A. F. & BILOTTA, E. (Eds.) *Proceedings of the 12th Annual Meeting of the Psychology of Programming Interest Group (PPIG-12).* Cosenza, Italy.

CRESWELL, J. W. (2003) *Research Design: Qualitative, Quantitative and Mixed Methods Approaches,* Thousand Oaks, Sage Publications.

CROTTY, M. J. (2004) *The Foundations of Social Research: Meaning and Perspective in the Research Process* London, SAGE Publishers.

DARKE, P., SHANKS, G. & BROADBENT, M. (1998) Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems Journal,* 8**,** 273-289.

DAVIS, A. M. (2005) *Just Enough Requirements Management: Where Software Development Meets Marketing,* New York, Dorset House.

DAVIS, G. B. (1982) Strategies for information requirements determination. *IBM Systems journal,* 21.

DE LUCA, J., (2009). IT Solutions that make a difference, [online]. Available from: http://www.nebulon.com/index.html [Accessed: 1 October 2009].

DE TROYER, O. M. F. (2001) Audience-Driven Web Design. IN ROSSI, M. & SIAU, K. (Eds.) *Information Modelling in the New Millenium.* IDEA Group Publishing.

DE TROYER, O. M. F. & LEUNE, C. J. (1998) WSDM: A User Centered Design Method for Web Sites. *7th International World Wide Web Conference.* Elsevier: Computer Networks and ISDN Systems.

DEMARCO, T. (1978) *Structured Analysis and Systems Specification,* Englewoood Cliffs, New Jersey, Prentice - Hall.

DENZIN, N. K. & LINCOLN, Y. S. (2005a) The Discipline and Practice of Qualitative Research. IN DENZIN, N. K. & LINCOLN, Y. S. (Eds.) *The SAGEHandbook of Qualitative Research.* 3rd ed. Thousand Oaks, SAGE Publishing.

DENZIN, N. K. & LINCOLN, Y. S. (Eds.) (2005b) *The Sage Handbook of Qualitative Research,* Thousand Oaks, Sage.

DIAZ, P., AEDO, I. & MONTERO, S. (2001) Ariadne, A Development Method for Hypermedia. IN MAYR, H. C. (Ed.) *DEXA, 2001.* Munich, Germany, Springer-Verlag, Berlin.

DSDM CONSORTIUM, (2009). Atern User Manual, [online]. Available from: http://www.dsdm.org [Accessed: 30 June 2010

EICHINGER, C. (2006) Web Application Architectures. IN KAPPEL, G., PROLL, B., REICH, S. & RETSCHITZEGGER, W. (Eds.) *Web Engineering.* English ed. Heidleberg, Germany, John Wiley and Sons.

ERDOGMUS, H., MORISIO, M. & TORCHIANO, M. (2005) On the Effectiveness of Test-first Approach to Programming. *IEEE Transactions on Software Engineering,,* 31.

ERICKSON, J. & SIAU, K. (2007) Theoretical and Practical Complexity of Modeling Methods. *Communications of the ACM,* 50.

ESCALONA, M. J. & ARAGON, G. (2008) NDT. A Model-Driven Approach to Web Requirements. *IEEE Transactions on Software Engineering,* 34**,** 377-390.

ESCALONA, M. J. & KOCH, N. (2004) Requirements Engineering for Web Applications - A Comparative Study. *Journal of Web Engineering,* 2**,** 193-212.

ESCALONA, M. J. & KOCH, N. (2006) Metamodeling the Requirements of Web Systems. *2nd International Conference on Web Information Systems and Technologies.* Setubal, Portugal.

FERNANDEZ, M., FLORESCU, D., KANG, J., LEVY, A. & SUCIU, D. (1997) STRUDEL: A Web Site Management System. *ACM SIGMOD Conference on the Management of Data.* Tucson, Arizona, USA.

FERNANDEZ, M., FLORESCU, D., LEVY, A. & SUCIU, D. (2000) Declarative Specification of Web Sites with Strudel. *VLDB Journal,* 9**,** 38-55.

FITZGERALD, B., RUSSO, N. L. & STOLTERMAN, E. (2002) *Information Systems Development: Methods in Action,* London, McGraw-Hill.

FLICK, U. (2006) *An Introduction to Qualitative Research,* London, SAGE Publishing.

FLYVBJERG, B. (2006) Five Misunderstandings about Case Study Research. *Qulaitative Inquiry,* 12**,** 219-245.

FRATERNALI, P. (1999) Tools and approaches for Developing Data-Intensive Web Applications: A Survey. *ACM Computing Surveys,* No 31.

FRATERNALI, P. & PAOLINI, P. (1998) A Conceptual Model and a Tool Environment for Developing More Scalable, Dynamic, and Customisable Web Applications. *6th International Conference on Extending Database Technology.* Valencia, Spain.

FRATERNALI, P. & PAOLINI, P. (2000) Model driven development of Web Applications: The AutoWeb System. *ACM Transactions on Information Systems,* 18**,** 323-382.

FUCCELLA, J. (1997) Using User Centered Design Methods to Create and Design Usable Web Sites. *Proceedings of 15th International Conference on Computer Documentation.* Salt Lake City, Utah.

GAEDKE, M. & GRAF, G. (2000) Development and Evolution of Web-Applications Using the WebComposition Process Model. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *WebEngineering 2000.* Springer-Verlag, Berlin, Heidleberg.

GALLIERS, R. D. & LAND, F. F. (2002) Choosing Appropriate Information Systems Methodologies. IN MYERS, M. D. & AVISON, D. (Eds.) *Qualitative Research in Information Systems: A Reader.* London, Sage.

GARZOTTO, F., MAINETTI, L. & PAOLINI, P. (1995) Hypermedia Design Analysis, and Evaluation Issues. *Communications of the ACM,* Vol 38**,** 74-86.

GARZOTTO, F., PAOLINI, P. & SCHWABE, D. (1993) HDM -A Model-Based Approach to Hypertext Application design. *ACM Transactions on Information systems,* 11**,** 1-26.

GELLERSON, H. W. & GAEDKE, M. (1999) Object-Oriented Web Application Development. *IEEE Internet Computing,* 3**,** 60-68.

GELLERSON, H. W., WICKE, R. & GAEDKE, M. (1997) WebComposition:An Object-Oriented Support System for the Web Engineering Lifecycle. *Computer Networks and ISDN Systems,* 29**,** 429-437.

GHOSH, G. (1999) Turbo-prototyping: Ultra rapid user centred web development. IN BREWSTER, S., CAWSEY, A. & COCKTON, G. (Eds.) *Human-Computer Interaction INTERACT 99.* Edinburgh, Scotland, IFIP.

GIFFORD, S. S., HENRY, S. M. & SCHOENHOFF, P. K. (2003) Personality and Type Considerations for Programming Teams in Computer Science Classes. *7th World Multiconference on Systems, Cybernetics, and Informatics,.* Orlando, Florida.

GINIGE, A. & MURUGESAN, S. (2001a) The Essence of Web Engineering: Managing the Diversity and Complexity of Web Application Development. *IEEE Multimedia,* 8**,** 22-25.

GINIGE, A. & MURUGESAN, S. (2001b) Web Engineering: An Introduction. *IEEE Multimedia,* 8**,** 14-18.

GLASS, R. (2001) Who's right in the Web Development Debate? *Cutter IT Journal,* 14**,** 6-10.

GLASS, R. L. (2004) Matching Methodology to Problem Domain. *Communications of the ACM,* 47**,** 19-21.

GLESNE, C. & PESHKIN, A. (1992) *Becoming Qualitative Researchers: An Introduction* New York, Longman.

GNAHO, C. (2000) Web-Based Information Systems Development - A User Centered Engineering Approach. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering 2000.* Springer-Verlag, Berlin Heidleberg.

GOESCHKA, K. M. & SCHRANZ, M. W. (2001) Client and Legacy Integration in Object-Oriented Web Engineering. *IEEE Multimedia,* 8**,** 32-41.

GOMEZ, J., CACHERO, C. & PASTOR, O. (2001) Conceptual Modelling of Device-Independent Web Applications. *IEEE Multimedia,* 8**,** 26-39.

GOMM, R., HAMMERSLEY, M. & FOSTER, P. (2007a) Case study and Generalization. IN GOMM, R., HAMMERSLEY, M. & FOSTER, P. (Eds.) *Case Study Method.* Los Angeles, Sage.

GOMM, R., HAMMERSLEY, M. & FOSTER, P. (Eds.) (2007b) *Case Study Method,* Los Angeles, SAGE Publications.

GOODLAND, M. & SLATER, C. (1995) *SSADM v4,* London, McGraw Hill.

GRAY, D. E. (2009) *Doing Research in the Real World,* Los Angeles, Sage.

GREEN, T. R. G. (1989a) Cognitive Dimensions of Notations. IN SUTCLIFFE, A. & MACAULAY, L. A. (Eds.) *People and Computers V.* Cambridge, Cambridge University Press.

GREEN, T. R. G. (1989b) Cognitive dimensions of notations. IN SUTCLIFFE, A. & MACAULAY, L. (Eds.) *People and Computers V.* Cambridge, UK, Cambridge University Press.

GREEN, T. R. G. & PETRE, M. (1996) Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages and Computing,* 7**,** 131-174.

GRIFFITHS, G., HEBBRON, B. D., LOCKYER, M. A. & OATES, B. J. (2002) A Simple Method and Tool for Web Engineering. *SEKE 2002.* Ischia, Italy, ACM.

GRUNBACHER, P. (2006) Requirements Engineering for Web Applications. IN KAPPEL, G., PROLL, B., REICH, S. & RETSCHITZEGGER, W. (Eds.) *Web Engineering: The Discipline of Systematic Development of Web Applications.* Chichester, John Wiley and Sons.

GU, A., HENDERSON-SELLERS, B. & LOWE, D. (2002) Web Modelling Languages: The Gap Between Requirements and Current Exemplars. *Australian World Wide Web Conference.* Queensland, Australia.

GUBA, E. G. & LINCOLN, Y. S. (1994) Competing Paradigms in Qualitative Research. IN DENZIN, N. K. & LINCOLN, Y. S. (Eds.) *Handbook of Qualitative Research,.* Thousand Island, Sage.

GUELL, N., SCHWABE, D. & VILAIN, P. (2000) Modeling Interactions and Navigation in Web Applications. IN LIDDLE, S. W., MAYR, H. C. & THALHEIM, B. (Eds.) *WWW and Conceptual Modelling Workshop.* Salt Lake City, USA, Lecture Notes in Computer Science.

HAKIM, C. (2000) *Research design: successful designs for social and economic research* London, Routledge.

HALLOWS, J. (2005) *Information Systems Project Management: How To Deliver Function and Value In Information Technology projects* New York, Amacom.

HALSTEAD, M. H. (1987) *Elements of Software Science,* Amsterdam, Elsevier.

HATZOPOULOS, M., VAZIRGIANNIS, M. & RIZOS, I. (1993) HADT: Hypermedia Application development Tool for Tourist Applications. *European Journal of Information Systems,* 2**,** 91-100.

HENDERSON-SELLERS, B. (1995) Who needs an object-oriented methodology anyway? *Journal of Object Oriented Programming,* 8**,** 6-8.

HOLCK, J. (2003) 4 Perspectives on Web Information Systems. *36th Hawaii International Conference on System Sciences,.* Hawaii.

HOLCK, J. & CLEMMENSEN, T. (2001) What makes web-development different? Copenhagen, Denmark, Dept of Informatics, Copenhagen Business School.

HOUBEN, G.-J., BARNA, P., FRASINCAR, F. & VDOVJAK, R. (2003) Hera: Development of Semantic Web Information Systems. *International Conference of Web Engineering.* Oviedo, Spain, Springer.

IBM, (2009). Rational Unified Process, [online]. Available from: [Accessed: 15 June 2009 2009].

IEEE (1990) IEEE Standard Glossary of Software Engineering Terminology,. IEEE Computer Society.

ISAKOWITZ, T., STOHR, A. & BALASUBRAMANIAN, E. (1995) RMM: A methodology for structured hypermedia design. *Communications of the ACM,* 38**,** 34-44.

JACKSON, M. (2001) *Problem Frames: Analyzing and structuring software development problems,* Harlow, Addison Wesley/ ACM Press.

JANSSEN, W. & STEEN, M. (2001) Rapid Service Development: An Integral Approach to e-Business Engineering. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering 2000.* Springer-Verlag, Berlin Heidelberg.

JAYARATNA, N. (1994) *Understanding and Evaluating Methodologies: NIMSAD A systemic framework,* London, McGraw Hill.

JEARY, S., PHALP, K. & VINCENT, J. (2007a) Are Web Development Methods Too Difficult to Use? *15th International Software Quality Management Conference/INSPIRE Conference.* Stafford, UK.

JEARY, S., PHALP, K. & VINCENT, J. (2007b) Web Development Methods and Techniques - A Survey. *15th International Software Quality Management Conference/INSPIRE Conference.* Stafford, England.

JEARY, S., PHALP, K. & VINCENT, J. (2009) An evaluation of the utility of web development methods. *Software Quality Journal,* 17**,** 125-150.

JEARY, S., PHALP, K., XU, L. & DEVRIEZE, P. (2010) A requirements framework for novice web developers. *1st Workshop on The Web and Requirements Engineering.* Sydney, Australia.

JOHNSON, R. B. & ONWUEGBUZIE, A. J. (2004) Mixed Methods Research: A research Paradigm Whose Time has Come. *Educational Researcher,* 33**,** 14-26.

KALPSING, R. & NEUMANN, G. (2000) Applying the Resource Description Framework to Web Engineering. *1st International Conference on Electronic Commerce and Web Technologies.* Greenwich, London.

KANYARU, J. M., JEARY, S., COLES, M., PHALP, K. & VINCENT, J. (2008) Assessing Graphical User Interfaces in Modelling Tools for MDA using the Cognitive Dimensions Framework. *16th Intl Software Quality Management Conference,.* Ulster.

KAPPEL, G., PROLL, B., REICH, S. & RETSCHITZEGGER, W. (2006) *Web Engineering*, John Wiley and Sons Ltd.

KING, N. (2004) Using templates in the thematic analysis of text. IN CASSELL, C. & SYMON, G. (Eds.) *Essential Guide to Qualitative Methods in Organizational Research.* London, Sage.

KITCHENHAM, B., BRERETON, O. P., BUDGEN, D., TURNER, M., BAILEY, J. & LINKMAN, S. (2009) Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology,* **,** 7-15.

KITCHENHAM, B., PICKARD, L. & PFLEEGER, S. L. (1995) Case Studies for Method and Tool Evaluation. *IEEE Software,* 12**,** 52-62.

KITCHENHAM, B. A. (1996) DESMET: A method for evaluating Software Engineering methods and tools. Keele, University of Keele.

KITCHENHAM, B. A., DYBA, T. & JORGENSEN, M. (2004) Evidence-based Software Engineering. *26th International Conference on Software Engineering.* Washington DC. USA, IEEE Computer Society.

KOCH, N. (1999) A Comparative Study of Methods for Hypermedia Development. Munchen, Ludwig-Maximilians-Universitat.

KOCH, N. (2000) Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modelling Techniques and Development Process. *Fakultat fur Mathematik und Informatik.* Munchen, Ludwig-Maximilians-Universitat.

KOCH, N. & KRAUS, A. (2002) The Expressive Power of UML-based Web Engineering. IN SCHWABE, D., PASTOR, O., ROSSI, M. & OLSINA, L. (Eds.) *Second International Workshop on Web-Oriented Software Technology.* Valencia, Spain, CYTED.

KOTONYA, G. (1999) Practical expereince with viewpoint-oriented requirements specification. *Requirements Engineering,* 4**,** 115-133.

KRUCHTEN, P. (2000) *The Rational Unified Process: An Introduction,* Boston, Addison-Wesley.

KUHNKE, C., SCHNEEBERGER, J. & TURK, A. (2000) A Schema-Based Approach to Web Engineering. *Thirty-third Hawaii International Conference on System Sciences.* Hawaii, IEEE.

KUSHWAHA, D. S., SINGH, R. K. & MISRA, A. K. (2006) Cognitive Web Based Software Development Process: Towards a more Reliable Approach. *ACM SIGSoft Software Engineering Notes,* 31**,** 1-6.

LANG, M. & FITZGERALD, B. (2005) Hypermedia Systems Development Practices: A survey. *IEEE Software,* 20**,** 68-75.

LANG, M. & FITZGERALD, B. (2007) Web-based systems design: a study of contemporary practices and an explanatory framework based on ''method-in-action''. *Requirements Engineering,* 12**,** 203-220.

LANGE, D. (1994) An Object-Oriented Design Method for Hypermedia Information Systems. *Twenty Seventh Annual Hawaii International Conference on System Sciences.* Maui, Hawaii.

LEE, A. S. (2001) Challenges to Qualitative Researchers in Information Systems. IN TRAUTH, E. M. (Ed.) *Qualitative Research in Information Systems: Issues and Trends.* Hershey, Idea Group Publishing.

LEE, C., SUH, W. & LEE, H. (2004) Implementing a community web site: a scenario-based methodology. *Information and Software Technology,* 46**,** 17-33.

LEE, H., KIM, J., KIM, Y. G. & CHO, S. (1999) A View Based Hypermedia Design Methodology. *Journal of Database Management,* 10**,** 3-13.

LEE, H., LEE, C. & YOO, C. (1998) A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems. *31st Annual Hawaii International Conference on System Sciences.* Maui, Hawaii, IEEE.

LEIGH, A., (2008). Target Audience, [online]. Available from: http://www.shire.net/learnwebdesign/target.htm [Accessed: 1 October 2009].

LIEBERMAN, B. (2003) The Art of Modelling - Part 1:Constructing an analytical framework. *The Rational Edge.*

LINCOLN, Y. S. & GUBA, E. G. (2007) The Only Generalization Is: There is No Generalization. IN GOMM, R., HAMMERSLEY, M. & FOSTER, P. (Eds.) *Case Study Method.* Thousand Oaks, Sage Publishing.

LOWE, D., (1998). Engineering the Web: Web Engineering or Web Gardening?, [online]. Available from: [Accessed: 14 February 2005

LOWE, D. (2003) Web system requirements: an overview. *Requirements Engineering,* 8**,** 102-113.

LOWE, D. & EKLUND, D. (2002) Client Needs and the Design Process in Web Projects. *Journal of Web Engineering,* Vol 1.

LOWE, D., GINIGE, A., SIFER, M. & POTTER, J. (1996) The MATILDA Data Model and its Implications. *Third International Conference on Multimedia Modelling.* Toulouse, France.

LOWE, D. & HALL, W. (1999) *Hypermedia and the Web: An Engineering Approach,* Chichester, John Wiley.

LU, M.-T. & YEUNG, W.-L. (1998) A framework for effective commercial Web application development. *Internet research: Electronic Networking Applications and Policy,* 8**,** 166-173.

MACIASZEK, L. (2007) *Requirements Analysis and System Design,* Harlow, Addison Wesley.

MCCABE, T. J. (1976) A Complexity Measure. *IEEE Transactions on Software Engineering,* SE-2**,** 308-320.

MCCRACKEN, D. D. & WOLFE, R. J. (2004) *User-Centered Website Development: A Human-Computer Interaction Approach,* Upper Saddle River, Pearson Prentice Hall.

MCCRACKEN, G. (1988) *The Long Interview,* Newbury Park, Sage.

MCDONALD, A. & WELLAND, R. (2001a) Agile Web Engineering Process. Glasgow, Department of Computing Science, University of Glasgow.

MCDONALD, A. & WELLAND, R. (2001b) Web Engineering in Practice. *Proceedings of the 4th WWW10 Workshop on Web Engineering*. Hong Kong.

MECCA, G., ATZENI, P., MASCI, A., SINDONI, G. & MERIALDO, P. (1998) The ARANEUS Web-Base Management System. *Sigmod Record,* 27**,** 544-546.

MELDRUM, M. & ROSE, J. (2004) Activity based generation of requirements for web-based information systems: the SSM/ICDT approach. *12th European Conference on Information Systems (ECIS 2004)*. Turku, Finland.

MERIALDO, P., ATZENI, P. & MECCA, G. (2003) Design and Development of Data-Intensive Web Sites: The ARANEUS Approach. *ACM Transactions on Internet Technology,* 3**,** 49-92.

MILITELLO, L. G. & HUTTON, R. J. B. (1998) Applied cognitive task analysis (ACTA): a practitioner's toolkit for understanding cognitive task demands. *Ergonomics,* 41**,** 1618-1641.

MONTERO, S., DIAZ, P. & AEDO, I. (2002) Requirements for Hypermedia Development Methods: A Survey of Outstanding Methods. IN BANKS PIDDUCK, A. (Ed.) *CAiSE 2002*. Springer-Verlag Berlin Heidelberg.

MUMFORD, E. (1983) *Designing Human Systems,* Manchester, Manchester Business School.

MURUGESAN, S., DESHPANDE, Y., HANSEN, S. & GINIGE, A. (2001) Web Engineering: A New Discipline for Development of Web-Based Systems. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering 2000.* Springer Verlag, Berlin Heidleberg.

MURUGESAN, S. & GINIGE, A. (2005) Web Engineering: Introduction and Perspectives. IN GINIGE, A. (Ed.) *Web Engineering*. Idea Group.

NANARD, J. & NANARD, M. (1995) Hypertext Design Environments and the Hypertext Design Process. *Communications of the ACM,* 38**,** 49-56.

NORTON, K. S. (2001) Applying Cross-Functional Evolutionary Methodologies to Web Development. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering 2000.* 2016 ed., Springer-Verlag, Berlin Heidelberg.

NUSEIBEH, B. & EASTERBROOK, S. (2000) Requirements Engineering: A Roadmap. *International Conference on Software Engineering (ICSE).* Limerick, Ireland.

OLLE, T. W., SOL, H. G. & VERRIJN-STUART, A. A. (1986) Information Systems Design Methodologies: Improving the Practice. *Proceedings of the IFIP WG 8.1 Working Conference on Comparative Review of Information Systems Design Methodologies: Improving the Practice.* Noordwijkerhout, The Netherlands, North-Holland.

OLSINA, L. (1997) Applying the Flexible Process Model to build hypermedia products. *Hypertext & Hypermedia: Products, Tools, Methods.* Paris, France, Hermes Editorial.

OLSINA, L., LAFUENTE, G. & ROSSI, G. (2000) Specifying Quality Characteristics and Attributes for Websites. IN MURUGESAN, S. & GINIGE, A. (Eds.) *Web Engineering 2000*. Springer-Verlag, Berlin, Heidleberg.

OOHDM, (2009). OOHDM resources, [online]. Available from: www.telemidia.puc-rio.br/oohdm/oohdm.html [Accessed: 7 September 2009].

OSTERWALDER, A. & PIGNEUR, Y. (2002) An e-Business Model Ontology for Modeling e-Business. *The 15th Bled Electronic Commerce Conference: e-Reality: Constructing the e-Economy.* Bled, Slovenia.

OVERMYER, S. P. (2000) Whats different about Requirements Engineering for Web Sites? *Requirements Engineering,* 5**,** 62-65.

PASTOR, O., FONS, J. & PELECHANO, V. (2003) OOWS: A Method to develop Web Applications from Web-Oriented Conceptual Models. *Third International Workshop on Web-Oriented Software Technologies.* Oviedo, Spain.

PASTOR, O., PELECHANO, V., INSFRAN, E. & GOMEZ, J. (1998) From Object Oriented Conceptual Modelling to Automated Programming in Java. *17th International Conference on Conceptual Modeling.* Singapore, Springer-Verlag, Berlin.

PATTON, M. Q. (2002) *Qualitative Research and Evaluation Methods,* Thousand Oaks, Sage.

PELECHANO, V., FONS, J., MANOLI, A. & PASTOR, O. (2003) Developing Web Applications from Conceptual Models. *Caise 2003.* Springer Verlag, Berlin.

PORTINARI, (2003). The Portinari Project, [online]. Available from: http://www.portinari.org.br/acervoIngles/document.asp?contexto=documento [Accessed: 1 February 2010].

POWELL, T. A., JONES, D. L. & CUTTS, D. C. (1999) *Web Site Engineering: Beyond Web Page Design,* Upper Saddle River, Prentice Hall.

PRESSMAN, R. S. (2000) What a Tangled Web We Weave. *IEEE Software,* 17**,** 18-21.

RETSCHITZEGGER, W. & SCHWINGER, W. (2000) Towards Modeling of DataWeb Applications - A Requirements perspective. *The Americas Conference on Information Systems.* Long Beach, California.

RICHARDSON, K. A., CILLIERS, P. & LISSACK, M. (2001) Complexity Science: A "Gray Science for the "Stuff in Between". *Emergence,* 3**,** 6-18.

RIEMENSCHNEIDER, C. K., HARDGRAVE, W. C. & DAVIS, F. D. (2002) Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models. *IEEE Transactions on Software Engineering,* 28**,** 1135 -1145.

ROSSI, M. & BRINKKEMPER, S. (1996) Complexity metrics for systems development methods and techniques. *Information Systems,* 21**,** 209-227.

RUBIN, H. J. & RUBIN, I. S. (2005) *Qualitative Interviewing: The Art of Hearing Data,* Thousand Oaks, Sage.

RUSSO, N. L. & GRAHAM, B. R. (1998) A first step in developing a web application design methodology: understanding the environment. IN WOOD-HARPER, A. T., JAYARATNA, N. & WOOD, J. R. G. (Eds.) *Sixth International Conference on Information Systems Methodologies.* Salford, Manchester, Springer, London.

SCHARL, A. (2000) *Evolutionary Web Development,* London, Springer-Verlag.

SCHWABE, D. & ROSSI, G. (1995) The Object-Oriented Hypermedia Design Model. *Communications of the ACM,* 38**,** 45-46.

SCHWABE, D. & ROSSI, G. (1998) An Object Oriented Approach to Web-Application Design. *Theory and Practice of Object Systems,* 4.

SCHWABE, D., ROSSI, G. & BARBOSA, S. D. J. (1996) Systematic Hypermedia Application Design with OOHDM. *Hypertext 96.* Washington DC, USA.

SERENA, (2008a). Collage Content Management System, [online]. Available from: http://www.serena.com [Accessed: 1 January 2008].

SERENA, (2008b). Collage no longer supported, [online]. Available from: http://community.serena.com/posts/7f0dac1697 [Accessed: 6 January 2010].

SIAU, K. & CAO, Q. (2003) How complex is the Unified Modelling Language. *Advanced topics in database research*

SIAU, K. & ROSSI, M. (1998) Evaluation of Information Modelling Methods - A Review. *31st Hawaii International Confernce on System Science.* Hawaii, IEEE.

SIAU, K. & WANG, Y. (2006) Cognitive evaluation of information modeling methods. *Information and Software Technology,* 49**,** 455-474.

SILVERMAN, D. (2006) *Doing Qualitative Research,* London, SAGE Publishing.

SOL, H. G. (1983) A Feature Analysis of Information Systems Design Methodologies:methodological Considerations. IN OLLE, W., SOL, H. G. & TULLY, C. (Eds.) *Proceedings of the IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies.* Amsterdam, North-Holland.

SOLTYS, R. & CRAWFORD, A., (Undated). JAD for business plans and designs, [online]. Available from: www.thefacilitator.com [Accessed: 1 June 2005].

STAKE, R. E. (2007) The Case Study Method in Social Inquiry. IN GOMM, R., HAMMERSLEY, M. & FOSTER, P. (Eds.) *Case Study Method.* London, Sage Publications Ltd.

STANDING, C. (2001) The Requirements of Methodologies for Developing Web Applications. *Proceedings of the 9th European Conference on Information Systems.* Bled, Slovenia.

STRAUB, D., GEFEN, D. & BOUDREAU, M.-C., (2004). The ISWorld Quantitative, Positivist Research Methods Website, [online]. Available from: http://www.dstraub.cis.gsu.edu:88/quant/ [Accessed: 12 September 2007].

STRAUSS, A. & CORBIN, J. (1998) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory,* Thousand Oaks, Sage.

TAKAHASHI, K. & LIANG, E. (1997) Analysis and Design of Web-based Information Systems. *6th International World Wide Web Conference.* Santa Clara, California.

TASHAKKORI, A. & TEDDLE, C. (1998) *Mixed Methodology: Combining Qualitative and Quantitative Approaches,* Thousand Oaks, Sage.

TAYLOR, M. J., ENGLAND, D. & GRESTY, D. (2001) Knowledge for Web site development. *Internet research,* 11**,** 451-462.

TAYLOR, M. J., MCWILLIAM, J., FORSYTH, H. & WADE, S. (2002) Methodologies and website development: a survey of practice. *Information and Software Technology,* 44**,** 381-391.

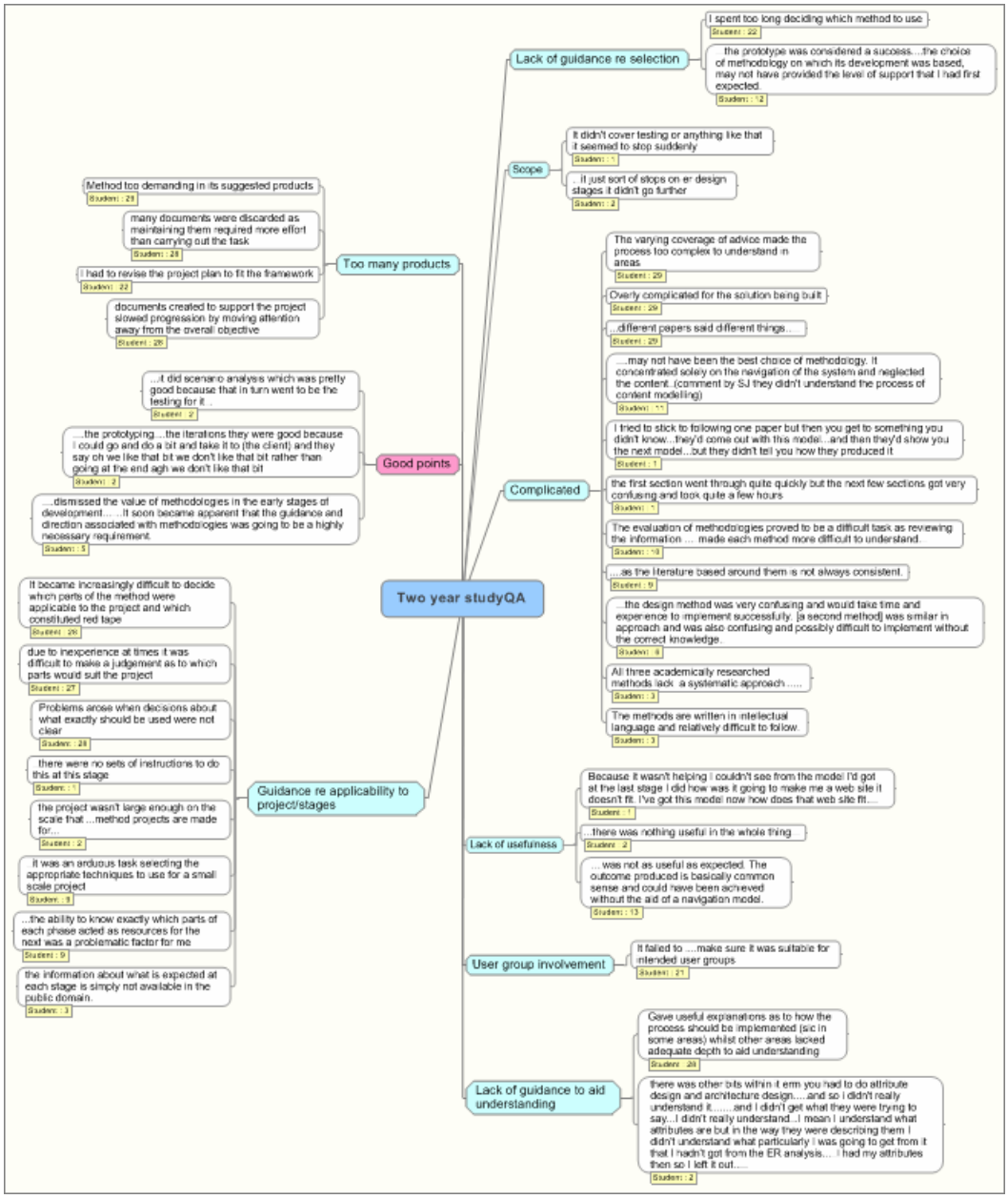
TONGRUNGROJANA, R. & LOWE, D. (2003) WebML+: a Web modeling language for forming a bridge between business modeling and information modeling. *Fifteenth International Conference on Software Engineering and Knowledge Engineering.* San Fransisco.

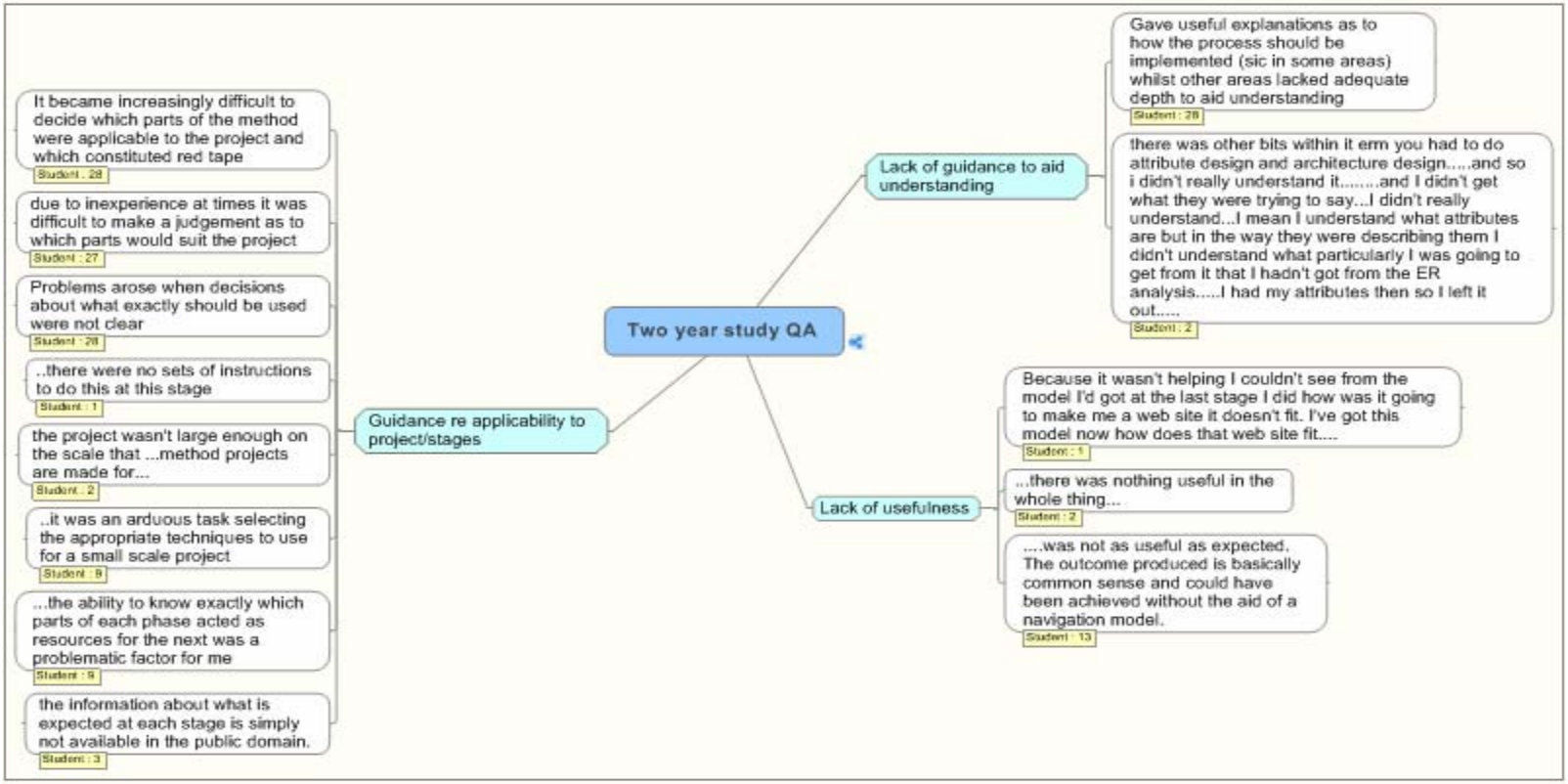
TRAUTH, E. M. (2001) Choosing Qualitative Methods in IS Research: Lessons Learned. IN TRAUTH, E. M. (Ed.) *Qualitative Research in Information Systems: Issues and Trends.* Hershey, Idea Group Publishing.

UDEN, L. (2002) Design Process for Web Applications. *IEEE Multimedia*, 47-55.
VAN LAMSWEERDE, A. (2000) Requirements engineering in the year 00: A research perspective. *22nd International Conference on Software Engineering, ICSE'00.* Limerick, Ireland, IEEE Computer Society Press.
VESSEY, I. & GLASS, R. (1994) Applications-based Methodologies: Development by Application Domain. *Information Systems Management,* 12, 53-56.
VIDGEN, R. (1998) Using the Multiview2 Framework for Internet-Based Information System Development. IN WOOD-HARPER, A. T., JAYARATNA, N. & WOOD, J. R. G. (Eds.) *Sixth International Conference on Information Systems Methodologies.* Salford, Manchester, Springer, London.
VIGDEN, R. (2002) Constructing a web information system development methodology. *Information Systems Journal,* 12, 247-261.
VORA, P. R. (1998) Designing for the Web: A Survey. *interactions.*
WALSHAM, G. (2002) Interpretive Case Studies in IS Research: Nature and Method. IN MYERS, M. D. & AVISON, D. (Eds.) *Qualitative Research in Information Systems: A Reader.* London, Sage.
WEBML, (2009). WebML resources, [online]. Available from: www.webml.org [Accessed: 7 September 2009].
WEED, D. L. (1997) Methodologic Guidelines for Review Papers. *Journal of the National Cancer Institute,* 89.
WHITLEY, E. A. (1998) Method-ism in Practice: Investigating the relationship between method and understanding in web page design. *19th International Conference on Information Systems.* Helsinki, Finland.
WIMMER, M., KEMPER, A. & SELTZSAM, S. (2006) Security for Web Applications. IN KAPPEL, G., PROLL, B., REICH, S. & RETSCHITZEGGER, W. (Eds.) *Web Engineering.* Chichester, John Wiley.
WOLCOTT, H. F. (2001) *Writing Up Qualitative Research,* Thousand Oaks, Sage Publications Inc.
WOOD-HARPER, A. T. & FITZGERALD, G. (1982) A Taxonomy of Current Approaches to Systems Analysis. *The Computer Journal,* 25, 12-16.
WOUKEU, A., CARR, L., WILLS, G. & HALL, W. (2003) Rethinking Web Design Models: Requirements for Addressing the Content. University of Southampton.
YIN, R. K. (2003) *Case Study Research:Design and Methods,* Thousand Oaks, SAGE Publications.
YIN, R. K. (2009) *Case Study Research: Design and Methods,* Los Angeles, Sage.
YOO, J. & BIEBER, M. (2000) Finding Linking Opportunities through Relationship-based Analysis. *Hypermedia & Hypertext.* San Antonio, Texas, USA, ACM Press, New York.
ZOLLER, P. (2001) HMT: Modelling Interactive and Adaptive Hypermedia Applications. IN ROSSI, M. & SIAU, K. (Eds.) *Information Modelling in the New Millenium.* Hershey, IDEA Group Publishing.

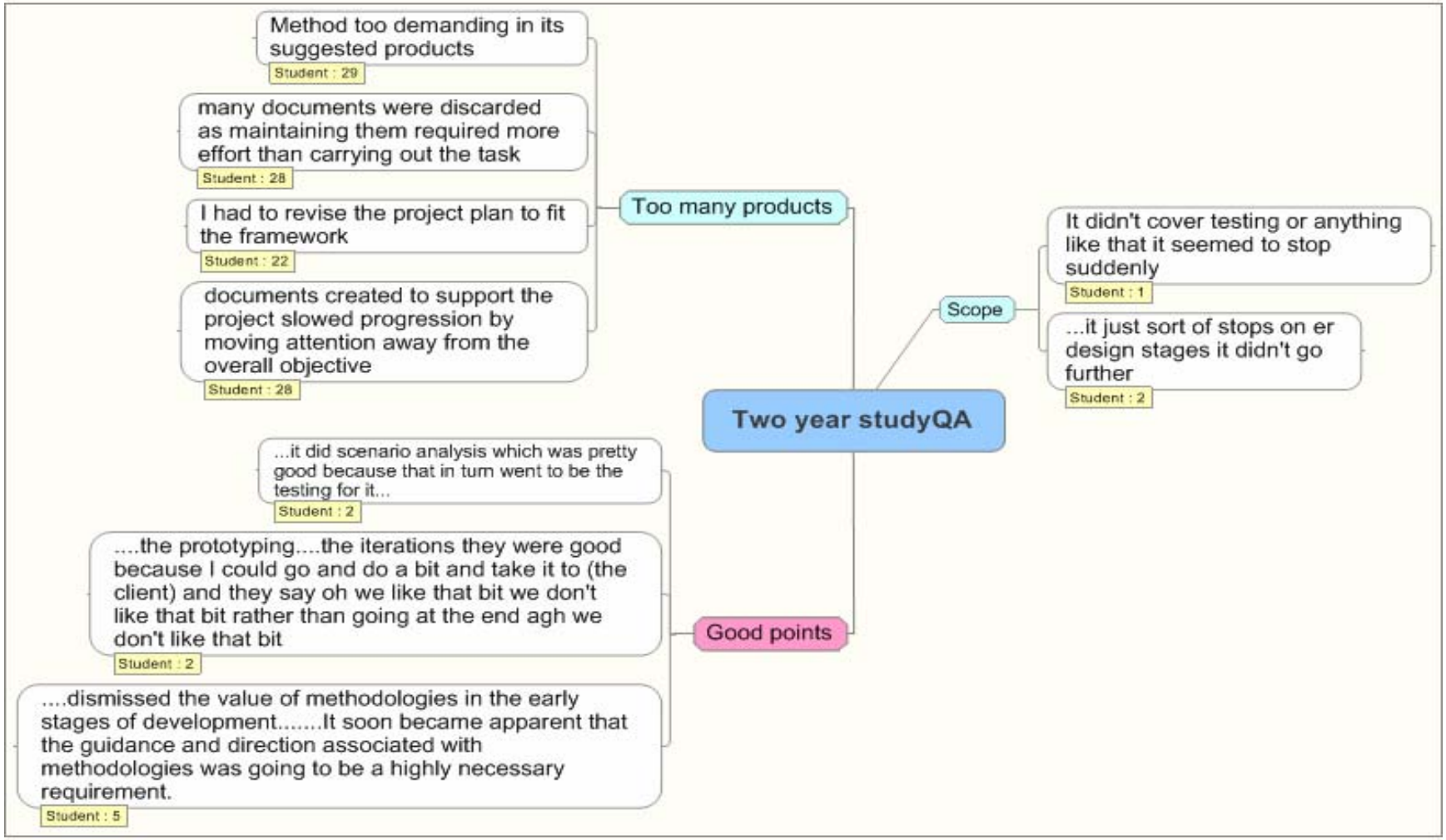
Appendix A

Sample Mind Map from Section 5.2.4 (Overview)

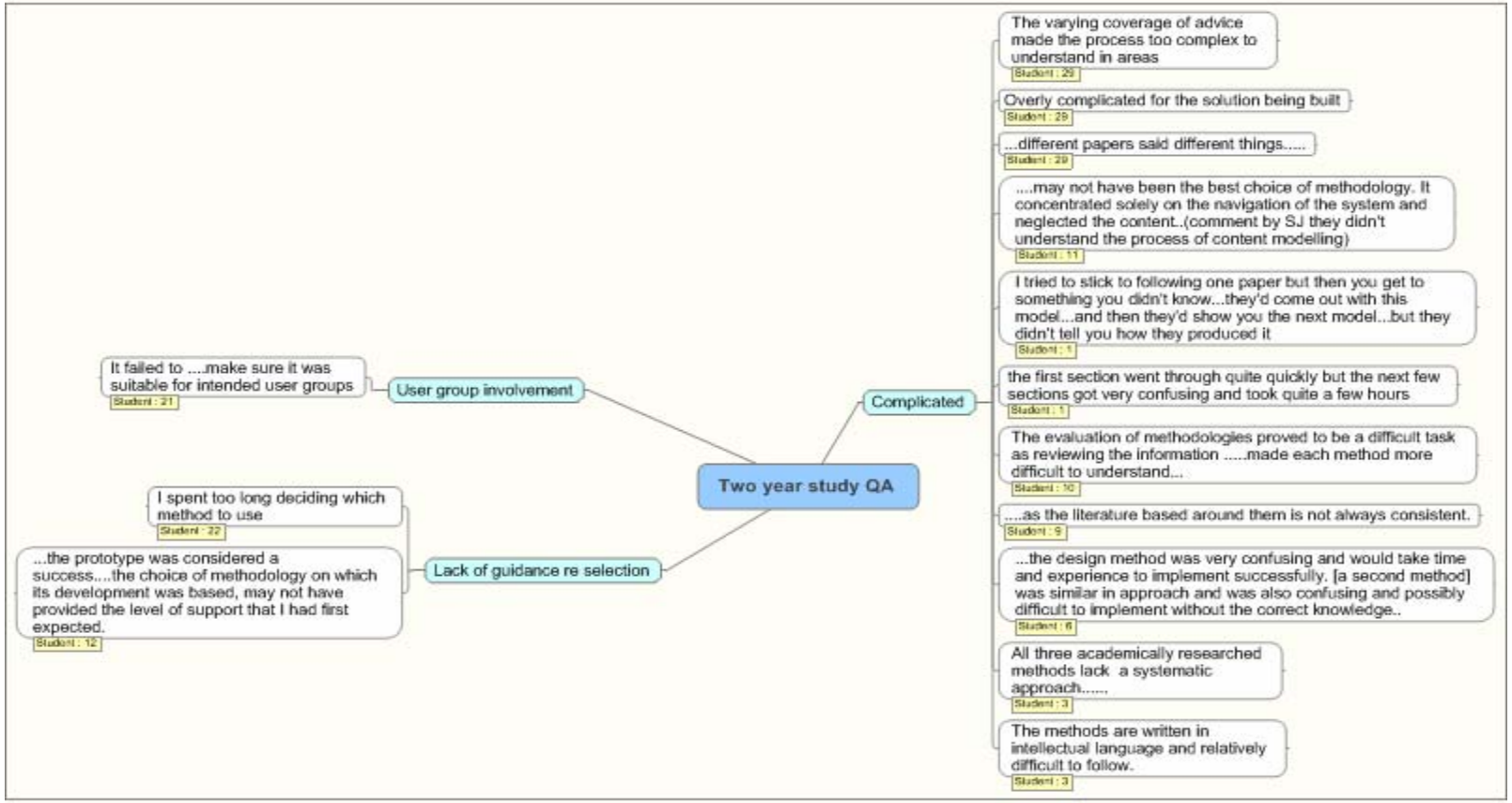




Breakdown 1 of Sample mind map from Section 5.2.4.



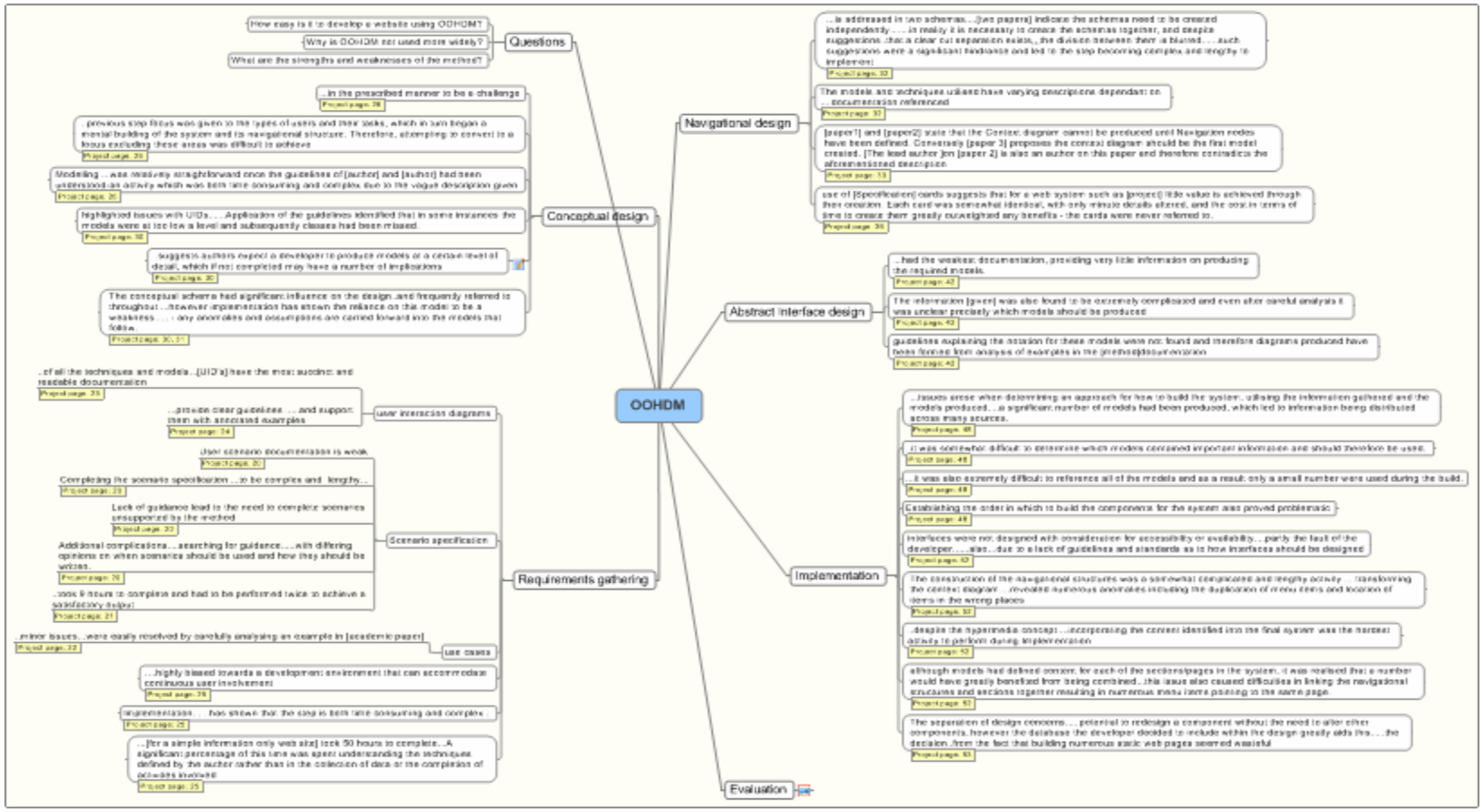
Breakdown 2 of Sample mind map from Section 5.2.4.



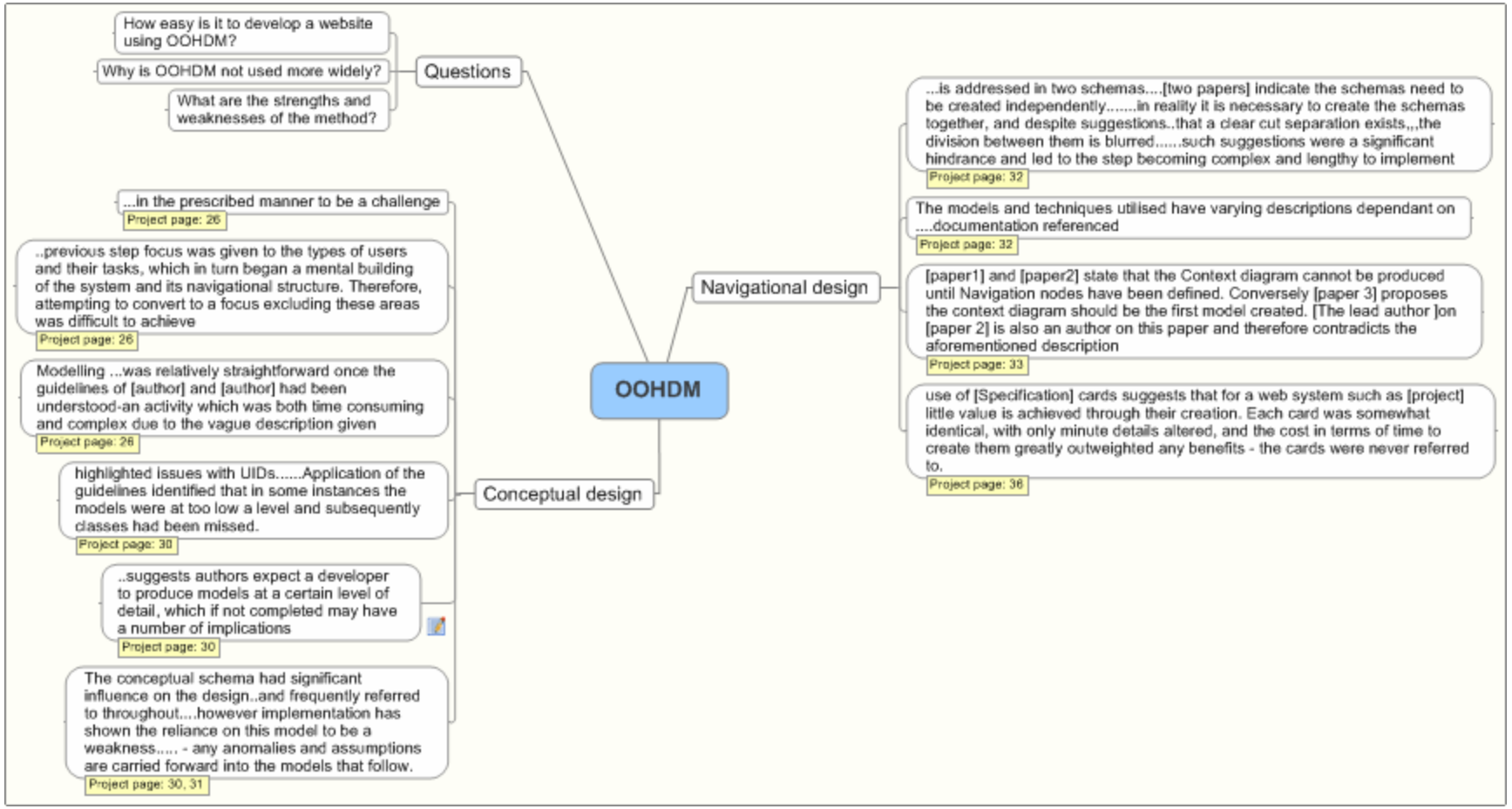
Breakdown 3 of Sample mind map from Section 5.2.4

Appendix B

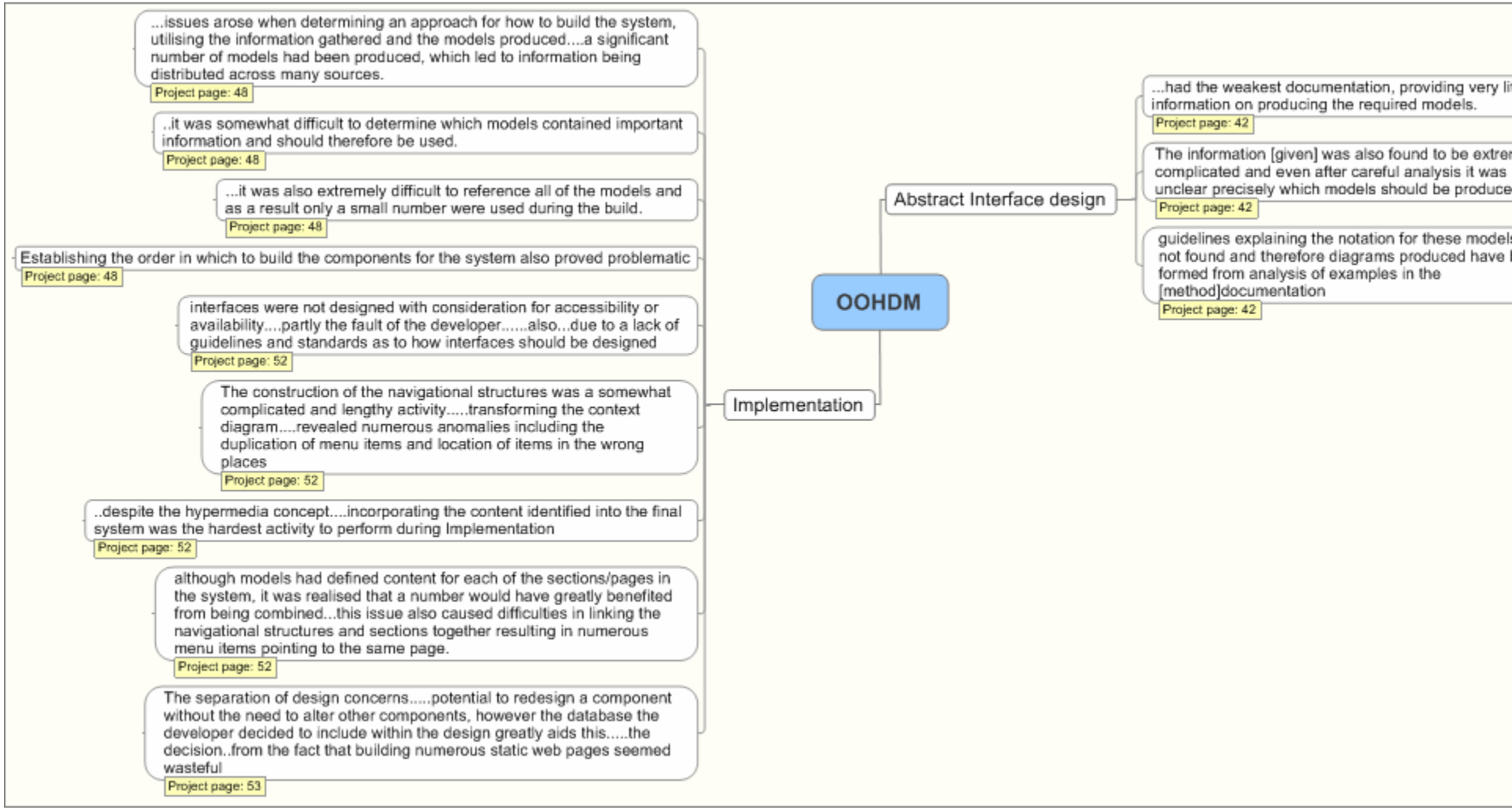
Sample Mind Maps from Section 5.4



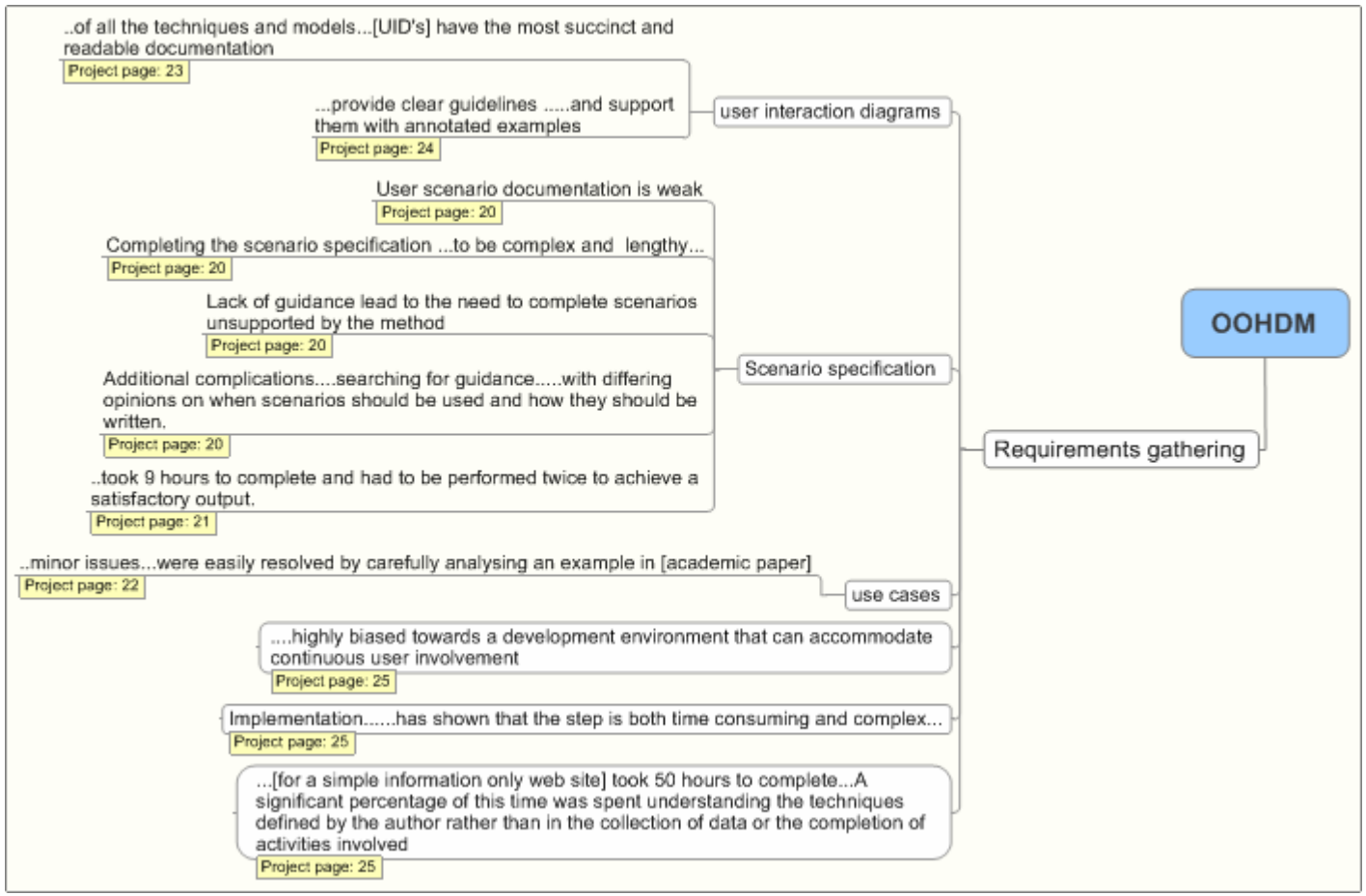
OOHDM Overview evaluation mind map described in section 5.4.6.



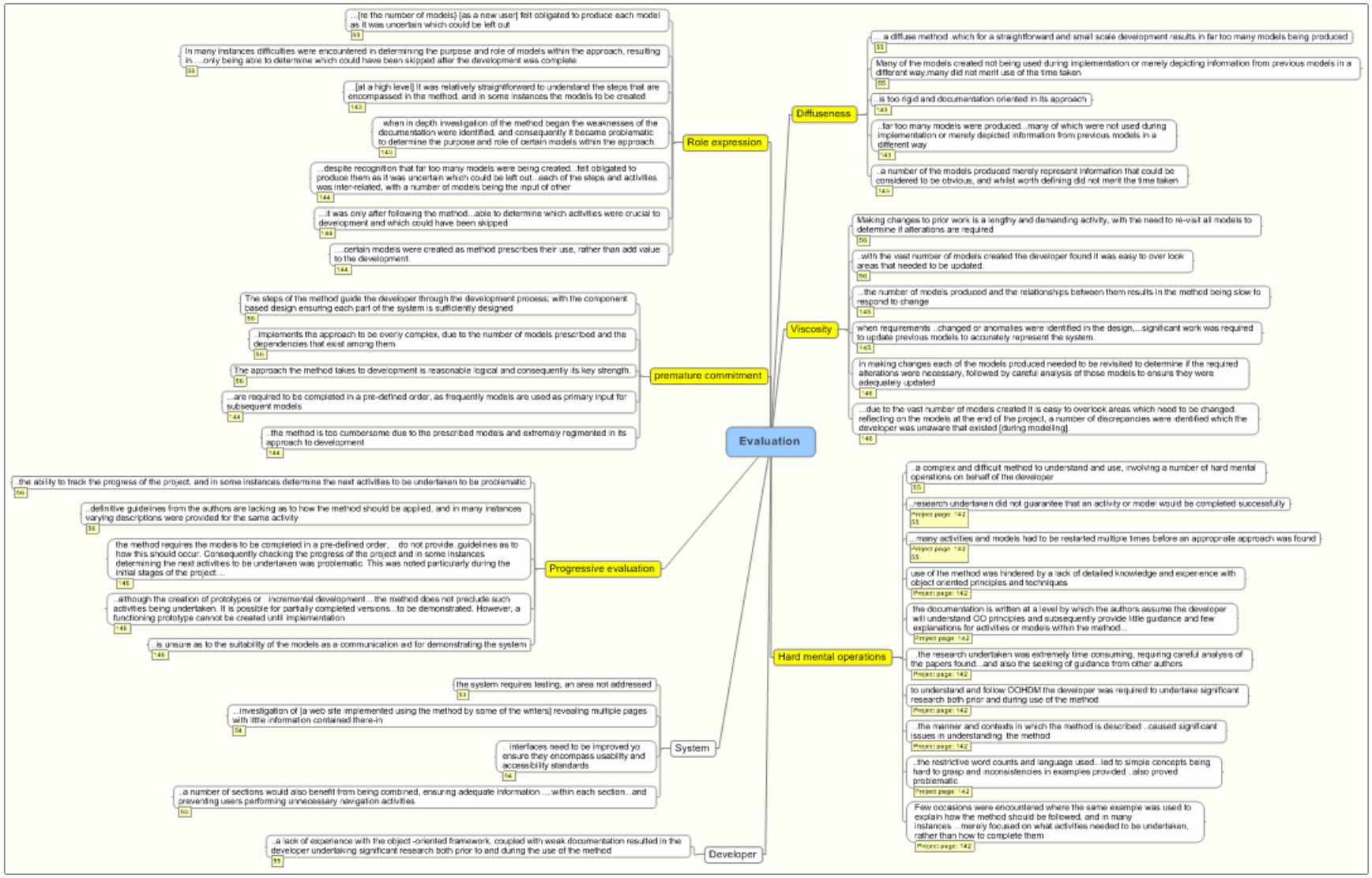
Breakdown 1 of OOHDM mind map described in section 5.4.6.



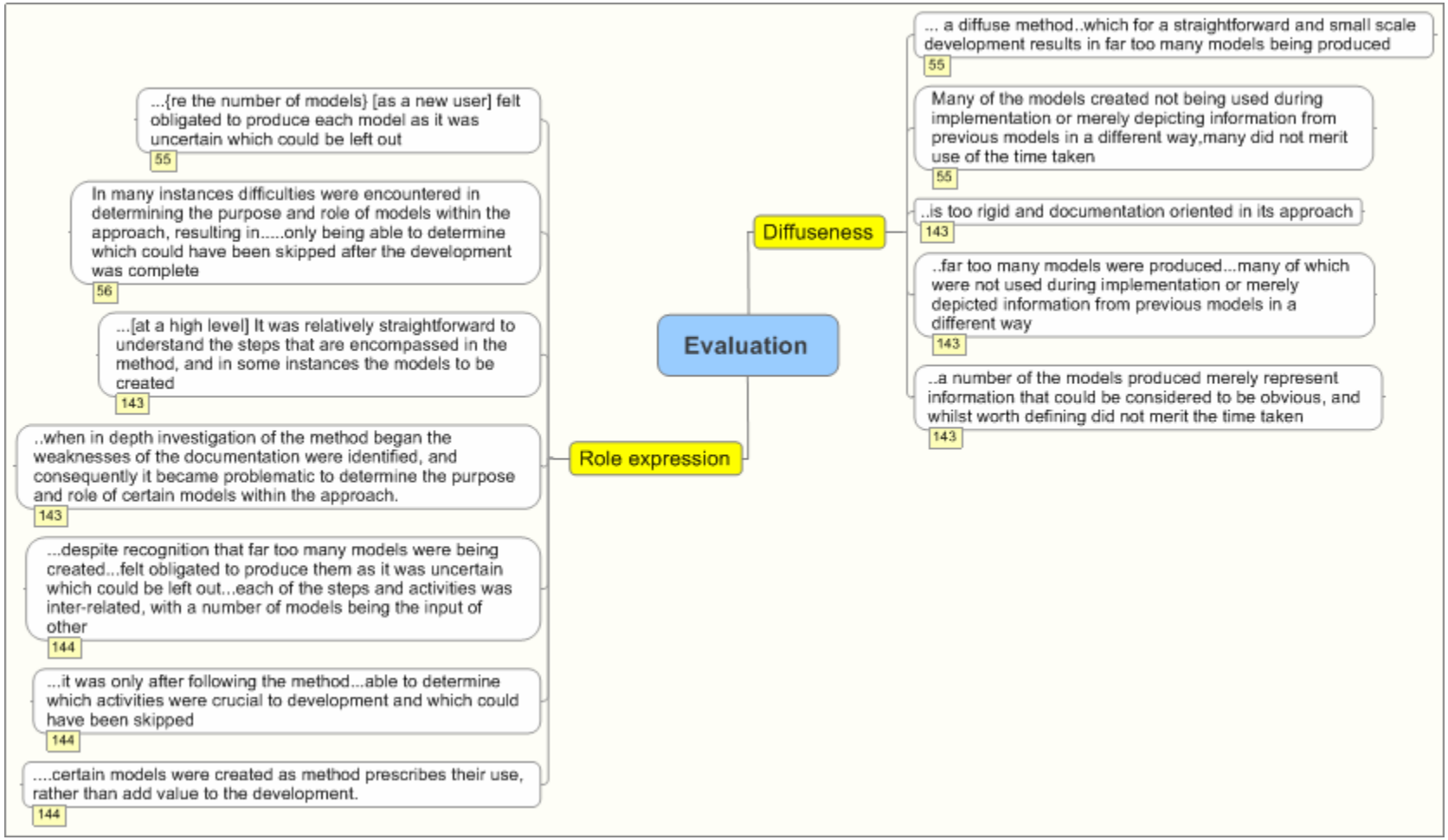
Breakdown 2 of OOHDM mind map described in section 5.4.6.



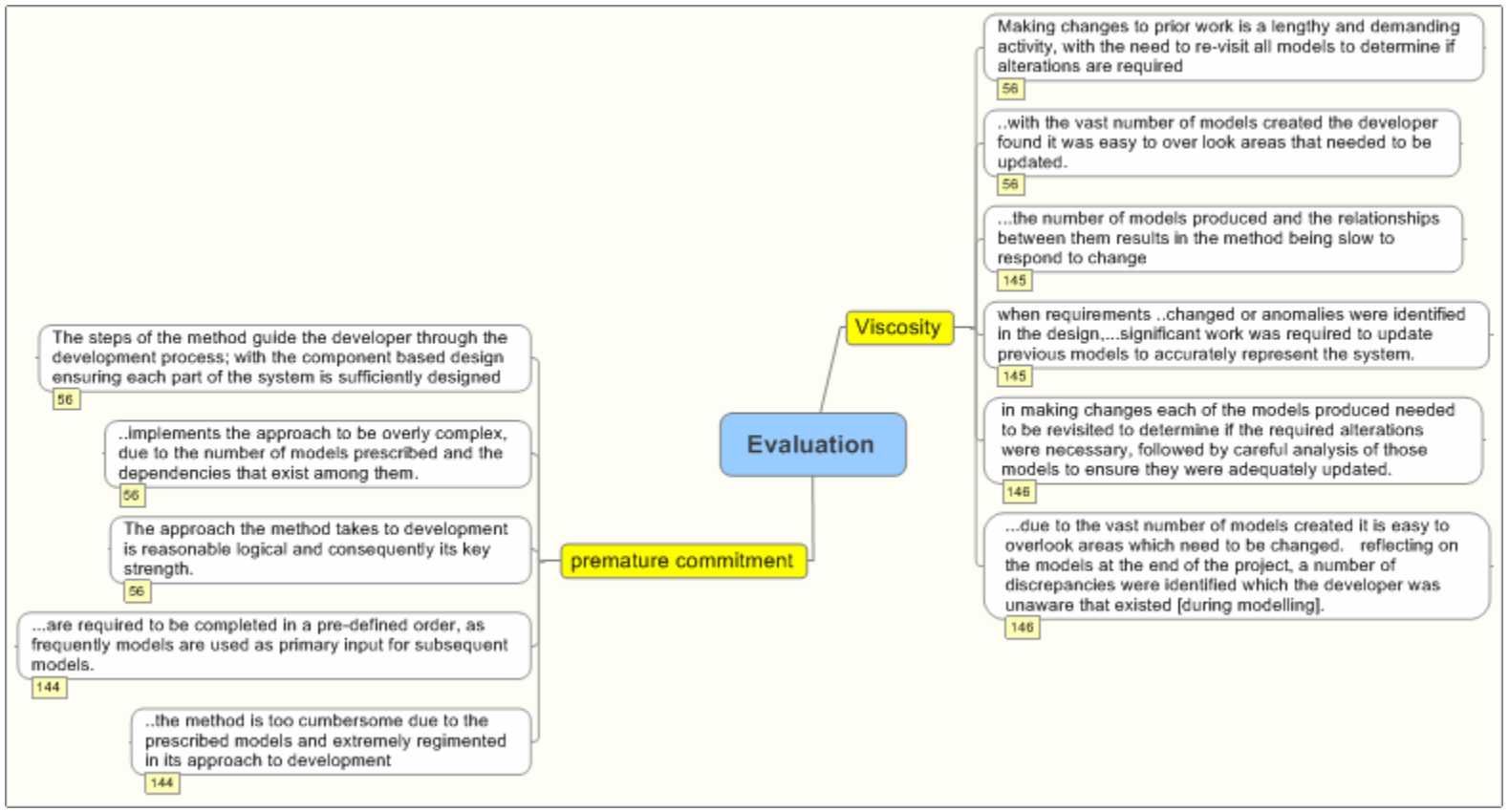
Breakdown 3 of OOHDM mind map described in section 5.4.6.



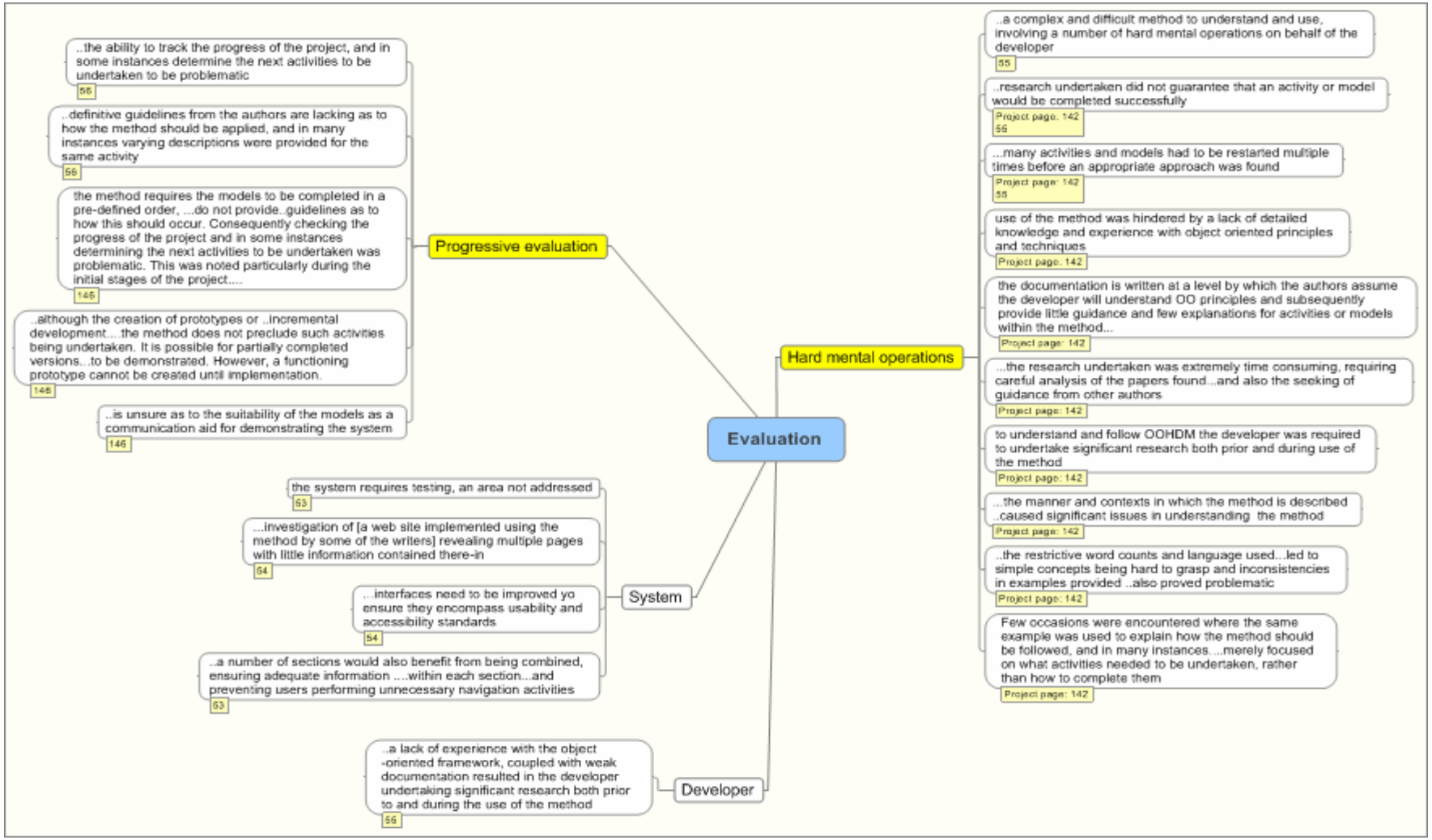
OOHDM Cognitive Dimensions evaluation mind map described in section 5.4.7.



Breakdown 1 of OOHDM Cognitive Dimensions evaluation mind map described in section 5.4.7.



Breakdown 2 of OOHDM Cognitive Dimensions evaluation mind map described in section 5.4.7.



Breakdown 3 of OOHDM Cognitive Dimensions evaluation mind map described in section 5.4.7.

Appendix C

Outline interview questions

Interviewee qualifications and experience

What is your age?

What qualifications do you have? In the area of computer science or web development?

Current position?

Length of time in present position?

Previous experience? Previous experience in web development and software development

If a manager, consider team and information about the team.

How many in the team? Qualifications and experience of team members?

Development process

Are you aware of any development methods?

Are you aware of any web development methods?

Do you use a development process? Where does this process start?

Describe a development through from a customer request to implementation.

Do you have a requirements process?

Getting from requirements to design – how does that happen?

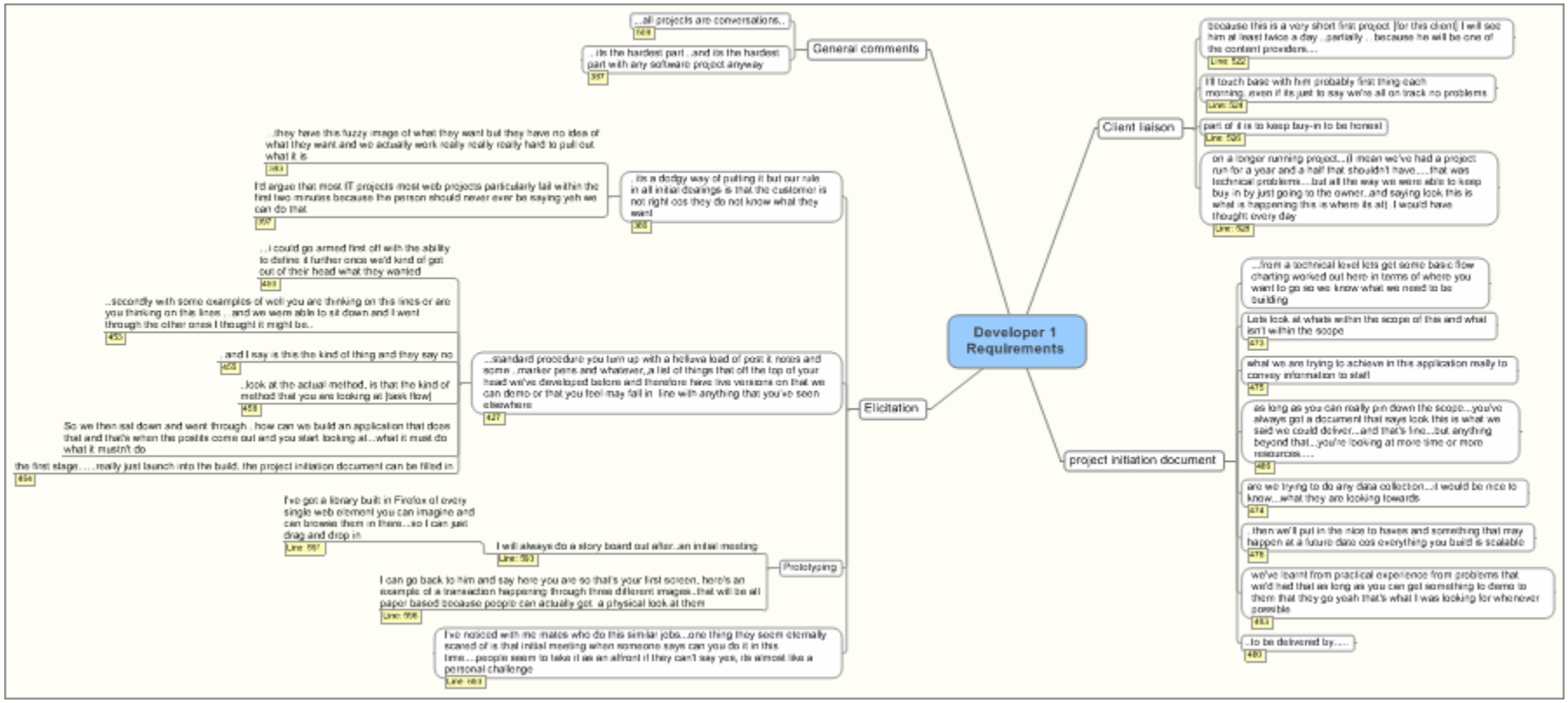
Do you use any models?

What happens with the data? Do you have a database expert?

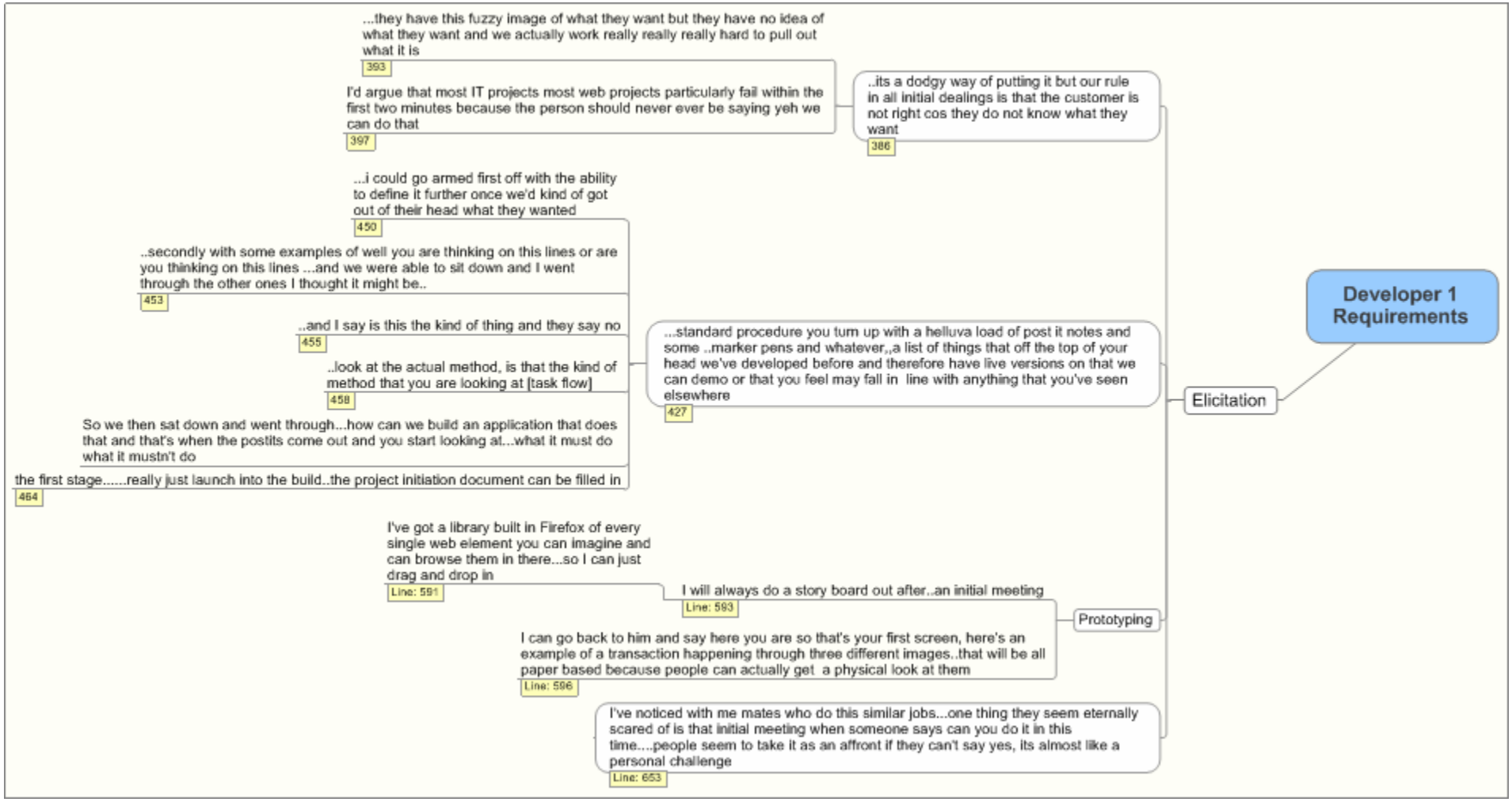
How do you evaluate what you do?

Appendix D

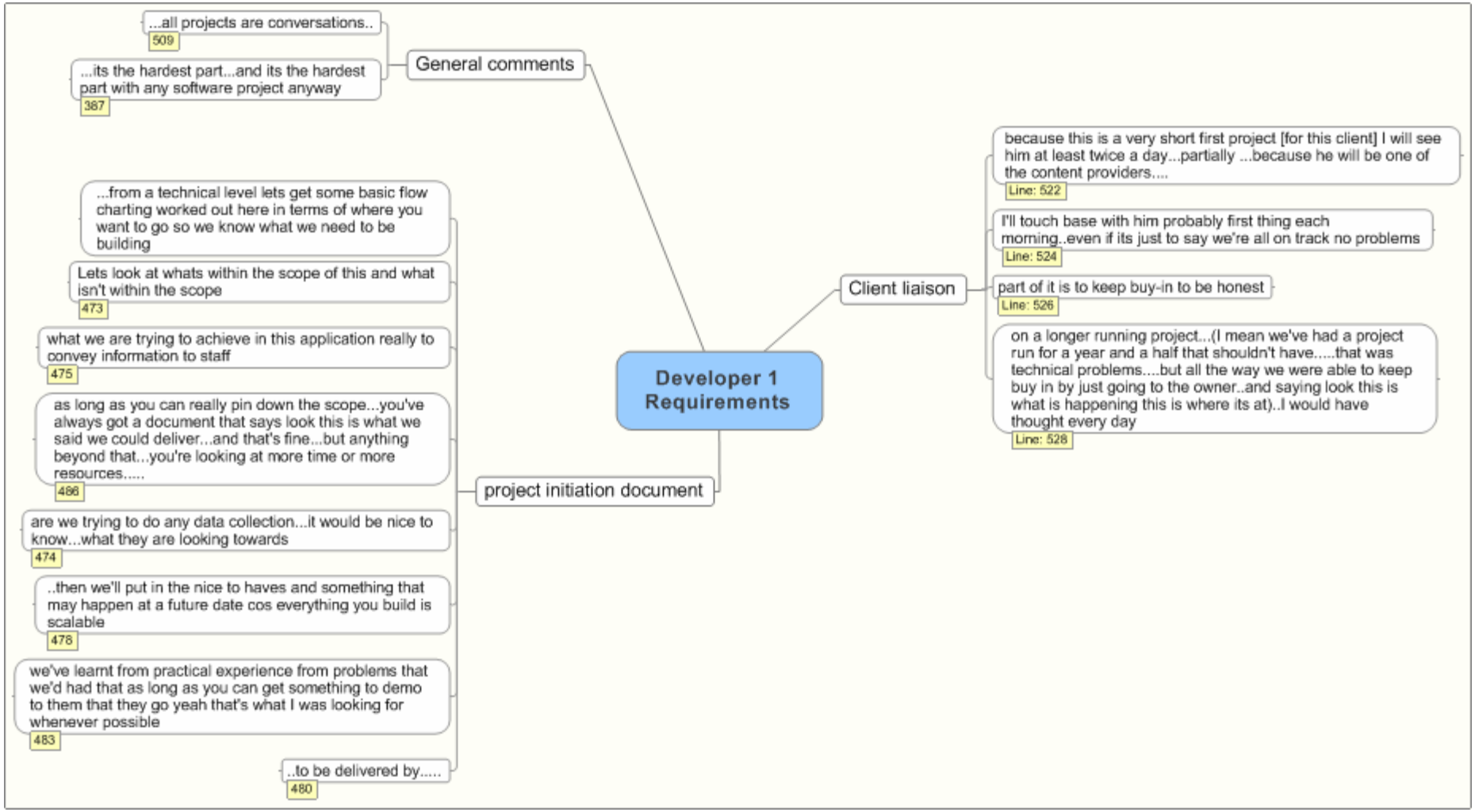
Sample mind maps from Section 5.5.



Overview mind map of Web1 interview - requirements section



Breakdown 1 of Web1 interview - requirements section



Breakdown 1 of Web1 interview - requirements section

Appendix E

School of Design, Engineering and Computing

CONTEXT		ASSIGNMENT	
Course	**Software Systems Framework**	**Title**	**Requirements Methods for the Web**
Level	**H**	**Assignment N^{o}**	**1b**
Unit Title	**Web Systems**	**Issued to students**	**21/10/2008**
Marker	**Sheridan Jeary**	**Assignment Hand in 1a**	**21/10/2008**
		1b	**24/11/2008**

This is an individual assignment

A satisfactory pass should be obtainable for about **25 hours** work

Web Systems Assignment

Background:
In the lecture you completed a first cut attempt at identifying the requirement categories that you thought you would need in a web development. This has been returned to you.
For reference the task was:

''You are a web developer for a web development company that specialises in providing web applications for the travel industry and you are working on a new project with two others. The skills the three of you have are:- programmer, graphic designer,and web developer, although the exact job boundaries of each job are irrelevant for the purposes of this exercise.

This project requires you to produce a portal. The portal is for a travel company called 'Your golf break' that specialises in golfing holidays. It has decided to produce a portal to display all the holidays which it covers, 50 per cent of which are sourced elsewhere. It would also like to show hotels, airlines and golf clubs that it features so that customers are able to visit a one stop shop. The portal will include the display and storage of information and some financial transactions when customers decide they wish to book online.In addition the client requires a number of additions to the interface that will attract investment into his company.

Your company director has asked you to provide a detailed set of requirements to enable him to:

a. manage the project and to allocate work
b ensure the final product meets the requirements
c. make a more accurate estimate for the cost of the work.

The assignment:
You are required to take the requirements categories that you identified and the requirements categories that have been presented to you in the lecture and reflect on them. You need to answer a number of broad questions:

1. What should be added to **your** requirements categories and why?
2. What would you remove from **your** requirements categories and why?
3. What should be added to **Sherry's** requirements categories and why?
4. What should be removed from **Sherry's** requirements categories and why?
5. Is there a 'perfect set' of requirements categories? identify what the list should have in it and explain your reasoning.

Remember you should consider all sets of requirements categories in terms of the proposed development. No marks will be awarded for the initial requirements list you made, however your reflection on it will be marked.

SUBMISSION FORMAT
You should produce a well researched essay of 2500 words which answers the five questions presented above.

MARKING CRITERIA
Evidence of research and reference list: 20%
Reflection on your original categories: 20%
Reflection on Sherry's categories: 30%
Reflection and comment on a 'super set': 30%

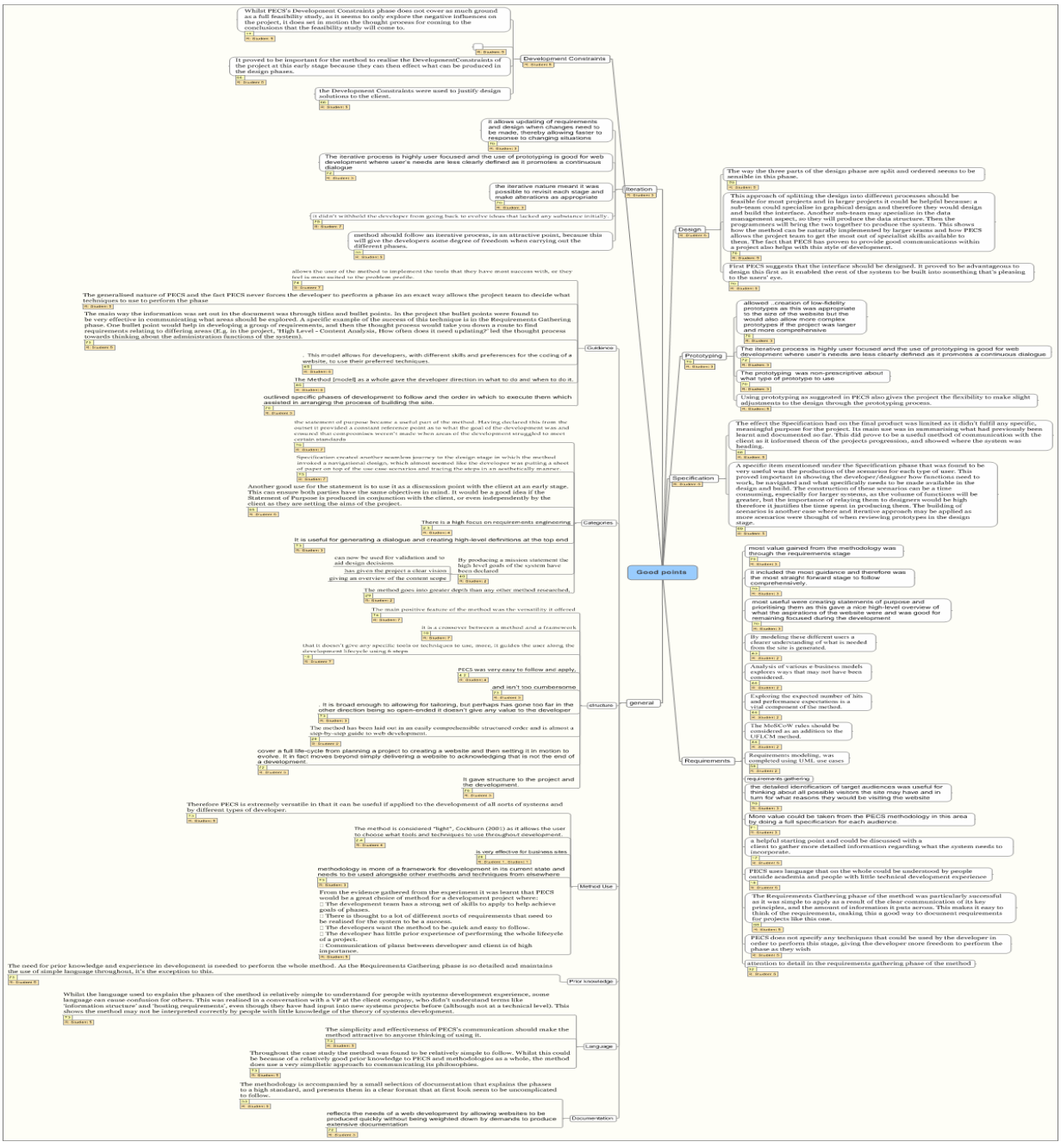
LEARNING OUTCOMES

1. Contrast and compare the selection of different web development methods and evaluate critically web development techniques as opposed to traditional software approaches and techniques.
2. Evaluate critically the impact upon the organisation, management, employees and stakeholders of adopting or including different e-Business models
3. To understand and evaluate critically the effects of Web 2.0 upon an organization.
4. To understand the legal, ethical and social issues deriving from e-Commerce and e-Business operations.
5. To understand the relationship among e-business processes, requirements and modern enterprise development.

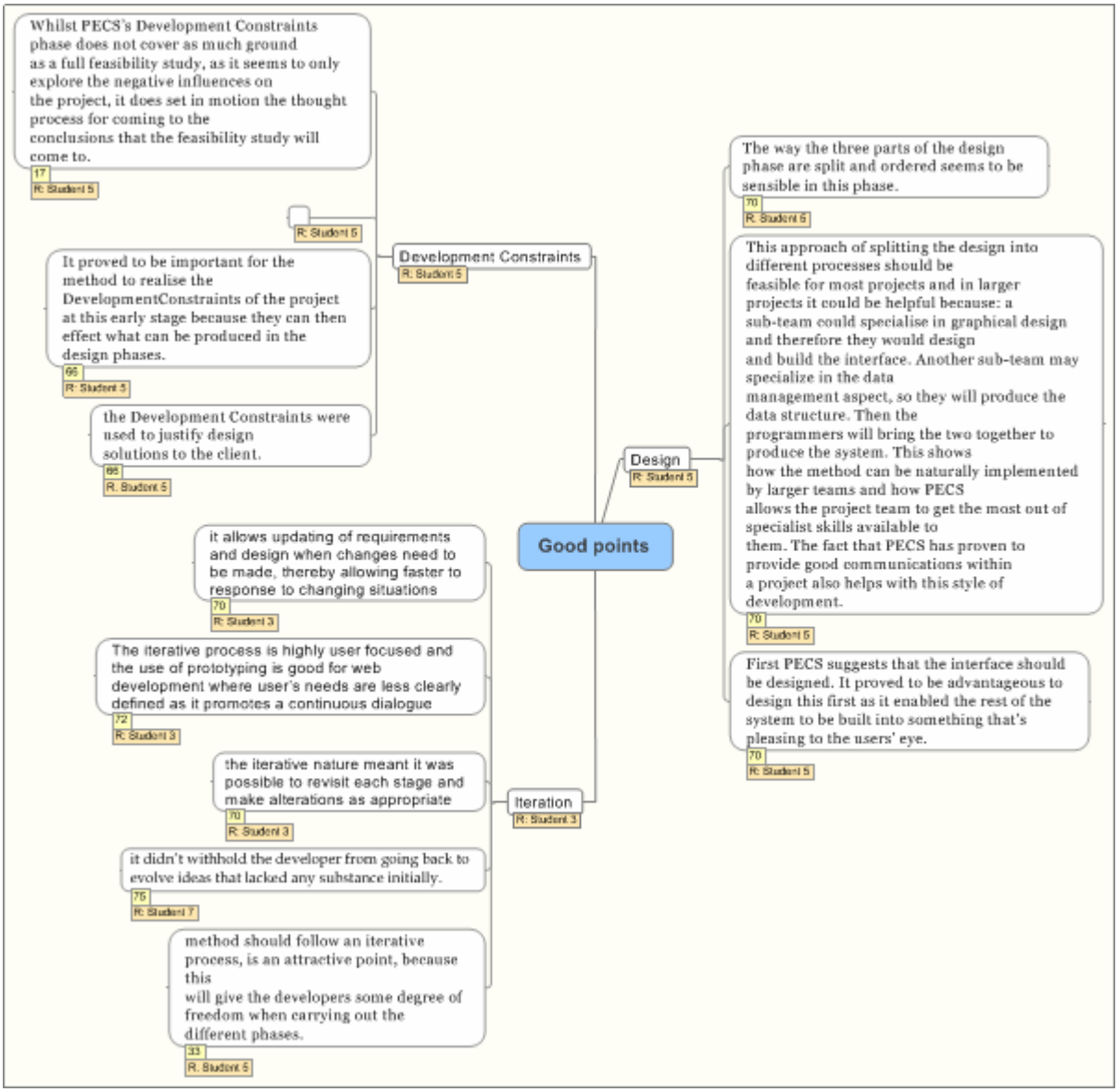
Signature Marker _____
Sheridan Jeary
Signature Quality Assurer _____
Keith Phalp

Appendix F

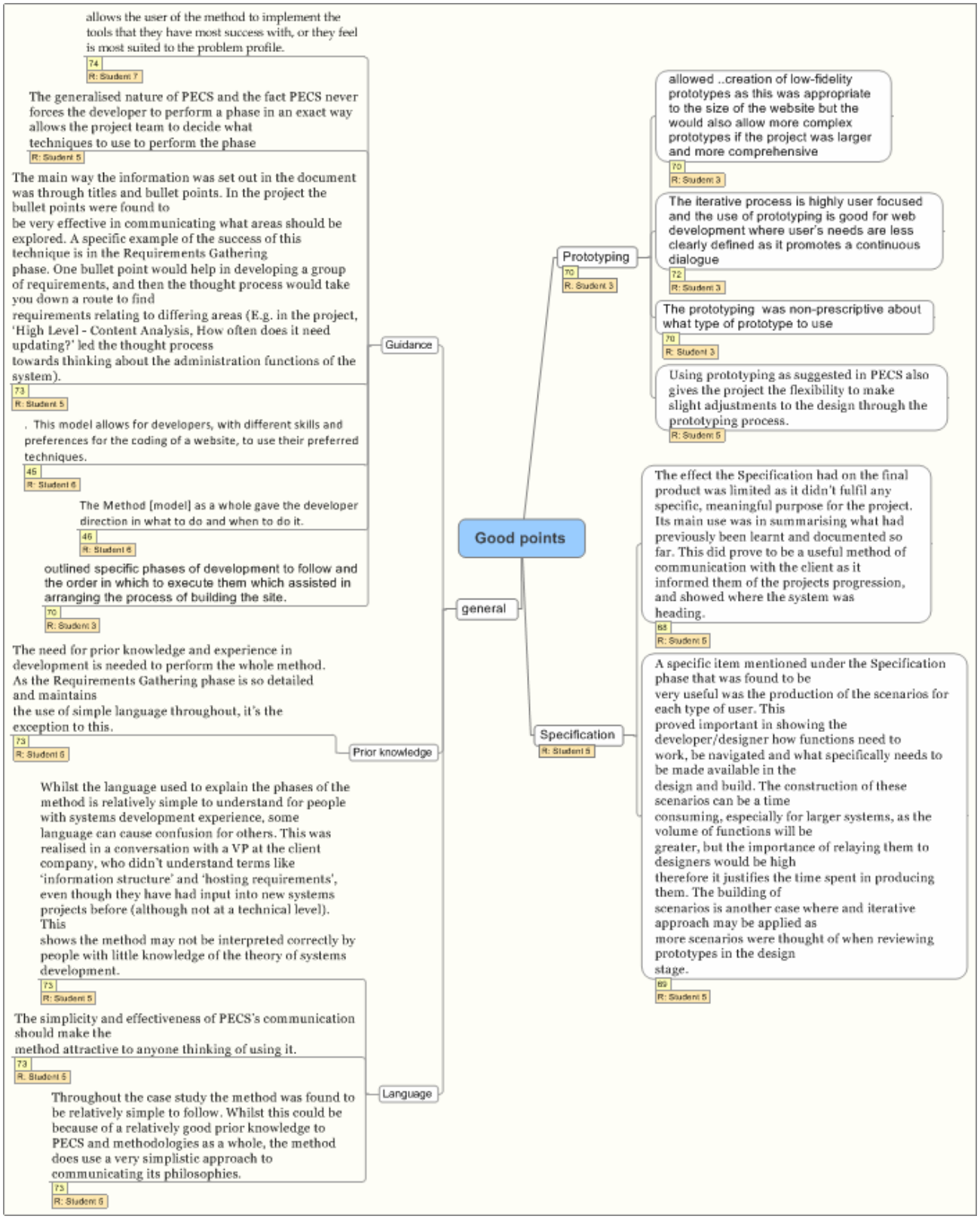
Sample Mind Map from Section 7.5



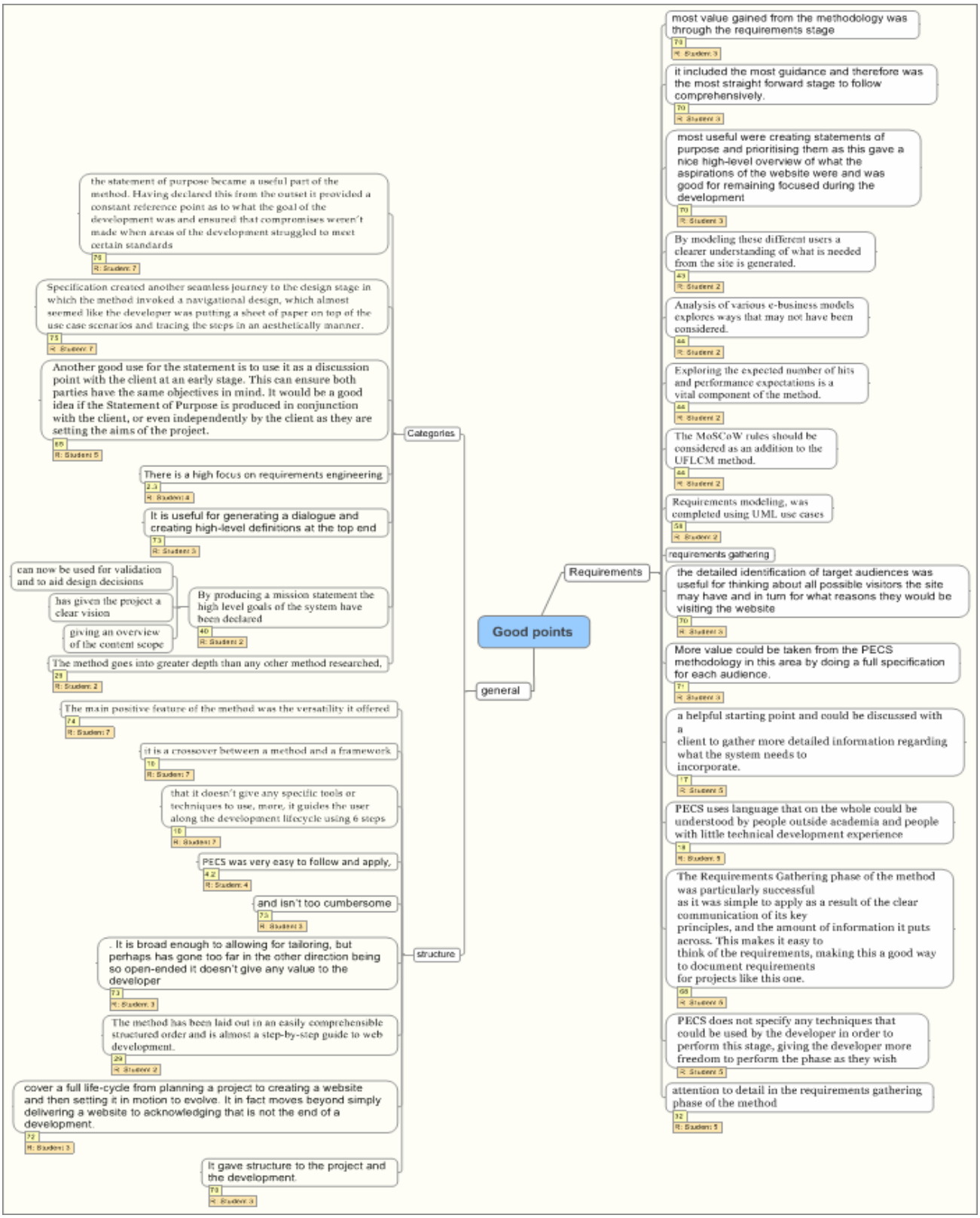
Overview 'good points' mind map from Section 7.5



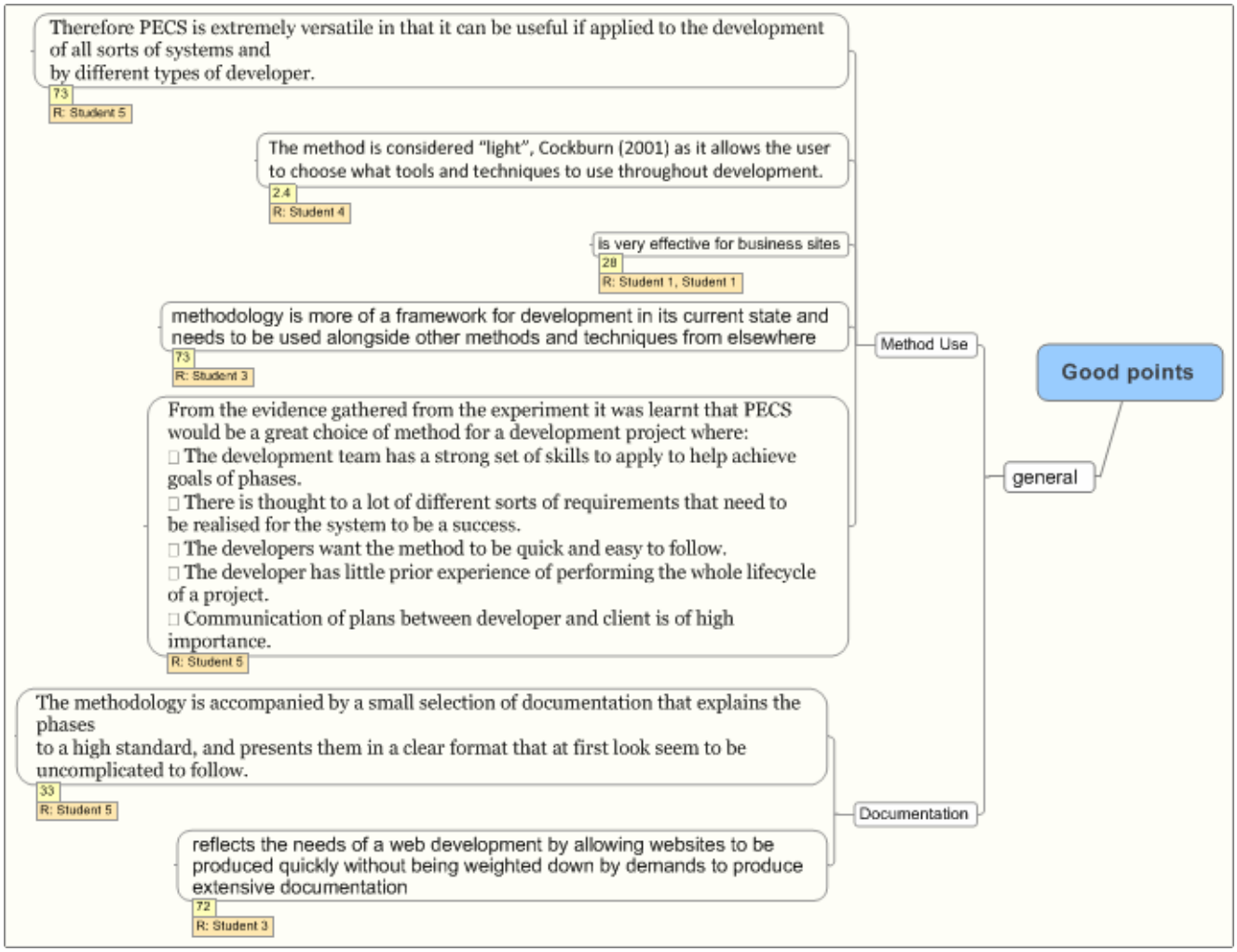
Breakdown 1 'good points' mind map from Section 7.5



Breakdown 2 'good points' mind map from Section 7.5



Breakdown 3 'good points' mind map from Section 7.5



Breakdown 4 'good points' mind map from Section 7.5

Appendix G

G.1. The initial requirements framework

Requirements Gathering

8. What is the site for?
    a. To provide a community
    b. To increase company visibility
    c. To provide a new business model
    d. To provide information
    e. To make money
9. Who are the target audience?
    a. How will you find this out? What problems are there to doing this?
    b. What are the expected types? What are the requirements of each type?
    c. Who are the stakeholder audience?
10. What is the business model?
11. High Level Content Analysis
    a. Who is providing the content?
    b. Marketing analysis and planning
        i. What are the expected visitor numbers and types
        ii. Product releases etc
    c. Who owns it?
    d. What about copyright?
    e. How often does it need updating?
    f. Who will update it?
    g. Consider privacy, accuracy, property, accessibility
12. Security implications?
    a. For different revenue models
    b. For different audience types
    c. Sensitive information
13. Timescale
    a. What is required by when?
14. Technical issues
    a. Hosting requirements
    b. Development environment
    c. Language
    d. Hardware

Specification

3. Identify audience types using:
    a. Statement of purpose
    b. Market research
    c. High level target audience identification
4. Write scenarios for each audience type
    a. From scenarios identify information requirements
    b. From scenarios identify functions they will need to complete

3. Identify audience structure linking similar information and functions

G.2. The initial outline method

1. Project planning
    - Is this part of the development method or separate?
2. Statement of purpose
    - High level
    - Who are the target audience?
3. Development Constraints
    - Staffing
    - Availability
    - Skills and experience
    - Both client company and development team
4. Requirements Gathering
5. Specification
6. Design
    - Information
        - Storage medium
        - Structure
    - Navigation
    - Presentation

G.3. Final requirements framework after evaluation

Method Phase	Discussion and see Section
1. Create a timeline for the project What is required by when? When is the final product required?	Moved from Phase 7. 7.4.1.1 7.4.1.5
2. Consider development constraints? a. Staffing b. Availability c. Skills and experience d. Both client company and development team	Moved from method outline 7.4.1.4
3. What is the site for? a. To provide a community b. To increase company visibility c. To provide a new business model d. To provide information e. To make money	Repeated at Phase 8 7.4.1.2
4. Who are the target audience? a. How will you find this out? b. What problems are there to doing this? c. What are the expected types? d. What are the requirements of each type? e. Who are the target audience? f. Who are the stakeholders?	Repeated at Phase 8 Stakeholder and audience definition separated 7.4.1.3
5 What is the business model?	
6. High Level Content Analysis a. Who is providing the content? b. Marketing analysis and planning i. What are the expected visitor numbers and types ii. Product releases etc c. Who owns it? d. What about copyright? e. How often does it need updating? f. Who will update it? g. Consider privacy, accuracy, property, accessibility	
7. Security implications? a. For different revenue models b. For different audience types c. Sensitive information	
8. Technical issues a. How will the site be created and using which technologies? b. Hosting requirements c. Development environment d. Language e. Hardware	Add a. 7.4.1.5

9. Revisit phase 2 and 3 a. What is the site for? b. Who are the target audience	
Specification	
1. Identify audience types using: c. Statement of purpose d. Market research e. High level target audience identification	
2. Create scenarios for each audience type a. From scenarios identify information requirements b. From scenarios identify functions they will need to complete (Storyboards or prototypes may be considered instead of scenarios)	Rewording create instead of write 7.4.1.6
3. Identify audience structure linking similar information and functions	

G.4. Further work

Addition to the method	**Discussion see**
Use of SWOT Analysis, User Needs Analysis and JAD should be considered as possible candidate techniques in further work	7.4.2.1
An evolution/maintenance phase should be added to the method. A specific testing regime should be considered as part of the method. Consider if security issues are well covered throughout the method. Consider adding a feasibility study.	7.4.2.2
Add specific iteration and prototyping to the method	7.4.2.3
Add the use of documentation as part of further work	7.4.2.5
Add a parallel project management activity and define the steps similar to Hallows (Hallows, 2005) at their respective points in the process. Use the concept of a timeline for the project	7.6.1
The addition of a selection of requirements gathering techniques is seen as outside the scope of this work, but will be considered as part of further work.	7.6.2
Consider the use of a specification template and investigate the use of specification in the method with more than one developer as further work.	7.6.3
Suggest suitable techniques for the design phase of the PECS method as part of further work	7.6.4
Further work could consider the role of method at the development stage for novice developers.	7.6.5
As discussed in earlier sections further work will explore the use of more guidance for complete novices and suggestions as to tools and techniques for specific areas.	7.6.6.2
Clearer definition of the iterations should be made in the guidance	7.6.6.3
A more comprehensive test plan will be explored in further work along with guidance as to when to complete documentation.	7.6.6.5
More detailed guidance as to where the method may be used should be published in the next iteration of the method.	7.6.6.6