WEB USER REQUIREMENTS: A SUPPORT FRAMEWORK FOR STUDENTS

Andrew Paul Bingham

June 2012 PhD

Abstract

This thesis proposes a framework to support the inexperienced student user to undertake the elicitation, analysis and specification of web user requirements. It is designed to support the student during web projects and to encourage more rigorous analysis by documenting web user requirements before the student commences design and implementation. The framework comprises a process meta-model, object model, rules model, support and guidance model, consistency, completeness and correctness model, learning model, student data model and a requirements specification model. The framework was transformed into an automated Computer Aided Web Environment (CAWE) tool and tested on a number of web modules within a Higher Education Institute (HEI).

The research programme adopts the Canonical Action Research (CAR) methodology, which involves one or more iterations of *diagnosing*, *action planning*, *action taking*, *evaluating* and *specifying* what has been learned through reflection and allows interventions to take place within the next research cycle. Students were active participants in the research programme and contributed to the development of the intervention with continuous feedback. Analysis of usage data generated by the CAWE tool provided a valuable insight into how the framework and support mechanism was used by the students.

Main contributions include the extension of knowledge and understanding of Web User Requirements in Web Engineering. Contribution is made to the curriculum of Web Engineering by identifying gaps in knowledge and understanding regarding the lack of analysis techniques used by the student.

Contribution is also made to Web and User Requirements Engineering by proposing, implementing and evaluating a range of novel methods and frameworks through student collaboration.

DEDICATION

To My Fiancée

Nicola

Who unreservedly supported me during my research – and whom I love very much xxxx

Acknowledgements

I would like to thank my supervisors Dr Gary Griffiths, Professor Briony Oates and Professor Mike Lockyer for their continuous supervision, support, guidance and understanding of the trials and tribulations of part-time research.

Thanks to all the students who participated in the research programme. Your enthusiasm engaging with the surveys and using the CAWE tool in the web project provided me with invaluable data.

I would like to acknowledge my mum and nana in encouraging me to continue my university education and academic career.

And finally to my Fiancée, Nicola. Thank you for your patience during the production of this thesis and for your continued support, encouragement and motivation throughout.

Contents

List of Figures

List of Tables

List of Abbreviations

CAR – Canonical Action Research

CASE – Computer Aided Software Engineering

CAWE – Computer Aided Web Engineering

eWURF – Electronic Web User Requirements Framework

HEI – Higher Education Institute

ICA – In Course Assessment

IEEE – Institute of Electrical and Electronics Engineers

IID – Integrated Development module

OBS – Online Business Systems module

PBL – Problem Based Learning

RE – Requirements Engineering

RUMM – Rapid User Modelling Method

SE – Software Engineering

SRS – Software Requirements Specification

VLE – Virtual Learning Environment

WAU – Web Authoring module

WE – Web Engineering

WURF – Web User Requirements Framework

Chapter 1 Introduction

1.1 Purpose of the Study

The initial motivation of this research was to investigate a way of improving students' analysis of their web projects and to address ways of enhancing their learning of Requirements Engineering (RE) within Web Engineering (WE). This is achieved by identifying gaps in knowledge and understanding regarding the lack of analysis techniques used by the student. A review of existing requirements analysis approaches found within Web and Software Engineering further highlights gaps in knowledge in this area. The research objectives evolved through progression over three cycles of research. This resulted in a much broader investigation that led to a set of contributions not envisaged at the outset. This included a review of existing requirements methods and incorporation of their ideas within three interventions to student practice. It also led to students collaborating in the research itself. The research initially focused on web design, but this evolved into investigating web development as a consequence of changes to the curriculum of web development modules at the Higher Education Institute (HEI).

The author had identified in his own teaching of web development that students were not undertaking sufficient analysis before designing and implementing websites. In addition there was an expectation that students had the necessary practical skills, proficiency and motivation to apply a suitable evidence based requirements gathering process to their projects. By making an intervention to existing practice, the students involved in the research were shown to have enhanced competence concerning RE within WE. The evaluation of the research programme provided evidence to suggest that the intervention also improved their professional skills, such as problem solving and evidence based analysis. Students played an important role in shaping an intervention to their own learning by becoming involved directly in this research programme. This aligns with Research Informed Teaching strategies adopted within the HEI.

1.2 Background of the problem

The discipline of Web Engineering has grown rapidly over the last 17 years due to demand from private and public sector projects driven by the ubiquity of access to products and services online. A number of web development methodologies have emerged to support developers in increasingly complex web projects. This ensures a systematic and structured approach is taken throughout the project lifecycle. The objective is that the web project is delivered on time and exhibits a minimum level of quality and conformance to both client and user requirements. Most notably these include, WebML (Ceri, *et al.*, 2000), December (December, 2008), UWE (Koch, 2006), OOHDM (Schwabe, *et al.*, 1996) and SWM2 (Griffiths, *et al.*, 2003).

Research undertaken in this area over the last ten years indicates that there are gaps in knowledge within Web Engineering specifically relating to web user requirements (Ginige, *et al.*, 2001), (Barry, *et al.*, 2001), (Escalona and Koch, 2004), (Escalona and Aragón, 2008). In addition to this there is evidence to suggest that existing web development methodologies tend to concentrate on design and implementation. “There are a significant number of proposals that provide a methodological solution for developing web applications. However, these proposals mainly focus on defining web applications from conceptual models that allow them to systematically obtain implementations. Very few of them rigorously state how to elicit and represent requirements and how to go from the requirements specification to the conceptual model with a sound methodological basis” (Valderas, *et al.*, 2007).

Griffiths, *et al*., 2006 identified a lack of requirements analysis in web development that the focus tends to be on the later phases of the development cycle such as implementation and testing. In addition the paper highlighted problems in current web development projects that are partly due to a lack of attention to systems analysis. Analysis techniques for web development methodologies include audience *definition*, *content analysis*, *market analysis* and *constraint analysis* which are all deemed to be useful techniques to employ before any design work takes place. In addition, the incorrect treatment at the requirements stage has been identified as a reason for failure in web projects. “Reasons for the failure of projects are mostly to be found in the process of Requirements Engineering, shown by several surveys of the Standish Group. Primarily, this is caused by missing or incomplete requirements” (Asarnusch, *et al*., 2006).

There is also evidence to suggest that traditional user requirements techniques do not match needs involved in dealing with web applications with increasing technological complexity and difficulties in analysing a diverse set of web user requirements. “Empirical results show that requirements should be treated carefully. Web systems are becoming more and more complex and it is necessary to know the requirements needed as soon as possible or to at least control their growth to guarantee the quality of the system. Moreover, the special characteristics of Web Systems require special necessities” (Escalona and Aragón, 2008)

Software Engineering methods have evolved over time to reflect the changing nature of both client and developer needs. Traditional Software Engineering techniques follow a linear workflow in which requirements specifications were created and frozen. Problems could then only be discovered at the end the lifecycle when they became costly to rectify. Contemporary software development practices adopt agile approaches where evolutionary prototyping is often used and where changes can be made to the requirements and design throughout. Agile development "embraces change as the norm, not something to be fought" (Avison and Fitzgerald, 2006). Requirements are discovered and modifications made to the software over the project lifecycle and mean that users can influence changes in requirements. The agile philosophy recognises that requirements are not fully understood at the outset of the project and that requirements evolve throughout the development lifecycle.

Within Web Engineering the difficulties in understanding user requirements are magnified, often due to the fact the websites have many more users and development time is compressed. This thesis argues that a more fluid approach for eliciting, analysing and specifying web user requirements should be adopted, reflecting both Web Engineering and to support the inexperienced student user.

Requirements Engineering (RE) is a well established discipline within Software Engineering that helps developers to elicit, analyse and specify requirements before they start the design and implementation of a software product. RE uses a highly ordered and structured approach to determine and communicate a set of requirements relating to the user. Traditional RE techniques assume that all requirements can be documented before design and implementation takes place. Paetsch, *et al*., 2003 argue that there is mismatch between traditional RE approaches and the Agile Software Development process. "Requirements Engineering, on the other hand, is a traditional Software Engineering process with the goal to identify, analyse, document and validate requirements for the system to be developed. Often, Requirements Engineering and agile approaches are seen as being incompatible: RE is often heavily relying on documentation for knowledge sharing while agile methods are focusing on face-to-face collaboration between customers and developers to reach similar goals" (Paetsch, *et al*., 2003).

A literature review on Software, Web and Requirements Engineering approaches highlighted gaps in knowledge in relation to an approach for web user requirements. In addition it has been identified from teaching web development on a number of programmes at a Higher Education Institution (HEI) that a problem exists with the students' understanding of web requirements.

In particular when the author was assessing student projects on his modules it was evident that some students were not specifying requirements in a structured way and were often developing for themselves and focusing solely on their implementation. The lack of analysis identified by Griffiths, *et al*., 2006, combined with the changes in the teaching of web development, (focusing more on programming and database integration), underlines the need to develop an alternative teaching approach regarding the combined disciplines of Web Engineering and Requirements Engineering. We therefore have an opportunity to explore ways to change the students' established practice regarding RE in WE and to support the inexperienced student in their learning.

To support students in their web requirements specification a number of interventions in their learning were deployed, evaluated and refined over a period of five years. The final intervention is represented in the Electronic Web User Requirements Framework (eWURF).

The following models led to the development of a framework in the final research cycle:

1. Process Meta-model.
2. Object model.
3. Rules model.
4. Support and Guidance model.
5. Consistency, Completeness and Correctness model.
6. Student Data model.
7. Requirements Specification model.
8. Learning model.

It was also found of benefit to the research programme to transform the framework into a Computer Aided Web Engineering (CAWE) tool and to allow the students to update their requirement specifications throughout the web project lifecycle. The CAWE tool would become a repository for web user requirements and would support the inexperienced student user throughout the requirements process.

1.3 Research Aims

1. Examine existing Requirements Engineering methods and techniques within Web and Software Engineering.

2. Facilitate the production of a novel method and prototype framework to aid the inexperienced student user to undertake elicitation, analysis and specification of web user requirements.

3. Specify an intervention and framework that comprises a process meta-model, object model, rules model, support and guidance model, consistency, completeness and correctness model, learning model, student data model and a requirements specification model that could be represented in an automated Computer Aided Web Environment (CAWE) tool.

The thesis argues that there is a gap in knowledge between Software Engineering and Web Engineering. It proposes an intervention that challenges established practices in the teaching of requirements analysis and bridges the gap between the two disciplines. The intervention will be evaluated, including its models, methods and frameworks in order to refine the intervention over three cycles of research. It also brings to the Web Engineering discipline improved practice regarding web user requirements and a way forward for further work in this area.

1.4 Research Objectives.

The following research objectives were developed during three cycles of research. Their development is closely aligned and informed by the findings of each cycle of action taking and reflection.

Cycle 1 Research Objectives	Cycle 2 Research Objectives	Cycle 3 Research Objectives
1) To investigate ways of changing current analysis of web requirements in student projects. 2) To evaluate students' opinions regarding the use of a tool to capture and communicate a set of requirements. 3) To demonstrate that a construct for developing user profiles can be used as a starting point within a requirements method.	1) To investigate ways of extending the meta-model to better support the inexperienced student user to define dynamic web requirements. 2) To establish how relationships between requirements and actors can be modelled. 3) To evaluate students' opinions regarding the updated meta-model. 4) To demonstrate that an updated meta-model can be used in the teaching of web development.	1) To investigate how a Computer Aided Web Environment (CAWE) tool can support the inexperienced student user in their requirements elicitation, analysis and specification using a natural language. 2) To investigate how a consistency, completeness, and correctness rules model can be incorporated into the CAWE tool. 3) To determine if usage of the CAWE tool influences assessment outcomes for the student.

Table 1.4 Research Objectives by Research Cycle.

1.5 Overview of Work

The research is undertaken in the following three Canonical Action Research cycles which are summarised below as an overview of work:

1.5.1 Research Cycle One commenced April 2005 within the School of Computing, Teesside University. The first cycle was concerned with establishing a way forward for the design of a suitable method to aid students with their RE process. This was achieved by investigating current analysis techniques used by the student and by exploring ways of enhancing practice through interventions to their learning. The literature review established a range of possible approaches that could be adopted, based on their alignment with RE and WE and the need to support the intervention with an appropriate learning model. A method named RUMM was designed and released on two modules; Design for Usability (DFU) a second year undergraduate module and Integrated Development (IID) a masters module. A student focused evaluation, together with extensive observation of its use by the student in the class room provided a basis for its evaluation. The main findings of this cycle were that it was too focused on non-functional requirements, that there was too much reliance on user profiling within the discovery process and more support was needed during its completion. Observation of student practice within their assessments had identified some tentative improvements with their analysis, although there were still gaps concerning consistency, completeness and correctness of the requirements.

1.5.2 Research Cycle Two aimed to address the weaknesses identified in the first cycle, especially relating to the findings of the survey and observations of the approach being used in the class room. User profiling as a way for discovering requirements was identified as being a weakness in the first cycle. Alternative requirements discovery techniques were sought in the second cycle, in addition to enhancing the web development aspect. In this regard, the modified meta-model adopted a new way to capture requirements, which aligned with similar work undertaken in this field.

To better support the diversity of the student cohort, modifications were also made to the learning model. The hybrid PBL model was adapted to ensure students with particular needs were better supported. The modified method and support mechanism combined together to form an overall framework named Web User Requirements Framework (WURF), which was released to students in the 2008/2009 academic year. An e-learning environment was adopted for its release to the students on two modules; Online Business Systems (OBS) a final year undergraduate module and Integrated Development (IID), a masters module that was also adopted in the first research cycle.

The main findings of this cycle were that the traceability between the requirements specification and implementation were much improved, as evidenced in the student assessments. There were still problems with the consistency, completeness and correctness of the requirements due to the document based approach adopted. Students were more inclined to adopt the framework in this cycle, as it was now an explicit element of the assessment process.

1.5.3 Research Cycle Three focused on refining the meta-model in order to transform it into a Computer Aided Web Engineering (CAWE) tool. An association model, enforcing a set of rules regarding the consistency, correctness and completeness, would address problems with the traceability of the requirements within the students' websites. The CAWE tool would also provide enhanced support during the discovery, analysis and documentation of the requirements. It would also provide an opportunity for further evaluation of the framework in use by the student, as collection of usage data is possible via the CAWE tool. Analysis of the usage and assessment data provided a deeper understanding of its effectiveness. In addition to observing students using the framework in the class room, students were also asked to contribute to an online survey and participate in a focus group at the end of the module.

The main findings of this cycle were that students were able submit more complete requirements specifications in their assessments. Reasons for this were attributed to the support mechanisms being an integral part of the framework and a more rigorous rules model enforcing completeness. In addition, the marking criteria was modified in the final year module (OBS) to encourage not only its adoption, but also its evaluation via an online questionnaire. This was reflected in the higher response rates for the questionnaires in relation to the online survey.

There was still a weakness in the rules model concerning associations between requirements and tasks/actors and changes to the rules model was identified as being needed. An opportunity to embed formative feedback mechanisms to support the student with their requirements specifications were also identified. Data collected by the CAWE tool identified that it was being used at franchised centres in other institutions such as Botswana and Sri Lanka.

1.6 Thesis Structure

The rest of the thesis is structured as follows:

Chapter 2 provides a background to the Requirements, Software and Web Engineering and an overview of the key principles that must be reflected in the interventions to student practice.

Chapter 3 demonstrates how the intervention reflects student needs in terms of learning support. An appropriate learning model is presented that will underpin the intervention for use in the curriculum of Web Design and Development modules.

Chapter 4 outlines the research methodology adopted for the study. It demonstrates how its selection matches the educational context of the research.

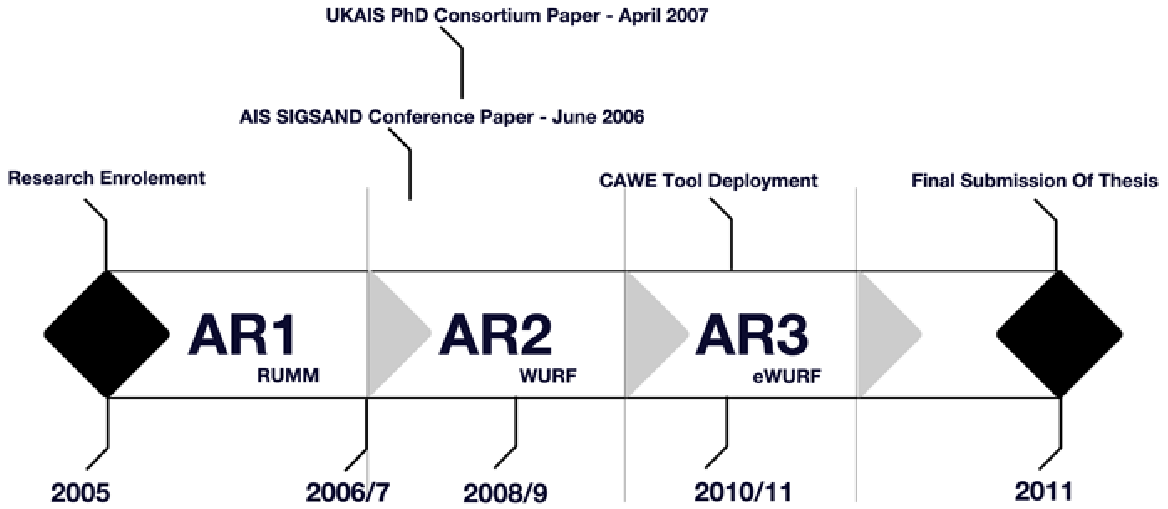
Chapter 5 presents the first experimental intervention and requirements method including its evaluation.

Chapter 6 presents a modified intervention and a collective of meta-models that comprise an overall framework, along with an evaluation.

Chapter 7 presents a final modified intervention and framework represented in a Computer Aided Web Engineering tool (CAWE), along with an in depth evaluation.

Chapter 8 concludes the thesis by reflecting on the research as a whole, presenting the main contributions to knowledge and future work.

1.7 Research Programme Timeline.


UKAIS PhD Consortium Paper - April 2007
AIS SIGSAND Conference Paper - June 2006
Research Enrolement
CAWE Tool Deployment
Final Submission Of Thesis
AR1
RUMM
AR2
WURF
AR3
eWURF
2005
2006/7
2008/9
2010/11
2011

AR1 = Action Research Cycle 1

AR2 = Action Research Cycle 2

AR3 = Action Research Cycle 3

Figure 1.7 Research Programme Timeline.

1.8 Contributions to Knowledge.

The major contributions of this thesis are summarised below:

- Extension of knowledge and understanding of User Requirements in Web Engineering. (Chapter 2 and 3).
- Identification of gaps in knowledge and understanding regarding the lack of analysis techniques used by the student. (Chapter 4, 5, 7 and 8).
- A range of novel methods and frameworks developed through student collaboration that can be adopted for teaching purposes in Web and Requirements Engineering modules. (Chapter 5, 7 and 8).

In addition a number of minor contributions emerged, as summarised below:

- How to implement the final Framework (WURF) within a CAWE tool to support the student user. (Chapter 7).
- How to collect and analyse log data produced by the students, including integration and visualisation within the CAWE tool for both student and tutor use. (Chapter 7).
- A way of visualising the process meta-model in a web user requirements CAWE tool through an automated rules model. (Chapter 7).
- A hybrid PBL model for Requirements Engineering in Web Engineering that aligns theory with practice, including the role of formative and summative assessment and student support. (Chapter 3).
- An evaluation of an action research methodology applied to a computer science based research programme. (Chapter 4, 5, 6, and 7).

1.9 Publications

Bingham, A., P., *et al.*, ‘A Framework For User Characteristics Capture: An Evaluation Of Using RUMM In The Teaching Of Web Design’, 1st AIS SIGSAND European Symposium on Systems Analysis and Design, Galway, Ireland, 2006.

Bingham, A., P., ‘Determining Web User Requirements Through User Modelling’, United Kingdom Academy For Information Systems (UKAIS) PhD Consortium, University Of Manchester, 2007.*

* A paper based on the first publication, but updated to show work in progress and progression of the study.

Chapter 2 Background

2.1 Introduction

The previous chapter introduced the motivation for this research. This chapter presents a literature review of Requirements Engineering and Web Engineering. It argues that a number of key principles found in the literature can be adopted for a web user requirements approach to support the inexperienced student user and change established student practice in requirements analysis.

2.2 Requirements Engineering

Sommerville and Sawyer provide one of the earliest definitions of Requirements Engineering (RE). "Requirements Engineering is a relatively new term which has been invented to cover all the activities involved in discovering, documenting and maintaining a set of requirements for a computer based system. The use of the term 'engineering' implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent and relevant" (Sommerville and Sawyer, 1997).

Requirements Engineering is a process that involves 'people' as stakeholders. Stakeholders can include the user; client; designer and developer, each having different perspectives and needs from the requirements process. Stakeholders need to be included at each stage in process in order to achieve success. "Success can be measured, for example: arriving at a complete unambiguous set of requirements or success in terms of the system" (Macaulay and Mylopoulos, 1995). Each stakeholder must therefore be able to interpret the requirements specification and understand it from their own problem domain.

RE as a computer science discipline has evolved along with recent contemporary software development practices such as, “Agile Development” (Avison and Fitzgerald, 2006). RE can be defined further as a process that encompasses a wide range of methods and tools to help elicit, analyse, specify and evaluate requirements to a range of audiences involved in system development. This section will review RE from the perspective of Web Engineering enabling us to set RE in context within this research programme. It will also inform the development of key criteria that will be used to benchmark existing approaches used within WE and SE.

There is a substantial amount of literature on Requirements Engineering but there is no definitive ‘one size fits all’ Requirements Engineering process. Indeed some authors believe that this is impossible to achieve due to the myriad of variables that exist when applying a requirements process to a given scenario. “There are many possible ways to organise Requirements Engineering processes and they do not transfer well from one organisation to another” (Sommerville and Sawyer, 1997).

One of the prominent researchers in this field is Berry whose work emphasises the *user* as an integral part of the requirements process. “User requirements provide a clear articulation of *how users currently work*, *what they expect to be able to do* and *how they wish to do it*” (Berry, *et al*., 2003). His work distinguishes the difference between a *requirement* and a *user*, emphasising the fact that the user should be considered at each stage of the requirements process. This distinction is pivotal to the approach taken in this research regarding a web user requirements framework as it is recognised that the *user* is perhaps the most important aspect of the requirements process. It is also an area that has been identified as a concern within student web projects.

The term ‘User Requirement’ is cited by many authors in the Requirements Engineering domain although there is also some disagreement with the term *user*. For example, Wiegers, 2006 asks the question of whether requirements should describe system behaviour and therefore should the requirement be written for the system rather than the user? He later adds that the requirement should be written in a way that best communicates the requirement to all stakeholders. It could be argued that both the user (*human*) and system (*logic*) should be considered within the RE process to overcome this.

2.2.1 The Requirements Process

A variety of terms have been found in the literature that describe stages within the RE process. For example, one of the first stages in the process is *eliciting*. The following terms can be found in the literature that describe this stage: *Eliciting; Recording; Capturing; Discovery; Collecting; Inquiry and Surveys*. Elicitation is considered to be one of the most important but often neglected aspect within RE and SE. “It is considered to be the most important activity in information systems development” (Pitts and Browne, 2007). Sommerville 2007 helps illustrate the Requirements Engineering process in Figure 2.2.1.

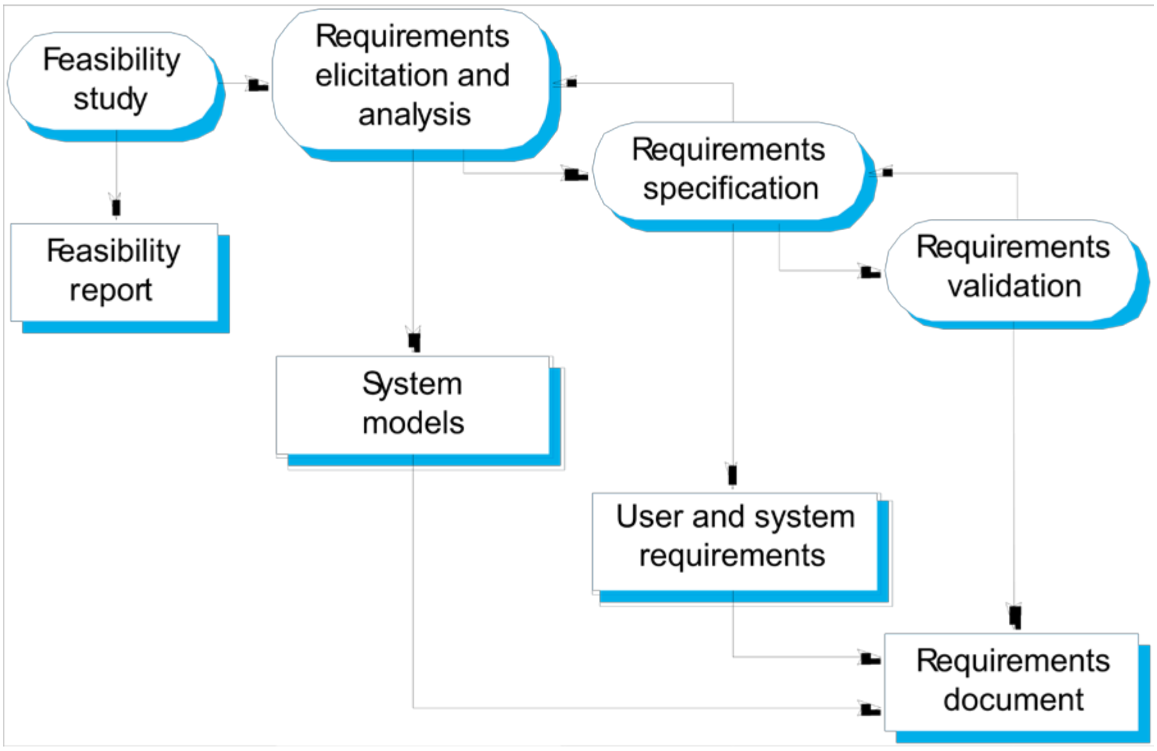


Figure 2.2.1 The Requirements Engineering Process (Sommerville, 2007).

In order to arrive at a *requirements document* that can be understood by all stakeholders a structured process must be followed. Sommerville defines the starting point as a *feasibility study*. The feasibility study provides assistance with the decision making process and documents the outcomes in a ‘feasibility report’. For example, the feasibility study might ask the following questions: *Is the proposed system worthwhile from an organisational perspective? Does it contribute to business objectives? What existing systems need to be considered? What will the technological platforms be on which the system will be built?* In essence it defines the organisational *problem* that will be solved by the development of the system. “Requirements Elicitation and Analysis describes the process of discovering the requirements for a system” (Sommerville, 2007). There are many different approaches that can be taken during this stage and much depends on the context, the organisation, system engineering approach and the type of software that is to be developed. The actual mechanics of discovering the requirements involve *communication*, often one-to-one structured interviews or brainstorming sessions that might be one-to-one or group based. Usually the session is initiated by an analyst who already has some understanding of the organisation and its business objectives and will drive the elicitation stage. The elicitation stage may take one or several iterations in order to arrive at a refined and clear set of detailed requirements. Wiegers also highlights the importance of taking an iterative approach to the elicitation stage. “Requirements elicitation is an exploration and discovery process and the requirements analyst is the guide. Analysts need to recognise that customers won’t be able to deliver all their requirements in a single workshop or discussion. Elicitation requires multiple cycles of refinement, clarification and adjustment as the participants move from high-level concepts to specific details, perhaps through a series of releases or iterations” (Wiegers, 2006).

As such it is important that the web developer or student is able to revisit requirements throughout the lifecycle of the web project as requirements evolve and become better understood. In addition this also reflects agile web development methods where the website is built incrementally over multiple iterations. Developers are able to discover and modify the code based upon multiple cycles of development and testing. Requirements can then be refined over the duration of the web project. Agile Methodologies and the implications for this investigation are discussed in more detail in section 2.3.

Analysis activity helps to derive a more detailed set of requirements captured during the elicitation stage. It is common for the elicitation stage to output *high level*, rather than *low level* requirements which describe the level of detail conveyed. The goal of analysis is to arrive at a consistent, correct and complete set of requirements agreed by all stakeholders. To achieve this, the analyst must refine the high level requirements whilst checking “for conflicts, overlaps, omissions and inconsistencies" (Sommerville, 1997). Analysis must be undertaken by the students themselves within any approach using information from the elicitation stage. A structured analysis could take place by enforcing a process on the student and by ensuring that they transform the information discovered in the elicitation into correctly formed requirements. It is therefore clear that a structured approach must be adopted for analysis to take place.

2.2.2Requirements Specification

The requirements specification includes functional and non-functional requirements. It is widely cited as a software requirements specification (SRS). Functional requirements help describe the behaviours, tasks, interactions and features of a software system. Non-functional requirements can impose constraints on the design or implementation of the software. These can include user interface, quality, performance, or technical requirements.

Both functional and non-functional requirements are usually written in a natural language along with graphical representations unless a formal specification approach is taken. Natural language offers advantages as requirements can be expressed and understood more easily by the stakeholders. "A recent study shows that several software development companies use common natural language for specifying requirements in the early phases" (Mich, 2004). The disadvantage is that they can also be written poorly which may lead to later problems with their usage and traceability.

Functional and non-functional requirements can also be written in a structured natural language that forces authors to adopt a lexicon that standardises the way that the requirements are expressed. This provides an advantage to readers who are expecting requirements to be expressed in a certain way.

Formal specification alternatives include graphical and mathematical notations to express requirements. In theory at least this should provide the benefit of an unambiguous, traceable and complete set of requirements. However, this approach is not without its disadvantages. “Formal methods are still not widely accepted in engineering practice. One of the reasons for this is difficulty of deriving formal specifications from large and complex requirements given in natural language” (Ilic, 2007). The author also adds that “one of the biggest disadvantages is the danger of stakeholders not understanding the mathematical notation”. The SRS is later embedded into the *Requirements Document* where it is then translated into a design and subsequently implemented by the development team.

2.2.3 Requirements Validation

This is a process that must be followed to ensure requirements are complete, correct and consistent and is one of the most important stages within the Requirements Engineering process. Validated requirements are normally locked or frozen within traditional software development and therefore must have the agreement of all stakeholders before proceeding to the requirements document. The main objective of requirements validation is to examine the functional and non-functional requirements to ensure the language used is clear, inconsistencies are removed and dependencies are added. In traditional software development it is also the last opportunity for the client or customer to change the scope of the software and this is something that is often misunderstood. The scope provides boundaries for the system to be developed and without this there is a danger of requirements creep with resulting time and cost overruns. Requirements creep is different to requirements modification as this is something that happens naturally in agile development. “Scope creep refers to the uncontrolled and continuous increase in requirements that makes it impossible to deliver a product on schedule”

(Wiegers, 2006). Scope creep is often driven by a third party, such as the client or other stakeholder, not the development team. Ownership of requirements and its validity is something that should be considered carefully under an agile approach in order to mitigate requirements creep.

Validation under agile approaches is more difficult to achieve as requirements may not be complete when the design and implementation commences. Requirements are discovered and existing ones evolve throughout the project lifecycle. “Requirements validation is an activity that requires different techniques in agile software development. Existing approaches to validation rely heavily on a requirements specification document which is not available in agile” (Gallardo-Valencia and Sim, 2009).

One solution to this is to continuously validate requirements using a set of rules to ensure requirements are complete, correct and consistent during elicitation and analysis process. Ensuring that requirements are valid contributes to the success of the project, both in terms of scope and end user acceptance of the software.

2.2.4 Requirements Document

A ‘requirements document’ is often used to describe a collection of documents or specifications produced from the requirements analysis. These are widely referred to in the literature as ‘functional specifications’, ‘requirements definition’ and ‘software requirements specification’. They are used to communicate system requirements to stakeholders involved in the project and in agile development can be best described as a living document. The living component stems from its role within the development lifecycle where requirements are often updated as discoveries are made which necessitate modifications.

Changes therefore need to be documented and it is standard practice to have a 'revision history' to accompany this as part of the document. It must be emphasised that the requirements document is not a design document and does not contain information pertaining to how the system should be built. For example, it should not detail how a particular behaviour should be implemented, but should instead, be more general and allow the developer to decide on the technology to use. It may include models that help explain functional requirements such as flow diagrams, class diagrams and entity relationship diagrams. The requirements document may integrate with project management systems used by the organisation and could also be used by quality assessors later in the project lifecycle. The requirements document may also bring together other details about the system such as business objectives and the overall vision/motivation for the system, its scope and purpose. For example, Bleistein, *et al.*, 2004 in their paper 'Strategy-Oriented Alignment in Requirements Engineering' describe the identification of business objectives as a means of decomposing requirements.

In the context of this research project this aspect is considered to be crucial as it sets the system in context with wider organisational use. Adoption of this approach would enable the student to see a 'rich picture' of the project before web user requirements are defined.

2.2.5 Requirements Management

Requirements Management is a term that is used widely amongst academics with many different interpretations of its meaning. The popular view is that requirements management is concerned with managing change, tracking and traceability. The Software Engineering Institute (SEI) defines requirements managements as follows. "The purpose of Requirements Management (REQM) is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products" (Chrissis, *et al*., 2003).

Gotel and Mäder explore this concept further in their mini-tutorial presented at the 2009 IEEE International Requirements Engineering Conference into the RE process. In their definition of requirements management they explicitly excluded 'tools to support elicitation and preliminary analysis' and focused more on controlling *consistency, completeness* and *correctness* of the requirements. "Requirements management is therefore the activity concerned with the effective control of information related to system requirements and, in particular, the preservation of the integrity of that information for the life of the system and with respect to changes in the system and its environment. Tools to support the wider aspects of Requirements Engineering, such as the initial exploration and negotiation of stakeholder needs, will not be the primary focus" (Gotel and Mäder, 2009).

Sommerville and Sawyer, 1997, argue that "requirements management is, therefore, a process which supports other Requirements Engineering activities and is carried out in parallel with them" (Sommerville and Sawyer, 1997). From this statement we can see some divergence from other authors in this field. Somerville and Sawyer describe requirements management as the layer that sits beneath the *overall requirements process*. Its function is to provide a mechanism to ensure that requirements are *consistent, complete* and *correct*, which is an essential aspect of requirements management. Requirements management can therefore be defined as a layer which underpins the whole requirements process.

Requirements management will be an essential function of the web user requirements process and methods of achieving this function will be explored within the design and evaluation stages of this research programme.

2.3 Requirement Hierarchy

Requirements can be described at different *levels* of detail which translates into a requirement hierarchy. A specification may define requirements at a high level, often mirroring the responses to questions posed by the analyst. These may be refined over multiple iterations to become more focused and more detailed. In addition, they can be decomposed into further individual requirements. Agile methodologies embrace this approach by starting off a project with high level requirements. As the project continues and requirements are better understood, they continue to be refined into lower level requirements with greater levels of detail. “A common agile practice is to perform some high-level requirements envisioning early in the project to help come to a common understanding as to the scope of what you're trying to accomplish. The goals at this point are to identify the business goals for the effort, develop a common vision and swiftly identify the initial requirements for the system at a high-level” (Ambler, 2002).

2.3.1 High Level Requirements

A High Level Requirement (HLR) is the most generalised breakdown of a system to be developed, usually expressed by the client or user group who have an in depth understanding of a task or business problem. For example, an administration area of a website requires a mechanism by which it can be securely accessed. This task could be represented in a HLR using the following statement:

> Secure Login: *When prompted to use a restricted area of the website, the user shall be able to enter an alphanumerical password to enter that part of the system*

Whilst an HLR is enough to provide some information to the developer, much more is needed before they can implement it. For example, this HLR does not describe the minimum and maximum digits that can be entered. The HLR does not convey enough detail required to fully implement the requirement. For example, it is not clear what happens if the user enters the wrong password or how many attempts may be made before the system locks out.

There is usually a direct relationship with a business problem or set of tasks that enables the user to solve that business problem. HLR are sometimes drafted to provide an opportunity for stakeholders to engage and contribute to the development and refinement of requirements moving increasingly to a lower level of abstraction to enable developers to move to the implementation phase.

2.3.2 Low Level Requirement

The language used to describe a Low Level Requirement (LLR) is more technical and is related to the software platform. LLR's are more likely to define procedures or execution boundaries and will take into account dependencies with other requirements. An LLR should have traceability back to a HLR to enable requirements to be managed. They should also be visible within the software and measurable in terms of an input and expected output in the design of the interface or in its conformance to a range of security or quality assurance tests. Returning to the previous example of an administration area of a website represented in an HLR, an LLR would provide much more detail, for example:

Secure Login: *On Accessing a restricted webpage the user is taken back to a login page. The user enters a 10 digit alphanumerical password. If more or less digits are entered, an pop up message will display to inform them to enter more or less digits. The system shall evaluate the authenticity of the user. If the password has been entered correctly, the system shall enable them to view the restricted page. If the password is incorrect, a pop up message will be displayed. If the wrong password is entered three times, the system will lockout for ten minutes. This requirement is related to the Secure Access requirement.*

Being able to describe a requirement at different levels is useful in the communication process. For example, initial HLR's are often specified in a draft requirements specification and signed off by stakeholders. An advantage of this is that "success-critical stakeholders can suggest additions and amendments early in the requirements process" (Kitapci and Boehm, 2006).

Perhaps the biggest advantage in using the HLR approach is that opinions emerge and a consensus is reached early on in the process. Although it could be argued that this approach extends the process in terms of time and resources there is evidence to suggest that this provides more benefits. "Effectively negotiating requirements from various stakeholders who have different roles and responsibilities during the early stages of the software development is a key factor of successful software projects" (Boehm and In, 1996).

2.4 Requirement Priorities

A requirement may have a 'Priority Level' expressed in the specification. The priority metric is based upon its importance within the overall system. A priority is often negotiated by the client and other stakeholders. One consideration, for example, is it is more important that a client can access information than it is to change text size on a web interface? “By addressing high-priority requirements before low-priority ones, one can significantly reduce project costs and duration. It is difficult enough for a customer to decide which of his requirements are most important; achieving consensus among stakeholders with diverse expectations is even more challenging. Factors concerning different stakeholders such as business value, risks, relation to other requirements, etc., should be considered while prioritising requirements” (Xiaoqing, *et al*., 2004).

The decision to prioritise a requirement is often made more complex by conflicting factors. The cost associated with implementing a requirement, perceived value to the client, technical feasibility and the obligation to meet minimum quality standards all have a part in the decision making process. A Cost-to-Value prioritisation approach is most widely used but this approach will sometimes have a negative impact. For example, in using a website the end user experience can be enhanced by integrating more costly technology. If important elements were removed due to cost, the website may not serve the end user as well as it could have, resulting in lower usage of the site. “Several prioritization methods have been proposed, each of which uses different mathematical or analytic approaches for requirements prioritization” (Karlsson, *et al*., 1998). An automated prioritisation process may not be in the interests of the project as a whole.

Once the priority decisions have been made, the development team will concentrate on developing the most critical features. High priority requirements will tend to be visible in early iterations of the development leading therefore to benefits in terms of stability and enabling stakeholders to contribute an evolving set of requirements.

The priority metric may be expressed within a specification using a scale. For example, *Essential – Conditional – Optional* or *High – Medium – Low* scales are commonly used. The metric is usually fully documented in the specification, for example:

- *High Priority - the requirement is critical for operations or performing the job.*
- *Medium Priority - the task can be performed without the requirement, but not very well.*
- *Low Priority - the requirement would be 'nice to have'.*

In a traditional RE process where requirements are specified and frozen this usually works well. In a scenario where requirements evolve or change iteratively this approach can cause problems. In order to mitigate this problem an effective requirements management process or *layer* can be used, providing a mechanism to track changes in priority and flag conflicts and dependencies. The 'priority level' is therefore something that needs to be considered in the design of the web user requirements method.

2.5 Model-Driven Requirements Engineering

A relatively new area to emerge within RE is ‘Model-Driven Requirements Engineering’ (MDRE). Although much of the work undertaken in MDRE so far is in the field of Software Engineering rather than Web Engineering, it is still worthy of further investigation as a possible approach for web user requirements.

MDRE is part of the ‘Model Driven Approach’ (MDA) that uses formal models that have the exact meaning of program code. “A model is an abstract representation of a systems structure, function or behaviour. MDA models are usually defined in UML. In principle, the MDA formally considers even classic programming languages as MDA languages that in turn maintain relationships with a platform” (Stahl and Völter, 2006).

Model Driven Engineering (MDE) adopts an MDA approach and offers a framework and set of tools in which to develop software. “Model-driven engineering (MDE) offers a technical framework that can relate software development activities around meta-models and model transformations” (Baudry, *et al*., 2007). Model Driven Requirements Engineering applies MDE techniques to RE. MDRE uses a constrained natural language for requirements definition. Using an environment for meta-models, functional requirements can be simulated in order to test and validate consistency of the requirements. Some MDRE frameworks are also able to link business logic with functional requirements.

MDRE is a formal process in that adoption requires an understanding of a modelling language such as UML. This presents opportunities, one being that if a developer is using an MDRE and MDE approach for software development both can be integrated readily. A disadvantage is that it may not suit all development approaches and some developers may be already used to other engineering approaches such as Object Orientated Programming (OOP).

From an educational point of view, the author can appreciate the benefits of adopting an MDRE approach in a web development module. Its use in a web design module, where the student may not have been exposed to any formal engineering methods may not be the correct approach to adopt. It is also outside the scope this research as it is not the intention to create code from the requirements process.

2.6 Unified Requirements Modelling Language

The Unified Requirements Modelling Language (URML) is an approach rather than a specific process. “The basic idea is to create a single, traceable and consistent requirements model instead of relying on thousands of pages of text with manually created traceability links. Requirements are captured in a visual language following the principles for *cross-disciplinary use*. URML contains not only diagrams but also semantics defining the relationships between requirements and rules for creating diagrams and textual artefacts” (Berenbach and Gall, 2006). The URML approach has recently started to emerge in the field of RE and a number of authors advocate its adoption for Software Engineering. For example, an investigation undertaken by Berenbach and Gall, 2006 at the University of Munchen, looked at URML as a way of solving the problem of communication between different analysts, where they were working across different disciplines such as web and software, as well as a method of combining functional and non-functional requirements into one common model. “Current practice is to partition functional and non-functional requirements such that they are often defined by different teams” (Berenbach and Gall, 2006).

The investigation found that by extending UML with specific requirements symbols to bind high level features to a UML model they were able to elicit requirements across different teams of analysts in different countries and provide traceability of requirements. They found that they could reuse requirements and visualise requirements dependencies. An example of a requirements model in URML can be found in Figure 2.6 below. The model is represented by a set of symbols attached to the requirements and relationships to provide developers with an interpretation mechanism that transcends language barriers within the organisation. For example, Berenbach and Gall, 2006 found that in their investigation, which focused on Siemens AG, the approach helped in the communication process where outsourcing resulted in different teams in different geographical locations.

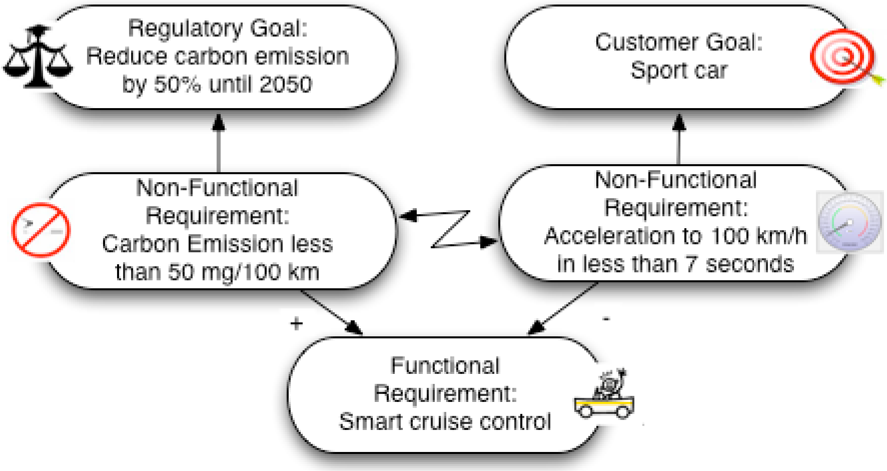


Figure 2.6 URML Requirements Model (Helming, *et al.*, 2010).

URML reflects the benefits in *visualising* requirements in order to better communicate the relationships between different requirements amongst the development team. URML also tends to concentrate on *features* of the system to be developed, which is a more tangible concept than an abstract model. In this respect, URML provides a useful direction to this investigation especially given the relationship between *tasks* that the user carries out within the website and the *high level functional requirement* that is used to describe this. This would especially suit those students' who are designers and more 'visual' in their problem solving skills. An automated production of the associations between tasks and functional requirements is also something that could prove useful in a web user requirements process.

2.7 Requirement Storage and Management

When considering the requirements management process, it is useful to choose a suitable mechanism to store and manage requirements. A *document* or *database* based approach could be adopted depending on which requirements process is chosen and whether the requirements document is distributed electronically or in hard copy.

A Requirements Management Database (RDB) embodies a range of tools that can be used within the RE process in order to *store*, *manage* and *maintain* links between requirements. RDB is a data centric approach that enables requirements to be stored, retrieved and analysed throughout the project lifecycle. Sommerville and Sawyer exemplify the benefits of setting up an RDB. "If you maintain your requirements in a database, you can design the requirements database to include traceability information" (Sommerville and Sawyer, 1997). A difficulty with this approach is both the cost and time required to setup the database and provide access to stakeholders. An advantage is that links between requirements can be maintained by use of foreign keys. An RDB can also be manipulated to output documents and can be tailored to meet the particular needs of the audience. For example, high level requirements can be extracted to form an initial 'draft' specification in order for stakeholders to contribute to the RE process. Low level requirements can then be extracted from the same data and targeted at developers with different needs. The same requirements are maintained in order to satisfy traceability, validation and change control.

A benefit to the web user requirements process in adopting the database based approach would be the ability for students to build an initial set of requirements and amend these throughout the web project. Requirements could be refined as the student discovers, through their agile development process, that requirements require modification or translation into LLR's. Using the database approach would also allow the student to express a relationship between requirements. For example, how functional requirements relate to user tasks or behaviours.

A document based approach is useful where stakeholders are accustomed to working with documents such as Microsoft Word. An advantage of this approach is that it can be setup quickly and relatively cheaply. A disadvantage is that it is harder to maintain and control requirements in this way. It may still be possible to monitor changes via 'track changes' in Microsoft Word, for example.

Alternatively a number of software packages can assist in the storing of requirements. These are effectively plug-ins for Microsoft Word, where the analyst writes the requirements in the document and the software stores part of the document in a database for retrieval at a later stage. An example of a document based requirement management plug-in is *RequisitePro*, which has been development by IBM. This software integrates with Microsoft Word to facilitate requirements definition, traceability and collaboration.

The document based approach may suit the initial design, deployment and testing of an experimental method in the first research cycle. Progression towards an RDB would then be possible in the final research cycle where this would provide greater capability for the storage and manipulation of requirements.

2.8 Web Engineering

Web Engineering (WE) is a relatively new discipline that helps unify a number of approaches to aid the web development process. It borrows heavily from Software Engineering in terms of process, methods and tools. This chapter looks at Web Engineering from the perspective of process, methods and tools and how Software Engineering has shaped Web Engineering. Implications for the treatment of web user requirements within web methodologies are outlined which in turn will inform the development of the web user requirements process within the first research cycle.

2.8.1 Software Engineering Philosophy

According to Pressman, Software Engineering is a discipline that incorporates a number of different layers such as quality, process, methods and tools. “Its foundation is an organisational commitment to quality and the process layer is the glue that holds the technology layers together and enables rational and timely development of computer software. It (process), forms the basis for management control of software projects and establishes the context in which technical methods are applied, work products (eg., models and documents) are produced, milestones are established, quality is ensued and change is properly managed” (Pressman, 2000).

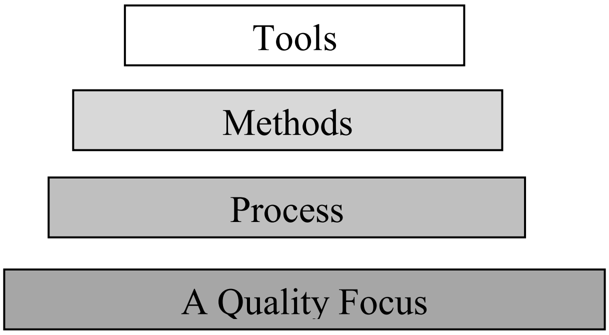


Figure 2.8.1 Software Engineering Philosophy (Pressman, 2000).

The *process* is emphasised as the layer that links together all layers in the philosophy and is considered to be the most important attribute. Pressman defines methods as a broad range of actions and tasks that can include *communication*, *requirements analysis*, *design modelling, program construction*, *testing* and *support*.

Process and *methods* together contribute to an overall software development methodology or *framework* as they are referred to by Pressman. For the purposes of consistency the term *methodology* will be used to distinguish a development methodology from a process that focuses on one aspect of software development.

Development methodologies help achieve a structured approach and are well established in Software Engineering. Many of the principles found in Software Engineering were adopted and subsequently adapted to fit the needs of Web Engineering. For example, the much cited 'waterfall method' has a philosophy of sequential development. Progress through development follows a rigid sequence of stages. Each stage is locked, with the consequence that you could not revisit previous stages iteratively. The Waterfall Method is formally referred to as the Software Development Life Cycle (SDLC) in the literature. SDLC was reviewed by Winston W Royce in 1970 who described it as a flawed non-working model. “I believe in this concept, but the implementation described above is risky and invites failure. The testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., are experienced as distinguished from analyzed” (Royce, 1970). Royce’s main concern was that testing appears after the implementation phase making it difficult to return to previous stages (iteratively) should changes to the requirements or design be required.

The SDLC has and still does influence general approaches to software development. A number of variants exist that represent different perspectives on a staged developmental process, however they all follow a basic structure:

1. Feasibility Study
2. System investigation
3. System analysis
4. System design
5. Implementation
6. Review and Maintenance

Within the analysis and design stage, techniques and tools that aid the software developer include *flowcharts*, *specifications*, *grid charts* and *entity relationship diagrams*. Many approaches based on SDLC emphasise planning and analysis and advocate strong adherence to the engineering aspect. A developer may use a variety of tools within the planning stage in order to arrive at a set of requirements. Figure 2.3.1.2 shows a Taxonomy of Software Development Methodologies. The evolution of these methodologies was dictated by the increasing complexity of software and the need to decompose systems further. The development of this approach was due to the increasingly complex interrelationships between systems and the user rather than predictable sequences of development in early software development. The move to *object orientated* methodologies became apparent during the 1990's, hence the plethora of methodologies that appeared from this period onwards, along with more conventional methodologies such as "Rapid Application Development (RAD)" (Gerber, *et al*., 2007).

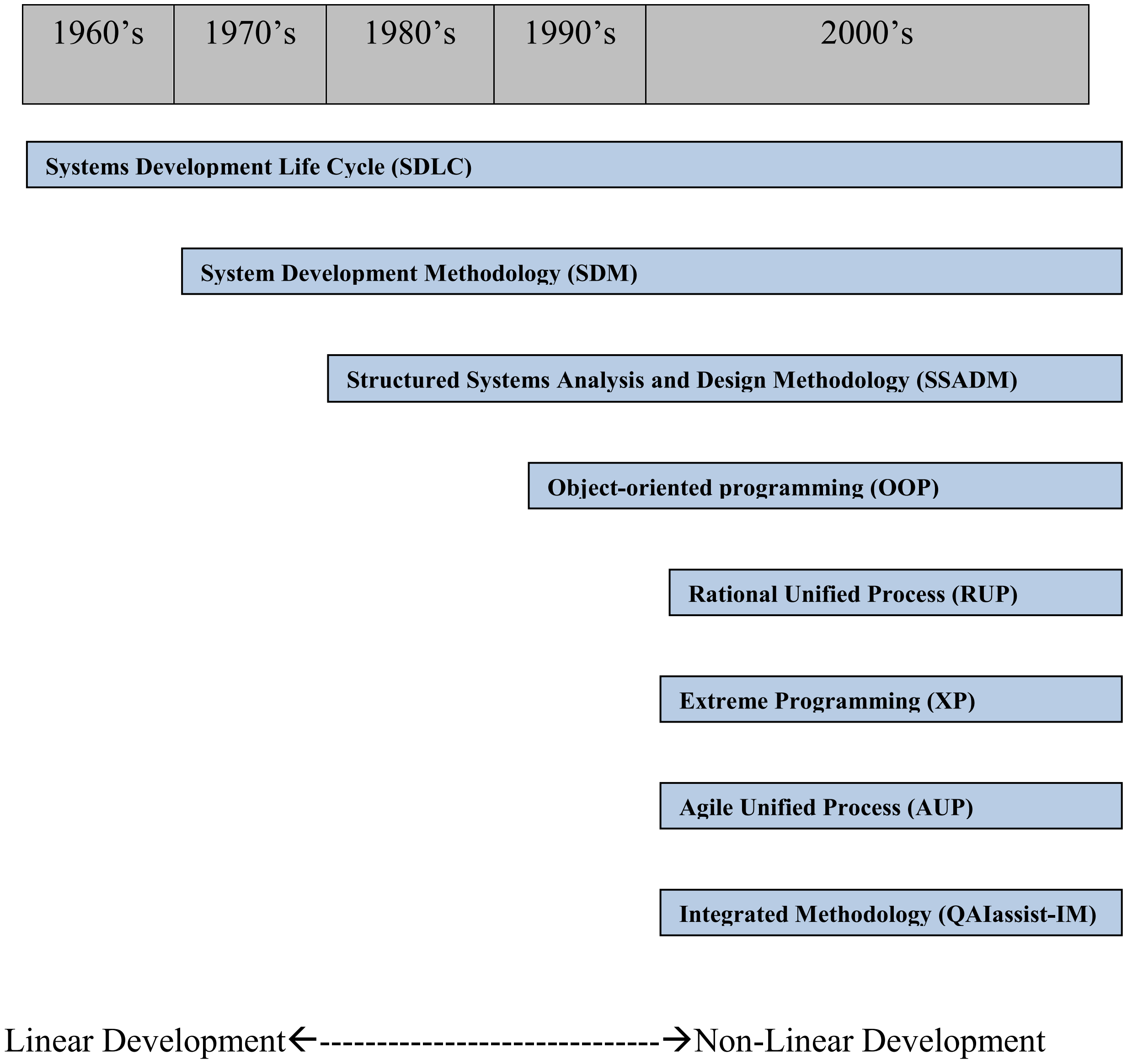

1960's
1970's
1980's
1990's
2000's
Systems Development Life Cycle (SDLC)
System Development Methodology (SDM)
Structured Systems Analysis and Design Methodology (SSADM)
Object-oriented programming (OOP)
Rational Unified Process (RUP)
Extreme Programming (XP)
Agile Unified Process (AUP)
Integrated Methodology (QAIassist-IM)
Linear Development←----------------------------→Non-Linear Development

Figure 2.8.2 Taxonomy of Software Development Methodologies.

2.8.2 Web Engineering Philosophy

Web Engineering encompasses a wide range of interdisciplinary areas such as; *analysis and design, usability, user experience design, Requirements Engineering, information engineering, testing, project management and graphic design*. It could be argued that the reason for this is due to the uniqueness of web application development. When web development was in its infancy, developers often used *hacking* approaches to implement websites. Quite often no structured process was employed to analyse, document, test or evaluate what they produced. Websites were often static in nature and very basic in the delivery of information with this being limited to hyperlinked text and images. It could be argued that a methodical approach was not necessary due to the simplicity of text and image based documents. However, web applications are now becoming more complex to implement due to changing requirements, hence "developing quality Web applications quickly and error free is one of the most challenging problems in the Web Engineering field. This kind of software always stresses development teams because requirements tend to change fast (the permanent beta syndrome)" (Luna, *et al*., 2010).

As the demand for websites grew business requirements for e-commerce functionality and the need to work with more complex technologies arose. Databases, dynamic server-side languages and integration with legacy systems were required in order to satisfy business objectives. 'Ad hoc' web development was no longer tolerated by established organisations and developers looked to existing methods that could help control code, establish quality procedures and solve reliability issues that were abounded in the early web applications. Most of the existing methods were to be found in Software Engineering.

There are many definitions around Web Engineering and its distinction from Software Engineering. For example, Pressman's view is that "*Web Engineering proposes an agile, yet disciplined framework for building industry-quality WebApps*" (Pressman and Lowe, 2008). Pressman emphasises agility as the 'degree of difference' to that of Software Engineering and believes that web engineers have to respond quickly to changing rules and requirement's, for example as stakeholders often change their minds. He therefore proposes that existing methodologies already established in Software Engineering cannot be simply *cloned* for Web Engineering. Pressman believes that in Web Engineering, stakeholders are more likely to ‘change their minds’ regarding requirements. This belief is probably held because the target user group is less well defined than in a typical software application. It may also be due to the way that web requirements evolve through the iterative ‘agile’ development processes that is now used within web development.

Ginige and Murugean’s view is that “Web Engineering is the application of scientific, engineering and management principles and disciplined and systematic approaches to successful development, deployment and maintenance of high quality Web-based systems and applications” (Murugesan, *et al*., 1999). They provide a distinction to the discipline of Software Engineering by using the term *maintenance* and emphasise that the frequency of this is much higher than that of Software Engineering. “Maintenance is a continual process” (Murugesan, *et al*., 1999). Ginige and Murugean’s distinction between ‘software’ and ‘web’ is that software tends to be built and then revised over time through version control. Websites on the other hand can be updated continuously and maintenance must an integral and recognised aspect of development and release.

Lowe and Hall’s view is that Web Engineering is “the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of web systems” (Lowe and Hall, 1999). Lowe and Hall’s views on Web Engineering are similar to those of Ginige and Murugean regarding the maintenance of websites, however, they also recognise that websites are ‘operated’ continuously. The degree to which a website may be ‘operated’ and ‘maintained’ would be reflected by its usage. A static website involving only text and images may require low levels of maintenance to sustain its operation. Conversely, an e-commerce site would require a high level of maintenance in its day to day operation.

2.8.3 Differences between Software and Web Engineering

There is much debate on the differences between *Software* and *Web* Engineering. On one hand it could be argued that web applications are executable programs and therefore can be defined as software. An alternatively view is that there is a need for a separate discipline that recognises the particularities of web development.

It can also be argued that software applications exhibits different *characteristics* to web applications and ergo Web Engineering methods and techniques cannot *duplicate* that of Software Engineering. Ginige and Murugean, for example, suggest that "Contrary to the perception of some professionals, Web Engineering is not a clone of Software Engineering, although both involve programming and software development" (Ginige and Murugean, 2001). They also add that although Software Engineering principles are used, Web Engineering has grown its own set of approaches, methodologies, tools and techniques to meet the unique requirements of web-based systems.

Powell believes "that when considered in their entirety, a complete set of WebApp characteristics do differentiate web-based systems from more conventional computer based systems" (Powell, *et al.*, 1998). In addition there is growing consensus that traditional software practices do not fit the needs of website developers. "Web application development differs from development of traditional software in several significant ways; therefore engineering for web applications entails new demands accordingly" (Powell, *et al.*, 1998).

Pressman 1998 in Al-Salem and Samaha, 2007 opposes this view. "Based on the argument that WebApps are an natural evolution for information systems, as a solution for problems exhibited by previous systems. Thus, the current traditional methods, tools and techniques from Software Engineering are still applicable". Al-Salem and Samaha, 2007 are of the opinion that Web Engineering is an evolution of Software Engineering, although 'web' has special characteristics that need to be recognised. For example, Al-Salem and Samaha believe that Web Engineering has special characteristics such as:

- A Multi-disciplinary development team;
- State-of-the-art technology;
- Diverse and volatile requirements;
- Vast and Unknown end users;
- Multiple Stakeholders;
- Short development lifecycle;
- Essential quality requirements;
- Heavy content;
- Integration with backend databases and third party applications;
- Adaptable architecture;
- Visibility;
- WebApps relevance and direct effect on business.
  (Al-Salem and Samaha, 2007).

It could also be argued that Software Engineering has moved further towards Web Engineering in recent years. This is especially true where software is distributed over the internet as opposed to being installed on a stand-alone platform. This includes applications that rely on web services for their data such as desktop widgets and software that relies on regular updates in order to run effectively. "Web Applications can be considered a special class of software applications. The web applications can serve part of a larger system: information, organisational, control, etc" (Casteleyn, *et al.*, 2009).

These key changes in both Software and Web Engineering therefore compel us to think about Web Engineering as an evolution, rather than revolution and ideas from both disciplines can influence an approach to web user requirements in this research programme. It is therefore valid to investigate approaches, methods and tools employed in Software Engineering with a view to adapt them for use in Web Engineering. This will be undertaken in a review of related work (see Appendix A3).

2.8.4 Web and Software Development Methodologies

A number of prominent development methodologies have emerged in order to address the differences between *web* and *software* development. Notable web development methodologies are presented in Table 2.8.4.

1	December (December, 1996)
2	IBM (IBM, 2000)
3	Relationship Management Methodology (RMM) (Isakowia , *et al*., 1995)
4	WebML (Ceri, *et al*., 2000)
5	Conallen's adaptation of the UML for web development (Conallen, 2003)
6	UML-based Web Engineering (UWE) (Koch, *et al*., 2008)
7	Web Semantics Design Method (WSDM) (De Troyer, *et al*., 2007)
8	Agile Process For Web-based Application Development (XWebProcess) (Sampaio, *et al*., 2004)
9	WebHelix (Whitson, 2006)
10	MPM (Chen and Heath, 2005)
11	OOHDM (Schwabe, *et al*., 1996)
12	SWM2 (Griffiths, *et al*., 2003)

Table 2.8.4 Web Development Methodologies

Table 2.8.5 presents a number of cross disciplinary methodologies that are used in both Software and Web Development.

1	Rapid Application Development (RAD) (Gerber, *et al*., 2007)
2	Model Driven Development (MDD) (Selic, 2003)
3	Iterative and Incremental Development (IID) (Larman and Basili, 2003)
4	Spiral Model (Boehm, 1986)
5	Feature-Driven Development (FDD) (Palmer and Felsing, 2002)

Table 2.8.5 Cross Disciplinary Development methodologies (CDD)

CDD methodologies have emerged to reflect changes in the way both software applications and websites are produced and in response to the mismatch between traditional waterfall or SDLC methodologies. "The waterfall model process was perfect for developing a file maintenance program for mainframes, but far too restrictive a process for building a Web application. Web application development needs to be an iterative process and most agree that a spiral approach is best" (Altarawneh and Shiekh, 2008).

Other methodologies reflect the way in which web applications are constructed and related to their models. For example, a number follow the *three tier approach* to development. "After requirement elicitation, a web application is usually designed in a three stage process that defines an application model, a navigational model and a presentation model" (Garrido, *et al*., 2009). Recognition of the 'modelling' aspect is clear in Garrido's research, where the website design is decomposed into three or more models.

These can then be transformed into iterative prototypes with traceability back through to the design models and documented requirements.

2.8.5 Requirements Analysis Integration within Web Development Methodologies

The methodologies listed in Table 2.8.4 and 2.8.5 on the previous two pages, were investigated further in order to better understand how they treat Requirements Analysis.

Methodology	Type	Requirements Analysis Undertaken
December	Web	No, although some mention of capturing information about the ‘web audience’, ‘website goals’ and ‘website vision’.
IBM	Web	Yes, very well defined in the process model.
RMM	Web	Yes, with a requirements document produced.
WebML	Web	Yes, with separate tasks for collection and specification.
Conallen’s UML	Web	Some, using USE cases.
UWE	Web	Some, using USE cases.
WSDM	Web	Yes, user analysis and user requirements class descriptions.
XWebProcess	Web	Yes, definition of initial requirements

		and then developed iteratively as the project evolves (define and revise).
WebHelix	Web	Some analysis undertaken, but does not specifically address requirements.
MPM	Web	Yes, defined as *user requirements*, *functional requirements*, *data requirements*, *interface and architecture requirements*.
OOHDM	Web	Not addressed.
SWM2	Web	Yes, undertaken in the analysis stage.
Rapid Application Development (RAD)	Web/Software	Yes, expressed as user requirements.
Model Driven Development (MDD)	Web/Software	Yes, expressed as functional and business requirements.
Iterative and Incremental Development (IID)	Web/Software	Yes, comes after business modelling.
Spiral Model	Web/Software	Yes, defined as system requirements.

Feature-Driven Development (FDD)	Web/Software	Yes, expressed as features.

Table 2.8.6 Requirements Analysis Treatment in Web and Software Development Methodologies

By carrying out this analysis into both Software and Web methodologies, it can be established that:

1. Requirements Analysis seems to be an embedded stage in most methodologies.

2. Methodology authors are poor at informing the developer how to go about eliciting, analysing and specifying those requirements.

3. There is no standard way of expressing requirements. For example, MDD supports business requirements and FDD intertwines functional requirements with website features.

4. Iterative development features heavily, resulting in the need for fluid requirements that can evolve throughout the web project lifecycle.

5. Most, but not all, have requirements analysis as the first stage in the process.

It has already been noted that a problem exists in the teaching of web modules where it is evident from the students' submissions that requirement analysis seldom takes place. However, if a student adopts a methodology it is the requirements stage that is often neglected and not completed as rigorously as is required. There is a gap in understanding between the inference of what the student should be doing in terms of requirements elicitation, analysis and specification and how this is achieved in practice. By bridging the gap in knowledge between Requirements Engineering and Web Engineering by proposing an educational focused approach it is hoped that the student will be able to undertake requirements analysis within their web methodology.

2.8.6 Problems in Web Engineering

It is widely acknowledged that a lack of requirements analysis affects web projects in terms of end user acceptance and non-conformance to the clients business objectives. For example, according to Lowe and Eklund "a major problem that occurs in Web Engineering projects is that the users get to know how to express *their requirements very late in the process*, i.e. after the design artefacts appeared" (Lowe and Eklund, 2002).

It was also found that web developers working in the field could be described as *indifferent* to the use of any formal methodology to control process. In a survey carried out by Lang & Fitzgerald in 2005, it was found that whilst many organisations were using a methodology which had a well defined process, only half were explicitly documented in practice. Many organisations had adopted a selection of methods and toolkits that enabled them to simply get the job done. This could be described as a hybrid approach, with many in-house hybrid methods being adopted to control parts of the web development process. It is widely believed that developers are not adopting academic methodologies partly due to the short development life cycle of web projects, as highlighted by Babanezhad, *et al.*, 2010. "Web development processes have special properties such as short development cycles" (Babanezhad, *et al.*, 2010).

The way *practitioners* are using methodologies has implications for this research, but these are less significant due to the educational context that this investigation is set in. However, it could be argued that although an academic approach is being investigated and proposed, the problems facing developers are still relevant.

Further evidence for the need for a systematic way to communicate requirements in web development is highlighted by Lockyer *et al*., 2003. "Today, many sites are large-scale and involve sophisticated interaction with visitors and databases; such sites are often regarded as mission critical. In parallel with this evolution, a need for Web Engineering has become apparent. Yet, within education, the plethora of web courses primarily address the implementation of web sites with very little about their *analysis* and design of web applications" (Lockyer, *et al*., 2003).

This point is reinforced by Whitson, 2006, who proposed a lightweight web development methodology named 'Helix' in order to address analysis and design in his teaching, whilst recognising that this does not happen in industry. "In spite of the large number of Web application development processes available, project managers often report that projects developed in the real world are done with little or no design methodology" (Whitson, 2006).

2.8.7 Differences between Web and Software Requirements

Section 2.3 has argued that Web Engineering is an evolution of Software Engineering and is a discipline in its own right due to the unique challenges to be found in web development.

In most instances web requirements are much more volatile than software requirements due to the variety of target users, short development lead times, web specific functional requirements and the fact that unproven technologies are more likely to be defined in response to user or client requirements. Web Requirements therefore need to be treated differently to those of Software

Requirements. A rationale for the way web requirements could be treated within the web user requirements approach is outlined in more detail within section 2.4.

2.9 Web Requirements

In section 2.8.4, an argument for the separation of Web Engineering from the discipline of Software Engineering was put forward. These key differences play a part in shaping an approach in the design of a web user requirements method as set out later, within Chapter 5. Section 2.8.4 further more highlights the need to make the requirements *human readable*, *precise*, *complete* and to be able to articulate the requirements to all stakeholders involved in the project.

It must be emphasised that one of the main aims of this investigation is to bridge the gap between RE and WE in an educational environment. This brings with it unique challenges, including the diversity of the learner and their learning styles, which will be addressed in Chapter 3. This section establishes a need for 'web specific requirements', including the treatment of functional and non-functional requirements within the Web Engineering domain.

2.9.1 Discovery of Web User Requirements

Section 2.2 explored the RE process, including a process model that is typical of a requirements method. Appendix 3 related work, further identified specific methods that could be adopted. A range of approaches were typified by the classification of functional and non-functional requirements. For example, OVID and ARM both classify requirements based on a behaviour, task or goal. Alternative classifications include the further refinement of requirements into *Business Requirements, Systems Requirements, Operations*

Requirements, Interaction Requirements, Actor Requirements and Information Storage Requirements.

It was also found that some methods used different models to aid the analyst in the exploration of requirements. For example, *Tasks*; *Behaviours*; *Interactions* and *Features* are used to form relationships with the user of the web application. These relationships are modelled to ensure the traceability of requirements in later stages of the RE process. To help *envision* requirements, existing models also included a stage that helped to define the project in context within the organisation. A mission or vision statement helped encapsulate the project and its objectives. Business Objectives were also found featured, sometimes expressed as Business Requirements. This represents an important aspect of the RE process, where ill-defined requirements need to be discovered and further refined throughout the web project lifecycle. By defining business objectives, it can be argued that the student would be able to see the *bigger picture* and then refine the web requirements based on business and user needs.

A number of requirements methods used the term ‘Actor’ to describe the user of the system. An Actor name provided the analyst with an archetypal user with which to associate tasks and functional requirements. A number of methods went further than this by envisioning to a greater extent a profile for the Actor. The profile can include *age*, *gender*, *usage* and *behavioural* traits.

2.9.2 Web Functional Requirements

Soares and Vrancken*,* 2008*,* describe functional requirements as they stand in the Software Engineering discipline. “Functional: describes what the system should do to be useful within the stakeholders’ context (the functionalities), including information about logical databases, such as frequency of use, data entities and integrity constraints” (Soares and Vrancken, 2008).

Web functional requirements should describe the behaviour of the web application being developed and have a direct relationship with a task or data entity. Model Driven Software Development (MDSD) approaches can mesh with tools that support the translation of the model into code. Whilst the advantages of this are recognised, a problem presents itself in respect of the students capability in this area where he/she may not be developing using the MDSD approach and may have a *design* rather than *developmental* background. Forcing the student to adopt MDSD approach may result in poor adoption of an RE process.

The variations in approach call into question the validity of a *one size fits all* definition for a web functional requirement. One way around this is to leave the web functional requirement component open, albeit with a guidance and validity checking model that enables the student to define functional requirements in a way that suits their development style. In order to provide traceability it is deemed essential to include a reference system for each requirement. Additionally, it is also thought essential to enable the student to create relationships between an actor and the task that they will perform within the web application.

2.9.3 Web Non-Functional Requirements

As described in section 2.2, Non-Functional Requirements (NFR) often impose constraints on the system. For example, security constraints or user interface constraints. They do not express a system behaviour, task or code generation. Within Web Engineering, these constraints would be similar to those of Software Engineering, but would need to be extended to reflect web design and development constraints such as user interface, usability, accessibility and technical server-side requirements.

NFR's in Web Engineering are sometimes only discovered after the project development has started. "The difficulty with articulating NFR's for Web system projects lies in identifying and predicting possible causes and impacts that NFR's have on the system and its domain. This is partially due to the uncertainty when the Web Developer does not understand the domain completely before building the Web system, leading organisations to make decisions without complete information" (Yusop, *et al*., 2006).

An opportunity exists in providing an NFR construct that helps the student to think about the various NFR's in respect of the web project that they are working on. A guidance system could be employed in a similar way to the functional requirements in order to encourage the student to a produce a consistent and complete set of NFR's. The NFR construct would need to consider the following:

1. User Interface (Screen size, navigation, text size).
2. Usability (Learnability, memorability and efficiency).
3. Technical (To include server platform, language support and database environment).
4. Marketing (To reflect search engine optimisation, metrics and conversion measurements).

2.9.4 Communicating Web User Requirements

One of the most important aims of RE is to effectively communicate requirements to all stakeholders involved in the project. As such, it is deemed important to design the meta-model in such a way as to provide flexibility in articulating the requirements. For example, in an NFR, a change in technology may result in the component being outdated quickly. An additional field, for example, the term '*non-standard requirement*' could provide a way of expressing these types of requirement within the construct.

For an FR, this may take the form a ‘notes’ field, where additional information can be provided to the stakeholder, either translating the technical language into an easily understood statement or the inclusion of a diagram or hyperlink to additional information.

2.10 Related Work Summary

The previous sections have provided a background to this research programme by outlining some of the problems facing Web, Software and Requirements Engineering. Important theory in respect of RE and SE sets the context for further work in terms of this research programme. This section provides a summary of related work which links to the review presented in Appendix A3. It demonstrates how each research cycle reflected the examination of existing approaches and integration of conceptual ideas found in these to the experimental method and framework.

Requirement Process, Methods and Tools

A requirements process is underpinned by specific methods and tools that are selected by the development team. Some methods reflect the whole requirements process (*elicitation, analysis, specification and validation*) and some address one or two stages of the typical RE process. For example, some focus on elicitation or the specification of requirements. It was found that some developers choose to combine methods in order to satisfy particular organisational or problem objectives, thereby creating hybrid methods suited to the organisation. The aim of this section is to demonstrate variations in the approaches that are in use by both academics and practitioners.

It was considered important to undertake a structured analysis of the methods by comparing their treatment against the requirements process as defined by Sommerville & Sawyer 1997 and Berry 2003.

Please refer to Appendix A3 Related Work - Table 3.3. This demonstrates how each approach addresses the whole requirements process criteria (*elicitation, analysis, specification and validation*). A discussion of how the review relates to the three action research cycles can be found below.

Relationship of Review to Research Cycles

Having reviewed a range of existing approaches a number of *conceptual ideas* emerged that provided the starting point for the design of a web user requirements approach:

User Modelling. *Define the target audience using an appropriate model that reflects their importance using profiling and classification models.*

- Primary and Secondary User Classification (UCA)
- User Profile (UCA), Person Profile (CI), Usage Scenarios / Persona (US), Actors (NDT) and Actors (UC)

Project Vision and Objectives. *Allows the developer to establish an overall vision and business objectives before defining functional/non-functional requirements.*

- Concept Vision Document (ARM), Business Vision (JAD) and Business Case (CRC)
- Business Requirements (MSF), Business Objectives (JAD)
- Requirements Generation (SSM/ICDT)

Task and Goal Association Model. *Describe what the users do within the web/software application by the Tasks they complete or by the Goals they want to achieve. An association model links these with specific users.*

- Task to Interface Object Association Model (OVID), and Tasks (CRC)(TBAS)(UC)

Computer Aided Web Engineering (CAWE). *Automation of a rules model. Compel the student to complete every aspect of the meta-model. Check correctness of associations and consistency of requirements. Conformance to the rules model represented in the student dashboard, with visual cues to indicate completeness of the process.*

- WebRatio and FlashWeb in the way it supports the developer to model aspect of the website before implementation commences.

The 'electronic Web User Requirements Framework' (eWURF) embodies work undertaken in three research cycles, where ideas evolved and changes were made in response to in class observation. Feedback from the students via module surveys and also indirectly from delegates at the conferences that were attended all played an important role in shaping the method and overall framework. It must be emphasised that review of related work was undertaken across a period of time, as demonstrated in Chapter 1, section 1.7.

Figure 2.10 shows how the review of related work maps onto the three research cycles.

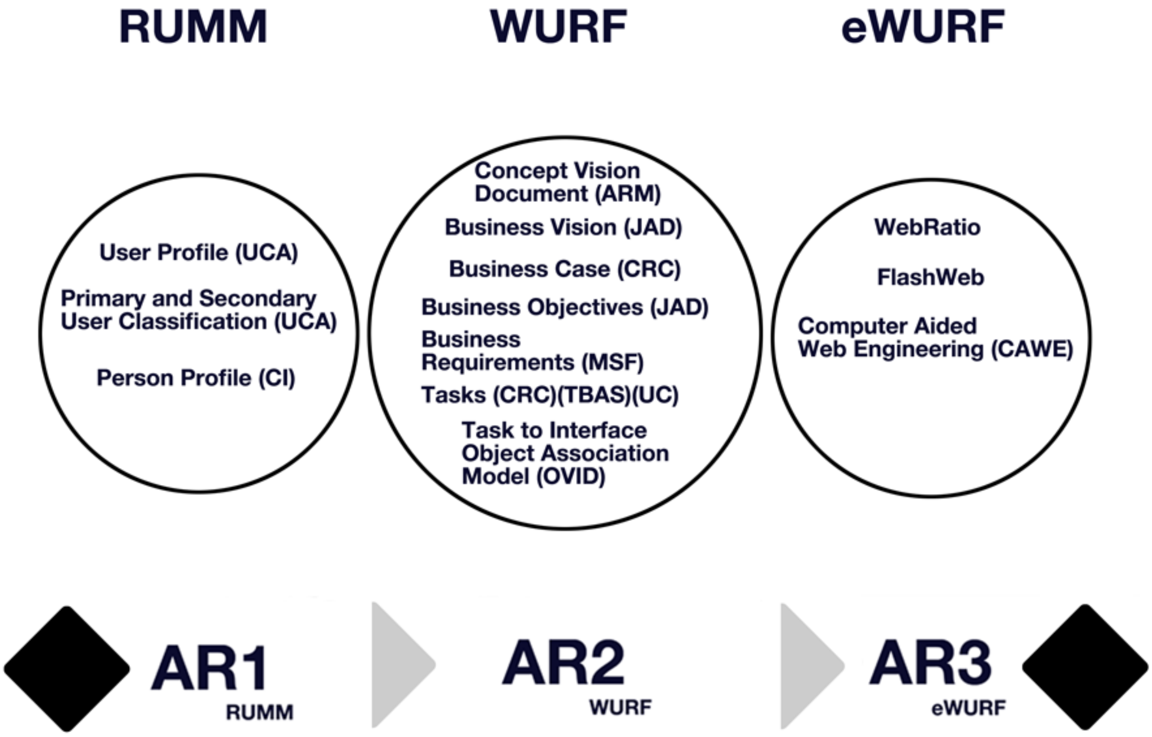

RUMM
WURF
eWURF
User Profile (UCA)
Primary and Secondary User Classification (UCA)
Person Profile (CI)
Concept Vision Document (ARM)
Business Vision (JAD)
Business Case (CRC)
Business Objectives (JAD)
Business Requirements (MSF)
Tasks (CRC)(TBAS)(UC)
Task to Interface Object Association Model (OVID)
WebRatio
FlashWeb
Computer Aided Web Engineering (CAWE)
AR1
RUMM
AR2
WURF
AR3
eWURF

Figure 2.10 How review of related work maps to the research cycles.

A number of existing requirements process, methods and tools are presented in Appendix A3. It is clear that from the review that: ARM; US; AMSF; and NDT address the whole requirements process, as defined by Sommerville 2007 (see Chapter 2, section 2.2.1). Five methods: NDT, AWARE, URN, SOARE and SSM/ICDT are aimed at projects that involve Web Engineering. These are characterised by a modelling technique that enables the web developer to draw out requirements based on the business vision, objectives, the user tasks or goals and before proceeding to define both functional and non-functional requirements.

JAD, AMSF, CRC and SOARE require the development team to draw out business objectives and to link these with interactions, tasks or functional requirements. Some use a modelling approach in eliciting and defining the requirements, such as URN, AWARE and NDT. Many approaches define the user of the system as an ‘Actor’, with various methods employed to show the importance of the Actor within the system.

In the context of a web user requirements approach aimed to support the student, it is felt that techniques that use field research, which involves the collection of primary data, would result in poor adoption and usage. This is due to students having little in the way of resources (time, budget and networks) in order to realistically achieve this. One way around this is to write a briefing document, for example, as part of the ICA that contains all the key information needed to start the elicitation process. The tutor could act as the client and user within a simulation exercise within the laboratory.

The student would therefore still proceed with elicitation, without having to construct questionnaires and carry out a survey. It was found that researchers had already debated the safety of generating requirements based on “intelligent guess work” (Cato, 2001 and Szekely, 1994). This was interesting, as any approach developed from this research programme will involve making informed decisions regarding the users. Their decisions would be informed by evidence found within the ICA briefing document and as well as the tutor acting as a user. Research suggests that it is plausible to generate requirements without directly questioning the target user. “Collect facts if you have them, or make reasonable guesses because even a reasonable guess provides a focus” (Cato, 2001). Cato encourages the use of this approach where it would prove impossible or difficult to undertake surveys to elicit requirements from the user. A technique defined by Szekely as ‘fast prototyping’, may also provide a way forward. “This approach facilitates elicitation, validation and revision through *discovery* of requirements. The discovery stage involves the production of a small scale version of a complicated system in order to acquire critical knowledge required to build a full system” (Szekely, 1994). This aligns with the Agile Approach to web development, involving iterative cycles of implementation and testing until the application is fit for release. Iterative development is widely used by students in the development process and is one which students would readily identify with. This method relies on revision or iterations which may prove valuable and align with the students' development practice.

Many of the existing approaches had gaps in the treatment of user requirements, notably OVID, TBAS and CI. Web specific requirements approaches such as MSF and NDT did meet the criteria, as did ARM and US. Approaches outlined in Appendix A3, Table 3.3 can be further characterised by:

1. Stakeholder involvement in the requirements elicitation process.
2. Detailed descriptions or profiles of the user often referred to as Actors.
3. The separation of functional and non-functional requirements.
4. Use of a meta-model to help define associations and dependencies.
5. Use of natural language to describe 'user journeys' or 'scenarios' or to map business objectives with tasks, features, behaviours and goals.

2.11 Summary

This Chapter has examined Requirements and Web Engineering and its influences from Software Engineering. Notable methodologies have been identified and implications for addressing requirements outlined, including a set of guidelines for the review of existing methods. The teaching of Web Engineering has been identified as an area of concern and a gap identified in the treatment of web user requirements.

Software Engineering has influenced the philosophical approach to Web Engineering, however most authors set out clear distinctions between the two. It is claimed that these differences are about the way web applications deliver their content and demands from those who commission them. In turn, this has led to subtle but significant differences in process and methods. The discipline is moving away from highly ordered processes to more agile methods, reflecting practice and therefore challenging orthodoxies. The latter has certainly led to differences of opinion and changed the way that Web Engineering is taught within HE. Problems are still apparent in Web Engineering and requirements analysis has been identified as an area of concern. In particular Lockyer *et al.,* 2003, identified that web courses concentrate on implementation, at the expense of analysis. McDonald and Welland, 2001 are also of the opinion that Web engineering needs to focus more on analysis, specifying requirements and testing. There is a need to investigate an approach that can specifically address web user requirements in the context of Web Engineering and to support the inexperienced student user.

Much of the literature focuses on ‘software’ rather than ‘web’, pointing to a gap in knowledge in this area. RE does provide a number of important principles which need to be taken forward when thinking about the design of a web user requirements process. These include:

1. A process that is transparent, logical and repeatable.

2. The ability to support the student through elicitation, analysis and specification process.

3. To reflect agile development methods that are adopted by the student, including the ability to refine and append additional requirements throughout the web project.

4. To provide a mechanism to establish a set of functional and non-functional requirements in a natural language, expressed as ‘high’ or low’ level in terms of detail.

5. To ensure requirements are *consistent, complete* and *correct.*

6. To enable requirements to be 'traced' through to the website artefact.

7. The ability to store, analyse and output requirements in a specification document.

Much of the literature concerning existing requirements approaches focuses on 'software' rather than 'web', pointing to a gap in knowledge in this area. RE does provide a number of important principles which need to be taken forward when thinking about the design of a web user requirements process. These include the transparency of the requirements process, where this should be logical and understood by the student together with the ability to produce valid requirements. It should also reflect agile development methods adopted by the student, including the ability to refine and append additional requirements throughout the web project period. It has been argued that web requirements are distinct and require an alternative treatment within the web user requirements method to that of software focused methods.

The investigation has so far focused on requirements within Software and Web Engineering, with little attention paid to the educational or learning aspects. Problems associated with the teaching of RE within WE will be discussed in Chapter 3. It is recognised that in order to support the student effectively in their web user requirements process, a suitable learning model must be adopted for each intervention in their learning. In Chapter 3, we investigate learning theory and propose a hybrid learning model to support the student.

Chapter 3 – Supporting Student Learning

3.1 Introduction

The aim of this section is to investigate how to incorporate a suitable model to underpin and support the students learning of web requirements. A learning model for teaching RE in WE is proposed which reflects the need for the learner to understand the whole Requirements Engineering process. The model proposed moves away from the conventional wisdom of RE as an 'effortless scientific process', where problems and solutions are discovered easily, to one that reflects real world 'unpredictability' and 'ill-structured problem definitions'.

3.2 The Problem of Learning and Teaching of RE in Web Modules

A problem has been identified in teaching Web Engineering by Griffiths and Lockyer, where the requirements and analysis stages are being neglected within the curriculum. "It is our contention that early lifecycle activities are also neglected within computer science education. We believe that there is often an over emphasis on the later stages of the development lifecycle – in particular programming" (Griffiths and Lockyer, 2004).

Reasons for this could be due to the focus on implementation, programming and technology, along with an over assessment of the 'artefact' rather than the process. Web development requires the application of knowledge and an in-depth understanding of client-side and server-side technology, which takes longer to learn. It can be classified as a knowledge intensive discipline, in that in order to successfully understand the discipline, the learner must put theory into practice.

It has already been suggested that Web Engineering is a separate discipline to that of Software Engineering, as it entails an understanding of design, usability, accessibility and server-side/client-side technology. In order to fully understand these topics, modules tend to provide the student with a ‘front loaded curriculum’ focusing on technology, at the expense of requirements and analysis. A gap therefore exists in the teaching of RE in the context of WE.

The problems identified in teaching RE are not new. Connor, *et al.*, 2009, have identified through their own investigation into effective teaching practices within RE that “Requirements Engineering is not taught to any depth in many universities. Students have only some vague knowledge through Software Engineering. Hence there is a lack of well trained requirements engineers” (Connor, *et al.*, 2009).

Teaching RE also requires an understanding of both its theoretical concepts and the application of that theory to a real world problem. RE requires knowledge and understanding of a wide range of interrelated subject areas and involves problem solving skills, analytical skills, system modelling skills, technological understanding and social skills required when working with stakeholders. “Practitioners and student’s, are seen to need conceptual knowledge in several overlapping domains in order to perform Requirements Engineering tasks successfully” (Armarego, 2007).

In a round table discussion at the 16th IEEE International Requirements Engineering Conference in 2008, the panel concluded that “unfortunately, many of these skills can simply not be learned by sitting in a classroom and listening to a lecture, or by performing an exercise at a computer. Furthermore, the face of today’s student’s and even the classroom environment is changing dramatically as an increasing number of universities offer distance learning opportunities" (Zowghi and Cleland-Huang, 2008).

In order to bridge the gap between RE and WE within higher education and for more effective teaching practice it is vital to view the whole picture. For example, within the author's own experience of teaching web development, a number of issues can be attributed to the students' inexperience, exhibited by specific behaviour traits in their approach to learning. Problems in teaching RE within WE, can be compounded by the diversity of the student cohort. For example, students will have backgrounds in diverse subjects such as design, business, marketing or information technology. These students tend not to have had any formal computer science teaching and therefore may not have been exposed to modelling languages that facilitate analysis. Some students are experienced in consuming web applications, rather than implementing them. Therefore students have preconceived ideas of what they wish to implement, rather than focusing what the user wants, articulated through the RE process. They are also inclined to implement too quickly rather than undertaking analysis activities first.

There are also pressures on students to work part-time in addition to study, resulting in more self directed learning taking place away from the traditional lab / lecture setting, requiring changes to teaching and learning strategies to better support them. Some students are reluctant to take control of their own learning, preferring a 'spoon-fed' learning experience. Students expect answers to be accessible in tutorial booklets rather than discovering the solutions themselves, leading to a lack of academic curiosity. It is hoped that this practice can be challenged by making changes to the traditional learning and teaching strategy underpinned by an alternative learning model. The learning model will support the overall intervention to the students' practice.

3.2.1 Teaching Web Engineering

The author has been teaching web development for ten years at a Higher Education institute (HEI). The curriculum within the modules has tended to concentrate on the technical aspects, using specific technologies, rather than providing opportunities to engage with aspects of Web Engineering that are concerned with critical thinking and problem solving. Specifically the early stages of the Web Engineering process have been neglected or not taught correctly, for example, focusing on the design documentation to the detriment of user requirements. Web Design has been the focus of the curriculum on many modules, with the result that students do not come to terms with the full 'engineering' process involved in producing an enterprise level web application. Moreover, incorrect methodologies have been taught in place of more contemporary web specific engineering processes. For example, until recently SDLC was still used on many modules, even though the student was actually following an entirely different approach. Lip service is therefore paid to the SDLC by the student, whereas it is not actually used in practice.

Attempts to address this at Teesside University have taken place, most notably by (Griffiths , *et al.*, 2002) who pioneered a new web development methodology named Simple Web Method (SWM). Tools to support the student in understanding the domain of Web Engineering include CASE tools such as PAWS (Project Administration Web Site) (Lockyer, *et al.*, 2003). This provides an opportunity for the student to use a systematic process and helps develop 'problem solving' skills that can be transferred from project to project and to new problem domains. Whilst this is true, a gap does exist at the requirements analysis stage, where its treatment is not explicitly defined and is largely left open to interpretation regarding an approach. This gap can be fulfilled by proposing a student focused framework to help them elicit, analyse, specify and document requirements for integration into a development methodology.

3.3 Theories of Learning

Students learn in different ways, processing and assimilating information and making their own connections between experiences they have had in the past in order to generate new understanding. Teaching styles also differ in approach, for example, some tutors prefer using a lecture, some prefer to demonstrate and others allow the student to discover things for themselves. Sometimes there is a mismatch between these learning and teaching styles, resulting in some students achieving better results than others.

In HE, there is evidence to suggest that academics are moving away from traditional ‘instructional methods’ towards those ‘learner centred methods’ that encourage critical reflection, allowing the student to discover and construct knowledge for themselves. This approach allows them to make discoveries for themselves and solve real world problems. “An influential paper published by Barr and Tagg in 1995 entitled ‘From Teaching to Learning: A New Paradigm For Undergraduate Education’ strongly advocated the need to move from what the authors termed the traditional ‘instructional paradigm’ with its focus on teaching and instruction to a ‘learning paradigm’ that enables student’s to discover and construct knowledge for themselves. Barr and Tagg (1995) present some powerful arguments to support this shift towards an environment in which students are empowered to take responsibility for what they learn (guided by explicit learning outcomes that clearly link to assessment)” (Maher, 2004).

3.3.1 Behaviourism

The behaviourist paradigm operates on the principle of ‘stimulus-response’, in which behaviour is governed by a response to an external stimuli or event. It assumes that the learner is a passive participant in responding to the external stimuli and advocates the theory of reinforcement as a key mechanism in the learning process. Many models of learning are based on behaviourism, for example, it was previously quite popular to learn multiplication tables by repetition and evidence for learning could be observed and measured with an exam paper.

Behaviourism is an instructional method of teaching that is mainly transmissional in nature and does not encourage learners to take control of their own learning. It is appropriate for learning facts and figures, but does not lend itself to a model of learning that encourages the development of transferable skills such as problem solving. Models that use behaviourism are not thought appropriate to WE or RE where learners need to generate ideas and apply their understanding and experiences to new problem domains.

3.3.2 Constructivism

A constructivist approach to teaching is learner centred and one where "the learner constructs their own knowledge from their own activities, building on what they already know" (Biggs, 2003). In a computer science field, students' activities tend to be practical in nature rather than purely theoretical, with the instructor being the *facilitator* rather than *instructor*. The theory advocates that for learning to take place learners must draw from their previous experiences of learning and re-encode it, so that they can make connections with previous knowledge. It encourages a deep approach to learning where the learner can use previous knowledge to solve new problems presented to them. "Learners themselves will be more flexible, transferable and useful than knowledge encoded for them by experts and transmitted to them by an instructor or other delivery agent" (Cobb, 1999).

In order for the student to benefit from this model, they must first have a foundation of knowledge on which they can facilitate further learning. In section 2.6.1, it was identified that a problem exists in the curriculum of the author's web development modules, in that where the focus is on implementation, rather than problem solving. Without a foundation of understanding, it could be argued that students would find themselves floundering as a result of not fully understanding how to solve the problem by applying a constructivist approach.

3.4 Models of Learning

Learning models describe the approach taken in the design of a unit of learning, whether this is a module or distinct learning object. The relevance to this research is that students are required to learn RE in the context of Web Engineering. As such, an effective learning model that underpins the development of the web user requirements method would provide a sound process for the design of an effective learning environment. Current trends in learning theory in computer science exhibit a move away from behaviourist models, to constructivist centred approaches. The role of the tutor is based in facilitations, collaborating with the students themselves in their learning experience. A number of models lend themselves to teaching Requirements Engineering in a web orientated module. These are presented in the next section.

3.4.1 Experiential Learning

'Experiential Learning' provides opportunities for the student to relate theory with practice in order for them to generate their own understanding. "Experiential learning theory defines learning as the process whereby knowledge is created through the transformation of experience. Knowledge results from the combination of grasping and transforming experience" (Kolb, 1984).

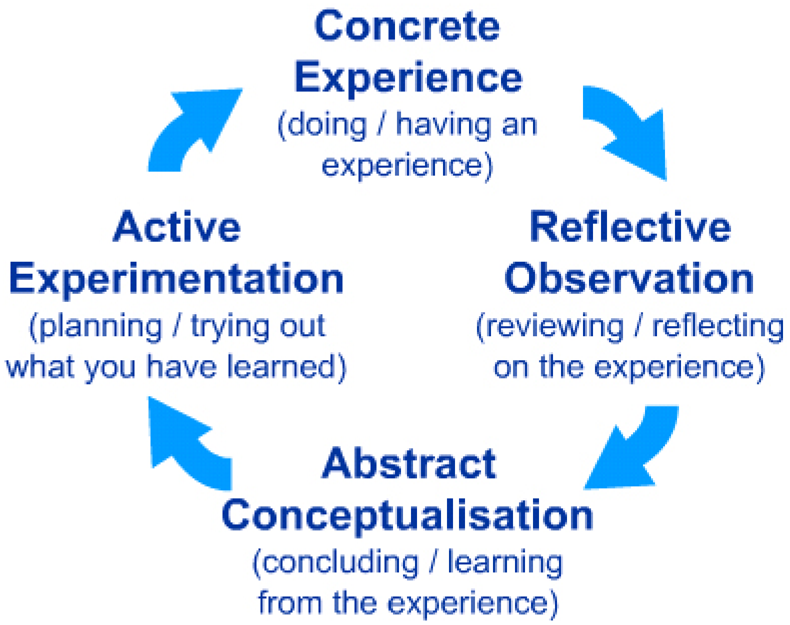

Concrete Experience
(doing / having an experience)
Reflective Observation
(reviewing / reflecting on the experience)
Abstract Conceptualisation
(concluding / learning from the experience)
Active Experimentation
(planning / trying out what you have learned)

Figure 3.4.1 David Kolb's Experiential Learning Cycle (Kolb, 1984).

Kolb's principle theory is based on a cycle of learning, (see Figure 3.4.1), which is based on immediate or concrete experiences, which in turn provide a basis for observation and reflection. Some abstract conceptualisation of learning must take place before completion of active experimentation. The approach is useful when thinking about the way in which RE and WE can be taught at HE. Experiential learning is considered to be 'reflective learning' using multiple iterations of experiences in order to solve problems. In each learning cycle, learners draw upon their experiences from earlier cycles, reflecting on what they have learned and applying this to complex problems.

This model may suit RE and in particular WE, where an iterative approach to development is taken.

3.4.2 Laurillard's Conversational Model

Laurillard's learning model is focused on the use of technology in higher education. The model's principle theory is that learning takes place with other people, for example, tutors and peers and that their opinions, arguments and experience all play a role in shaping one's own knowledge and understanding. It also emphasises that interactions must take place between the learner and the tutor, along with their experiences and theoretical concepts. The model advocates experiential learning, specifically a situation where interactions between the tutor and subject take place. For example, where the tutor and learner collaborate together to form new understanding of a problem via a message board within a VLE.

3.4.3 Problem Based Learning (PBL)

Constructivist and behaviourist theories of learning underpin most models of learning where the theoretical body of knowledge is transmitted to the learner (behaviourist) and reliance in placed on memorisation or where the learner is more autonomous in their learning (constructivist). There is still an element of transmission and memorisation in the learning models that use their theory, which may not be suitable in an engineering domain. In both theories, information is imparted up front, using a structured method. For example, learning takes place in sequential order via a module and tutorial plan. The learner is then expected to 'do something' with the information so that they fully understand it.

In PBL, this traditional model is turned on its head. Instead of transmitting information to the learner, the learner is presented with a problem to solve. Analysis of the problem is the first stage in the learning model. Students determine what they need to do to solve the problem, propose a solution and disseminate their findings to others. “PBL is a way of constructing and teaching courses using problems as the stimulus and focus for student activity. Problem-based courses start with problems rather than with exposition of disciplinary knowledge” (Boud and Feletti, 1998).

PBL exhibits similarities with other models. For example, Laurillard’s Conversational Model is based upon interactions with other learners and the tutor. Their arguments, opinions and solutions shape the learning of others. This is similar to the last stage in PBL, where learners share their solutions with others and where it is evaluated by the same people.

A number of PBL approaches exist, each varying in style. For example, some PBL approaches use a structured framework, where students progress through a defined curriculum in order to check progress. Others use a ‘real life’ project to simulate a problem, where the student is prompted to ask questions of the facilitator in order to solve the problem.

PBL is not without its own set of challenges. The author has experimented with PBL models of learning on a second year undergraduate web authoring module. The model was adopted to teach one aspect of the module that was thought to lend itself to this approach. Adherence to the principles of PBL (facilitating and mentoring) meant that no information was imparted to the students in class and they were presented with a problem to solve. They were encouraged to ‘buddy up’ with a partner in the class in order to solve the problem. The author adopted the role of facilitator, stepping back from his usual role as a teacher and provided only low level responses to questions that were posed by the students in a mentoring role.

It was observed that some students found it difficult to engage with PBL, but others were liberated, often seen working outside of class. Some reported that they wanted ‘tutorial booklets’ and others did not know how to manage their own time effectively. A number of students called for direct instruction. The ‘active learning’ aspect of PBL was too much for some learners, so they switched off from the process entirely.

PBL is not used widely in the author’s department and it could be that as these students were not used to a PBL model of learning in other modules, when it was used in isolation, it caused them problems. Some students preferred the traditional model of learning, where theory is transmitted in a lecture and they then followed up by experimenting with theory in the construction of an artefact (experiential learning).

Positive aspects of the PBL model include its focus on solving real world problems. This matches RE, in that the solutions to problems are not going to be realised easily. It is thought that this aspect of PBL could be used in the teaching of RE in WE.

3.4.4 Blended Learning Model

A number of definitions exist that help to explain blended learning, for example, Procter *et al.,* 2003*,* define blended learning thus “Blended learning is the effective combination of different modes of delivery, models of teaching and styles of learning” (Procter, 2003). The Department for Education (DET) defined blended learning as “Learning which combines online and face to face approaches” (DET, 2003).

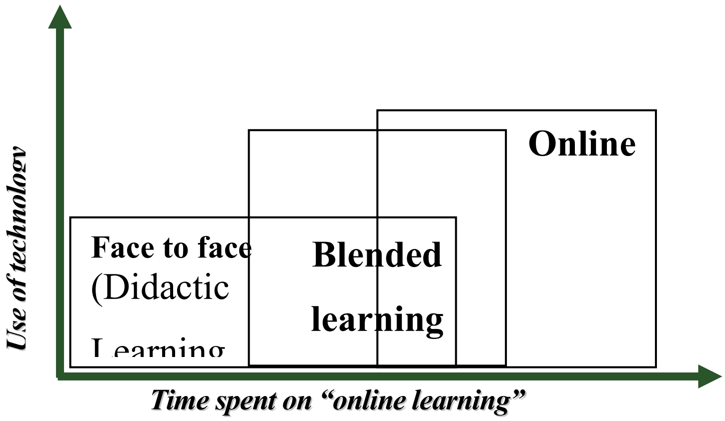


Figure 3.4.4 Conception of Blended Learning (Heinze and Proctor, 2004).

It advocates that ways of learning should vary, determined by what is being learned and the types of learners that are resident on the module. The model is especially useful when technology is employed to facilitate learning or where e-learning methods are employed.

E-learning is of particular importance to this investigation, since the web user requirements method will be tested within an e-learning environment. An opportunity exists to enhance the learning model and to support the student. For example, the method could include electronic support and guidance, with examples of completed requirements, as well as student support via a help system. It would also facilitate non-face to face teaching, where some learners prefer to learn in their own spaces and in their own time. It would permit students to re-visit content as often as necessary, reflecting the fact that some students learn faster than others. It would also be possible to track students' progress, providing the tutor with a rich picture of the student cohort. A 'dynamic dashboard' could display completions of the method in real time. This would allow early interventions to take place if students were not completing their work. The same system could also provide feedback to the student regarding their completion of the requirements, including if they were completing the requirements more slowly than the rest of the cohort. This aspect is covered in more detail within Chapter 7, where an approach is tested to provide visual feedback to the learner regarding the completion status of their requirements.

3.5 Assessment

An effective assessment can measure, at a given point in time, the students' knowledge and understanding of a given assessment criteria. “Assessment is at the heart of the undergraduate experience. Assessment defines what students regard as important, how they spend their time and how they come to see themselves as student's and then as graduates” (Brown and Knight, 1994).

When we think of assessment in the context of Higher Education, measurement alone cannot best describe its aim and purpose. Assessment is an intricate component of the design of effective learning. There are many reasons why we actually carry out assessment which can usefully be split into three distinct spheres.

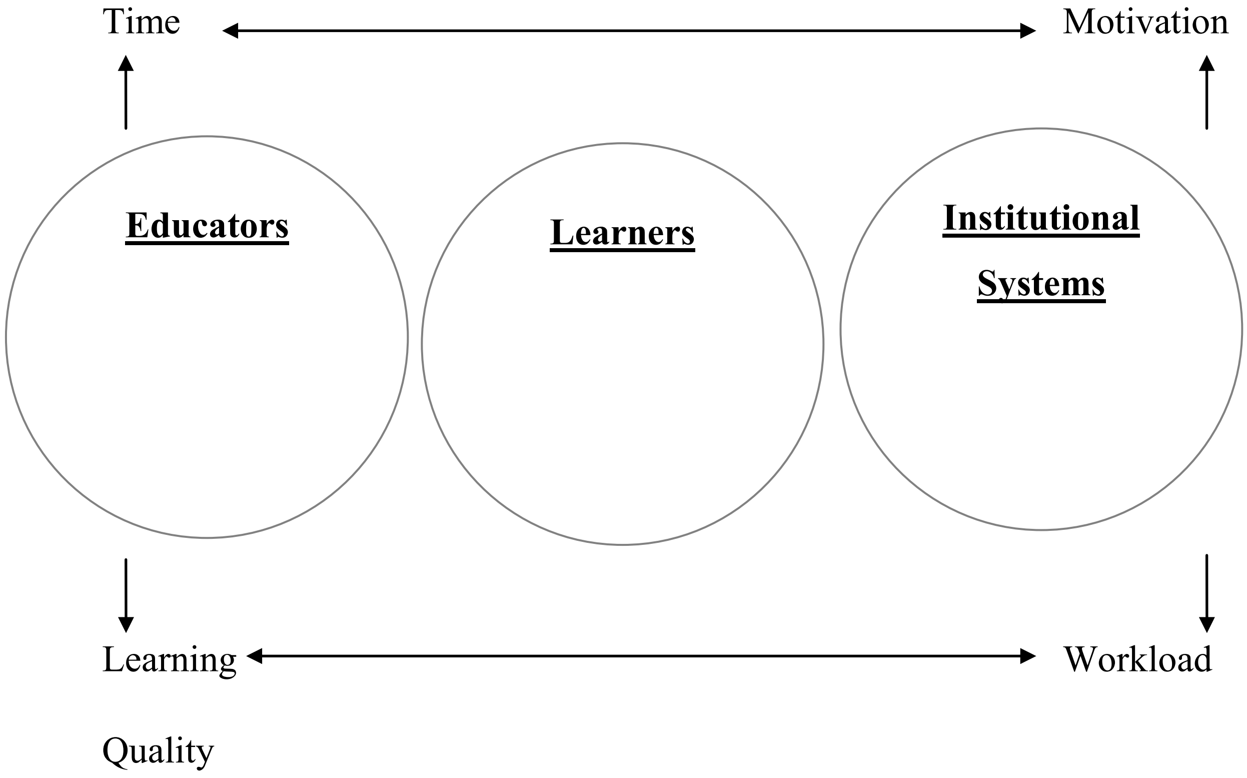


Figure 3.5 Aims, Purposes and Forces of the Assessment Process.

Figure 3.5 is based upon ‘purposes of assessment’, (Brown and Knight, 1994) and adapted to show the interrelationship and forces that are at play within the assessment process. Whilst there are huge motivational factors behind assessment, there are also hidden forces, attempting to ‘pull apart’ the fabric of the assessment process. At the centre of the process is the learner sphere, where it can be argued that assessment is part of learning. Students expect assessment and are motivated by it throughout the duration of a module. Assessment also provides feedback and encourages critical reflection.

The educator is also motivated by assessment, but for different reasons. The educator is motivated to assess by the learner and by the institutional systems that consume the results. It can be seen as a basis for the design of learning, given the need to deliver an effective learning experience with a measurable end product. Whilst this is true, there are a number of forces which act on the educator’s ability to assess, namely workload and time. The interrelationship between these factors and the learner cannot be underestimated, given the increased number of learners and what they require from the assessment process, such as individualised feedback.

Institutional systems consume data produced by the learner and educator. This is required for degree classification, student/institutional performance indicators required by internal and external parties, such as progression boards and employers/governments. Systems demand validated data, based upon the outcomes of a programme of study. They are motivated by averages, means and the distributions of marks. This underpins the perception of ‘quality learning’ purely from a statistics point of view, which may conflict with the point of view of learners and educators. For example, if a programme of learning has enhanced knowledge and understanding and which is then reflected in a set of results, systems will call into question the educators marking. Systems always require averages that are fixed to a certain level, which is usually set at an average mark of fifty six percent. This can de-motivate both educators and learners as marking is sometimes skewed to fit exam board requirements.

It can clearly be understood that assessment is far from a straight forward process. There is a danger that it can be diluted by external and internal forces, until the measurement no longer becomes 'safe', in the context of the learning process. To ensure that tutors assess effectively and that all stakeholders needs can be addressed, the assessment strategy needs to be a pivotal aspect to the design of learning. "An alternative view has emerged in schools and higher education, namely that 'Students assessment is at the heart of an integrated approach of student learning" (Knight, 2002).

How RE is assessed is therefore important in the design of a web user requirements method. For example, does one assess the process of RE - the actual words written for a functional requirement or does one assess the traceability of the functional requirements within the artefact? The learner could produce a requirements specification that is consistent, correct and complete, but with no relationship to the end artefact. The assessor would then provide appropriate feedback for the validation process and separate feedback for the artefact, which would be conflicting. It is therefore deemed essential that consideration is given to both the process and artefact when assessing RE.

Many learners are assessment driven in their learning, often focusing on what needs to be done to get a high grade. It can be argued that this approach results in a narrow field of learning, rather than a breadth and depth of understanding that is expected at HE level. In order to mitigate this, attention must be paid to the weightings in the assessment process of RE. For example, from the author's personal experience an assessment criteria of five percent is sometimes ignored by the student and instead focus is shifted onto the higher weightings where it provides the student with greater reward.

In addition, there is also an opportunity to enhance the teaching of RE in web modules by the adoption of formative assessment opportunities. The traditional teaching and assessment cycle is one of teaching, summative assessment and feedback at the end of the module. It can be argued that feedback at this stage is not going to be beneficial to the student, as they have finished the module. Reflection that takes place six weeks after the assessment has been marked diminishes the quality of learning as it delays the students critical reflection.

By designing the learning model so that formative assessment opportunities are integrated at key points feedback can be provided part way through the module, rather than at the end.

3.6 Proposed Learning Model

The proposed learning model that will underpin the intervention, (see Figure 3.6), draws upon the learning theory described in the previous sections, especially relating to PBL and constructivism. Within the module a facilitator provides instruction and guidance as required. The model reflects the need for underpinning theory as a starting point to help put the problem in context and identify resources available to solve the problem. A problem definition in the form of an ICA brief leads the leaner towards greater autonomy in their learning. It also reflects the need for experimentation to occur early in the process in order for the learner to make mistakes in a safe environment. Experimentation occurs in iterations that suits their learning style. The experimental stage allows the student to get to grips with the learning environment and so that they can explore and discover how this works for themselves. This is often referred to as ‘surface learning’ in the literature and although the learner might be able to describe the requirements process from memory, they are unlikely to posses the transferable problem solving skills required in later stages of the model. Tutors' aim for students to learn skills at a much deeper level (deep learning) than simply being able to recall information.

Active learning involves the student using their previous experiences in trying to solve problems. Again, this may take one or several iterations depending on the learning style of the student. Active learning involves the use and consultation of a range of resources, for example, the tutor may facilitate a question and answer session. External sources of information may also be used, for example, journals, books, websites and primary sources of data such as user questionnaires.

Reflective learning refers to a situation where deep learning is expected to occur. Deep learning is a state in which through the students' previous experience, greater knowledge and understanding can emerge. Peer learning and formative feedback from the tutor, all play an important role within the proposed learning model

Assessed learning is a situation in which students finalise the solution to their problem, via the Requirements Engineering process. As discussed in section 3.5, students are motivated by assessment and this can be used in the effective design of their learning. Evidence of their learning will be resident in the requirements document and website, with traceability between the two. As such the process can also be assessed.

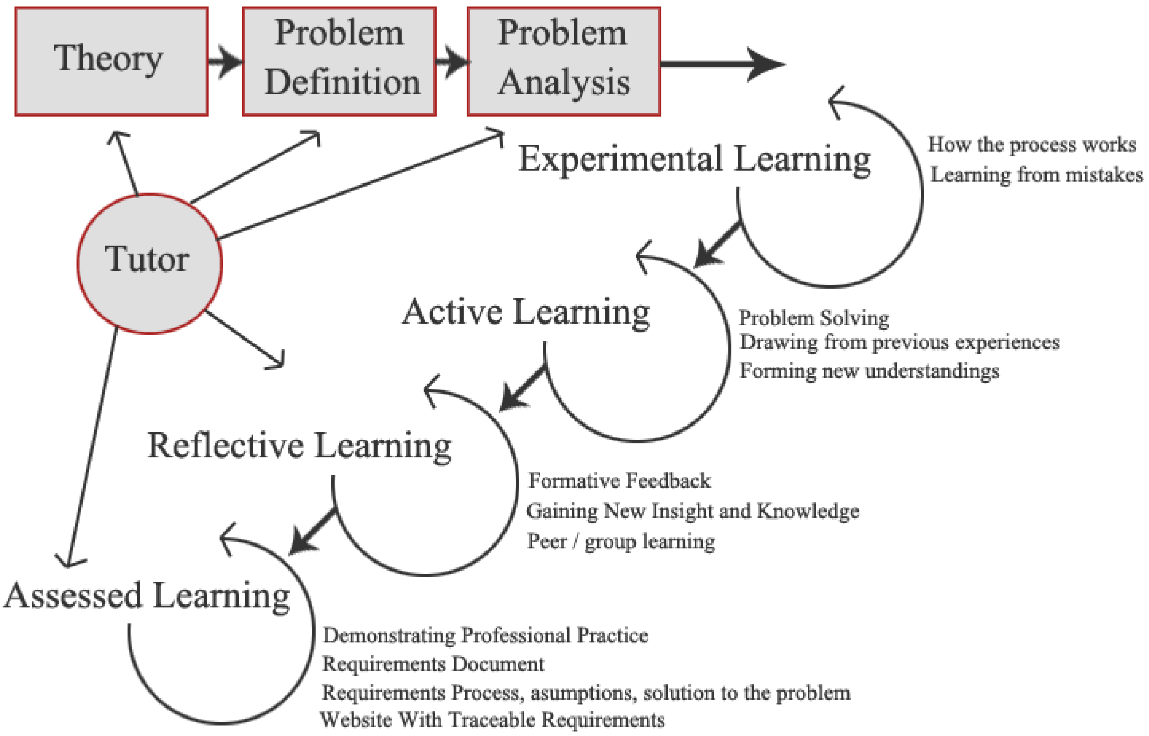


Figure 3.6 A Hybrid Problem Based Learning Model.

3.7 Summary

A problem in the teaching of RE and WE has been identified and a learning model proposed in order to underpin the intervention as a whole. As the web user requirements method and framework is aimed at students studying Web Engineering, it is important to recognise the role that learning and teaching practice will have on its successful adoption within the curriculum.

The hybrid PBL model proposed in Figure 3.6 reflects the needs of the cohort of students studying web design and development modules. Its aim is to encourage deep learning and problem solving skills that, once learned, can be applied to situations which demand solutions to new problems. This is an important dimension given the need to discover, elicit and analyse web requirements in particular. The learning model will be adopted as the basis for lesson plans and to support the student within the laboratory whilst using the web user requirements method.

The next chapter is concerned with the research paradigm and in particular the research method that will be adopted for the programme. Consideration for the setting of the research programme is discussed, with this being focused on educational research. Strengths and weaknesses of the approaches are also documented.

Chapter 4 Research Approach

4.1 Introduction

The preceding chapters have described current state of the art thinking regarding RE and WE. The main aim of this investigation is to "make a contribution to the discipline of Web Engineering and Requirements Engineering and to support the inexperienced student user." This chapter describes and justifies a research philosophy and methodology that has been adopted for this investigation.

4.2 Philosophical Research Paradigm

Oates 2005 outlines the characteristics of various research paradigms, for example, positivism, interpretivism and critical research. The identification of a paradigm is seen as essential, as it underpins the research design, how one acquires knowledge and how one interprets and evaluates the results. Positivism is a scientific orientated paradigm, as it uses experiments to test hypothesis using results objectively collected from repeatable surveys. It uses quantitative data analysis with mathematical modelling and statistics to provide logical interpretations. It seeks 'the truth' and 'the proof' and makes generalisations based on the findings of the research. It is also concerned with empirical testability and replicability of experiments, hypothesis and theories. "The scientific and positivism were developed for studying the natural world, for example, in physics, chemistry and biology. Positivism is less suited to studying the social world." (Oates, 2005: p288).

Interpretivist research is concerned with understanding how the development of an information system or web application affects the social setting. People are recognised as having values and beliefs in a group or individual setting and these combine to develop a 'world view' or 'standpoint'. These views and standpoints influence the outcomes of the research and it is recognised that these can change over time or by the interventions taken in carrying out the research. For example, in an educational context the research findings may differ from cohort to cohort, even though the same process of intervention, reflection and evaluation has been carried out. Interpretive research is different from positivism research as its aim is not to prove or disprove a hypothesis. Instead it tries to ‘explore’ and ‘explain’ how the investigation outcomes are related and interdependent. It relies on qualitative data analysis, where data is generated and analysed based on ‘words’, ‘metaphors’ and ‘meanings’ of those involved in the study. Interpretative research also recognises that the research reflexivity, the cause and effect, will shape the research process. For example, “assumptions, beliefs, values and actions will inevitably shape the research process and affect the situation” (Oates, 2005: p292).

Critical Research sets out to challenge and question ‘taken for’ assumptions. These can prevail in systems of economic, political and cultural authority or within organisations. Researchers who prescribe to this view seek to highlight and confront sources of domination and alienation. The main aim is to reveal, criticise and explain how an established order or view point is affecting people within an organisation and also aims to empower stakeholders in transforming those viewpoints. Critical research is not aligned with specific methods and relies on the adoption of interpretivist methods such as; critical ethnography (Myers 1997, Thomas 1993), participatory action research (Baskerville, 1999), critical discourse analysis (Fairclough, 1995), which are all proposed to be distinctly critical. Dubravaka, 2007, explicitly highlights ‘participatory action research’ as exhibiting ‘traits’ of critical research. “Participatory action research can be also seen as a distinctly critical method to the degree to which it identifies specific critical concerns and focuses on practical intervention to address these concerns and transform practice (such as IS development)" (Dubravaka, 2007).

4.3 Educational Research

A number of learning theories have been outlined in Chapter 3, that have been born out of educational research. Educational research covers a disparate number of areas, such as investigating the behaviour of learners (students), educators or institutions. The majority of educational research focuses on monitoring educational quality, demographics of the student population and grade monitoring. It also includes educational changes, developments and interventions, such as the proposal of new ways of learning. A number of educational research methodologies exist, with the majority having their origins in the positivism paradigm, drawing from scientific methods, however it is increasingly evident that interpretivist paradigms are emerging, using largely social and behavioural science centred methodologies.

There is continuing debate as to whether educational research should adopt a *positivist* or *interpretivist* paradigm. For example, Rowbottom and Aiston, 2006, argue that the scientific method should not be adopted on the basis that it is the only valid approach. "Recognise that good inquiry-rather than 'doing science'-is what really matters" (Rowbottom and Aiston, 2006). A suitable 'method of inquiry' that is valid in terms of educational research must therefore be adopted for this research programme.

What is clear is that positivism and interpretivism are both valid paradigms, but that they see research from different perspectives. "Positivism strives for objectivity, measurability, predictability, the construction of laws and rules of behaviour and the ascription of causality; the interpretative paradigms strive to understand and interpret the world in terms of its actors" (Cohen, *et al.*, 2007).

Methodologies associated with the interpretivism paradigm reflect participant observation, role playing, collaborative investigation and the reflection of social interaction, with the researcher often becoming part of the process rather than being an objective observer. Some investigations in educational research lend themselves to specific methodologies. For example, a controlled experiment requires an environment where variables can be isolated, controlled and manipulated. This can suit a scientific method, grounded in the positivism paradigm. One that involves the observation of people in an environment that cannot be controlled in order to change a way of working or learning can suit an interpretivist paradigm.

Adoption of a suitable research methodology must reflect the nature of what is being investigated. The purpose of this research programme is to make a contribution to knowledge in the area of RE and WE by understanding relationships between the theory, the practice, the learner and the adoption of models that combine in the form of an overall intervention. A positivist view is that knowledge is there to be discovered by the researcher. If this is true a hypothesis can be defined and experiments designed to test it. The results should be the same if other researchers were to attempt the same experiment again. The results would therefore be valid and trustworthy.

There are a number of issues with this approach, the first being the ability to create an unambiguous and testable hypothesis in educational research and in particular this investigation. For example, even though the framework succeeds in situation X, this does not necessarily mean it will succeed in situation Y. This is due to the variance in environmental conditions, such as the cohort and teaching methods used, or indeed the tutor. Secondly, the knowledge that is being discovered is created and re-created by its participants. People behave differently in certain situations, making it difficult to retest experiments or approaches under the scientific method. Responses to the research will be different as people have different viewpoints and understandings of the same subject, making it difficult to filter out these factors in the analysis and interpretation of any data. Thirdly, the actual data that is being collected varies from quantitative to qualitative. For example, measuring access times in log data is classified as quantitative data whereas descriptions of beliefs, opinions and suggestions would be classified as qualitative data. By interpreting qualitative data it is possible to gain an understanding of a problem under certain conditions, however it is difficult to apply statistical analysis to this data or to make generalisations.

This investigation is undertaken in an educational context which by nature is a social discipline. It involves constructing rich understandings by an ongoing process of intervention, interpretation and reflection. It involves groups of people with perceptions, views, expectations and shared understandings that change over time. The researcher is participating actively in this investigation, delivering teaching in RE and WE and interacting with the students who are being studied. This raises possibility that the researcher will influence their thinking and actions in relation to what is being studied. This research programme can therefore be said to be underpinned by the interpretivism research paradigm, with a slight overlap with critical research, due to the participatory aspect of its approach. The selection of a suitable methodology therefore lies within the interpretivist paradigm.

4.4 Design Science Research

Design Science Research (DSR) is an approach that reflects real-world, relevant problems and where the research makes a significant contribution to the field in which the investigation is taking place. Hevner *et al.,* 2004, states that in a DSR methodology “knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artefact” (Hevner, *et al*., 2004).

The artefact can take on a number of different forms, for example, a model, method or a prototype application. "Design-science research must produce a viable artefact in the form of a *construct*, a *model*, a *method*, or an *instantiation*" (Hevner, *et al*., 2004). Hevner et al further defines these as:

1. Constructs (vocabulary and symbols).
2. Models (abstractions and representations).
3. Methods (algorithms and practices).
4. Instantiations (implemented and prototype systems).

Within DSR a number of authors propose that building 'innovative and creative systems' in itself could be considered 'original contribution to knowledge'. For example; March and Smith 1995 comment that building an innovative and creative system, including its *evaluation*, can be considered as a contribution to the research discipline in which it takes place. "Building the first of virtually any set of constructs, models, methods, or instantiations is deemed to be research, provided the artefact has utility for an important task. The research contribution lies in the novelty of the artefact and in the persuasiveness of the claims that it is effective" (March and Smith, 1995). Persuasiveness of its effectiveness can be found in the critical evaluation of the research.

In order to address the main research aim and objectives, this research is proposing a number of different models that will combine to produce an empirical 'experimental framework'. The framework will be tested and evaluated on a number of modules in an HEI. It will be make a contribution by bridging the gap in knowledge between Requirements Engineering and Web Engineering by proposing an educational focused web user requirements framework. The models that it contains and the instantiation will be tested and evaluated on a number of web based modules in a HEI.

4.5 Action Research

Action Research sits within the 'interpretivism' research paradigm. It is a generic term, but with a central premise of combining theory and practice in an iterative cycle of application and reflection in order for new understandings and knowledge to emerge. "Action research, as a method of inquiry, is founded on the assumption that theory and practice can be closely integrated by learning from the results of interventions that are planned after a thorough diagnosis of the problem context" (Davison, *et al*., 2004).

Contribution to knowledge is made by making connections between the emerging evidence between each cycle and where the understanding is used to inform the next cycle of *problem definition*, *planning*, *designing*, *testing* and *evaluation*. New research objectives often emerge from each cycle and this in itself can be thought of as original contribution, given that without undertaking the cycle, the new objectives would have remained undiscovered. Unlike positivist methodologies, the enquiry process develops throughout each cycle, without being constrained by an overarching hypothesis. Each research cycle has the potential for making a contribution to the field in its own right.

There is no overall homogenous 'action research methodology' and a substantial number of variations exist. For example, Checkland 1991, Baskerville and Wood-Harper 1998, Chandler and Torbert 2003 and Davison 2004, all propose action research methodologies. These are grounded in cyclical models where each cycle informs the next in terms of understanding and testing of new ideas. In its simplest form, a typical action research methodology can be broken down into four distinct stages, as depicted in Figure 4.5.

1. **Reflection.** (Initial problem or thematic concern)

2. **Plan.** (Examination of the problem or theme in order to further define it, proposal of possible solutions)

3. **Action.** (Application of the solution and changes to practice)

4. **Observe.** (Changes to practice are observed and results examined)

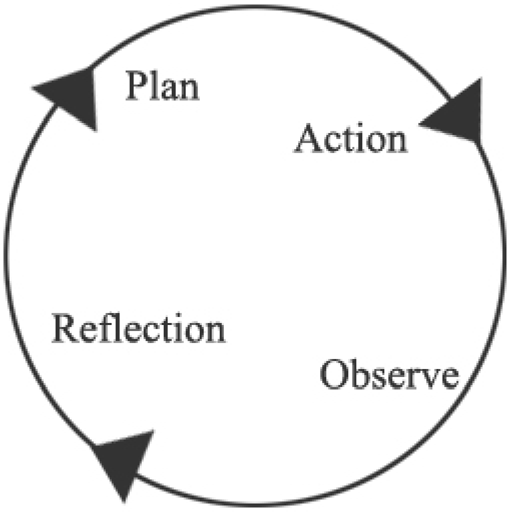


Figure 4.5 A Typical Action Research Cycle.

4.5.1 Dual Imperatives of Action Research

Checkland's 1991 model of action research is based upon the 'one cycle view' where *research themes*, *real world problem situations* and *reflections* are based on the framework and methods. Checkland's premise is to "suggest an alternative to positivistic research" (Checkland, 1991). Checkland uses a 'cycle' to describe the action research process.

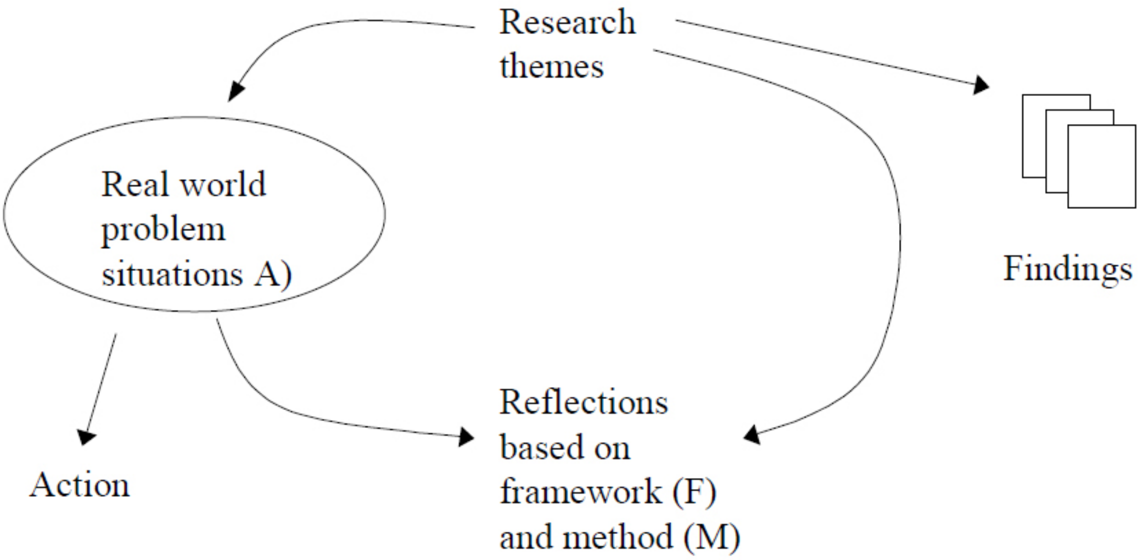

Research themes
Real world problem situations A)
Findings
Action
Reflections based on framework (F) and method (M)

Figure 4.5.1 The cycle of action research (Checkland, 1991).

Other variations of this approach claim that action research is composed of two interlinked cycles (dual imperatives) (see Figure 4.5.2). It comprises a problem solving cycle and a research cycle that operate in tandem. This duality seeks to separate theory and practice. The first cycle seeks to solve the real world problem and the second cycle deals with research questions, objectives or themes and helps identify problems posed at the outset and throughout each cycle.

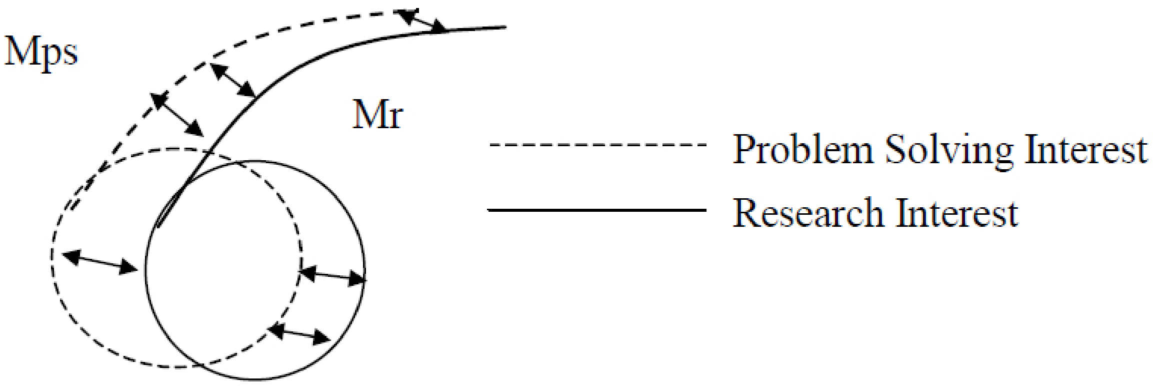


Figure 4.5.2 Action research viewed as a dual cycle process (McKay and Marshall, 2001).

4.5.2 Educational Action Research

Action Research is becoming increasingly accepted as a valid method for educational research. 'Educational Action Research' is founded on work undertaken by Dewey, who believed that educators should become involved in the research process, rather than being impartial observers. It offers a number of advantages such as its ability to reconstruct theory and knowledge in order to enhance practice and by challenging established beliefs. Kemmis and McTaggart also believe that action research can only be defined as such if the researcher collaborates in the investigation. "The approach is only action research when it is collaborative, though it is important to realise that the action research of the group is achieved through the critically examined action of individual group members" (Kemmis and McTaggart 1988).

Educational Action Research provides a valid methodological approach, given the positioning of the investigation within an educational context, where interventions to existing assumptions and values can take place more openly and where the researcher is part of the collaboration. It allows the educator to see changes in patterns of behaviour resulting from interventions that they have brought about. Observing patterns of behaviour and gathering evidence from those that are collaborating in the research will allow new knowledge to emerge, which in turn can provide the basis for new cycles of research.

4.6 Choice of Approach

It has already been established that a scientific method is not appropriate for this research programme, given its collaborative nature and the problems generating a valid and testable hypothesis under the positivist research paradigm. There are similarities between design science and action research, in particular the 'the real world problem identification; 'demonstrable solutions in an educational context'; 'cyclical approach to the research process'; 'mixed methods for obtaining and analysis of data resulting from the research' and the 'models and instantiations' used in the action part of the approaches.

However, DSR concentrates on the localism in viability of the theory, whereas this research hopes to make a contribution to the wider body of knowledge. Its focus on the artefact, rather than the wider underpinning theory also detracts from its adoption in this research. Within action research, it is acknowledged that contribution to knowledge is brought about by the experimentation and intervention of the research, rather than the construction and testing of the artefact.

The main principle of the researcher becoming an active participant in the research, making interventions, rather than being an observer, is perhaps the strongest argument for adoption of action research. In particular it is recognised that this research programme will reflect:

1. The interpretative nature of the data that will be analysed.

2. The social and collaborative construct of understanding that will emerge.

3. Emerging problem identification, rather than hypothesis testing.

4. Iterative problem solving process.

5. The researcher as an active participant in the research.

The Action Research approach that is to be adopted for this research programme closely addresses the framework proposed by Baskerville and Wood-Harper, 1998. Sometimes this is referred to as; 'Canonical Action Research' (CAR), cited by Davison, *et al*., 2004, that involves one or more iterations of diagnosing, action planning, action taking, evaluating and specifying what has been learned through reflection, for interventions to take place within the next cycle. The latter stage distinguishes the approach from other action research methods, as it is focused on learning as well as building on the knowledge and understanding of RE in WE. Davison, *et al*., 2004 propose five principles for CAR:

1. The Researcher-Client Agreement (RCA).
2. The Cyclical Process Model (CPM).
3. Theory.
4. Change through Action.
5. Learning through Reflection.

"Throughout each cycle, AR is focused on both organizational improvement and the generation of knowledge" (Baskerville and Wood-Harper, 1998). It is envisaged that a greater understanding of RE in WE will emerge in each cycle. The main aim of the research reflects both a contribution to the knowledge and understanding of RE in WE and the scientific knowledge gained by understanding by the creation of an overall framework and intervention.

It is recognised that action research is not without its critics. For example, some see action research as a 'consultation' rather than being a scientific research methodology. Heller, 1986, cite the lack of generalisability or external validity of action research, where the knowledge and understanding gained in a particular situation is hard to transpose to another situation (Heller, 1986). Other criticisms include the lack of impartiality, lack of basic research methods, ethical issues and that the research is context-bound, with the findings unique to this investigation and not generalisable. Frideres, 1992, is a strong critic of Action Research claiming that it misleads participants and does not generate knowledge. Others suggest that the principle threat to Action Research is validity in terms of the lack of impartiality and lack of proof. As discussed in section 4.2, these concerns come from the positivistic scientific community.

To counter to these criticisms, Baskerville and Wood-Harper, 1996, argue that "the rejection of action research as a method is rooted in the philosophical supremacy", (Baskerville and Wood-Harper, 1996), of those who reject its findings. The 'lack of relevance' can equally be applied to other methodologies that use qualitative methods, not just Action Research. Baskerville and Wood-Harper also highlight the differences in Action Research from consultation, noting that the "(i) researchers require more rigorous documentary records than consultants; (ii) researchers require theoretical justifications and consultants require empirical justifications; (iii) consultants operate under tighter time and budget constraints; (iv) the consultation is usually linear – engage, analyse, action, disengage – while the action research process is cyclical" (Baskerville and Wood-Harper, 1996).

In order for the action research to be classified as a scientific method, Susman and Evered 1978, in Baskerville *et al.,* 1996, propose five stages, which are cyclical in nature and are described as "an ideal 'exemplar' of the original formulation of action research" (Baskerville and Wood-Harper, 1996: p237).

Baskerville further illustrates this approach in Figure 4.6 and characterises it as "client-system infrastructure or research environment. Then, five identifiable phases are iterated: (1) Diagnosing, (2) Action Planning, (3) Action Taking, (4) Evaluating and (5) Specifying Learning" (Baskerville *et al.,* 1996). This approach differs from that of 'dual cycle' research, where the research and problem solving interests are separated.

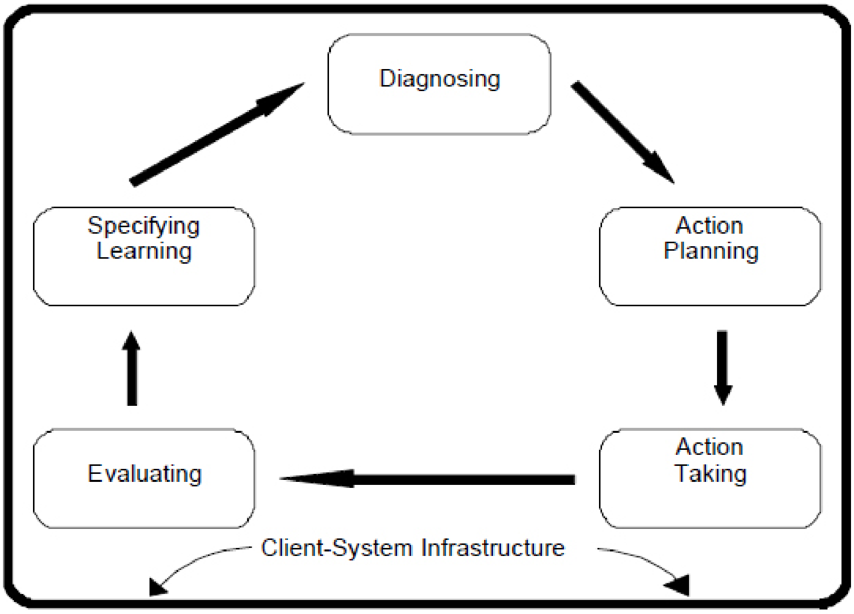


Figure 4.6 The Action Research Cycle (Susman 1985 in Baskerville and Wood-Harper 1996).

The client-system infrastructure can be defined as the agreement between the researcher and host practitioners or stakeholders, where boundaries and limitations of the research are expressed as the research environment. It also details how the research and learning will be disseminated.

4.7 Strengths and Weaknesses of Canonical Action Research

4.7.1 Strengths of Canonical Action Research

- Recognises the researcher as part of the programme, not divorced from it.

- Recognises the 'cause and effect' of researcher involvement given the context and setting of educational research.

- Allows for the evolution of ideas over multiple cycles of action planning, taking, reflection and evaluation leading to a greater understanding of the problem.

- Is responsive to the needs of the student, where changes can be made in light of problems, difficulties or ethical issues that affect the research programme.

4.7.2 Weaknesses of Canonical Action Research

- Transferability and generalisability of findings may be limited to the context and setting of the research. The validity therefore may be called into question.

- A perception of Action Research being 'less rigorous' than that of alternative methods set in the positivist paradigm.

- The possibility of becoming too personally involved in the programme and influencing its findings.

In this research programme, successive cohorts of students will collaborate in the research by using and evaluating an overall intervention over three cycles of research. By combining Susman and Evered's 1978 'exemplar action research' approach, together with Checkland's 1991 'dual cycle action research', this research programme will adopt a valid research approach. Original contribution will emerge in each cycle (see Figure 4.7.1), where reflection on both the problem solving and research interest will take place.

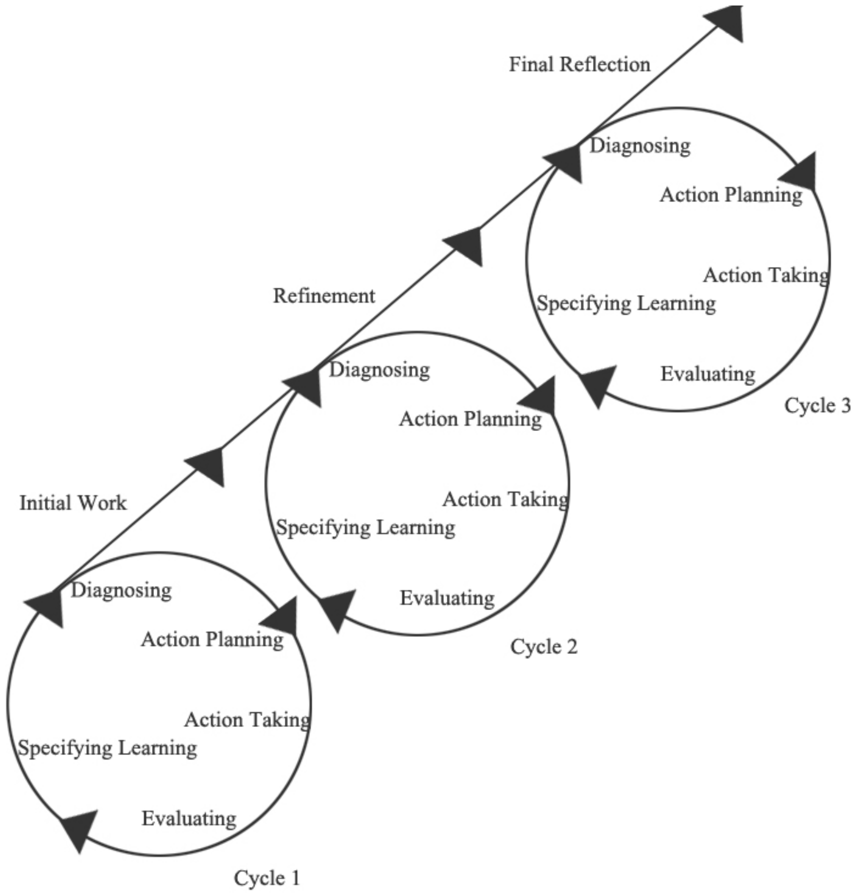


Figure 4.7.1 Proposed Cycles of Action Research.

The researcher will be actively involved in the research, rather than being a passive observer and will collaborate and seek opinions of the student in each research cycle. It is proposed to undertake three cycles of research, each with dual cycles of problem solving and research interests. Each cycle will begin with a diagnostic stage to further refine the problem and define additional research needed to inform the planning stage. Each cycle will also comprise an action taking stage, where the modified intervention will be tested. In the evaluation stage, learning through reflection will allow further problem diagnoses to take place in the subsequent cycles (see 4.7.2 Action Research Model).

Stages		Dual Cycle Action Research	
		Problem Solving Interest	Research Interest
	Diagnosing		
	Action Planning		
	Action Taking		
	Evaluating		
	Learning Through Reflection		

Figure 4.7.2 Adopted Action Research Model.

4.8 Summary

This chapter has described the research paradigm and research method that has been adopted for the investigation. Emphasis is placed on how the investigation is set within an educational establishment, along with suitable methods that can be adopted. Action Research is adopted, as it reflects the need to change teaching practice through dual cycles of problem solving and research interests. The validity of the research was discussed and a model proposed to control multiple cycles of research within the investigation.

Over the next three chapters Canonical Action Research is demonstrated as an effective method in respect of its educational dimension and charting the progress of the study from the perspective of the problem solving and research interests. It also shows how multiple cycles of research can reveal and build upon new knowledge through continuous evaluation and reflection.

Chapter 5 Research Cycle One

5.1 Introduction

This chapter presents experimental work in relation to a web user requirements method. The main aim of conducting the initial work was to explore how students were using requirements documentation in their web development projects and to plan and test an intervention in their practice. Two modules were chosen to test the intervention, including a second year undergraduate degree module called ‘Design for Usability’ (DFU) and a postgraduate module called ‘Integrated Development’ (IID). The chapter reports on the first research cycle, aligned to the canonical action research approach adopted for this investigation, as described in Chapter 4.

5.2 Action Research Cycle One – Initial Work

5.2.1 Diagnosing and Problem Identification

The diagnostic stage of the first research cycle commenced through reflection in professional practice. Having taught on a number of web design and web development modules at a HEI for a number of years, the author identified a problem with the students approach to analysis, design and implementation. Students paid inadequate attention to the analysis stage and were inclined to move straight to the implementation stage. Students often designed and developed for themselves, rather than for the target user group and the majority of students did not document requirements in their written reports. The curriculum of web modules has grown and matured in recent years, but most of this maturity is in the technology rather than the academic theory relating to Web and Requirements Engineering. In particular, analysis techniques have been neglected, a view shared by other authors who teach Web Engineering. For example, Griffiths and Lockyer, 2004, (see Chapter 3, section 3.2).

There is also evidence to suggest that traditional user requirements techniques do not match the needs of dealing with web applications with increasing technological complexity and difficulties in analysing a diverse set of web user requirements, for example, Escalona and Aragón, 2008, (see Chapter 1, section 1.2). Traditional requirements specifications in the waterfall method are created and then frozen. However, web projects tend to be iterative cycles of 'prototype' development and refinement, necessitating a change in the way that requirements are approached. Web user requirements therefore need to be 'fluid', rather than being frozen, in order for continual refinement to take place over the web project lifecycle.

The discipline of Web Engineering has grown its own set of web specific methodologies. The objective is that the project is delivered on time and exhibits a minimum level of quality and conformance. Most notably these include, WebML (Ceri, *et al.*, 2000), December (December, 2008), UWE (Koch, 2006), OOHDM (Schwabe, *et al.*, 1996) and SWM2 (Griffiths, *et al.*, 2003).

In addition, Avison and Fitzgerald 2003 argue that in industry, web based systems are generally produced in an ad-hoc fashion without the use of systematic planning, process, quality assurance or management practices. In addition, Kautz, *et al.*, 2007 argue that "according to similar studies, this means that traditional methods and management techniques are unfit for the development of web-based applications. Therefore, there is a need for new methods and tools for web development and Web Engineering" (Kautz, *et al.*, 2007). A mismatch between current industrial practice and the new practices students should be adopting is hardly surprising, given the curriculum devoid of suitable methods for students to use in RE and in web development methodologies.

Recent research has indicated that there are gaps within Web Engineering relating to web user requirements (Ginige, *et al*., 2001), (Barry, *et al*., 2001), (Escalona and Koch, 2004). In addition to this, there is evidence to suggest that existing web development methodologies tend to concentrate on design and implementation. "There are a significant number of proposals that provide a methodological solution for developing web applications. However, these proposals mainly focus on defining web applications from conceptual models that allow them to systematically obtain implementations. Very few of them rigorously state how to elicit and represent requirements and how to go from the requirements specification to the conceptual model with a sound methodological basis" (Valderas, *et al*., 2007).

Context For Research Cycle One

In order to provide a greater understanding of the setting for the research, this section aims to show how the students, module learning outcomes and environment affected the research findings.

This research programme is set within the School of Computing, Teesside University, a Higher Education Institute within the United Kingdom. The School has a balanced portfolio of programmes on offer, from games development to computer networks. It offers a range of levels to suit the student population, including Higher National Diploma, Undergraduate and taught Postgraduate programmes.

One of the programmes on offer is the BA Creative Multimedia that incorporates a core second year module named Design for Usability (DFU). It has a curriculum that reflects 'web design' such as exploring how to design effective user interfaces, accessibility and usability. It also involves the student in building a website at the end of the module. As such, it was selected for inclusion in this research programme. See appendix A2.1 for DFU module specification.

As previously discussed, the DFU module along with others in the programme, suffers from poor analysis techniques as evidenced in the students' assessment submissions. An opportunity to enhance the curriculum of this module exists in adopting a requirements method as part of new teaching practices on the module. In 2005/6 there were approximately 70 students taking the module. They were made up predominantly by the BA Creative Multimedia students, with other students also able to enrol from generic pathways such as BSc Computing.

A module named Integrated Development (IID) is a core module on the MSc Multimedia Applications, MSc Web Enterprise, MSc Web Services Development, MSc Mobile Computing Applications, MA Web Design and MA Creative Digital Media. Its curriculum is slightly different to that of DFU, with more emphasis on the implementation phase. Its role as a core module on a number of programmes and as an option on others influences the curriculum. As such IID is placed in the programme to allow students learn key web development skills needed to complete assessments on other modules. There were approximately 40 students studying the module in 2005/6. See appendix A2.2 for IID module specification.

Both modules were supported by a lecture each week, with follow up work in a laboratory. Theoretical concepts were first addressed in a lecture and followed up in the laboratory, where students were expected to explore and build on their knowledge and understanding of a given topic. Continuous feedback was an integral aspect of the learning and teaching philosophy, given the nature of the subject and complexities presented to the student.

Research Cycle One Objectives

1) To investigate ways of changing current analysis of web requirements in student projects.

2) To evaluate students' opinions regarding use of a tool to capture and communicate a set of requirements.

3) To demonstrate that a construct for developing user profiles can be used as a starting point within a web user requirements method.

5.2.2 Action Planning

The action planning stage of research cycle one is informed by the investigation and review of related work (see Appendix A3). The mechanism in which the student obtains the data about the user and their requirements is seen as crucial. It was felt that techniques that use primary data, for example, collecting data from a representative target user group, would result in poor adoption of the method. Collecting data of this type would place both time and cost constraints on the student. It was felt that information about the user and requirements could be written into an in-course assessment (ICA) brief. A PBL approach to teaching and adoption of the learning model as described in Chapter 3, section 3.4.3, would allow the student to analyse and make decisions based on this information. A perceived benefit to this would be the development of the students' problem solving skills and to associate design decisions based on evidence rather than 'gut feeling'. A disadvantage could be the potential for the students to make incorrect assumptions based on the evidence provided in the ICA briefing document. The only way to establish this is to propose, implement and evaluate an experimental web user requirements method.

Having undertaken a review of related work and an investigation into the challenges faced within Web Engineering, a number of key principles of Requirements Engineering emerged.

The proposed method should reflect:

1. **Different Types of Requirements.** In AWARE the meta-model describes the 'requirements construct' as; Navigation; Presentation; Content; Access; Structure; Interaction; User Operation; and System Operation. This closely represents the design aspects of a web design. A direct relationship between the meta-model and the website design is seen as a distinct advantage, given the attributes of the inexperienced student user.
2. **Multiple Stakeholders.** The term stakeholder is cited by many authors in RE and WE. Stakeholders reflect a diverse range of user groups, from those who consume content on the website, to those who commission and fund it. A mechanism to distinguish between different user groups includes the term 'priority'. For example, *high*, *medium* and *low* priority. The AWARE approach includes a construct to further describe the user through a 'user profile'.
3. **Include a Suitable Meta-model 'Starting Point'.** The starting point in the requirements meta-model is crucial for the success 'in use'. For example, the AWARE method starts with the 'stakeholder construct'. The user is profiled, tasks and goals associated and a priority level set. Requirements are then derived, having first modelled end user tasks. Alternatively, SOARE starts by defining business objectives and a strategy for attaining those objectives. High level goals are then defined for the user, with subsequent requirements then being defined. The SSM/ICDT starts with the identification of a problem, thereby allowing the developer to understand a 'rich picture' for the subsequent stage of mapping the problems onto a matrix.

4. **Support the Inexperienced Student User.** The inexperienced student user must be provided with some of the background to the problem in order to make informed decisions about the user and their requirements. Use of complex language and a process that is hard to understand may deter the student from adopting a method to express web user requirements. Bolchini *et al*., 2003, identified that AWARE is a ‘lightweight’ approach that uses a meta-model that expresses ‘natural language’ requirements that is intuitive and usable by web analysts. The approach should also require “little training effort for adoption and effective integration into current practices” (Bolchini, *et al*., 2003). The similarities between Bolchini’s work and the inexperienced student user are evident. Students on the web design modules are not used to using methods and additionally adopt a build it now approach, rather than undertaking analysis before commencing the implementation. Design artefacts also feature in AWARE, allowing the inexperienced student user to make connections between requirements and the design. This traceability of requirements is important for measuring the success of the method in the students’ web designs within their assessment submission.

5. **Allow assessment of Traceability of requirements to reflect their validity.** Requirements that are documented in the method should be visible in the submitted assessed web design. In AWARE one of the project objectives was “as a traceability concern, we tried to diminish the gap between requirements and design, trying to iterate during the process” (Perrone and Bolchini, 2005). The gap between requirements and design could be objectively measured, for example, a mark could be given in the students’ feedback, by cross referencing the requirements document and the website design. The SOARE approach also has a concept of traceability, but this focuses on high level business objectives and user goals to low level

requirements traceability. This provides an association between the various constructs within the meta-model.

Problem Solving Objectives

It was important that the method should integrate with existing methodologies used in the teaching of both modules. Having established the need for students to adopt a structured and systematic method to express user requirements, the next step was to outline the problem solving objectives which are:

1. To provide a mechanism to profile the user.

2. To ensure the process is rapid, without the need to collect primary data.

3. It must be structured, repeatable and provide a way of tracing the requirements to the design artefact.

3. The student must be able to identify and understand the language used.

4. It should be accessible to students who, are by definition, less experienced.

5. To express a set of requirements for the web interface design.

6. It should consider the notion of multiple users of websites, rather than one.

In order to reflect the user profiling aspect of the method, a suitable name that the students could identify with was sought. The name ‘Rapid User Modelling Method’ (RUMM) was chosen for this reason.

5.2.3 Action Taking

The main aim of the first cycle is to test an initial experimental method for defining web requirements. It should allow the student to express requirements in a language that they are able to readily identify with, reflecting the fact the most web students have had little formal computer science education. They mainly have a web design background and primarily concentrate on the interface design, usability and accessibility aspects of the website.

Meta-model Construct and Taxonomy

The purpose of RUMM is to help the student elicit, analyse and specify requirements for their web design projects. One of the objectives of the first cycle of research was to '*Investigate ways of changing current analysis of web requirements in student projects.*' It was therefore necessary to determine what student's did first in their web requirements process. Appendix A3 presents a review of existing approaches and offers some guidance on how to achieve this from the perspective of Software and Web Engineering. RUMM also had to integrate into the curricular of web modules and the learning model explicitly sets out how this is to be achieved, with an emphasis on developing the students problem solving skills. It was therefore decided to start the RUMM process by encouraging the student to think about the target user. This is in line with existing requirements approaches such as; AWARE, URN and CE.

In RUMM, the 'User Profile Construct' (see Figure 5.2.3), captures information about the user, thereby enabling the student to build a rich picture of the target user group before requirements are documented. The User Profile Construct includes the following taxonomy:

1) **Characteristics**: *general information about the user* (age, gender, level of computer expertise, employment).

2) **Usage**: *how the user will use the website* (In the course of their job? At home in their own leisure time? Platform? Assistive Technology? Speed? In library or other public access point?).

3) **Goals**: *what the user expects to do with the website* (Use it as the main part of their job? Assist them to do their job? Allow them to buy something online? Assist them to find out information - specific and/or general? Provide fun or leisure activities? Help them learn something?).

4) **Persona**: *a written description about the user.*

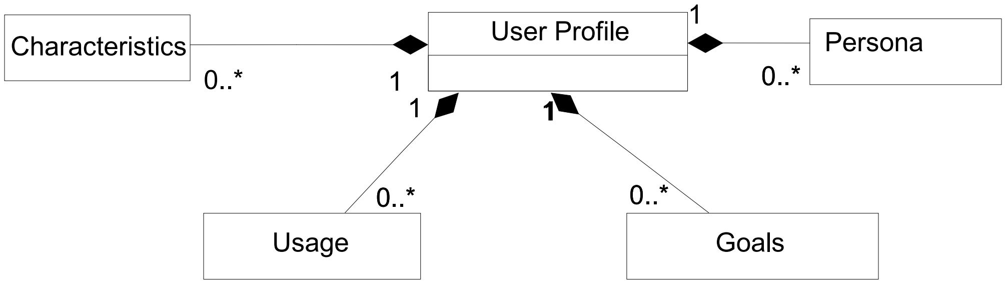


Figure 5.2.3 RUMM – User Profile Construct.

The student is required to identify two types of user for the construct (see Figure 5.2.7):

1) **Primary User**: The main target user who will be using the website. Most of the design decisions will be made based on their profile.

2) **Secondary User**: Consideration is given to secondary users, but their needs have a lower priority.

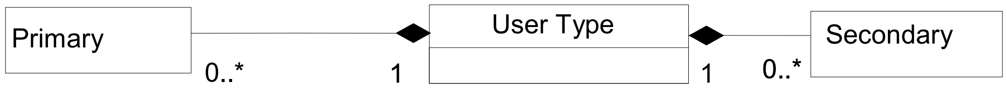


Figure 5.2.4 RUMM – User Type Construct.

The Web Requirements Construct (see Figure 5.2.5) is the next stage in the process. In order to represent the web design requirements, the construct comprises the following taxonomy for the organisation of requirements:

1) **Layout:** *Provide requirements for the layout given the usage and goals of the target user.eg, Liquid Layout, Fixed Layout and Screen Resolution.*

2) **Colour:** *Provide requirements for the visual design. eg, colours to reflect the current colours in corporate image, buttons and logos.*

3) **Content:** *Provide requirements for the content, referring to text, images and audio and video. eg, In the case of a theatre website a requirement might be: 'present details in multiple images and text about each event'.*

4) **Navigation:** *Provide requirements for the navigation, including what content will appear in the 1st level and subsequent levels of the navigation hierarchy. eg, in the case of a theatre website a requirement might be 'Each event to have a link to a 'book tickets' screen'.*

Each construct requires the student to complete the 'fill in forms' (see Figure 5.2.6 and 5.2.7) that represent the overall RUMM meta-model (see Figure 5.2.8.). The student is provided with the forms in both paper and electronic versions during the learning activity as described in the learning model (see Chapter 3, section 3.6). The electronic version allows revision to take place before completion of a final version. The final version may reflect changes in requirements based on prototype development and the discovery of further requirements or the refinement of others.

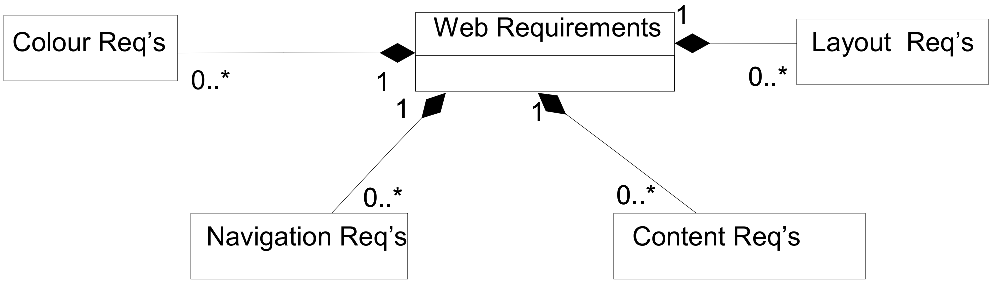
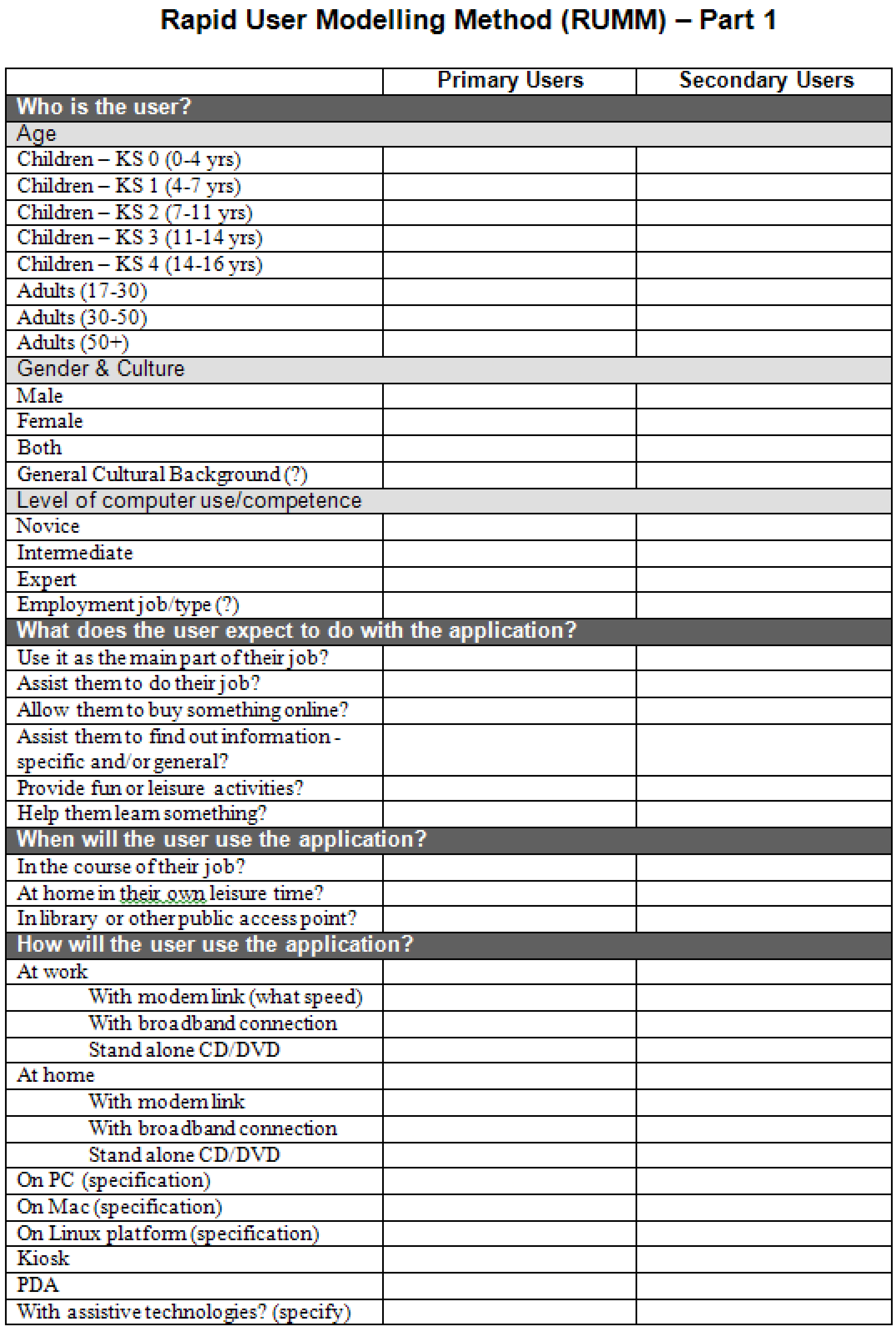


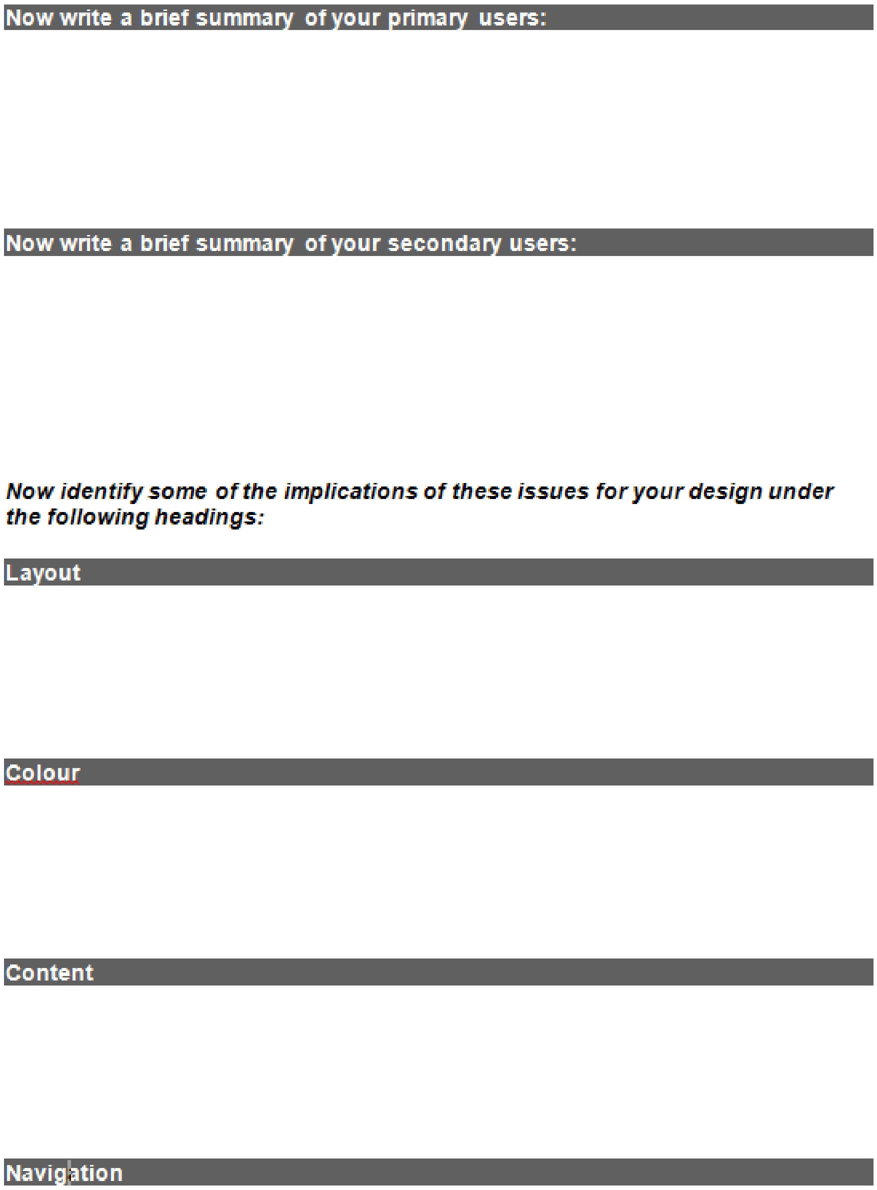
Figure 5.2.5 RUMM – Web Requirements Construct.



Rapid User Modelling Method (RUMM) – Part 1

	Primary Users	Secondary Users
Who is the user?		
Age		
Children – KS 0 (0-4 yrs)		
Children – KS 1 (4-7 yrs)		
Children – KS 2 (7-11 yrs)		
Children – KS 3 (11-14 yrs)		
Children – KS 4 (14-16 yrs)		
Adults (17-30)		
Adults (30-50)		
Adults (50+)		
Gender & Culture		
Male		
Female		
Both		
General Cultural Background (?)		
Level of computer use/competence		
Novice		
Intermediate		
Expert		
Employment job/type (?)		
What does the user expect to do with the application?		
Use it as the main part of their job?		
Assist them to do their job?		
Allow them to buy something online?		
Assist them to find out information - specific and/or general?		
Provide fun or leisure activities?		
Help them learn something?		
When will the user use the application?		
In the course of their job?		
At home in their own leisure time?		
In library or other public access point?		
How will the user use the application?		
At work		
With modem link (what speed)		
With broadband connection		
Stand alone CD/DVD		
At home		
With modem link		
With broadband connection		
Stand alone CD/DVD		
On PC (specification)		
On Mac (specification)		
On Linux platform (specification)		
Kiosk		
PDA		
With assistive technologies? (specify)		

Figure 5.2.6 RUMM User Profile Construct – Fill-In Form.



Now write a brief summary of your primary users:

Now write a brief summary of your secondary users:

Now identify some of the implications of these issues for your design under the following headings:

Layout

Colour

Content

Navigation

Figure 5.2.7 RUMM Web Requirements – Fill-In Form.

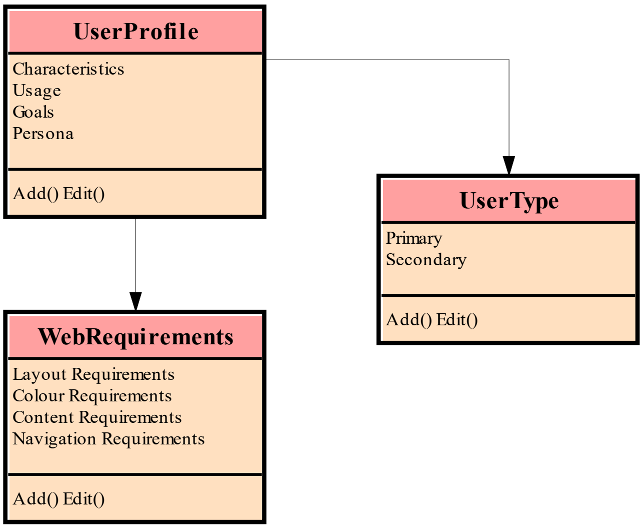


Figure 5.2.8 RUMM Meta-model.

Application and Testing

RUMM was used in the teaching of a second year module named ‘Design for Usability’ (DFU) and a Postgraduate module named ‘Integrated Development’ (IID). These modules used the ‘User Centred Web Development Methodology’ (UCWDM) to support the development process in order to align the curriculums as closely as possible. Use of UCWDM mirrored aspects of RUMM, where the development process begins with the ‘identification of the target audience’. RUMM therefore integrated with the analysis stage of the development methodology which the students had to undertake.

Both DFU and IID had their curriculum divided into two parts, the first being the analysis and design of a website. The second part concerned the website implementation such as XHTML and CSS that transforms the design into a website. It was deemed advantageous to use RUMM only on the design aspects of the module since its meta-model was limited to the design requirements. Adopting the learning model for web user requirements (see Chapter 3, section 3.6) an initial lecture provided the students with some theoretical underpinning regarding RE and WE, including an explanation of the research and how the students themselves would become participants in that research.

The students on the module were provided with the ICA brief that described in detail the ‘problem’ and defined the steps and tools required in order to satisfy the learning objectives. Students then worked on the problem throughout the duration of the first part of the module, with the author acting as a facilitator. The ICA contained detailed information from which the student could undertake further analysis. For example, after the student had identified the user and created the profile, they started to ask more questions about how this would affect the layout and navigation. A number of students were dismissive of the requirements process, preferring instead to move straight to the implementation phase. The facilitator encouraged them to re-visit RUMM once they had an initial prototype finished in order to check that they had satisfied the ‘problem’ identified in the ICA brief.

As part of the module assessment submission and reflecting the learning model, the student had to submit their design documentation, requirements analysis and design artefact, part of the way through the module. Formative feedback was provided both from peers and the tutor, from which the student was expected to make changes in the final version of both artefact and documentation.

5.2.4 Evaluation and Learning

The purpose of this stage within the action research cycle was to assist in determining if the first cycle of research fulfilled its problem solving objectives in relation to the method set out in section 5.2.2. It also provides the researcher with data to interpret for reflection purposes so that changes can be made to the method in the next research cycle.

Key Findings

This section presents the findings from the student survey and feedback from SIGSAND conference. It evaluates both the action taking and research activities, using the objectives developed in the initial stages of this chapter as a basis for the evaluation.

Student Opinion Survey

At the end of the module, students were asked to complete a paper based questionnaire regarding their opinion of RUMM (no pilot questionnaire was used in this instance due to time constraints). The aim of the survey was to establish if students felt that the process was useful to them, how it compared to other methods they may have used, if they felt that there was anything missing and any improvements that could be made. Additionally, students were asked if they would use a more advanced method based on RUMM. It was hoped that through the analysis of the results, improvements could be made to the process meta-model. (Please refer to appendix A1 for an example of the RUMM Survey Questionnaire).

Completed questionnaires were returned anonymously to ensure an un-biased response. Seventy six students were asked to complete the questionnaire on a second year degree and on a postgraduate module. Fifteen questionnaires were returned by the undergraduates and six questionnaires were returned by the postgraduates, giving a total of twenty one responses overall. The response rate of (n=21) 16% is considered to be normal for a questionnaire survey, but disappointing in terms of what had been expected. The response rate does raise some issues regarding the limitations of the survey and analysis of the survey data must be made with this in mind. Methods of enhancing response rates would be looked at in the next research cycle.

Analysis of the data was undertaken by means of coding the responses to each question in a codebook (see appendix A2.3) and then further analysis of the data in a spreadsheet. Responses were converted to percentages in the spreadsheet in order to create relevant charts. Table 5.3.1 below provides a summary of the data of the responses, from which the charts were derived.

Questions	Response Data (n=21)				
	a	**b**	**c**	**d**	**e**
1. How useful did you find using this approach was in helping you define your audience? *a. Very Useful/helpful b. Useful/helpful c. Neither helpful nor unhelpful d. A little unhelpful/confusing e. Complicated & very unhelpful.*	9 43%	11 52%	1 5%	0 0%	0 0%
2. If you have used other user defining approaches in the past how do you think this approach compares? *a. Never used other approaches b. Much more useful c. More useful d. Not much better really e. I'll stick with my original approach!*	4 19%	5 24%	8 38%	4 19%	0 0%
3. Do you think there are any points missing from the list that should be considered? *a. Yes b. No*	9 43%	12 57%	n/a	n/a	n/a
4. Do you think there is any unnecessary information being gathered through this method? *a. Yes b. No*	2 10%	19 90%	n/a	n/a	n/a
5. The next stage of the model will be to link the user model you have created with issues that you need to consider in your design. If such a tool were available to you how useful and helpful do you think it would be to *you* as a designer? *a. Very useful & helpful b. Useful and helpful c. Neither helpful nor unhelpful d. Unhelpful e. Very unhelpful*	13 62%	7 33%	0 0%	0 0%	1 5%

Table 5.3.1 Summary of data collected from the student survey.

Discussion of Results

Question 1 asked how useful the approach was in helping to define an audience. Twenty students (95%) indicated that they found RUMM 'very useful or useful' in defining the audience for the projects that they were working on. As RUMM was an experimental method, with user profiling as a mechanism to discover requirements by defining the target audience in detail, this aspect deemed to be successful.

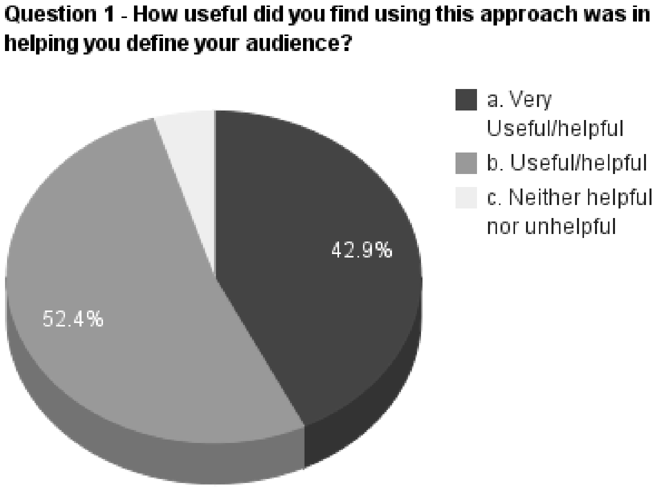


Figure 5.3.2 Results indicating how useful RUMM was in target audience definition.

Question 2 asked if they had used other approaches for defining the user. 4 students (19%) said that they had never used other approaches and of those who had used an alternative method for defining the audience, (62%) said that it was more useful compared to the approaches used previously. Four students (19%) expressed the opinion that it was not much better than an approach they had used in the past.

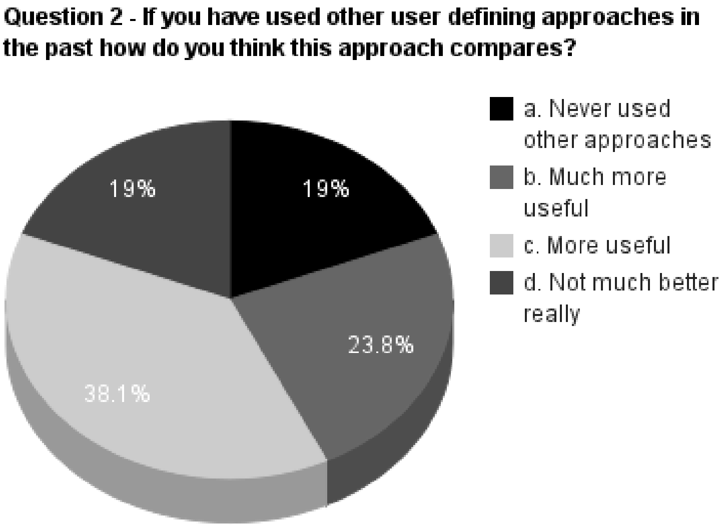


Figure 5.3.3 Results indicating if the student had used other approaches for defining users.

Question 3 asked if there were any points missing from RUMM that should be considered. The responses were evenly distributed, indicating that the student felt that something was missing in the process.

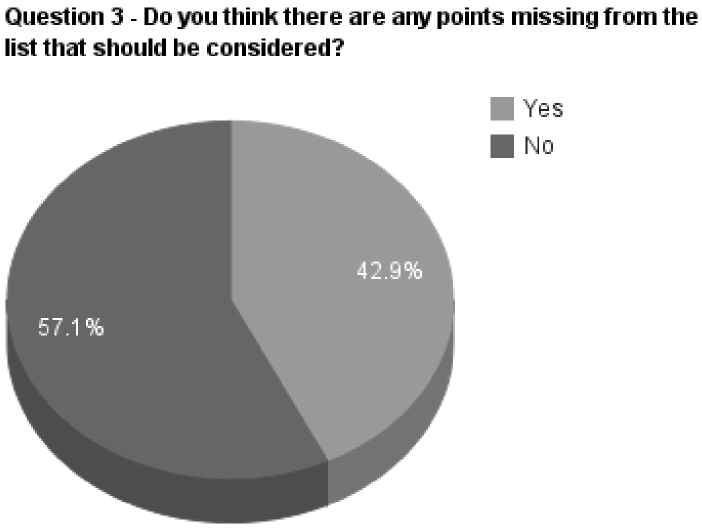


Figure 5.3.4 Results indicating if there were any points missing from RUMM.

The open responses were quite varied and include:

- “More Focus on personal background of users, including profession, hobbies and customs to help understand them better” (anonymous student).
- “Non-native English speakers (specify level), users reading from right to left or from bottom to top, disabled users with/without assistive technologies, use at school/college/university” (anonymous student).
- “Level of computer use/competence section, we need to know what is actually meant by novice, intermediate, expert” (anonymous student).

This feedback is again quite interesting in that the students wanted to include more detail in the user profile. Some students felt that more characteristics should be integrated into RUMM, such as hobbies and a profession. It could be argued that these are perhaps too detailed and specific but are none the less, important in thinking about the user.

Question 4 asked if they felt there were any unnecessary information being gathered through the method. Only one student answered this question responding that they thought there should be a clearer way of differentiating primary users from secondary users. It was felt that this is a very important aspect of RUMM that could have implications for the way in which a priority system could be integrated to ensure no conflicts arise.

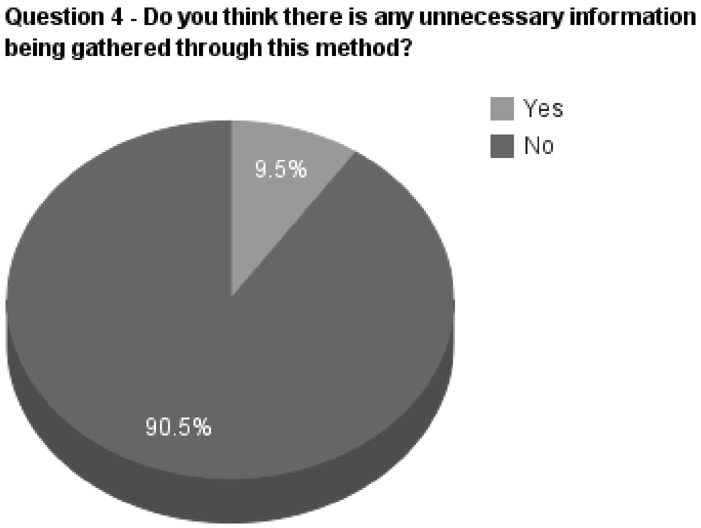


Figure 5.3.5 Results indicating if the students felt there was any unnecessary information being gathered.

Question 5 asked if RUMM provided further support via a tool for students in completing the method, would they find this useful? Thirteen student's (62%) said that they would find this very useful and seven students (33%) said that they would find it useful. The results of this question indicates additional support is required in some way, perhaps moving away from the paper based method to a more electronic representation of the process.

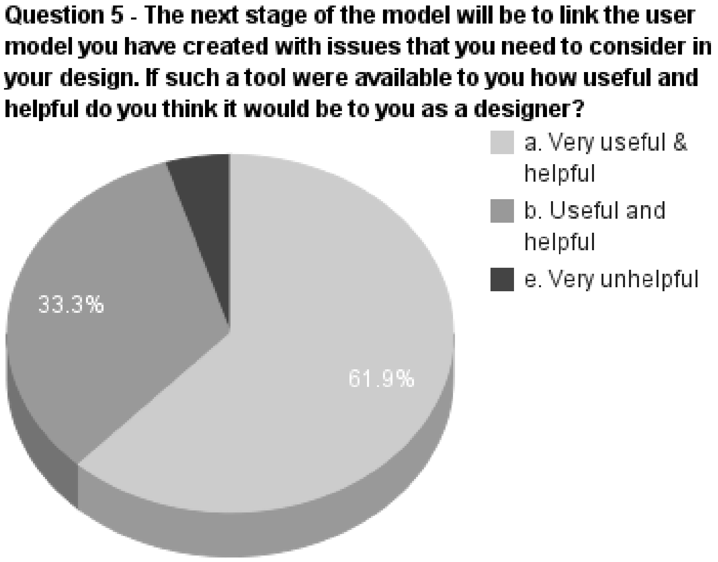


Figure 5.3.6 Results indicating if the students felt additional support would be useful.

The results from the survey are very encouraging in terms of acceptance of using a method for expressing web user requirements. Students have also indicated gaps in the method, where additional work is needed. Further support and lack of user differentiation (too fixed on primary and secondary users) were apparent. Although the method is still at an experimental stage of development, students felt that it did help them to define who the user is, but felt that the main weakness of the tool was within the requirements construct. This aspect needs more work in the second cycle to enable the student to discover and define much more precise web specific requirements relating to the design and development aspects. Respondents indicated that the existing method for describing design requirements were too loosely defined. Other students also indicated that they felt the method was too restrictive and wanted to modify the process to suit their own needs. Having an more open mechanism to define requirements is needed in the second version of the method.

Feedback from the SIGSAND conference

Preliminary findings from the first research cycle were presented as a conference paper at the European Symposium on Systems Analysis & Design: Practice and Education (AIS SIGSAND) on the 6th of June 2006.

Feedback was provided orally from the delegates regarding the initial work, its key findings and areas to look at in future work. The following feedback points were noted in written form at the end of the session:

- Delegates noted that this could be “yet another method”.
- The safety of allowing the students to make assumptions or guess work was called into question.
- Focused too much on user modelling at the expense of modelling features of the web design.
- Liked the idea of RUMM being referred to throughout the development lifecycle, but what happens if something changes? Should be flexible to changes in requirements.
- Could students upload their RUMM documents to a common shared area, and re-use profiles?
- Need to differentiate design / application type development in the second part of RUMM.
- Could look at ways of improving the students’ assumptions by providing some lightweight documentation such data sets for them to analyse.

- The term users may be confusing for the student. Suggested terms included ‘Audience’ and ‘Actor’.
- Investigate the following: SSM (Meldrum and Rose 99), Task Oriented Requirements Engineering (TORE) and Sysiphus.

5.2.5 Learning Through Reflection

Reflections on the Problem Solving Interest

Observation of RUMM in use took place within a laboratory setting whilst the students worked on their In Course Assessment (ICA). Students used the method within a laboratory setting, where the researcher was active in terms of providing guidance to students in its completion. During this time notes were taken in order to identify issues, behaviours of the student towards the method and as a way of documenting what happened. Excerpts from the Observational Logs can be found in Appendix A2.4.

Observations were recorded in an observation log, mainly concerning the way students were interacting with RUMM. Further observations were undertaken in the assessment process, mainly pertaining to any changes in behaviour relating to RE and to enable evaluation of the learning model.

Review of Problem Solving Objectives

Seven objectives were established for the method. (*Note: these objectives are part of the problem solving cycle. Research objectives are reviewed independently of these. (See Dual Cycle Canonical Action Research, Chapter 4 section 4.7.1*). These are now reviewed as part of the evaluation and learning stage of the problem solving interest.

Objective 1. *Provide a mechanism to profile the user.*

A user profile construct provided the student with a rich picture of those persons who would interact with the web application. In this regard, the profile construct was successful, but required more work in terms of associating the user to specific goals or tasks and the website requirements represented in the design and development. There were also difficulties observed in using the profile construct as the first stage process and an alternative starting point may have provided advantages in envisioning requirements.

Objective 2. *The process must be rapid, without the need to collect primary data.*

An oversimplification of the meta-model resulted in the students themselves extending the method to 'make it fit' their projects. Initial work pointed to potential problems in the collection of primary data to model the user and their requirements. In an educational setting, this was deemed too costly in terms of time and the ICA brief proved to be effective in providing the student with a source of data in this regard.

Objective 3. *It must be structured, documented, repeatable and provide a way of tracing the requirements to the design.*

Work on the meta-model ensured that a structured process could be followed by the student. On reflection the model lacked an effective association model, for example, there was little linkage between the user and their goals. In the assessment process requirements that were documented by the student could be identified easily in the design and this aspect was deemed to be successful.

Objective 4. *The student must be able to identify and understand the language used.*

Attention was focused on using language that the student would expect to see, given their background in web design. Web Requirements were expressed as: *navigation*, *content*, *colour* and *layout*. Although this provided familiarity to the student, some indicated that they would like to extend the construct to include dynamic development requirements.

Objective 5. *Should be accessible to students who are by definition less experienced.*

Fill-in-forms were used to represent the meta-model. Electronic and paper based forms were provided to the student along with an orientation session regarding their use. Illustrative examples were provided in response to the students 'first use'. Further work was required in order to better support the inexperienced student user. For example, a support and guidance model could be integrated into the method to aid completion.

Objective 6. *To express a set of requirements for the web interface design.*

Some students identified that an emphasis on web design (layout, content, colour and navigation) was a weakness, particularly on the Masters module. It was established that these students had wanted to use the method on their final projects. It could be argued that this indicated a gap in the availability of a suitable web requirements method.

Objective 7. *It should consider the notion of multiple users of websites, rather than one.*

This was perhaps the greatest weakness identified in the evaluation. The user profile construct had a limited classification system that composed of a primary and secondary user. Additional users could not be added without a modification of the meta-model. Relationships and dependencies were also vague between these two types of users. For example, it was not clear how conflicts could be resolved if two competing requirements arose in the web requirements construct.

Overall Reflection on the Problem Solving Interest

Within initial laboratory session, the majority of students were enthusiastic about RUMM, but wanted know more from the tutor directly, rather than exploring the process themselves. This was contrary to the learning model adopted to teach RE, as students were expected to undertake explorative work in order to discover how things worked, facilitating deeper learning as argued in Chapter 3. Although an underpinning theory lecture was provided on RE and RUMM within the learning model, the students indicated that they wanted a demo or walk through in order to more fully appreciate how they would go about using it in their project. An illustrative example was provided via the VLE, based on a project that the researcher had completed recently, but had to be 'reverse engineered' as the project was completed before RUMM. This raised some interesting questions about RUMM as an effective teaching aid and the learning model. In particular, can RUMM be extended to incorporate dynamic development? Are there additional process models required in order to facilitate the inexperienced student user? Is PBL the correct approach given the students previous experience?

It was also observed that the use of RUMM in the laboratory was sometimes sporadic, with some students completing it in one go and others completing part of it before moving onto another aspect of their project. Although the pathway through the process was thought to be simple, there was a general feeling on the part of the students that it was too simplistic. The web requirements construct in particular was treated by the students in different ways, with some only writing a limited amount of requirements and others extending the taxonomy. In the assessment process, it was also evident that some students had completed RUMM entirely, but others had only part completed it. This raised the issue of completeness and an additional question that is born out of the first cycle of research. How can we ensure that the student had completed RUMM and arrived at a valid and complete set of requirements?

User profiling seemed to strike a accord with the majority of students and it was noted in the assessment process that some had gone further with the 'persona' aspect and provided additional detail such as a photograph and character name. There was some uncertainty as to how the user profile informed and shaped the requirements in the 'web requirements' construct. An aspect of the meta-model that could be improved is the association between users, goals and requirements, resulting in the following question; is there an association model that could be used to achieve this in the next iteration?

The learning model adopted for the intervention incorporated a hybrid PBL approach and an iterative 'experiential learning paradigm'. The learning model encouraged the student to explore the process model by solving problems facilitated by the tutor, with little intervention expected. In reality, more interventions took place than was expected, mainly related to issues concerning the 'primary' and 'secondary' user relationship. Some students could not grasp the difference, but perhaps more significantly, others wanted to define more than two users. For example, one student wanted a classification system that comprised a 'consumer', 'casual browser' and 'administrator'. A conflict of interest between wanting to help the student and adherence to the learning model ensued.

Perhaps the most pertinent observation came to light on the postgraduate module. Some students wanted to use RUMM on their Masters project and asked for modifications in order to reflect 'dynamic web development' as opposed to the 'web design'. A more open taxonomy for the capture of functional requirements in particular became evident and which was impossible in the current meta-model. A number of questions resulted from this, for example, how would one extend the web requirements construct to reflect dynamic development? Is there a way of allowing for more flexibility in the construct?

Some students questioned the starting point of the method, with RUMM using the ‘user profile’ construct as the first stage of the process. A number of students had an initial discussion with the tutor about the overall vision for the project and what the client wanted the website to do. Without realising it, the students had found a problem and a potential solution to this issue. The vision and objectives of the website have a direct impact on the requirements, but this is something that is missing from the meta-model at present. This aspect requires further investigation in the next research cycle.

Reflections on the Research Interest

Chapter 4 indicated that the research approach adopted for this investigation is ‘Canonical Action Research’ which comprises dual cycles that involves one or more iterations of diagnosing, action planning, action taking, evaluating and specifying what has been learned through reflection, for interventions in the next cycle. Having already undertaken a reflection of the problem solving interest, attention is now paid the research approach undertaken.

The collaborative nature of action research, with the students actively becoming involved in the project, was found to be highly successful. Students were informed of the project aim and research methodology before starting work on their projects. The students provided the researcher with vital feedback in real time and this feedback could be recorded for later analysis. A perceived weakness of action research is the involvement of the researcher in the experimental stage. For example, was the researcher influencing and intervening too much, thereby in-validating the learning model? Were the students also providing feedback in a skewed way, in order to get a better mark?

There was also a feeling through the first cycle of researcher of the students being ‘subjects’ and ‘being experimented on’. The direct intervention in the learning, not only in the method, but also the learning model, led to some questioning of the ethics of the research.

In terms of the validity of the research, it is noted that the research:

- Has followed a documented canonical action research approach that is defined in the literature.
- Use of Action Research in an Educational Environment is aligned with research methods in the interpretative paradigm.
- Is underpinned by a clear identification of a gap related to web user requirements in a teaching context.
- Has led to planning and action taking concerning an intervention underpinned by a review and analysis of RE and WE.
- Actively encourage a partnership between the researcher and student
- Embraced the spirit of ‘research informed teaching’ through a reciprocal relationship between teaching and research, each benefiting from activities undertaken.

Reflection of Research Cycle 1

Three **research objectives** were established within the first cycle of research. These are now reviewed in the context of the research interest within the action research approach.

1) To investigate ways of changing current analysis of web requirements in student projects.

A review of similar work had informed the action planning stage of the action research approach. Although this review had not found investigations directly relating to educational web requirements methods, the findings were valuable in providing a starting point for the research.

2) *To evaluate students' opinions regarding use of a tool to capture and communicate a set of requirements.*

Students were asked to complete a questionnaire concerning their opinions of using RUMM on the module. In addition, the researcher observed the students in a laboratory setting, which provide a rich insight into how they used it, along with real time feedback. This provided insight into the investigation that would not have been possible using a scientific research paradigm.

3) *To demonstrate that a construct for developing user profiles can be used as a starting point within a requirements method.*

The review of similar work had found a number of approaches, each starting the process at different points. Bolchini *et al.,* 2004, had found that envisioning the user helps to provide the web analyst with an overall vision for the web project before requirements are elicited. Following Bolchini's work, RUMM started with the user profile before moving onto defining the web requirements. It was found that students had difficulty with this, in particular they could not then relate the profiles to goals or requirements as the meta-model was lacking a way of expressing these relationships.

Reflection on the first cycle as a researcher

Reflections on the first cycle are centred on the design of research, mainly in the questionnaire design and deployment. It was disappointing to hand out paper based questionnaires, only to receive very few back. The response rate was in line with normal expectations for a questionnaire of this type, but it was felt that more could be done to enhance the response rate. It was also difficult to keep track of responses and their relationship to modules, as this had to be manually completed by the student.

Lessons learned for the next stage would include creating an explicit relationship with the questionnaire and its completion by the student, perhaps using the assessment process. Problems associated with tracking could be solved by moving to an online questionnaire.

5.3 Conclusions

The initial diagnostic stage within the action research method identified specific problems in teaching Requirements Engineering within Web Engineering at a HEI. A number of research objectives were defined as part of the method of research that could be later evaluated. An initial experimental method was produced that comprised a meta-model that contained various constructs to aid the inexperienced student user in defining their requirements elicitation, analysis and specification. User profiling was used as a way of encouraging the student to think and specify web requirements.

An action taking stage within the research cycle provided an opportunity to test the intervention with students taking web design modules, where the learning model was used to deliver the necessary theory relating to requirements and a PBL process. Observation of the method and a student survey was undertaken in order to gather evidence for its effectiveness.

Evaluation of the problem solving interest provided the basis for further refinement of the meta-model to better represent the inexperienced student user in defining web user requirements. In particular, the first meta-model was found to be too restrictive to be used for dynamic web development due to the closed ‘design specific’ requirements construct. Other issues were also found during the evaluation of the problem solving interest, for example, consistency, completeness and correctness of requirements; the starting point in the process; too much reliance on ‘user profiling’ as a mechanism to further define requirements; additional guidance and support mechanisms required in order to better reflect the adopted learning model and no mechanism to associate requirements to specific users. The evaluation of the research interest included the first cycle objectives for the programme of research.

The following points will form the basis of an updated set of research objectives for the second research cycle:

1. Examine ways of extending the meta-model to include a more open taxonomy for requirements capture, including dynamic web development.
2. Review the user profile construct within the meta-model, especially concerning the need to express more user profiles.
1. Examine how to extend the meta-model to provide a richer understanding of business requirements and overall vision for the website.
2. Explore ways of incorporating a support and guidance model to help the student user complete the process.
3. Investigate how the student can be encouraged to complete the whole process, rather than selectively completing and submitting parts of the process for assessment.

These points will be used as the basis for the diagnosis of the second research cycle in Chapter 6.

Chapter 6 Research Cycle Two

6.1 Introduction

The first research cycle provided the researcher with a valuable understanding of how the student approached RE within their web design projects. A review of similar work in Appendix A3 provided evidence for a range of interventions to challenge students perception of RE and how they can adopt a method to specify requirements in their web projects. Evaluation of the first cycle utilised qualitative research methods and reflection on the part of the researcher in relation to the learning aspect of the intervention. A conference paper based upon this initial work was accepted and presented at the 'Special Interest Group on Systems Analysis & Design' (SIGSAND) conference, held at Galway University in 2006 (see Appendix A3.1).

The second research cycle commenced by reviewing the feedback from the qualitative research, research reflection on the learning aspects and from the feedback received from delegates at the SIGSAND conference. In particular, the second cycle shifted emphasis away from *web design requirements* to *web development requirements*, reflecting the need to address dynamic web development. The starting point of the RE process was also modified, addressing the comments from the student questionnaire survey. A further review of similar work in the area of RE and WE provided the basis for the development of a modified method. The method was tested on a more focused cohort of students undertaking 'dynamic development' in order to test the new meta-model. The evaluation of the second cycle included qualitative research methods and by reviewing traceability of requirements as evidenced in the students assessments. The work was then presented at the United Kingdom Academy for Information Systems (UKAIS) Phd Consortium in 2007 and feedback on the second cycle was offered by delegates. Feedback from both conferences provided the basis for the third cycle of research that is presented in Chapter 7.

6.2 Action Research Cycle Two

6.2.1 Diagnosing and Problem Identification

From undertaking the first cycle of research it was found that the intervention had a positive impact on the students' attitudes to RE, but there were a number of substantial issues that need further attention in the second cycle. The main strength of intervention became apparent in the students' behaviour in the laboratory, where their practice had changed and more attention had been paid to the user and their requirements, due to the fact that much more analysis was taking place. In the majority of ICA submissions, evidence that the students had followed a systematic approach could be found in their written reports with traceability in their web design. The main weaknesses of the intervention included, its closed taxonomy, which could not be extended by the student, focus on design requirements at the expenses of dynamic development requirements, a confusing entry point into the process model and the lack of a mechanism to associate users to specific requirements.

Context For Research Cycle Two

In order to provide a greater understanding of the setting for research cycle two, this section aims to show how the students, module learning outcomes and environment affected the research findings.

In the second research cycle a module named Online Business Systems was chosen due to its focus on web development, as opposed to web design. This module is used as an option or core on a number of programmes within the school, such as BSc International Business Information Technology (core) and BSc Computing (option). Students enrolled on the module therefore have a wide range of subject knowledge and require greater learning support.

Problems identified in the teaching of Online Business Systems includes poor documentation in terms of what the student has built, stemming from poor analysis techniques. As with the first research cycle, an opportunity to enhance the curriculum of this module exists by modifying the students' practice in the way they discover, analyse and specify their requirements. In 2008/9 there were approximately 40 students taking the module. See Appendix 2.5 for OBS module specification.

Integrated Development (IID) is again adopted for the second cycle. It is still a core module on the MSc Multimedia Applications, MSc Web Enterprise, MSc Web Services Development, MSc Mobile Computing Applications, MA Web Design and MA Creative Digital Media. In 2008/9, there were approximately 40 students studying the module. See appendix A2.2 for IID module specification.

The learning and reflection stage of the first action research cycle documented in Chapter 5, 5.3 section highlighted areas that needed to be explored further in order to enhance the method. These are now discussed in more detail, along with a justification for changes that were made.

1. Examine ways of extending the meta-model to include a more open taxonomy for requirements capture, including dynamic web development.

Evaluation of the first cycle of research identified that the taxonomy of web requirements (*navigation, colour, content and layout*) concentrated too much on design aspects and could not be extended by the student. An alternative taxonomy needed to be established to reflect both design and dynamic development requirements, whilst recognising that some students would want to define their own 'web requirements'.

Returning to the literature on RE provided the basis for a modification to the requirements taxonomy. Sommerville 1997 describes the 'requirements specification' within the RE process, that are further refined into two main categories:

1. Functional Requirements – *Describe tasks, interactions, behaviours and features of the system to be developed.*

2. Non-Functional Requirements – *Impose constraints on design and implementation.*

A similar taxonomy to the one outlined by Sommerville 1997 can be found within Appendix A3; Usability Context Analysis (UCA) (see Appendix A3, section 3.4.2), Object View And Interaction Design (OVID) (see section 3.5.1), Task Based Audience Segmentation (TBS) (see Appendix A3, section 3.5.3) and Navigational Development Techniques (NDT) (see Appendix A3, section 3.5.6). The main advantage in modifying the taxonomy to include a 'functional requirements construct' would be the ability to add requirements that relate to dynamic development, based upon an actor *task* or *interaction*.

Non-functional requirements could be fixed in terms of taxonomy and would relate to web design, web environment and usability requirements. This would enhance the web aspects in both design and development. It would also prompt the student to consider these aspects in their design and development. An opportunity to extend the taxonomy also exists for non-functional requirements in order to better reflect design features and constraints and the dynamic nature of the web. MSF (see Appendix A3, section 3.5.4) provides a taxonomy that divides requirements into operational and system requirements. By sub dividing non-functional requirements into separate constructs it is hoped that greater flexibility will be achieved by allowing the student to select non-functional requirements that matches their project.

As argued in Chapter 2, the differences between 'Software Engineering' and 'Web Engineering' are distinct. These differences present a set of challenges that concern the platforms on which websites are deployed. Issues of *usability* and *accessibility* present the web developer with a multitude of problems to solve and choices to make before development commences. The student must also decide upon a suitable server-side language in order to facilitate database connectivity, file processing and authentication management. It is essential that these constraints are reflected in the new taxonomy for non-functional requirements. It is proposed to include four constructs to represent non-functional requirements within the extended taxonomy:

- Application Development Environment Requirements
- User Interface Requirements
- Security Requirements
- Usability Requirements

An opportunity to enhance the traceability of requirements also presents itself in adopting this approach. To achieve this each Functional Requirement should have a unique identifier, along with a mechanism to associate each requirement with a specific actor. Non-functional requirements would not need an association model since these describe the constraints of the website and are therefore implicit in the design.

2. Review the user profile construct, especially concerning the need to express more user profiles.

The evaluation and reflection of RUMM indicated that a deficiency existed in the user profile construct. A restriction on the number of users that could be created caused problems for the student. An alternative approach was therefore required and it was proposed to dispense with the notion of a *primary* and *secondary* user.

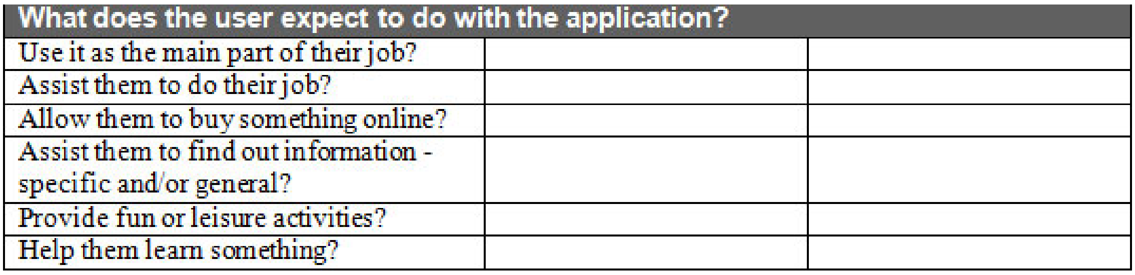
The term 'actor' is widely used in other requirements methods to represent the user. For example, NDT uses the term 'webActor', which can be both human and system. An actor could be represented as an external system such as a web service or application programming interface (API), which interacts with the website. As such, it was thought advantageous to change the term 'user' to 'actor' within the construct.

The issue of having a fixed number of actors within the meta-model was more of a challenge to resolve. An alternative way to express the importance of the actor within the web application is the idea of 'priority'. Bolchini, *et al*., 2003, expresses actors as 'stakeholders' and attributes a priority status in order to differentiate which actors are more important than others. "A priority may thus be associated to each stakeholder in order to help analysts weigh properly the goals and the needs expressed by each stakeholder and consequently plan effort and resources for the analysis in a more efficient way" (Bolchini, *et al*., 2003).

This approach allows the production of unlimited actors, rather than having a fixed primary and secondary user. By using an 'actor priority', it is possible to have a number of actors, each with their own profile and varying levels of importance within the web application. It was therefore proposed to keep the profiling aspect, but to split the existing user construct into:

1. An Actor Construct
2. An Actor Profile Construct

Within the first meta-model, the user profile construct taxonomy also included a question concerning what the user would do within the application (see Figure 6.2.1). Again, this proved to be too inflexible for the student. In order to better represent how the actor would interact with the web application, it was proposed to remove this aspect of the actor profile and replace it within something that was more flexible.



What does the user expect to do with the application?		
Use it as the main part of their job?		
Assist them to do their job?		
Allow them to buy something online?		
Assist them to find out information - specific and/or general?		
Provide fun or leisure activities?		
Help them learn something?		

Figure 6.2.1 RUMM User Profile Construct Taxonomy.

A number of options existed for replacing this aspect of the construct. For example, Use Cases provide a conceptual model that includes modelling the actor *behaviour*, by defining *goals* that they will perform. “The Actors and Goals conceptual model is handy, since it applies equally to businesses and computer systems. The actors can be individual people, organisations or computers” (Cockburn, 2001). Other web requirements methods such as the ‘NDT Model-Driven Approach’ (Escalona and Aragón, 2008) and ‘A transformational approach to produce web application prototypes from a web requirements model’ (Valderas, *et al*., 2007) use the concept of Tasks. In the latter, the work is of particular importance as it offers a way forward to discover and specify functional requirements, based on the tasks that are modelled. Valderas, *et al*., 2007, describes how a *task metaphor* is “widely accepted for the capture of functional requirements” (Valderas, *et al*., 2007).

Furthermore, their work extends this concept by explicitly associating the task to a specific user.

The benefits of using this approach are twofold. Firstly, it allows the student to model individual tasks that can then be used to generate functional requirements. Secondly, an association can be generated between the task and the actor in order to enhance requirements traceability. It was therefore proposed to offer an additional construct as part of the meta-model, which helped the student to model the tasks that the web actor carried out within the web application.

3. Examine how to extend the meta-model to provide a richer understanding of business requirements and overall vision for the website.

Students indicated that they felt the starting point of the method, using the user profile as the entry point, was somewhat confusing. On reflection, it was identified that although the student had to undertake analysis of the problem, the evidence for this was not documented within the method. Providing a way of evidencing this in the method could resolve this issue and at the same time combine with the PBL model. In particular the problem analysis stage of the learning model would be documented more fully and enable more precise feedback to be provided to the student.

A 'statement of purpose document', (variously referred to as a 'vision document', 'business case' and 'mission statement'), was used in both SSM/IDT (see Appendix A3, section 3.5.11) and in CRC (see Appendix A3, section 3.5.8). In addition Al-Salem and Samaha, 2007 argue that the characteristics of web applications are "directly stemmed from and influenced by strategic business vision and goals" (Al-Salem and Samaha, 2007).

Two additional constructs are proposed that will help the student in their problem solving process as well as providing documentary evidence.

1. Statement of Purpose / vision - provide a consensus on what the site will do and will not do; motivations for its creation or modification to the website success criteria and how these will be measured.

2. Web Objectives - describe specific activities within the website that will achieve the overall vision for the website.

4. Investigate the possibility of incorporating a guidance model to help the student user complete the processes.

The reflection and learning stage identified a weakness in the existing method in cases where some students wanted the tutor to provided illustrated examples. This presented the researcher with an opportunity to extend the meta-model to include a guidance model or help system. This took the form of supplementary notes that exist alongside the ‘fill in forms’.

5. Investigate how the student can be encouraged to complete the whole process rather than selectively completing and submitting parts of the process for assessment.

One way to achieve this was to change the assessment criteria so that partially completed requirements analysis would receive lower marks. If the process was electronic, the requirements process could be controlled more precisely and could be restricted so that all parts of the meta-model had to be completed. A requirements document could then be printed out by the student.

Research Cycle Two Objectives:

The updated research objectives for the second cycle of research were based upon reflection and learning from the first cycle of research. The objectives are:

1) To investigate ways of extending the meta-model to better support the inexperienced student user to define dynamic web requirements.

2) To establish how relationships between requirements and actors can be modelled.

3) To evaluate students' opinions regarding the updated meta-model.

4) To demonstrate that an updated meta-model can be used in the teaching of web development.

6.2.2 Action Planning

This section describes the modifications and restructuring of the intervention in the second cycle of research. The restructured meta-model was translated into a new teaching tool that utilises fill in forms in both paper and electronic formats (see Appendix A2).

Problem Solving Objectives

Having identified the major changes needed to the framework, the next step was to outline the objectives of the modified framework, which include:

1. Enable students to envision the web project before undertaking analysis of functional and non-functional requirements.

2. The framework should enhance student support by integration into a VLE.

3. Reflect changes to the learning and teaching materials to improve problem solving skills in relation to web development.

4. Better reflect web development in the meta-model constructs.

5. Provide a means of tracing requirements through an association model.

The structure of the learning model was not modified, although there were changes to the theory that underpin the intervention. As argued in section 6.2.1, substantial changes were necessary in order to enhance the student experience in the analysis of dynamic web application type requirements. The inclusion of additional models, such as support, guidance and an association model meant that the approach evolved beyond a method and could be better described as a framework.

In addition there was a shift from the use of user profiling as a way of eliciting requirements. It was thought advantageous to change the name of the approach to better reflect how it was to be used going forward. It was decided to rename the approach Web User Requirements Framework (WURF).

Figure 6.2.2 shows the updated structure of the WURF meta-model. The user profile construct was the only aspect that remained unchanged. The starting point became the statement of purpose / vision construct, with a sequential flow through to *Web Application Objectives*; *Web Actors*; *User Tasks*; *Functional Requirements* and *Non-Functional Requirements* constructs. The meta-model included the ability to model relationships between functional requirements and actors, something that was not possible in the first instantiation of the meta-model. The relationships within the meta-model are managed by the use of a number based identifier. No relationships are possible within non-functional requirements, since these are constraints that apply to the web application environment, such as the *user interface*, *security* and *usability requirements*.

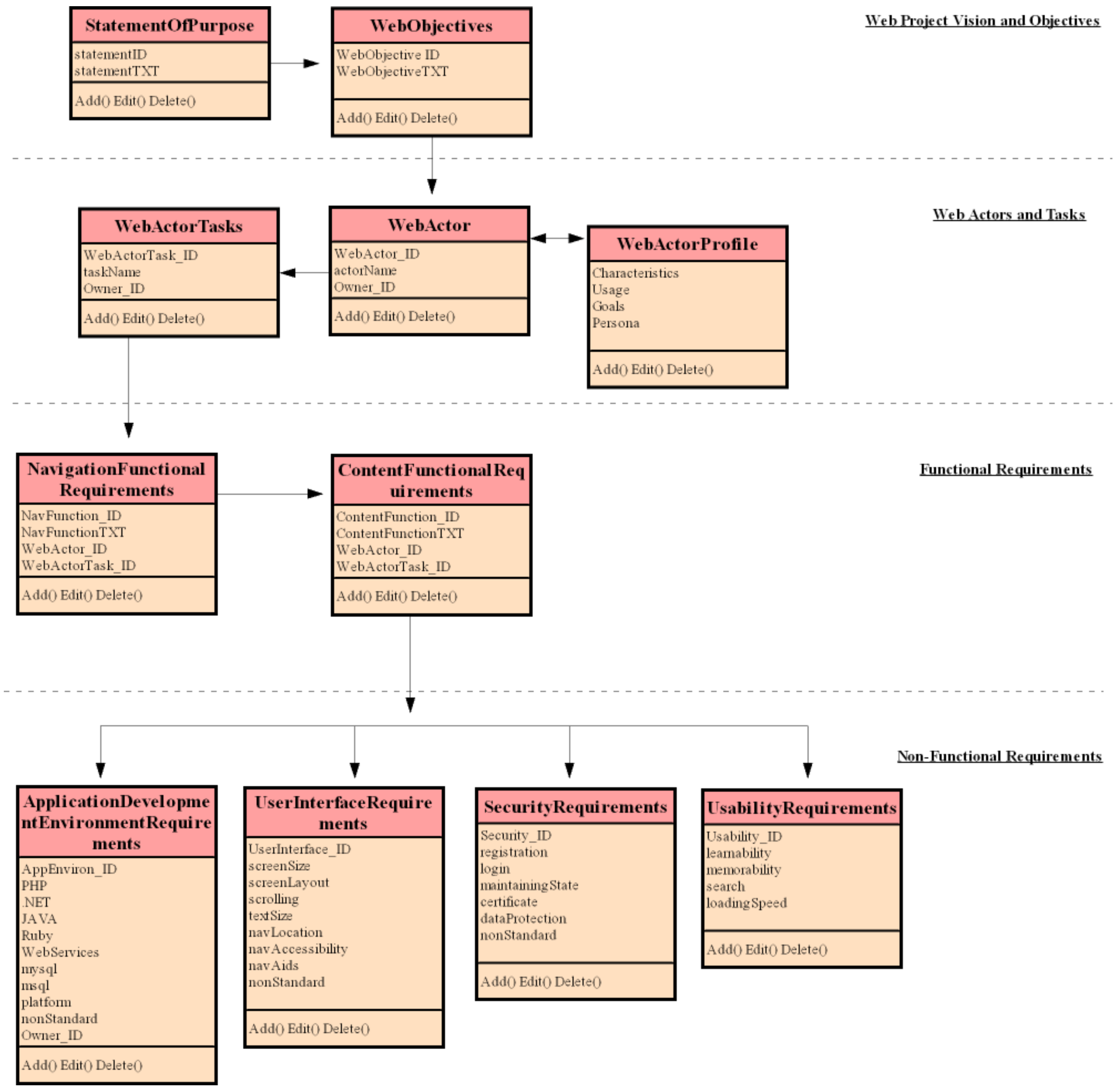


Figure 6.2.2 Updated Meta-Model (WURF).

The learning model was not modified in terms of structure, although a number of minor changes were made to the content in order to better support students with special needs. For example, it was identified that more attention should be paid to the underpinning theory, including access to real world examples, in order to address early problems in students' understanding of the approach. Video based tutorials were also provided in order to support part-time students, who needed to access learning material away from the laboratory. Confidence in the hybrid PBL model was established in the reflection and learning stage within the first cycle of research, but it was recognised that student support needed to vary.

6.2.3 Action Taking

The modified intervention was tested on two modules; a final year module named Online Business Systems (OBS) and a postgraduate module name Integrated Development (IID). Both used individual rather than group assessment. Their selection was deemed appropriate due to their focus on dynamic development and a curriculum that reflected the whole lifecycle of a web project. These modules extended over two terms, providing the researcher with an extended observation period and an opportunity for a deeper insight into how WURF was being used by the student.

Since the new meta-model had been extended by, for example, the ability to model relationships and a guidance model, the fill-in forms were only available to the students in an electronic format. It was decided to package and disseminate these via the institutional Virtual Learning Environment (VLE) in order to facilitate additional support. A written commentary was provided to the student within this package, along with a general overview of how the process worked and examples of the filled in forms.

The learning model used in the first cycle of research was again adopted to teach the RE and WE aspects of the curriculum. An introductory lecture on RE and WE provided the student with the necessary background to the project, including their role as active participants in the research programme. Evaluation of the first cycle of research identified that some learners had floundered in the initial stages due to ill structuring of the learning outcomes within the module guide. In the second research cycle the laboratory sessions were more clearly defined in terms of what was expected from the student, including more explicit learning outcomes for each session. This mirrored the learning model more tightly than in the first cycle of research and it was thought that this should reassure the student that they are on schedule.

As part of the initial sessions delivered to the students, the ICA was once again used to deliver information about the project and the types of users that would be using the web applications. Changes to the 'Actor Construct' in the meta-model facilitated enhancements to the ICA briefing document and more meaningful information could be provided to the student. The assessment process was conducted over the full academic year, rather than at the end, in order to reflect the real world web development lifecycle.

As in the previous research cycle, the 'User Centred Web Development Methodology' was adopted to support the web development process. Use of this was mandatory and explicitly written into the assessment criteria. Learning and reflection in the first cycle of research had identified 'completeness' as an area of concern. The assessment criteria for requirements analysis was modified to encourage the student to submit complete requirements documentation.

Formative and summative feedback was provided direct to the student by the tutor. In the case of the OBS module, summative feedback was provided at week twelve due to a staged hand-in regime. Formative feedback was provided within the IID module within the same timescale. At the end of both modules, more detailed summative feedback was provided. By providing timely feedback part way through the module, it was expected that the student could respond more positively and make changes, rather than at the end of the module when it is too late.

In light of the modifications to the meta-model, dissemination of support material and assessment criteria, it was found that the students were much more positive in their RE analysis. The initial hesitation and apprehensions that were apparent in the first cycle of research had lessened and the students seemed to comprehend the process much earlier than before. They were motivated to discover how things worked on their own, rather than requiring the tutor to explain things to them. In particular, the dissemination of the WURF package via the VLE resulted in some student's coming to the laboratory pre prepared, as they had engaged with the material after the initial introductory lecture.

The module guide had described the content of each lab session in much more detail than before, mirroring the main stages within the learning model. This more rigid approach seemed to reassure the student. One of the disadvantages of a PBL model is that the student can sometimes feel disorganised, as the pace of learning is dictated to them. The incidences of student intervention where the tutor had to provide one-to-one support were much lower, better reflecting the aims of the learning model. Some interventions proved unavoidable where the student had special needs or other learning difficulties. This type of student required additional support, as they could not manage their own learning in a PBL situation.

6.2.4 Evaluation and Learning

The purpose of this stage within the action research cycle was to assist in determining if the second cycle of research fulfilled its objectives set out in section 6.2.1 and to provide the researcher with data to interpret and for reflection purposes. It enabled the researcher to form an understanding of the modifications necessary to inform the third cycle of research.

Key Findings

This section presents the findings from the student survey and feedback from UKAIS conference. It evaluates both the action taking and research activities, using the objectives developed in the initial stages of this chapter as a basis for the evaluation.

Student Opinion Survey

Student feedback in relation to their use of the updated framework was undertaken in March 2009. The online questionnaire link was emailed from the institutions VLE after they had undertaken learning activities in relation to (see Appendix A5).

Questionnaires were returned anonymously to ensure an un-biased response and data was stored on a secure database. Seventeen students in total (n=17) out of a total of 84 responded, making a response rate of 20%. Some respondents chose not to answer some questions. Where this was the case, the value of 'n' is clear indicated in Table 6.2.5. The response rate was disappointing, as it had been expected that the move from paper to online would have enhanced response rates. Nevertheless, the response rate was higher than first survey.

The questions posed to the student focused on the 'construct' aspects and how WURF assisted them translating the requirements into the design for traceability purposes. A question concerning additional help required in order to complete WURF was thought essential, given the adopted learning model and its emphasis on self-directed learning. The survey also aimed to discover which aspects of WURF were most and least understood. The final question asked if students felt any changes were necessary to the meta-model or constructs.

Questions	Response Data (n=17)					
	a	**b**	**c**	**d**	**e**	**f**
1. Please indicate which course you are currently enrolled. a: Masters b: Computer Studies c: Web Design d: Creative Multimedia / Web and Multimedia Design e: IBIT	3 18%	6 35%	4 23%	3 18%	1 6%	
2. Please Indicate Your Mode Of Study. *a. Full-time* *b. Part-time*	13 76%	4 24%	n/a	n/a	n/a	n/a
Q3 Did you use WURF in your in-course assessment. a. Yes b. No	15 88%	2 12%	n/a	n/a	n/a	n/a
Q4a.I understood the process of WURF without the need to ask for help. (n = 15)	2 13%	11 74%	2 13%	0	0	n/a
Q4b. WURF helped me think about the user in terms of characteristics and their requirements within the web application.(n = 15)	3 20%	11 73%	1 7%	0	0	n/a
Q4c. WURF helped me think about translating requirements into tasks and functions within the application. (n = 15)	2 13.33%	11 73%	2 13.33%	0	0	n/a
Q4d. WURF takes too much time to complete. (n = 15)	0	3 20%	4 27%	7 46%	1 7%	n/a
Q4e. There's no benefit for me in using WURF. (n = 15)	0	1 7%	3 20%	7 46%	4 27%	n/a

Q4f. Before using WURF I had not thought enough about the user and their requirements. (n = 15)	2 13%	7 47%	3 20%	2 13%	1 7%	n/a
Q5. What step within WURF did you least understand? (n=14) a. Statement Of Purpose / Vision / Define Web Application Objectives b. Define Web Actors / Define Web Actor Tasks c. Define Web Functional Requirements d. Define Web Non-Functional Requirements	1 7%	3 22%	1 7%	9 64%	n/a	n/a
Q6. Which aspect of WURF did you feel helped the most? (n=15) a. Statement Of Purpose / Vision / Define Web Application Objectives b. Define Web Actors / Define Web Actor Tasks c. Define Web Functional Requirements d. Define Web Non-Functional Requirements	9 60%	5 33%	1 7%			
Q7. Would You Use WURF again? a. Yes b. No	15 100%	0 0%				
Q8. Now thinking about the 'user characteristic' stage. Would you prefer to use 'pre-written' persona's / profiles here rather than writing these yourself? a. Yes b. No	7 47%	8 53%				

Q9. Are there any additional enhancements that you would like to see incorporated? Please indicate these in the box below;	No responses to this question.					
Q10 Have you used another method for User Requirements Analysis? (n=17) a. Yes b. No	9 53%	8 47%				
	RUMM, Modified OOHDM, RUMM, RUMM V1, RUMM2, Personas.					

Table 6.2.5 Responses to online student survey.

Discussion of Results

Significant results are dealt with below where these have implications for the second research cycle.

Question 1 asked students to indicate their course enrolment. The majority of students were enrolled on the generic pathways. Computer studies, Web Design, IBIT and Web and Multimedia Design were all undergraduate programmes. Masters students accounted for 18% of responses. As such the results should be viewed in the context that the majority of responses were from the undergraduate programmes.

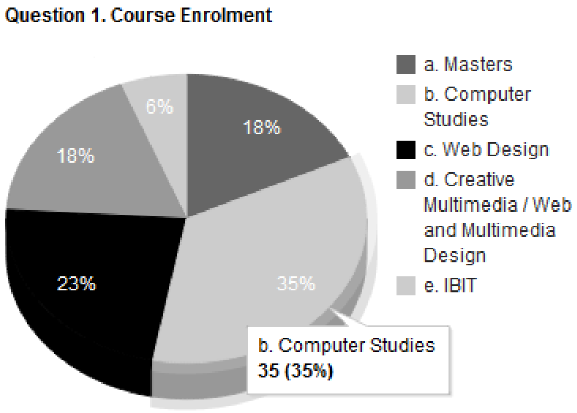


Figure 6.2.5 Results indicating student profile

Question 3 Asked students if they had used WURF within their in-course assessment. The majority had used WURF directly, with rest recording their requirements using an alternative mechanism.

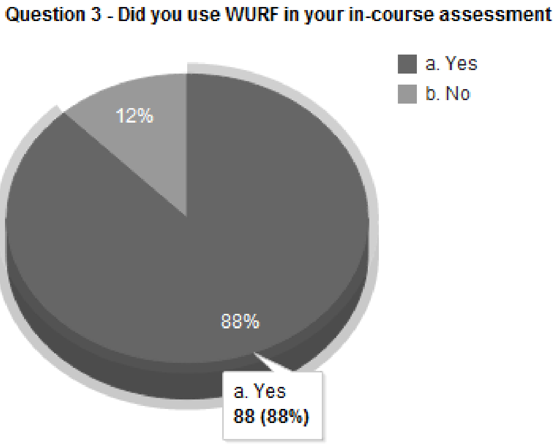
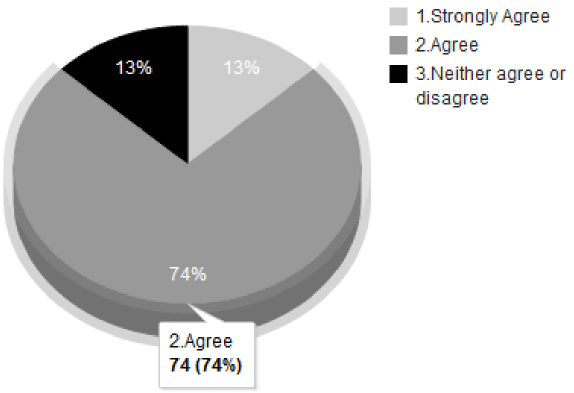


Figure 6.2.6 Use of WURF in the assessment process

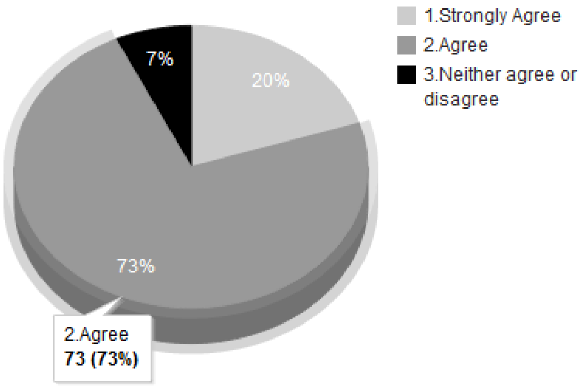
Question 4 contained six statements in which the student could record their responses using a scale based on the suitability of WURF for performing the requirements analysis. (The scale key is as follows: 1 = Strongly Agree 2 = Agree 3 = Neither agree or disagree 4 = Disagree 5 = Strongly Disagree)

The significant results from question 4 were that the majority of students were able to use WURF without the need to ask for help. This is an important result, as the second cycle introduced more student support to help them learn requirements without the need for continual intervention from the tutor. Student also indicated that WURF benefited them in their learning of requirements engineering and that this was not too time consuming for them.

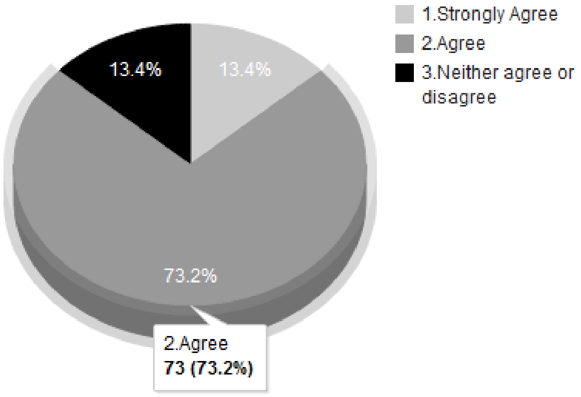
Statement 4a. I understood the process of WURF without the need to ask for help

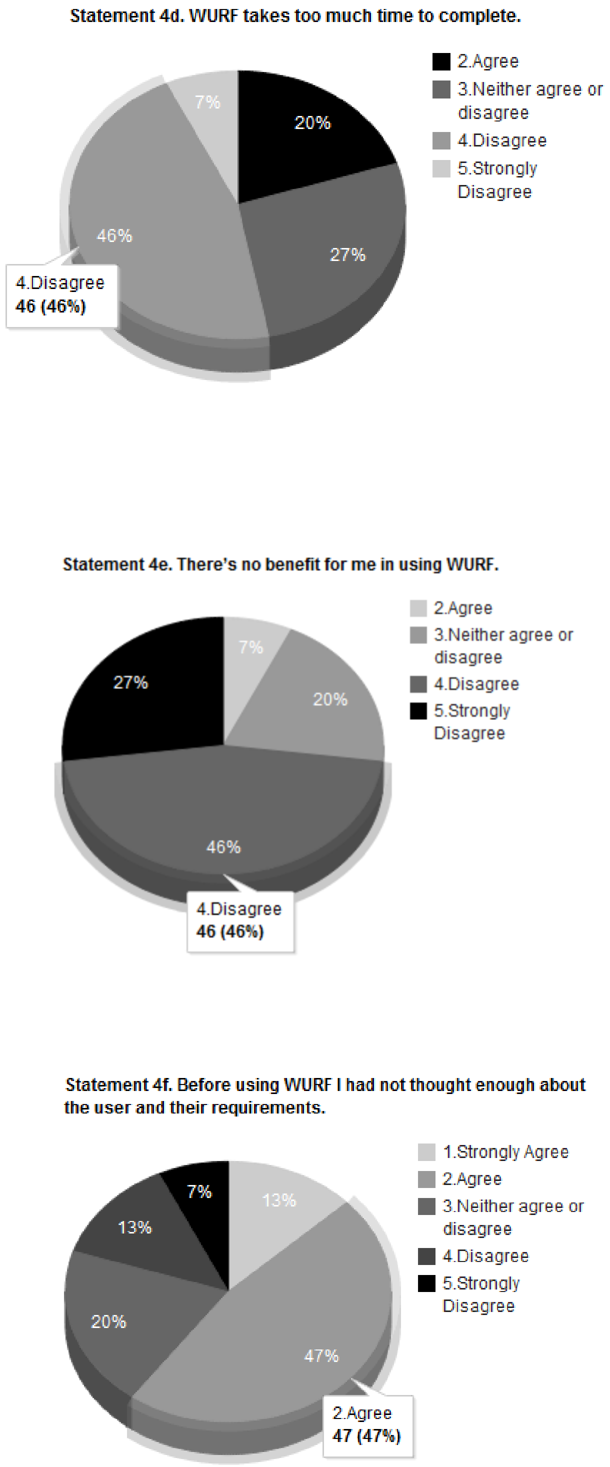


Statement 4b. WURF helped me think about the user in terms of characteristics and their requirements within the web application.



Statement 4c. WURF helped me think about translating requirements into tasks and functions within the application.





Statement 4d. WURF takes too much time to complete.

Statement 4e. There's no benefit for me in using WURF.

Statement 4f. Before using WURF I had not thought enough about the user and their requirements.

Figure 6.2.7 Responses to Question 4 statements (a to f).

Question 5 asked students which step within WURF was least understood. It had been expected that 'functional requirements' would cause the majority of problems for students due to its openness. Non-functional stood out for the students as the least understood step which was surprising given the 'check box' metaphor adopted. It is therefore evident that more work is required here.

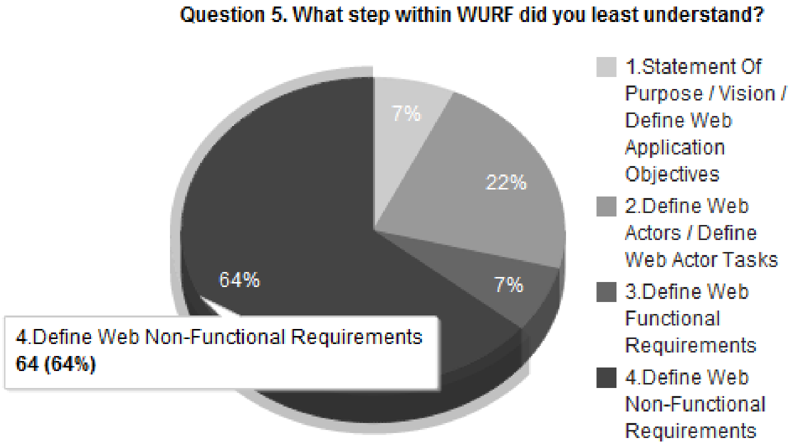


Figure 6.2.8 Least understood aspect of WURF.

Question 6 asked the student to indicate what aspect of WURF helped them the most. The responses to this were that it was the initial stage of WURF that helped them set the project in context. This supports the modifications to the entry point of the meta-model, where in the first cycle students indicated that this was confusing.

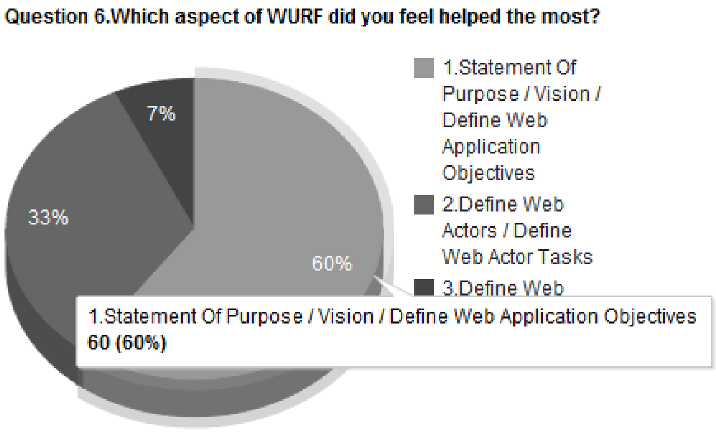


Figure 6.2.9 Most helpful aspect of WURF.

Question 10 asked students to indicate if they had used another method for requirements analysis. Although the majority said yes, the significance of this is not great, due to them being exposed to the first version of RUMM. The majority of students indicated that they had used this, with only one student highlighting that they had used OOHDM.

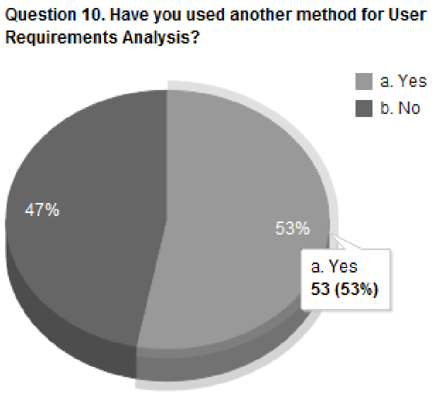


Figure 6.2.10 Use of another Requirements Analysis method

The overall interpretation of the results from the student survey is that students acknowledge benefits of its adoption in their learning. Changes to the student support mechanism had a positive impact, although there is still work to be done concerning the non-functional requirements. Reflecting on the way that the survey was undertaken, it is clear that the sample data is not well matched to the student profile (a mismatch between numbers on the undergraduate programmes that has to be taken into consideration regarding interpretation of the results). In the next research cycle it is hoped to make the completion of the survey part of the assessment to ensure a more representative sample. Open questions sometimes generate more meaningful insight into the students' perception of their learning. As such, it is hoped to complement the student survey with a follow up focus group.

Feedback from UKAIS conference

An updated research paper based on the key findings from the first research cycle (and presented at the SIGSAND conference in 2006) was written and presented at the United Kingdom Academy For Information Systems (UKAIS) PhD Consortium held at University Of Manchester in April 2007. The main aim of the phd Consortium was to allow phd students to present their work in progress to participants. Although attention was paid to the topic of the research project itself, the aim of the session was also to discuss methods of the research, including work in progress. It is important to recognise that work presented to this conference was 'work in progress' during the second cycle, before WURF was released to students. Feedback was oral in nature after the author had presented his work and feedback from the delegates was recorded in note taking form. A summary of the main points are outlined below:

- Research methodology adopted (CAR) is fine for this type of study.

- Need to bring out ‘educational use’ much more, as the context for the study is important.

- Can requirements ever be defined correctly? How does the adopted development methodology influence the approach to requirements analysis? Iterative nature of requirements needs to be emphasised if using agile methods.

- The study seems to cross a number of different disciplines such as Information Systems (IS), Web Engineering (WE), Software Engineering (SE) and educational research. Selection of external examiners will be crucial.

- Students as co-producers of the research must be brought out.

- Traceability of requirements needs further work.

6.2.5 Learning Through Reflection

Reflections on the Problem Solving Interest

Observation of WURF took place within a laboratory setting over the duration of one full academic year in two web based modules. Example of observational logs can be found in Appendix 2.4. Students engaged with WURF in conjunction with the User Centred Web Development Methodology (UCWDM) which was adopted for both modules.

Review of Problem Solving Objectives.

Objective 1. *Enable students to envision the web project before undertaking analysis of functional and non-functional requirements.*

The statement of purpose construct was perceived to be an important element missing from the first meta-model and inclusion in the new version was expected to reflect the discovery stage and web project envisioning that the student must undertake. Little intervention was needed at this stage, although it was observed in the ICA submissions that some learners were copying and pasting text from the briefing document.

Objective 2. *The framework should enhance student support by integration into a VLE.*

A VLE was used to distribute updated learning and teaching materials including additional support content. Students were often coming into the laboratory pre-prepared. Feedback was then provided regarding their progress. This cycle of continual feedback had a positive impact on the students' assessment submissions overall. Success was due in part down to the VLE, but this was only a means of transmission, with no real learning taking place within the VLE itself. Flexibility of learning was perhaps a successful effect of using the VLE, with students learning in their own time and at a distance. There is an opportunity to explore ways of enhancing learning in a dedicated environment.

Objective 3. *Reflect changes to the learning and teaching materials to improve problem solving skills in relation to web development.*

Students were provided with an updated set of learning materials, including an example in use. They were still expected to 'problem solve' on their own, although again some interventions were needed. WURF had a clear 'pathway' in terms of process, but some students still jumped straight to the definition of functional requirements, then returned to defining web actors, web objectives and tasks.

This problem was identified in the first cycle of research and it was argued that the issue could be resolved by modifying the meta-model to include the 'statement of purpose' construct in which they were expected to all complete in a particular laboratory session. The continuation of this practice was not envisaged in the second intervention. Consistency was therefore an issue in the meta-model impacting the validity of requirements and subsequently the quality of the implementation of the web application.

On a positive note, it was observed that the temptation for students to move straight to the implementation phase had been lessened due to the changes made to the intervention, including tighter integration with UCWDM and modifications to the meta-model. In particular, the fact that students had to pre-define requirements and receive feedback on these before they were implemented in the website, led to a change in their practice.

Objective 4. *Better reflect web development in the meta-model constructs.*

Non-functional requirements describe the web environment and constraints on the web application to be developed. These were fixed within the construct and the student could not modify or extend these further. Students were able to relate well to the constructs, for example, the web application development environment contained a taxonomy that described the server-side language and database. Further work is required in order to better represent the 'web aspects' of non-functional requirements.

Objective 5. *Provide a means of tracing requirements through an association model.*

As part of the assessment criteria students had to produce evidence in their submission for completing each stage of UCWDM including defining a vision document, defining actors and requirements, a conceptual design, a physical prototype and testing and evaluation. UCWDM is an established and documented methodology and students were introduced to it as part of the theoretical lecture. In the lab, students were much more confident in following the process and used WURF directly as part of their development. Evidence for this was the fact that the majority of assessment submissions contained comprehensive analysis and design documentation. Therefore in this respect, use of WURF in combination with UCWDM was successful.

Overall Reflection on the Problem Solving Interest

The wording of some of the constructs did appear to cause issues for some students' and this was where the majority of interventions had to take place. The term 'functional requirement' had to be explained in more detail. Some students mistook the term and described the functional requirement in technical language and others wrote code to describe a particular requirement. The term 'task' was less of an issue, but some students hesitated with this until some guidance was provided. This was especially true with the term 'web objectives' and students had difficulty distinguishing the difference between a 'task' and a 'web objective'.

Describing web objectives as high level business-type goals that could be measured in some way after the website had gone live, helped the student to distinguish the difference. Changes to the language used and the guidance model were therefore needed at this point. Tasks were easier to describe to the student, as the tutor could explicitly cite everyday examples, such as a login or registration task, linked to a specific actor.

Varying levels of detail were found in the overall submission of functional requirements, with some students using a cut down vocabulary and others writing in much more detail. It was not anticipated that this would be an issue before commencing the second cycle of research, but the type of language used could impact the quality of the requirement and the traceability within the design and implementation. This is something that needed to be addressed in the next cycle of research.

Modifications to the meta-model had other effects, such as an increased amount of time required for the student to complete it. It was felt that the increased work load placed on the student was offset by the availability of more complete and consistent requirements. The association model was also found to be an issue, in cases where an association of actors to functional requirements was problematic for the student to complete. Mapping each requirement to an actor within the electronic form proved extremely difficult for the student to achieve and this aspect of the meta-model required further work.

A solution to the issues outlined so far could have been to release the paper-based framework in phases together with the learning activities. This would have ensured each stage within WURF was completed and signed off by the facilitator, thereby guaranteeing consistency. Another solution would have been produce an electronic version of the framework.

Moving forward in terms of the next research cycle, the framework could be developed into a Computer Aided Software Engineering (CASE) tool, where the students 'flow' is controlled by a rules model. A rules model would define which parts of the meta-model had to be completed in order to progress onto the next. It could also control associations that have to be manually created at present within the paper based approach.

Additional benefits would also present themselves in adopting this approach, such as the inclusion of a data model to capture and log user activity. A guidance model would also provide student support during completion and could be referred to in 'real time' within the CASE tool. Revision and version control of the requirements would be possible within the CASE tool, as well as automated production of the requirement document.

Reflection on the Research Interest

The purpose of this section is to reflect upon the second cycle of research with regard to Canonical Action Research and the methods employed to evaluate both the method and learning aspects. Action research has proved useful in the way in which it controlled iterative feedback and the involvement of the researcher within the investigation. In the first cycle of research it was identified that direct involvement of the researcher in the testing and evaluation may impact the validity of the research. This was due to the possibility of bias on the part of the student. For example, the student could 'tell the tutor what he/she wants to hear to get a better mark' and the possibility of coaching the students in the use of the intervention to achieve a more positive outcome. On reflection, the responses to the survey indicated that students were able to provide critical feedback in the spirit of the research programme.

Response rates to the online survey were lower than had been expected. In the second cycle it had been envisaged to more tightly couple the student survey to the completion of their assessment. What is needed in the third cycle is a way of attributing a mark to the completion of the student survey. It is hoped that response rates will be much higher, providing much more confidence in the resulting findings based on their interpretation.

In the second research cycle the uncertainties surrounding the validity of research have been lessened but not completely dismissed. Feedback from the student was balanced and informative, leading to the identification of modifications and additions to the meta-model. Effective working relationships established with the students over the duration of the year proved to benefit the quality of feedback, as it was found that a mutual understanding emerged in discussions within the lab. This openness resulted in useful insights that would not have emerged if the researcher had been isolated in the evaluation process.

Uncertainties were mainly centred upon the student interventions that were required in some situations. For example, the web actor / functional requirement association model proved to be problematic for the student to understand. The validity of the research could be called into question here, as without the intervention, some students may not have been able to produce their requirements. In defence of this, it can be argued that this research took place in a teaching situation, which made interventions inevitable. By not making those interventions, the delivery of learning would impact the validity of the module itself. Research and teaching interests are so tightly integrated they are impossible to separate. It can also be argued that the benefits of adopting action research outweighed any negative aspects to the research overall. The adoption of action research and its use over two research cycles elicited further benefits such as:

- The ability to set clear research objectives at the outset of each cycle.

- A separation of the problem solving interest and research interest reflections.

- The direct contribution of the student in shaping the direction of the investigation, resulting in student collaboration in the research.

- The emergence of themes and issues that could only be brought about by cycles of problem identification, testing and evaluation.

Perhaps the most important outcome of using action research in this investigation was the feedback from the students and observation of the intervention in use by the student within a laboratory setting. In particular it is felt that the contribution by the learners has strengthened the validity of the research. Canonical Action Research benefited the research programme by ensuring a considered process was followed in terms of research methods and in the interventions used to test ideas born out of the reflection process.

Reflection of Research Cycle 2

Four research objectives were established within the second cycle of research. These will now be reviewed in the context of the research interest within the action research approach:

1) *To investigate ways of extending the meta-model to better support the inexperienced student user to define dynamic web requirements.*

   In the evaluation of the problem solving interest it was established that the meta-model was restrictive in defining requirements relating to dynamic development. In the second cycle of research the investigation reviewed a number of existing web specific approaches that enabled the definition of dynamic web requirements, as well as a more open taxonomy. Changes were made to the meta-model and electronic fill-in forms.

   Significant modifications were also made to the entry point of the meta-model in order to better reflect the requirements elicitation and analysis process. This was found to be problematic in the first cycle. It was found that by representing the 'cognitive flow' of the student, for example, from reading the initial ICA brief, through to establishing specific web objectives, learners were able to see a 'richer picture' of the web project. They were also able to extrapolate more detailed information about the project than they would have achieved in cycle one.

2) *To establish how relationships between requirements and actors can be modelled.*

   By modelling the relationships between requirements and actors it was hoped that the student would set out more focused requirements. In terms of the ‘association model’ this was achieved in the electronic forms by manually associating an actor to a specific requirement by the use of number based identifiers. However, it was less certain if this had any direct correlation to the quality of the requirement that was developed by the student. They also provided feedback on the association model and it was observed that the manual association was problematic in its existing configuration. The principle weakness was that the association was based in a document centric application.

3) *To evaluate students’ opinions regarding the updated meta-model.*

At the end of the module students were asked to complete an electronic questionnaire detailing their experiences using WURF. The move to an electronic questionnaire was thought advantageous in both response rates and analysis of the survey data. In actuality a much lower response rate was achieved, although useful data was still collected. On reflection, completion rates could have been enhanced by better promoting the online questionnaire in class time or by rewarding students in some way. Means of enhancing completion rates were investigated in the next cycle of research.

4) *To demonstrate that an updated meta-model can be used in the teaching of web development.*

Reflection on the problem solving interest highlighted that WURF achieved a much improved process successfully mirroring the dynamic nature of web development. It was possible for both functional and non-functional requirements to be produced, with the former providing a more open taxonomy for the students. Non-functional requirements provided a mechanism to capture both the design and technical aspects of the web application. On reflection, the decision to extend and modify the meta-model had a positive effect.

Reflection on the second cycle as a researcher

In terms of enhancing the validity of the research in the next cycle we can reflect on the methods of evaluation in this cycle. A number of methods were used to evaluate the intervention, but these were perhaps too focused on qualitative evaluation. In the next cycle there is an opportunity to employ multiple methods including statistical analysis of assessment data. Richer data may also be gathered from the students by interviewing them directly, rather than at a distance using a questionnaire. By employing multiple methods, cross comparison of data may also be possible thereby providing a greater understanding of the impact on student learning and support.

Use of CASE tools to support WURF has already been mentioned. Ways of recording usage data associated with the completion of the process would also provide a means of statistical analysis. For example by analysing usage data particular patterns of student behaviour may highlight problems or provide opportunities to enhance support mechanisms.

Student engagement in the surveys was an issue in the first and again in the second research cycle. In order to increase response rates, it is hoped to more tightly couple survey completion with the assessment process. Use of a survey prior to completion and then again post completion may also provide a greater level of understanding of how students perceive RE within their own development practice.

6.3 Conclusions

The second cycle of research focused on the refinement and expansion of the meta-model, where deficiencies had been identified in the evaluation of the first research cycle. A number of possible solutions to the problems identified were found in a review similar of work, where the treatment of dynamic requirements, the principle problem in the first meta-model, could be achieved by the separation of requirements into two distinct constructs. A functional requirements construct enabled a more open taxonomy, where this had been limited in the first meta-model. Non-functional requirements could be expanded to include web environment constraints such as server, language and databases. Significant modifications to the learning model were not required, although changes were made to the learning materials and assessment to better support the student. Changes to the entry point of the meta-model, along with additional constructs such as tasks and web objectives, provided further enhancements to the intervention.

Supporting documentation and 'examples in use' were created to enhance existing materials as part of the overall package. Fill-in forms included a range of electronic documents that worked collectively to aid the student user in their elicitation, analysis and specification of web requirements. These were delivered via a VLE to better reflect the needs of the student. The action taking stage enabled the intervention to be tested on two year long modules in order to provide the researcher with an extended observational period.

Evaluation of the problem solving interest within the second cycle identified successful aspects of the intervention, as well as opportunities for changes and enhancements. It was established that the traceability between requirements and the design/development artefact was much improved, with evidence for this in the ICA submissions. The new entry point for the process, which better represented the formulation of web objectives and a statement of purpose, had resulted in a closer relationship to the learning model than had previously been seen. An association model had been included within the meta-model in order to demonstrate to the student that each requirement had a distinct association with an actor. This aspect was not deemed to be successful, due to the limitations of the electronic forms that were used to distribute WURF to the student.

During the reflection on the learning aspects of the research method, it was established that some of the problems raised in the evaluation could be resolved by use of a CASE tool. Potential enhancements to the intervention included the transformation of all the models into one coherent framework. For example, a rules model could control the consistency, completeness and correctness aspects of the process model by ensuring ordered completion. A guidance model could provide support during the completion of each stage in the process model. The collection of completed constructs could automatically generate a requirements specification. In addition, a data model would ensure that information could be saved and amended thereby providing a version control mechanism.

The evaluation of the research interest identified the strengths and weaknesses of using action research in this investigation. Concerns about the legitimacy of the researcher's direct involvement in the delivery of learning were raised. It was argued that the benefits outweighed the concerns about the validity of the research outcomes. The biggest advantage in using canonical action research was the way in which new understandings emerged through the cycles of diagnosing problem, action planning, action taking and evaluation. A contribution to knowledge in this area has been made by undertaking an investigation into RE and WE within an educational context and by understanding the complex relationships between how the student user develops their understanding of WE and its traceability to their design and development artefacts. The next chapter represents the third and final cycle of research undertaken as part of this research programme.

Based on the reflection of the research interest in section 6.2.6 the following points will form the basis of an updated set of research objectives in the third research cycle:

1. Examine how the various models that comprise WURF can be incorporated into a CASE tool that students can use within a Virtual Learning Environment (VLE).

2. Explore ways to better represent requirements that are generated by the student user in the final specification.

3. Investigate how consistency, completeness and correctness can be incorporated into the process of generating valid requirements for the specification.

4. Examine how WURF is being used by the student user within a CASE tool including the possibility of logging usage data.

5. Understand how to enhance student support within a CASE tool during its completion.

6. Establish a way of measuring traceability between requirements and design/development artefacts.

The next Chapter demonstrates how a modified framework can offer way of addressing the concerns expressed in the reflection of the second cycle. It also demonstrates how students can be better supported through encapsulating WURF in a CAWE tool.

Chapter 7 Research Cycle Three

7.1 Introduction

The previous chapter reported on the second cycle of research that focused on the diagnosis of the problem, the gaps in knowledge within RE and WE and the planning and testing of an intervention. An evaluation of the problem solving interest in the first cycle of research found that deficiencies in the meta-model caused problems for the student in the completion of their web requirements, although there were also positive outcomes, particularly in the acceptance on the part of the students to follow a requirements process. In the second cycle, efforts were undertaken to address these deficiencies whilst building on the positive aspects. Again, the intervention was tested on two modules. In the evaluation of the action taking in the second cycle of research, it was established that although there were improvements to be made, the meta-model could provide the basis for further development and refinement in the third cycle of research. A number of ideas emerged during the evaluation of the learning interest, the most significant being an opportunity to package the various models that comprise the framework within a CASE tool in order to better support the student user. By consolidating the rules model, support and guidance model, association model and learning model into the CASE tool, it was hoped to solve outstanding issues highlighted in the evaluation. These included issues with the association model where it was difficult to associate actors to specific requirements. In addition, the separate and unorganised set documents that made up the ‘requirements specification document’ in the second cycle, could also be transformed and consolidated automatically in a CASE tool.

7.2 Action Research Cycle Three

7.2.1 Diagnosing and Problem Identification

The second cycle of research had identified a number of strengths and weaknesses of the modified intervention. Changes to the Web Actor construct, which enabled multiple Actors to be allocated to specific tasks and functional requirements, was the most successful aspect. The modified entry point, which used the statement of purpose instead of the user profile proved to be more intuitive for the student. Feedback from the students and analysis of their ICA submissions found that the modified framework encouraged them to create requirements based on evidence. The evidence was documented within the ICA brief document handed out to them before the project commenced. Using the 'statement of purpose' as the new entry point proved to be critical in allowing the student to comprehend the project as a whole, before analysis of functional and non-functional requirements took place.

The reclassification of design type requirements into functional and non-functional brought with it benefits, but some problems for the student. To mitigate this, they were provided with working examples in order to learn more about how functional requirements work in practice. Packaging WURF for dissemination within VLE allowed the student to access materials in their own time, as evidenced by some students coming to the laboratory with WURF 'fill in forms' pre-populated. In the second cycle of research the modified intervention resulted in a closer adherence to the learning model, with students taking charge of their own learning within the module, both in the lab and in their self directed learning. Additionally there were changes in practice concerning analysis of requirements in the lab, with the student less likely to move to the implementation phase before completion of the analysis and design stages within the User Centred Web Development Methodology.

Weaknesses in the second intervention included the *consistency, completeness* and *correctness* of the requirements, for example some students submitted incomplete documentation within their ICA submissions. It was also found that some students were skipping parts of the process, for example, defining Functional Requirements before Web Actors. As a consequence, the association between Functional Requirements and Web Actors was not complete leading to an in-correct requirement specification.

Some students indicated that they wished to extend Non-Functional Requirements, but were prevented from doing so in the second iteration of WURF. Greater flexibility in the taxonomy would result in the students defining more precise Non-Functional Requirements more accurately reflecting the web project. The association of both 'Web Actor Tasks to Web Actors' and 'Functional Requirements to Web Actors' had to be manually completed by the student. It was found that some students had trouble keeping track of this, especially if Web Actors were later modified.

Context For Research Cycle Three

A second year module named Web Authoring (WAU) and a final year module named Online Business Systems (OBS) were chosen for the third research cycle. Both modules were year long, enabling an extended period of observation and an opportunity for evaluation to take place at key points throughout the year. Key points included in-class observation where formative feedback was provided direct to the student; an evaluation survey from the students' perspective; a focus group at the end of the module and observation of assessment and usage data.

All students were provided with a theoretical session, in line with the learning model, that provided a background to RE and WE. Students were also advised of the ‘research informed teaching’ aspect and that they would become part of and contribute to the research programme.

Changes to the module schedule, assessment criteria and briefing document had to be carried out to reflect the changes in approach in this research cycle. For example, the electronic dashboard was demonstrated along with how the association and dependencies checking routine worked. In common with research cycle two, formative feedback opportunities were explicitly set out within the curriculum and scheduled for both modules within the module guide.

Both modules followed the unmodified learning model, again adopting a hybrid PBL approach that allowed the student to discover how RE would be applied to their web projects. The CAWE tool was used mostly in the early stages of the module and then referred to by the student in their design and implementation phases. This allowed them to refine requirements in parallel to their implementation.

In addition, the learning and reflection stage of the second action research cycle, documented in Chapter 6, additionally highlighted topics that needed to be explored further in order to address the research objectives. The review was presented in Appendix A3. The following section provides a summary of how the review has influenced an updated approach in the third research cycle.

1. Examine how the various models that comprise WURF can be incorporated into a CASE tool that students can use within a Virtual Learning Environment (VLE).

The evaluation of learning in the second research cycle identified an opportunity to consolidate the various models that comprise WURF into one tool. In Software Engineering these tools are often referred to as Computer Aided Software Engineering (CASE) tools that support a variety of stages within the software development lifecycle.

The review undertaken in Appendix A3, describes in detail the work of Casteleyn, et al., 2009 that outlines number of CASE tools which specifically address Web Engineering. Casteleyn defines the term Computer Aided Web Engineering (CAWE) in order to differentiate these from CASE tools. WebRatio and VisualWade exhibited characterictics of CAWE tools in the way they are able to support hypertext design, data abstraction and code generation. It is felt that although WURF does not support code generation, there are aspects of these tools that would benefit WURF. These include:

- Automated production of requirements before the student commences implementation.
- A graphical representation of requirements.
- Offer additional ‘in tool’ support to the student, including help and feedback opportunities.
- Impose a set of rules for completion of requirements.

The CAWE tool would support the early stages in the web development lifecycle which are neglected at present and support the inexperienced student user in their requirements elicitation, analysis and specification. Additionally, a CAWE tool could assist the completion of requirements by enforcing a process model. This would solve issues seen in the assessment stage, where student have submitted partially completed requirements documentation.

In addition, a CAWE tool could enable automation of the rules model in other ways. For example, the completeness of non-functional requirements, Web Actors and Tasks and the association of Web Actors to Tasks and Functional Requirements. These have to be manually represented in WURF at present, whereas in a CAWE tool, their association could be part of an additional 'in screen' process.

A requirements specification could also be automated and based on a rules model. Conformance to the rules model could be represented in the student dashboard, with visual cues to indicate completeness of the process.

Tutors could also be provided with a rich picture about the degree to which the students are completing their requirements, with the potential benefit that they would be able to detect students who have not engaged with their requirements analysis. Much of this would be dependent on storing user data, represented by datasets of each aspect of the meta-model. By storing data, students would also be able to edit their requirements throughout the web project lifecycle. The importance of this within Web Engineering was discussed in Chapter 2, as requirements cannot be understood well at the outset and are refined over time.

2. Explore ways to better represent requirements that are generated by the student user in the final specification.

In the second iteration of the framework the student had to manually translate the information contained within WURF into a human readable format. This is often cited in the RE literature as a Software Requirements Specification (SRS). It was identified in the evaluation in the second cycle of research that students often used non-standard specification templates and this sometimes proved difficult to assess at the end of the module for feedback purposes.

The review of similar work in Appendix A3, describes two example templates that could be adopted for the automated requirements document. Robertson and Robertson, 2010 provide a commercial template called VOLRERE that mirrors some aspects of WURF including the ability to summarise project details, functional requirements and non-functional requirements. These aspects could help support the student by standardising the way their requirements are documented. In addition, the VOLRERE approach provides a way of tracing individual requirements via a unique identity. Again, this is something that could be achieved in WURF by adopting a CAWE tool approach. Individual requirements could be saved in a repository and given a unique identity and documented in a template.

An additional example cited in Appendix A3 Related Work, section 3.6.5 was a software requirements specification used by a large public sector organisation. This provided some useful guidance on the need to track multiple versions of documents that would be produced by the student. Version control would need to be built into the requirements document to ensure the most up to date version is in use.

3. *Investigate how consistency, completeness and correctness can be incorporated into the process of generating valid requirements for the specification.*

This thesis has argued for the need to capture requirements in a natural language in order to reflect the needs of the inexperienced student user and that the intervention is centred upon a natural language approach. The proposal to transform WURF into a CAWE tool raises some questions concerning the direction that should be taken. Adoption of a more formal approach to represent requirements would enable the rules model to valid requirements more precisely. This would be at expense of ease of use, especially as the student would need to learn a new way of expressing requirements. Alternatively, use of a natural language adopted in the first two methods would bring continuity in learning for the student, although a rules model would be more difficult to enforce.

A key attribute of the CAWE tool would be to control *consistency, completeness and correctness* of the requirements. This would be achieved by incorporating the following:

- An early warning detection in student dashboard via 'requirement change flag' so as to alert the student to check that there are no conflicts with other requirements or that key information has been lost.
- Only allowing production of the Requirement Specification once all FR/NFR's have been written. Pre-conditions for this could be written into a rules model. For example, the rules model would check the *consistency, completeness and correctness of requirements* before allowing the production of a SRS Document.
- Version control to ensure the most up to date specification is in use. This would also facilitate the tracking of changes to the requirements in order to revert to a previous version.

4. *Examine how WURF is being used by the student user within a CASE tool including the possibility of logging usage data.*

Data mining is an established and proven technology within e-learning environments. In research carried out by Romero and Ventura into educational data mining, a number of techniques to record data generated by the learning environment included use of server logs, session cookies, transactions and condition filtering. This enabled data mining techniques to be used in order to improve the learning environment. "Web-based learning environments are able to record most learning behaviours of the students and are hence able to provide a huge amount of learning profile. Recently, there is a growing interest in the automatic analysis of student interaction data with web-based learning environments. In order to provide a more effective learning environment, data mining techniques can be applied" (Romero and Ventura, 2007).

The learning behaviour of the student could be recorded via system generated logs that could be retrieved for future analysis. The benefit of being able to track student behaviour includes the ability to identify trends in usage that may indicate problems. In addition, it was thought that the logs could be accessed in real time to provide a rich data set for the 'student dashboard' within the CAWE tool. This would represent 'automatic analysis of interaction data' cited by Romero and Ventura, 2007

In order to achieve this, a data logging function must exist within the CAWE tool. This would log each interaction that takes place including time and date stamps as a way of tracking usage of individual students within the CAWE tool. This is essential for the discovery of trends that may not be apparent at the outset of the third cycle of research and for automatic analysis for 'student dashboard' purposes by a rules model.

5. Understand how to enhance student support within a CASE tool during its completion.

   Additional support would include a help system that provides general support, such as 'in screen contextual pop-up boxes', along with 'examples in use'. The help system could be included within each screen via a recognisable feature in order to provide instant assistance with its completion. The student dashboard would provide additional information and visual references as to the completeness of the requirements and what is needed next within the process. The student dashboard would provide real time feedback on different aspects of the process and control the rules model of the CAWE tool. In common with CASE tools, other support mechanisms would need to be provided. For example, students would need to be able to access support systems to help them access the CAWE tool, including a password recovery and password changing facility.

Accessibility and meeting the needs of a diverse range of student's with different learning styles also needs to be considered within the CAWE tool. Opportunities to extend the type of support on offer include being able to embed video based tutorials as an alternative to text based materials. This would better support the visual learner.

6. Establish a way of measuring traceability between requirements and design/development artefacts.

One of the main themes that emerged from the evaluation of learning in the second research cycle was traceability between the requirements specification and the student web implementation. For example, a student requirement specification document could contain a consistent and complete set of requirements that does did appear in the final website implementation. This would result in *complete* requirements, but a website that would be invalid.

Gotel, 1995 in Valderas and Pelechano, 2009, defines Requirements Traceability as "the ability to describe and follow the life of a requirement, in both a forward and backward direction. Forward traceability looks at both tracing the requirements source to the resulting requirements and tracing the resulting requirements to the work products that implement them" (Valderas and Pelechano, 2009). Requirements traceability is an emerging area within Web Engineering due to the growing number of model-driven development methods that are being proposed.

Valderas and Pelechano, 2009, propose an extension to their Web Engineering method (OOWS) by extending the meta-model to include graph transformations as a means of tracing requirements. Although not a specific requirements method, OOWS is able to model different aspects of the system using class diagrams which include dynamic and functional models to describe system behaviour. Traceability mapping is easier to achieve due to linkage to one or more of these models. However, WURF does not have the ability to model web application data structures, as this is not the aim of WURF. The aim is to aid the inexperienced student user to elicit, analyse and specify requirements, which would then link to a web methodology where further modelling and design would take place.

The connection between the physical web design and its interactions are therefore lost in the second iteration of the meta-model. One way to re-establish a link between the requirements and the physical website is to propose a testing and evaluation strategy to the student user. Within the User Centered Web Methodologies (UCWM), usability testing is an important phase, which focuses on determining if the prototype meets the requirements of the target user. Therefore UCWM could be extended to include a requirements traceability map. The map template could be produced at the end of the WURF process to aid the student user with the usability testing stage.

Research Cycle Three Objectives:

Three updated research objectives were determined from the evaluation and reflection on the key findings in the previous chapter.

1) To investigate how a Computer Aided Web Environment (CAWE) tool can support the inexperienced student user in their requirements elicitation, analysis and specification using a natural language.
2) To investigate how a consistency, completeness and correctness rules model can be incorporated into the CAWE tool.
3) To determine if usage of the CAWE tool influences assessment outcomes for the student.

7.2.2 Action Planning

The previous section outlined problems with the second intervention and discussed possible solutions based upon a review of similar work and by returning to the literature on Web Engineering and Requirements Engineering. In particular, these areas focused on how to better represent the requirements produced by the student in their specifications, how *consistency*, *completeness* and *correctness* and can be incorporated, how to enhance student support and the traceability of requirements. The overall intervention, including the CAWE tool, WURF and the various models that support it, are illustrated in Figure 7.2.2.

Problem Solving Objectives

1. Automate the requirements specification document thereby ensuring its completeness.

2. Facilitate the student in constructing consistent and correct requirements through additional support and guidance mechanisms.

3. Provide a student dashboard for completion of the requirements process.

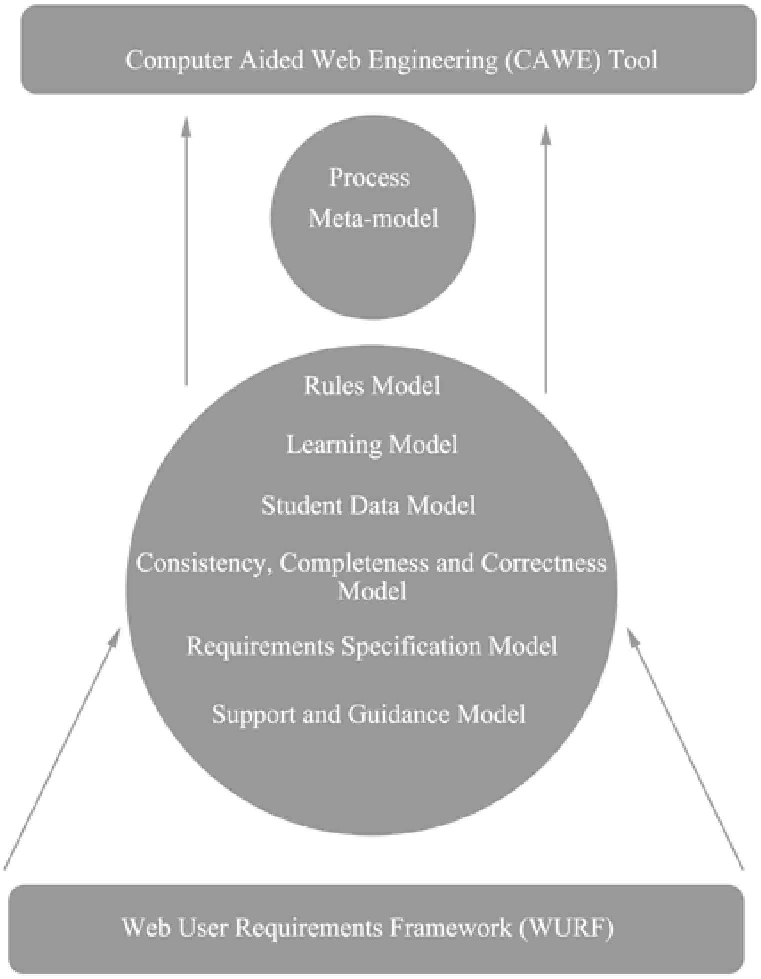

Computer Aided Web Engineering (CAWE) Tool
Process
Meta-model
Rules Model
Learning Model
Student Data Model
Consistency, Completeness and Correctness Model
Requirements Specification Model
Support and Guidance Model
Web User Requirements Framework (WURF)

Figure 7.2.2 Intervention Overview.

Modifications to the meta-model were needed in order to address the issues and requirements discussed in section 7.2.1. To better represent the transformation into a CAWE tool and to aid the development of a data and object model, the meta-model is now represented as a class diagram (see Figure 7.2.3). This describes the system (CAWE tool), attributes and relationships between the classes. The class diagram will be used for the transformation into programme code within the development environment. Appendix B1 describes the development of the CAWE tool, including a rationale for the chosen platform of delivery and the significant problems associated with transforming WURF into a CAWE tool.

Significant changes were not made to the learning model at this stage. Adaptations to learning and teaching materials were needed to ensure students were aware of the CAWE tool and how to use it effectively. In previous interventions, examples in use were provided in written form. Adoption of the CAWE tool opened up additional means of providing examples in use, such as video based tutorials. These were highlighted in the relevant learning and teaching material, in addition to their incorporation into the CAWE tool itself to provide 'in tool' support. A help system was also written for each individual screen, again incorporating an example in use.

Changes were not made to the assessment criteria from previous years. However, a new scenario was adopted in each of the modules in order to satisfy the institutions assessment regulations. An opportunity to provide more detailed information within the ICA brief was taken. This was achieved by modelling the ICA brief on a commercial briefing document that had been undertaken by the author in the previous year.

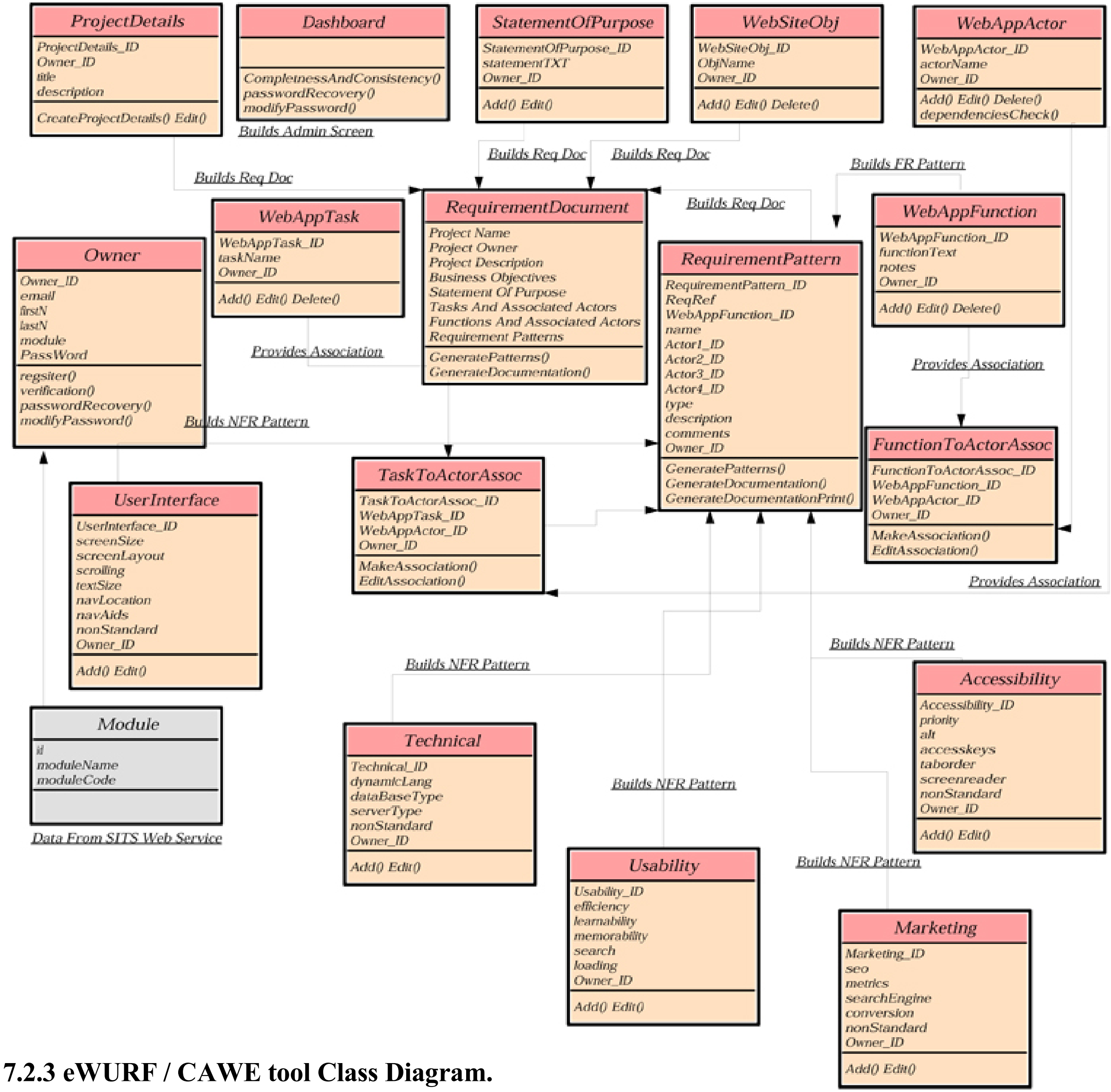


Figure 7.2.3 eWURF / CAWE tool Class Diagram.

7.2.3 Action Taking

The main aim of the third cycle is to establish how the CAWE tool could better support the inexperienced student user by solving a number of issues highlighted in the evaluation of the second research cycle. Some of the most prominent issues were centred on the *consistency, completeness and correctness* of the requirements that the students were submitting to be assessed.

Having planned and built the CAWE tool, two modules were once again identified to support students in their web projects in the 2009/10 academic year. The context for the modules is provided in section 7.2.1. In addition to these two modules, the undergraduate final year project leader had approached the author regarding adoption of the CAWE tool to support final year students undertaking web development type projects. A separate lecture was held with this group of students in order to facilitate an orientation session in line with the learning model. This was important, as there was an opportunity to capture data associated with their usage of the tool. As with the previous research cycle, the modified intervention was adopted as part of the curriculum and integrated into the module schedule over a full academic year.

Perhaps the biggest challenge in the final research cycle was the monitoring activity associated with the CAWE tool. In previous years, paper based documents were released using the institutions VLE. In the final cycle a bespoke application was designed and developed which required daily monitoring in the first few months to ensure there were no problems with the integrity of student data.

In common with the previous cycle, students were provided with a theoretical lecture on requirements engineering, along with a specific lab session for orientation purposes. Students were then expected to use the CAWE tool as they progressed through their web development process.

Observation of the students using the CAWE tool was recorded via a log book in the same way as the previous two research cycles. A detailed evaluation based on the observation can be found in section 7.2.2. The way in which the final intervention was to be evaluated was planned for in advance of the release of the CAWE tool, including building in a pre-use questionnaire and how different cohorts of students could be represented in the usage data. The final questionnaire would also be used in conjunction with a focus group to provide further evaluation opportunities.

7.2.4 Evaluation and Learning

Appendix B1 demonstrated how the CAWE tool was released and how the data it generated provided the basis for its evaluation. This section sets out the evaluation methods employed in the third research cycle, describes how the data was analysed and presents a discussion of the findings. It also provides a critical evaluation and reflection of the research programme as a whole.

Key Findings

The purpose of the evaluation stage was to assist in determining if the third cycle of research fulfilled its objectives set out in section 7.2.1 and to provide the researcher with data in which to interpret and reflect upon. It enabled the researcher to form an understanding of how the CAWE tool influenced the inexperienced student user to produce web user requirements in their modules. The following methods were used to evaluate the CAWE tool:

1. Pre-Use Survey.
2. Student Opinion Survey.
3. Student Feedback.
4. Student Focus Group.
5. Usage and Assessment Data – Comparative Analysis.

Pre-Use Survey

In order to better appreciate the student’s understanding of both Requirements and Web Engineering, participants were asked to complete a short questionnaire during the registration process of the CAWE tool.

A summary of the data is provided below:

1. Number of students who indicated that they have used a requirements analysis tool before using eWURF:

OBS = Online Business Systems (a final year undergraduate module).
WAU = Web Authoring (a second year undergraduate module).

OBS: **9 out of 45 students indicated that they had used a requirements analysis tool.**
WAU: **6 out of 40 students indicated that they had used a requirements analysis tool.**

Comments: *The data indicated that very few students had used a requirements analysis tool in prior modules. Students on the modules broadly fall into two categories 1.Web Design and 2.Web Development, with the latter more likely to have been exposed to requirements analysis.*

2. Three most recognised existing requirements analysis tools.
OBS: **Use Cases, Agile Requirements Method, Goal Analysis.**

WAU: **Use Cases, Usage Scenarios, Goal Analysis.**
Comments: *Use cases come through as the most recognised requirements analysis tool, although this does not necessarily mean that the student had used it in practice.*

3. Experience Level:

OBS: Experienced (**3**) Reasonable Experience (**16**) Inexperienced (**22**)

WAU: Experienced (**3**) Reasonable Experience (**19**) Inexperienced (**13**)

Comments: *Participants on the OBS module indicated that they were less experienced in Web Development. This is accounted for by the module being an elective option on a number of pathways, with some of these being non-web. As* *such, their knowledge of web methods may not be as high as on the WAU module, where the majority indicated that they had reasonable experience.*

Student Opinion Survey

An online survey was conducted between February 2010 and May 2010 for both modules using ‘Google Forms’ which is part of the ‘Google Docs’ cloud based service. ‘Google forms’ is a useful tool, as it automates the collection and production of spread sheets for later analysis. An email was sent to the students instructing them how to complete the survey, informing them of its purpose in the context of this research programme, a hyperlink to follow and an explanation that the information that they provided would be used confidentially. An example of the email sent to the students can be found in Appendix B1.2. Individual responses were tracked by use of a unique identifier so that results could be compared to previous questionnaire responses and cross referenced to their module mark and usage data. The questionnaire was also now part of the assessment process in order to increase responses from the students. In preparation for the online survey, a test questionnaire was generated to ensure data could be collected.

A total of **41** students responded to the survey, which represented **38%** of those involved on the modules. The response rate was much higher than the second cycle due to the linkage to the assessment process. OBS students generated the most responses, which represented **57%** of the module cohort, with WAU responses representing **31%** of the module cohort. Survey data, including a graphical representation of the data, are available in Appendix C3. The data was analysed by module and by combining both sets of results in order to identify general trends. The data provided an indication of how students used the CAWE tool across an illustrative sample. It is recognised that there are limitations with the data, such as its focus on two modules on distinct pathways.

Some general observations became apparent in the analysis of the student opinion survey:

- **96%** of OBS students and **92%** of WAU student felt that their web development process had been enhanced through the use of the CAWE tool.

- Respondents indicated that the *least useful* aspect of the CAWE tool was the **non-functional requirements**.

- Respondents indicated that the *most useful* aspect of the CAWE tool was the way it produced the **requirements specification document**.

- **84%** of respondents had accessed the student help and guidance system.

Comments: It was evident in the response from the student survey that although there were areas that need improvement, generally, the CAWE tool had enhanced their learning of web user requirements. In particular, the survey supported the way in which the CAWE tool had been designed to provide support and guidance. It also had enhanced their web development process overall. Surprisingly, the respondents indicated that the least useful aspect was the non-functional requirements. This was perhaps due to the fact that this was not editable by the student, for example, they felt constrained by the fixed nature of selection boxes. Respondents felt that the automatic production of the SRS document to be the most useful aspect of the CAWE tool, as this was its main goal from the student's perspective.

Student Feedback

Students were asked to suggest modifications that would enhance their student experience. Comments included (note: these are un-edited from the questionnaire responses, which include spelling/grammatical mistakes):

- *Clarification on how to remove tasks. Formatting on associated tasks another areas could be improved, this maybe a Safari bug, a lot of scrolling is required. Sometimes radio buttons are not on the same line as the related answer, this can be seen in the Actor Profile page. I also nearly missed out creating Actor Profiles as they did not show under the Incomplete tasks on the home page.*

- *Although I managed to complete the accessibility and usability non-functional requirements sections I found that it could have been made clearer as to what information was trying to be received.*

- *When attempting to use the print safe option upon completion I found that if a lot of information had been entered then the forms wouldn't accommodate it and so would not be displayed. Other than that it was a very useful assistance tool.*

Most comments related to usability issues within the various screens, including variations in positioning of elements from browser to browser. The CAWE tool was built using XHTML/CSS and was tested on a web standards compliant browser and most of the issues here were related to Internet Explorer 6 and 7, as these browsers use a different 'engine' to render the html tags. This was fixed in future iterations of the tool, but it should be noted that this did not prevent students using the CAWE tool, but rather caused some usability issues regarding the on screen objects. Some of the students indicated that the non-functional requirements were not applicable to their project, but they still had to 'enter something'. This was an important point and will need to be investigated in future work.

Focus Group

A focus group was organised in order to collect qualitative data as a follow up to the online student survey. All students involved in the OBS and WAU modules were invited by email to attend an hour and a half session. This took place on the 26th of May 2010 at Teesside University, facilitated by the author. Five Students in total attended the focus group. Students were informed that the session audio would be recorded for later analysis. In the questionnaire survey a number of issues emerged that required further clarification and exploration in the focus group. In addition, some general questions emerged from the observation of the CAWE tool in use by the students within the laboratory. These formed the basis for the topics for discussion and were structured in the following way:

1. Usage of the Framework. *Influence on final design, user experience, traceability of requirements to the design and interactions. Use of natural language and terminology, use of the specification document, feedback, difficulties encountered. Understanding of the requirements process and group working.*

2. Methodologies, Adaptation and Evaluation Mechanisms. *Use of eWURF within the chosen development methodology, use without a methodology. Adaptation of the requirements specification document, extending eWURF to include an evaluation stage.*

3. Open Discussion. *Feedback from the participants on general usage of the CAWE tool.*

Analysis Of Focus Group Data

The audio recording of the focus group was analysed by selective transcription. A number of key themes emerged from the analysis:

Theme 1. eWURF influenced the web project as a whole, not just the requirements.

Participants Comments Taken From Audio Transcript (unedited)
“Acted like a checking device”, “helped initially to define the functional and information architecture”, “I Explicitly linked a php function to a functional requirement in eWURF”, “Able to keep track of what I was implementing”, “Helped keep track of what I was developing”, “Did not describe a php function, Did not go into fine detail in the functional requirement, just a general high level description”, “It is quite important that you can check back. Sometimes, you did something in eWURF, but did not implement it, but that is now documented and you can go back later to implement it”.

Comment: It was found that participants felt strongly that using eWURF had a positive impact on their web projects. This reinforced the main objective of the research, which was to support and guide the inexperienced student user to deliver a consistent and completed requirements specification. The consistency, completeness and correctness model within eWURF ensures that the user undertook activities in a sequential way. This enabled the student to arrive at a complete specification, rather than an unfinished document, that was indicative before eWURF was adopted within the curriculum.

Theme 2. A danger of using eWURF as a development methodology.

Participants Comments Taken From Audio Transcript (unedited)
"I got carried away using the tool, I have so much information contained within it" "..I did not use an overall development methodology. I used elements from other methods. I used a prototyping method and used eWURF to control the project."

Comment: Some participants had used eWURF as a development process, rather than using it to facilitate their elicitation, analysis and specification of their web user requirements. As a consequence, the students may not have considered the whole project lifecycle and concentrated on the early stages of the project to the detriment of others. Students' who had expressed that they used eWURF in this way, had used a prototyping approach in their web development.

Theme 3. Participants indicated no strong preference for natural language or formal notation for expressing requirements.

Participants Comments Taken From Audio Transcript (unedited)
"Formal notation or natural language – either would work for me. But a natural language is better in team development as the requirements document can be passed around the team and everyone will understand it."

Comment: One participant indicated that either natural language or formal notation would work for them in expressing requirements, as long as it enabled them to arrive at a specification for their requirements. Another participant expressed a view that natural language would be better for novices or in a team situation. As eWURF was aimed at the inexperienced student user, the decision to use natural language was thought to give more flexibility. Conversely, if a formal notation approach had been used, perhaps this would have provided greater validity and traceability to the specification.

Theme 4. Students had a better understanding of Web Engineering once they had used eWURF, but prior 'theoretical' learning helped them gain a better understanding before using eWURF itself.

Participants Comments Taken From Audio Transcript (unedited)
"Understood it better once I had used eWURF.", "had some prior appreciation of functional/non-functional as I had watched the video tutorial. This helped me use eWURF later on.", "The lecturer on Enterprise Web Development did some exercise with it first that helped when I used it later on.", "Lecturer did some theory on usability and accessibility, this helped me better understand these sections in eWURF.", "At the outset it was not clear to me what I need to do within eWURF. But I found the really good examples, how you do it and what you write down. What about other language support, eg Mandrin, French, German. Translator integrated into the guidance and support system. Google translator?"

Comment: It was found that participants preferred some prior learning material before they commenced using eWURF. Key terminology that was unfamiliar to the student presented problems, for example, the term ‘Accessibility and Usability’ used in the non-functional requirements caused some problems, but this was overcome by providing working examples. Participants indicated that they preferred to view a video tutorial and would want this to be available at the bottom of each screen.

Theme 5. Asynchronous feedback should be part of the framework in the future.

Participants Comments Taken From Audio Transcript (unedited)
“... to have tutor based feedback. Could ask questions, can’t ask a computer questions. A Dialog between the tutor and student is needed. It depends on the type of feedback. Eg, if its simply that I have not filled in field, the this could be computer (logic) based feedback. Anything else would need to be human based feedback.”, “Email function could be built in, email tutor the requirements document and then feedback will come back into eWURF.”

Comment: Participants indicated a strong preference for human asynchronous feedback. At present the guidance system is text driven, with no dialog or communication between ‘it’ (CAWE tool) and the student. One participant suggested that an *email work to tutor* button be incorporated, so that asynchronous feedback on the work could be achieved. The participant went on to describe how the tutor based feedback return to the CAWE tool, which would be annotated on the students SRS document. Other participants in the focus group agreed with this. It was felt that this was an important outcome of the focus group, reaffirming the need to cater for a wide range of learning styles.

Theme 6. The process meta-model was useful, but greater flexibility is required.

Participants Comments Taken From Audio Transcript (unedited)
“There were Enough stages, but would have liked to see requirements specification straightway”, “More flexibility required. Eg, jumping stages, rather than forcing to go through in sequence”.

Comment: Participants strongly indicated that they would have preferred to complete the various processes in a non-sequential way. Presently, the process meta-model does afford some flexibility, but does impose constraints on the order of completion. The ‘Consistency, Completeness and Correctness’ model that is part of the eWURF framework does provide a visual prompt for the user. This could be adapted in some way, in order to provide greater flexibility, whilst recognising that there are sequences in completion within the framework which must be adhered to.

Theme 7. Ability to continually redefine requirements was important.

Participants Comments Taken From Audio Transcript (unedited)
"I thought that I would only need 2 actor when I started, but then changed my mind as I went through", "more than 10 Iterations to get to a finalised specification document", "More than 5 iterations", "Changing and updating ability of requirements was a positive. Ideas developed when implementing functions and I found that these had to be then reflected / reaffirmed iteratively in eWURF", "Iterative nature of web development means that it was a necessity to be able re-visit requirements in cycles".

Comments: A feature that was noted in the review of existing requirements methods was a mechanism to 'lock' requirements once they had been defined. This was not something that eWURF incorporated, due to the changing nature of web requirements and the necessity to continually modify requirements to match iterative development. Participants thought that this was a key benefit within eWURF. Analysis of the usage and assessment data indicated that students had used eWURF in this way, often returning five or more times to define additional or redefine existing requirements.

Theme 8. Data and knowledge sharing would enhance eWURF from the perspective of the inexperienced student user.

Participants Comments Taken From Audio Transcript (unedited)
“Peer learning (eWURF forum) student’s could post to a forum, student (peers) could suggest ways / responses that build a knowledge base. A rich resource that can be built up”, “Re-use data that is in eWURF, eg, functional requirements that are common across different projects. Actors, could be re-used. Students would be willing to share data amongst themselves.”

Comments: Participants suggested ways to reuse data that eWURF had captured. They indicated a strong preference for reusing existing functional requirements, where eWURF users would share their own and reuse data in their projects. A discussion took place on data protection and the dangers of using a ‘pick list’ approach, without analysis taking place. Participants still thought that whilst this is true, the benefits of learning from others would override this negative aspect.

Theme 9. Evaluation and testing of requirements.

Participants Comments Taken From Audio Transcript (unedited)
“This would be useful, after you have used it”, “A prompt to think about, aspects of design. Like a heuristic evaluation”, “Still ok to carry out evaluation and testing (still valid)”, “I did use eWURF as a prompt, so used it in this way”.

Comments: Participants were asked about traceability of requirements in their web artefact and if they had checked completion. A number of participants indicated that as eWURF had been used as a checking mechanism, they were to some degree already doing this, but in an un-structured way. A discussion then took place of possible ways to record which requirements were embedded within the website. Traceability of requirements is an important consideration and the whole premise of using eWURF is that the requirements will make it through into the actual website implementation. At present this is left to the student and assessor to validate. By modifying the meta-model to include a process for ‘requirements testing’, it is thought that the validity of eWURF be enhanced. Participants suggested that testing could be incorporated by the inclusion of a tick box against each requirement to indicated that it had been completed. Evaluation would be much harder to achieve, as this is often subjective, but would need to be end user driven, rather than completed by the student themselves.

Usage and Assessment Data

Usage data was generated by all student users of the CAWE tool, including student's on other modules and those completing their final year projects. During registration, users selected the module to which their requirements related. This provided a way to distinguish users on specific modules for analysis purposes.

Analysis of the CAWE Tool in use

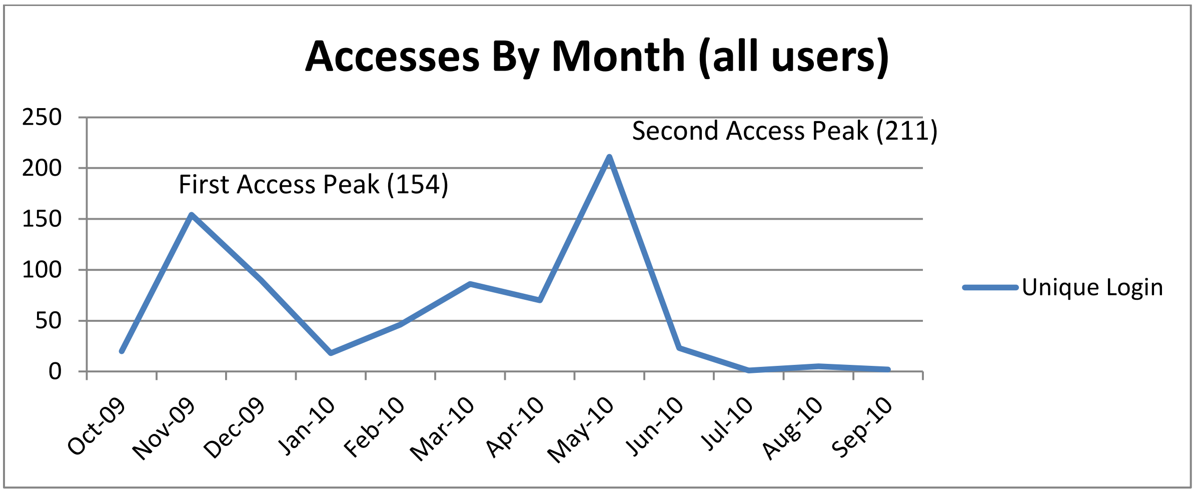
All Students (OBS, WAU and other modules)

Total Number of Registered students: **143**

Average Accesses Over 12 Months Per Individual: **4.1** (Standard Deviation: **6.2**)

Average Accesses Per Month of eWURF by all students: **60.5** (Standard Deviation: **66.1**)

Average Requirements Specification Documents Produced all students: **3** (Standard Deviation: **3.7)**



Comment: First access peak during November relates to the CAWE tool being used in class during the theoretical / practical tutorials. The second access peak during May relates to it being used prior to the assessment hand-in.

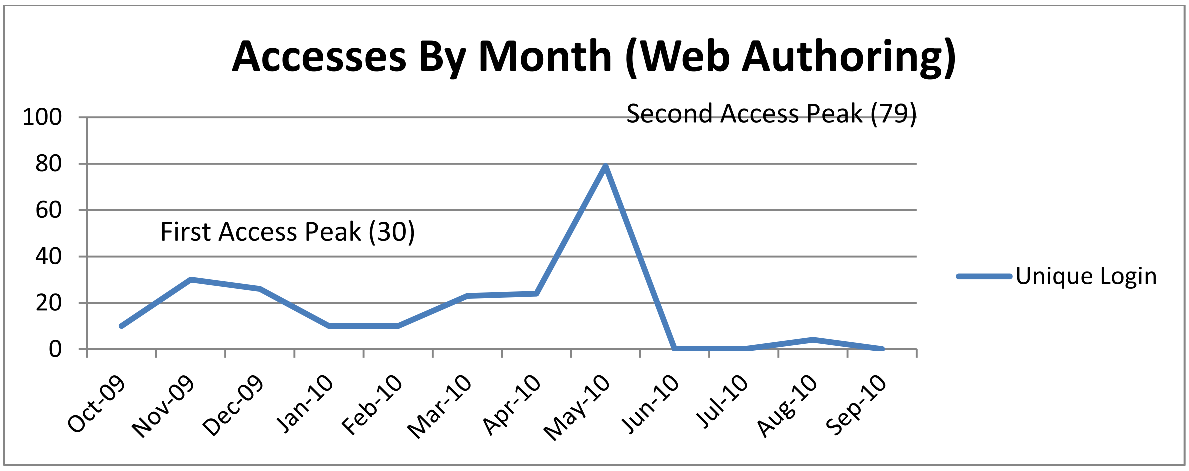
Web Authoring (WAU) Usage

Total Number of Web Authoring WAU students: **40**

Average Accesses Over 12 Months Per Individual: **5.4** (Standard Deviation: **7.1**)

Average Accesses Over 12 Months of eWURF by all WAU students: **18** (Standard Deviation: **22.1**)

Average Requirements Specification Documents Produced Per Individual: **3.8** (Standard Deviation: **5.3)**



Comment: Access peaks in November 2009 and May 2010 followed the general trend evident in the data for all users. Again Novembers' access peak related to eWURF being used in class during the theoretical / practical tutorials, with students logging in for the first time. The second access peak during May related to it being used prior to assessment hand-in, most likely due to changes in requirements and the generating of the SRS document. Interestingly, there was some usage apparent during August 2010 which can be attributed to the reassessment period.

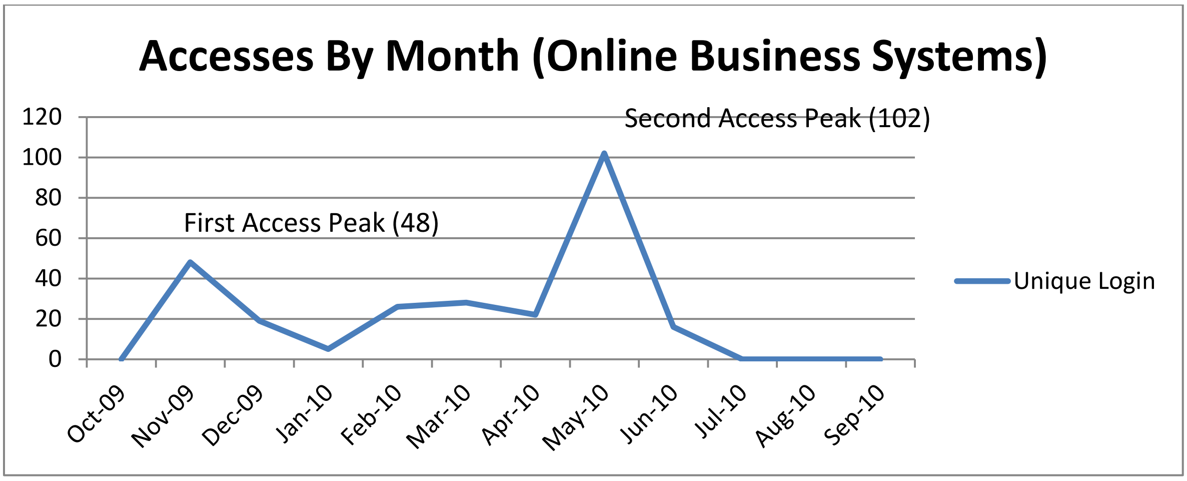
Online Business Systems (OBS) Usage

Total Number Of Online Business Systems (OBS) Students: **49**

Average Accesses Over 12 Months Per Individual: **7** (Standard Deviation **6.2**)

Average Accesses Per Month of eWURF by all OBS Students: **22.6** (Standard Deviation: **29.2**)

Average Requirements Specification Documents Produced Per Individual: **3.7** (Standard Deviation: **2.4)**



Comment: In general terms eWURF was used more on the Online Business Systems module than on the Web Authoring module. One explanation for this was attributed to the level of the Online Business Systems module being final year, and used to a greater depth.

Assessment Mark vs Usage Relationship Analysis

Statistical analysis was undertaken to determine if there was a relationship between the usage of the CAWE tool and the assessment mark attributed to the requirements analysis on two separate modules, Online Business Systems and Web Authoring. Although this research did adopt Canonical Action Research methodology, this aspect of the research entailed ‘quantitative data analysis’ generated via the log system within the CAWE tool and assessment data generated by the marking process. It was decided that the best way to analyse this data was through quantitative data analysis techniques, using hypothesis testing, as opposed to interpreting the data using alternative analysis techniques. (Please refer to Appendix C4 for data tables used in the statistical analysis).

Although the analysis presented here is useful, it is recognised that there are limitations with the data in relation to generalisability and transferability. Two distinct cohorts of students on particular modules have used the intervention and the same results might differ year to year and cohort to cohort. The nature of in course assessment is subjective and a different set of results may have been achieved under varying conditions, including the teaching methods adopted.

The **independent variables** used were:

1. Frequency of logins per student.
2. Frequency of requirements document production per student.

The **dependent variable** used was the assessment mark attributed to requirements analysis in the assessment submission per student.

Hypothesis:

HA1: Increased usage of the CAWE tool would result in higher marks for requirements analysis.

HA0: Increased usage of the CAWE tool would not result in higher marks for requirements analysis.

Spearman's Rank Correlation Coefficient was used having the following formula:

$$\boxed{r = 1 - (6 \Sigma d^2) / n(n^2 - 1)}$$

A. Online Business Systems (OBS)

n = 44

Mark to Frequency of Requirements Document Production Relationship Correlation Score: **0.329**
Mark to Frequency Of Logins Relationship Correlation Score: **0.328**

Common Significance Level: **.05**
Degrees of Freedom (n -2): **42**

Statistical Significance of Correlation Two-Tailed T Test: **0.304**

Comment: The 'Level of Significance Two-Tailed T Test' demonstrated the direction of the correlation and not its strength. In this case, where the correlation value is higher (**0.328**) than the significance level (**0.304**), it proved that the correlation is 'statistically significant' and we can reject a null hypothesis (H_{A0}). The difference between the T-test and correlation score supported the hypothesis that a relationship between usage and the mark attributed for requirements analysis existed. **Note:** 44 students out of 49 on the module undertook the assessment, which accounted for the variation in the value of 'n' from the previous usage data analysis.

B. Web Authoring (WAU)

n = 38

Mark to Frequency of Requirements Document Production Relationship Correlation Score: **0.226**

Mark to Frequency Of Logins Relationship Correlation Score: **0.372**

Common Significance Level: **.05**

Degrees of Freedom (n -2): **36**

Statistical Significance of Correlation Two-Tailed T Test: **0.325**

Comment: The correlation score for the frequency of requirements document production relationship to the mark given to the student was **0.226.** This demonstrated a weak correlation based on this test. The statistical significance level T-Test of **0.325** meant that the confidence of a correlation based on this test is lower than average and in this test the null hypothesis (H_{A0}) was therefore accepted.

The correlation score for the frequency of logins relationship to the mark given to the student was **0.372**. The statistical significance level T-Test of **0.325** meant that the confidence of a correlation based on this test was higher than average and in this test the hypothesis (H_{A1}) was accepted.

Analysis of the mark and usage data provided a useful insight into how the CAWE tool influenced the assessment artefact for requirements documentation. Interpretation of the correlation data pointed to some influence of the CAWE tool on the student's assessment mark, albeit weak.

7.2.5 Learning Through Reflection

Reflections on the Problem Solving Interest

Observation logs were recorded during lab and also during the assessment of student submissions. Excerpts of the logs can be found in Appendix A2.4. It must be noted that laboratory observations provided the basis for further analysis, especially in relation to the completeness of requirements and correlation between usage and assessment. The CAWE tool was produced to address the following objectives identified in the second research cycle:

Review of Problem Solving Objectives

Objective 1. *Automate the requirements specification document thereby ensuring its completeness.*

The assessment data indicated that requirements specification documents were more complete than they were before the paper based method was converted into a CAWE tool. A rules model prevented the student from printing off the requirements specification document until it was complete and so this was an expected outcome. During the assessment process it was found that some students did not print off their specifications and instead included a reference to the electronic version. An access tool had to be built to view their requirements and this is something that was not foreseen.

Generally, the standard of the requirements produced by the student were improved and tended use the type of language that one would expect Functional Requirement to consist of. Whilst this aspect was successful, the consistency of requirements was still an area for concern. For example, it was possible to submit a Functional Requirement without an associated actor, which resulted in inconsistencies within the association model.

The usage data indicated that the tool was accessed on average five times over the duration of the web project. This led to continual refinement of the requirements, reflecting the agile web development methods adopted by the student. As a consequence, it was felt that this aspect was the most successful.

Analysis of the student usage and assessment mark took place in order to establish if a relationship existed between how the student used the CAWE tool and their final mark. Spearman's Rank Correlation Coefficient was used to test the significance level of two independent variables (Frequency of logins per student and Frequency of requirements document production per student) against the dependent variable (assessment data). Interpretation of the correlation data pointed to some influence of the CAWE tool on the students assessment mark, although this was much weaker than expected. On reflection, trying to measure learning through these statistics had limitations. In particular, there was no benchmark data that could be used to make a comparison, for example, the previous years' assessment data. Even if there was, a direct comparison could not be achieved, as the CAWE tool had not been used in previous years. It could also be said that it was impossible to isolate the independent variables, as these would vary from year to year and cohort to cohort. Nevertheless, interpretation of data provided some generalised observations that benefited this research overall.

Objective 2. *Facilitate the student in constructing consistent and correct requirements through additional support and guidance mechanisms.*

Students found the embedded support and guidance mechanisms useful at the outset and during completion of their web requirements. Whilst they found this to be of benefit, the evaluation of eWURF highlighted the need to extend the support mechanisms further. Participants in the survey suggested that inclusion of an asynchronous feedback system would benefit them.

Objective 3. *Provide a student dashboard for completion of the requirements process.*

Students were provided with a dashboard screen in which they could visualise their completion of the requirements process. Use of colour to indicate complete or incomplete requirements seemed to work well in the class room. In addition, students were able to continually refine requirements and track different versions of the requirements specification. This was only possible by incorporation of eWURF into a CAWE tool, where usage data was generated each time the student interacted with the tool.

Overall Reflection on the Problem Solving Interest

In addition to the issues highlighted in the survey and usage data, there were also a number of areas which required further discussion from the perspective of the researcher and facilitator. Having observed the student using the CAWE tool, there were certain aspects that worked well and there were areas that needed to be improved in future work:

1. The Web Actor definition was one aspect of eWURF that was understood most fully by the students and required little intervention. To this end, eWURF helped the student to appreciate that their website could be used by a multitude of users, rather than one. In the first method, the meta-model constrained the student in their user definition, adopting a single primary and secondary user in which to map requirements to. In eWURF, the user was defined as a 'Web Actors' and the student was not constrained in the number of Web Actors that could be generated or in their profile definition. Students were able to create subsets of Web Actors linked to specific Tasks and Functional Requirements. Future work could look at grouping Web Actors, where these share common Tasks or Goals and represent these visually within the CAWE tool and SRS document.

2. Tasks helped the student to understand how the user would interact with the website. The majority of students managed to understand what the tasks involved conceptually and how these could be refined throughout the project period. Further work should include how to implicitly link a Web Actor Task to a Functional Requirement, to enhance forwards and backwards traceability. A method of enforcing associations between Web Actors, Functional Requirements and Web Actors Tasks should also be investigated further.

3. Functional Requirements were the least understood aspect of the eWURF tool, reflected in what was observed within the laboratory. Interventions were needed in order to help the student understand how they should be written. It was observed that students from a technical background understood these better than students with a design background. More work is required in the support mechanisms in regard to this, including whether a fixed lexicon or dictionary should be introduced to standardise the language used for functional requirements.

4. The Actor Association provided a way to link Web Actors to specific Tasks and Functional Requirements and this proved to be a useful way of helping the student to understand the traceability aspect of Requirements Engineering. Further work is needed to enhance the association model, especially concerning Web Actor Tasks to Functional Requirements and a mechanism to automate the evaluation of the traceability through the actual website implementation.

5. Usability 'non-functional requirements' were not well understood by the student and is a weakness in the non-functional requirements. The language used within the usability requirements could have been the reason for this. As a consequence, the traceability of usability requirements was poor in the student assessments. More work is required within the learning model in order to address this aspect.

6. User Interface and Technical requirements proved to be most useful to students and this was evident in traceability within the student assessments. For example, the screen resolution and navigation type could be traced through to their website, as well as the choice for the dynamic development environment. Further work could address ways of enhancing this by reflecting the myriad of devices that can now consume web content.

7. It was observed that the student support and guidance mechanism within the CAWE tool were well used by the student. Some student's indicated that they found it most useful where examples were included and some felt that more could have been included, such as video based tutorials within each screen. More work is required on the asynchronous feedback potential of the CAWE tool, including the possibility of sending SRS documents to the tutor for comments and annotation.

8. Perhaps the most successful aspect of eWURF was the requirements specification document which included version control, a summary of students' requirements and an identification system for each functional requirement. Students were able to link to the requirements specification from their design documentation, providing evidence for their design choices. Students resisted the temptation of moving into the implementation phase straightaway, a practice observed and outlined in the problem solving interest in the first cycle of research and addressed within each of the interventions proposed.

9. Having observed the students using eWURF and assessed their submissions, the least successful aspect of the framework was the validity of requirements. Validity was only carried out by the assessor, after the requirements specification and website was submitted to be marked. This was too late for the student to rectify their requirements or indeed their website and is something that requires further work.

Reflection on the Research Interest

The purpose of this section is to reflect upon the third cycle of research with regard to canonical action research and the methods employed to evaluate both the method and learning aspects. Within the final research cycle, a number of qualitative and quantitative methods were used to collect data about how the CAWE tool was being used by the student in order to address the research objectives. These included An Opinion Survey; An Observational Study within the laboratory; A Focus Group and Comparative Analysis of the Usage and Assessment data.

Usage and Assessment data was collected and stored to enable further analysis, such as *average usage*, *how many times the student created an requirements specification document* and *relationship between usage and assessment marks*. Although this provided a useful insight, it was recognised that reliability of the results could be limited. For example, students who used the CAWE tool may have engaged more with the theoretical aspects and received a higher mark whichever RE technique was adopted by them.

Analysis of the survey and usage data in section 7.2.1 provided evidence for the evaluation of eWURF. The reliability of the data from the survey and usage logs was something that proved difficult to establish due to relatively small sample size. Interpretation of the results provided the basis for generalisations to be made, rather than concrete evidence of the way eWURF was being used by the student. Having a mixed approach to the research methods employed in this final cycle had a positive impact. In particular, the interpretation of survey data pointed to issues in the way the CAWE tool was being used. This led onto to the formation of questions for the focus group, which in turn allowed a number of themes to emerge which were followed up in the analysis of the usage data.

Three research objectives were established within the third cycle of research. These are now reviewed in the context of the research interest within the action research approach:

1. *To investigate how a computer aided web environment (CAWE) tool can support the inexperienced student user in their requirements elicitation, analysis and specification using a natural language.*

   The third research cycle has shown that the CAWE tool provided a mechanism to allow the student to elicit, analyse and specify their web user requirements throughout the web project lifecycle. It allows the student to control the flow of their ideas, from initial client and user expectations, through the refinement of functional requirements using a natural language. Further work is required to better understand how to enhance the traceability of requirements, especially the association model and evaluation of requirements in the website implementation.

2. *To investigate how a consistency, completeness* and *correctness rules model can be incorporated into the CAWE tool.*

   Reflection of the problem solving interest raised a number of issues with the correctness aspect within the association model, as well as the positive impact it had on the students' learning and resulting requirements documentation. The rules model had constrained the student to complete their documentation, although the correctness of the requirements requires further work. For example, it was still possible to have a complete set of Functional Requirements and Web Actors defined, but no association between them in the documentation. Further work could also determine how some flexibility could be introduced concerning the non-functional requirements, where these are not editable by the student.

3. *To determine if usage of the CAWE tool influenced assessment outcomes for the student.*

   Analysis of the usage and assessment data demonstrated that there was some weak correlation between the usage and assessment outcomes. It was argued that there were limitations with this interpretation, concerning the direct relationship between this particular intervention and the student assessment score. It could be argued that an alternative requirements analysis tool would have led to the same statistical outcome. From an observational perspective and assessor of the student work, it was found that the students analysis was much improved, although there will always be a variance in assessment scores concerning this aspect. It was also difficult or indeed impossible to measure student learning through this type of statistical analysis.

Reflection on the third cycle as a researcher

Limitations of the results have already been mentioned in the previous section, with the greatest of these centred on the correlation of assessment to usage data. Whilst this was true, learning how to undertake this type of statistical analysis has been worthwhile. Working with multiple data sets including usage and assessment data provided an opportunity to engage with quantitative data analysis, which was something that was not achieved in the first and second cycles. Recognising where the results have limitations has also proved to be useful and transferable to other research projects.

An aspect of the research that proved to be the most useful to the evaluation of the third cycle, but at the same time challenging to undertake, was the focus group. It is felt that by linking the focus group to issues highlighted in the student survey, responses from the students were enhanced. Conversely, recording and transcribing the audio recorded during the focus group proved to be time consuming.

The main lesson learned from this cycle was that the sheer amount of work needed to conduct a valid evaluation involving multiple data collection and analysis methods cannot be under estimated.

7.3 Conclusions

The third cycle of research focused on further refinement of the meta-model and transformation of the meta-model into a CAWE tool. A review of similar work concerning CASE tools found that some research had been carried out with reference to CASE tools within the Web Engineering domain and that these were referred to as CAWE tools. Transformation of the paper based method into a more fully fledged CAWE tool could provide solutions to problems highlighted in the evaluation of the second research cycle. This included a way of enforcing a rules model to enable the student to submit *consistent, complete* and *correct* requirements and to enhance the students support and guidance during completion of the process and a method of storing user data.

Development of the CAWE tool was achieved, along with its adoption on two modules, supported the hybrid PBL model. This thesis has argued that in order to positively change the established practice of students not undertaking requirements analysis in web projects, an intervention is needed in their learning. The third research cycle demonstrated how a number of models can combine together to better support the inexperienced student user in their web user requirement analysis. A hybrid PBL model has played an important role in the intervention as a whole and in itself has influenced the outcomes of this research programme. The theoretical stage within the hybrid PBL model was important, due to the fact that Requirements Engineering required some understanding and recognition of the process before it could be attempted. Without this, it could be argued that the student would have had a very different experience and a different set of findings would have been reported. Further work regarding the learning model is needed and it is recognised that evaluation of this part of the intervention is weaker than that of the meta-model or CAWE tool. It would be beneficial to understand how to better represent variance in learning styles within the PBL model, for example.

The researcher was an active participant in the research programme and a teacher on both modules adopted to test the intervention. It could be argued that this has influenced the findings of the research and interpretation of the findings must take into consideration Canonical Action Research as the research method. The dual cycles of problem solving and research interest has played an important role, particularly in the way that it has allowed in depth and extended observation of the intervention in use. It also allowed the intervention to be integrated with the assessment strategy for both modules, including the briefing document and marking criteria, which again has influenced the findings of the research.

The impact on assessment was measured through the collection of usage data within the CAWE tool. It was found that there was a weak correlation between the usage of the tool and the assessment score by individual students. A weakness in this regard was identified, concerning the safety of the hypothesis in relation to the way the CAWE tool itself influenced the student. The hybrid PBL model also played an important part in changing the established practices of the students on these two modules. To conclude this chapter, the third cycle fulfils the objectives set out in the second research cycle.

The next Chapter draws conclusions from the main body of work and outlines implications concerning the major and minor contributions to knowledge. The initial research aims are used as a means of concluding the main findings of this research programme. A reflection on how the author conducted the research programme is also provided. As discussed in this Chapter, there are potential areas of development in continuing this research and these are discussed in terms of future work in the next Chapter.

Chapter 8 Conclusions, Contribution and Future Work

8.1 Introduction

This chapter concludes the research programme by reviewing the research aims and presents the main contributions that this thesis has made. The chapter concludes with a discussion about future work.

Preliminary work was undertaken in Chapter 2 to better understand the background to Software, Requirements and Web Engineering. A review of related work was undertaken in Appendix A3 which contributed to each intervention in terms of their construct and how to support the inexperienced student user in Chapter 3. Canonical Action Research was chosen as the research method due to the nature of the study and was examined in Chapter 4. Action research suited the educational dimension in the way its participants (students contributing directly to the research) and the way in which it became involved in shaping the intervention over multiple cycles of research. Chapter 5, 6 and 7 describe the activities taken in the planning, design and implementation of the three interventions. Each chapter also discusses the evaluation of each intervention through interpretation of questionnaire responses, assessment and observational data. By comparing these data sets and reflecting on their significance in the context of the research objectives it was possible to modify the intervention in the next research cycle. Chapter 8 concludes by reflecting on the research aims, outlining the main contributions to knowledge and a discussion about future work.

8.2 Reflection on Research Aims

8.2.1 Examine existing Requirements Engineering methods and techniques within Web and Software Engineering.

A review and analysis of Requirements Engineering found a number of useful approaches, but none specifically addressing the learning and teaching aspect. It was found that some approaches were incompatible with Web Engineering such as: ARM, US, and AMSF. Five methods: NDT, AWARE, URN, SOARE and SSM/ICDT were aimed at projects that involved Web Engineering. These were each characterised by a modelling technique that enabled the web developer to draw out requirements based on *business vision*, *objectives*, *user tasks* or *goals* and recommended both functional and non-functional requirements within their taxonomy.

Initial work focused on user profiling as a mechanism to discover and document web user requirements. The approach named Rapid User Modelling Method (RUMM) was based on the work of Sommerville & Sawyer 1997 and Berry 2003, who defined a Requirements Engineering process. RUMM was also influenced by the work of Bolchini and Paolini, 2003 and their proposal of an approach encouraging web developers to discover requirements by profiling the user.

RUMM focused on encouraging the student to create a user profile through profiling techniques as a *starting point* within the meta-model. Once this had been completed by the student, it was thought that they would have a better understanding of web design requirements, such as navigation, content, colour and layout. Participants in the first research cycle found the entry point into the meta-model confusing, possibly due to insufficient information been available to them. The basis for web design decisions taken by the student could therefore be said to be unsound, as the student should not rely on guesswork, but undertake systematic analysis based on visible evidence.

8.2.2 Facilitate the production of a novel method and prototype framework to aid the inexperienced student user to undertake elicitation, analysis and specification of web user requirements.

The first cycle of research established that the student was prepared to follow a requirements process and document this within their ICA submissions. It was evident from the ICA submissions that more analysis was undertaken by student than before and this aspect was a successful outcome. It also provided the opportunity for further work in improving the method, particularly with regard to extending the meta-model to include a more appropriate entry point into the process and to reflect dynamic web development.

Based on the learning and reflection achieved in the first cycle of research, the direction of the research in the second cycle focused on ensuring that the restructured meta-model better supported the student in a number of ways. This included changing the meta-model to better reflect both design and dynamic web development. Sommerville's notion of functional and non-functional requirements provided the basis for the updated meta-model. A number of established requirements approaches such as Object View And Interaction Design (OVID) (see Appendix A3, section 3.5.1), Usability Context Analysis (UCA) (see Appendix A3, section 3.4.2) and Navigational Development Techniques (NDT) (see Appendix A3, 3.5.6), were found to use the same taxonomy and adoption of the meta-model was therefore considered to be an appropriate way forward.

In the evaluation of the second research cycle, it was found that the student was able to relate to and better comprehend functional and non-functional requirements, especially as this provided a way for adding unlimited dynamic web requirements, whilst constraining design web requirements within non-functional requirements.

Further work on user profiling resulted in a modified meta-model that reflected the requirement for a more flexible user profile definition. In the first method, the user profile was limited to two types of users *primary* and *secondary*. Participants in the research indicated that this was too restrictive. An alternative approach was found that led to changes in the way a user was defined and to reflect this the label for the user was changed to 'Web Actor'. The Web Actor could either be human or system. The latter could be, for example, a web service or other external API that interacted with the website in some way. Two constructs were offered to help the student to define a Web Actor and its profile. Bolchini and Eric, 2004 provided an approach to express varying levels of importance within the web application, using a priority attribute as part of the construct. The student could then determine Web Actor importance within the application in order to resolve conflicts that arose during analysis.

Linking closely to the Web Actor construct, the *Task* construct provided a mechanism for the student to think about the interaction that the Actor would perform within the web application and to enable the student to move onto the next stage of the process by forming Functional Requirements. A number of students indicated that they had problems understanding the term *Task*, in particular the difference between a *Task* and a *Functional Requirement*. However, these issues were easily resolved by providing some guidance to the student.

Two elements were deemed to be the most success aspects of the modified meta-model: *the separation of Tasks from the User Profile* and *the production of Web Actors and Functional Requirements*. In addition, work was also undertaken to improve student support and guidance within the overall intervention. To this end, supplementary notes were written to help students comprehend key stages within the process. To support the student further, the VLE was used to disseminate the method as electronic fill in forms and guidance notes. Additionally, downloadable examples of the WURF in use were provided via the VLE.

During the second cycle of research, it was identified that the traceability of requirements within the website implementation was very important in terms of the validity of the approach. To support traceability, an association model was proposed that enabled Tasks and Functional Requirements to be linked visually. The resulting requirements could then be traced forwards and backwards through to the website. In the evaluation, it was determined that this was unsuccessful due to the way the student had to manually ‘draw’ the association and that there was no way that this could be enforced. It was recognised that traceability was still an important aspect of the web requirements method and an alternative way of modelling the association was sought.

One of the main outcomes of the reflection on the problem solving interest in the second cycle of research was the need to further enhance the support and guidance model. In order to address deficiencies in the meta-model in the second cycle of research, further work in the third cycle of research would focus on enhancing the association model, consistency, completeness and correctness of the requirements and by providing more support and guidance electronically. Transforming WURF into a CAWE tool was acknowledged as a method of achieving this, as well as providing additional student support capability. Being able to provide the student with a ‘dashboard’ within the CAWE tool and being able to provide support and guidance in real time were also cited as possible solutions to the problems identified in the evaluation. In particular the consistency, completeness and correctness model could be enforced more rigorously within the CAWE tool.

8.2.3 Specify an intervention and framework that comprises a process meta-model, object model, rules model, support and guidance model, consistency, completeness and correctness model, learning model, student data model and a requirements specification model that could be represented in an automated Computer Aided Web Environment (CAWE) tool.
The third and final research cycle demonstrated how an updated and refined meta-model could be transformed into a CAWE tool to better support the inexperienced student user in their web user requirements analysis. The contribution was made by understanding how the CAWE tool influences the student, their Web Engineering process, usage of the CAWE tool and the modifications to established teaching and learning practices.

A rules model was embedded within the CAWE tool to enforce a number of constraints in relationship to the meta-model. This included the ability to automatically generate the requirements specification document, but only if certain conditions had been met, such as the completeness of the requirements. The way in which the rules model was visually represented within the student dashboard was a significant outcome and contribution of the research programme.

An additional contribution was made by understanding how to visually represent the student usage data within the CAWE tool dashboard. Usage data was generated during each student initiated event within the CAWE tool and was stored for automated analysis within the CAWE tool itself. For example, the rules model was able to consume usage regarding the completeness of the requirements. The requirements specification document was also able to consume usage data to record version history.

Analysis of the student opinion survey in the third research cycle showed how the majority of students felt that their web development process had been influenced through the use of the CAWE tool. The survey also demonstrated that the majority of students had accessed the help and guidance system, although there were concerns about the non-functional requirements aspect. User feedback also indicated that there were areas within the student dashboard that need to be refined.

During the focus group, a number of themes emerged that reinforced the findings from the student opinion survey, for example, that the CAWE tool influenced the web project as a whole not just the implementation. It also raised some concerns. For example, a number of students had adopted the CAWE tool as a method to control the whole web development lifecycle, rather than using it to facilitate their requirements elicitation, analysis and specification.

Perhaps the most important outcome of the focus group was that participants felt that the CAWE tool lacked a means of receiving feedback from the tutor. Although the help system provided some guidance on its completion, participants felt that feedback from the tutor on the requirement specification document would be beneficial to their learning.

It was established that the PBL model influenced the usage of the CAWE tool and that the intervention as a whole had to include multiple learning and teaching strategies and the ability to respond to student needs as they arose. It was felt that the students needed some theoretical underpinning before they could commence effective requirements analysis. It is recognised that further work regarding the learning model is needed and it is also accepted that evaluation of this part of the intervention is weaker than that of the meta-model or CAWE tool. In particular, it would be beneficial to understand how to better represent variance in learning styles within the PBL model.

8.3 Contributions to Knowledge

The main contributions of this research are centred on the need to change practice regarding how students undertake requirements analysis in their web projects. Through undertaking this study the thesis has provided the following main contributions:

8.3.1 Extension of knowledge and understanding of User Requirements in Web Engineering. This is evidenced by the review and analysis of Requirements Engineering methods and techniques within Web and Software Engineering. A review map demonstrates how a range of methods and techniques for Web User Requirements Engineering can be adopted for web projects. The study also established how the intervention as a whole modified the students' practice regarding their approach to requirements elicitation, analysis and specification in their web projects.

8.3.2 Identification of gaps in knowledge and understanding regarding the lack of analysis techniques used by the student. Through identifying problems with current teaching practices and making interventions to those practices the thesis provides a way forward for more effective teaching of Web Engineering. A way of achieving this was proposed, tested and refined over three cycles of research leading to an evidenced based approach that informs the curriculum of Web Engineering at HE level.

8.3.3 A range of novel methods and frameworks developed through student collaboration that can be adopted for teaching purposes in Web and Requirements Engineering modules. Three interventions to practice were deployed over three research cycles including a novel method (RUMM) to aid the inexperienced student user to undertake user profiling, elicitation, analysis and to document web user design requirements. This was achieved through a collaborative endeavour with the students and has greatly influenced this project. A more fully developed framework (WURF) to aid the inexperienced student user to elicit, analyse and document web user development requirements, has been achieved. A proposal for a Framework (eWURF) that comprises a Process meta-model, Object model, Rules model, Guidance model, Consistency, Completeness and Correctness model, Student Data model, Learning model and a Requirements Specification model, that could be represented in an automated Computer Aided Web Environment (CAWE) tool, has been achieved.

In addition a number of minor contributions emerged:

8.3.4 How to implement the final Framework (WURF) within a CAWE tool to support the student user.

8.3.5 How to collect and analyse log data produced by the students, including its integration and visualisation within the CAWE tool for both student and tutor use.

8.3.6 A way of visualising the process meta-model in a web user requirements CAWE tool through an automated rules model.

8.3.7 A hybrid PBL model for Requirements Engineering in Web Engineering that aligns theory with practice, including the role of formative and summative assessment and student support.

8.3.8 An evaluation of an action research methodology applied to a computer science based research programme.

8.4 Future Work

A number of ideas emerged within the reflection of the research interest regarding future work. This section describes how this research could be extended, including its limitations and the areas that warrant further investigation.

Student and tutor support - Further work is required on enhancing student support mechanisms within the CAWE tool, especially relating to asynchronous feedback. At present summative feedback is provided at the end of the assessment process. Further work could look at how formative feedback affects the usage of the CAWE tool, including the possibility of sending documents to the tutor for comments and annotation for formative feedback purposes.

More work is required concerning the tutor and their specific requirements as educators, including investigating the role of the tutor and using user data as means of tracking student progression. For example, further work could look at how the data could be represented within a 'tutor dashboard' that could act as an early warning system for the student who has not engaged with their requirements analysis.

Evaluating the CAWE tool on other modules and institutions - The CAWE tool could be evaluated on other courses in order to test the robustness of the intervention as a whole and as a standalone teaching tool. If the CAWE tool is to be adopted by other institutions further work is also required to ensure that it matches their learning and teaching approaches. For example, the intervention relies on a hybrid PBL model which may not suit learning and teaching methods employed at other institutions. At present, the meta-model is focused on Web Engineering. Further work could investigate how the meta-model could be adapted to support the Software Engineering student, who could use the CAWE tool to elicit, analyse and specify their Software Requirements.

Address problems with the meta-model - There are also some deficiencies and limitations within existing meta-models that require further work, with the association model in particular requiring further attention. Being able to associate Tasks to Functional Requirements is something that is recognised as deficient at present, affecting the traceability and validity of the requirements. Further work would need to be carried out on how to measure traceability and how to represent this within the student dashboard.

Web actor and task clustering - Web Actors are modelled individually at present, but it is recognised that some actors would share common tasks. Being able to cluster actors together is something that warrants further investigation. For example, by clustering actors and their tasks, a greater understanding of their requirements may emerge through the analysis stage, especially where tasks may conflict with one another. Specific requirements may also emerge through the cohesion between actors and tasks within the cluster, which is something that is missing in the current meta-model. Requirements may therefore be written with greater *consistency*, *completeness* and *correctness*, further enhancing the validity of the approach.

Enhancing non-functional requirements - Greater flexibility in defining non-functional requirements is something that the participants in the survey highlighted as being important to them. At present, the non-functional requirements are limited to User Interface; Usability; Accessibility; Marketing and Technical requirements. Further work could investigate how the student could propose their own non-functional requirement construct. The work would need to establish how this could be achieved, whilst adhering to the *consistency*, *completeness* and *correctness* of requirements.

Usability ‘non-functional requirements’ were not well understood by the student and this is a weakness in the non-functional requirements construct. The language used within the usability requirements could be the reason for this. As a consequence, the traceability of usability requirements was poor in the student assessments. More work is required within the learning model in order to address this aspect.

User Interface and Technical Requirements proved to be most useful to the student according to the feedback from the focus group and as evidenced within the student assessments. For example, the screen resolution and navigation type could be traced through to their website, as well as the choice for dynamic development environment. Further work could look at ways of enhancing this by reflecting the myriad of devices that can now consume web content.

Sharing User Data - Agile Web Development Methodologies reflect the iterative nature of development, where prototypes are built incrementally and where requirements are discovered through this process. Code reuse is encouraged, where code may be reused from a library of previously tested and validated classes. Further work could look at reusing previously generated requirements, actor profiles and tasks. The work would need to investigate the safety of reusing and sharing user data and how to extend the meta-model to represent the shared objects. Any investigation would also need to look at the role of sharing user data and its impact on the learning model.

Requirements Validation - Having observed the student using the CAWE tool in the lab and assessed in their projects, the least successful aspect of the meta-model is the ‘validity of requirements’. Validity is only carried out by the assessor, after the requirements specification and website is submitted to be marked. In essence, there is a missing link within the meta-model, between the requirements and the actual website. The student should be able to test requirements validity before project completion and any further work would need to look at how this could be achieved in a modified meta-model. Mechanisms in which the student could be supported in this process would also need to be established.

8.5 Closing Remarks

Main Lessons Learned as a Researcher

Reflecting on the research programme from the perspective of a researcher, it must be said that the direction of the study changed quite significantly over the first two years. Key to this change were ideas developed through undertaking the background research and reflecting on the outcome of other studies. Changes in direction also came about through collaborations with the students involved in the research by taking their feedback into consideration in the way that the interventions were refined over three research cycles. Feedback from the two conferences attended also led to changes in direction, highlighting the benefit of discussing work in progress with peers.

Feedback and refection were two important aspects integrated into the research method adopted for this study. Canonical Action Research was the most appropriate, given the educational context in which it was set and the need to reflect and allow other people to contribute to the interventions. The dual cycles of problem solving and research interests has also played an important role, particularly in the way that it has allowed in depth and extended observation of the intervention in use. The main benefits of Action Research to the programme were that:

- It provided a structure in which to define the research objectives and solutions over multiple iterations.

- Students as contributors to the study and shaping the intervention in their learning.
- Increasing the momentum of each research cycle due to the continual identification of problems, planning, action taking and evaluation.
- The transformative nature of the research matches both the action research methodology adopted and the educational environment in which it took place.
- On-going laboratory observation allowed the researcher to form a deep insight of how the intervention was being used by the participants.
- Cycles of continuous reflection, evaluation and modifications to the intervention benefitted the education of the participants themselves.
- It allowed ideas to emerge over a period of time by on-going reflection and learning as part of the dual cycle Canonical Action Research methodology.
- Opportunities for early evaluation from peers. For example, the research undertaken in the first and second cycle were disseminated via a conference paper and presentations at SIGSAND and the UKAIS PhD consortium.

Some issues also became apparent in its use in this programme, namely:

- That the role of the researcher influences the findings, especially the active participation within student learning.
- The transferability of the findings may be limited to the institution, programmes and cohorts.
- The problem solving interest was initiated by the researcher from an educational perspective. The literature review then dictated a response to the problem. On reflection, perhaps more perspectives on the problem should have been sought, such as educators and practitioners in the field.

This research programme was undertaken on a part-time basis over 6 years, which brought with it a unique set of challenges that would not have affected full time research. Changes in Web Engineering practice and promotions in the school for the researcher affected the programme and influenced the final outcome in a way that was not envisaged at the outset. On reflection, perhaps more research cycles would have led to a further enhanced intervention and different set of findings and contributions to knowledge. However, it must be emphasised that the researcher has confidence in the final intervention and this is reinforced by its adoption in a number of programmes in the school, including franchised partners in Botswana and Sri Lanka.

In Conclusion

This research programme makes a number of contributions to the discipline of Web Engineering and Requirements Engineering. In particular, the research has made a fundamental contribution to the teaching of Web Engineering by identifying gaps in knowledge and addressing the lack of analysis techniques used by the student. The intervention as a whole has changed student practice with regard to their requirements analysis. The result of this research, specifically the teaching and learning aspect, may be useful to other institutions, where module programme learning outcomes could take into account the contribution of this research in curriculum and module design.

Significant contribution has been made by undertaking a review and analysis of RE methods and techniques found within Software and Web Engineering and by proposing, implementing and evaluating a set of new methods and an overall framework for web user requirements. These methods and frameworks are supported by a number of models, which in turn, contributed to an overall intervention to student practice.

Further contribution has been made by understanding how to embed the framework within a CAWE tool and in particular how this can provide enhanced student support and guidance. Usage data, generated by the rules model and the CAWE tool, provided additional contribution, especially in the visualisation and data mining areas. Evaluation of the third research cycle unveiled areas for further work, demonstrating that this research can be extended and continue to provide a contribution.

References

(Ambler, 2002) Ambler, S.;, "Agile Modelling: Effective Practices for Extreme Programming and the Unified Process", John Wiley & Sons, pp.246, 2002.

(Ambler, 2004) Ambler, S.; W.; , "The Object Primer: Agile Model-Driven Development with UML 2.0", Cambridge University Press; Edition 3, 22 Mar 2004.

(Anderson, 2005) Anderson, D.; J.; , "Stretching Agile to fit CMMI Level 3 - the story of creating MSF for CMMI Process Improvement at Microsoft Corporation", AGILE 2005 Conference, IEEE Computer Society, pp.193-201, July 2005.

(Armarego, 2007) Armarego, J.; , "Educating Requirements Engineers in Australia: effective learning for professional practice", phd thesis, School of Computer and Information Science, University of South Australia, 2007.

(Arnold, et al., 2010) Arnold, D.; Corriveau, J.; P.; Wei Shi.; , "Scenario-Based Validation: Beyond the User Requirements Notation," Software Engineering Conference (ASWEC), 2010 21st Australian , pp.75-84, 6-9 April 2010.

(Asarnusch, et al., 2006) Asarnusch, R.; Meder, D.; Wiesenberger, J.; Astrid, B.; , "Visual Requirement Specification In End-User Participation", First International Workshop on Multimedia Requirements Engineering (MERE'06 - RE'06 Workshop) pp.6, 12 Sept 2006.

(August, 1991) August, J.; H.;, "Joint Application Design The Group Session Approach to System Design". Yourdon Press, 1991.

(Avison and Fitzgerald, 2003) Avison, D.; and Fitzgerald, G.;, "Where Now for Development Methodologies? ", Communications of the ACM 46 (1) : pp.79-82, 2003.

(Avison and Fitzgerald, 2006) Avison, D.; and Fitzgerald, G.;, "Information Systems Development: Methodologies, Techniques and Tools", McGraw-Hill Higher Education, pp.135, 2006.

(Babanezhad, et al., 2010) Babanezhad, R.; Bibalan, Y.M.; Ramsin, R.; , "Process Patterns for Web Engineering", Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual, pp.477-486, 19-23 July 2010.

(Barry, et al., 2001) Barry, C.; Lang, M.;, "A Survey of Multimedia and Web Development Techniques and Methodology Usage," IEEE Multimedia, pp.52-56, Apr-June 2001.

(Baskerville and Wood-Harper, 1996) Baskerville, R. L.; and Wood-Harper, A.; T.;, "A critical perspective on action research as a method for information systems research", Journal of Information Technology, vol.11, pp.235-246, 1996.

(Baskerville and Wood-Harper, 1998) Baskerville, R.;L.; and Wood-Harper, A.;T.; , "Diversity in information systems action research methods", European Journal of Information Systems, vol.7, no.2, pp.90-107, June 1998.

(Baskerville, 1999) Baskerville, R.; L.;, "Investigating Information Systems with Action Research", Communications of the Association for Information Systems, vol.2, no.19, October 1999.

(Baudry, et al., 2007) Baudry, B.; Nebut, C.; Le Traon, Y.; , "Model-Driven Engineering for Requirements Analysis", Enterprise Distributed Object

Computing Conference, 2007. EDOC 2007. 11th IEEE International, pp.459-459, 15-19 Oct 2007.

(Berenbach and Gall, 2006) Berenbach, B.; Gall, M.; , "Toward a Unified Model for Requirements Engineering," Global Software Engineering, 2006. ICGSE '06. International Conference on, pp.237-238, Oct 2006.

(Berry, et al., 2003) Berry, Hungate and Temple, IBM Systems Journal, Vol 42, No 4, 2003.

(Berry, et al., 2007) Berry, D.; Isensee, S.; Mullaly, J.; Roberts, D.; , "OVID: an overview", http://www-03.ibm.com/easy/page/104, last accessed 09/09/07.

(Bevan, 1997) Bevan, N.;, "Usability Context Analysis: a Practical Guide", NPL Usability Services, Teddington, UK, 1997.

(Biggs, 2003) Biggs, J.;, "Teaching for Quality Learning at University", The Society for Research into Higher Education, 2003.

(Bleistein, et al., 2004) Bleistein, S.J.; Aurum, A.; Cox, K.; Ray, P.K.;, "Strategy-oriented alignment in Requirements Engineering: linking business strategy to requirements of e-business systems using the SOARE approach", Journal of Research and Practice in Information Technology, vol.36, no.4, pp.259-276, 2004.

(Boehm and In, 1996) Boehm, B.; In, H.; , "Identifying quality-requirement conflicts," Requirements Engineering, 1996., Proceedings of the Second International Conference on, vol.13, no.2, pp.218, 15-18 Apr, 1996

(Boehm, 1984) Boehm, B.; W.;, "Verifying and validating software requirements and design specifications", IEEE Software 1, pp.75-88, 1984.

(Boehm, 1986) Boehm, B.; w.;, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes, ACM,vol. 11, no.4, pp.14-24, August 1986.

(Bolchini and Eric, 2004) Bolchini, D.; Eric, S.;, "Modelling User Requirements For Web Application Design", Studies in Communication Sciences, Università della Svizzera Italiana, pp.175, 2004.

(Bolchini and Paolini, 2003) Bolchini, D.; and Paolini, P.;, "Goal-Driven Requirements Analysis for Hypermedia-intensive Web Applications", Requirements Engineering Journal, Special Issue RE03, 2003.

(Bolchini, et al., 2003) Bolchini, D.; Paolini, P.;, Randazzo, G.;, "Adding Hypermedia Requirements to Goal-Driven Analysis", 11th IEEE International Requirements Engineering Conference (RE'03), pp.127, 2003.

(Boud and Feletti, 1998) Boud, D.; and Feletti, G.;, "The Challenge Of Problem Based Learning", Kogan Page, 1998.

(Brown and Knight, 1994) Brown, S.; and Knight, P.;, "Assessing Learners in Higher Education", London: Kogan Page, pp.12, 1994.

(Carr and Kemmis, 1986) Carr, W.; and Kemmis, S.;, "Becoming Critical: Education, Knowledge and Action Research", Falmer, London, 1986.

(Carroll, 2002) Carroll, J.; M.;, "Scenarios and design cognition" Proceedings IEEE Joint International Conference on Requirements Engineering (RE02), pp. 3- 5, 2002.

(Casteleyn, et al., 2009) Casteleyn, S.; Daniel, F.; Dolog, P.; Matera, M.;, "Engineering Web Applications", Springer, 2009.

(Casteleyn, et al., 2009) Casteleyn, S.; Daniel, F.; Dolog, P.; Matera, M.;, "Engineering Web Applications (Data-Centric Systems and Applications)", pp.349, 2009.

(Cato, 2001) Cato, J.;, "User-Centered Web Design", London: Addison-Wesley, 2001.

(Ceri, et al., 2000) Ceri, S.; Fraternali, P.; and Bongio, A.;, "Web modelling language (WebML): A modelling language for designing Web sites". Proceedings of 9th International World Wide Web Conference, Amsterdam, 2000.

(Chandler and Torbert, 2003) Chandler, D.; and Torbert, B.;, "Transforming Inquiry and Action: Interweaving 27 Flavors of Action Research", Action Research, vol.1 no.2, pp.133-152, 2003.

(Checkland, 1991) Checkland, P.;, "From framework through experience to learning: the essential nature of action research", Information Systems Research: Contemporary Approaches and Emergent Traditions, North-Holland, Amsterdam, pp.397-403, 1991.

(Chen and Heath, 2005) Chen, J.; Q.; and Heath, R.; D.;, "Web Application Development Methodologies," In Web-Engineering: Principles and Techniques, W.Suh, Ed., Idea Group, , pp.76-96, 2005.

(Chrissis, et al., 2003) Chrissis, M.; B.; Konrad, M.; Shrum, S.;, "CMMI: guidelines for process integration and product improvement", Addison Wesley, pp.485, 2003.

(Cobb, 1999) Cobb, T.;, "Applying Constructivism: A test for the learner as scientist", Educational Technology Research & Development, Vol.47, No.3, pp.15-31, 1999.

(Cockburn, 2001) Cockburn, A.;, "Writing Effective Use Cases", Addison-Wesley Professional, pp.24, 2001.

(Cockburn, 2001) Cockburn, A.;, "Writing Effective Use Cases", The Crystal collection for software professionals, The Agile software development series, Addison-Wesley, 2001.

(Cohen, et al., 2007) Cohen, L.; Manion, L.; Morrison, K.;, "Research Methods in Education", Taylor & Francies Group, pp.28, 2007.

(Conallen, 2003) Conallen, J.;, "Building Web Applications With UML", Addison-Wesley, pp.3, 2003.

(Connor, et al., 2009) Connor, A.;M.; Buchan, J.; Petrova, K.; , "Bridging the Research-Practice Gap in Requirements Engineering through Effective Teaching and Peer Learning," Information Technology: New Generations, ITNG '09, Sixth International Conference, pp.678-683, 27-29 April 2009.

(Cornish, et al., 2003) Cornish, R.; Moore, T.; Pavoni, D.; Rockenbach, E.;, "NET Solution Architectures: Gathering and Analyzing User Requirements", Jul 4 2003.

(Davison , et al., 2004) Davison, R.; Martinsons, M.; G.; Kock, N.;, "Principles of canonical action research", Information Systems Journal, Vol. 14, No.1, pp.65-86, 2004.

(De Troyer, et al., 2007) De Troyer, O.; Casteleyn, S.; Plessers, P.;, "WSDM: Web Semantics Design Method", Web Engineering: Modelling and Implementing Web Applications, Human-Computer Interaction Series Vol. 12, pp. 303-352, 2007.

(December, 1996) December, J.;, "An Information Development Methodology for the World Wide Web" Technical Communication, Vol. 43, No. 4, pp. 369-375, 1996.

(December, 2008) December, J.;, Home Page, <http://www.december.com/web/develop/>. last accessed 5/08/08.

(DET , 2003) DET, Blended Learning, NSW Department of Education and Training, 2003.

(Dubravk, 2007) Dubravka, C.; K.;, "Critical Research in Information Systems: The Question of Methodology", In Proceedings of the Fifteenth European Conference on Information Systems, pp.1446-1457, 2007.

(Escalona And Aragón 2008) Escalona, M.; J.; Aragón, G.;, "NDT. A Model-Driven Approach for Web Requirements", IEEE Transactions On Software Engineering, Vol. 34, No. 3, pp.377-390, May/June 2008.

(Escalona and Koch, 2004) Escalona, M.; J.; Koch, N.;,"Requirements Engineering for Web Applications: A Comparative Study", Journal of Web Engineering, vol .2, no. 3, pp193-212, 2004.

(Fairclough, 1995) Fairclough, N.; "Critical Discourse Analysis: The Critical Study of Language", Longman, London, New York, 1995.

(Frideres, 1992) Frideres, J.; S.;, "Participatory research: An illusionary perspective, A world of communities: Participatory research perspectives", North York, Ontario: Captus University Publications, 1992.

(Gaffney, 2007) Gaffney, G.;, "Information and Design", http://www.infodesign.com.au/usabilityresources/analysis/contextualenquiry.asp, Last Accessed, 28/08/07.

(Gallardo-Valencia and Sim, 2009) Gallardo-Valencia, R.;E.; Sim, S.;E.;, "Continuous and Collaborative Validation: A Field Study of Requirements Knowledge in Agile Methodologies", Managing Requirements Knowledge (MARK), Second International Workshop on, pp.65-74, 1-1 Sept 2009

(Garrett, 2002) Garrett, J.; J.; "The Elements of User Experience: User-Centered Design for the Web", User-Centered Design for the Web, New Riders Publishing, 2002.

(Garrido, et al., 2009) Garrido, A.; Rossi, G.; and Distante, D.;, "Systematic Improvement of Web Application Design", Journal Of Web Engineering, vol.8, no.4, pp.371-404, 2009.

(Gause and Weinberg) Gause D.; C.; and Weinberg, G.; M.;, "Exploring Requirements: Quality Before Design", Dorste House, New York, 1989.

(Gerber, et al., 2007) Gerber, A.; Van der Merwe, A.; Ronell, A.;, "Implications of Rapid Development Methodologies", CSITEd, Mauritius, November 2007.

(Ginige and Murugean, 2001) Ginige, A.; and Murugean, S.;, "Web Engineering: An introduction", IEEE Multimedia, vol.8, no.1, pp.14-18, 2001.

(Ginige, et al., 2001) Ginige, A.; Murugesan, S.;, "The Essence of Web Engineering Managing the Diversity and Complexity of Web Application Development", IEEE MultiMedia, vol 8, no 2, pp.22-25, 2001.

(Gotel and Mäder, 2009) Gotel, O.; Mäder, P.;, "How to Select a Requirements Management Tool: Initial Steps", 17th IEEE International Requirements Engineering Conference, pp.365-367, 2009.

(Gotel, 1995) Gotel, O.;, "Contribution structures for requirements traceability", Ph.D. Thesis, Imperial College, Department of Computing, London, England, 1995.

(Griffiths , et al., 2002) Griffiths, G.; Hebbron, B.; D.; Lockyer, M.; A.; and Oates, B.; J.; "A simple method & tool for Web Engineering", In Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE '02). ACM, New York, NY, USA, pp.755-762, 2002.

(Griffiths , et al., 2004) Griffiths, G.; Lockyer, M.;A.; Oates, B.;J.; Hebbron, B.;D.;, "Web engineering in Internet degrees," Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference, pp. 218- 222, 2004.

(Griffiths and Lockyer, 2004) Griffiths, G.; and Lockyer, M.;A.;, "Early Life Cycle Emphasis", Grand Challenges in Education, Newcastle, pp.29-31, March 2004.

(Griffiths et al., 2006) Griffiths, G.; Oates, B.; J.; Lockyer, M.; A.;, "Systems Analysis Techniques For The Web", Proceedings of AIS SISSAND Symposium on Systems Analysis & Design, Galway, Ireland, pp.95-104, 2006.

(Griffiths, et al., 2003) Griffiths, G.; Lockyer, M.; A.; Oates, B.; J.; Hebbron, B.;D.; "Simple Techniques for Complex Web Applications", ICWI, pp.1068-1071, 2003.

(Haag, et al., 2006) Haag, S.; Cummings, M.; McCubbrey, D.; J.; Pinsonneult, A.; and Donovan, R.;, "Information Management Systems for the Information Age", McGraw-Hill Ryerson, 2006.

(Heinze and Proctor, 2004) Heinze, A.; and Procter, C.;, "Reflections On The Use Of Blended Learning", Education in a Changing Environment, 13th-14th September 2004.

(Heller, 1986) Heller, F.;, "The Use and Abuse of Social Science", Sage, London, 1986.

(Helming, et al., 2010) Helming, J.; Koegel, M.; Bruegge, B.; Berenbach, B.;, "Unified Requirements Modeling for Environmental Systems", Second International Workshop on Software Research and Climate Change, 2010.

(Hevner, et al., 2004) Hevner, A.; R.; March, S.; T.; Park, J.;, "Design Science in Information Research", MIS Quarterly Vol. 28, No. 1, pp. 75-105, March 2004.

(Hoffmann, et al., 2004) Hoffmann, M.; Kuhn, N.; Weber, M.; Bittner, M.;, "Requirements for requirements management tools", Requirements Engineering Conference, Proceedings. 12th IEEE International, pp. 301-308, 6-11 Sept 2004.

(IBM, 2000) Design @ IBM, URL: http://www.ibm.com/ibm/hci/guidelines/web/print.html. Last Accessed: 7/12/2010.

(Ilic, 2007) Ilic, D.;, "Deriving Formal Specifications from Informal Requirements", Computer Software and Applications Conference, COMPSAC 2007, 31st Annual International , vol.1, pp.145-152, 2007.

(Isakowia , et al., 1995) Isakowia T.; Stohr, E.; Balasubramaniam, P.;, "RMM: a methodology for structured hypermedia design", Commun. ACM vol.38, no. 8, pp.34-44, 1995.

(ITU-T, 2008) URL: http://jucmnav.softwareengineering.ca/ucm/bin/view/UCM/DraftZ151Standard, last accessed 15/02/11)

(Jakob, et al., 2007) Jakob, O.; S.; Schwarz, H.; Kaiser, F.;, "flashWeb: graphical modeling of web applications for data management", ACM International Conference Proceeding Series; Vol. 334, pp.59-64, 2007.

(Karlsson, et al., 1998) Karlsson, J.; Wohlin, C.; and Regnell, B.;, "An Evaluation of Methods for Prioritizing Software Requirements", Journal of Information and Software Technology, Vol.39, No. 14-15, pp. 939-947, 1998.

(Kautz, et al., 2007) Kautz, K.; Madsen, S.; and Nørbjerg, J.;, Persistent problems and practices in information systems development. Information Systems Journal, Vo.17, No.3, pp. 217-239, July 2007.

(Kemmis and McTaggart, 1988) Kemmis, S. and McTaggart, R., The Action Research Planner, 3rd ed, Geelong: Deakin University, 1988.

(Kitapci and Boehm, 2006) Kitapci, H.; Boehm, B.W.; , "Using a Hybrid Method for Formalizing Informal Stakeholder Requirements Inputs," Comparative Evaluation in Requirements Engineering, 2006. CERE '06. Fourth International Workshop on , vol., no., pp.48-59, Sept. 2006.

(Knight, 2002) Knight, P.;, "Being A Teacher In Higher Education", Open University Press, pp.11, 2002.

(Koch, 2006) Koch, N.;, "Transformations techniques in the model-driven development process of UWE", Proceedings of the 2nd Workshop on Model-driven Web Engineering (MDWE), Palo Alto, California, 2006.

(Koch, et al., 2008) Koch N.; Knapp, A.; Zhang, G.; and Baumeister, H.;, "UML-Based Web Engineering, An Approach Based on Standards", Web

Engineering: Modelling and Implementing Web Applications Human-Computer Interaction Series, pp.157-191 2008

(Kolb, 1984) Kolb, D.; A.;, "Experiential Learning: Experience As The Source Of Learning And Development", Case Western Reserve University, pp.41, 1984.

(Lang and Fitzgerald, 2005) Lang, M.; and Fitzgerald, B.;, "Hypermedia Systems Development Practices: A Survey", IEEE Software, Vol .22, No.2, pp.68-75, March-April, 2005.

(Larman and Basili, 2003) Larman, C.; Basili, V.;R.;, "Iterative and incremental developments. a brief history" Computer, vol.36, no.6, pp. 47-56, June 2003.

(Lee, et al., 1997) Lee, A.;S.; Liebenau, J.; and DeGross, J.;I.; "Qualitative Research", pp.276-300, Chapman and Hall, London, 1997.

(Leffingwell and Widrig, 2003) Leffingwell, D.; and Widrig, D.;, "Managing Software Requirements: A Use Case Approach", Second Edition, 2003.

(Leffingwelland Widrig, 2010) Leffingwell, D.; and Widrig, D.;, "Agile Software Requirements: Lean Requirements Practices for Teams", Programs and the Enterprise (Agile Software Development), Addison Wesley, pp.228, 2010.

(Lockyer, et al., 2003) Lockyer, M.; A.; Griffiths, G.; Hebbron, B.; D.; Oates, B.; J.;, "A Teaching Method & Tool for Web Engineering", Third IEEE International Conference on Advanced Learning Technologies (ICALT'03), pp.284,2003.

(Lockyer, et al., 2003) Lockyer, M.; A.; Griffiths, G.; Hebbron, B.; D.; Oates, B.; J.;,"A framework for a Web-based toolset to support the teaching of software and Web engineering" Frontiers in Education, 2003. FIE 2003. 33rd Annual , vol.3, pp.5-8, Nov 2003.

(Lockyer, et al., 2003) Lockyer, M.; A.; Griffiths, G.; Hebbron, B.; D.; and Oates, B.; J.;, "A Teaching Method & Tool for Web Engineering", Advanced Learning Technologies: Technology Enhanced Learning, IEEE Computer Society, Athens, Greece, 9-11 July 2003.

(Lowe and Eklund, 2002) Lowe, D.; and Eklund, J.;, "Client Needs and The Design Process in Web Projects", Journal of Web Engineering, Vol.1,No.1, 2002.

(Lowe and Hall, 1999) Lowe, D.; and Hall, W.;, "Hypermedia and the Web: An Engineering Approach", John Wiley and Sons, pp.6, 1999.

(Luna, et al., 2010) Luna, E.; R.; Panach, J.;I.; Grigera, J.;I.; Rossi, G.; Pastor, O.;, "Incorporating Usability Requirements In a Test/Model-Driven Web Engineering Approach", Journal of Web Engineering, pp.132-156, 2010.

(Macaulay and Mylopoulos, 1995) Macaulay, L.; and Mylopoulos, J.;,"Requirements Engineering: An Educational Dilemma", Automated Software Engineering, Vol.2, No.4, pp.343-351, September 1995.

(Macaulay, 1993) Macaulay, L.;, "Requirements capture as a cooperative activity", Requirements Engineering, Proceedings of IEEE International Symposium on, pp.174-181, 4-6 Jan 1993.

(Macleod, 1994) Macleod, M.;, "Usability in Context: Improving Quality of Use", Proceedings of the International Ergonomics Association 4th International Symposium on Human Factors in Organizational Design and Management, Stockholm, pp.1, 1994.

(Maher, 2004) Maher, A.;, "Learning Outcomes in Higher Education: Implications for Curriculum Design and Student Learning", Journal of Hospitality, Leisure, Sport and Tourism Education, Vol.3, No.2, 2004.

(March and Smith, 1995) March, S.; T.; and Smith, G.;F.;, "Design and natural science research on information technology", Decision Support Systems, vol. 15, pp.251-266, 1995.

(McDonald and Welland, 2001) McDonald, A.; Welland, R.;, "Web Engineering in Practice", Proceedings of the Fourth WWW10 Workshop on Web Engineering, pp.21-30, May 2001.

(McKay and Marshall, 2001) McKay, J.; and Marshall, P.;, "The Dual Imperatives of Action Research", Information Technology and People, pp.46-59, 2001.

(Meldrum and Rose, 2004) Meldrum, M.; and Rose, J.;, "Activity Based Generation of Requirements for Web-Based Information Systems: The SSM/ICDT Approach", ECIS 2004 Proceedings, Paper 143, 2004.

(Mich, 2004) Mich, L.; Franch, M.; and Novi Inverardi, P.;L.;, "Market Research for Requirements Analysis using Linguistic Tools", Requirements Engineering, Springer, vol. 9, no. 1, pp.40-56, 2004.

(Murugesan, et al., 1999) Murugesan, S.; Deshpande, Y.; Hansen, S.; and Ginige, A.;, "Web Engineering: A New Discipline for Development of Web_based Systems", Proceedings of the First International Conference of Software Engineering (ICSE) Workshop on Web Engineering, Los Angeles, USA, 1999.

(Myers, 1997) Myers, M.; D.;, "Critical Ethnography in Information Systems", In Proceedings of the IFIP TC8 WG 8.2 international conference on Information systems and qualitative research, Chapman \& Hall, Ltd., London, UK, pp.276-300, 1997.

(Oates, 2005) Oates, B.; J.;, "Researching Information Systems and Computing", Sage Publications, 2005.

(Paetsch, et al., 2003) Paetsch, F.; Eberlein, A.; Maurer, F.; "Requirements Engineering and Agile Software Development", Enabling Technologies, IEEE International Workshops on, Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp.308, 2003.

(Palmer and Felsing, 2002) Palmer, S.; R.; and Felsing, J.; M.;, "A Practical Guide to Feature-Driven Development", Prentice Hall, 2002.

(Perrone and Bolchini, 2005) Perrone,V.; and Bolchini, D.;, "Designing Communication-Intensive Web Applications: Experience and Lessons from a Real Case", JCS&T, Vol.5, No.2, 2005.

(Pitts and Browne, 2007) Pitts ., M.; G.; and Browne, G.; J.;, "Improving requirements elicitation: an empirical investigation of procedural prompts", Information Systems Journal, vol.17, no.1, pp.89-110, 2007.

(Powell, et al., 1998) Powell, T.; A.; Jones, D.; L.; Cutts, D.; C.;, "Web site engineering: beyond Web page design", Prentice Hall PTR, 1998.

(Pressman and Lowe, 2008) Pressman, R.; S.; Lowe, D.;, "Web Engineering: A Practioner's Approach", McGraw-Hill Higher Education, 2008.

(Pressman, 1998) Pressman, R.;S.;, "Can Internet-Based Applications Be Engineered?", IEEE Software (September/October), pp.104-110, 1998.

(Pressman, 2000) Pressman, R.; S.;, "Software Engineering: A Practitioner's Approach Paperback", pp.13, 2000.

(Procter, 2003) Procter, C.;, "Blended Learning in Practice", In Inaugural Education in a Changing Environment conference, University of Salford, Salford, 2003.

(Ramesh, et al., 2002) Ramesh, B.; Pries-Heje, J.; and Baskerville, R.;L.;, "Internet Software Engineering: A different class of processes", Annals of Software Engineering, pp.169-195, 2002.

(Roberts, et al., 1997) Roberts, D.; Berry, D.; Isensee, S.; Mullaly, J.;, "Developing Software Using OVID," IEEE Software, vol.14, no.4, pp. 51-57, Jul/Aug 1997.

(Robertson and Robertson, 2010) Robertson, J.; and Robertson, S.;, "Volere Requirements Resources", URL: http://www.volere.co.uk/template.htm, last accessed 13/12/2010.

(Romero and Ventura, 2007) Romero, C.; and Ventura, S.;, "Educational data mining: A survey from 1995 to 2005", Expert Systems with Applications, Vol.33, No.1, pp.135-146, July 2007.

(Rowbottom and Aiston, 2006) Rowbottom , D.; and P.; Aiston, S.; J.;, "The Myth of 'Scientific Method' in Contemporary Educational Research", Journal of Philosophy of Education, Vol.40, No.2, pp.137-156, 10 Jul 2006.

(Royce, 1970) Royce, W.; W.;, "Managing the Development of Large Software Systems: Concepts and Techniques", In: Technical Papers of Western Electronic Show and Convention (WesCon), Los Angeles, USA, August 25-28 1970.

(Sampaio, et al., 2004) Sampaio, A.; Vasconcelos, A.; and Sampaio, P.;, "Design and Empirical Evaluation of an Agile Web Engineering Process" Proc. 18th Brazilian Symposium on Software Engineering (SBES'04), pp.194-209, 2004.

(Schwabe, et al., 1996) Schwabe, D.; Rossi, G.; and Barbosa, S.; D.; J.;, "Systematic hypermedia application design with oohdm", In Proceedings of The Seventh ACM Conference on Hypertext, pp.116 -128, 1996.

(Selic, 2003) Selic, B.;, "The pragmatics of model-driven development", Software, IEEE , vol.20, no.5, pp.19- 25, Sept-Oct 2003.

(Soares and Vrancken, 2008) Soares, M.; D.; S.; and Vrancken, J.;, "Model-Driven User Requirements Specification using SysML", JOURNAL OF SOFTWARE, Vol.3, No.6, JUNE 2008.

(Sommerville 2007) Sommerville, I., "Software Engineering", 7th Edition, Addison Wesley, (pp.63, pp.111, pp.143, and pp.345), 2007.

(Sommerville and Sawyer, 1997) Sommerville I.; Sawyer, P.;, "Requirements Engineering: A Good Practice Guide", Wiley, (pp.5, pp.11, pp.216, and pp.236) 1997.

(Stahl and Völter, 2006) Stahl, T.; Völter, M.;, "Model-Driven Software Development", Technology, Engineering, Management, Wiley, 2006.

(Susman and Evered, 1978) Susman, G.;I.; and Evered, R.;D.;, "An Assessment of the Scientific Merits of Action Research", Administrative Science Quarterly, pp. 582-603, 1978.

(Szekely, 1994) Szekely, P.;, "User Interface Prototyping: Tools and Techniques", USC/Information Sciences Institute, 1994.

(Thomas, 1993) Thomas, J.;, "Doing Critical Ethnography", SAGE Publications, Newbury Park, CA, 1993.

(Valderas and Pelechano, 2009) Valderas, P.; and Pelechano, V.;, "Introducing requirements traceability support in model-driven development of web applications", Information and Software Technology, Vol.51, No.4, pp.749-768, April 2009.

(Valderas, et al., 2007) Valderas, P.; Pelechano, V.; and Pastor, O.;, "A transformational approach to produce web application prototypes from a web requirements model", Int. J. Web Engineering and Technology, Vol. 3, No. 1, pp.4-42, 2007.

(Whitson, 2006) Whitson, G.;, "WebHelix: Another Web Engineering Process", Computing Sciences in Colleges, vol. 21, no. 5, pp.21-27, May 2006.

(Wiegers, 2006) Wiegers, K., "More about software requirements", Microsoft Press,(pp.17, pp.57, and pp.119), 2006.

(Xiaoqing, et al., 2004) Xiaoqing Liu.; Veera, C.; S.; Sun, Y.; Noguchi, K.;, Kyoya, Y.; , "Priority assessment of software requirements from multiple perspectives," Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International, vol.1, pp. 410- 415, 28-30 Sept 2004.

(Young, 2007) Young, I.;, "Task-Based Audience Segmentation 2005", http://www.adaptivepath.com/publications/reports/task/task_exec.pdf, last accessed 09/09/07.

(Yu, 1997) Yu, E.;S.;K.;, "Towards modelling and reasoning support for early-phase Requirements Engineering", Requirements Engineering, Proceedings of the Third IEEE International Symposium on, pp.226-235, 6-10 Jan 1997.

(Yusop, et al., 2006) Yusop, N.; Zowghi, D.; Lowe, D.;, "THE IMPACTS OF NON-FUNCTIONAL REQUIREMENTS IN WEB SYSTEM PROJECTS", European and Mediterranean Conference on Information Systems (EMCIS), 2006.

(Zhang, et al., 2003) Zhang, J.; Chang, C.; K.; Chung, J.; Y.;, "Mockup-driven fast-prototyping methodology for Web Requirements Engineering", Computer Software and Applications Conference, 2003. COMPSAC 2003, Proceedings 27th Annual International, pp. 263- 268, 3-6 Nov 2003.

(Zowghi and Gervasi, 2003) Zowghi, D.; and Gervasi, V.;, "On the interplay between consistency, completeness and correctness in requirements evolution", Information and Software Technology, Eighth International Workshop on Requirements Engineering: Foundation for Software Quality, Volume 45, Issue 14, (pp.993-1009 and pp.763-779), 2003.

(Zowghi and Cleland-Huang, 2008) Zowghi, D.; and Cleland-Huang, J.;, "Transforming the Requirements Engineering Classroom Experience", International Requirements Engineering, RE '08 , pp.297-297, 8-12 Sept 2008.

Appendix A1 - RUMM Survey Questionnaire

Please fill in the following questionnaire as fully and honestly as you can, your response and comments will help determine where the next stage of the research will go. *Thank you.*

1. **How useful did you find using this approach was in helping you define your audience?**

Very Useful/helpful	Useful/helpful	Neither helpful nor unhelpful	A little unhelpful/confusing	Complicated & very unhelpful

Please tick a box

2. **If you have used other user defining approaches in the past how do you think this approach compares?**

Never used other approaches	Much more useful	More useful	Not much better really	I'll stick with my original approach!

Please tick a box

3. **Do you think there are any points missing from the list that should be considered?**

□ Yes □ No If "Yes" Please state what point(s):

4. **Do you think there is any unnecessary information being gathered through this method?**

□ Yes □ No If "Yes" Please state what information:

5. **The next stage of the model will be to link the user model you have created with issues that you need to consider in your design. It is anticipated that this would be generated online and would provide a list of design guidelines that is UNIQUE to the user model you have created. If such a tool were available to you how useful and helpful do you think it would be to *you* as a designer?**

Very useful & helpful	Useful and helpful	Neither helpful nor unhelpful	Unhelpful	Very unhelpful

Please tick a box

Appendix A2 - Rapid User Modelling Method (RUMM) Fill In Forms

	Primary Users	Secondary Users
Who is the user?		
Age		
Children – KS 0 (0-4 yrs)		
Children – KS 1 (4-7 yrs)		
Children – KS 2 (7-11 yrs)		
Children – KS 3 (11-14 yrs)		
Children – KS 4 (14-16 yrs)		
Adults (17-30)		
Adults (30-50)		
Adults (50+)		
Gender & Culture		
Male		
Female		
Both		
General Cultural Background (?)		

Level of computer use/competence		
Novice		
Intermediate		
Expert		
Employment job/type (?)		
What does the user expect to do with the application?		
Use it as the main part of their job?		
Assist them to do their job?		
Allow them to buy something online?		
Assist them to find out information -specific and/or general?		
Provide fun or leisure activities?		
Help them learn something?		
When will the user use the application?		
In the course of their job?		
At home in their own leisure time?		

In library or other public access point?		
How will the user use the application?		
At work		
With modem link (what speed)		
With broadband connection		
Stand alone CD/DVD		
At home		
With modem link		
With broadband connection		
Stand alone CD/DVD		
On PC (specification)		
On Mac (specification)		
On Linux platform (specification)		
Kiosk		
PDA		
With assistive technologies? (specify)		

Now write a brief summary of your primary users:

Now write a brief summary of your secondary users:

Now identify some of the implications of these issues for your design under the following headings:

Layout

Colour

Content

Navigation

Appendix A2.1 - DFU Module Specification

TEESSIDE UNIVERSITY
MODULE SPECIFICATION

Module Title	**Design for Usability**		
Module Status	New	**Date of Official Approval**	2003
SITS Module Code	**MUL2003-N**		
Module Descriptor	The module aims to give the students an understanding of the design of usable yet creative interfaces. Much of the focus of the module is on web based applications, but many of the approaches and techniques discussed could be applied to the development of other computer interfaces. The module emphasises the importance of the relationship between the user and the system. Key issues include the understanding of human cognitive abilities - memory, vision, problem-solving and reasoning - then builds upon this to consider interaction methods, Usability Engineering Life Cycle, guidelines for 'good' and creative design, and evaluation of interfaces. There is a strong element of practical work in prototyping an interactive interface, including user testing and evaluation.		
Key Words (10 max)	User Centred Design, human cognition, perception & creativity, interactivity, interface development		

Module Type	Standard	**Credits** (UoT CAMS)	20
Total Learning Hours	200	**Level**	II

Summative Assessment	ICA 100%
Assessment Marking Scheme	2004 UG Module
Assignment Submission	School of Computing

Available as Open/Distance Learning	No
Assignment Re-submission possible	No
Roll On / Roll Off	No
Compensatable	Yes

Pre-requisites	None
Co-requisites	None
Other Requirements	None

TEESSIDE UNIVERSITY QUALITY UNIT
APPROVAL OF A MODULE - MODULE SPECIFICATION

Module Aims

1. To provide students with the knowledge needed to understand the role of the human user as a component in a human-computer system.
2. To challenge the students attitudes towards applying creative and aesthetically pleasing designs to interface development.
3. To provide the student with tools and techniques required to analyse, design, develop and evaluate an interactive user interface, with a focus on Usability Engineering techniques and practices including evaluating if the application meets the needs of the target user group.
4. To provide the student with an understanding of a variety of interaction methods and the appropriate methodologies and guidelines for good design and development.
5. To provide the student with the practical experience of using a prototyping tool for interactive user interface product development.

Indicative Content

1. Introduction to human-computer interaction.
2. The User Centred Development Methodology, including establishing user requirements, preparing a mission statement, conceptual design, implementation, testing and evaluation.
3. Understanding the user: cognition [perception, memory, attention, problem-solving and reasoning]; semiotics, society, culture and gender.
4. Producing interactions to enhance the users experience.
5. Creativity and its role in the design of an effective user interface solution.
6. Evaluation and comparison of traditional design methods and strategy.
7. Navigation principles and strategies.

8. Conventions, constraints and limitations of the implementation process.
9. Analysis, design, development and evaluation of the human-computer interface.
10. Techniques for building 'user friendly' interfaces.
11. Practical interface development, prototyping and testing.

Learning Strategy

The overall strategy for teaching and learning is described in the Framework document. Specific features of this module are:Delivery of underlying theoretical concepts of the field of creative yet usable interface development through a series of lectures, with further in-depth exploration by students via tutorial tasks and structured, directed research.Exploration of practical aspects is via laboratory-based activities, including the use of prototyping tools for interface development, and the use of evaluation techniques for refining and user testing of that interface.Much of the module revolves around active participation by the student, and uses a variety of delivery styles supported by a range of resources to encourage participation, learning and reflection. Activities undertaken by students include completion of lecture-related tasks; tutorial work on aspects such as user analysis, production of design documentation and various evaluation techniques and practices; practical work on interface design for prototype development and user testing and evaluation. There is a mix of group and individual activities.

Learning Outcomes

On successful completion of this module, the student will be able to:
1. Demonstrate and document an understanding of the role of the human as a component in a human-computer system.
2. Apply an aesthetic and creative solution to the design of an interface.
3. Analyse, design, develop and evaluate an interactive interface.
4. Use an appropriate design methodology and design guidelines in the production of an application.
5. Implement a prototype interface using a specified application development tool.

Assessment

Assessment Strategy

The formal mechanism for assessing student achievement is via In-Course Assessment. This will take the form of the design and development of a web-based application that should reflect good usability design principles. The students will also have to demonstrate their knowledge of the needs of the user through a form of scholarly discourse. The assessment will measure all the learning outcomes of the module. The School operates a standard procedure for providing (at least) a minimum level of feedback to students.

Assessment Criteria

Indicative Resources

Purchase
not applicable

Essential
Tutor Based Resources. Lecture Notes. Practical Tutorials. Tutorial Tasks. Electronic media.

Recommended
"User-Centered Web Development" by Jonathan Lazar, Jones & Bartlett Computer Science,
2001."Dreamweaver MX for Windows & Macintosh, Visual Quick Start" by J. Tarin Towers, Peachpit,
2003."The Essential Guide To User Interface Design : An introduction to GUI design principles and techniques" by Wilbert O. Galitz, Wiley Computer, 2002

Journals
not applicable

Electronic
Blackboard Virtual Learning Environment.

Accessibility

The School of Computing endeavours to make all of its modules inclusive and does its best to adopt accessible and inclusive practices but we are aware that we cannot anticipate every possible special needs or requirements. There may be elements of this module (resources, assessment, learning and teaching methods, etc) that may present difficulties for students with special needs. You are strongly advised to check the module details carefully and discuss any potential problems with the Special Needs tutor so that your particular needs can be accommodated wherever possible.

Appendix A2.2 - IID Module Specification

TEESSIDE UNIVERSITY
MODULE SPECIFICATION

Module Title	**Integrated Development**		
Module Status	New	**Date of Official Approval**	2005
SITS Module Code	**MUL4002-N**		
Module Descriptor	This module will provide the student with skills necessary to manage the multimedia implementation process. The module will take an integrated approach to the use of multimedia development tools. Students will also be provided with concepts and skills for utilising an appropriate scripting language. This module is appropriate to students who already have an understanding of multimedia production and who wish to extend these skills further. The module will be supported by lectures and practical tutorials. Other resources and additional learning opportunities will be presented via Blackboard and alternative web delivery systems. The module will be assessed by an in course assessment which will involve the development of a substantial multimedia product.		
Key Words (10 max)	Multimedia, Implementation, Scripting, Workflow Management		

Module Type	Standard	**Credits** (UoT CAMS)	20
Total Learning Hours	200	**Level**	IV

Summative Assessment	ICA 100%
Assessment Marking Scheme	2004 PG Module

Assignment Submission	School of Computing
Available as Open/Distance Learning	No
Assignment Re-submission possible	No
Roll On / Roll Off	No
Compensatable	Yes

Pre-requisites	None
Co-requisites	None
Other Requirements	None

TEESSIDE UNIVERSITY QUALITY UNIT
APPROVAL OF A MODULE - MODULE SPECIFICATION

Module Aims

The module aims to:
1. Establish the principles of a well managed multimedia implementation process.
2. Introduce key scripting concepts to enable the development of interactive multimedia.
3. Present appropriate web technologies for the use in developing web based multimedia.
4. Give students the opportunity to gain experience in developing multimedia using an industry standard integrated development package.
5. Encourage the student to reflect on the implementation process.

Indicative Content

Introduction to a 'work flow model' for the management of the implementation process.Effective use of Implementation tools, including industry standard Multimedia/Web Authoring tools. Utilisation of Macromedia Studio, (Flash, Dreamweaver and Fireworks). Fundamentals and concepts of scripting for multimedia and the web. - Problem solving and providing efficient solutions using a range of scripting languages, including Actionscript and Javascript. - Working with objects, functions, conditionals and loops.- Data Types, Variables, Assignment, Input and Output.Overview of future technologies and standards, and how these affect the implementation process. Writing XHTML compliant mark-up. Using CSS to efficiently present content for the World Wide Web. Awareness of the need to address accessibility requirements in the implementation of multimedia and web based applications. Introduction to Server-side technologies, including FTP and client to server communication. Data validation and security issues. The evaluation process and its integration into the product lifecycle.

Learning Strategy

The module will be taught by blend of lectures, practical tutorials and online learning. Blackboard will be used to integrate learning objects, where this is deemed appropriate. This will also include online discussion, which should combine to foster a collaborative learning experience. Lectures will be used to develop understanding of the underpinning theory and concepts involved in implementing integrated multimedia. Case studies will be used to illustrate how the theory relates to commercial application implementation. In addition, active lectures, using a variety of techniques, will enable students to investigate key concepts in relation to the tools, within the session itself. Practical tutorials will concentrate on developing key skills within the implementation environment, including graphical asset production, implementation of multimedia/web based objects and access to server side technologies. Self directed learning, taking into account the level of the module, will be a required from the student.

Learning Outcomes

Knowledge & Understanding

On successful completion of this module, the student will be able to:
1. Appraise different web technologies and apply an appropriate solution to a given scenario.

Cognitive & Intellectual Skills

On successful completion of this module, the student will be able to:
1. Critically evaluate their implementation approach.

Practical & Professional Skills

On successful completion of this module, the student will be able to:
1. Compose a well formed scripting solution to address the requirements of a design brief.
2. Design and construct a multimedia solution addressing the requirements of a design brief.

Key Transferable Skills

On successful completion of this module, the student will be able to:
1. Manage the implementation phase of a multimedia development lifecycle.

Assessment

Assessment Strategy

The module will be assessed by in-course assessment (ICA): Using a project brief and specification, students will design and develop a multimedia application, along with a linked 'micro web site'. This will incorporate both the theoretical and practical components of the module, such as writing functions for the development of interactions and animations and post-implementation evaluation. Clear assessment marking criteria will be presented within the ICA specification. Individual feedback will be given in line with these criteria. The School operates a standard procedure for providing (at least) a minimum level of feedback to students in line with Minimum Standards.

Assessment Criteria

Assessment criteria will be provided, related to the set tasks, stating how marks will be allocated.

Indicative Resources

Purchase
Not Applicable

Essential
This module will require access to a lecture theatre and laboratory facilities that support digital presentations, as well as access to web development tools as outlined below.
1. Non web based tools (e.g. Email, FTP, SSH).2. Dynamic HTML enabled Web Browsers (eg. Mozilla Firefox, Internet Explorer).3. Web Site Development tools (eg. Macromedia Dreamweaver).4. Multimedia Development tools (eg. Flash).5. Asset creation tools (eg. Adobe Photoshop, Macromedia Fireworks).
1. Chun, Russell, Macromedia Flash MX 2004 Advanced for Windows and Macintosh (Visual QuickPro Guides),
2004.2. J.Tarin Towers, Macromedia Dreamweaver MX 2004 for Windows and Macintosh (Visual QuickStart Guides),
2004. 3. Cohen, Leonard, Macromedia Fireworks MX 2004 for Windows and Macintosh (Visual QuickStart Guides),
2004.

Recommended
1. Oliver, D., and Morrison, M., Sams Teach Yourself HTML and XHTML in 24 hours, Sams,
2003.2. Vaughan, Tay, Multimedia : Making it work, McGraw-Hill Technology Education,
2004.3. Chapman, Nigel P, Digital multimedia, Wiley,
2004.4. Chapman, Nigel P, Digital media tools, Wiley,
2002.

Journals
International Journal of Web Engineering and TechnologyInteracting with Computers New Review of Hypermedia and Multimedia

Electronic
Blackboard. http://www.blackboard2.tees.ac.ukMedia-drive.co.uk. http://www.media-drive.co.uk (2005)

Accessibility

The School of Computing endeavours to make all of its modules inclusive and does its best to adopt accessible and inclusive practices but we are aware that we cannot anticipate every possible special needs or requirements. There may be elements of this module (resources, assessment, learning and teaching methods, etc) that may present difficulties for students with special needs. You are strongly advised to check the module details carefully and discuss any potential problems with the Special Needs tutor so that your particular needs can be accommodated wherever possible. Further advice is available from the University Student Services staff.

Appendix A2.3 - RUMM Survey Codebook

Question 1. How useful did you find using this approach was in helping you define your audience?

Very Useful/helpful:	**a**
Useful/helpful:	**b**
Neither helpful nor unhelpful:	**c**
A little unhelpful/confusing:	**d**
Complicated & very unhelpful:	**e**

Question 2. If you have used other user defining approaches in the past how do you think this approach compares?

Never used other approaches:	**a**
Much more useful:	**b**
More useful	**c**
Not much better really:	**d**
I'll stick with my original approach:	**e**

Question 3. Do you think there are any points missing from the list that should be considered?

Yes: Y
No: N

Question 4. Do you think there is any unnecessary information being gathered through this method?

Yes: Y
No: N

Question 5. The next stage of the model will be to link the user model you have created with issues that you need to consider in your design. It is anticipated that this would be generated online and would provide a list of design guidelines that is UNIQUE to the user model you have created. If such a tool were available to you how useful and helpful do you think it would be to *you* as a designer?

Very useful & helpful	**a**
Useful and helpful	**b**
Neither helpful nor unhelpful	**c**
Unhelpful	**d**
Very unhelpful	**e**

Appendix A2.4 - Excerpts of Observational Logs (AR1, 2 and 3)

Sample of Observational Logs. (Recorded in notebook form and significant observations translated).

AR1 Observations

Date	Location	Observational Remarks
Oct 2005	Lab	Students appear to be adopting RUMM in their initial planning stages, but quickly becoming bored! Some students asking how to gather information about the user. Students not completing RUMM 'at one sitting'.
Nov 2005	Lab	Students generally need a lot more support in using RUMM than had been expected. It is feared that the reasons for this is that I have not provided them with sufficient support material.
December 2005	Lab	Some students on the masters module IID wish to modify/extend RUMM to suite their own development practices. Unsure if this is valid, but it does raise some questions regarding the limitations of RUMM in its current form.
Jan 2006	Lab	Some student expressing if they are able to use persona's to model the user, rather than the built in 'user characteristics' of RUMM.
June 2006	Office	Some students are submitting incomplete

		requirements within their ICA's. From assessing their websites, it seems that there is some relationship to their documented requirements. Generally, much more analysis is taking place than in previous years.

AR2 Observations

Date	Location	Observational Remarks
Oct 2006	Lab	Students are jumping into different stages in the modified WURF process meta-model. It had been expected that a sequential approach to be taken. This is potentially problematic, as analysis of functional requirements is dependent on previously completed stages such as tasks.
Oct 2006	Lab	The association system is causing issues for student. Some students are transposing WURF into their own requirements document.
Nov 2006	Lab	The separation of users in to Actors seemed to be working well with students, particularly those developing web applications that consume a web service. Having unlimited users is providing students with much more scope for modelling the whole application, rather than being limited to primary and secondary (RUMM).

June 2006	Office	Assessment of the student submissions is providing some useful insights in the way that they are adapting WURF to suite their development approach. Student still submitting incomplete requirements! It should be possible to automate WURF in some way in order to resolve issues regarding its completeness.

AR3 Observations

Date	Location	Observational Remarks
Oct 2009	Lab	Major issue with the association system within eWURF. Student indicated that the screen on the association page goes blank after they attempt to define the association. The only way to recover is to close the browser.
Oct 2009	Lab	Some students are able to add empty objectives to the system, which causes major problem with the automated SRS system.
Nov 2009	Lecture	Provided a research method lecture to the final year students embarking on their dissertations. Also showed them eWURF. Some web students indicated that they wanted to use previous ‘paper based’ versions (RUMM) (WURF), as they prefer there approach.
June 2009	Office	Starting to undertake data analysis of student marks vs usage of eWURF (data captured by student usage of online WURF). Preliminary indications show that students used the online tool 7 to 8 times over the duration of the module. I had expected this to be much higher.
Oct 2010	Office	eWURF registration system still being actively used even though the release phase is now finished. New

		users traced to Franchised programmes running in Botho College in Botswana and London Tec in Sri Lanka.

Appendix 2.5 – OBS Module Specification

TEESSIDE UNIVERSITY
MODULE SPECIFICATION

Module Title	**On-Line Business Systems**		
Module Status	New	**Date of Official Approval**	2003
SITS Module Code	**MUL3030-N**		
Module Descriptor	This module takes students, who may have no specific computing knowledge, through the theoretical and technical skills required to design and develop dynamic web applications. A practical approach will be adopted throughout and students will be expected to develop a dynamic database driven website utilising appropriate authoring tools and server side scripts. Students will be assessed individually on their ability to analyse user requirements and develop dynamic web applications. Students will also be introduced to a methodology to underpin project development. This methodology will be user centred and will include tools to model the user characteristics and build a set of requirements specific to the project.		
Key Words (10 max)	Web Development, Web Solutions, Dynamic Website		

Module Type	Standard	**Credits** (UoT CAMS)	20
Total Learning Hours	200	**Level**	6

Summative Assessment	ICA 100%
Assessment Marking Scheme	2004 UG Module
Assignment Submission	School of Computing
Available as Open/Distance Learning	No
Assignment Re-submission possible	Yes
Roll On / Roll Off	No
Compensatable	Yes

Pre-requisites	None
Co-requisites	None
Other Requirements	Forbidden combination with Web Scripting - MUL2021-N.

TEESSIDE UNIVERSITY QUALITY UNIT
APPROVAL OF A MODULE - MODULE SPECIFICATION

Module Aims

This module aims to:

1. Develop an understanding of the programmatic controls for both the content and presentation of web pages e.g. XHTML and CSS-P.
2. Introduce theoretical and practical considerations when working with authoring tools, server-side scripting languages and databases.
3. Encourage the use of development methodologies and analysis tools throughout the development process in order produce a user and client centred solution.
4. Provide students with an opportunity to gain experience in developing web applications using an industry standard integrated development environment.
5. Provide students with the opportunity to design and develop server side scripts and database tables.
6. Highlight the need for continual evaluation, maintenance and marketing during the lifecycle of an online business system.

Indicative Content

- Introduction to methodologies for the management of the implementation process. Use of a user modelling method to help understand user requirements.
- Writing XHTML compliant mark-up. Using CSS-P to separate presentation from content. Awareness of the need to address accessibility requirements in the implementation of web based applications.
- Introduction to Server-side technologies, e.g. PHP and MySQL
- Client to server communication model. Data validation and security issues when developing and deploying online business systems.
- Concepts of scripting for the web.
    1. Problem solving and providing efficient solutions using a range of scripting languages, including PHP and Actionscript.
    2. Working with objects, functions, conditionals and loops.
    3. Writing database connection and query scripts.

    4. Data Types, Variables, Assignment, Input and Output.

- Introduction to Flash for the production of multimedia components and applications.

Learning Strategy

The module will be taught by a blend of lectures, practical tutorials and online resources. Blackboard will be used to provide learning objects and supporting materials where this is deemed appropriate. Students will be provided with the necessary theoretical concepts in relation to SQL and PHP. Extensive tutorial support is provided for the IDE and to enable the connection to the database for query execution.

Lectures will be used to develop understanding of the underpinning theory and concepts involved in implementing web applications. Active lectures, using a variety of techniques, will enable students to investigate key concepts in relation to the tools within the session itself.

Practical tutorials will concentrate on developing key skills within the implementation environment, including design and application implementation. Students will be expected to dedicate a substantial amount of time working on their web applications outside of class time. This self-directed learning will include research and developing practical skills.

Learning Outcomes

Knowledge & Understanding

On successful completion of this module, the student will be able to:

1. Explain what is involved in the development and design of a dynamic, database enabled web application using contemporary tools and techniques.

Cognitive & Intellectual Skills

On successful completion of this module, the student will be able to:

2. Justify the design and development of web applications and critically evaluate their implementation approach.

Practical & Professional Skills

On successful completion of this module, the student will be able to:

3. Design and develop web applications using an industry standard integrated development environment (IDE).

Key Transferable Skills

On successful completion of this module, the student will be able to:

4. Document the web application development process.

5. Critically evaluate appropriate methodologies and recognise the need for continual development and marketing of an online business system.

Assessment

Assessment Strategy

The module will be assessed by one component weighted at 100% that assesses the learning outcomes 1,2,3,4 and 5. Students will receive the ICA documentation at the start of the module and will work towards the ICA throughout the module through the ongoing development of a portfolio. The assessment will include the implementation of a dynamic website and a written report of approximately 2500 words documenting the design and implementation of the website.

Formative feedback is provided throughout the module.

The School operates a standard procedure for providing (at least) a minimum level of feedback to students in line with the University's Assessment and Feedback Policy.

Assessment Criteria

Students will submit a dynamic website for a given scenario. The website will assess learning outcome 3.

Students will also be required to submit a report based on the design and implementation of their dynamic website (approx 2500 words). The report will assess learning outcome 1, 2, 4 and 5. They will be assessed on the quality of their design documentation, the justification of their design choices and a discussion of their implementation.

Students will be provided with a detailed assignment brief.

Indicative Resources

Purchase

Negrino, T. & Smith, D. (2007) *Dreamweaver CS5 for Windows and Macintosh:Visual QuickStart Guide.* Peachpit Press.

Ullman, L. (2011) *PHP for the Web: Visual Quickstart Guide (Visual QuickStart Guides).* 4th ed. Peachpit Press.

Essential

Felke-Morris, T (2008) Web Development and Design Fundamentals with XHTML 4th ed. Addison Wesley,

2008.

Lazar, J (2001) User Centered Web Development - - Jones and Bartlett Publishers

Elmasri, R. & Navathe, S. (2010) Fundamentals of database systems. 6th ed. BostonPA.: Pearson.

Gilmore W J, (2004) Beginning PHP 5 and MySQL : from novice to professional,, Berkeley, Calif. : London : Apress ; Springer

Meloni, J.C.. & Morrison, M. (2010) *Sams Teach Yourself HTML and CSS in 24 Hours.* 8th ed. Sams.

Connolly, T M & Begg C E (2010) Database systems : a practical approach to design, implementation and management 5th ed London : Addison-Wesley,

Recommended

Garrett, J.J. (2011) *The Elements of User Experience.* 2nd ed. New Riders.

Sommerville, I. (2011) *Software Engineering.* 9th ed. Pearson Education.

Journals

International Journal of Web Engineering and Technology

Interacting with Computers

New Review of Hypermedia and Multimedia

Electronic

University of Teesside , *Elearning@Tees*. http://eat.tees.ac.uk.

Accessibility

The School aims to make this module accessible to any student who may benefit by studying it. Students who are concerned about their ability to access the module are advised to contact the School Disability Coordinator for academic advice and the University Student Services staff for details of available support.

Appendix A3 – Related Work

3.1 Introduction

Chapter 2 provided a background to this research programme by outlining some of the problems facing Web, Software and Requirements Engineering. Important theory in respect of RE and SE sets the context for further work in terms of this research programme. This section presents a literature review of related work, subdivided into three sections. It demonstrates how each research cycle reflected the examination of existing approaches in order to solve problems identified from each evaluation.

3.2 Requirement Process, Methods and Tools

A requirements process is underpinned by specific methods and tools that are selected by the development team. Some methods reflect the whole requirements process (*elicitation, analysis, specification and validation*) and some address one or two stages of the typical RE process. For example, some focus on elicitation or the specification of requirements. It was found that some developers choose to combine methods in order to satisfy particular organisational or problem objectives, thereby creating hybrid methods suited to the organisation. The aim of this section is to demonstrate variations in the approaches that are in use by both academics and practitioners.

It was considered important to undertake a structured analysis of the methods by comparing their treatment against the requirements process as defined by Sommerville & Sawyer 1997 and Berry 2003. Table 3.3 demonstrates how each approach addresses the whole requirements process criteria (*elicitation, analysis, specification and validation*). A discussion of how the review undertaken in section 3.3 relates to the three action research cycles can be found in section 3.7.

3.3 Review of Existing Requirements Processes, Methods and Tools

There is a strong argument for suggesting that there is no *definitive* or *one-size fits* all requirements process, method or tool that can be adopted. "Of course, there is no "one-size-fits-all" approach to this critical front-end challenge and development teams will likely is a combination of techniques for any particular circumstance" (Leffingwell and Widrig, 2010). Much depends on the context in which the software is being produced, the organisational expertise in Requirements Engineering and constraints experienced by the organisation such as human resources and budget. It also depends on the approach taken in terms of the development methodology, where agile approaches need to be supported by agile requirements. The aim of this research is to investigate how to bridge the gap between RE and WE by proposing a web user requirements approach aimed to support the inexperienced student user. As such, it is deemed advantageous to consider what already exists in this domain in order to address specific contextual needs. Although there are no organisational constraints there are time pressures from the perspective of the students' support and learning needs, which will influence the proposal of a web user requirements approach.

A number of specific methods and tools can be found in the literature that assists the developer with their requirements discovery, elicitation, analysis and specification. Some of these focus on one stage in the process, whilst others cover the whole process. It should be emphasised that there does not seem to be a definitive method dealing with the *elicitation*, *analysis* and *specification* of web requirements in an educational context. Much of the research therefore investigated methods within the Software Engineering (SE) domain. Whilst it is recognised that there are differences between WE and SE, there are key principles that can be applied to both, thereby contributing to understanding of how RE can be applied to WE. This section investigates and analyses existing methods within SE and a limited number of WE specific approaches.

Requirement Process Stages Method	Elicitation	Analysis	Specification	Validation	Meets All Criteria (elicitation, analysis, specification and validation)
ARM	■	■	■	■	▲
OVID			■	■	
UCA	■	■	■		
TBAS	■				
CI	■				
US	■	■	■	■	▲
MSF	■	■	■	■	▲
NDT	■	■	■	■	▲
JAD	■	■	■		
CRC	■	■			
UC		■			
AWARE	■	■	■	■	▲
URN	■	■	■	■	▲
SOARE	■	■	■	■	▲
SSM/ICDT	■	■	■	■	▲

Table 3.3 Comparison of Requirements Methods.

3.4 Related Work Research Cycle 1 (RUMM)

3.4.1 AWARE Approach

Work undertaken by Bolchini and Paolini, 2002 and 2004 into ‘Goal-Driven Requirements analysis’ provides some useful direction for a student focused web user requirements approach. Bolchini’s work culminated in the development a model for the Analysis of Web Application Requirements (AWARE). This is described by Al-Salem 2007 as “a light-weight methodology based on the ‘i*frame-work’, which blends goal-directed RE and scenario-based techniques for defining hypermedia requirements concerning aspects such as *content*, *interaction*, *navigation* and *presentation* for Web applications” (Al-Salem and Samaha, 2007). An interesting point to note here is the reference to the ‘i* framework’, which can be described as a user modelling approach. It was first proposed by Eric Yu in his doctoral thesis and later refined in his research into “Early-Phase Requirements Engineering” (Yu, 1997). This work proposes that ‘agents’ which are composed of multiple users can be modelled in terms of their characteristics and attributes such as goals, tasks and abilities. The ‘i* framework’ is now part of an international standard for user requirements notation (URN).

AWARE is user orientated, concentrating on *high level communication goals* in relation to the design aspects of the web application. The AWARE meta-model, (see Figure 3.4.1), contains *constructs* that enable developers to build up a rich picture of the web application to be developed. A construct model helps define these in more detail, for example, a ‘Stakeholder’ construct defines a set of users, expressed as *Actors*, with these described in further granularity of detail via the user profile. Another construct named *Goal* provides a mechanism to express informal stakeholder goals. The Goal construct additionally provides a way of classifying relevance in relationship to other goals, using a hierarchy of numerical or alpha numeric notations. An actor may also have a varying priority, reflecting the actors that have more importance within the web application. The meta-model enables relationships between the goals and actors to be defined.

The *Task Construct* builds and refines the actor goals into high level user activities. A *Requirement Construct* is composed of refined goals and tasks in a natural language that can be understood by all those involved in the web application design and implementation. The meta-model does not imply how to capture all requirements with only those considered critical in shaping the user experience listed in the documentation.

Further, the meta-model links requirements to conceptual design specifications using what could be described as agile techniques. "In particular, the following issues are addressed: a) high-level communication goals and user requirements have to be taken carefully into account in the requirements analysis; b) requirements should be tied up coherently with conceptual design of hypermedia specifications; c) Requirements Engineering techniques should be extremely lightweight, intuitive and usable by web analysts; d) the model proposed must show a relative advantage to project managers, requiring little training effort to be adopted and effectively integrated into current practices" (Bolchini, *et al*., 2003).

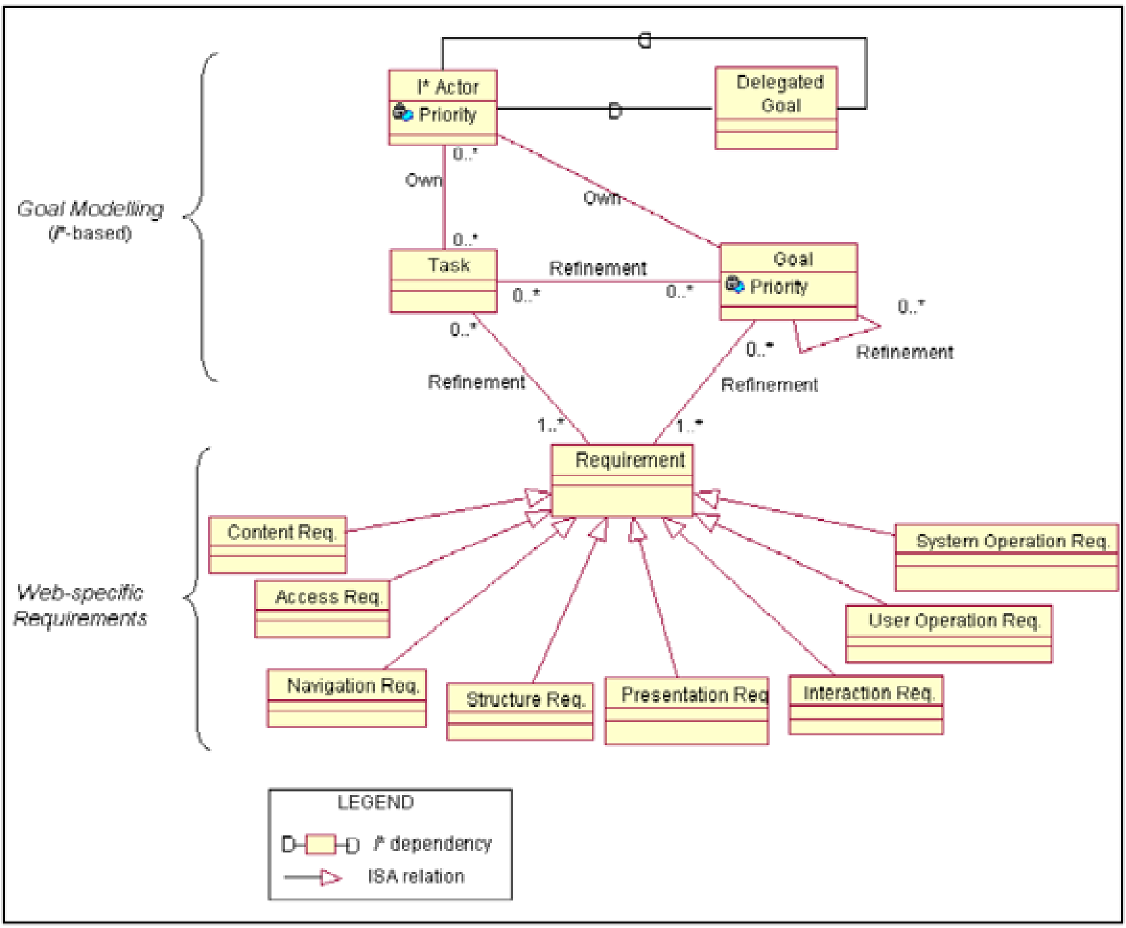


Figure 3.4.1 AWARE meta-model for hypermedia web applications (Bolchini, *et al*., 2003).

Work undertaken by Bolchini and Eric 2004 provides an alternative approach with the identification of the user profile as the starting point of the user requirements analysis. “A user profile describes stable archetypal qualities of a relevant target user segment (Carroll, 2002) and may comprise a variety of attributes based on demographic, for example, *age*, *gender*, *occupation*, *disabilities* etc. or “webographic” eg. *net usage habits*, *interests* and *software constraints*, *favourite sites* etc” (Garrett, 2002). Profiles can be discovered through a variety of requirements elicitation techniques based on user research, such as surveys, contextual inquiry, focus groups and structured interviews. “The user profile can be broken down into four elements; *a person profile*, *a web use profile*, *a context of use profile* and *personalising the user*.

The profile should be as accurate, clear and realistic as possible" (Bolchini *et al*., 2003).

3.4.2 Usability Context Analysis

UCA is described by Macleod 1994, as an informal approach, due to the fact that the process is not represented by any recognised notation or universally accepted diagrams. "Usability Context Analysis (UCA), is a practical cooperative method for identifying and recording contextual aspects of usability in system development or redesign and for helping ensure that usability evaluations reflect the context of use" (Macleod, 1994). UCA was born out of a project run by UK National Physical Laboratory (NPL) and HUSAT Research Institute as part of the MUSiC (Measuring Usability of Systems in Context) toolkit. In the context of research into a suitable approach for web user requirements one of the interesting aspects of this approach is that usability and evaluation are explicit within the process. A mechanism to evaluate what is produced at the end of the project is beneficial as it demonstrates the traceability of requirements to the web design. "There are two prerequisites for any evaluation, if it is to have valid and useful results: the data must be drawn from circumstances which have acceptable ecological validity; and an appropriate method must be applied for analysing the data" (Macleod, 1994). By eliciting, analysing and documenting requirements, students would have a valid set of requirements in which to base their evaluation. The method in which this could be achieved requires further investigation in the research programme.

Usability Context Analysis does involve a high number of steps and the involvement of a number of people in order to successfully complete it. One of the steps within the method is the 'Context Questionnaire and Guidance'. The questionnaire attempts to capture 'User Types' and classifies users as Primary and Secondary. The modelling process is interesting as it allows the production of a profile of the user using the headings: *Skills and Knowledge, Physical Attributes, Mental Attributes, Job Characteristics and Tasks.* The method uses natural language to build the profile which then flows into the evaluation plan in the form of measurable criteria. The evaluation headings include Users, Tasks and Environment (organizational, technical and physical). It is felt that this aspect of UCA warrants further investigation.

UCA provides a well formed structured approach to the requirements elicitation and analysis process. It pays particular attention to the users, classifying them as 'Types' which are then modelled as generic entities for the evaluation process. UCA could be partly defined as a cognitive aid, as it helps in the logical process of linking assumptions based on documented facts, forcing consistency and traceability of the requirements through to the web design. Perhaps the most interesting ideas evident in this method are 'user types' and 'profiling'. This would be something that students may readily identify with, especially as it uses natural language to describe user characteristics.

The requirements classification system used by UCA is based upon *functional* and *non-functional* requirements, along with a meta-model to support design requirements, usability requirements, user interface requirements and features of the system that the user interacts with.

3.4.3 Contextual Enquiry

Bolchini's approach is to develop an in-depth understanding of the target users. One of his methods is *contextual inquiry (CE)*. "Contextual Enquiry is a technique for examining and understanding users and their workplace, tasks, issues and preferences. CE can be used to produce user needs analyses and task analyses. The results of a CE feed directly into the design process" (Gaffney, 2007).

A web user requirements approach aimed at students could adapt CE to form part of a learning exercise, where the tutor acts as the target user and the student asks a number of questions to elicit the required information. The information could then be captured within a *user profile*. "Profiles can be discovered through a variety of requirements elicitation techniques based on user research, such as surveys, *contextual inquiry*, focus groups and structured interviews. The user profile can be broken down into four elements; a *person profile*, a *web use profile*, a *context of use profile* and *personalising the user*. The profile should be as accurate, clear and realistic as possible" (Bolchini and Eric, 2004).

Bolchini highlighted profiling as a vehicle for the way that user requirements are presented. A person profile is perhaps the most interesting aspect from the point of view this research and as way forward for a student centred learning activity. A profile could include the characteristics of the user, including age, gender, motivations, preferences and skills. Having formed a mental model of the user, the student could then think much more deeply about user requirements. In common with the two previous approaches, *contextual inquiry* does not cover the entire requirements process. Most of the activity concentrates on the elicitation stage. In addition, little attention is given to analysis and it is not clear how functional and non-functional requirements are treated.

3.4.4 Usage Scenarios

This method relies on user involvement in the requirements process by eliciting information from them directly. "Usage scenarios and the use cases from which they are derived should describe an application's expected behaviour in multiple business processes. Gathering this information requires sitting down with users and determining every possible scenario for a specific function" (Cornish, *et al.*, 2003). This method works best where the complexities of the application arise from multiple processes which need to be constructed. By writing a usage scenario with the user, the development team can focus much more on the problem.

The scenarios can include steps, events and actions which occur during the business process. This makes the method quite flexible in terms of interfacing with more complex system modelling techniques such as UML or other system development methodologies. "Usage scenarios are applied in several development processes, often in different ways. In derivatives of the Unified Process (UP), such as the Rational Unified Process (RUP), ICONIX and the Agile Unified Process (AUP) they are used to help move from use cases to sequence diagrams. The basic strategy is to identify a path though a use case, or through a portion of a use case and then write the scenario as an instance of that path" (Ambler, 2004). The Usage Scenario process involves writing a conceptual description of what the application must do. This is then translated into the conceptual description in a logical representation of the design, comprising of components in the application, often carried out using 'use cases'.

Usage Scenarios help with an often neglected stage within the requirements process, *elicitation*. It achieves this by gathering information directly from the user and represents this in the design using system modelling techniques. In an educational context, the fact that usage scenarios rely on direct involvement with a user may detract from its usefulness, although again the tutor could *act* as a user for this purpose either in class or by providing information in the assessment documentation. In terms of requirements classification, Usage Scenarios concentrate on functional, rather than non-functional requirements.

3.4.5 Key Ideas That Emerged From The Review (Research Cycle 1)

Having undertaken a review of related work in the first cycle, a number of ideas formed concerning an initial web requirements method aimed to support inexperienced student:

- Students need to define different types of requirements, such as: Navigation; Content; Colour; Structure/Layout; that represents the design aspects of a web design.

- Encourage the Student to Model the User through profiling techniques as the starting point in the requirements discovery process. The user could be defined as a primary or secondary user, that encourages the student to think about the roles that the need to be considered.

- A lightweight method that supports students design decisions. Encourage the student user to make connections between requirements and the design, thereby making the traceability of requirements explicit and assessable in the ICA submissions.

- An enquiry based method may enable the student to envision a rich picture of the project.

- Requirements discovery encourages the student to analyse the problem domain more thorough and to document these in design centred requirements.

The ideas were more thoroughly developed in the first research cycle, see Chapter 6.

3.5 Related Work Cycle 2 (WURF)

3.5.1 Object View and Interaction Design (OVID)

Object View and Interaction Design (OVID) (Roberts*, et al*., 1997) cannot be described as a ‘whole process requirements method’, as it concentrates upon matching *tasks* to *interface objects.* It represents this through class diagrams, but it is not clear how *elicitation* takes place. OVID is an example of a user centred approach, as its main aim is to bridge the gap between the implementers and end users. “In interface design, we call the models in the users heads the ‘user's model’. In our designs, we try to take advantage of how users employ this model. There are also two other models involved in product design, the designer's model and the implementation model. The designer's model is what users are supposed to see when they use the product and the implementation model is what the implementers actually program” (Roberts, *et al*., 1997).

OVID provides a way forward to link tasks to user driven interface objects whilst recognising that there are differences between design and development. However, it is argued that OVID focuses too much on this aspect. “OVID is a method for representing requirements during the analysis and communication phases. It assumes that the acquisition phase has already taken place and indeed some analysis in the form of tasks. The output of the method is an abstract diagram that describes the architecture of the desired design, from the users' point of view. The diagram is used in conjunction with the visual specifications to enable implementation of the final diagram” (Berry, *et al*., 2007).

OVID is focused on the User Interface, something that use case modelling does not specifically address. As a result, the design may progress more quickly with fewer cycles of iteration. This is something that could be tested in the design and evaluation of an approach aimed to support the inexperienced student user. “OVID specifies the interface design in a format and notation well-suited for code design and feeds directly into tools and methods commonly used for programming, thus reducing the risk of introducing interface errors later in development” (Berry, *et al.*, 2007).

OVID uses natural language to describe *interactions*, *tasks* and *objects*, but still draws upon technical approaches in the form of class diagrams. These diagrams can be described as abstract or conceptual as their interpretation is not yet represented in the visual design or behaviours linked to specific interactions. The requirement classification system employed by OVID is based around the term *functional* and *non-functional* requirement.

3.5.2Agile Requirements Method

The Agile Requirements Method (ARM) (Leffingwell and Widrig, 2003) brings together a range of tools in a flexible way. It starts by envisioning a concept for the overall project. A concept is the root of the project, which might be fully understood at the outset, but might be an idea that needs to be explored more fully. This is achieved through workshops or interviews undertaken with end users and other stakeholders. A ‘vision’ for the project is a document that describes the features to be implemented. Requirements are then refined using a use case diagram, with each use case containing its own specification template. The classification system used by the Agile Requirements Method utilises the functional and non-functional definitions. ARM also recognises the need for a management of requirements, but leaves this open in terms of the tools which would be used to support it. Project management is also a feature of this method, reflecting the need to mesh with other processes that the development team might be using.

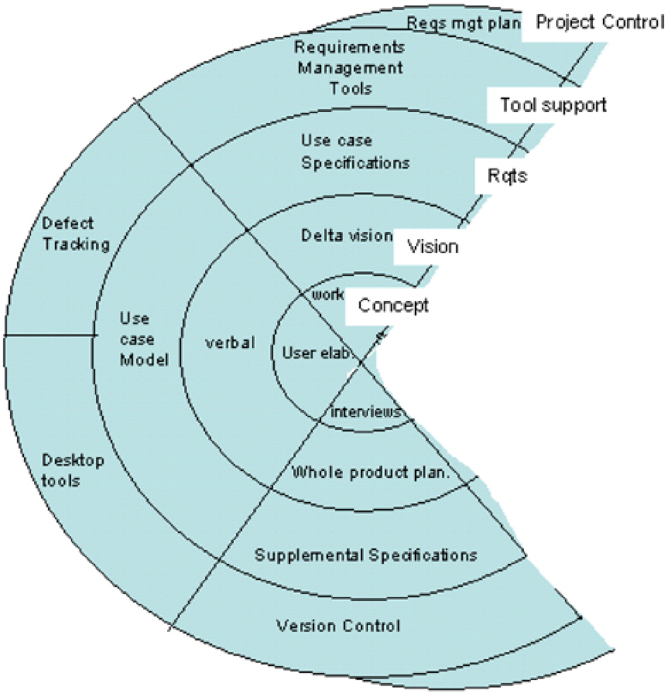


Figure 3.5.2 Agile Requirements Method (ARM) (Leffingwell and Widrig, 2003).

According to Leffingwell and Widrig, the Agile Requirements Method suits projects that are ground breaking in terms of innovation and the tools that are used to control requirements. It could be described as a complex approach, as some stages are explicit in terms of the tooling that can be used to achieve it, but other stages are left open. For example, the vision stage offers no template could be used by the developer. This may elicit advantages and disadvantages if it was adopted in an educational context. Students could pick and choose from a selection of tools that best fit particular projects, but with the underpinning of a process that is documented and repeatable. Some students might find this openness difficult and choose to complete the stages superficially or not at all. Nevertheless, ARM provides some useful ideas that could contribute to the development of a student focused web user requirements approach. In particular, the vision document is used as a starting point and could be useful in the initial stages of an approach, as it enables students to see a richer picture of the web project.

3.5.3 Task Based Audience Segmentation

Thus far this section has focused on three academic approaches. A commercial requirements approach could provide a useful contrast to the approaches discussed earlier. Task Based Audience Segmentation (TBAS) is a commercial approach that helps the developer to think about the user and the tasks they complete in order to achieve a goal. “It is an invaluable foundation for conceptual research, that is to say it can reveal how people think about completing online tasks” (Young, 2007). TBAS cannot be defined as a true requirements method as it is does not include an elicitation stage. Instead it concentrates upon the analysis stage, where it reveals users and their needs by the tasks that they undertake. In terms of requirements classification, both Functional and Non-Functional requirements are absent from this particular method.

In order to carry out TBAS, a number of steps are required as described by Young, 2007:

1. Team Preparation.

2. Brainstorming (tasks, goals and users).

3. Task Grouping (to create user goal sets).

4. Link Users (Audience) to Tasks.

5. Define Audience Segments.

The TBAS method encourages the developer to think about tasks and users and to link a user to specific a task. After an initial brain storming session, groupings are used for tasks which are then matched with users to create a conceptual map. Relationships between users and tasks are then identified and named accordingly, using mainly marketing terminology. This activity may prove useful in an educational setting, where students could undertake brainstorming activities to think about the end users and their tasks. Linking a task to a user would also enhance the traceability of requirements within the overall process.

3.5.4 Agile Microsoft Solutions Framework (MSF) – Requirements Stage

According to Anderson 2005, MSF enables the software development team to follow a proven Software Engineering lifecycle. It has an adaptable meta-model, which contains rules, frames and constraints, that can be adapted to suit the type of application being developed. The need for a flexible approach is due to the divergences in software development, such as agile, rapid and prototyping. "Many software developers are suspicious of process generally. Process often gets in their way and slows the pace of software development to a frustrating level" (Anderson, 2005). This idea is reinforced by Leffingwell and Widrig, 2010, who cite the reason for developers wanting more flexible approaches as "failures in the waterfall model, along with increasing time-to-market pressures and advances in software development tools and technologies" (Leffingwell and Widrig, 2010).

MSP achieves a flexible or agile approach by providing specific process guidance and templates which map onto the adaptive process. These stages include; *Envisioning*; *Planning*; *Developing*; *Stabilising* and *Deploying*. One of the stages within MSF that relates to this research investigation is planning. Within the planning stage, a number of templates enable the production of a detailed overall specification for the project. These help formulate highly detailed requirements and include "Business Requirements; Conceptual Design; Functional Specification; Logical Design; Operations Requirements; Physical Design; System Requirements; Usage Scenarios; and User Requirements" (Anderson, 2005).

What is interesting from the planning stage is the separation of business requirements from user requirements. The planning stage within MSF broadly reflects the requirements process, although excludes requirements management. Usage scenarios are separated from user requirements. In many of the previous approaches, requirements were grouped together as one set including; *functional requirements, tasks, goals, usage scenarios and UI behaviours*. MSF classifies requirements into; *Business Requirements*, *Operations Requirements, System Requirements and User Requirements.*

The flexible meta-model approach embedded within MSF allows for adoption or rejection in any of these specification or requirements templates, essentially allowing the developer to *pick and mix* meta-models to fit the project. In terms of the design of a web user requirements approach aimed at students, this approach has potential. For example, it could allow the student to adopt an appropriate meta-model to fit their project. Where a set of requirements are not applicable, for example, a set of non-functional requirements, they would not need to appear in the final specification.

3.5.5 Use Case Diagrams and Templates

Use Cases are part of UML, which is a notation for modelling software systems. "UML expresses system models and designs in an object-oriented fashion" (Conallen, 2003). Use Case Diagrams are used as starting point in the analysis of requirements within a number of contemporary requirements methods. For example, webRE , from which requirements are expressed in more formal notations including NDT patterns and UWE activity diagrams. "Use cases help to determine the functionality and features of the software from the users perspective" (Escalona and Aragon, 2008).

A Use Case helps to describe the goal of the user by the definition of steps required to achieve it and the use case diagram provides a graphical relationship of all the Use Cases. The graphical aspect is important as it allows the analyst to *visually* depict the main functions to be implemented within the software. Users are represented by the term 'Actor' and are typically represented in the use case diagram by a 'stick figure'. Relationships between the Actor and Use Case are represented by a drawn line, with multiple actors and multiple relationships often illustrated.

Actors can be further defined as having a *primary* or *supporting* (*secondary)* role within the system. "The primary actor of a use case is the stakeholder that calls on the system to deliver one of its services" (Cockburn, 2001: p54). Supporting actors are usually external to the system that is being developed, for example, a web service or hardware such as a printer. The supporting actor is important in a web project, as it possible to interact with multiple external systems. Figure 3.5.5 shows a typical Use Case diagram with associations between actors.

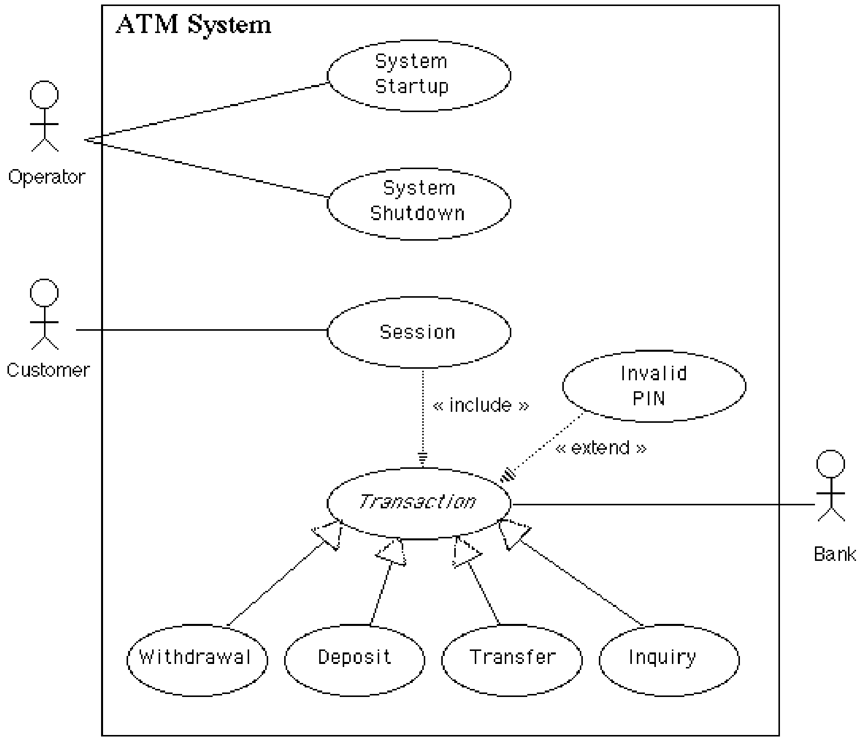


Figure 3.5.5 Example Use Case Diagram.

A *Use Case Template* provides a mechanism to further refine the use case including the flow of events triggered by the function and its pre-conditions. A use case may also be useful for early development or brainstorming of initial system requirements. “It can be used to focus discussion about upcoming software system requirements, but not to be the requirements description” (Cockburn, 2001: p7).

Within a web user requirements approach, Use Cases could provide a way of defining users who will interact with the website. In particular, the way that Use Cases classify users (*primary* and *secondary)* would allow students to model their users and to express their profiles in a universally recognised language.

3.5.6 Navigational Development Techniques (NDT)

NDT has been developed specifically for use in web development in order to capture, analyse and specify web requirements. NDT is the outcome of extensive research at the University of Seville, by Maria Escalona and Gustavo Aragon in 2008. NDT is a method that reflects all stages within the requirements process. It achieves this by offering a strict workflow using ‘Model Driven Web Engineering’ (MDWE). “MDWE proposes representing concepts using meta-models. The development process is supported by a set of transformations and relations between concepts that leads to agile developments and assures consistency between models” (Escalona and Aragon, 2008). Using this workflow the requirements are acquired and then defined according to their nature. This includes a three stage process, with some guidelines and heuristics to help during their production. These stages include:

Stage 1 Requirements Capture (*)

1. Information Storage Requirements.
2. Actors Requirements.
3. Functional Requirements.
4. Interaction Requirements.
5. Non-functional Requirements.

Stage 2 Requirements Definition

1. Content Model (using a class diagram).
2. Navigational Model (represented in a navigation chart).
3. Abstract Interface Model (showing part of the web user interface using prototypes).

Stage 3 Requirements Validation

4. Evaluation of Prototypes.
5. Requirements Specification to be used in Web Methodologies to deal with Design and Implementation.

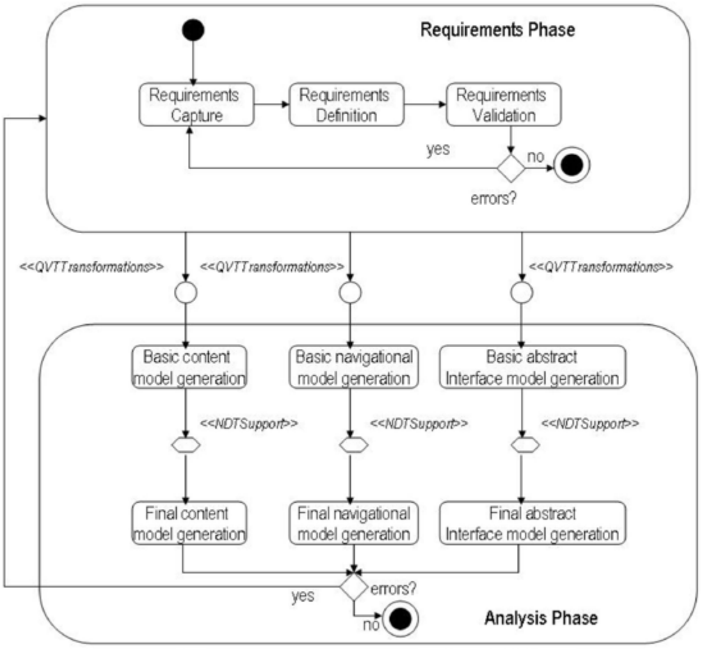


Figure 3.5.6 Navigation Development Techniques Model (Escalona and Aragon, 2008).

It is felt that this method has enormous potential for Web Engineering and for teaching Web Engineering due to the transparency of the stages and its ability to link with a number of existing web methodologies. The method does not add unnecessary complexity and tools used within the method include Use Cases; Navigation Mapping; and Entity Relationship Diagrams. In common with other approaches, the user is expressed as an actor within the web application. The framework deals with this within an NDT package named 'Behaviour'. The *WebActor Class* models in detail their *behaviours*, *attributes* and their *relationships*. Functional requirements are represented by a *WebUseCase* class. Navigational activities are represented by *Browse*, *Phrase* and *Transaction* classes.

In terms of a web user requirements approach, the teaching of *classes* and *class diagrams* would suit many web development modules that deal with databases, as their fundamental concept is an embedded feature of the curricula. That does leave a gap, where students studying web design modules may find it difficult to engage with the notion of classes and diagrams to represent the requirements model. This cohort of students represents a significant overall number on web modules within the HEI.

3.5.7 Joint Application Development (JAD)

Joint Application Design (JAD) was written and developed by Drake, Josh and Crawford of IBM. The principle idea is to bring together developers and users together in a productive and creative setting. JAD overlaps RE and SE, where both requirements and analysis are undertaken. Nevertheless it is still felt that there are some stages within JAD that could inform the structure of a web user requirements method. "JAD is a method whereby system stakeholders work together in facilitated group sessions to specify and perform preliminary development (Requirements Engineering and analysis) of a system. JAD sessions include representatives in the following roles:

session leader (facilitators), user representative, specialist, analyst, information systems representative, executive sponsor" (August 1991).

A JAD session starts with the *business vision* and moves onto the definition of *high level requirements* and the setting of *business objectives*. It attempts to capture both functional and non-function requirements in the same session, including security requirements and constraints. It also supports the documenting of requirements in addition to analysis and modelling. JAD relies heavily on facilitation and stakeholder involvement and when used on projects can take days to complete. "Knowledge workers and IT specialists meet, sometimes for several days, to define and review the business requirements for the system" (Haag, *et al*., 2006).

Setting an overall vision for the project, along with measurable business objectives is an interesting aspect of JAD and again this is something that could prove beneficial to a web user requirements method. Students would be able to see a richer picture of the project and would be able to think about what the website hopes to achieve by expressing its business objectives.

3.5.8 Cooperative Requirements Capture (CRC)

The principle concept behind CRC is that mathematical and technological notations are insufficient in reflecting user requirements. According to Macaulay, the social element is "explicitly managed through use of a human facilitator and which provides a structured approach to the management of the requirements capture task. The process comprises seven stages; 1.The business case; 2.Workgroups; 3.Users; 4.Objects; 5.Tasks; 6.Interactions; 7.Consolidation" (Macaulay, 1993).

The process encourages the production of documentation in each stage although it does not provide a template for this. The process uses the idea of *user tasks* to help the stakeholders think about requirements. A hierarchy of tasks are produced and their interactions associated to specific users. The consolidation stage allows stakeholders to revisit the business case and reassess the credibility of the information gathered.

However, it is not clear what happens after these stages have been completed. For example, it offers little advice on the analysis and specification stages. CRC is therefore focused on the elicitation stage. CRC could be useful in helping define the elicitation stage for web user requirements. For example, the way that CRC focuses on ‘cooperative interaction’ between stakeholders in drawing out requirements may prove useful. Obviously this would need to be simulated by the tutor, rather than interacting with the users themselves.

3.5.9 User Requirements Notation (URN)

URN is supported and ratified as an international standard by the International Telecommunications Union, a United Nations agency for information and communication technology issues. This is a body that has responsibility for co-ordinating the global use of the radio spectrum and for assigning satellite orbits and so is very well recognised in various fields. The fact that URN is now a ITU standard underlines the importance of work into requirements and the way in which it should be approached.

URN is able to model functional and non-functional requirements by adopting the Goal-Requirements Language (GRL) to model tasks and procedures. It achieves this by the use of scenarios to draw out ‘high level’ functional and non-functional requirements. “The URN is a two-headed proposal. URN-NFR addresses non-functional requirements (NFRs), capturing them using the Goal-Requirements Language (GRL).Such a model aims at highlighting how some facets of a system (e.g., tasks, procedures) contribute (positively or negatively) to the satisfaction of NFRs” (Arnold, *et al*., 2010).

As discussed in section 2.2.1, high level requirements are generalised descriptions that are written in a natural language and can be understood by all stakeholders involved in the project. In URN these are then transformed and organised into textual 'use cases'. "URN combines modelling concepts and notations for goals (mainly for non-functional requirements and quality attributes) and scenarios (mainly for operational requirements, functional requirements and performance and architectural reasoning). The goal sub-notation is called Goal-oriented Requirements Language (GRL) and the scenario sub notation is called Use Case Map (UCM)" (ITU-T, 2008).

Within URN, a 'topmost' meta-model describes the attributes, relationships and constraints between the models (see Figure 3.5.9).

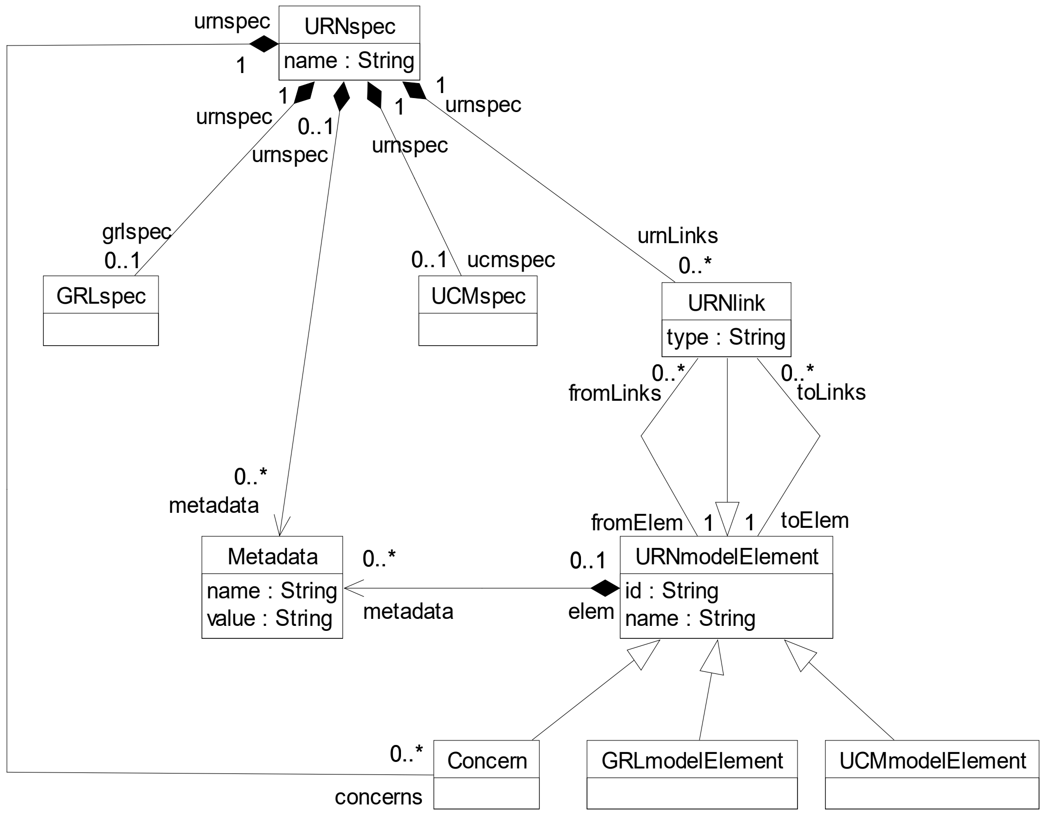


Figure 3.5.9 URN 'Topmost' Meta-model.

In addition, the GRL meta-class allows the developer to model the goals and intentions of the stakeholders, with the latter modelled as actors. URN would be difficult for students to adopt and use for a number of reasons. URN is quite a complex modelling approach, using a language that design students would not be familiar with. However, this would bring an additional overhead in terms of learning for students before they could use it. It would also require an overhead in terms of time and resources in order to support it. It cannot be therefore described as a 'light weight' approach that matches the agile requirements method that is needed to support the inexperienced student user. Whilst this is true, there are some useful aspects within URN that could be explored further. This includes the ability to profile the characteristics of the users (actors); the way that it is able to model tasks and the association of tasks to actors, represented within the meta-model.

3.5.10 SOARE Approach

Bleistein, *et al.*, 2004 have also investigated web requirements and produced an approach named 'Strategy-Oriented Alignment in Requirements Engineering' (SOARE). This provides a basis for defining 'e-business web requirements'. SOARE is aimed at e-business projects that are both web and non-web based. It focuses on business objectives and the 'real world goals of the system'. SOARE starts with the identification of business objectives as a means of decomposing requirements. "The SOARE approach incorporates means for analysing and decomposing business strategy, employing goal modelling both to represent business strategy in a Requirements Engineering context and to link high-level strategic objectives to low-level requirements through goal refinement" (Bleistein, *et al.*, 2004). It also provides traceable links between business objectives and goals, making these measurable within the end product. Business strategies are expressed in a natural language and derived from business plans, annual reports, stakeholder interviews and executive reports. This approach is very different from AWARE, as it is business focused and uses language that requires some background understanding of business strategy and planning. SOARE offers a model for the requirements process, as illustrated in Figure 3.5.10.

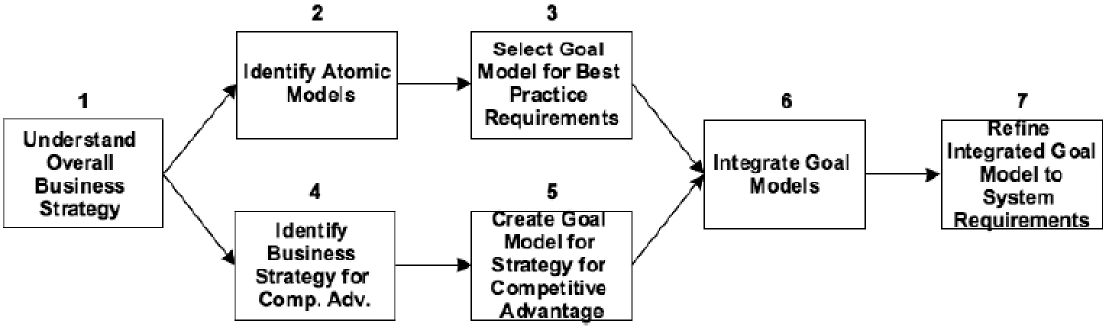

1
Understand Overall Business Strategy
2
Identify Atomic Models
3
Select Goal Model for Best Practice Requirements
4
Identify Business Strategy for Comp. Adv.
5
Create Goal Model for Strategy for Competitive Advantage
6
Integrate Goal Models
7
Refine Integrated Goal Model to System Requirements

Figure 3.5.10 SOARE Process (Bleistein, *et al.*, 2004).

3.5.11 SSM/ICDT Approach

Meldrum and Rose, 2004, argue that “there is a need for an approach to requirements generation for web systems that combines the recognition of multiple user views of a complex human activity system with techniques to help creatively map existing and potential business functions to a Web-based environment” (Meldrum and Rose, 2004). This view supports the idea that an approach must be accessible to those that have no IT background and who do not have an understanding of formal notations to express requirements, or other I.T specific terminology. It employs Soft Systems Methodology (SSM) to deliver various meta-models that demonstrates how describe business processes relate to human activity within the web application. Ramesh, *et al.*, 2002 in Meldrum and Rose, 2004, identifies the difference in characteristics between software and web development. “These can include time pressure, vague requirements, a prototyping orientation, frequent releases and evolutionary development, parallel development and an emphasis on small teams of highly competent programmers coding their way out of problems” (Meldrum and Rose, 2004).

The process model, see Figure 3.5.11, starts with the analysis and expression a problem faced by the developer. “Using rich pictures and root definitions and conceptual models of relevant human activity Systems” (Meldrum and Rose, 2004).

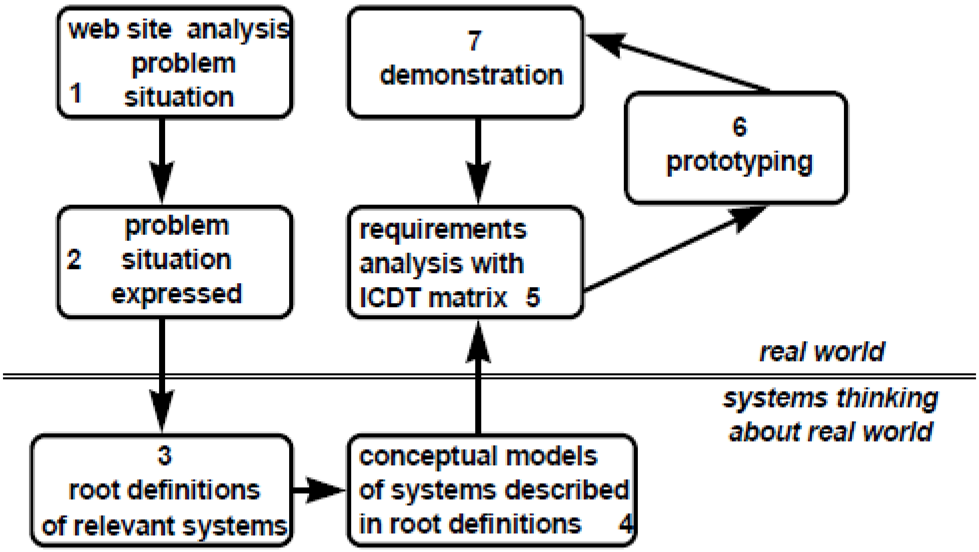
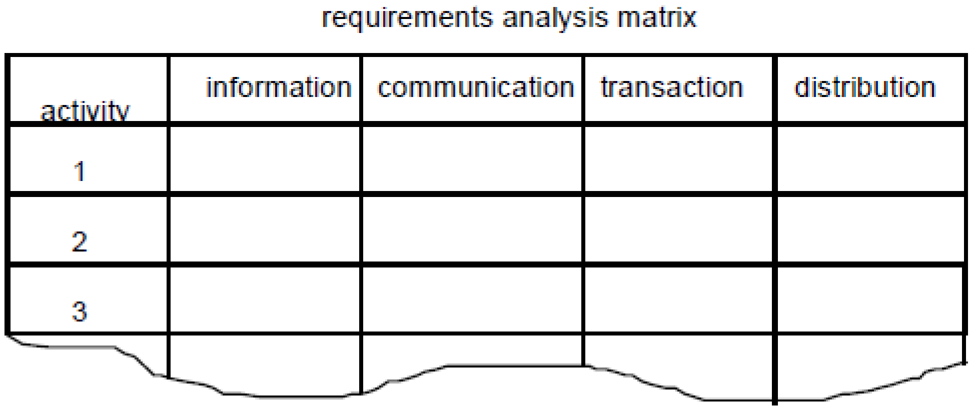

web site analysis problem situation 1
7 demonstration
6 prototyping
problem situation expressed 2
requirements analysis with ICDT matrix 5
real world
systems thinking about real world
3 root definitions of relevant systems
conceptual models of systems described in root definitions 4

Figure 3.5.11 Web-site development using SSM, the ICDT model and prototyping (Meldrum and Rose, 2004).

The activities are *mapped* onto a requirements analysis matrix and categorised into *information*, *communication*, *transaction* and *distribution* (see Figure 3.5.12). “Each activity is then analysed using the question ‘How could a Website support the information/communication /distribution /transaction potential for the activity? The answers can be mapped onto the matrix resulting in a creative set of potential ideas. These can then be prioritised (in terms of feasibility and desirability) in discussion with users to arrive at an agreed set of requirements. This is followed by a prototyping development strategy” (Meldrum and Rose, 2004: p195).



requirements analysis matrix

activity	information	communication	transaction	distribution
1				
2				
3				

Figure 3.5.12Requirements generation matrix using ICDT model (Meldrum and Rose, 2004: p143).

3.5.12 Key Ideas That Emerged From The Review (Research Cycle 2)

The second review was undertaken in the early stages of research cycle 2 in order to address weaknesses identified in the evaluation of the experimental method. The context for the review was informed by the evaluation of the first research cycle. A number of ideas emerged that informed the re-working of the method into a more fully developed framework:

- More emphasis on the 'discovery phase' by focusing more on the vision behind the website, its objectives and the tasks that the user will perform. This replaced the 'user profiling' aspect of the first method, which proved to be confusing for the student.

- In order to address the main weakness of the experimental method, which was the 'closed taxonomy for requirements', it was thought that it would be more appropriate to break requirements into functional and non-functional. Functional requirements would reflect the web project in terms of tasks and features to be built, in which the student would better relate to. Dynamic 'server-side' requirements could also be considered in terms of functional requirements, as it is possible to describe their interactions and behaviours. Design type requirements could still be represented in non-functional requirements, where these impose constrains on the web design and implementation environment.

- A number of key terms cited in the method were changed in order to help the student. The term ‘user’ was dropped in favour of ‘actor’ in order to better represent both human and non-human agents who interact with the website. A change to the meta-model in this regard makes it possible to model more than two actors, again something that was identified as a weakness in the first method.
- The addition of an association model. To make explicit that there is a link between actors, tasks and functional requirements. A direct relationship between the requirements and traceability to the website could then be achievable.

3.6 Related Work Cycle 3 (WURF / CAWE)

3.6.1 Computer Aided Web Engineering

The first method was deployed to the student in a paper based format, where this was completed and submitted along with the students ICA. In the second cycle of research, it was deemed advantageous to use an electronic means of deploying the document, whilst also supporting the student with examples in use. These were packaged for distribution within the institutions virtual learning environment (VLE). In the final research cycle the problem of deployment was once again considered with a premise of integrating more support mechanisms. One of the options for deployment was to re-engineering WURF into a computer based program in which the students could access continually throughout their web project. This approach is a well established in various fields of computer science, where they are referred to as Computer Aided Software Engineering (CASE) tools.

CASE tools can support a variety of stages within the software development lifecycle. A similar acronym can be found in the literature for Web Engineering, although the frequency of citation is much lower than that of CASE. Computer Aided Web Engineering (CAWE) is used to describe tools that help support one or more stages within the web development lifecycle, such as translating designs into code.

The approach offers solutions raised in the evaluation of the second research cycle, such as the *consistency, completeness and correctness* of requirements that the students are producing. It was found that students were able to produce incomplete requirements documentation in their ICA's. By adopting a CAWE tool approach, a strict rules model could be enforced thereby ensuring students submit complete requirements. It also offers a way to automate the associations between actors, tasks and functional requirements in order to enhance the traceability of requirements through to the end website.

3.6.2 WebRatio

Casteleyn, *et al*., 2009, describe a number of CAWE based tools such as WebRatio and VisualWade that support model driven web development. WebRatio provides support for five main web development areas *Data Design, Hypertext Design, Data Mapping, Presentation Design and Code Generation* (Casteleyn, *et al*., 2009, p232) (see Figure 3.6.2).

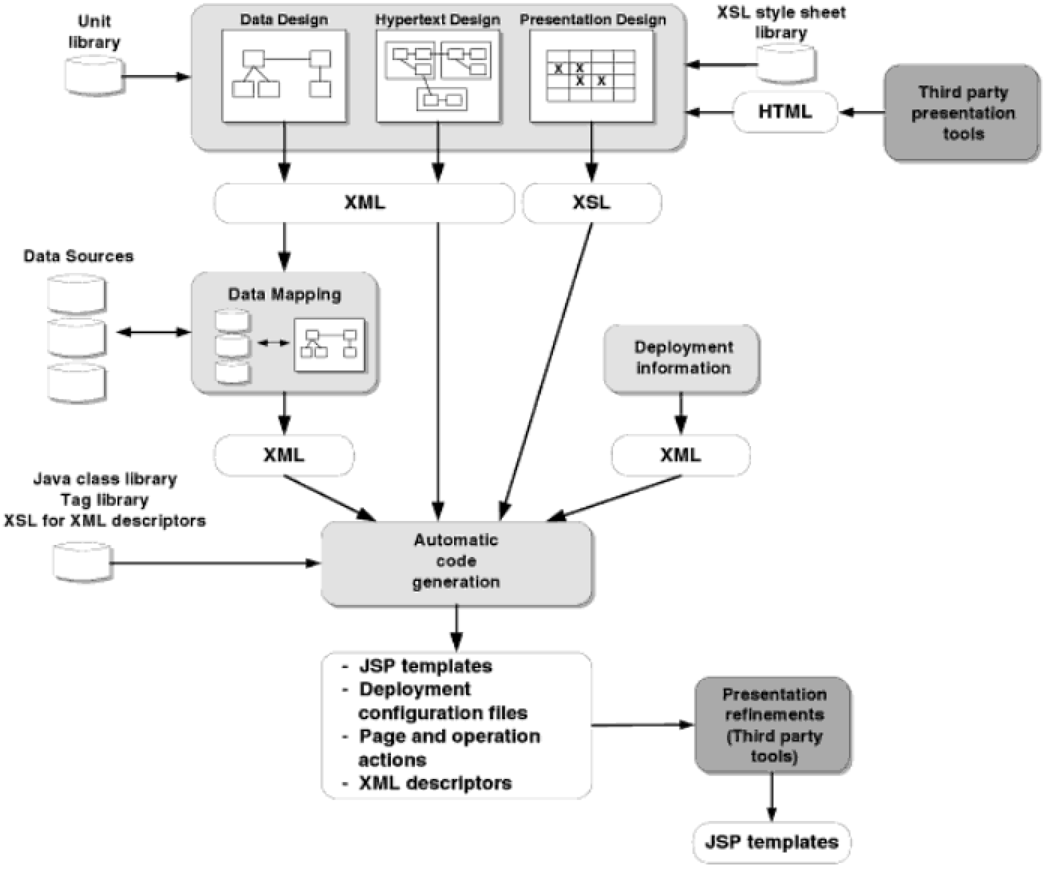


Figure 3.6.2 The WebRatio Development Process (Casteleyn, *et al*., 2009: p233).

The goal of WebRatio is to automate code generation by modelling the *data*, *logic* and *presentational* design phases. It is also able to produce 'java server pages'(JSP) templates that can be refined further in order to modify presentational designs or data interactions. WebRatio supports rapid prototyping, thus shortening the overall lifecycle, which is useful in web projects due to the short time to market expectations. Unlike other prototyping tools, where mock web pages have to be recoded, WebRatio generates code that can be used in the production version of the website.

3.6.3 FlashWeb

FlashWeb, (see Figure 3.6.3), is a CAWE tool that focuses on data management as a mechanism to model a web application. “The FashWeb CAWE tool supports the model-driven development of web applications that provide advanced data management functionality. It utilises graphical models throughout the entire development process. Different aspects of the web application are captured with different models assuring a clear separation of concerns” (Jakob, *et al.*, 2007).

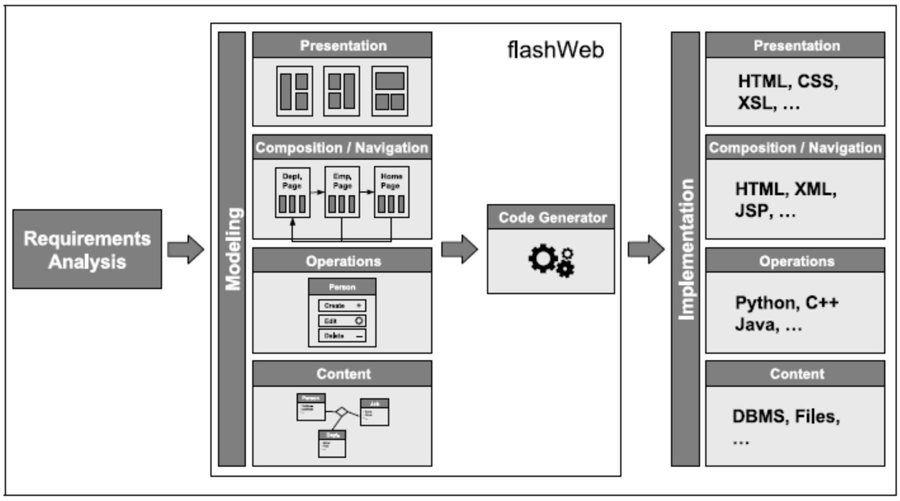


Figure 3.6.3 FlashWeb Development Process (Jakob, *et al.*, 2007).

FlashWeb provides a good example of how a CAWE tool can provide support to web developers by helping them to model various aspects of the web application, before implementation commences. It shares some similarities with WebRatio, such as the ability to model aspects of the web application including a graphical representation, which then informs the code generation process. FlashWeb aims to provide a rapid development environment, but still focuses on the modelling of data, presentation and interactions. However, it does not consider requirements first, although the process model of FlashWeb does acknowledge requirements analysis.

WebRatio, FlashWeb and other related tools tend to focus on implementation, rather than the early phases of the web development lifecycle. "*There are many tools available to support the building of web sites, but few that support their planning or design*" (Griffiths, *et al*., 2004).

3.6.4 VOLERE Requirements Template

An opportunity to use the CAWE tool to enhance the SRS exists via production of an automated SRS document. This could be achieved by using the stored user data to populate an SRS template, after consistencies and completeness checking had been completed by the rules model. Requirements could also be attributed with a unique identification number for tracking purposes. A summary of the Statement of Purpose, Web Objectives, Tasks and Web Actors could also be included within the SRS. By standardising the specification, it is also envisaged that the traceability between the requirements and website implementation would become more apparent during the marking and feedback process, and for validation purposes.

It was found that a number of requirements specification templates existed, including a number of commercial offerings. For example, VOLERE (Robertson and Robertson, 2010) is a commercial specification template that comprises the following elements:

1. Project Drivers.
2. Project Constraints.
3. Functional Requirements (FR).
4. Non Functional Requirements (NFR).
5. Project Issues.

(Robertson and Robertson, 2010)

The interesting point about the requirement template is that it offers a standard way of setting out Functional Requirements (see Figure 6.2.4). The designers refer to this as the ‘requirement shell’, which acts as a template for requirements.

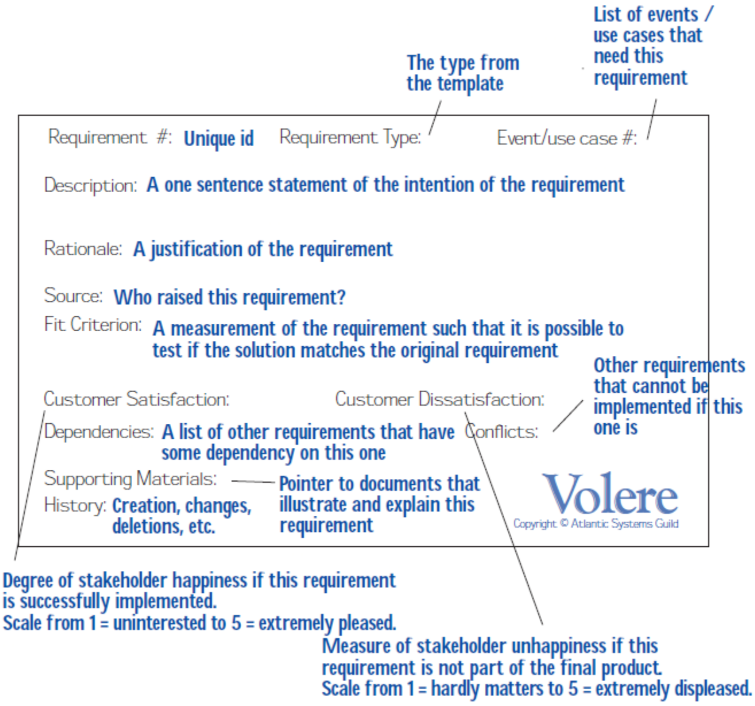

List of events / use cases that need this requirement
The type from the template
Requirement #: Unique id
Requirement Type:
Event/use case #:
Description: A one sentence statement of the intention of the requirement
Rationale: A justification of the requirement
Source: Who raised this requirement?
Fit Criterion: A measurement of the requirement such that it is possible to test if the solution matches the original requirement
Other requirements that cannot be implemented if this one is
Customer Satisfaction:
Customer Dissatisfaction:
Dependencies: A list of other requirements that have some dependency on this one
Conflicts:
Supporting Materials:
Pointer to documents that illustrate and explain this requirement
History: Creation, changes, deletions, etc.
Volere
Copyright © Atlantic Systems Guild
Degree of stakeholder happiness if this requirement is successfully implemented.
Scale from 1 = uninterested to 5 = extremely pleased.
Measure of stakeholder unhappiness if this requirement is not part of the final product.
Scale from 1 = hardly matters to 5 = extremely displeased.

Figure 3.6.4 VOLERE SRS Requirement Template (Robertson and Robertson, 2010).

VOLERE was not found to be an automated tool, but offered a way forward for a requirements template to be expressed within WURF. In particular, as the VOLERE template allowed for the standardisation of information, this would have benefits not only for the student user, but also the facilitator, in their marking and feedback process.

3.6.5 SRS Requirements Template

Another example of an SRS document discovered during this research was one in use by a major United Kingdom public sector organisation (see

Appendix B1.1). This document did not have a classification for a functional requirement and merely lists requirements in alphabetical order, using written natural language statements, along with references to additional resources. The SRS did include a version control mechanism using date tracking. This would suit a situation where the requirements are being constantly updated in a team project, but is nonetheless a valuable feature of the SRS. It also includes a mission statement and a project description as an executive summary.

In the two SRS examples reviewed, it would be seen that differences in approach occur, perhaps relating to specific organisational needs. Both reference requirements, either as features, or by a singular reference 'requirements'. Templates can consist of simple text and bulleted lists or more complex layouts that contain the information. Some specifications also included additional information and links to resources such as a screen shot of a high fidelity prototype.

Although the term 'SRS' is software centric, it was expected that some of the examples addressed in this section could be adapted for use within WURF. In particular, an executive summary that includes the Statement of Purpose, Web Objectives, Tasks and Web Actors, would have provided a useful overview of the project to a third party. A more formal template for functional and non-functional requirements would then follow, each with a unique identifier to facilitate tracking and traceability.

Consistency, completeness and correctness are all established and well understood topics within RE and are often cited as the 'three Cs' of requirements in the literature. "Consistency requires that no two or more requirements in a specification contradict each other" (Zowghi and Gervasi, 2003). Correctness relates to the needs of the user, their tasks or business objectives and the corresponding requirement in much the same way as being able to trace requirements to the physical design of a website. Completeness reflects the need to include all requirements within the SRS and the necessity that all the information contained within the requirements statements is complete. “To be considered complete, the requirements document must exhibit three fundamental characteristics: (1) No information is left unstated or ‘to be determined’, (2) The information does not contain any undefined objects or entities, (3) No information is missing from this document” (Boehm, 1984).

The use of natural language in the definition of requirements makes validating or proving that requirements are consistent, correct and complete difficult if not impossible. “To perform consistency, completeness and correctness checking effectively and to be able to automate this process (in order to assist the requirements engineers in some of their more difficult and mundane tasks), the specification has to be expressed in a formal notation. This is because computer-based analysis requires an explicit formal semantics which provides the basis for the algorithms that carry out the analysis. This is precisely the approach that has been taken by proponents of formal methods in RE. Indeed much of the RE research effort over the last three decades has been concentrated on developing new formal requirements specification languages so that tasks such as syntax correctness, reasoning about requirements and checking their consistency can be automated in ways that are similar to how programs are compiled and managed” (Zowghi and Gervasi, 2004).

3.6.6 Key Ideas That Emerged From The Review (Research Cycle 3)

- Adopt a CAWE tool to represent the WURF process meta-model in order to resolve problems with the document based approach used in the second cycle.
- Represent completeness of requirements in a student dashboard within the CAWE tool where usage data could be captured and represented within a student dashboard. Their learning behaviour could also be tracked and analysed within the evaluation stage of the third research cycle.
- Incorporate a rules model to enforce the completeness of requirements and allow automated production of the SRS document. Consistency and correctness checking could also be performed as the student progresses through the requirements production process.
- Automated production of functional requirements including a visual representation of their relationships.
- Further address student support needs by providing guidance and examples in use within each screen within the CAWE tool. Non-textual support could also be incorporated, such as screen casts and audio feedback.

3.7 Relationship of Review to Research Cycles

Having reviewed a range of existing approaches a number of *conceptual ideas* emerged that provided the starting point for the design of a web user requirements approach:

3.7.1 User Modelling. *Define the target audience using an appropriate model that reflects their importance using profiling and classification models.*

- Primary and Secondary User Classification (UCA)

- User Profile (UCA), Person Profile (CI), Usage Scenarios / Persona (US), Actors (NDT) and Actors (UC)

3.7.2 Project Vision and Objectives. *Allows the developer to establish an overall vision and business objectives before defining functional/non-functional requirements.*

- Concept Vision Document (ARM), Business Vision (JAD) and Business Case (CRC)
- Business Requirements (MSF), Business Objectives (JAD)
- Requirements Generation (SSM/ICDT)

3.7.3 Task and Goal Association Model. *Describe what the users do within the web/software application by the Tasks they complete or by the Goals they want to achieve. An association model links these with specific users.*

- Task to Interface Object Association Model (OVID), and Tasks (CRC)(TBAS)(UC)

3.7.4 Computer Aided Web Engineering (CAWE). *Automation of a rules model. Compel the student to complete every aspect of the meta-model. Check correctness of associations and consistency of requirements. Conformance to the rules model represented in the student dashboard, with visual cues to indicate completeness of the process.*

- WebRatio and FlashWeb in the way it supports the developer to model aspect of the website before implementation commences.

The ‘electronic Web User Requirements Framework’ embodies work undertaken in three research cycles, where ideas evolved and changes were made in response to in class observation. Feedback from the students via in module surveys and also indirectly from delegates at the conferences that were attended all paid an important role in shaping the method and overall frameworks. It must be emphasised that research into the related work was undertaken across a period of time, as demonstrated in Chapter 1, section 1.7. Figure 3.8.1 shows how the review of related work maps onto the three research cycles.

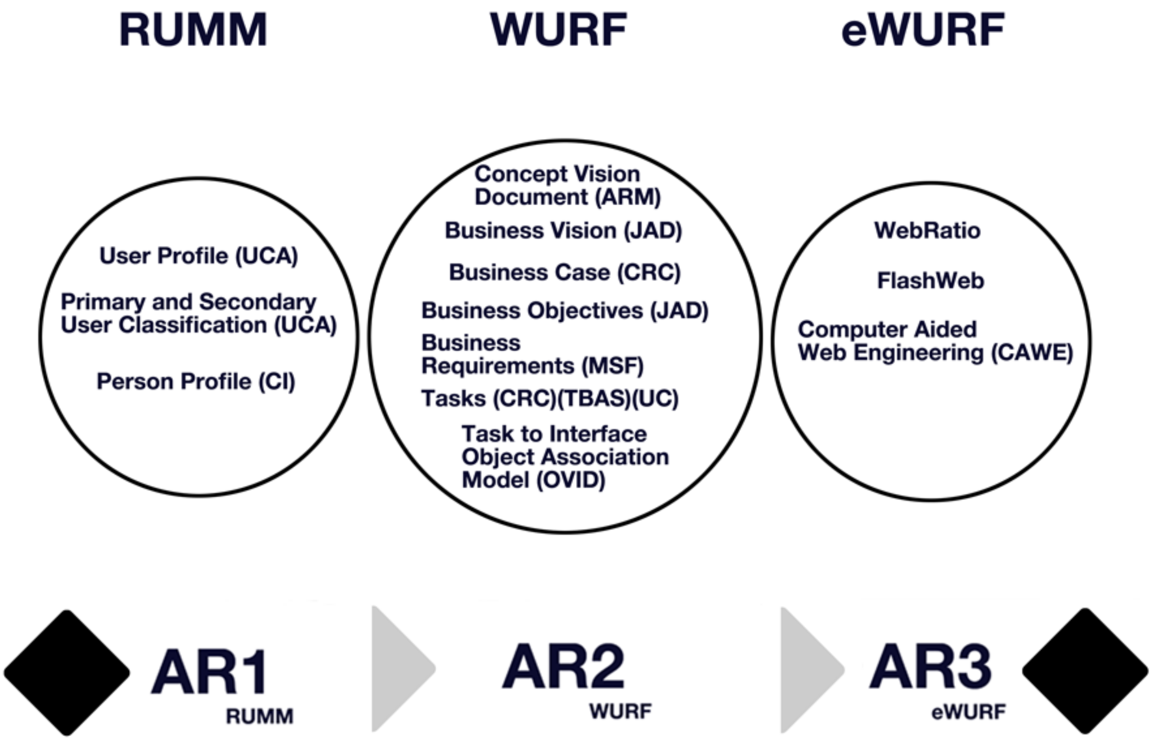

RUMM
WURF
eWURF
User Profile (UCA)
Primary and Secondary User Classification (UCA)
Person Profile (CI)
Concept Vision Document (ARM)
Business Vision (JAD)
Business Case (CRC)
Business Objectives (JAD)
Business Requirements (MSF)
Tasks (CRC)(TBAS)(UC)
Task to Interface Object Association Model (OVID)
WebRatio
FlashWeb
Computer Aided Web Engineering (CAWE)
AR1
RUMM
AR2
WURF
AR3
eWURF

Figure 3.8.1 How review of related work maps to the research cycles.

A number of existing requirements process, methods and tools have been presented in this review. It is clear that from the review that: ARM; US; AMSF; and NDT address the whole requirements process, as defined by Sommerville 2007 (see Volume, Chapter 2, section 2.2.1). Five methods: NDT, AWARE, URN, SOARE and SSM/ICDT are aimed at projects that involve Web Engineering. These are characterised by a modelling technique that enables the web developer to draw out requirements based on the business vision, objectives, the user tasks or goals and before proceeding to define both functional and non-functional requirements.

JAD, AMSF, CRC and SOARE require the development team to draw out business objectives and to link these with interactions, tasks or functional requirements. Some use a modelling approach in eliciting and defining the requirements, such as URN, AWARE and NDT. Many approaches define the user of the system as an ‘Actor’, with various methods employed to show the importance of the Actor within the system.

In the context of a web user requirements approach aimed to support the student, it is felt that techniques that use field research, which involves the collection of primary data, would result in poor adoption and usage. This is due to students having little in the way of resources (time, budget and networks) in order to realistically achieve this. One way around this is to write a briefing document, for example, as part of the ICA that contains all the key information needed to start the elicitation process. The tutor could act as the client and user within a simulation exercise within the laboratory.

The student would therefore still proceed with elicitation, without having to construct questionnaires and carry out a survey. It was found that researchers had already debated the safety of generating requirements based on “intelligent guess work” (Cato, 2001 and Szekely, 1994). This was interesting, as any approach developed from this research programme will involve making informed decisions regarding the users. Their decisions would be informed by evidence found within the ICA briefing document and as well as the tutor acting as a user. Research suggests that it is plausible to generate requirements without directly questioning the target user. “Collect facts if you have them, or make reasonable guesses because even a reasonable guess provides a focus" (Cato, 2001). Cato encourages the use of this approach where it would prove impossible or difficult to undertake surveys to elicit requirements from the user. A technique defined by Szekely as 'fast prototyping', may also provide a way forward. "This approach facilitates elicitation, validation and revision through *discovery* of requirements. The discovery stage involves the production of a small scale version of a complicated system in order to acquire critical knowledge required to build a full system" (Szekely, 1994). This aligns with the Agile Approach to web development, involving iterative cycles of implementation and testing until the application is fit for release. Iterative development is widely used by students in the development process and is one which students would readily identify with. This method relies on revision or iterations which may prove valuable and align with the students' development practice.

Many of the existing approaches had gaps in the treatment of user requirements, notably OVID, TBAS and CI. Web specific requirements approaches such as MSF and NDT did meet the criteria, as did ARM and US. Approaches outlined in Table 3.3 can be further characterised by:

6. Stakeholder involvement in the requirements elicitation process.
7. Detailed descriptions or profiles of the user often referred to as Actors.
8. The separation of functional and non-functional requirements.
9. Use of a meta-model to help define associations and dependencies.
10. Use of natural language to describe 'user journeys' or 'scenarios' or to map business objectives with tasks, features, behaviours and goals.

3.8 Summary

This section has reviewed, analysed and benchmarked a set of methods against key criteria developed in section 3.3. Their suitability for adoption in part, or as a whole, was also discussed. A number of ideas emerged from the review and analysis that could provide the basis of a web user requirements method to support the inexperienced student user. Mechanisms in which to resolve the 'elicitation' problem, where students do not have the resources to question the users directly were explored. Using simulations, briefing documents and tutors acting as the clients were all cited as a way forward.

Much of the literature on existing requirements approaches focuses on 'software' rather than 'web', pointing to a gap in knowledge in this area. RE does provide a number of important principles which need to be taken forward when thinking about the design of a web user requirements process. These include the transparency of the requirements process, where this should be logical and understood by the student together with the ability to produce valid requirements. It should also reflect agile development methods adopted by the student, including the ability to refine and append additional requirements throughout the web project period. It was argued that web requirements are distinct and require an alternative treatment within the web user requirements method to that of software focused methods.

Appendix A3.1 - SIGSAND Conference Paper

A Framework For User Characteristics Capture: An Evaluation Of Using RUMM In The Teaching Of Web Design

Andrew Bingham

School Of Computing, University Of Teesside

a.p.bingham@tees.ac.uk

Gary Griffiths

School Of Computing, University Of Teesside

g.griffiths@tees.ac.uk

Briony Oates

School Of Computing, University Of Teesside

b.j.oates@tees.ac.uk

Julie Turnell

School Of Computing, University Of Teesside

j.d.turnell@tees.ac.uk

ABSTRACT

Students need to capture user profiles to ensure their web design projects are focused on the needs of the users. This paper outlines existing methods and proposes a new framework targeted at students on web design modules. The framework developed is named 'Rapid User Modelling Method' (RUMM), and acts as a cognitive aid to help students think about the target users and implications for the interface design. The framework was tested and evaluated on a group of students, who were shown to benefit from its use. In addition, the students indicated that the framework could be enhanced and developed further. The findings support the view that a method for thinking about *who* the user is and *their* requirements, can be successfully used in the teaching of web design.

Keywords

User Characteristics, User Analysis, Evaluation, Method-In-Action, Persona, Cognitive Aid, Web Design.

INTRODUCTION

Over the last decade web engineering has matured in terms of its approach to formalised development, starting with the adoption of information systems methodologies and progressively adopting its own methods and tools. Whilst this has led to better quality of web applications, it is believed that there is still room for improvement. The web is still in a state of massive change, with new web standards and technology pushing the need for continual professional development in web engineering. In addition there has been a paradigm shift in defining the web engineering discipline. "Web engineering is a holistic approach, and it deals with all aspects of web based systems development, starting from conception and development to implementation, performance evaluation and continual maintenance.” (Ginge And Murugesan 2001).

Within the institution a number of courses provide an opportunity for students to engage in the development of web applications using a variety of development methodologies. These include the ‘Simple Web Method’ (Lockyer, Griffiths, Hebbron, Oates, 2003) that supports the web engineering process, through a defined process of stages. Although this provides the student with a distinct process, it was found that students were not considering the user in their designs. The need to address issues of accessibility and usability within the development process has become essential, with most web development now becoming much more user focused. To address these issues, research was undertaken to support students in their analysis of the users and with a focus on interface design requirements.

This paper describes the importance of understanding the users before any design work begins, outlines existing methods for user analysis and introduces a new cognitive aid to help the student think about the user in a structured and repeatable framework. The framework was tested by 76 students during 2005/6. Students were asked to complete a survey regarding its use within the context of their projects. A proposal for further development, based upon the feedback from the student survey, is provided.

The Need For A Method To Record User Characteristics

Accessibility and the need to enhance usability within web applications has had an impact in web engineering, calling for a change in the requirements gathering phase, implementation approach and evaluation to ensure conformance to a set of pre-defined requirements. This has led to a shift towards an iterative process in web projects, where requirements conformance is measured at each stage. In addition, web projects are characterised by short development times and continual maintenance, as highlighted by Ginige and Murugesan, “Web based system development is not a one time event, as practiced by many; it’s a process with a long lifecycle.” This reinforces the notion of the web application being an evolving product, with the need for web methodologies to adopt the same approach.

The need to capture user characteristics and requirements is becoming an essential aspect of effective web engineering. A major problem that occurs in Web Engineering projects is that the users get to know how to express their requirements very late in the process, i.e. after the design artefacts appeared. (Lowe and Eklund, 2002) To address these issues, there has been a move towards more user centered methods that have a user requirements stage explicitly integrated into the process. It is recognised that while this does offer a more user focused product, the ‘user centered design’ ethos needs to go beyond this. “The need for a systematic approach to capture user navigational requirements has some merit and perhaps some urgency.” (Barry and Lang, 2001)

User analysis for a commercial web project can be undertaken using a variety of techniques, including questionnaires and focus groups. The data could then be used to influence subsequent stages in the development process. Within the context of an educational web project, questionnaires are often an unrealistic proposition for students due to cost and time implications. Whilst this constraint is a major barrier to conducting user analysis, it was still felt that a tool or framework was still required to enable the student to think about users characteristics and produce a set of guidelines or requirements for the design of a web interface.

Before any user requirements can be documented, a method to capture their requirements must be used. Existing approaches include;

1. Concur Task Trees. Allows the envisioning user interaction with the application and derive information, navigation and presentation design accordingly. (Bolchini, Mylopoulos, 2003)

2. Task Based Audience Segmentation. A technique that defines the target audience by the tasks they perform to achieve a goal. (Young, 2005)

3. Ethnographic Approach / Contextual Enquiry. Provides a framework that helps understand users and their requirements using a structured interview. (Gerry Gaffney, 2004)

Construction Of The Method

The main aim of the method is to ensure that the student thinks about who the users are and the subsequent implications for the design of an interface, before any design work starts. It was also important that the framework should integrate with existing methodologies, or indeed be used in isolation to ensure adoption amongst a diverse student population.

Having established the need for students to think more deeply about the users, the next step was to outline the objectives of the framework.

1. The process must be rapid, without the need to collect primary data.

2. The student must be able to identify and understand the language used.

3. Should be accessible to students who are by definition less experienced.

4. To communicate a set of requirements for interface design.

5. It should consider the notion of multiple users of websites, rather then one.

Current practice in web development considers the identification of the user profile as the starting point of the user requirements analysis (Bolchini & Eric, 2004). A user profile describes stable archetypal qualities of a relevant target segment (Carroll 2002) and may comprise a variety of attributes based on demographic eg. age, gender, occupation, disabilities etc. or "webographic" eg. net usage habits, interests and software constraints, favourite sites etc.(Garrett, 2002). Profiles can be discovered through a variety of requirements elicitation techniques based on user research, such as surveys, contextual inquiry, focus groups and structured interviews. (Bolchini et al., 2004). The user profile can be broken down into four elements; a person profile, a web use profile, a context of use profile and personalising the user. The profile should be as accurate, clear and realistic as possible.

The mechanism in which the student obtains the data about the user is crucial. In the context of their projects, it was felt that techniques that use primary data would result in poor adoption of the framework. This is due to time and cost implications that would be placed on the student. As such, the framework must use an alternative method to generate the profile of the users. Any such method would rely upon judgment and reasonable guesses from the student themselves. "Collect facts if you have them, or make reasonable guesses because even a reasonable guess provides a focus". (Cato, 2001). Although it could be argued that this would result in an un-safe profile, the student can obtain some limited information about the user. This can found in ICA briefing documents where the user is loosely defined, but where much more analysis is expected.

Having undertaken the background research into existing methods for creating user profiles and more general user analysis, it was felt that a new cognitive aid and frame was required. The research would help shape the framework in terms of characteristic capture and how this could be communicated and synthesised into a set of requirements for the interface design.

Making the student think about ‘who’ the users are and how they would use the application would be the main objective. The framework would rely upon decision making from the student themselves, since no actual data would be collected. It must be emphasized that RUMM must be completed by students themselves, in the context of a project that they are working on. Subsequently the only reasonable outcome of the system would be merely to make the student think, rather than automatically produce the requirements.

The following questions provided a basis for the full development of RUMM;

1. Who are the users?

(age/gender/culture)

2. What do the users expect to do with the application?

(To learn, purchase online, assist them in their job, provide fun or leisure activity).

3. When will the users use the application?

(In the course of their job, in library or other access point, at home in their leisure time).

4. How will the users use the application?

(At home/work, on a modem connection, broadband, cd-rom, PC/Mac/Linux, Kiosk, PDA and with assistive technologies).

Once the student has thought about the characteristics of the users and how they intend to use the application, they are then expected to write a description of the Primary and Secondary user. By providing a framework for the student to think about the users, it was hoped that a much more detailed and focused description could be developed for the project. After the description is developed, they must think about the considerations for the layout, colour, content and navigation of the user interface. It is at this stage that the student would form the requirements for the interface design.

To view the RUMM document please refer to appendix 1.0.

Results Of The Student Survey

At the end of the module students were asked to complete a questionnaire regarding their opinion of using RUMM. These were returned anonymously to ensure an un-biased response. Seventy six students were asked to complete the questionnaire on a final year degree and on a postgraduate module. Fifteen questionnaires were returned by the undergraduates and six questionnaires were returned by the postgraduates. The response rate of 16% is considered to be normal for a questionnaire survey. The questionnaires asked the students if it helped them define users for their project, whether there were any elements missing that they felt should be included or if any unnecessary information was included. Additionally students were asked if they would use a more advanced tool based on the RUMM framework. To view the questionnaire used in the survey please refer to appendix 2.0.

Twenty students (95%) expressed that they found RUMM 'very useful/useful' in defining the audience for the projects that they were working on. When asked if they had used other approaches for defining the user 4 students (19%) said that they had never used other approaches, and of those who had (62%), said that it was more useful compared to the approaches used previously. Four students (19%) expressed the opinion that it was not much better then an approach they had used in the past.

The results from this section of the survey are very encouraging in terms of acceptance of the framework from the students. Although the framework is still at a very early stage of development, students felt that it did help them to define who the user is and a significant number of students felt RUMM was more useful than approaches they had used before. An interesting outcome of question two was the fact that students had been using previous approaches. Whilst this is true, the definition of an approach is critical here, as this had an ambiguous meaning. An approach could be defined as a rigid framework or conversely a more loose process of simply writing a paragraph about the user. The latter fits in well with the evidence available in many student in-course assessments (ICA) where they had not used RUMM and the user was only briefly considered.

Question three asked if there were any points missing from RUMM that should be considered. The responses were quite varied and include;

More Focus on personal background of users, including profession, hobbies and customs to help understand them better.

Non-native English speakers (specify level), users reading from right to left or from bottom to top, disabled users with/without assistive technologies, use at school/college/university.

Level of computer use/competence section, we need to know what is actually meant by novice, intermediate, expert.

This feedback is again quite interesting in that the students want to break down the characteristics even further. Some students felt that more characteristics should be integrated into RUMM, such as hobbies and the profession. It could be argued that these are perhaps too detailed and specific but are none the less, important in thinking about the user. Whilst it is felt these characteristics are valid, there might be a better way of thinking about the user at this level of detail. This is discussed in more detail in next section of this paper.

Question four asked if they felt that there was any unnecessary information being gathered through the method. Only one student responded to this question, where they thought there should be a clearer way of differentiating primary users from secondary users. It was felt that this is a very important aspect of RUMM that could have implications for the way in which a priority system could be integrated to ensure no conflicts arise.

Question five asked the student if an enhanced version of RUMM which produced design guidelines was available for them to use, would they find this useful. Thirteen students (62%) said that they would find such a tool very useful, seven student (33%) said that they would find it useful.

Evaluation & Further Development

The results of the survey reinforce the positive impact that RUMM has had on teaching. Whilst the framework is still in early development, it is evident that students who have used it are confident in its application. One area for improvement would be the second half of RUMM, where they have to write a brief summary of the primary, secondary users and think about the navigation, content, layout and colour. During this part of RUMM, reliance is placed on the knowledge of the student and there is a preconception that they understand usability and conventions associated with the application of colour and layout in interface design. This stage of RUMM requires more investigation, especially pertaining to the safety of assumptions made by the students. The approach of using characteristics to model the user is central to the framework, but it is recognised that this can be difficult for the student.

One area which may help the student identify more with the process, is the use of persona's. RUMM could be used to generate a number of different persona’s, based on a predefined model . For example the use of persona’s could help provide a rich contextual model of a group of primary targeted users based on conjoint analysis theory. The persona analysis process is divided into the two processes of simultaneous disjoint clustering, and identification of personas. These persona’s could be clustered to generate standardised requirements, based upon a minimal set of the most influential users. (Mikio Aoyama, 2005)

The framework thus so far has concentrated on user characteristics. But users also exhibit other traits such as motivation and needs. It is felt these are features lacking in the current version of RUMM, and as such more development is needed to consider their integration. In addition, further work will explore the traceability of RUMM to evaluate its influence on the design process. This would involve the investigation of the application of profiles upon a web site to determine differences in design approaches using a test project.

The use of RUMM has had a positive impact on the way students think about the user, and we believe the main objective of the framework has been realised. Once the tool had been used by students, some weaknesses became apparent. These mainly related to the use of primary and secondary users and dealing with conflicts and priorities which may arise. Additionally we had expected the students to understand the language used to define the characteristics, an assumption that we perhaps should have first tested in a pilot of the framework.

Conclusions

This paper has established that there is a need to encourage students to think more deeply about who the users are, what they do with the application, when and how they use it. There are very few methods that can achieve this without resorting to expensive and time consuming techniques, and which would not fit with the needs of the students themselves. The majority of students indicated that they found RUMM useful in helping them define the users for their projects. They also felt that there needs to be some mechanism to automate the requirements for layout, content, navigation and colours, perhaps via the production of guidelines. User characteristics could be expanded further, with additional components and the modification of others.

It is hoped that the next cycle of research would build upon the positive elements of the RUMM framework, and test other approaches to enhance the process. One of the ways in which this could be achieved is the integration of persona's and clustering. The clustering technique itself would require more research, in addition to investigating how to turn the outcomes into usable guidelines.

When we first investigated the need for a better approach for thinking about the user, we set out our objectives for the framework. These included the need for a rapid approach, helping the student to communicate requirements through a simple framework that they could use without any previous experience. It should also reinforce the notion of multiple users. The investigation and subsequent development of the RUMM framework showed that most objectives were attained and that RUMM provides a basis for the evolution of a more advanced tool.

Acknowledgements

We would like to thank the students in the School of Computing, University of Teesside who were willing to use RUMM during their projects and answer our survey during a very busy period.

REFERENCES

1. Athula Ginige, San Murugesan. (2001) The Essence of Web Engineering Managing the Diversity and Complexity of Web Application Development, IEEE MultiMedia, 8, 2, 22-25.

2. M A Lockyer, G Griffiths, B D Hebbron, B J Oates. (2003) A Teaching Method & Tool for Web Engineering, Third IEEE International Conference on Advanced Learning Technologies, 284.

3. Lowe, D., Eklund, J. (2002) Client Needs and The Design Process in Web Projects. Journal of Web Engineering, 1,1.

4. Chris Barry, Michael Lang. (2001) A Survey of Multimedia and Web Development Techniques and Methodology Usage, IEEE MultiMedia, 8, 2, 52-60.

5. D, Bolchini, J, Mylopoulos. (2003) From Task-Oriented to Goal-Oriented Web Requirements Analysis, Fourth International Conference on Web Information Systems Engineering, 166.

6. Indi Young, Adaptive Path, "Task Based Audience Segmentation",http://www.adaptivepath.com/publications/essays/archives/000496.php, accessed 13th February 2006.

7.GaffneyG.,Contextualenquiry",http://www.sitepoint.com/article/contextual-enquiry-primer, accessed 13th February 2006.

8. Bolchini,D., Eric, S., (2004) Modelling User Requirements For Web Application Design. Studies in Communication Sciences, Università della Svizzera italiana, 175.

9. Carroll, J.M. (2002) Making Use. Scenario-based design of human-computer interactions, Boston: MIT Press.

10. Garrett, J.J. (2002) The Elements of User Experience, Indianapolis: New Riders.

11. Bolchini,D., Eric, S. (2004) Modelling User Requirements For Web Application Design. Studies in Communication Sciences, Università della Svizzera italiana, 173-196.

12. Cato, J. (2001) User-Centered Web Design, London: Addison-Wesley.

13. Mikio A. (2005) Persona-and-Scenario Based Requirements Engineering for Software Embedded in Digital Consumer Products. 13th IEEE International Conference on Requirements Engineering, 85-94.Appendix A4 - UKAIS Phd Consortium Conference Paper

Appendix A4 - UKAIS Phd Consortium Conference Paper

Determining Web User Requirements Through User Modelling: A Framework For User Characteristics Capture

Andrew Bingham

School of Computing, University of Teesside, Borough Rd, Middlesbrough, TS1 3BA, UK.

Abstract

The research explores the need to capture user requirements in the context of web engineering projects. The objective of the research is to develop a cognitive aid and framework. The author argues that the need for such a framework exists because of the changing nature of web development, with time to market pressures for developers, and a fluid and constantly changing and evolving user. The first cycle of research has been completed and evaluated. The evidence suggests that such a framework benefits students when working on web development projects.

Keywords (User Characteristics, User Analysis, Evaluation, method-in-action, Persona, Web Design)

1. Introduction

Over the last decade web engineering has matured in terms of its approach to formalised development, starting with the adoption of information systems methodologies and progressively adopting its own methods and tools. Whilst this has led to better quality of web applications, it is believed that there is still room for improvement. The web is still in a state of massive change, with new web standards and technology pushing the need for continual professional development in web engineering. In addition there has been a paradigm shift in defining the web engineering discipline. “Web engineering is a holistic approach, and it deals with all aspects of web based systems development, starting from conception and development to implementation, performance evaluation and continual maintenance.” (Ginge And Murugesan 2001).

Within the school of computing, a number of courses provide an opportunity for students to engage in the development of web applications using a variety of development methodologies. These include the ‘Simple Web Method’ (M A Lockyer, G Griffiths, B D Hebbron, B J Oates, 2003) that supports the web engineering process, through a defined process of stages. Although this provides the student with a distinct process, it was found that students were not considering the user in their designs. The need to address issues of accessibility and usability within the development process has become essential, with most web development now becoming much more user focused. To address these issues, research was undertaken to support students in their analysis of the users within the context of the project. The term ‘users’

This paper describes the importance of the research within the web engineering discipline and establishes the position the framework will take within the development lifecycle. It will outline the proposed research paradigm and propose an appropriate methodical research approach to be taken by the researcher.

2. The Need For A Method To Record User Characteristics

Accessibility and the need to enhance *usability* within web applications has had an impact in web engineering, calling for a change in the requirements gathering phase, implementation approach and evaluation to ensure conformance to a set of pre-defined requirements. This has led to a shift towards an iterative process in web projects, where requirements conformance is measured at each stage. In addition, web projects are characterised by short development times and continual maintenance, as highlighted by Ginige and Murugesan, “Web based system development is not a one time event, as practiced by many; it’s a process with a long lifecycle.” This reinforces the notion of the web application being an evolving product, with the need for web methodologies to adopt the same approach.

The need to capture user characteristics and requirements is becoming an essential aspect of effective web engineering. A major problem that occurs in Web Engineering projects is that the users get to know how to express their requirements very late in the process, i.e. after the design artefacts appeared. (Lowe and Eklund, 2002) To address these issues, there has been a move towards more user centered methods that have a user requirements stage explicitly integrated into the process. It is recognised that while this does offer a more user focused product, the ‘user centered design’ ethos needs to go beyond this. “The need for a systematic approach to capture user navigational requirements has some merit and perhaps some urgency.” (Barry and Lang, 2001)

User analysis for a commercial web project can be undertaken using a variety of techniques, including questionnaires and focus groups. The data could then be used to influence subsequent stages in the development process. Within the context of an educational web project, questionnaires are often an unrealistic proposition for students due to cost and time implications. Whilst this constraint is a major barrier to conducting user analysis, it was still felt that a tool or framework was required to enable the student to think about users characteristics and produce a set of guidelines or requirements for the design of a web interface.

Students need to consider the role of the user in their web development projects to ensure embedded artefacts are focused on the needs of the users. Usually the student is using an existing methodology or creating their own hybrid to best fit the project. As such it is essential that the framework can be used with a variety of development methodologies or indeed used in isolation.

Before any user requirements can be documented, a method to capture their requirements must be used. Existing approaches include;

1. *Concur Task Trees*. Allows the envisioning user interaction with the application and derive information, navigation and presentation design accordingly. (Bolchini, Mylopoulos, 2003)

2. *Task Based Audience Segmentation*. A technique that defines the target audience by the tasks they perform to achieve a goal. (Young, 2005)

3. *Ethnographic Approach / Contextual Enquiry.* Provides a framework that helps understand users and their requirements using a structured interview. (Gerry Gaffney, 2004)

3. Construction Of The Method

The main aim of the method is to ensure that the student thinks about *who* the users are and the subsequent implications for the design of an interface, before any design work starts. It was also important that the framework should integrate with existing methodologies, or indeed be used in isolation to ensure adoption amongst a diverse student population.

Having established the need for students to think more deeply about the users, the next step was to outline the o*bjectives of the framework*.

1. The process must be rapid, without the need to collect primary data.

2. The student must be able to identify and understand the language used.

3. Should be accessible to students who are by definition less experienced.

4. To communicate a set of requirements for interface design.

5. It should consider the notion of multiple users of websites, rather then one.

Current practice in web development considers the identification of the *user profile* as the starting point of the user requirements analysis (Cato,2001). A user profile describes stable archetypal qualities of a relevant target segment (Carroll 2002) and may comprise a variety of attributes based on demographic eg. age, gender, occupation, disabilities etc. or “webographic” eg. net usage habits, interests and software constraints, favourite sites etc.(Garrett, 2002). Profiles can be discovered through a variety of requirements elicitation techniques based on user research, such as surveys, contextual inquiry, focus groups and structured interviews. (Bolchini & Eric, 2004).

The mechanism in which the student obtains the data about the user is crucial. In the context of their projects, it was felt that techniques that use primary data would result in poor adoption of the framework. This is due to time and cost implications that would be placed on the student. As such, the framework must use an alternative method to generate the profile of the users. Although it could be argued that this would result in an un-safe profile, the student can obtain some limited information about the user. This can found in ICA briefing documents where the user is loosely defined, but where much more analysis is expected.

Having undertaken the background research into existing methods for creating user profiles and more general user analysis, it was felt that a new student centered method was required. The research would help shape the framework in terms of characteristic capture and how this could be communicated and synthesised into a set of requirements for the interface design.

Making the student think about 'who' the users are and how they would use the application would be the main objective. The framework would rely upon decision making from the student themselves, since no actual data would be collected. Subsequently the only reasonable outcome of the system would be merely to make the student *think,* rather than automatically produce the requirements.

The following questions provided a basis for the full development of RUMM;

- **Who are the users?** (age/gender/culture)

- **What do the users expect to do with the application?** (To learn, purchase online, assist them in their job, provide fun or leisure activity).

- **When will the users use the application?** (In the course of their job, in library or other access point, at home in their leisure time).

- **How will the users use the application?** (At home/work, on a modem connection, broadband, cd-rom, PC/Mac/Linux, Kiosk, PDA and with assistive technologies).

Once the student has thought about the characteristics of the users and how they intend to use the application, they are then expected to write a description of the *Primary* and *Secondary* user. By providing a framework for the student to think about the users, it was hoped that a much more detailed and focused description could be developed for the project. After the description is developed, they must think about the considerations for the *layout*, *colour*, *content* and *navigation* of the user interface. It is at this stage that the student would form the requirements for the interface design.

4. Results Of The Student Survey

At the end of the module students were asked to complete a questionnaire regarding their opinion of using RUMM. These were returned anonymously to ensure an un-biased response. Seventy six students were asked to complete the questionnaire on a final year degree and on a postgraduate module. Fifteen questionnaires were returned by the undergraduates and six questionnaires were returned by the postgraduates. The questionnaires asked the students if it helped them define users for their project, whether there were any elements missing that they felt should be included or if any unnecessary information was included. Additionally students were asked if they would use a more advanced tool based on the RUMM framework. Please see *appendix 2.0* for a copy of the questionnaire.

Twenty students (95%) expressed that they found RUMM ‘very useful/useful’ in defining the audience for the projects that they were working on. When asked if they had used other approaches for defining the user 4 students (19%) said that they had never used other approaches, and of those who had (62%), said that it was more useful compared to the approaches used previously. Four students (19%) expressed the opinion that it was not much better then an approach they had used in the past.

The results from this section of the survey are very encouraging in terms of acceptance of the framework from the students. Although the framework is still at a very early stage of development, students felt that it did help them to define who the user is and a significant number of students felt RUMM was more useful than approaches that they had used. An interesting outcome of question two was the fact that students had been using previous approaches. Whilst this is true, the definition of an approach is critical here, as this had an ambiguous meaning. An approach could be defined as a rigid framework or conversely a more loose process of simply writing a paragraph about the user. The latter fits in well with the evidence available in many student ICA’s where they had not used RUMM and the user was only briefly considered.

Question three asked if there were any points missing from RUMM that should be considered. The responses were quite varied and include;

- More Focus on personal background of users, including profession, hobbies and customs to help understand them better.
- Non-native English speakers (specify level), users reading from right to left or from bottom to top, disabled users with/without assistive technologies, use at school/college/university.
- Level of computer use/competence section, we need to know what is actually meant by novice, intermediate, expert.

This feedback is again quite interesting in that the students want to break down the characteristics even further. Some students felt that more characteristics should be integrated into RUMM, such as hobbies and the profession. It could be argued that these are perhaps too detailed and specific but are none the less, important in thinking about the user. Whilst it is felt these characteristics are valid, there might be a better way of thinking about the user at this level of detail. This is discussed in more detail in section five of this paper.

Question four asked if they felt that there was any unnecessary information being gathered through the method. Only one student responded to this question, where they thought there should be a clearer way of differentiating primary users from secondary users. It was felt that this is a very important aspect of RUMM that could have implications for the way in which a priority system could be integrated to ensure no conflicts arise.

Question five asked the student if an enhanced version of RUMM which produced design guidelines was available for them to use, would they find this useful. Thirteen students (62%) said that they would find such a tool very useful, seven student (33%) said that they would find it useful.

5. Evaluation & Further Development

The results of the survey reinforce the positive impact that RUMM has had on teaching. Whilst the framework is still in early development, it is evident that students who have used it are confident in its application. One area for improvement would be the second half of RUMM, where they have to write a brief summary of the primary, secondary users and think about the navigation, content, layout and colour. During this part of RUMM, reliance is placed on the knowledge of the student and there is a preconception that they understand usability and conventions associated with the application of colour and layout in interface design. This stage of RUMM requires more investigation, especially pertaining to the safety of assumptions made by the students. The approach of using characteristics to model the user is central to the framework, but it is recognised that this can be difficult for the student.

One area which may help the student identify more with the process, is the use of persona's. RUMM could be used to generate a number of different persona's, based on a predefined model . For example the use of persona's could help provide a rich contextual model of a group of primary targeted users based on conjoint analysis theory. The persona analysis process is divided into the two processes of simultaneous disjoint clustering, and identification of personas. These persona's could be clustered to generate standardised requirements, based upon a minimal set of the most influential users. (Mikio Aoyama, 2005)

The framework thus so far has concentrated on user characteristics. But users also exhibit other traits such as *motivation* and *needs.* It is felt these are features lacking in the current version of RUMM, and as such more development is needed to consider their integration.

The use of RUMM has had a positive impact on the way students think about the user, and we believe the main objective of the framework has been realised. Once the tool had been used by students, some weaknesses became apparent. These mainly related to the use of primary and secondary users and dealing with conflicts and priorities which may arise. Additionally we had expected the students to understand the language used to define the characteristics, an assumption that we perhaps should have first tested in a pilot of the framework.

6. Conclusions

The need to *think* about users before designing an interface is important. There are very few methods that can achieve this without resorting to expensive and time consuming techniques. The majority of students indicated that they found RUMM useful in helping them define the users for their projects. They also felt that there needs to be some mechanism to automate the requirements for layout, content, navigation and colours, perhaps via the production of guidelines. User characteristics could be expanded further, with additional components and the modification of others.

It is hoped that the next cycle of research would build upon the positive elements of the RUMM framework, and test other approaches to enhance the process. One of the ways which this could be achieved is the integration of persona's and clustering. The clustering technique itself would require more research, in addition to investigating how to turn the outcomes into usable guidelines.

When we first investigated the need for a better approach for thinking about the user, we set out our objectives for the framework. These included the need for a rapid approach, helping the student to communicate requirements through a simple framework that they could use without any previous experience. It should also reinforce the notion of multiple *users.* The investigation and subsequent development of the RUMM framework showed that most objectives were attained and that RUMM provides good basis for the evolution for a more advanced tool.

References

1. Athula Ginige, San Murugesan. "**Guest Editors' Introduction: The Essence of Web Engineering—Managing the Diversity and Complexity of Web Application Development**," *IEEE MultiMedia*, vol. 8, no. 2, pp. 22-25, April-June, 2001.

2. M A Lockyer, G Griffiths, B D Hebbron, B J Oates. "**A Teaching Method & Tool for Web Engineering**" *icalt*, p. 284, Third IEEE International Conference on Advanced Learning Technologies (ICALT'03), 2003.

3. Lowe, D., Eklund, J. **Client Needs and The Design Process in Web Projects**. Journal of Web Engineering, Vol. 1, No.1. 2002.

4. Chris Barry, Michael Lang. "**A Survey of Multimedia and Web Development Techniques and Methodology Usage**," *IEEE MultiMedia*, vol. 8, no. 2, pp. 52-60, April-June, 2001.

5. Davide Bolchini, John Mylopoulos. "**From Task-Oriented to Goal-Oriented Web Requirements Analysis**," *wise*, p. 166, Fourth International Conference on Web Information Systems Engineering (WISE'03), 2003.

6. Indi Young, Adaptive Path, "**Task Based Audience Segmentation**", http://www.adaptivepath.com/publications/essays/archives/000496.php, accessed 13th February 2006.

7. Gaffney G., **Contextual-enquiry**, http://www.sitepoint.com/article/contextual-enquiry-primer, accessed 13th February 2006.

8. Cato, J. **User-Centered Web Design**, London: Addison-Wesley. 2001.

9. Carroll, J.M. **Making Use. Scenario-based design of human-computer interactions**, Boston: MIT Press. 2002

10. Garrett, J.J. **The Elements of User Experience**, Indianapolis: New Riders. 2002

11. Bolchini,D., Eric, S,. **Modelling User Requirements For Web Application Design**. Studies in Communication Sciences, Università della Svizzera italiana. p.173-196. 2004

12. Mikio A. **Persona-and-Scenario Based Requirements Engineering for Software Embedded in Digital Consumer Products**. 13th IEEE International Conference on Requirements Engineering (RE'05). pp. 85-94. 2005.

Appendix A5 - Online Survey Questionnaire

Question 1. Please indicate which course you are currently enrolled.

BSc International Business Information Technology ●

BSc Computer Studies ●

BA Web Design ●

BA Creative Digital Media ●

BSc Web Development ●

BSc IT and Networks ●

Creative Multimedia ●

Masters Course ●

Other ●

Question 2. Please Indicate Your Mode Of Study

Full-time: ●

Part-time: ●

Question 3. Did you use WURF in your in-course assessment

Yes: ●

No: ● If no, please move onto Question 10.

Now thinking about how you used WURF in your in-course assessment (ICA). The next four statements use a 'Likert Scale' rating scheme based on the suitability of WURF for performing the requirements analysis.

The scale key is as follows:

1 = Strongly Agree 2 = Agree 3 = Neither agree or disagree 4 = Disagree 5 = Strongly Disagree

Statement 4a. I understood the process of WURF without the need to ask for help.

- 1●
- 2●
- 3●
- 4●
- 5●

Statement 4b. WURF helped me think about the user in terms of characteristics and their requirements within the web application.

- 1●
- 2●
- 3●
- 4●
- 5●

Statement 4c. WURF helped me think about translating requirements into tasks and functions within the application.

- 1●
- 2●
- 3●

- 4●
- 5●

Statement 4d. WURF takes too much time to complete.

- 1●
- 2●
- 3●
- 4●
- 5●

Statement 4e. There’s no benefit for me in using WURF.

- 1●
- 2●
- 3●
- 4●
- 5●

Statement 4f. Before using WURF I had not thought enough about the user and their requirements.

- 1●
- 2●
- 3●
- 4●
- 5●

Question 5. What step within WURF did you least understand?

a. Statement Of Purpose / Vision / Define Web Application Objectives ●

b. Define Web Actors / Define Web Actor Tasks ●

c. Define Web Functional Requirements ●

d. Define Web Non-Functional Requirements ●

Question 6. Which aspect of WURF did you feel helped the most?

a. Statement Of Purpose / Vision / Define Web Application Objectives ●

b. Define Web Actors / Define Web Actor Tasks ●

c. Define Web Functional Requirements ●

d. Define Web Non-Functional Requirements ●

Question 7. Would You Use WURF again?

Yes:●

No:●

Question 8. Now thinking about the 'user characteristic' stage. Would you prefer to use 'pre-written' persona's / profiles here rather than writing these yourself?

Yes:●

No:●

I Don't Understand this question:●

Question 9. Are there any additional enhancements that you would like to see incorporated? Please indicate these in the box below;

Question 10. Have you used another method for User Requirements Analysis?

Yes: (please indicate this the text feild below) ●

No: ●

if Yes, please write here which one you used.

Submit Questionnaire

Appendix A6 - Survey Data Table

	Course	Mode	Used WURF in ICA	4a	4b	4c	4d	4e	4f	What step within WURF did you least understand?	Which aspect of WURF did you feel helped the most?	Use again?		enhancements		Have you used another method for User Requirements Analysis?
1	Masters	Full-time	yes	2	1	2	4	5	2	Define Web Non-Functional Requirements	Statement Of Purpose / Vision / Define Web Applica	yes	yes		no	
2	ComputerStudies	Part-time	yes	1	1	1	4	5	2	Define Web Non-Functional Requirements	Define Web Actors / Define Web Actor Tasks	yes	yes		no	
3	WebDesign	Full-time	yes	2	2	2	3	4	3	Define Web Non-Functional Requirements	Define Web Actors / Define Web Actor Tasks	yes	no		yes	RUMM
4	ComputerStudies	Full-time	yes	2	2	2	3	3	1	Define Web Non-Functional Requirements	Statement Of Purpose / Vision / Define Web Applica	yes	yes		no	
5	ComputerStudies	Full-time	yes	2	2	2	4	5	4	Define Web Actors / Define Web Actor Tasks	Statement Of Purpose / Vision / Define Web Applica	yes	no		no	
6	ComputerStudies	Part-time	yes	1	1	2	4	4	3	Define Web Actors / Define Web Actor Tasks	Statement Of Purpose / Vision / Define Web Applica	yes	no	n/a	yes	RUMM
7	CreativeMultimedia	Full-time	yes	2	2	2	2	2	2	Statement Of Purpose / Vision / Define Web Applica	Statement Of Purpose / Vision / Define Web Applica	yes	yes		yes	
8	ComputerStudies	Part-time	yes	2	2	2	4	4	2		Define Web Actors / Define Web Actor Tasks	yes	no	N/A	yes	RUMM V1
9	ComputerStudies	Part-time	yes	3	2	2	4	3	2	Define Web Actors / Define Web Actor Tasks	Statement Of Purpose / Vision / Define Web Applica	yes	yes		no	
10	Masters	Full-time	yes	2	2	2	4	4	3	Define Web Non-Functional Requirements	Statement Of Purpose / Vision / Define Web Applica	yes	no		no	
11	IBIT	Full-time	no												yes	RUMM2
12	Masters	Full-time	yes	2	3	3	2	4	1	Define Web Non-Functional Requirements	Define Web Actors / Define Web Actor Tasks	yes	yes		no	
13	WebDesign	Full-time	yes	2	2	3	5	3	4	Define Web Non-Functional Requirements	Define Web Actors / Define Web Actor Tasks	yes	no		yes	RUMM
14	WebDesign	Full-time	yes	2	2	2	3	5	5	Define Web Functional Requirements	Statement Of Purpose / Vision / Define Web Applica	yes	no		yes	RUMM, questionnaire, personas
15	CreativeMultimedia	Full-time	no												yes	RUMM
16	WebDesign	Full-time	yes	3	2	2	3	4	2	Define Web Non-Functional Requirements	Statement Of Purpose / Vision / Define Web Applica	yes	no		yes	rumm version 1
17	CreativeMultimedia	Full-time	yes	2	2	1	2	4	2	Define Web Non-Functional Requirements	Define Web Functional Requirements	yes	yes		no	

Appendix B1 - CAWE Development and Implementation

Development Environment

A number of options were considered for the implementation of the CAWE tool. These were all centred upon contemporary 'web application development platforms' governed by the need to integrate with existing systems such as the institutions VLE, intranet and to store user data. The researchers own skill set was also taken into consideration. Standalone platforms that had to be installed on a PC were considered, for example, Adobe Integrated Runtime (AIR) was a platform that could be used, as it provides an installation package that can be distributed via the VLE. AIR can also access user data via a web service for the purposes of consuming user data. This was an important consideration given the need to access and update data generated within CAWE tool. PHP and MySQL could then used to control the rules model, but would require the data to be transformed into a web service for consumption by the AIR application. This adds a layer of complexity and additional overhead in terms of time in order to develop and test. Alternatively, a browser based solution using XHTML and CSS would offer a number benefits:

- The author has ten years experience of developing XHTML, CSS, PHP and MySQL web applications and has a number of pre built libraries that could be utilised.

- Any problems with the CAWE tool can be updated and rolled out instantly using a browser based web application. If AIR was adopted, the application would need to be re-installed on each machine.

- It would not be necessary to build a web service for AIR to consume user data.

The CAWE tool is as a mission critical application in terms of this research programme and for its use within the module curricular. Using new technologies such as AIR may provide benefits from a student experience perspective, but with an additional overhead in respect of time to learn. A pragmatic approach to development was needed due to the very short development lifecycle, as design and development time was limited to two months in total. It was decided to use XHTML, CSS, PHP and MySQL as the development environment, which would expedite the release of the CAWE tool in time for the new term.

Analysis and Design

The existing meta-model was modified and transformed into a class diagram in order to address the issues highlighted in the evaluation, as discussed in the previous section. A rules model had to be written to control a number of aspects of the CAWE tool, including the consistency, completeness and correctness and checking of requirements and the student dashboard. This is represented within the 'Dashboard class' as CompletenessAndConsistency(), a core function within the CAWE tool that drives the student dashboard. The dashboard is dynamically created, based on the completeness of the requirements and provides feedback to the student regarding which aspects need to be completed next.

The CompletenessAndConsistency() function is dependent on the TaskToActorAssoc() and FunctionToActorAssoc() functions. In the second cycle, it was identified that being able to associate tasks and functional requirements to specific actors would enhance the traceability aspect of WURF, but was something that could not be achieved correctly in the paper based tool. The CAWE tool provides an opportunity to represent and control associations more precisely and impose a set of rules to govern how these are created and subsequently modified. For example, if an Web Actor is deleted part way through the project, the dependenciesCheck() function within the webAppActor class is instantiated and provides feedback within the student dashboard. Once the student confirms that they wish to remove the actor, the association functions are instantiated to remove the actor in all association instances.

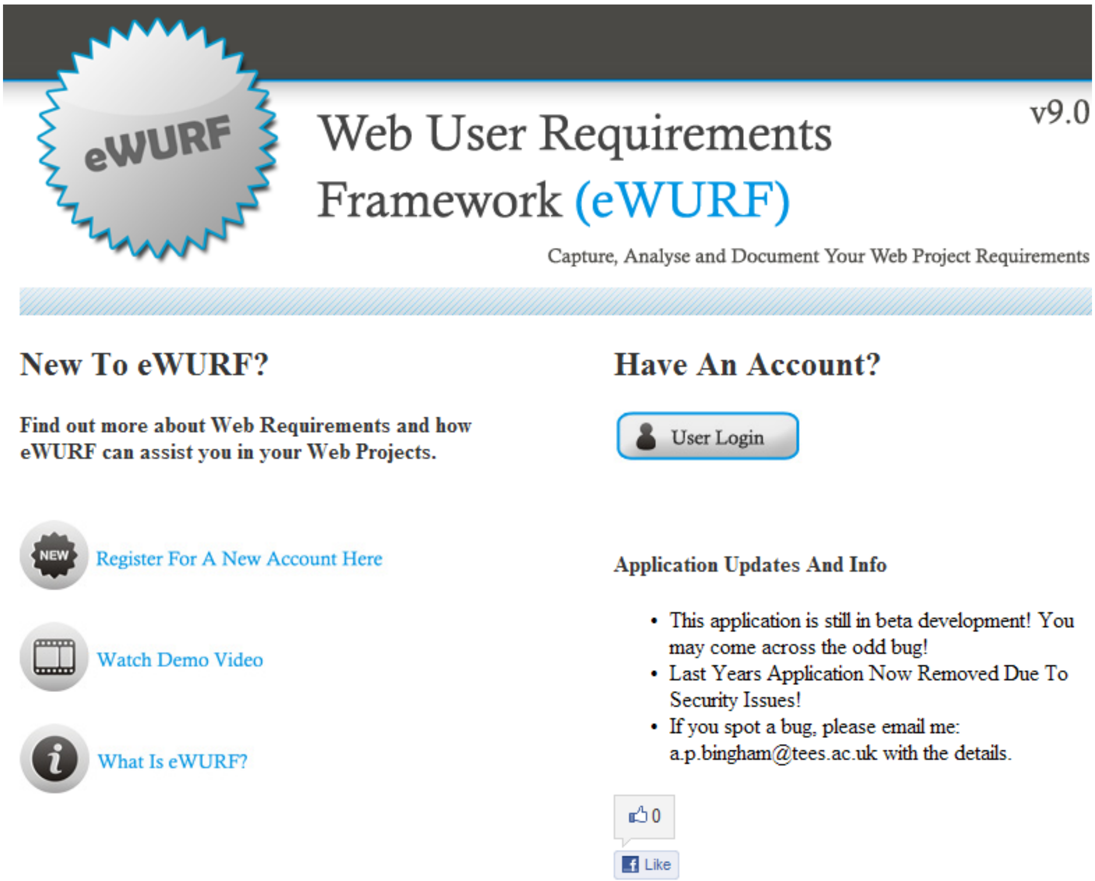
One of the main benefits to the student in using the CAWE tool is the ability to automatically generate an SRS document, which has version control to ensure they can revert to a previous version if required. This necessitated the use of two classes, RequirementDocument and RequirementPattern, each with its own set of functions to transform user data into an SRS. The latter class is required for the generation of a ‘virtual document’, held in a separate data store for later retrieval by the RequirementDocument class. This ‘virtual document’ ensures that each SRS is unique for version control purposes and so that unique identifiers can be faithfully reproduced in each document for traceability purposes.

Figure 8.2.3 (see page 224), illustrates all the classes within the CAWE tool, including the consumption of data outside the scope of the application, represented in the ‘module class’. These provided the basis for the implementation and transformation into PHP classes within the CAWE tool.

CAWE Tool Implementation

A suitable Integrated Development Environment (IDE) was chosen based upon the need to undertake both client-side and server-side scripting and to test key classes during development. Dreamweaver was chosen due to its ability to achieve this and the way it was able to provide a mechanism to write valid PHP classes. Rapid Prototyping was used to develop the CAWE tool, reflecting the short development time available. A total of nine iterations within the rapid prototyping approach provided stable release versions. Classes were written and immediately tested on a live server environment, rather than on a local server. This ensured syntax and logic errors could be quickly identified, rectified and re-tested to ensure conformity to the model. Error reporting was switched on within the PHP configuration settings to help identify syntax errors.

A client-side HTML template was produced, (see Figure 6.2.5), using both XHTML and CSS that included generic content holders to enable dynamic server-side functions to be embedded later. The class diagram provided the basis for transformation of the model into PHP classes. A simple version control system was adopted, with the benefit of being able to return to a previous working version if required.



Web User Requirements Framework (eWURF)

Capture, Analyse and Document Your Web Project Requirements

New To eWURF?

Find out more about Web Requirements and how eWURF can assist you in your Web Projects.

Register For A New Account Here

Watch Demo Video

What Is eWURF?

Have An Account?

Application Updates And Info

- This application is still in beta development! You may come across the odd bug!
- Last Years Application Now Removed Due To Security Issues!
- If you spot a bug, please email me: a.p.bingham@tees.ac.uk with the details.

Figure B1.1 Client-side HTML template.

The starting point for the development was dictated by the need to solve the most difficult problems first. The author had already implemented a library of authentication, database connectivity, user data (adding, editing and deleting) classes and so these could be integrated easily when required. It was identified that the 'Web Actor to Task' association class was the most challenging aspect of the implementation phase due to the multi-dimensional relationships between both Task/Functional Requirements Web Actors. The rules model had to be represented within the application logic. For example, the dependenciescheck() function had to be able to access relationships between Tasks and Web Actors. Any routine must be able to check for changes to multiple dependencies and be able to report these to the student dashboard. Accordingly, the TaskToActorAssoc() and FunctionToActorAssoc() association classes were chosen as the starting point for the first iteration within the implementation phase.

Association Classes – First Iteration

Initial ideas regarding the association class included the ability to relate a maximum number of four actors to any one task or functional requirement. The rules model imposed certain constraints on this process, for example, in order to reflect changing tasks and functional requirements, dependencies checking must take place. If an actor was no longer required but was associated to a task or functional requirement, this should be indicated within the student dashboard during the deletion routine.

Before the association routine could be written, it was important that a limited number of classes were written and tested, as both the *TaskToActorAssoc()* and *FunctionToActorAssoc()* classes consumed pre-existing data. The following classes formed the basis of the first iteration; *WebAppActor, WebAppTask, WebAppFunction* (see Figure B1.2).

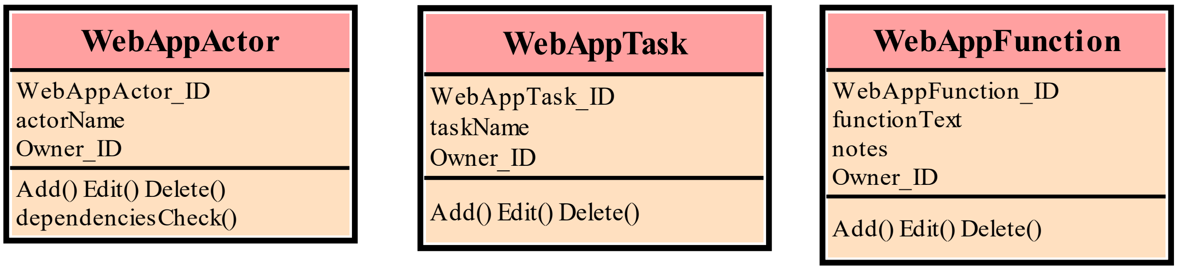


Figure B1.2 WebAppActor, WebAppTask and WebAppFunction Classes.

An association between individual Web Actors and Tasks had to be modelled, with the routine that enabled the student to *add*, *edit* and *delete* both Tasks and Web Actors in later versions of their requirements specifications. It was therefore essential that dependencies could be checked and edited by the dependenciesCheck() function without recourse to elaborate SQL queries. Early attempts included limiting the association data structure to a maximum of four actors. Each association would have a unique id (TaskToActorAssoc_ID), a foreign key (WebAppTask_ID) and names of the actors for the association. In order to check ownership of the association, an Owner_ID key was included (see Figure B1.3).

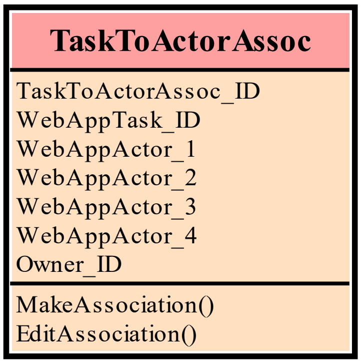


Figure B1.3 TaskToActor Data Structure in prototype 1.

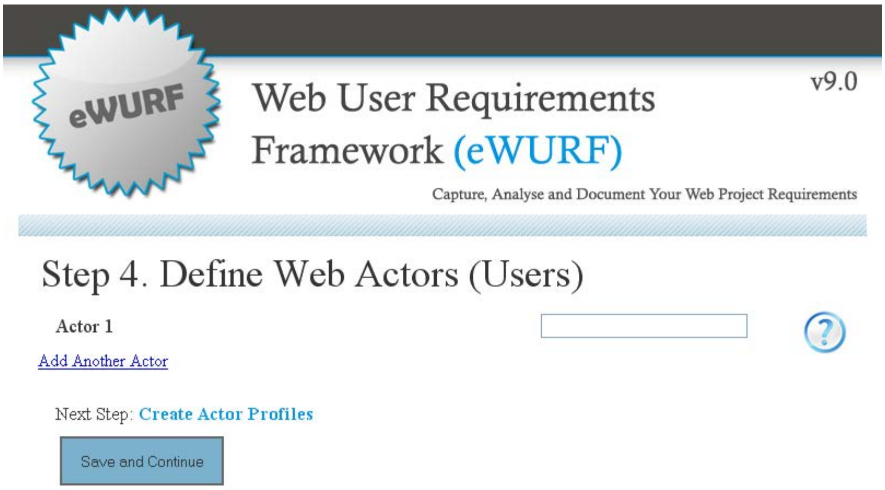
The first attempt was partially successful, in that it was possible to create the record for the association, but it was recognised that two issues would become a barrier for further progression:

1. It was only possible to associate four actors.
2. Data integrity problems. To address this, the primary key of the WebAppActor table should have been used.

In particular, the second problem regarding data integrity issues would have meant writing complex SQL to retrieve the correct data for the actor id association and would result in explicitly relating both the TaskToActorAssoc and WebAppActor when the student edits an actor. The first problem would prevent extension of the tool to include an unlimited number of associations. In addition, the student was presented with a limited number of actors and tasks in their dashboard screen.

Association Class – Second Iteration

In order to resolve some of the issues presented in the first iteration, attention was paid to the student's 'Define Web Actor' dashboard screen. In the first iteration this was 'hard coded' to a maximum of ten actors which proved to be too restrictive. The screen was redeveloped to include a feature that enabled the student to add unlimited actors. This was achieved by inclusion of a button 'Add Another Actor', written in javascript (see Figure B1.4).


eWURF
Web User Requirements Framework (eWURF)
v9.0
Capture, Analyse and Document Your Web Project Requirements

Step 4. Define Web Actors (Users)

Actor 1

Add Another Actor
Next Step: Create Actor Profiles
Save and Continue

Figure B1.4 Add Web Actors Screen with ‘Add Another Actor’ button.

The TaskToActorAssoc class structure was modified to enable a one-to-four (one task-to-four actors) association model. An advantage of this was the ability to model relationships using the primary key of the actors table, rather than the name of the actor. If an actor name was changed, the integrity of its relationship is maintained and preservation of the association remains (see Figure B1.5). Actor associations were limited within this iteration, but the approach offered a way forward for *one-to-many relationships*.

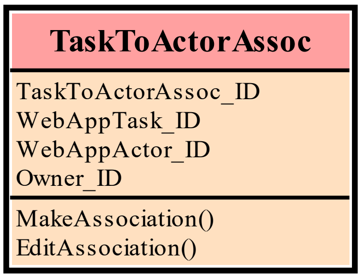
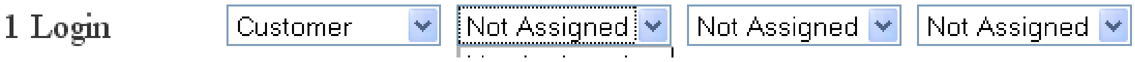


Figure B1.5 Modified TasktoActorAssoc class.

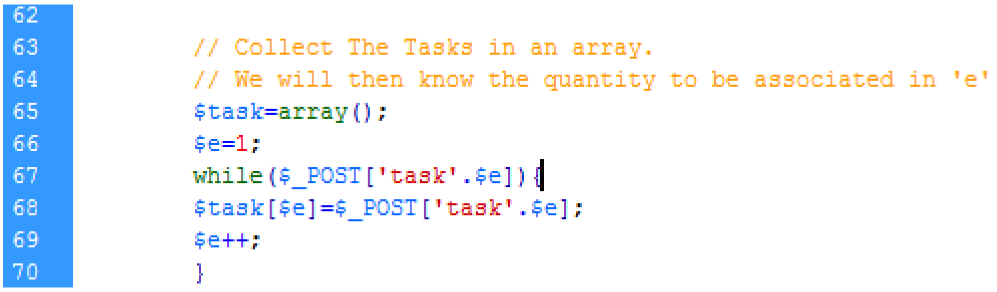
The ‘Task To Actor’ association dashboard was then developed to enable the student to select specific actors to be associated with a particular task. A routine was developed to list each task in ascending order, which then selected actor data from the WebAppActor table. Four dropdown menus were dynamically created next to each task (see Figure B1.6).



1 Login Customer Not Assigned Not Assigned Not Assigned

Figure B1.6 Task to Actor Association Dashboard.

Within the TasktoActorAssoc class, a function named MakeAssocation() was written. A number of programmatic issues had to be resolved in order to capture an unknown and unlimited amount of data posted from the student dashboard. In normal circumstances, variable data coming from a ‘web form’ is pre-determined including its key and value (key-value pairs). Allowing the student to send unlimited key-value pairs necessitated the production of a ‘while loop’ that populated an array. Within the array, an index contained a reference to the number sequence of the task as it appeared in the dashboard, along with the WebAppTask_ID reference (see Figure B1.7).



```

// Collect The Tasks in an array.
// We will then know the quantity to be associated in 'e'
$task=array();
$e=1;
while($_POST['task'.$e]){
$task[$e]=$_POST['task'.$e];
$e++;
}
```

Figure B1.7 Array to capture unlimited key-value pairs from POST data in student dashboard.

Once the WebAppTask_ID’s had been collected, the next part of the routine was to collect the four actors associated with the task. A subroutine was written to achieve this. It had been hoped to take advantage of PHP’s ability to create multidimensional arrays, for example, **array (array(“task1”, “actor1_ID”, “actor2_ID”, “actor3_ID”, actor4_ID”))** Implementing using this approach proved to be unreliable and an alternative solution was sought.

```
e = Total Number Of Tasks
b = Array Counter
s = Comparison Counter

While Loop One > Collect all post data and increment e by 1.
End Loop

While Loop Two > while e<=s (executes to maximum value
contained in e, using s as a comparison)

        c = Inner Array Counter

        Nested While Loop Three > while c<=4 (execute to a
        maximum of 4, using c as a comparison )

        Build SQL for Insert (based upon task and actor id's)

        End Loop

End Loop
```

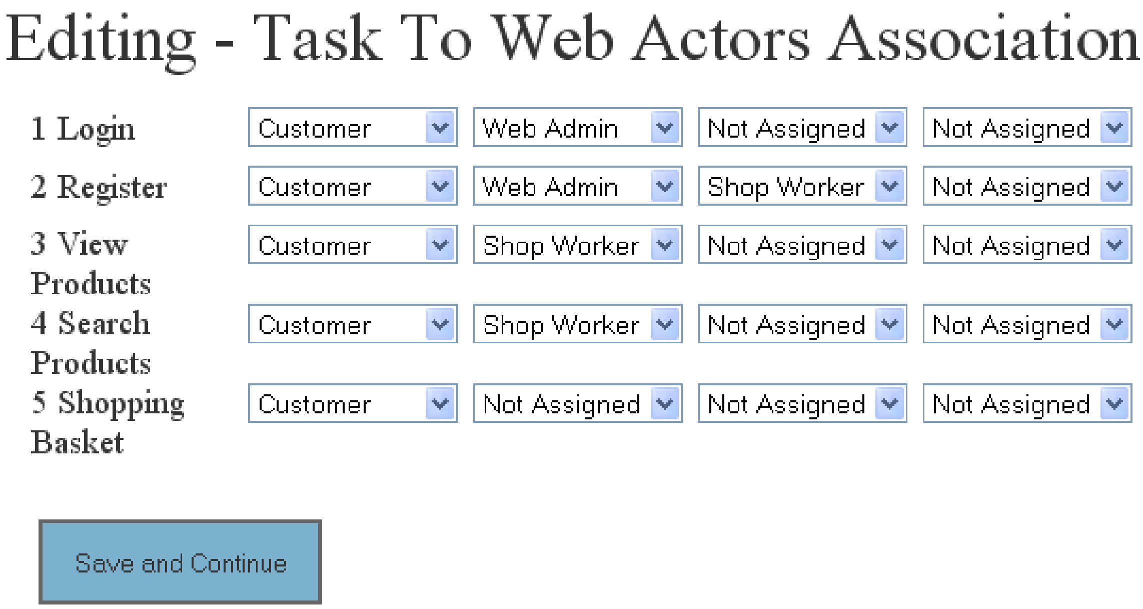
Figure B1.8 Association Classes - Subroutine Logic.

Figure B1.8 represents a solution to the problems of displaying multiple associations in the student dashboard. This was perhaps the most challenging aspect of the development and one which took the most time to write, test and validate. Much of the understanding developed in writing and developing this subroutine was applied to other association classes, such as the *FunctionToActorAssoc()* association class and would also be used in the editing subroutines of these functions (see Appendix C1 - MakeAssociation PHP class). The subroutine provided the basis for the

FunctionToActorAssoc(), with some modifications necessary to reflect variations in data labels. Iteration two proved to be stable in terms of its ability to add associations. The next iteration focused on editing facilities within the student dashboard.

Association Class – Third Iteration

The second iteration of the association class provided a stable code base in which to explore further solutions to a number of programmatic problems. This was achieved in the third iteration of the prototype. Although the 'add association' function proved to be stable, an editing facility was also required to reflect the changing nature of requirements over the duration of the web project. It was found that only slight modifications were needed to this function and the advantages of using the webActor_ID as the foreign key was a central factor in its correct execution. The student 'edit task to web actor association' dashboard provided a list of all associations (see Figure B1.9 Edit Task to Web Actor Association).



Editing - Task To Web Actors Association

1 Login	Customer	Web Admin	Not Assigned	Not Assigned
2 Register	Customer	Web Admin	Shop Worker	Not Assigned
3 View Products	Customer	Shop Worker	Not Assigned	Not Assigned
4 Search Products	Customer	Shop Worker	Not Assigned	Not Assigned
5 Shopping Basket	Customer	Not Assigned	Not Assigned	Not Assigned

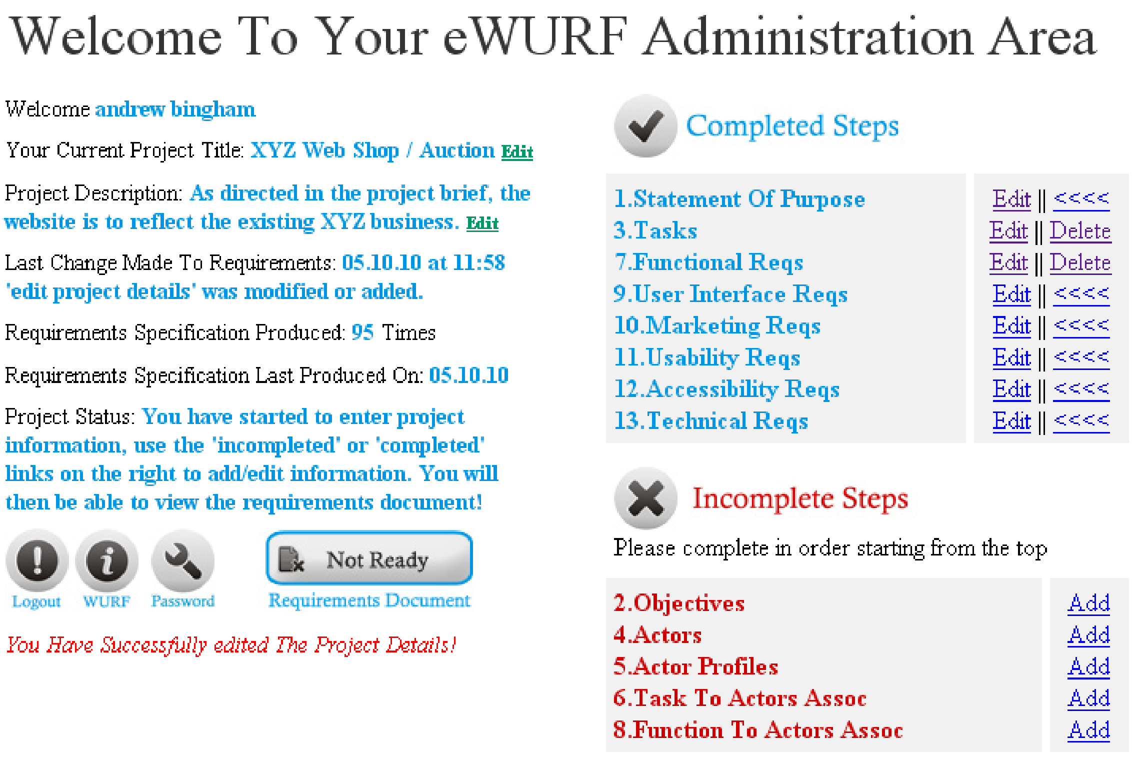
Figure B1.9 Edit Task to Web Actor Association.

The *EditAssociation()* function within both the *FunctionToActorAssoc()* and *TaskToActorAssoc()*, classes collected ‘post data’ and initiated an ‘update’ query. This function utilised much of the logic developed in the *MakeAssociation()* function, with additional routines written to ensure data validity. Work continued on additional classes that enabled the student to *add*, *edit* and *delete* Web Actors; Tasks; Functional Requirements; and non-functional requirements.

Consistency, Completeness and Correctness and Dependencies Checking Class – Fourth Iteration

One of the benefits of moving to a CAWE tool was an ability to impose constraints within a rules model. The rules model in turn provided a way forward to incorporate some of the ideas developed the diagnosing and problem identification section of this section, especially relating to *completeness* of SRS Document.

The rules model was represented in the *consistencyAndCompleteness()* function within the Dashboard class, where checks were to be performed to ensure that the student could not produce the SRS Document until certain rules within the model were met. Additionally, the class became host to a number of core functions that governed the student dashboard. For example, the functions it contained provided a visual representation of the meta-model, including those aspects which were completed or incomplete (see Figure B1.10).



Welcome To Your eWURF Administration Area

Welcome andrew bingham

Your Current Project Title: XYZ Web Shop / Auction Edit

Project Description: As directed in the project brief, the website is to reflect the existing XYZ business. Edit

Last Change Made To Requirements: 05.10.10 at 11:58 'edit project details' was modified or added.

Requirements Specification Produced: 95 Times

Requirements Specification Last Produced On: 05.10.10

Project Status: You have started to enter project information, use the 'incompleted' or 'completed' links on the right to add/edit information. You will then be able to view the requirements document!

You Have Successfully edited The Project Details!

Completed Steps

1.Statement Of Purpose	Edit || <<<<
3.Tasks	Edit || Delete
7.Functional Reqs	Edit || Delete
9.User Interface Reqs	Edit || <<<<
10.Marketing Reqs	Edit || <<<<
11.Usability Reqs	Edit || <<<<
12.Accessibility Reqs	Edit || <<<<
13.Technical Reqs	Edit || <<<<

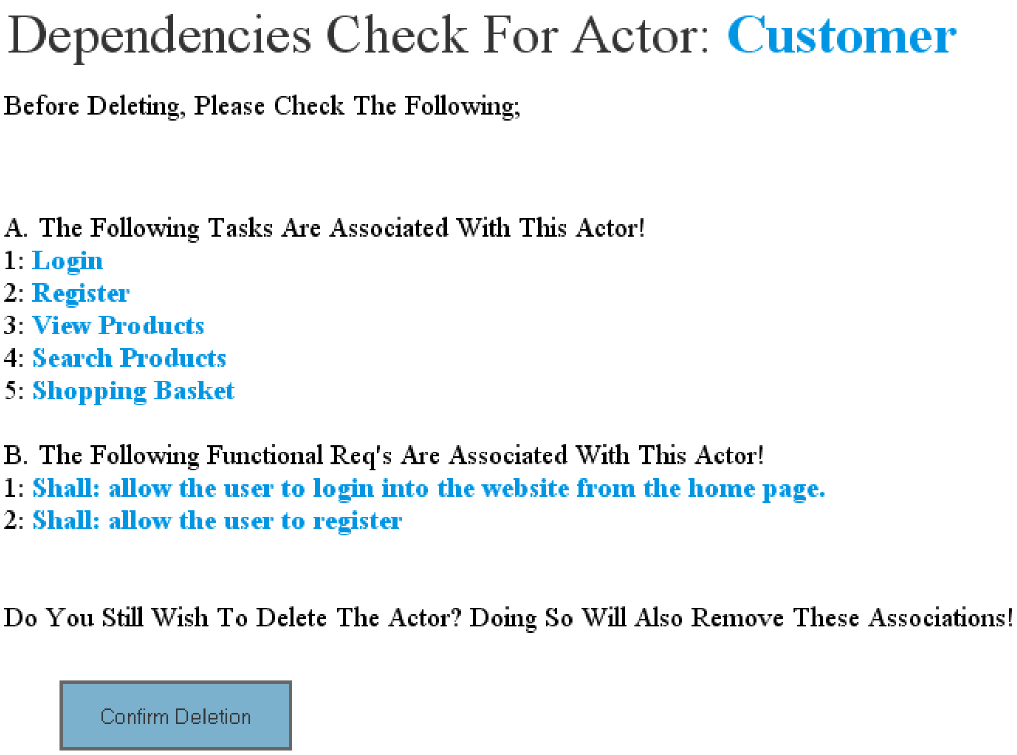
Incomplete Steps

Please complete in order starting from the top

2.Objectives	Add
4.Actors	Add
5.Actor Profiles	Add
6.Task To Actors Assoc	Add
8.Function To Actors Assoc	Add

Figure B1.10 CAWE Tool - Student Dashboard.

The student dashboard also controlled access to the editing, deletion and dependencies checking functions. For example, if an web actor was removed, a dependencies check was performed on the *FunctionToActorAssoc()* and *TaskToActorAssoc()* associations (see Figure B1.11). The student could then confirm deletion in acknowledgement of the dependencies checking routine.



Dependencies Check For Actor: Customer

Before Deleting, Please Check The Following;

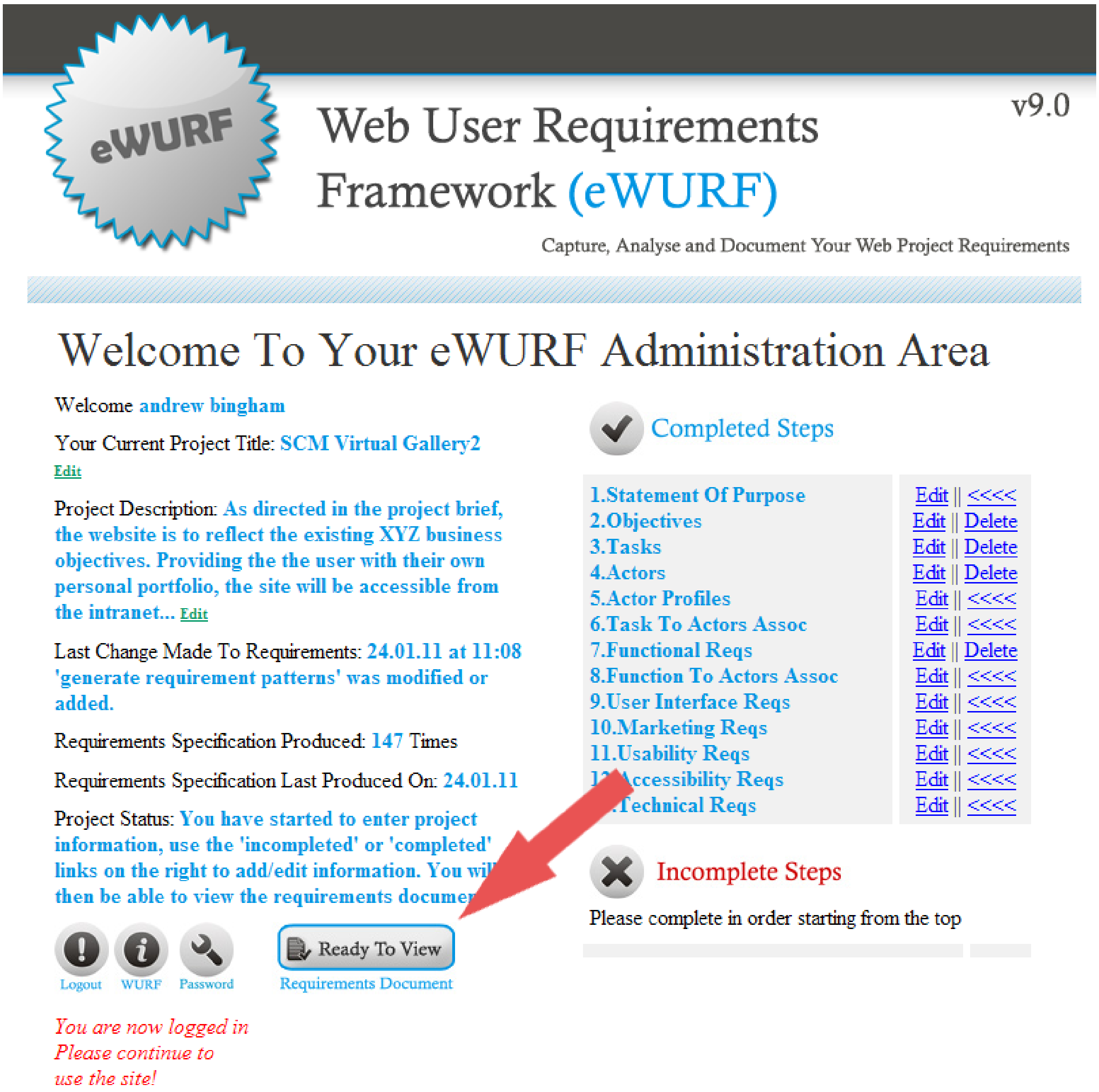
A. The Following Tasks Are Associated With This Actor!
1: Login
2: Register
3: View Products
4: Search Products
5: Shopping Basket

B. The Following Functional Req's Are Associated With This Actor!
1: Shall: allow the user to login into the website from the home page.
2: Shall: allow the user to register

Do You Still Wish To Delete The Actor? Doing So Will Also Remove These Associations!

Figure B1.11 Dependencies Checking Function.

The consistencyAndCompleteness() function in the dashboard class also contained a routine that controlled the students ability to create the SRS document. Each time the dashboard was accessed the class was initiated. It then executed a number of functions to check for the completeness of the requirements. Dashboard objects were also generated that allowed the student to *add*, *edit* or *delete* data as appropriate. For example, the student may have edited the 'statement of purpose' or delete 'tasks'. Once all of the steps were completed, the student dashboard allowed the student to view the SRS document via a highlighted button (see Figure B1.12).



Web User Requirements Framework (eWURF)

Capture, Analyse and Document Your Web Project Requirements

Welcome To Your eWURF Administration Area

Welcome andrew bingham

Your Current Project Title: SCM Virtual Gallery2 Edit

Project Description: As directed in the project brief, the website is to reflect the existing XYZ business objectives. Providing the the user with their own personal portfolio, the site will be accessible from the intranet... Edit

Last Change Made To Requirements: 24.01.11 at 11:08 'generate requirement patterns' was modified or added.

Requirements Specification Produced: 147 Times

Requirements Specification Last Produced On: 24.01.11

Project Status: You have started to enter project information, use the 'incompleted' or 'completed' links on the right to add/edit information. You will then be able to view the requirements document

You are now logged in Please continue to use the site!

Completed Steps

1.Statement Of Purpose	Edit || <<<<
2.Objectives	Edit || Delete
3.Tasks	Edit || Delete
4.Actors	Edit || Delete
5.Actor Profiles	Edit || <<<<
6.Task To Actors Assoc	Edit || <<<<
7.Functional Reqs	Edit || Delete
8.Function To Actors Assoc	Edit || <<<<
9.User Interface Reqs	Edit || <<<<
10.Marketing Reqs	Edit || <<<<
11.Usability Reqs	Edit || <<<<
12.Accessibility Reqs	Edit || <<<<
13.Technical Reqs	Edit || <<<<

Incomplete Steps

Please complete in order starting from the top

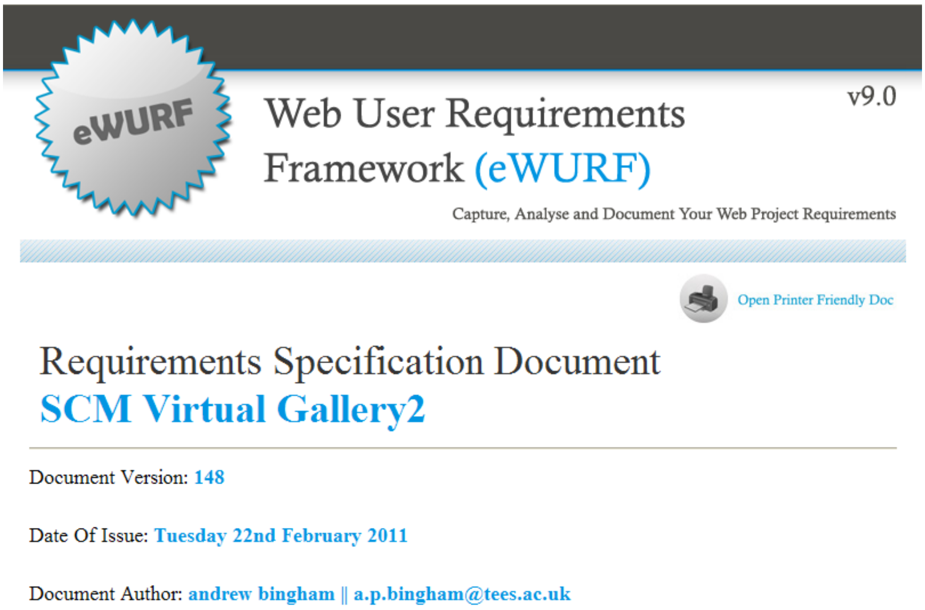
Figure B1.12 Requirements Document – Ready To View Button.

The fourth iteration proved to be stable in terms of being able to *add*, *edit*, and *delete* user data and to check for the consistency, completeness and correctness of the requirements. Chapter 2 highlighted the need capture and output the requirements in a ‘requirements specification document’ and suggested ways that this could be achieved. Stability of the web application and confidence in the validity of the user data, meant work could start on this aspect of the CAWE tool.

Requirements Specification Document Class – Fifth Iteration

The final iteration within the implementation phase of the CAWE tool included a mechanism to represent the user data within an overall SRS document. Since the consistencyAndCompleteness() function within the Dashboard class had already checked requirements, the resulting document would contain validated data and therefore deemed to be complete. In line with investigation into Requirements Specification Documentation, additional informational items were added, such as a version number and date stamps. Log data that was captured whenever a student performed an action within the CAWE tool provided the date and revision number that is displayed within the SRS document (see Figure B1.13).

Within the rest of the requirements specification, information entered by the student during the analysis stage was formatted and presented. Functional and non-functional requirements are also displayed. Associations are made explicit within the SRS document and each functional requirement has a unique id that was consistent even if the requirements are subsequently modified. Non-functional requirements were displayed at the end of the SRS document (see Figure B1.14, B1.15 and B1.16).



Web User Requirements Framework (eWURF)

Capture, Analyse and Document Your Web Project Requirements

Requirements Specification Document

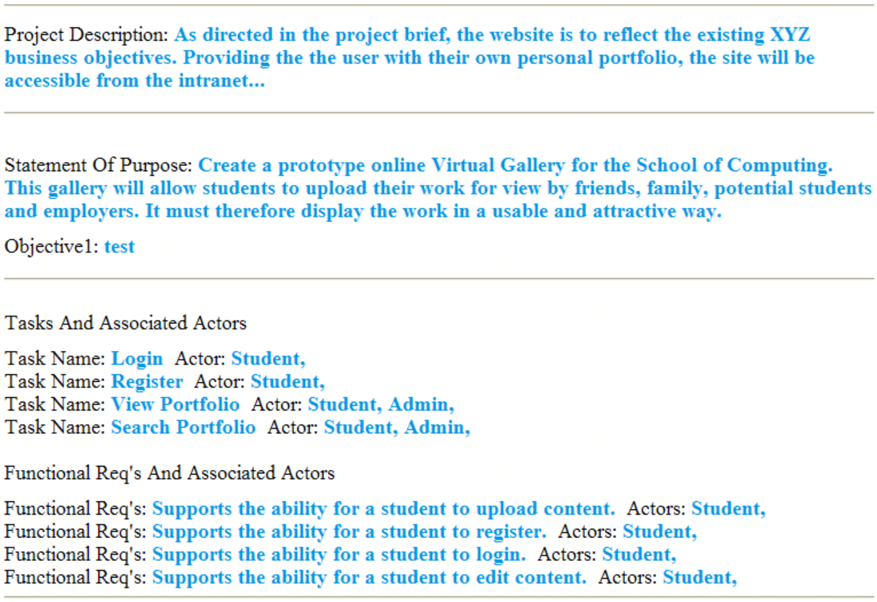
SCM Virtual Gallery2

Document Version: 148

Date Of Issue: Tuesday 22nd February 2011

Document Author: andrew bingham || a.p.bingham@tees.ac.uk

Figure B1.13 Requirements Specification Documentation – Version Control.



Project Description: As directed in the project brief, the website is to reflect the existing XYZ business objectives. Providing the the user with their own personal portfolio, the site will be accessible from the intranet...

Statement Of Purpose: Create a prototype online Virtual Gallery for the School of Computing. This gallery will allow students to upload their work for view by friends, family, potential students and employers. It must therefore display the work in a usable and attractive way.

Objective1: test

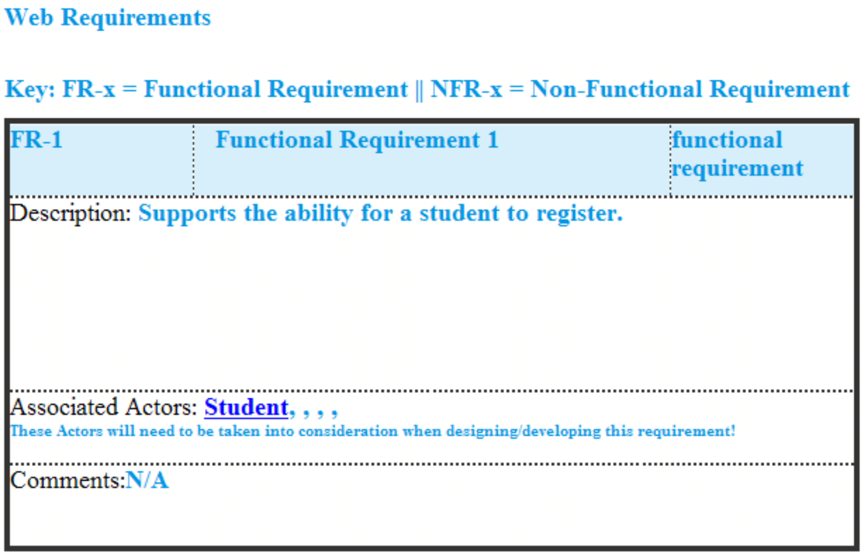
Tasks And Associated Actors

Task Name: Login Actor: Student,
Task Name: Register Actor: Student,
Task Name: View Portfolio Actor: Student, Admin,
Task Name: Search Portfolio Actor: Student, Admin,

Functional Req's And Associated Actors

Functional Req's: Supports the ability for a student to upload content. Actors: Student,
Functional Req's: Supports the ability for a student to register. Actors: Student,
Functional Req's: Supports the ability for a student to login. Actors: Student,
Functional Req's: Supports the ability for a student to edit content. Actors: Student,

Figure B1.14 Requirements Specification Documentation – Additional Info.



Web Requirements

Key: FR-x = Functional Requirement || NFR-x = Non-Functional Requirement

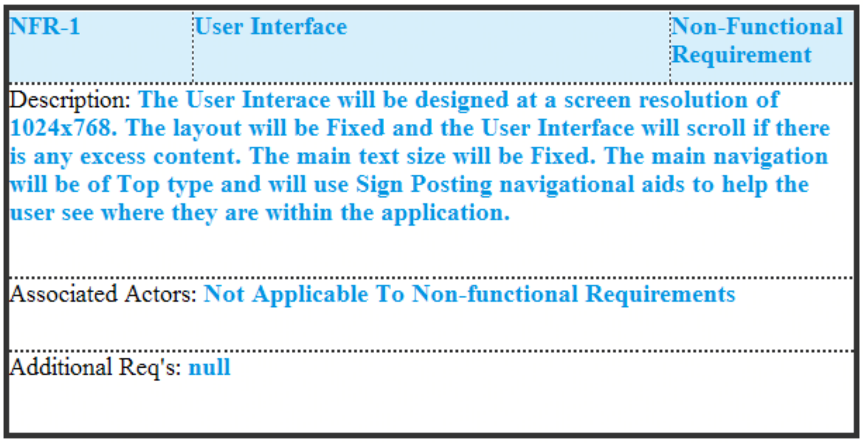
FR-1	Functional Requirement 1	functional requirement

Description: Supports the ability for a student to register.

Associated Actors: Student, , , ,
These Actors will need to be taken into consideration when designing/developing this requirement!

Comments:N/A

Figure B1.15 Requirements Specification Documentation – Functional Requirement Template.



NFR-1	User Interface	Non-Functional Requirement

Description: The User Interace will be designed at a screen resolution of 1024x768. The layout will be Fixed and the User Interface will scroll if there is any excess content. The main text size will be Fixed. The main navigation will be of Top type and will use Sign Posting navigational aids to help the user see where they are within the application.

Associated Actors: Not Applicable To Non-functional Requirements

Additional Req's: null

Figure B1.16 Requirements Specification Documentation – Non-Functional Requirement Template.

The traceability of requirements was highlighted as being an issue in the second research cycle. Creating an identifier for each requirement was one mechanism to allow the student to link the requirement through to the design documentation and into the implementation for validation purposes.

It was also important to format the requirements specification document so that it was usable by a range of people involved in the web project. Other group members, tutors and assessors would all need to read the information that it contained. The student would also need to print off the document for inclusion within the ICA submission. Two versions of the document were therefore offered to the student:

1. A screen only version that allowed the student to review the information quickly and easily.

2. A print version that allowed the student to print off information ready for inclusion within their ICA for assessment purposes. A printer friendly button was provided in the screen only version.

In order to facilitate traceability via the identifier system, during initial testing it was found that it was not possible to select a requirement and display it within the requirements specification document. On further investigation it was established that as requirements were constantly updated and deleted, the primary key of the WebAppFunction table was not a reliable mechanism to use for the identifier for each requirement. The MySQL documentation provided an explanation for this behaviour and a possible solution. It was found that unlike other relational database management systems, MySQL was able to 'reuse primary keys' once a record had been deleted, a behaviour that had not been envisaged in the initial design of the RequirementPattern class.

A more reliable approach was written and tested which would ensure forward and backwards traceability even if the requirements were modified. A RequirementPattern class was initiated from within the student dashboard once they had selected the ‘View Requirements Document’ button (see Figure B1.17 RequirementPattern Class).

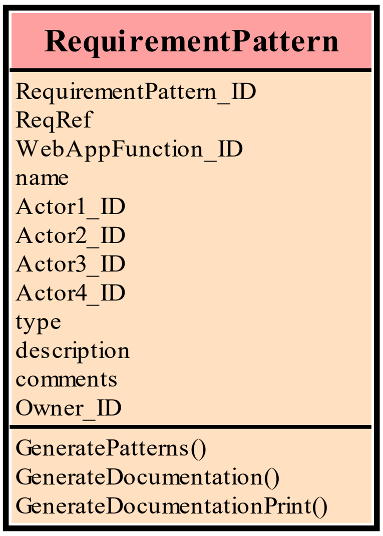


Figure B1.17 RequirementPattern Class.

The class selected and transformed the data for insertion into the RequirementPattern table. A RequirementPattern_ID provided the unique identifier for each requirement, which in turn was used by the RequirementDocument class that formats and displayed the information to the student within the SRS document. A print version was accessible from within the screen, which in turn called an additional class named requirementDocumentationPrint that formatted the specification so that it was compatible for printers (see Appendix C2, requirementDocumentPrint PHP class).

CAWE Tool Deployment

Prior to the website being launched, a number of additional classes had to be integrated within the CAWE tool to facilitate common tasks and to enhance the user experience. This included *user registration*; *login*; *password recovery*; *password change*; *help and guidance*; *actor profiles*; *video tutorials*; *log data* and *student profile questionnaire*.

To facilitate the student registration and to enable later tracking of individual students via the log system, the registration screen consumed a web service that provided a list of modules from the HEI student registry application. This would enable precise tracking of each action the student undertook within the CAWE tool and would be used for later analysis in the evaluation of the CAWE tool in use.

Alpha Testing

Stability of the registration system and user authentication was paramount, as it would become a mission critical tool in terms of use on a number of modules and in particular for the assessment process. Testing of individual classes which had taken place during the iterative prototyping stage had uncovered a number of issues and these had been resolved before moving to the next version. It was found that the validity of the data within the association classes was problematic and traced to an SQL query that was not correctly formed. Alpha testing was necessary to ensure that no conflicts or errors existed when executing functions and classes together. A number of dummy user accounts were created to ensure validity of data generated by the individual users. Alpha testing was conducted on a live server environment, but hidden behind a firewall to ensure no one else could access it.

During alpha testing a number of problems were identified:
1. Null data was being inserted into the Actor, Task and Website Objective tables. A knock on effect persisted within the association class, in particular PHP generated infinitive loops due to missing data. The application crashed at this point, resulting in a server time out problem. The issue was resolved by validating and cleaning POST data sent from the student dashboard and sending the student an appropriate message for them to address the missing form data.

2. MYSQL primary key column sequencing problem on the Actor and Task tables. A 'referential integrity bug' presented itself when deleting an Actor or Task where these had previously being associated. MYSQL reuses primary keys when using the 'DELETE WHERE' clause and this resulted in an 'out of sequence id pattern' within the tables, which caused the application to crash. This is a documented feature of MYSQL which adversely affected testing until a solution could be found. The solution centred upon resequencing the primary key column using the 'ALTER TABLE' clause after deletion occurred in both Web Actor and Task tables. For example:

```
ALTER TABLE WebAppActor ORDER BY WebAppActor_ID ASC ;
```

3. Due to the stateless nature of the platform and the object orientated approach for initiating functions within classes, it was necessary to adopt two technologies in order to 'maintain state'. These included session variables, where key-value pairs reside on the server and can be accessed by the application as long as the session persists. Global variables were used in order to assign variables for use in multiple functions, as some classes were run simultaneously from within a single class call. Persistence of sessions presented the author with the majority of programmatic issues to resolve. An

appropriate convention was setup to aid this, with global variables always declared in a group at the top of the relevant function.

4. A vulnerability was identified with user data, where an SQL query could accidentally delete data not belonging to the student. All SQL statements were re-written with the following additional clause: *WHERE Owner_ID = '$Owner_ID'*. Data that was being modified at this stage now had to belong to the student that had logged in. In addition, the vulnerability called into question the safety of holding all user data in one database. It was therefore decided to run a backup for the user data each day. An automatic database backup function was written for use in the final version, which relied on the UNIX 'Cron' call, a system that enables a time based job schedule to run at predetermined intervals. It was later found that this feature was not accessible on the HEI server. Manual backups were therefore the only option available.

Final Release

A dedicated web address '**www.scm.tees.ac.uk/WURF**' was setup to host the CAWE tool to ensure it could be easily accessed and that it was available outside of the institution's firewall. To assist with future error tracing and to ensure integrity of student generated data, two versions of the application were created to run concurrently:

1. A live application that would be used by staff and students.

2. A test application that would be used by the author to recreate and fix errors as they arose on the live site. Test data from the live site could be ported over to the test application in order to accurately recreate the error. Critical updates to the test application could be made before being applied to the live site.

Some differences were noted on the server running the live site. Most notably the version of PHP on the live server was 4.6 and on the test server it was

PHP 5.3. The reason for this was legacy software running on the intranet. No major issues presented themselves due to the difference in PHP versions, but it did prevent the running of generic PHP classes that had been developed for automatic database backup, as this used PHP 5 libraries to facilitate file output.

Appendix B1.1 - Software Requirements Specification

xxxxx Access Requirements

Status:	Second Release
Author:	xxxxx
Date of issue:	xxxxx
Reference:	xxxxxx
Number of pages:	xxxx

Table of Contents

Document Control

Issue	*Date of issue*	*Comments*
0.1	23 April 2xxx	Pre workshop draft
0.2	21 May 2xxx	Draft for Review
0.3	25 May 2xxx	Updated following review
0.4	25 May 2xxx	Updated following review
0.5	25 May 2xxx	Updated following review
0.6	29 May 2xxx	Updated following review
0.7	30 May 2xxx	Updated following review
0.8	12 June 2xxx	Updated following review
1.0	13 June 2xxx	First Issue

Executive Summary

Mission Statement for the Project

The mission of this project is to "put in place systems that enable The xxxxxx to become the point of first resort for anyone who wants to access a comprehensive archive of material from the United Kingdom Web domain."[1]

The Access Requirements project will generate a list of requirements for accessing information stored within the xxxxxxx. The project has a fixed time scope of xxx weeks, and is intended to gather the majority of the requirements, including all the key requirements and possibly a list of areas that require further investigation.

The scope of this project is limited to the Access to the xxxxxx, i.e. harvest / ingest of websites, other archive material (e.g. books, journals and newspapers) are out of scope.

Priority of Requirements

Once the requirements had been identified (see **Error! Reference source not found.** for the methodology used to identify requirements), they were categorised into one of five priorities, i.e.:

- “**Must Have**” requirements are basic functionality that is essential for the web archive;
- “**Should Have**” requirements are not essential for an initial version of the Web Archive, however they will have a substantial influence on the impact of the Web Archive, its usability and presentation and should be implemented as soon as possible;
- “**Could Have**” requirements are not essential for the Web Archive, however they are useful functionality that could be implemented in future versions of the Web Archive.
- “**Would Be Nice To Have**” requirements are not essential for the Web Archive; however they may be useful for a later version of the Web Archive, although they have a low priority.
- “**Rejected**” requirements are those which are either out of scope of this project, undesirable or untenable / undeliverable.

The assignment of priority to requirements has been done through direct ranking by external sources, e.g. the Oxford University Library Service survey, and from consensus of opinion from internal workshops.

Potential Implementation Solutions

During the course of gathering requirements, there were numerous times where potential implementation solutions have been suggested. These potential implementation solutions are out of scope of this document.

However, they have been captured and for completeness included with the underlying requirement. They can be seen as indented italic text below the underlying requirement.

4. How and where will users be able to access?

There are three fundamental requirement areas to consider here, i.e. User Access Controls, Access Routes and Material Access Control.

User Access Controls, which are requirements focused on users, i.e. how to define the level of access for different types of user.

Access Routes, which are requirements focused on where a user can gain access to archived material.

Material Access Controls, which are requirements focused on archived material, i.e. how to define the level of availability for different material.

User Access Controls

Must Have

1. A level of access that does not require Username and Password, i.e. for unregistered users.

Should Have

2. Functionality to allow different access rights for different types of user, i.e. multiple levels of access (e.g. Reader, Registered Researcher and Non-Registered Researcher).

    If there is a requirement (from Legal Deposit legislation) to restrict access, then it would be better to have this restriction controlled using a login, which will all allow access through the internet anywhere, i.e. not just in reading rooms.

3. The ability to offer value added services linked to a user, e.g. user created collections, and favourites.

    This could be facilitated through the use of a user account or an identification layer, e.g. user registration, identification and password or Virtual Readers Card. Any user registration process for this should be user friendly, i.e. not onerous.

Could Have

4. Remote access for site owners to the archived version of their site, even if it is not publicly available.
5. Access controlled through an existing standard, e.g. Shibboleth.

Would Be Nice To Have

No User Access Control requirements have been given this priority.

Rejected

Must be a way to configure user access. The minimum requirement will be to have access set by generic user profile and the maximum requirement will be to have access set on a user by user basis. This requirement has been rejected as there have been repeated suggestions that restrictions should not be levied at individuals.

Appendix B1.2 - Student Email for Survey

Dear Student.

Following on from last weeks' lecture announcement, please find link to the e-WURF Evaluation Questionnaire for Online Business Systems, as required for your ICA!

Your views are greatly appreciated and should only take 5mins to fill in.The data that you submit will not be used for any other purpose.

http://www.scm.tees.ac.uk/WURF/questionnaire2.php?userID=f6047269

Please cut and paste into your browsers address bar if the above link is not active

All the best.

Andrew Bingham

Senior Lecturer (Web Development)

Associate Teaching Fellow

School Of Computing

University Of Teesside

TS1 3BA

T: 01642 342666

E: a.p.bingham@tees.ac.uk

Appendix C1 – MakeAssociation Class

```
<?php

session_start();

//////////////////////////////////////////////////

// MakeAssociation Function - Actors And Tasks
Association

//////////////////////////////////////////////////

include ('../common.php');

DBConnect();

//COLLECT GET data and create an array

$formValueGET=array();

foreach ($_GET as $key => $value) {

$formValueGET[$key] = strip_tags($value);

}

//Conditional Statement To Check If We Are Adding or
Editing Associations

$edit = $formValueGET['edit'];

$fromAdmin = $_GET['fromAdmin'];

    if($edit == "true"){
```

```
    EditAssociation($Host, $User, $Password,
    $DBName, $table_1, $table_2, $table_3, $table_4,
    $table_5, $table_6, $table_7, $table_8);

    }else{

    MakeAssociation($Host, $User, $Password,
    $DBName, $table_1, $table_2, $table_3, $table_4,
    $table_5, $table_6, $table_7, $table_8,
    $fromAdmin);

    }

function MakeAssociation($Host, $User, $Password,
$DBName, $table_1, $table_2, $table_3, $table_4,
$table_5, $table_6, $table_7, $table_8, $fromAdmin
){

//include "checkLoginSession.php";

        $Link = mysql_connect($Host, $User,
        $Password);

        $Owner_ID  = $_SESSION['Owner_ID'] ;

//We Need To Collect The Task ID And the next 4
Actor ID's.

        $task=array();

        $e=1;

        while($_POST['task'.$e]){

        $task[$e]=$_POST['task'.$e];

        //echo $image[$i].'  ';
```

```
         $e++;

         }

         $e = $e -1;

//var $b will count the total tasks submitted by the user

         $b = 1;


//var $s as our comparison operator

         $s = 1;


//Collect the task value from the array using while loop

         while($s<=$e){


//For collecting multi-dimensional array values

         //var $q = the actor array
         //var  $k = actor / task association array

         $q = 1;

         $k = 1;

//A Maximum of 4 actors can be associated with each task
```

```
//Therefore, run the nested while loop 4 times to
see if we have an association with the task that has
been submitted by the user

        while($c <= 4){



        $actor[$q] =
    $_POST['actorAssoc'."_".$k."_".$b];



//For each task / actor association, create a new
record

        $Query = "INSERT INTO $table_7 VALUES
    ('0','".$task[$b]."', '".$actor[$q]."',
    '$Owner_ID')";



        mysql_db_query ($DBName, $Query, $Link);

        $c++ ;

        $k++;

        $q++;

        }//close nested while loop






        $s++;
```

```
          $b++;

          }//Close while loop


//Invoke Subsystem Usage Data

          global $Subsystem ;

          $Subsystem = "add task to actor
association";

          include "../classes/class.Log.php";

          //End Usage Data


          //Determine if function has been called
from student dashboard or from the initial setup
screen

          if($fromAdmin==true){

          //Return to student Dashboard

          $message = "Tasks and Actors Associated
Successfully! ";

          header("Location:
../index.php?message=$message");


          }else{

          //Move on one step in the setup process
```

```
        $message = "Tasks and Actors Associated
Successfully! ";

        header("Location:
../Functions.php?message=$message");

        }


    }//Close Function


//////////////////////////////////////////////////

//////////////////////////////////////////////////

// EditAssociation - Actors And Tasks Link

//on WURF_WebAppTaskAssoc

//////////////////////////////////////////////////

//////////////////////////////////////////////////

function EditAssociation($Host, $User, $Password,
$DBName, $table_1, $table_2, $table_3, $table_4,
$table_5, $table_6, $table_7, $table_8 ){

$Link = mysql_connect($Host, $User, $Password);

$Owner_ID  = $_SESSION['Owner_ID'] ;


//We Need To Collect The Task ID And the next 4
Actor ID's.
```

```
          $task=array();

          $e=1;

          while($_POST['task'.$e]){

          $task[$e]=$_POST['task'.$e];

          //echo $image[$i].'  ';

          $e++;

          }

//We Are Going To Have build this function
differently,

//But we DO have a known quantity for the Tasks!
//Lets collect the Tasks using the SELECT query in
EditAssociate_WebAppTaskTo_Actors.php

//Then do a while loop 4 times to collect each
actor!


//Get Our Task List Again so that we know what we
are expecting and can use result set as a comparison

    $Query = "SELECT * FROM $table_5 WHERE Owner_ID
= '$Owner_ID' ORDER BY 'WebAppTask_ID' ASC";


    $Result = mysql_db_query($DBName, $Query,
$Link);

    while ($Row = mysql_fetch_array($Result)){
     $taskID = $Row['WebAppTask_ID'];
     $taskName = $Row['taskName'];


     }//Close while loop
//We Need To Collect The Task ID And the next 4
Actor ID's that have been submitted by the user
         $task=array();
         $e=1;
         while($_POST['task'.$e]){
         $task[$e]=$_POST['task'.$e];
         //echo $image[$i].'  ';
         $e++;
         }//close while loop



//var $b will count the tasks submitted by the user
         $b = 1;

//var $s will be our comparison operator
         $s = 1;
```

```
//Collect the task from the array

        while($s<=$e){


//For collecting our multi-dimensional array values

        $q = 1;

        $c = 1;

        $k = 1;

        $once = true ;


//We have a fixed ability for 4 actors to be
associated with each task

//Run the loop 4 times to see if we have an
association with the task that has been submitted

while($c <= 4){


//SELECT ON $task[$b] to find the
TaskToActorAssoc_ID // But We will need to a switch
to say record + 1;


        if($once==true){
         $Query = "SELECT TaskToActorAssoc_ID FROM
$table_7 WHERE WebAppTask_ID = '".$task[$b]."' &&
Owner_ID = '$Owner_ID' ORDER BY 'WebAppTask_ID'
ASC";

         }else{

//Then select the next 3 TaskToActorAssoc_ID in
sequence

         $ID = $TaskToActorAssoc_ID +1 ;

         $Query = "SELECT TaskToActorAssoc_ID FROM
$table_7 WHERE TaskToActorAssoc_ID = '".$ID."' ";

         }//close conditional statement

         $Result = mysql_db_query($DBName, $Query,
$Link);

         $Row = mysql_fetch_array($Result);
```

```
        $TaskToActorAssoc_ID =
$Row['TaskToActorAssoc_ID'];


        $actor[$q] =
$_POST['actorAssoc'."_".$k."_".$b];


        $once = false ;

        //UPDATE THE RECORD


        $Query2 = "UPDATE $table_7 SET
WebAppTask_ID = '".$task[$b]."', WebAppActor_ID =
'".$actor[$q]."' WHERE TaskToActorAssoc_ID =
'".$TaskToActorAssoc_ID."'";





        mysql_db_query ($DBName, $Query2, $Link);

        $c++ ;

        $k++;

        $q++;

        }//close nested while loop
        $s++;

        $b++;

        }//close while loop


//Invoke Subsystem Usage Data

        global $Subsystem ;

        $Subsystem = "edit task to actor
association";

        include "../classes/class.Log.php";

//End Usage Data

        $message = "Tasks and Actors Association
Updated Successfully!";

        header("Location:
../index.php?message=$message");


        }//Close Function




?>
```

Appendix C2 – RequirementDocumentationPrint Class

```
<?php

session_start();

//class.requirementDocumentationPrint.php    //



////////////////////VERSION 4.0 ///////////////////

//////////////////////////////////////////////////

include ('../common.php');

DBConnect();



GenerateDocumentation($Host, $User, $Password,
$DBName, $table_1, $table_2, $table_3, $table_4,
$table_5, $table_6, $table_6_A, $table_7, $table_8,
$table_9, $table_10, $table_11, $table_12,
$table_13, $table_14, $table_14_A, $table_15,
$table_16, $table_17);

//////////////////////////////////////////////////

//////////////////////////////////////////////////

//Development Notes 12/08/09
```

```
// This Function Generates The Requirement
Documentation - Printable Version*******

//This Class is called by the following class:
class.requirementPatterns.php

//Initial Prototype to Produce HTML

//This can be converted into a PDF file later if
required, once PDF extension is installed on server.

//////////////////////////////////////////////////

//////////////////////////////////////////////////




function GenerateDocumentation($Host, $User,
$Password, $DBName, $table_1, $table_2, $table_3,
$table_4, $table_5, $table_6, $table_6_A, $table_7,
$table_8, $table_9, $table_10, $table_11, $table_12,
$table_13, $table_14, $table_14_A, $table_15,
$table_16, $table_17 ){


//Object Model For The Document

global $Doc_P1, $Doc_P2, $Doc_P3, $Doc_P4, $Doc_P5,
$title ;
```

```
// Rewind array's internal pointer to the first
element and returns the value of the first array
element.

function Array_Dimensional_Reset(&$arrRef) {

    foreach ($arrRef as $key=>$val) {

        if (is_array($val)) {

            $this->Array_Dimensional_Reset($val);

            reset($arrRef[$key]);

        }

    }

}


//include "checkLoginSession.php";

$Link = mysql_connect($Host, $User, $Password);

$Owner_ID  = $_SESSION['Owner_ID'] ;




//1. Project Details - First lets select the project
details


$Query = "SELECT t1.Owner_ID, t1.title,
t1.description, t2.firstN, t2.lastN, t2.email,
t2.Owner_ID FROM $table_16 AS t1, $table_1 AS t2
WHERE t2.Owner_ID = '$Owner_ID' AND t1.Owner_ID =
'$Owner_ID'";

$Result = mysql_db_query($DBName, $Query, $Link);

$Row = mysql_fetch_array($Result);

$title = $Row['title'] ;

$description = $Row['description'] ;

$email = $Row['email'] ;

$Owner_firstName = $Row['firstN'] ;

$Owner_lastName = $Row['lastN'] ;

//Use HEREDOC method to build the first object of
our document
$Doc_P1 = <<<DOC
<hr />
```

```
<div>Project Owner: <span
class="bluetext">$Owner_firstName $Owner_lastName
</span></div><br />

<div>Owner Email Address: <span
class="bluetext">$email@tees.ac.uk  </span></div><br
/>


<div>Project Description: <span
class="bluetext">$description</span></div>

//Close HEREDOC

DOC;


//2. Website Objectives


$Query = "SELECT * FROM $table_4 WHERE Owner_ID =
'$Owner_ID'";

$Result = mysql_db_query($DBName, $Query, $Link);

$i = 0 ;

$k = 1 ;

while ($Row = mysql_fetch_array($Result)){

$obj.$i = $Row['ObjName'] ;

$objOut = $objOut. "<div>Objective".$k.":<span
class='bluetext'>$obj.$i</span></div>";
$i++ ;

$k++ ;

}//close while loop



//3. Statement Of Purpose

$Query2 = "SELECT * FROM $table_3 WHERE Owner_ID =
'$Owner_ID'";

$Result2 = mysql_db_query($DBName, $Query2, $Link);

$Row2 = mysql_fetch_array($Result2);


$statementTXT = $Row2['statementTXT'] ;


$Doc_P2 = <<<DOC1

<hr />

<div>Statement Of Purpose AND Website
Objectives:</div><br />

<div><br />Statement Of Purpose: <span
class="bluetext">$statementTXT</span></div>

<br /><br />

$objOut

<hr />
```

```
//Close HEREDOC

DOC1;



//4. Task And Associated Actors For Project

//This is the tricky bit!!!!


$Query3 = "SELECT WebAppTask_ID, WebAppActor_ID,
Owner_ID FROM $table_7 WHERE Owner_ID =
'$Owner_ID'";


$Result3 = mysql_db_query($DBName, $Query3, $Link);

$i=1;

$Doc_P3 = "<br /><div>Tasks And Associated
Actors</div>";

$z = 1 ;


//Get The Task ID's

//LOOP A

while ($Row3 = mysql_fetch_array($Result3)){

$TaskName[] = $Row3['WebAppTask_ID'];

$myActor[] = $Row3['WebAppActor_ID'];
```

```
$z++;

}//CLOSE LOOP A


//We Now Have Our Comparison Data Which We Can Use
In The Loop To Look For The Actors Associated With
Each Task

//Remove duplicate values from the array, so we only
print off consistent requirements, not duplicates.

$result = array_unique($TaskName);

$j=1;

$a=1;

//LOOP B

for ($i=0;$i<=$z; $i++){

//Select The WebActors Table

if(empty($result[$i])){

//null value returned, ensure we don’t see this in
our document

}else{

//Now We Look Up The 4 Actors Associated For This
Task

$Query3_3 = "SELECT WebAppActor_ID, WebAppTask_ID
FROM $table_7 WHERE WebAppTask_ID = '$result[$i]'";
```

```
$Result3_3 = mysql_db_query($DBName, $Query3_3,
$Link);
$h = 1;



//LOOP C
while($Row3_3 = mysql_fetch_array($Result3_3)){
$ActorKey = $Row3_3['WebAppActor_ID'];
$TaskKey =  $Row3_3['WebAppTask_ID'];

//***Additional Query To Get The Task Name (2/9/09)
$Query3_3_4 = "SELECT WebAppTask_ID, taskName FROM
$table_5 WHERE WebAppTask_ID = '$TaskKey'";
$Result3_3_4 = mysql_db_query($DBName, $Query3_3_4,
$Link);
$Row3_3_4 = mysql_fetch_array($Result3_3_4);
$Taskname = $Row3_3_4['taskName'];

//Now We Can Do A SELECT To Return Actor Data
$Query3_3_3 = "SELECT WebAppActor_ID, actorName FROM
$table_6 WHERE WebAppActor_ID = '$ActorKey'";
$Result3_3_3 = mysql_db_query($DBName, $Query3_3_3,
$Link);

$Row3_3_3 = mysql_fetch_array($Result3_3_3);

if(empty($Row3_3_3['actorName'])){

//Null record returned

}else{

$ActorName_Task.$a = $Row3_3_3['actorName'];
$myActorList_Task = $myActorList_Task."
".$ActorName_Task.$a.", " ;


$h ++;

$a++;






}//Close if empty conditional statement

}//CLOSE LOOP C


$a =1;
$Doc_P3 = $Doc_P3."Task Name: <span
class='bluetext'>
$Taskname</span>"."&nbsp;&nbsp;Actor: <span
```

```
class='bluetext'>$myActorList_Task</span>"."<br />"
;


$myActorList_Task = "";

}//Close if empty conditional statement

$j++;

}//CLOSE LOOP B


$g=1;

$s =1;

$i++;

//****END OF TASK & ACTORS ASSOCIATION******//

//5. Functions And Actor Associations.

$Query3 = "SELECT WebAppFunction_ID, WebAppActor_ID,
Owner_ID FROM $table_10 WHERE Owner_ID =
'$Owner_ID'";


$Result3 = mysql_db_query($DBName, $Query3, $Link);

$i=1;


$Doc_P4 = "<br /><div>Functional Req's And
Associated Actors</div>";
```

```
$z = 1 ;



//Get The Task ID's

//LOOP A

while ($Row3 = mysql_fetch_array($Result3)){

$FunctionName[] = $Row3['WebAppFunction_ID'];

$myActor[] = $Row3['WebAppActor_ID'];

$z++;

}//CLOSE LOOP A

//We Now Have Our Unique Number Which We Can Use In
The Loop To Look For The Actors Associated With The
Function

$result = array_unique($FunctionName);

$j=1;

//LOOP B

for ($i=0;$i<=$z; $i++){

//Select The WebActors Table

if(empty($result[$i])){

//Null data returned, ensure this is not displayed
in the documentation

}else{
```

```
//Now We Look Up The Actors For Just That Task

$Query3_3 = "SELECT WebAppActor_ID,
WebAppFunction_ID FROM $table_10 WHERE
WebAppFunction_ID = '$result[$i]'";

$Result3_3 = mysql_db_query($DBName, $Query3_3,
$Link);

$h = 1;





//LOOP C

while($Row3_3 = mysql_fetch_array($Result3_3)){


$ActorKey = $Row3_3['WebAppActor_ID'];

$FunctionKey = $Row3_3['WebAppFunction_ID'];

//***Additional Query To Get The Function Name
(2/9/09)

$Query3_3_4 = "SELECT WebAppFunction_ID,
functionText FROM $table_9 WHERE WebAppFunction_ID =
'$FunctionKey'";
```

```
$Result3_3_4 = mysql_db_query($DBName, $Query3_3_4,
$Link);

$Row3_3_4 = mysql_fetch_array($Result3_3_4);


$Functionname = $Row3_3_4['functionText'];

//Now We Can Do SELECT To LOOK UP Who The Actor Is
On The Actors Table

$Query3_3_3 = "SELECT WebAppActor_ID, actorName FROM
$table_6 WHERE WebAppActor_ID = '$ActorKey'";


$Result3_3_3 = mysql_db_query($DBName, $Query3_3_3,
$Link);

$Row3_3_3 = mysql_fetch_array($Result3_3_3);

if(empty($Row3_3_3['actorName'])){

//Null data returned

}else{

$ActorName_Function.$a = $Row3_3_3['actorName'];

$myActorList_Function = $myActorList_Function."
".$ActorName_Function.$a.", " ;

$h ++;

$a ++;
}//Close If EMPTY



}//CLOSE LOOP C

$a =1;

$Doc_P4 = $Doc_P4. "Functional Req's: <span
class='bluetext'>$Functionname</span>"."&nbsp;&nbsp;
Actors: <span
class='bluetext'>$myActorList_Function</span>"."<br
/>" ;

$myActorList_Function = "";

}//CLOSE IF EMPTY

$j++;

}//CLOSE LOOP B


$g=1;

$s =1;

$i++;

//****END OF TASK & ACTORS ASSOCIATION******//

/* This fix is to align the id in the correct order
in the RequirementPattern table to ensure functional
```

```
requirement id’s are in the correct order within the
requirements document*/

$Query_alter = "ALTER TABLE $table_17 ORDER BY
RequirementPattern_ID ASC";

mysql_db_query($DBName, $Query_alter, $Link);


///Now We Generate Each Requirement Pattern



$Query5 = "SELECT * FROM $table_17 WHERE Owner_ID =
'$Owner_ID' ORDER BY RequirementPattern_ID ASC";


$Result5 = mysql_db_query($DBName, $Query5, $Link);


$Doc_P5 = "<div class='breakhere'></div><hr /><br />

<br /><div class='bluetextbig'>Requirements
Specification Document</div><br /><div
class='bluetextsmall'>Key: FR-x = Functional
Requirement || NFR-x = Non-Functional
Requirement</div>";


$p = 1 ;

while ($Row5 = mysql_fetch_array($Result5)){
```

```
$FunctionnameQ = $Row5['name'] ;

$Doc_P5 = $Doc_P5. "<div class='PatContainer'>";


$Doc_P5 = $Doc_P5. "<div
class='Row1Left'>$Row5[ReqRef]</div>";

//We can tailor the description feild here

$type = $Row5['type'] ;


if($type=="Non-Functional Requirement"){

$name = $Row5['name'];

$description = $Row5['description'];

}else{

$name = "&nbsp;&nbsp;&nbsp;Functional Requirement
$p";

$description = $FunctionnameQ ;

}


$Doc_P5 = $Doc_P5. "<div
class='Row1Cent'>$name</div>";


$Doc_P5 = $Doc_P5. "<div
class='Row1Right'>$Row5[type]</div>";
```

```
$Doc_P5 = $Doc_P5. "<div class='Row2'>Description:
<span class=bluetext> $description</span></div>" ;


//*****Here we need to do a Sub Query to get the
Task Name for the Function ID.****

//*****Here we need to do a Sub Query to get the
Actor Name .****

$Actor1 = $Row5['Actor1_ID'];

$Actor2 = $Row5['Actor2_ID'];

$Actor3 = $Row5['Actor3_ID'];

$Actor4 = $Row5['Actor4_ID'];

if((empty($Actor1)) && (empty($Actor2)) &&
(empty($Actor3)) && (empty($Actor4))){

$actorEmpty = true ;

}

$a =1 ;

$s = 1;

for ($f=0;$f<=3; $f++){

if($s==1){

$q = "WebAppActor_ID = '$Actor1'";

}else if($s==2){
```

```
$q = "WebAppActor_ID = '$Actor2'";


}else if($s==3){

$q = "WebAppActor_ID = '$Actor3'";


}else{

$q = "WebAppActor_ID = '$Actor4'";

}

//Lets Look Up Who The Actor is

$Query5_2 = "SELECT WebAppActor_ID, actorName FROM
$table_6 WHERE ".$q."";

$Result5_2 = mysql_db_query($DBName, $Query5_2,
$Link);

$Row5_2 = mysql_fetch_array($Result5_2);

$ActorName.$a = $Row5_2['actorName'];

$WebAppActor_ID = $Row5_2['WebAppActor_ID'];

//Check that there is a profile for the actor

$Query5_3 = "SELECT WebAppActor_ID, actorName FROM
$table_6_A WHERE ".$q."";

$Result5_3 = mysql_db_query($DBName, $Query5_3,
$Link);

$Row5_3 = mysql_fetch_array($Result5_3);
```

```
$check = $Row5_3['WebAppActor_ID'];

if(empty($check)){

$myActorList = " ".$myActorList.$ActorName.$a.", " ;


}else{

$myActorList = " ".$myActorList." "."<a
href='../ViewActorProfile.php?WebAppActor_ID=$WebApp
Actor_ID' target='_blank'>".$ActorName.$a."</a>".",
" ;

}

//Lets see if we can have a statement if no actors
are assigned

if($actorEmpty== true){

$myActorList = "No Actors Assigned!";

}

$s = $s +1;

$a++;

}


$a =1;

$s=1;

$type = $Row5['type'] ;
```

```
if($type=="Non-Functional Requirement"){


$myActorList = "Not Applicable To Non-functional
Requirements";

$addinfo = "";

$comments = "Additional Req's: <span
class=bluetext>".$Row5['comments']."</span>" ;

}else{

$comments = "Comments:<span
class=bluetext>".$Row5['comments']."</span>" ;

$addinfo = "<span class=bluetextsmall>  <br />These
Actors will need to be taken into consideration when
designing/developing this requirement!</span>";

}


$Doc_P5 = $Doc_P5. "<div class='Row4'>Associated
Actors: <span class=bluetext>".$myActorList."
$addinfo</span></div>";


//Now the comments from the templatePattern table


$Doc_P5 = $Doc_P5. "<div class='Row5'>
$comments</div>";
```

```
$Doc_P5 = $Doc_P5. "</div><div
class='breakhere'></div>";

$p++;

$myActorList="";

}

//T2;




//Invoke Log and Usage data

global $Subsystem ;

$Subsystem = "create requirements document print
version";

include "../classes/class.Log.php";

//End Invoke Log

}//Close GeneratePatterns Function

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
```

```
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />


<link rel="stylesheet" type="text/css"
href="../styles1.css"/>
<style type="text/css">
<!--
body{
background-image:none;
}
.txt1{
     font-size: 17px;
     color:#333333;
     width: 100%;
}




/*Requirement Pattern CSS */
.PatContainer{
position:relative;
height: 300px;

width: 600px;

/*background-color:#69F;*/


border-left-width:medium;

border-left-color:#333;

border-left-style:solid;


border-top-width:medium;

border-top-color:#333;

border-top-style:solid;


border-bottom-width:medium;

border-bottom-color:#333;

border-bottom-style:solid;


border-right-width:medium;

border-right-color:#333;

border-right-style:solid;

margin-top: 5px;

}
```

```
.Row1Left{

position:absolute;

left: 0px;

top:0px;

height: 50px;

width:130px;


background-color:#d7effb;


border-right-color:#333;

border-right-style:dotted;

border-right-width:thin;


color:#0095e2;

font-weight:bold;


}
```

```
.Row1Cent{

position:absolute;

left: 131px;

top:0px;

height: 50px;

width:338px;


background-color:#d7effb;


border-right-color:#333;

border-right-style:dotted;

border-right-width:thin;


color:#0095e2;

font-weight:bold;


}

.Row1Right{

position:absolute;

left: 470px;

top:0px;
```

```
height: 50px;

width:130px;

background-color:#d7effb;

color:#0095e2;

font-weight:bold;

}

.Row2{

position:absolute;

left: 0px;

top: 44px;

height: 136px;

width: 599px;

background-color:#FFF;

border-top-color:#333;

border-top-style:dotted;

border-top-width:thin;

border-bottom-color:#333;

border-bottom-style:dotted;

border-bottom-width:thin;

}

.Row3{
```

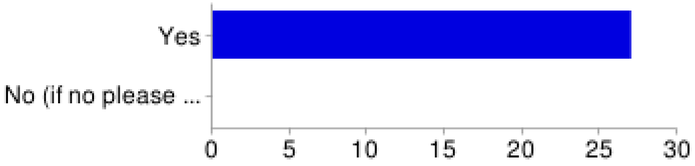
```
position:absolute;

left: 0px;

top: 146px;

height: 50px;

width: 100%;

background-color:#9CF;

}

.Row4{

position:absolute;

left: 0px;

top: 184px;

height: 50px;

width: 100%;

border-bottom-color:#333;

border-bottom-style:dotted;

border-bottom-width:thin;

}

.Row5{

position:absolute;

left: 0px;

top: 236px;
```

```
height: 58px;

width: 100%;

}

.breakhere {page-break-before: always}

.greyTxt{

color:#999;

}

-->

</style>

<title>Requirements Documentation</title>

</head>

<body>

<div class="greyTxt">Please Note: This is a Printer
Friendly Page. Each Requirement will be printed off
individually to ensure correct formatting</div>

<h2>Requirements Documentation For: <span
class="bluetext"><?php echo $title ; ?></span></h2>

<!-- Output Requirements Documentation Here -->

<?php print $Doc_P1 ; print $Doc_P2 ; print $Doc_P3
; print $Doc_P4 ; print $Doc_P5 ;?>

</body>

</html>
```

Appendix C3 – Student Opinion Survey

OBS Module Responses:

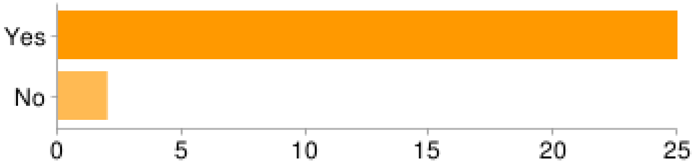
Q1. Have You Used e-WURF for the Online Business Systems Module?



Yes: 100%

No: 0%

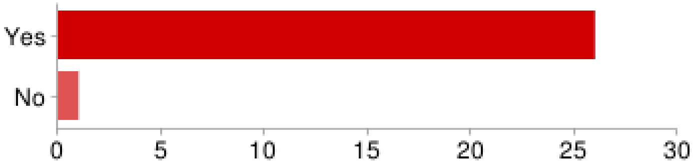
Q2. Have you been able to produce a requirements specification document?



Yes: 93%

No: 7%

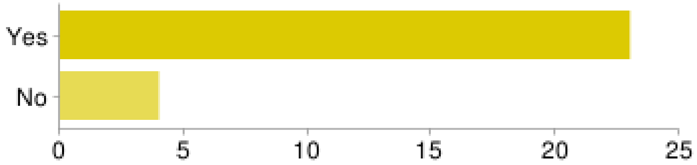
Q3. Do you feel that the web development process has been enhanced by using e-WURF?



Yes: 96%

No:4%

Q4. Did you use the e-WURF assistant?

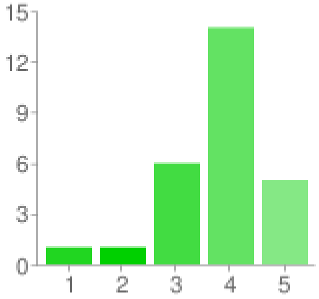


Yes: 85%

No: 15%

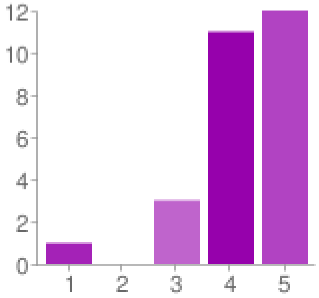
Q5. Please rate the following in terms of usefulness in the e-wurf process

Statement Of Purpose



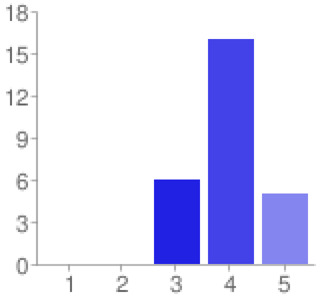
1 - least useful -	4%
2	4%
3	22%
4	52%
5 - most useful	-19%

Objectives



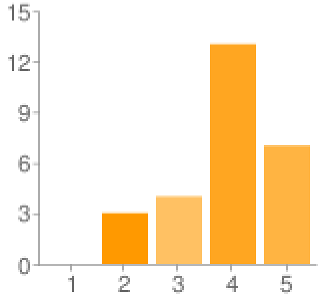
1 - least useful -	4%
2	0%
3	11%
4	41%
5 - most useful	-44%

Tasks



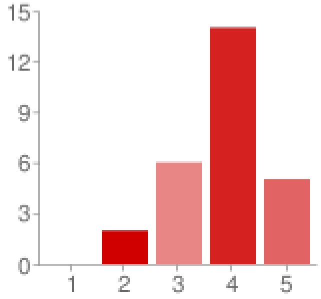
1 - least useful -	0%
2	0%
3	22%
4	59%
5 - most useful	-19%

Actors and Actor Profiles



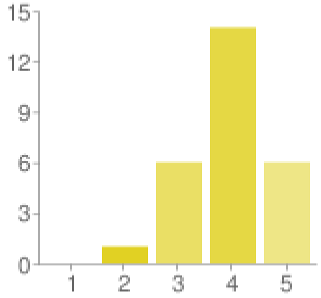
1 - least useful -	0%
2	11%
3	15%
4	48%
5 - most useful	-26%

Task To Actors Association



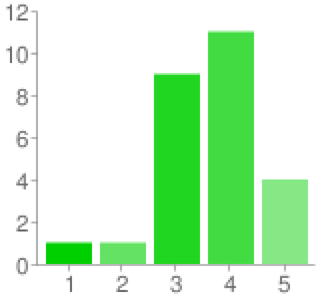
1 - least useful -	0%
2	7%
3	22%
4	52%
5 - most useful	-19%

Functional Requirements



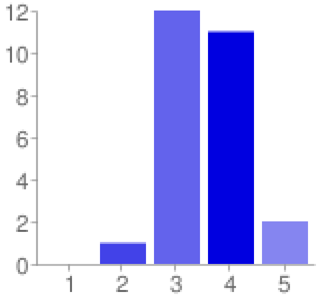
1 - least useful -	0%
2	4%
3	22%
4	52%
5 - most useful	-22%

Functional Req to Actor Association



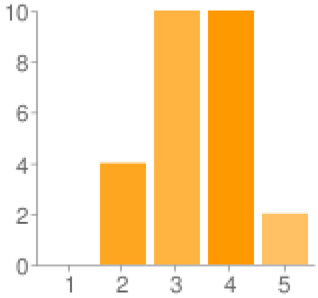
1 - least useful -	4%
2	4%
3	35%
4	42%
5 - most useful	-15%

User Interface (non-functional Req)



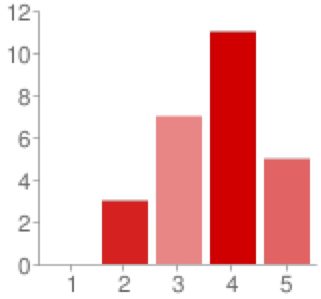
1 - least useful -	0%
2	4%
3	46%
4	42%
5 - most useful	-8%

Marketing (non-functional Req)



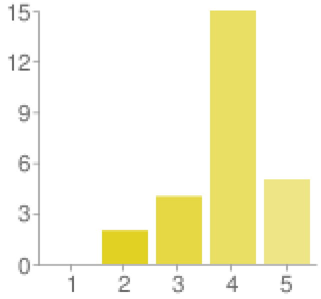
1 - least useful -	0%
2	15%
3	38%
4	38%
5 - most useful	-8%

Usability (non-functional Req)



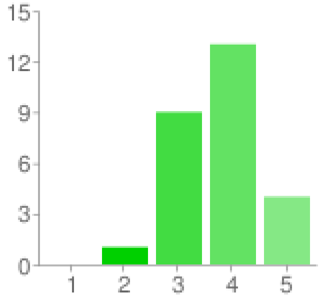
1 - least useful -	0%
2	12%
3	27%
4	42%
5 - most useful	-19%

Accessibility (non-functional Req)



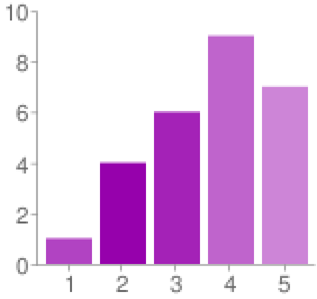
1 - least useful -	0%
2	8%
3	15%
4	58%
5 - most useful	-19%

Technical (non-functional Req)



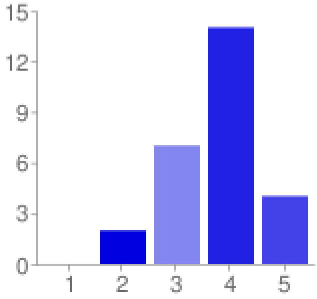
1 - least useful -	0%
2	4%
3	33%
4	48%
5 - most useful	-15%

e-WURF Assistant



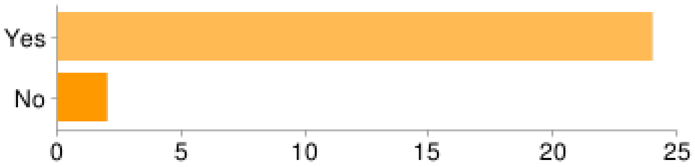
1 - least useful - 4%
2 15%
3 22%
4 33%
5 - most useful -26%

Requirements Specification Document



1 - least useful - 0%
2 7%
3 26%
4 52%
5 - most useful -15%

Did you understand the difference between functional and non-functional requirements after you had used eWURF

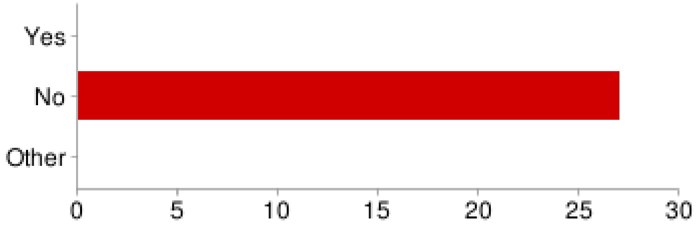


Yes: 92%

No: 8%

Q6. Did you encounter any difficulties with the following

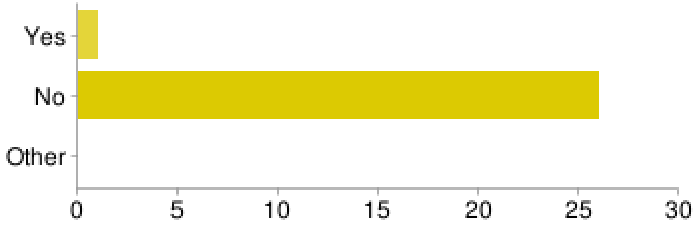
Statement Of Purpose



Yes: 0%

No: 100%

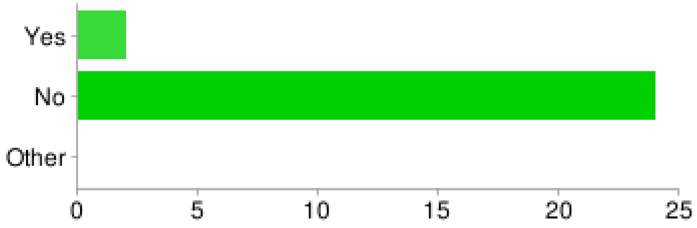
Objectives



Yes: 4%

No: 96%

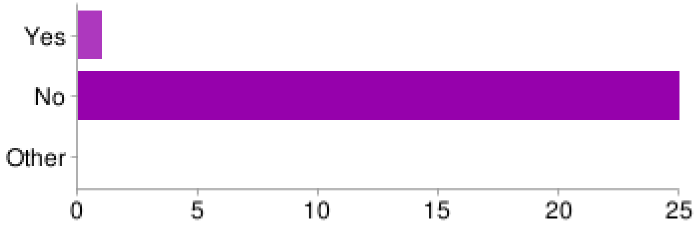
Tasks



Yes: 8%

No: 92%

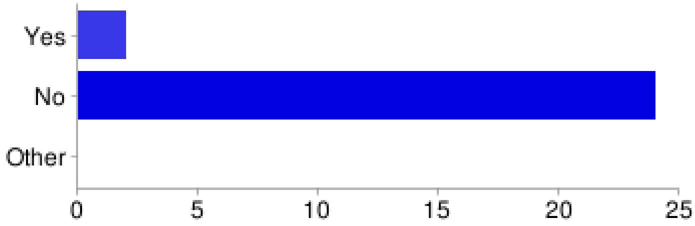
Actors and Actor Profiles



Yes: 4%

No: 96%

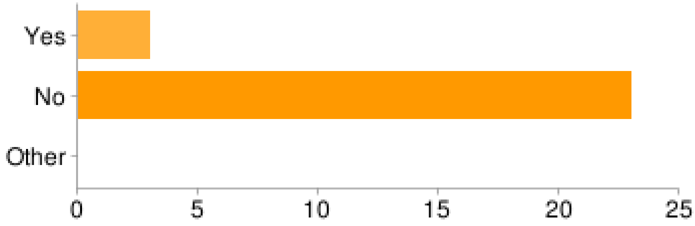
Task To Actors Association



Yes: 8%

No: 92%

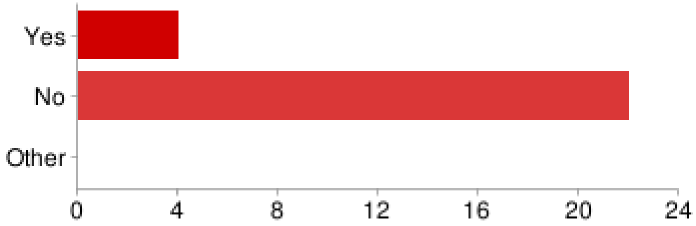
Functional Requirements



Yes: 12%

No: 88%

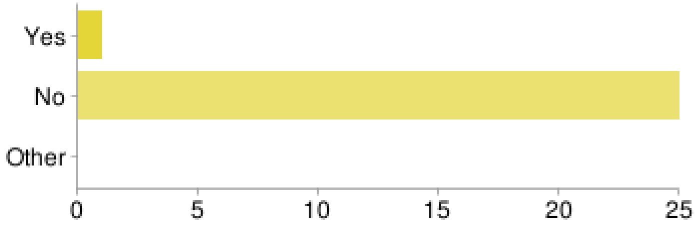
Functional Req to Actor Association



Yes: 15%

No: 85%

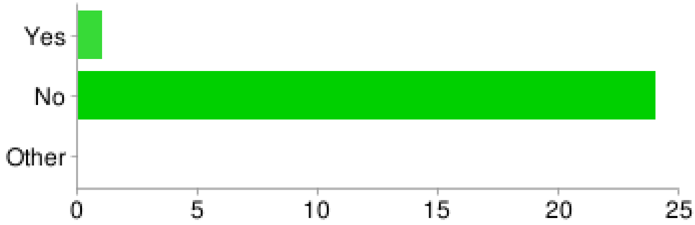
User Interface (non-functional Req)



Yes: 4%

No: 96%

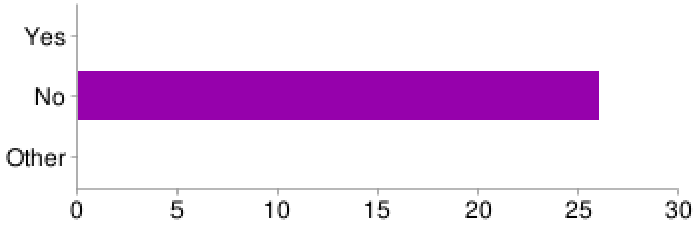
Marketing (non-functional Req)



Yes: 4%

No: 96%

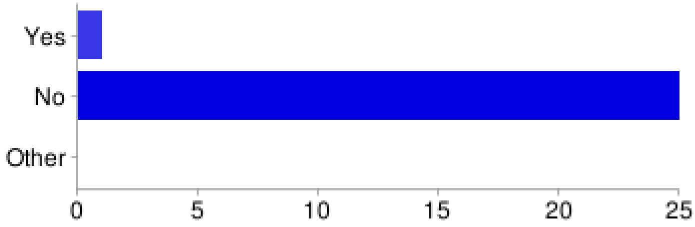
Usability (non-functional Req)



Yes: 0%

No: 100%

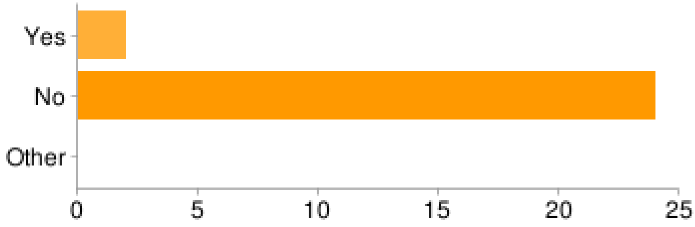
Accessibility (non-functional Req)



Yes: 4%

No: 96%

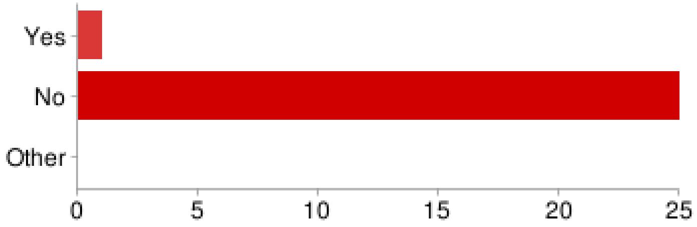
Technical (non-functional Req)



Yes: 8%

No: 92%

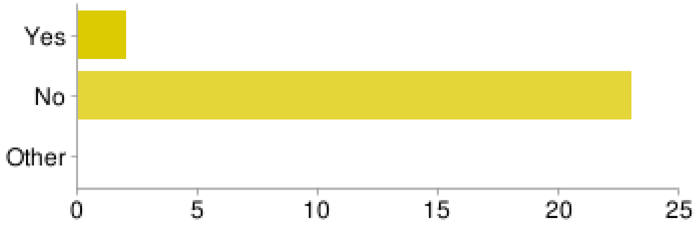
e-WURF Assistant



Yes: 4%

No: 96%

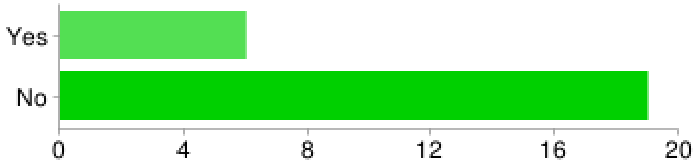
Requirements Specification Document



Yes: 8%

No: 92%

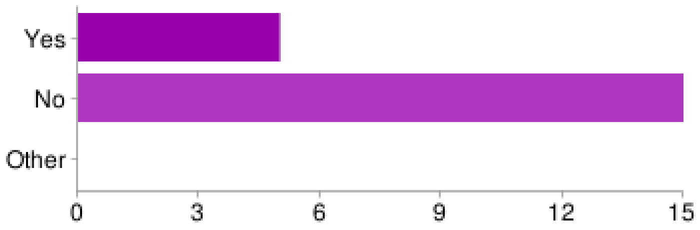
If you answered yes to any of the above, did you contact e-WURF about this



Yes: 24%

No: 76%

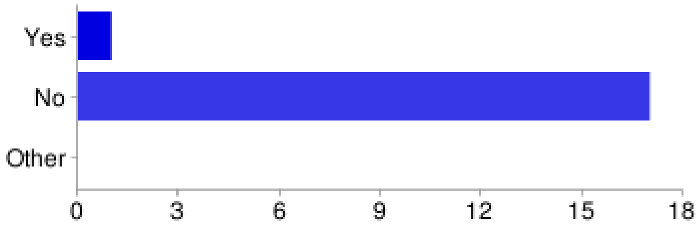
If you answered yes to the above, was the problem resolved?



Yes: 25%

No: 75%

If you did not use e-WURF, have you used another requirements tool?



Yes: 6%

No: 94%

Are there any additions/modifications that you would like to see? Please enter them in the box below (Student Responses unedited).

- if you enter data into one of the sections and exit by mistake it should save it and say its unfinished, since i clicked the banner a number of times by mistake.

save link as .doc, so can print document in word. heard of difficulties printing from other students.

-To close the help dialog box you have to scroll back to the top to select the close link. A additional close link at the bottom would help prevent this.

- Better english skills being utilised on the site, example "build" being used where "Built" should have been.

- Consider the diverse browser (compatibility) or the platforms the website will run on as well as the requirements for mobile applications.

-When using the e-WURF Assistant you can only close the popup box by scrolling back to the top of the page. A bookmark back to the top of the page or close function in the footer would prevent additional scrolling.

-The whole structure felt a bit too rigid, giving no flexibility for any of the requirements.

- The ability to group 'Actors' into primary and secondary (even tertiary) target audience groups. This would make it easy to identify the most important actors.

-Add a PDF Format to see the document on the screen without necessary have an internet connection.

-No this is ok

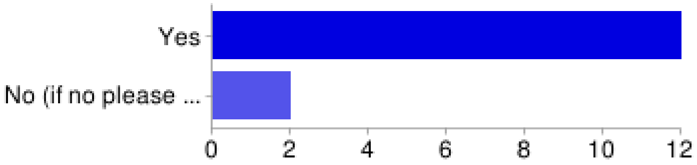
- Consider carefully where radio buttons and where check-boxes are appropriate.

- Clarification on how to remove tasks. Formatting on associated tasks another areas could be improved, this maybe a Safari bug, a lot of scrolling is required. Sometimes radio buttons are not on the same line as the related answer, this can be seen in the Actor Profile page. I also nearly missed out creating Actor Profiles as they did not show under the Incomplete tasks on the home page.

- Although I managed to complete the accessiblity and usability non-functional requirements sections I found that it could have been made clearer as to what information was trying to be received.

WAU Module Responses:

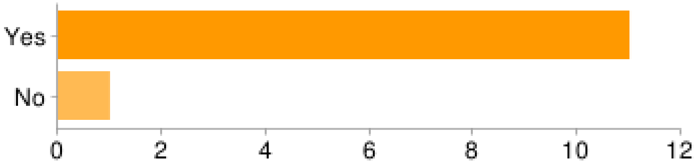
Q1. Have You Used e-WURF for the Web Authoring Module?



Yes: 86%

No: 14%

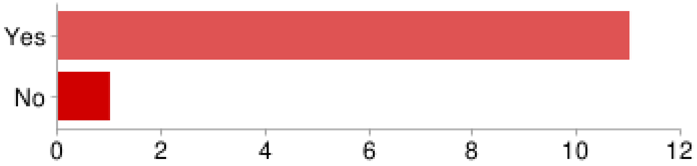
Q2. Have you been able to produce a requirements specification document?



Yes: 92%

No: 8%

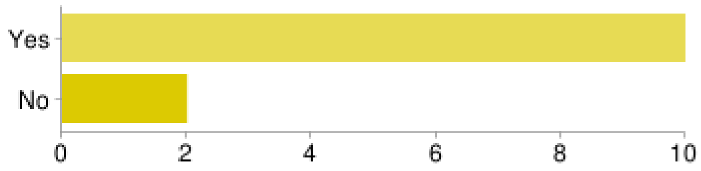
Q3. Do you feel that the web development process has been enhanced by using e-WURF?



Yes: 92%

No 8%

Q4. Did you use the e-WURF assistant?

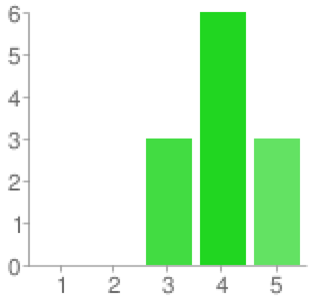


Yes: 83%

No: 17%

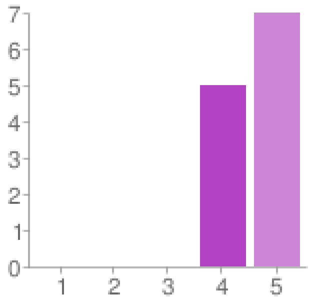
Q5. Please rate the following in terms of usefulness in the e-wurf process

Statement Of Purpose



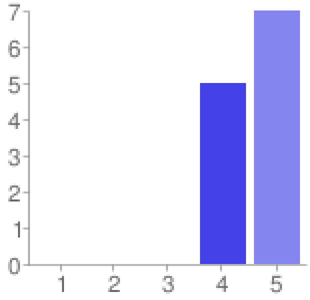
1 - least useful -	0%
2	0%
3	25%
4	50%
5 - most useful	-25%

Objectives



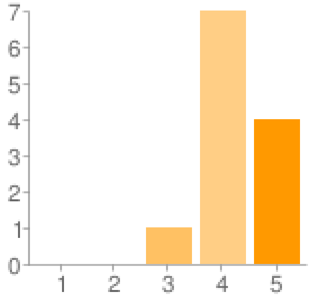
1 - least useful -	0%
2	0%
3	0%
4	42%
5 - most useful	-58%

Tasks



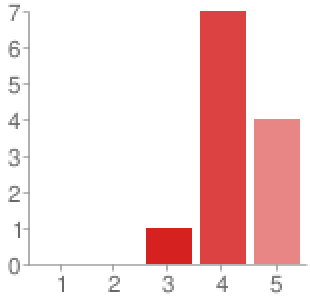
1 - least useful -	0%
2	0%
3	0%
4	42%
5 - most useful	-58%

Actors and Actor Profiles



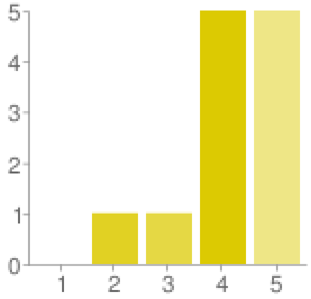
1 - least useful -	0%
2	0%
3	8%
4	58%
5 - most useful	-33%

Task To Actors Association



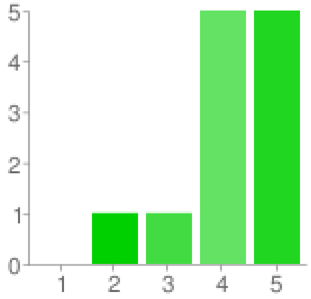
1 - least useful -	0%
2	0%
3	8%
4	58%
5 - most useful	-33%

Functional Requirements



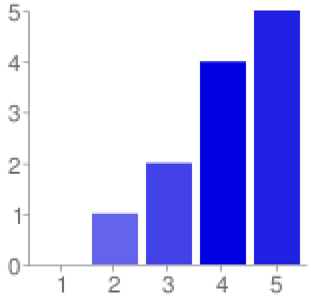
1 - least useful -	0%
2	8%
3	8%
4	42%
5 - most useful	-42%

Functional Req to Actor Association



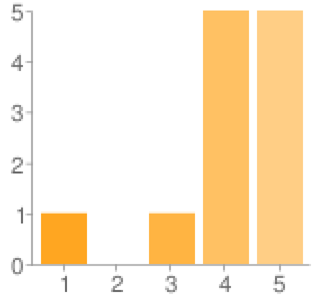
1 - least useful -	0%
2	8%
3	8%
4	42%
5 - most useful	-42%

User Interface (non-functional Req)



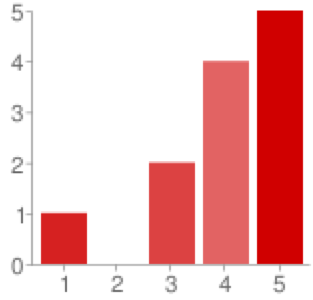
1 - least useful -	0%
2	8%
3	17%
4	33%
5 - most useful	-42%

Marketing (non-functional Req)



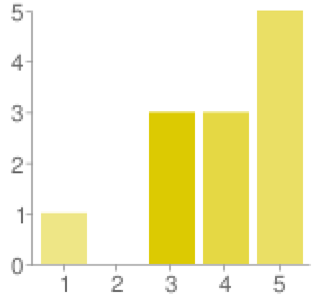
1 - least useful -	8%
2	0%
3	8%
4	42%
5 - most useful	-42%

Usability (non-functional Req)



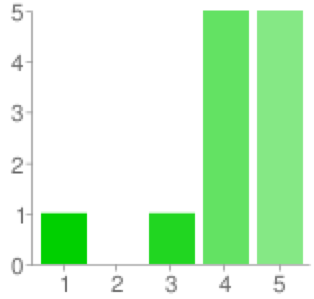
1 - least useful -	8%
2	0%
3	17%
4	33%
5 - most useful	-42%

Accessibility (non-functional Req)



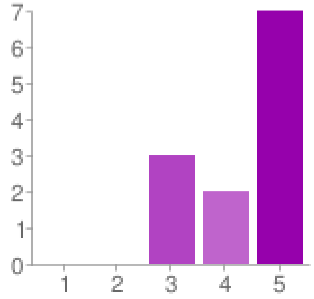
1 - least useful -	8%
2	0%
3	25%
4	25%
5 - most useful	-42%

Technical (non-functional Req)



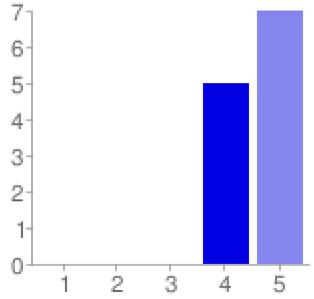
1 - least useful -	8%
2	0%
3	8%
4	42%
5 - most useful	-42%

e-WURF Assistant



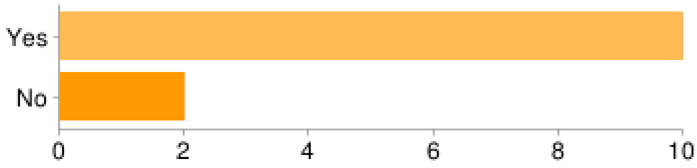
1 - least useful -	0%
2	0%
3	25%
4	17%
5 - most useful	-58%

Requirements Specification Document



1 - least useful -	0%
2	0%
3	0%
4	42%
5 - most useful	-58%

Did you understand the difference between functional and non-functional requirements after you had used eWURF

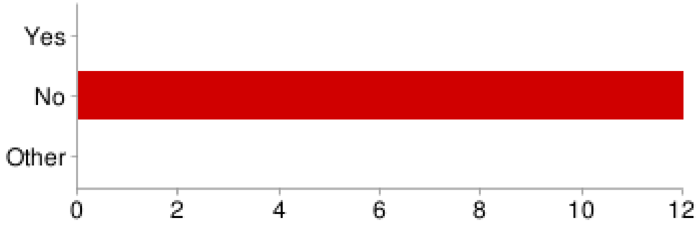


Yes: 83%

No: 17%

Q6. Did you encounter any difficulties with the following

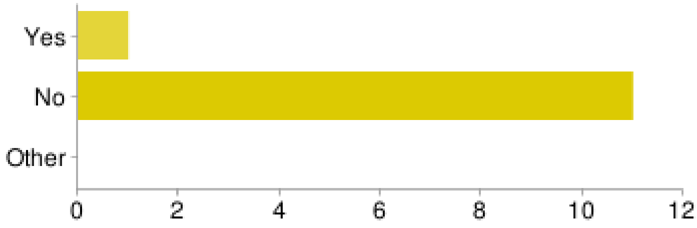
Statement Of Purpose



Yes: 0%

No: 100%

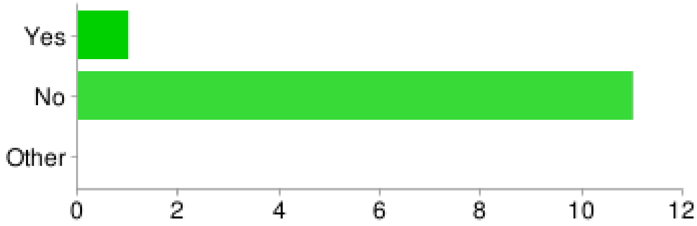
Objectives



Yes: 8%

No: 92%

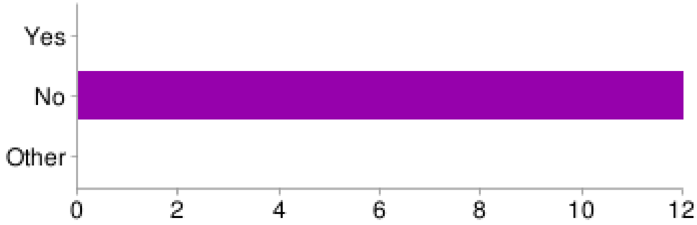
Tasks



Yes: 8%

No: 92%

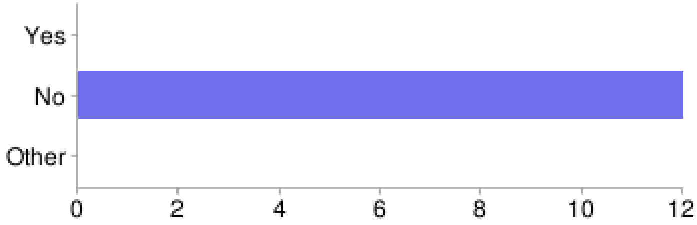
Actors and Actor Profiles



Yes: 0%

No: 100%

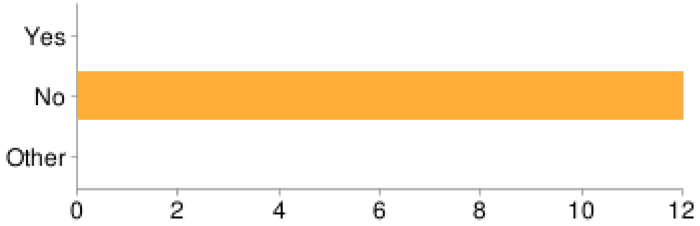
Task To Actors Association



Yes: 0%

No: 100%

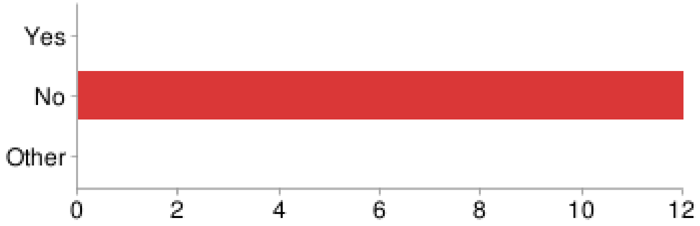
Functional Requirements



Yes: 0%

No: 100%

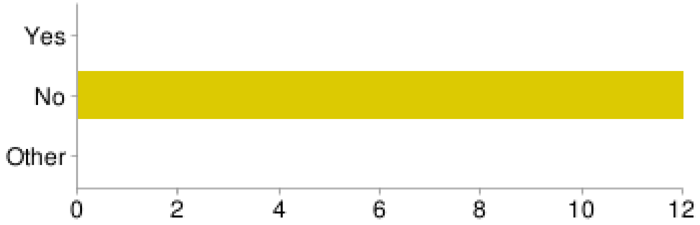
Functional Req to Actor Association



Yes: 0%

No: 100%

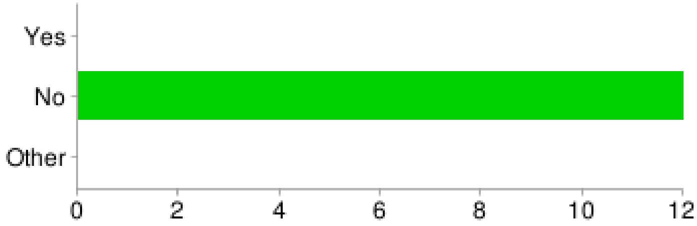
User Interface (non-functional Req)



Yes: 0%

No: 100%

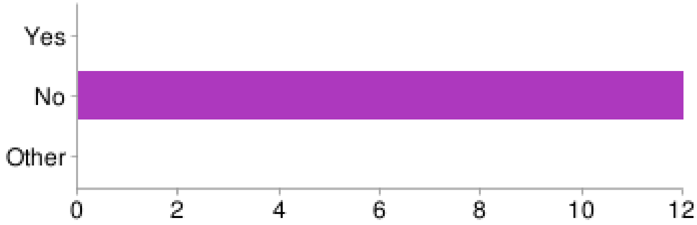
Marketing (non-functional Req)



Yes: 0%

No: 100%

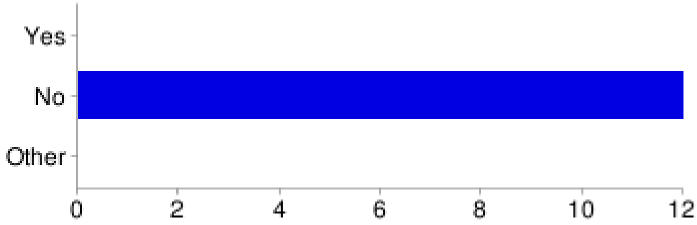
Usability (non-functional Req)



Yes: 0%

No: 100%

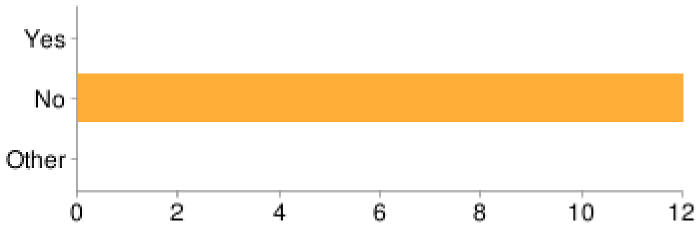
Accessibility (non-functional Req)



Yes: 0%

No: 100%

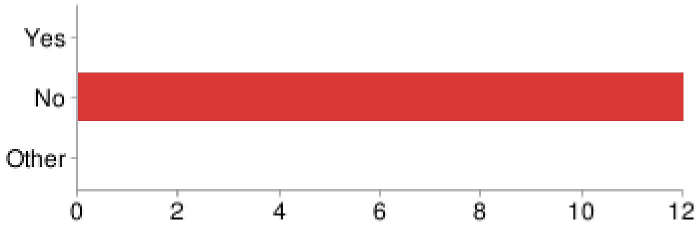
Technical (non-functional Req)



Yes: 0%

No: 100%

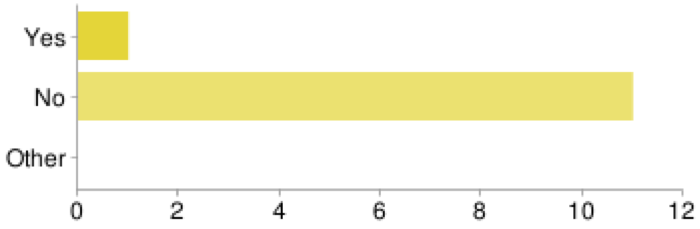
e-WURF Assistant



Yes: 0%

No: 100%

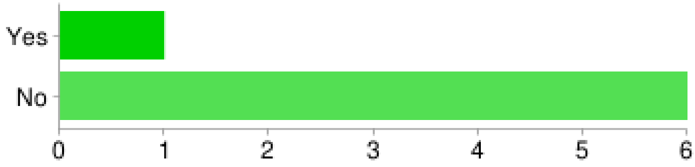
Requirements Specification Document



Yes: 8%

No: 92%

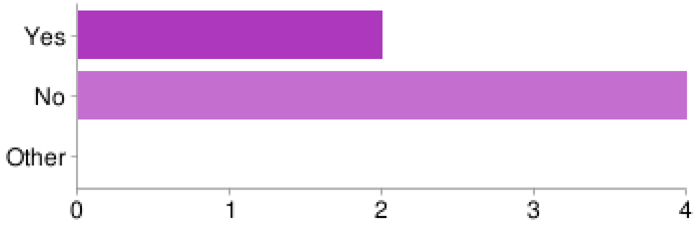
If you answered yes to any of the above, did you contact e-WURF about this?



Yes: 14%

No: 86%

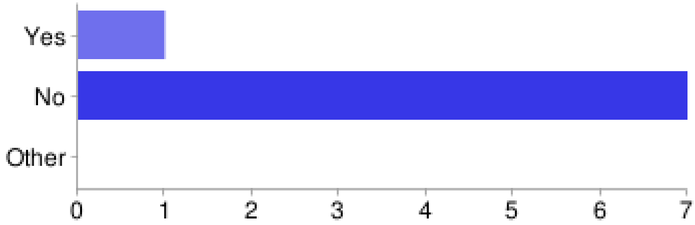
If you answered yes to the above, was the problem resolved?



Yes: 33%

No: 67%

If you did not use e-WURF, have you used another requirements tool?



Yes: 13%

No: 88%

Are there any additions/modifications that you would like to see? Please enter them in the box below (Student Responses unedited).

No, Very well developed.

When attempting to use the print safe option upon completion i found that if alot of information had been entered then the forms wouldnt accomodate it and so would not be displayed. Other than that it was a very useful assistance tool

Appendix C4 – Usage Data Tables

Module: OBS - Assessment Mark vs Usage Relationship Analysis

HA1: Increased usage of the CAWE tool will result in higher marks for requirements analysis.

HA0: Increased usage of the CAWE tool will not result in higher marks for requirements analysis.

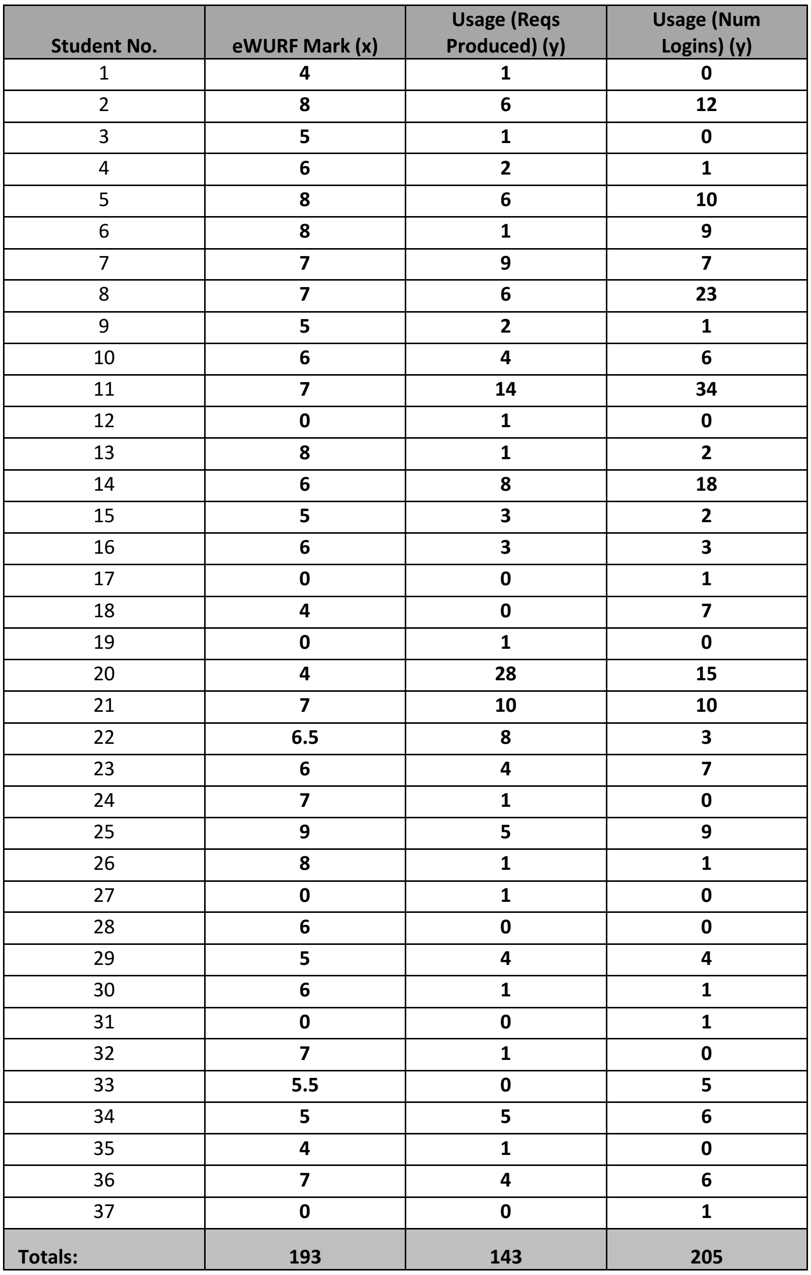
Test 1. Requirements documents produced vs Marks	
Spearman's Rank Correlation Coefficient r =	**0.226230711**
Statistical Significance of Correlation Two-Tailed T Test =	**0.325**
∴ null hypothesis (HA0) must be accepted	
Test 2. Frequency of Logins vs Marks	
Spearman's Rank Correlation Coefficient r =	**0.372954685**
Statistical Significance of Correlation Two-Tailed T Test =	**0.325**
∴ hypothesis (HA1) can be accepted	

Formula Used For Correlation Is:
(Spearman's Rank Correlation Coefficient)

n = number of pairs of scores
$\sum xy$ = sum of the products of paired scores
$\sum x$ = sum of x scores
$\sum y$ = sum of y scores
$\sum x_2$ = sum of squared x scores
$\sum y_2$ = sum of squared y scores

Analysis was undertaken on the following data sets:

Table 1. x and y scores



Student No.	eWURF Mark (x)	Usage (Reqs Produced) (y)	Usage (Num Logins) (y)
1	4	1	0
2	8	6	12
3	5	1	0
4	6	2	1
5	8	6	10
6	8	1	9
7	7	9	7
8	7	6	23
9	5	2	1
10	6	4	6
11	7	14	34
12	0	1	0
13	8	1	2
14	6	8	18
15	5	3	2
16	6	3	3
17	0	0	1
18	4	0	7
19	0	1	0
20	4	28	15
21	7	10	10
22	6.5	8	3
23	6	4	7
24	7	1	0
25	9	5	9
26	8	1	1
27	0	1	0
28	6	0	0
29	5	4	4
30	6	1	1
31	0	0	1
32	7	1	0
33	5.5	0	5
34	5	5	6
35	4	1	0
36	7	4	6
37	0	0	1
Totals:	193	143	205

Table 2. sum of squared x and y scores (Test 1 and Test 2)

(Test 1)				(Test 2)		
x*y	x*x	y*y		x*y	x*x	y*y
4	16	1		0	16	0
48	64	36		96	64	144
5	25	1		0	25	0
12	36	4		6	36	1
48	64	36		80	64	100
8	64	1		72	64	81
63	49	81		49	49	49
42	49	36		161	49	529
10	25	4		5	25	1
24	36	16		36	36	36
98	49	196		238	49	1156
0	0	1		0	0	0
8	64	1		16	64	4
48	36	64		108	36	324
15	25	9		10	25	4
18	36	9		18	36	9
0	0	0		0	0	1
0	16	0		28	16	49
0	0	1		0	0	0
112	16	784		60	16	225
70	49	100		70	49	100
52	42.25	64		19.5	42.25	9
24	36	16		42	36	49
7	49	1		0	49	0
45	81	25		81	81	81
8	64	1		8	64	1
0	0	1		0	0	0
0	36	0		0	36	0
20	25	16		20	25	16
6	36	1		6	36	1
0	0	0		0	0	1
7	49	1		0	49	0
0	30.25	0		27.5	30.25	25
25	25	25		30	25	36
4	16	1		0	16	0
28	49	16		42	49	36
0	0	0		0	0	1
859	1257.5	1549		1329	1257.5	3069

Module: WAU - Assessment Mark vs Usage Relationship Analysis

HA1: Increased usage of the CAWE tool will result in higher marks for requirements analysis.

HA0: Increased usage of the CAWE tool will not result in higher marks for requirements analysis.

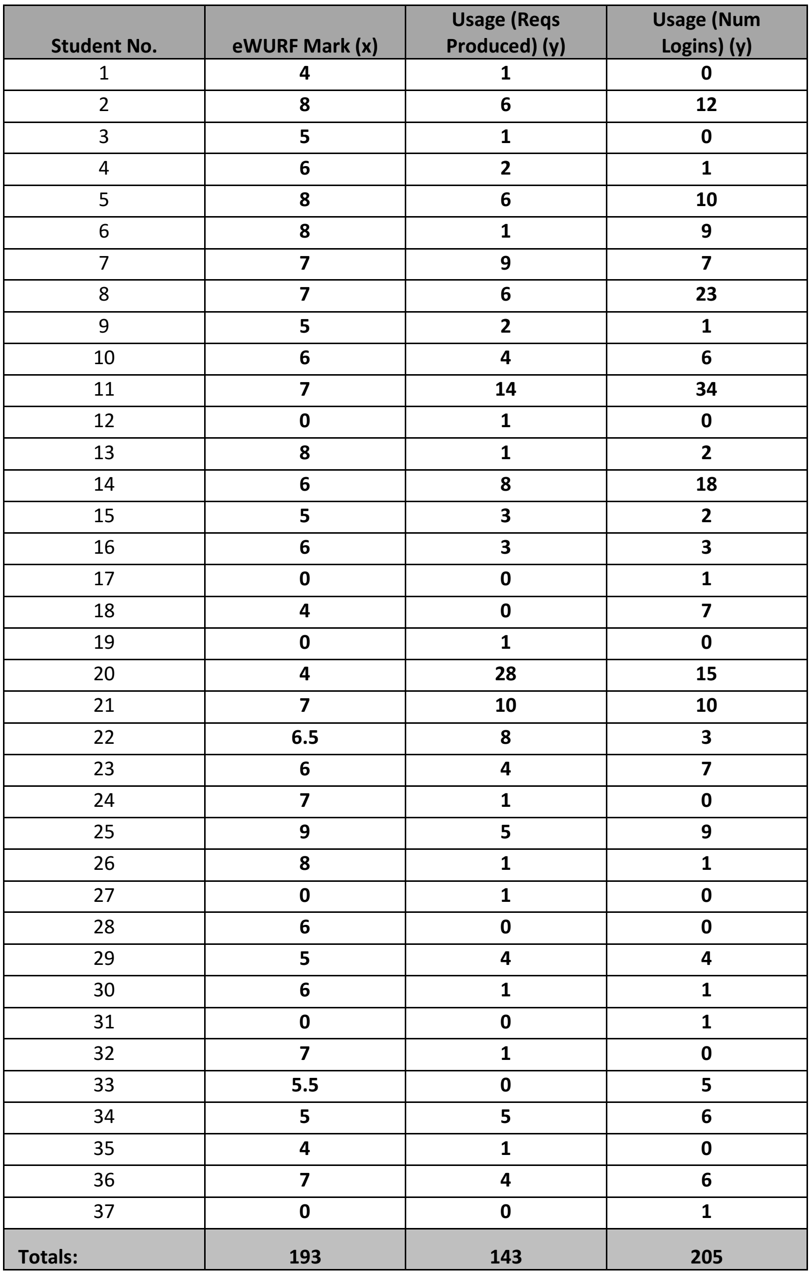
Test 1. Requirements documents produced vs Marks	
Spearman's Rank Correlation Coefficient r =	**0.226230711**
Statistical Significance of Correlation Two-Tailed T Test =	**0.325**
∴ null hypothesis (HA0) must be accepted	
Test 2. Frequency of Logins vs Marks	
Spearman's Rank Correlation Coefficient r =	**0.372954685**
Statistical Significance of Correlation Two-Tailed T Test =	**0.325**
∴ hypothesis (HA1) can be accepted	

Formula Used For Correlation Is:
(Spearman's Rank Correlation Coefficient)

n = number of pairs of scores
$\sum xy$ = sum of the products of paired scores
$\sum x$ = sum of x scores
$\sum y$ = sum of y scores
$\sum x_2$ = sum of squared x scores
$\sum y_2$ = sum of squared y scores

Analysis was undertaken on the following data sets:

Table 1. x and y scores



Student No.	eWURF Mark (x)	Usage (Reqs Produced) (y)	Usage (Num Logins) (y)
1	4	1	0
2	8	6	12
3	5	1	0
4	6	2	1
5	8	6	10
6	8	1	9
7	7	9	7
8	7	6	23
9	5	2	1
10	6	4	6
11	7	14	34
12	0	1	0
13	8	1	2
14	6	8	18
15	5	3	2
16	6	3	3
17	0	0	1
18	4	0	7
19	0	1	0
20	4	28	15
21	7	10	10
22	6.5	8	3
23	6	4	7
24	7	1	0
25	9	5	9
26	8	1	1
27	0	1	0
28	6	0	0
29	5	4	4
30	6	1	1
31	0	0	1
32	7	1	0
33	5.5	0	5
34	5	5	6
35	4	1	0
36	7	4	6
37	0	0	1
Totals:	193	143	205

Table 2. sum of squared x and y scores (Test 1 and Test 2)

	(Test 1)				(Test 2)	
x*y	x*x	y*y		x*y	x*x	y*y
4	16	1		0	16	0
48	64	36		96	64	144
5	25	1		0	25	0
12	36	4		6	36	1
48	64	36		80	64	100
8	64	1		72	64	81
63	49	81		49	49	49
42	49	36		161	49	529
10	25	4		5	25	1
24	36	16		36	36	36
98	49	196		238	49	1156
0	0	1		0	0	0
8	64	1		16	64	4
48	36	64		108	36	324
15	25	9		10	25	4
18	36	9		18	36	9
0	0	0		0	0	1
0	16	0		28	16	49
0	0	1		0	0	0
112	16	784		60	16	225
70	49	100		70	49	100
52	42.25	64		19.5	42.25	9
24	36	16		42	36	49
7	49	1		0	49	0
45	81	25		81	81	81
8	64	1		8	64	1
0	0	1		0	0	0
0	36	0		0	36	0
20	25	16		20	25	16
6	36	1		6	36	1
0	0	0		0	0	1
7	49	1		0	49	0
0	30.25	0		27.5	30.25	25
25	25	25		30	25	36
4	16	1		0	16	0
28	49	16		42	49	36
0	0	0		0	0	1
859	1257.5	1549		1329	1257.5	3069

Appendix C5 – Student Generated SRS Document

Online Portfolio

Document Version: **6**

Date Of Issue: **Tuesday 26th April 2010**

Document Author: **xxxxx xxxxxx || xxxxxx@tees.ac.uk**

Project Description: **An online system that will allow users to register and login to the website. They can then upload their cv and images and view all uploaded content. There will also be an administration section.**

Statement Of Purpose: **I will develop a website using a mixture of HTML, CSS, PHP and mySQL. The user interface of the website will be designed in HTML. Server side scripts will be developed using PHP to produce dynamic objects. mySQL will be used to create database tables, and CSS will be used to define the layout and appearance of the website. SQL queries will also be implemented into the PHP code. The purpose is to develop functionality that will allow a user to register and then log in to the website. The member can then upload images of themselves along with information from their curriculum vitae. The website will include a section to update any information stored about the member and an option to edit or delete their curriculum vitae and image. The website will also include an Administrators section that will allow an administrator to log in to the system and view and edit all information from all user accounts.**

Objective1: **To help job seekers promote themselves to potential employers**

Objective2: **Improve my PHP skills**

Objective3: **Improve Photoshop skills**

Objective4: **Understand SQL queries in further detail**

Objective5: **To allow general browers to view the website but not secure content**

Objective6: **To allow people to register and upload information from their CV and include images**

Tasks And Associated Actors

Task Name: **Create a registration form that will require email activation** Actor: **Developer,**
Task Name: **Create a log in section** Actor: **Developer,**
Task Name: **Create a log out section** Actor: **Developer,**
Task Name: **Create a section that will allow a user to upload an image of themselves** Actor: **Developer,**
Task Name: **Create a section that will allow a user to update any details stored about themselves** Actor: **Developer,**
Task Name: **Create a section that will allow a user to upload their CV** Actor: **Developer,**
Task Name: **Create an administration section** Actor: **Developer,**
Task Name: **Create a section that allows a user to edit and delete all information and images for their own user account** Actor: **Developer,**

Functional Req's And Associated Actors
Functional Req's: **Shall allow people to register. Shall not allow people to access secure area without logging in. Must allow people access with the correct credentials.** Actors: **General browsers of the website, Person seeking**

employment (Upload their CV),

Functional Req's: **Must allow users to upload information, cv and images. Must not allow unauthorised users access to information. Must not allow unauthorised users to edit information.** Actors: **General browsers of the website, Person seeking employment (Upload their CV), Administrator,**

Functional Req's: **Must allow registered users to log in to the system. Must not allow unregistered users to log in to the system. Must only accept the correct username and password combination.** Actors: **General browsers of the website, Administrator, Person seeking employment (Upload their CV),**

Web Requirements

Key: FR-x = Functional Requirement || NFR-x = Non-Functional Requirement

FR-1

Functional Requirement 1

functional requirement

Description: **Shall allow people to register. Shall not allow people to access secure area without logging in. Must allow people access with the correct credentials.**

Associated Actors: General browsers of the website, Person seeking employment (Upload their CV), , ,

These Actors will need to be taken into consideration when designing/developing this requirement!

Comments:**Registration**

FR-2

Functional Requirement 2

functional requirement

Description: **Must allow users to upload information, cv and images. Must not allow unauthorised users access to information. Must not allow unauthorised users to edit information.**

Associated Actors: General browsers of the website, Person seeking employment (Upload their CV), Administrator, ,

These Actors will need to be taken into consideration when designing/developing this requirement!

Comments:**Portfolio**

FR-3

Functional Requirement 3

functional requirement

Description: **Must allow registered users to log in to the system. Must not allow unregistered users to log in to the system. Must only accept the correct username and password combination.**

Associated Actors: General browsers of the website, Administrator, Person seeking employment (Upload their CV), ,

These Actors will need to be taken into consideration when designing/developing this requirement!

Comments:**Log In**

NFR-1

User Interface

Non-Functional Requirement

Description: **The User Interface will be designed at a screen resolution of 800x600. The layout will be Fixed and the User Interface will scroll if there is any excess content. The main text size will be Scaleable. The main**

navigation will be of Top type and will use Bread Crumb navigational aids to help the user see where they are within the application.

Associated Actors: **Not Applicable To Non-functional Requirements**

Additional Req's: **null**

NFR-2

Marketing

Non-Functional Requirement

Description: **The website will use Domain Name for Search Engine Optimisation. In order to track the metrics of any marking campaigns, the site will use in order to evaluate it effectiveness in terms of traffic. In terms of website visibility and rankings on search engines, we will use Manual Submission in order to achieve this. In order to evaluate goal conversion we will use the following; Review in house.**

Associated Actors: **Not Applicable To Non-functional Requirements**

Additional Req's: **Website hosted on Universities internal web server so search engine optimisation is not applicable.**

NFR-3

Accessibility

Non-Functional Requirement

Description: **The Accessibility of the site will be enhanced by meeting Priority 3 WAI Guidelines. The following assistive technology will be used: ALT Tags Applied To Images, .**

Associated Actors: **Not Applicable To Non-functional Requirements**

Additional Req's: **null**

NFR-4

Usability

Non-Functional Requirement

Description: **The type of website that is being designed requires the Efficiency of tasks to be: Very Important, the Learnability to be: Medium Importance, and Memorability of tasks to be: Very Important. The site will use Basic searching to help users find information on the site. It is required that that loads in less than 10 seconds.**

Associated Actors: **Not Applicable To Non-functional Requirements**

Additional Req's:

NFR-5

Technical

Non-Functional Requirement

Description: **The development team must use PHP and MySQL as the serverside technology. The server will therefore be: Linux.**

Associated Actors: **Not Applicable To Non-functional Requirements**

Additional Req's: **null**