

A requirements framework for novice web developers

Sheridan Jeary, Keith Phalp, Lai Xu, Paul de Vrieze
Software Systems Research Centre
Bournemouth University,
Bournemouth, UK
sjeary/kphalp/lxu/pdevrieze@bournemouth.ac.uk

Abstract—This paper introduces a requirements framework intended to guide novice web developers. The work is based on two earlier studies which found that the requirements phase is not well served in web development methods and that there is no agreed set of requirements from practitioners as to what they would like to see in a web development method. The requirements framework outlined here is developed by novice practitioners and later evaluated by them as flexible, simple and easy to use.

Keywords—web requirements; web development methods; novice developers;

I. INTRODUCTION

Much of the literature on web application development argues that it is different from traditional software application development citing numerous reasons [1; 2; 3]. We have previously divided these differences into three areas [4]: the user domain, the developer and development team domain and finally, the environment. The user domain is of particular interest to the requirements engineer because the web at the interface is of primary importance to the user and is an area where there are a number of differences. The user will enter any web site at any point and not, as traditionally expected, at the start of an application. They will also leave at any point [5]. In addition, the technology is more visible to the user and they will blame any delay in the internet technology transmission on the site they are visiting [6; 7]. When creating a new web application for a client the web developer rarely has any contact with any expected user of the site, except through email and survey, and thus the user is often misrepresented in a development [8]. In addition, if they do not like the results of the development they are only a mouse click away from going to the competition [9]. This creates a number of problems for the web developer in obtaining and validating requirements.

Many web developers are immature in their experience of the web, and have little knowledge of computer science or development methods, thus the use of traditional methods could be an issue [1; 7; 10]. From our experience, this lack of expertise is particularly prevalent in the development of web applications within Small and Medium Enterprises (SME's).

Section 2 explores the background to requirements prior to detailing a study in Section 3 which shows the creation of a requirements framework for web development and

discusses the various components prior to its evaluation in Section 4. Section 5 sets out conclusions and further work.

II. REQUIREMENTS BACKGROUND FROM THE LITERATURE

The requirements phase of traditional software development has been researched extensively over the last thirty years [11]. Nuseibeh and Easterbrook [12] define requirements as “the process of discovering that purpose (for which the software was intended) by identifying stakeholders and their needs and documenting them in a form that is amenable to analysis, communication, and subsequent implementation”. There has been a great deal of work in software engineering to ensure the adoption of systematic requirements gathering and an engineering based approach to analysis and modelling [12].

Web development has a number of issues that make the requirements process more difficult to apply than in typical application development. Lowe and Eklund [13] discuss the issues inherent in obtaining requirements late in the development process, when clients are unable to articulate them until after the initial design has been created. The issues that arise for the requirements stage include shorter development cycle times and uncertainty and volatility in user/client requirements coupled with the user/client only knowing their requirements when presented with a solution [13]. This uncertain environment causes developers to commence implementation with very little knowledge of the user /client requirements and these requirements may only become evident as the system evolves over its life [6]. Consequently, the requirements phase is part of an iterative cycle of requirements elicitation, design and prototyping as the clients needs are better understood and the system evolves [13].

It is also evident that many web development methods are weak in the area of requirements [14]; assuming that these will already be in the possession of the developer before development begins and often not specifying how or what should be obtained. In addition, there is a need to create a new taxonomy of requirements that can be linked to business issues[15].

This method weakness may be because the method authors make assumptions about developer understanding and they expect developers to be able to liaise with clients,

stakeholders and users to obtain a complete set of requirements which can subsequently be documented, prioritised, actioned and finally used to test and evaluate the finished system. However, novice developers consider the requirements section a vital part of their requirements for a method and expected to be advised on the elements that needed to be considered and how to handle such elements [16]. This observation has been corroborated by our own anecdotal evidence from undergraduate project supervision, which suggests that novice developers are often aware that they need to find out what the user needs and/or wants of any system, but are unable to work out the scope of the requirements process for a web development. Interviews with developers in industry confirm this view [16]. Those industry developers interviewed even suggested that the user 'does not know what they want' and therefore need to be guided into making appropriate decisions. That is, they need to be guided as to what are the important issues that need to be explored and what can usefully be omitted.

Requirements have traditionally been divided into functional or non-functional requirements [17]. The functional requirements are requirements that can be met by appropriate behaviour or functionality on the part of the solution system [18]. However, Davis [19] terms the non-functional requirements as non-behavioural requirements and adds that they will include some requirements that are not easily categorised; such as response time, capacity, degradation, maintainability and adaptability, reliability, tailorability and portability [19]. Bray [18] is somewhat more pragmatic and terms them as performance requirements using a slimmed down list which includes some of the Davis list and includes usability [18]. Web development however, involves all the categories from Davis [19] and usability as suggested by Bray [18]. Bray [18 p.17] identifies a third category of requirements which he terms design constraints, 'the true non-functional requirements which identify how the system is built but not what it does'. If, as Lowe [6] believes, web development means that traditional non-functional requirements become paramount, then arguably the requirements process should have greater significance in the development process, and the traditional requirements approaches need to be re-considered to take account of the increasing importance of non-functional requirements.

III. THE CREATION OF A REQUIREMENTS FRAMEWORK

The creation of a requirements framework for web development leads on from our earlier work which surveyed web development methods and techniques and found that, despite assertions to the contrary, most methods did not cover the entire lifecycle from requirements to maintenance in enough depth to allow the method to be followed in all phases [20]. This was particularly true of the requirements phase. We also found that there had been little research into understanding what practitioners were looking for, or required, from a web development method. Furthermore, that

there were no agreed set of requirements for a web development method from either theoretical or industrial perspectives [20]. We next conducted a two year study of 23 small scale web developments attempting to use a range of web development methods taken from academic literature. All but one developer abandoned the methods stating they were difficult to learn and apply [21]. Feedback from the study showed that the formality of the method used and the ambiguity of the terminology caused problems in the application of the method. If this was true of participants with specific training and knowledge in computing, then it suggested that typical practitioners, who often have no formal computing background, would also view development methods as difficult. In addition, many of the methods were incomplete. For example, many mention feasibility and requirements but assume the method user will have the knowledge to undertake those aspects. In addition, they assume that the emphasis taken by the method user will be the same as that envisaged by the method author. Again, the variety of backgrounds of web developers means that such assumptions may be inappropriate [21].

For the purposes of the study now described, that is to create a requirements framework for novice developers, the subjects were final year undergraduate students from a British University on BSc (Hons) degree courses in Business Information Technology and Computing and Internet Technology. They had all been in industry for an industrial placement for the third year of their studies and, thus, could be considered as novice developers. The requirements framework was created over five years using an interpretive, inductive approach using an adaptation of template analysis [22]. An initial template was designed as a series of questions that could be discussed. The template was updated after each session and seven updates took place before the template was complete. For a full discussion see [16]. Surprisingly, although expectations might be that the students would have unclear ideas as novice developers, they had strong opinions about what should and should not be included in the prospective requirements part of the method. A number of them felt that some guidance, particularly for people doing web development early in their careers was essential.

The findings became a series of headings of components that the students considered should be in the framework. As previously reported, web development methods are difficult to use for novice developers [4] however, a number of the students identified with two elements of WSDM [23] that the students believed could be used to good effect in any method. The framework is shown at Figure 1.

<u>Requirements Framework</u>	
1.	What is the site for?
a.	To provide a community?
b.	To increase company visibility?
c.	To provide a new business model?
d.	To provide information?
e.	To make money?

2. Who are the target audience?
 - a. How will you find this out? What problems are there to doing this?
 - b. What are the expected types? What are the requirements of each type?
 - c. Who are the stakeholder audience?
 3. What is the business model?
 4. High Level Content Analysis
 - a. Who is providing the content?
 - b. Marketing analysis and planning
 - i. What are the expected visitor numbers and types
 - ii. Product releases etc
 - c. Who owns it?
 - d. What about copyright?
 - e. How often does it need updating?
 - f. Who will update it?
 - g. Consider privacy, accuracy, property, accessibility
 5. Security implications?
 - a. For different revenue models
 - b. For different audience types
 - c. Sensitive information
 6. Timescale
 - a. What is required by when?
 7. Technical issues
 - a. Hosting requirements
 - b. Development environment
 - c. Language
 - d. Hardware
- Specification
1. Identify audience types using:
 - a. Statement of purpose
 - b. Market research
 - c. High level target audience identification
 2. Write scenarios for each audience type
 - a. From scenarios identify information requirements
 - b. From scenarios identify functions they will need to complete
 3. Identify audience structure linking similar information and functions

Figure 1. Requirements Framework

It is accepted that there are categories that could be considered to be missing from the list such as personalization and requirements resulting from marketing strategy; however, this framework is presented as constructed by the study participants. Further work will consider additional items. The following is a discussion on the selection of the individual components and the reasoning behind their inclusion.

A. What is the Purpose of the Site?

This element is included, at the students direction; it comes from both WSDM [23] and Turbo prototyping [24]. De Troyer [25] believes that not having a mission statement or statement of purpose will mean that goals may never be reached and there is no high level basis for any evaluation into the effectiveness of the site.

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

B. Who is the target audience?

This is another element taken from WSDM that the students considered important to include. The identification of the audience is a complex part of web development. There may be market research data available, in which case this may be used, but without this the site may not cater to the tastes of its audience. The target audience is 'that specific audience who is interested in and will benefit from the site' [25]. Audience identification is a complex process which many developers do not consider in enough depth; for example with a commercial site as well as customers, for example, there are likely to be potential customers, suppliers, distributors, potential investors etc. One of the major issues is that users are divorced from the development and may not be able to engage with the process [8]. Users of web sites are often asked to complete market surveys about their usage of the site and it is questionable whether market research surveys offer additional value. User forums are often a source of useful information, customer service blogs and wikis can be helpful. Interestingly, De Troyer [25] suggests looking at the business activities and decomposing them into target audience classes. (The link between the business processes and the web site is not followed up in her work, and is considered outside the scope of the work here). However, this link is something that should be explored in further work, and some authors have suggested using business process models as part of web development methods (e.g., Bleistein and others [26]). The two steps consisting of statement of purpose and identification of the target audience are important at a high level at the beginning of any project, and will be re-visited and undoubtedly identified in greater detail during requirements gathering and specification. Web sites dealing with web design also give guidance such as that described by Leigh [27] which suggests thinking about the following categories:

- Age
- Sex
- Religious aspects
- Ethnic backgrounds

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

C. *What is the business model?*

The business model is introduced at this stage. Osterwalder and Pigneur [28] summarise five reasons why a business model is essential to a business; three of which can be seen as important in requirements elicitation. These are first, identification of elements in a domain and the linkages between them; second, a formal e-business model allows useful discussion between managers and stakeholders and third, the mapping of the model provides a foundation for discussion and facilitates change. This model is likely to become part of the discussion among the developer, the client and other stakeholders relating to the statement of purpose and target audience, and is added to ensure that the topic is covered during elicitation. The site will typically need to be paid for either by the client, sponsor or stakeholder and if the site needs to be self-sustaining the business model becomes an important issue. Therefore, both business and revenue models become important in the setting up and operation of the site.

D. *Development Constraints*

Originally conceived from a development point of view, constraints could be considered in terms of budget, development environment or programmer availability. However, Chevalier and Bonnardel [29] believe that constraints can be split into either client or user constraints. Client constraints are those that result from interactions with the client or are inferred from interactions with previous clients, and user constraints which are developed as a result of a web developers own usage of web sites. They may address general aesthetics of web sites or specific aspects relating to usability such as the navigation.

E. *High Level Content Analysis*

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

F. *Security implications*

Different business and revenue models will require a developer to consider the amount of security and privacy that are required. For example business applications are expected to be secure so that personal data belonging to users is safe. Providers of services need to ensure that they are not open to third party breaches such that business critical information can be accessed or modified. Much of the information available to developers is in the form of how to and there is little reference to the developer reflecting on the different business and revenue models to decide what security will be needed such as integrity, non-repudiation, confidentiality, authentication etc. [30].

G. *Timescale*

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

H. *Technical issues*

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

I. *Towards specification*

Having identified the users of the site at a high level, a more detailed analysis is then conducted to identify the different user groups. The form that the students liked for this section was where each different class of audience was drawn life-size on a wall chart, and their habits, typical likes and dislikes were added by the development team so that the way they used a site was shown in a visual manner enabling the developer to have greater understanding of the user. This also ensures that individual developers have input into the user classification and gives a greater degree of inclusivity.

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

IV. EVALUATION

There is a large body of work on method comparison for example,[31; 32; 33; 34] but little work specifically on evaluation. The work of Abrahao and Poels [35] considers method evaluation using an experimental approach, which could have been used. However, the approach does not give

the richness of data that is necessary to further inform the framework. Sol [36] describes five different ways of undertaking method comparisons which are all with the final purpose of selecting a method to use. To this end, they can, therefore, be usefully examined to see if they could be adapted to allow the evaluation of a requirements framework. Jayaratna [37] usefully summarises Sol's five different ways of method evaluation:

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

The first two approaches, the comparison with an ideal method and the generalised measurement tool would appear to be useful techniques. Criticisms of the ideal methodology comparison and the generalised measurement tool are particularly related to the concept that any views are of necessity subjective. To counteract these criticisms the evaluation of the requirements framework could be given to a number of different people and thus a generalised view of the results could be obtained; this would be a variation on the outlined approaches.

The evaluation of the framework was made by a group of developers of a similar experience level to the developers that informed the creation of the framework. Thus, 63 students in the final year of their undergraduate studies on the Software Systems Framework at a British University made the evaluation. They had all undertaken an industrial placement for one year. They were aware that the framework was part of a PhD study and were briefed fully about being critical and their choices. The students were self selecting in that they had opted to complete a final year unit and the evaluation was part of their assessment. The students were asked to create their own list of requirements categories prior to seeing the framework and doing any research themselves. They were then asked to answer a series of questions - what should be added to their categories and why? What should be removed from their categories and why? What should be added to the requirements framework and why? What should be removed from the requirements framework and why? Is there a perfect set of requirements categories? What should the perfect list include?

The framework was generally received well, with the students believing it was straightforward and simple to apply. Of the initial 63, 48 students allowed their work to be used, although 3 students made no useful contribution.

Therefore 45 students gave useful contributions and made 569 comments which were analysed using an inductive process. Of these 569 comments 257 were related directly to the categories in the framework and 188 suggested other categories that could be included. 124 comments were negative in that they suggested a category was missing or wanted to remove a category. One student had 10 negative comments. (Six students made suggestions that were based on incorrect suppositions and were, therefore, just incorrect.) The comments were placed in framework categories and analysed. The numbers of comments are summarised in Table 1.

TABLE I. FRAMEWORK COMMENTS

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

Full details of the feedback and actions can be found in [16] but a selection are included here for illustration.

A. Statement of purpose?

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

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

B. Who is the target audience?

The most common comment here was the suggestion to split the target user and stakeholders (4 students) because their requirements would be different and thus could be separated. Only one student would not have used this as a category of the framework with eight agreeing as to the usefulness of the approach. Two students recognized the use of the target audience again later in the framework and the greater detail that would be added later in the iterations.

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

C. Requirements gathering

This was the section that resulted in the most comments. 33 students made specific comments in this section and 121 comments were recorded of which 54 were general in nature. Two students suggested the use of time-boxing and MOSCOW rules [38] for prioritizing requirements and three suggested the framework should outline some techniques for telling novices where to find information that provides guidance as to how to obtain requirements because traditional elicitation techniques were unlikely to be useful. Student 43 suggested that a list of possible techniques should be given so that novices could select a technique. The list they provide shows, for example, 8 different elicitation techniques.

The aim of the framework was not to force users to use specific techniques, but to allow the developer a free choice and provide what Cockburn terms a ‘lite’ method [39]. (However, the idea of providing a ‘Body of Knowledge’ of possibilities is attractive. Hence, although this suggestion has not been incorporated within the current framework a further work will explore technique choices for novice developers).

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

There were a further 67 comments which were categorised into each area of the requirements gathering phase. The category with the most specific comments was the technical section. This is possibly as a result of the technical nature of the degrees that many of these students are following; however some points are felt to be relevant. Of the 28 comments in the technical section there was debate by some students as to whether performance requirements should be included, student 54 believed that *“when designing the site there will be many factors that have to be taken into account, which may not be specified as functional requirements, but rather as performance requirements. These days, however, performance has become a large part of the functionality, so the two overlap”*. This thought is articulated well and reflects the nature of the framework, thus technical requirements will remain. Student 5 agrees and states that *“by defining exactly what technologies that are going to be used the web developers can educate the clients on the technologies and their capabilities – which may have a major impact on other parts of any development method, such as design and navigation”*. There is some evidence that client understanding of technological capabilities is low at the outset of projects [13] and by explicitly introducing it at the requirements stage it may help the development process.

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

A second approach to evaluation is to create a dialogue as discussed by Jayaratna [37] which will enable shortcomings and issues with the framework to be identified. Therefore, the second part of the evaluation involved seven final year Software Systems Framework students on BSc Business Information Technology, Computing and Software Engineering degrees completing their 60 credit final year project (which makes up half the final year mark) using the requirements framework. The students were again self selecting in that they were producing a web application and decided to use the framework as their project development method. The projects resulted in seven different web applications of differing success; this would be expected from a cross section of student projects.

Interestingly, nearly all students commented, as benefits of the framework, on its simplicity, flexibility and ease of use; although the same simplicity and flexibility were also picked out as negative points suggesting that the developer needed experience to be able to follow the framework. This was the section of the framework that received the most positive comments and was *“the most straightforward stage to follow comprehensively”*. The simplicity of the language *“that on the whole could be understood by people outside academia and people with little technical development experience”* and meant that the steps *“could be discussed with a client to gather more detailed information regarding what the system needs to incorporate”* were considered important points by the students. Another student used *“the points as the centre for brainstorming sessions helping to note and develop further requirements”*. One student believed it *“simple to apply as a result of the clear communication of its key principles, and the amount of information it puts across. This makes it easy to think of the requirements, making this a good way to document requirements for projects like this one”*. Although another student felt that they had too much information and created *“a disorganised requirements document”* and felt it *“could be a reflection of the method, ...or the developer...but a technique to ..fathom....documentation may have been useful here”*. Inclusion of several steps received positive comments, including statements of purpose which *“gave a high level overview of the aspirations of the website and was good for remaining focussed during the development”* and user identification when comments such as a *“...clearer understanding of what is needed from the website is generated”* were made.

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

V. CONCLUSIONS AND FURTHER WORK

This paper presents a framework for web requirements for novice developers.

Our earlier work found that web development methods presented in academic literature often did not cover the whole lifecycle, from requirements to maintenance, and that the requirements phase was particularly poorly served. In addition, that methods were rarely described in enough detail to allow their application and that often the requirements were assumed to have already been collected. Further, that these same web development methods were too difficult to apply in most cases for novice developers, who were likely to abandon using them. There has also been no set of agreed requirements for what practitioners would expect in a ‘web’ method. These findings were matched in an industrial study which has not been described here.

In contrast to these findings we wished to base our work on practice. We used input from final year undergraduates after their placement year in industry, who represented novice web developers. Discussions were carried out over a five year period involving seven iterations of a template.

The framework is evaluated using a generalized approach by similar novice developers (final year undergraduate students). A further evaluation using a dialogue approach by seven developers creating a variety of applications gave rise to comments such as easy to use, simplicity and flexibility. The framework has been very successful and taken up by a number of project students (outside this study) who were asked to evaluate it and have then gone on to use it successfully in industry.

There is still the possibility of further evolution but the main advantages that we see are:

- It is simple and easy to use
- It is not prescriptive as far as modeling notations are concerned
- It gives a novice developer many of the requirements issues to consider at the beginning of a development
- It opens a dialogue between the client and the developer

In addition, a general finding, or position, would be the need to inform our methods with a greater sense of the needs of practitioners, particularly the novice web developer (which is especially important given the demographic of web teams).

REFERENCES

- [1] S. Murugesan, Y. Deshpande, S. Hansen, and A. Ginige, Web Engineering: A New Discipline for Development of Web-Based Systems. in: S. Murugesan, and Y. Deshpande, (Eds.), Web Engineering 2000, Springer Verlag, Berlin Heidelberg, 2001.
- [2] D. Lowe, and B. Henderson-Sellers, Characterising Web Systems: Merging Information and Functional Architecture. in: V.K. Murthy, and N. Shi, (Eds.), Architectural Issues of Web-Enabled Electronic Business, Idea Group Publishing, Hershey, 2003, pp. 227-239.
- [3] S.P. Overmyer, Whats different about Requirements Engineering for Web Sites? Requirements Engineering 5 (2000) 62-65.
- [4] S. Jeary, K. Phalp, and J. Vincent, An evaluation of the utility of web development methods. Software Quality Journal 17 (2009) 125-150.
- [5] C. Barry, and M. Lang, A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE Multimedia 8 (2001) 52-59.
- [6] D. Lowe, Web system requirements: an overview. Requirements Engineering 8 (2003) 102-113.
- [7] T.A. Powell, D.L. Jones, and D.C. Cutts, Web Site Engineering: Beyond Web Page Design, Prentice Hall, Upper Saddle River, 1999.
- [8] J. Holck, and T. Clemmensen, What makes web-development different?, Dept of Informatics, Copenhagen Business School, Copenhagen, Denmark, 2001.

- [9] R. Glass, Who's right in the Web Development Debate? Cutter IT Journal 14 (2001) 6-10.
- [10] P.R. Vora, Designing for the Web: A Survey, interactions, 1998, pp. 13-30.
- [11] A. van Lamsweerde, Requirements engineering in the year 00: A research perspective, 22nd International Conference on Software Engineering, ICSE'00, IEEE Computer Society Press, Limerick, Ireland, 2000, pp. 5-19.
- [12] B. Nuseibeh, and S. Easterbrook, Requirements Engineering: A Roadmap, International Conference on Software Engineering (ICSE), Limerick, Ireland, 2000.
- [13] D. Lowe, and D. Eklund, Client Needs and the Design Process in Web Projects. Journal of Web Engineering Vol 1 (2002).
- [14] M.J. Escalona, and N. Koch, Requirements Engineering for Web Applications - A Comparative Study. Journal of Web Engineering 2 (2004) 193-212.
- [15] L.S. Al-Salem, and A.A. Samaha, Eliciting Web application requirements - an industrial case study. The Journal of Systems and Software 80 (2007) 294-313.
- [16] S. Jeary, The requirements for, and use of, web development methods, Software Systems Research Centre, PhD Thesis, Bournemouth, 2010.
- [17] L. Maciaszek, Requirements Analysis and System Design, Addison Wesley, Harlow, 2007.
- [18] I. Bray, An Introduction to Requirements Engineering, Pearson Education, 2002.
- [19] A.M. Davis, Just Enough Requirements Management: Where Software Development Meets Marketing, Dorset House, New York, 2005.
- [20] S. Jeary, K. Phalp, and J. Vincent, Web Development Methods and Techniques - A Survey, 15th International Software Quality Management Conference/INSPIRE Conference, Stafford, England, 2007.
- [21] S. Jeary, K. Phalp, and J. Vincent, Are Web Development Methods Too Difficult to Use?, 15th International Software Quality Management Conference/INSPIRE Conference, Stafford, UK, 2007.
- [22] N. King, Using templates in the thematic analysis of text. in: C. Cassell, and G. Symon, (Eds.), Essential Guide to Qualitative Methods in Organizational Research, Sage, London, 2004.
- [23] O.M.F. De Troyer, and C.J. Leune, WSDM: A User Centered Design Method for Web Sites, 7th International World Wide Web Conference, Elsevier: Computer Networks and ISDN Systems, 1998, pp. 85-94.
- [24] G. Ghosh, Turbo-prototyping: Ultra rapid user centred web development. in: S. Brewster, A. Cawsey, and G. Cockton, (Eds.), Human-Computer Interaction INTERACT 99, IFIP, Edinburgh, Scotland, 1999.
- [25] O.M.F. De Troyer, Audience-Driven Web Design. in: M. Rossi, and K. Siau, (Eds.), Information Modelling in the New Millennium, IDEA Group Publishing, 2001.
- [26] S. Bleistein, K. Cox, J. Verner, and K. Phalp, Requirements Engineering for e-Business Advantage. Requirements Engineering Journal 11 (2006) 4-16.
- [27] A. Leigh, Target Audience. Retrieved 1 October 2010 from <http://www.shire.net/learnwebdesign/target.htm>.
- [28] A. Osterwalder, and Y. Pigneur, An e-Business Model Ontology for Modeling e-Business, The 15th Bled Electronic Commerce Conference: e-Reality: Constructing the e-Economy, Bled, Slovenia, 2002.
- [29] A. Chevalier, and N. Bonnardel, Articulation of web site design constraints: Effects of the task and designers' expertise. Computers in Human Behavior 23 (2007) 2455-2472.
- [30] M. Wimmer, A. Kemper, and S. Seltzsam, Security for Web Applications. in: G. Kappel, B. Proll, S. Reich, and W. Retschitzegger, (Eds.), Web Engineering, John Wiley, Chichester, 2006.
- [31] A.T. Wood-Harper, and G. Fitzgerald, A Taxonomy of Current Approaches to Systems Analysis. The Computer Journal 25 (1982) 12-16.
- [32] T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart, Information Systems Design Methodologies: Improving the Practice, Proceedings of the IFIP WG 8.1 Working Conference on Comparative Review of Information Systems Design Methodologies: Improving the Practice, North-Holland, Noordwijkerhout, The Netherlands, 1986.
- [33] B. Fitzgerald, N.L. Russo, and E. Stolterman, Information Systems Development: Methods in Action, McGraw-Hill, London, 2002.
- [34] D. Avison, and G. Fitzgerald, Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, London, 2006.
- [35] S. Abrahao, and G. Poels, A Family of Experiments to Evaluate a Functional Size Measurement Procedure for Web Applications. Journal of Systems and Software 82 (2009) 253-269.
- [36] H.G. Sol, A Feature Analysis of Information Systems Design Methodologies: methodological Considerations. in: W. Olle, H.G. Sol, and C. Tully, (Eds.), Proceedings of the IFIP WG 8.1 Working Conference on Feature Analysis of Information Systems Design Methodologies, North-Holland, Amsterdam, 1983.
- [37] N. Jayaratna, Understanding and Evaluating Methodologies: NIMSAD A systemic framework, McGraw Hill, London, 1994.
- [38] DSDM Consortium, Atern User Manual. Retrieved 30 June 2010 from <http://www.dsdm.org>.
- [39] A. Cockburn, Selecting a Project's Methodology. IEEE Software 17 (2000) 64-71.