158.225
Assignment 1

Analysis, Modelling, and Design Planning

John Lamb # 04314883

Simon Parr # 09233113

Table of Contents

Introduction: 3

General Requirements: 3

Brief: 3

Assumptions: 3

Considerations: 4

The Application must: 4

The Application should: 4

System Request Document: 5

Recommended System Development Methodology: 6

Discussion: 6

Recommendation: 7

Acquisition Considerations: 9

COTS Vs Bespoke: 9

Recommendations: 10

Description of Required Team: 11

The Team in General 11

Key Members of the Team Include 11

In the Context of our Project 12

Technical Feasibility Analysis: 13

Requirements Gathering Plan: 14

Stakeholder Engagement Plan: 16

Documents and Diagrams Suite 18

Bibliography: 19

Introduction:

Throughout the course of this assignment we will be examining the planning and high-level conceptualisation of an assigned, theoretical software development project. We will review the brief, and cover a range of assumptions that the brief implies. Using the brief, and the assumptions made, we will draft a Systems Request Document and begin to think about Acquisition. We will engage in a discussion about which Development methodologies and the type of team that needs to be involved. Lastly, we will incorporate requirements and stakeholder engagement plans.

General Requirements:

Brief:

We have been engaged to create an application to connect environmental scientists with each other, and with the public. It is intended that the application allow scientists and members of the public to discuss and debate topics of current importance to society, and to allow scientists to ensure that their research is more closely connected to the needs and priorities of members of the public.

The project is primarily being funded by the World Bank.

Assumptions:

While the brief is scant, it does make several things very clear. Firstly, there are two distinct groups of users;

- Members of the Environmental Scientific Community: There is no further information provided about this group other than that they are members of a wider, presumably global, scientific community.
- Members of the General Public: Again, as with the scientific community, very little information is provided to help specify who exactly (of the global community) will be using this system.

Secondly, the system is to help them discuss issues that are of current importance to society. It must be open to both user groups creating topics for discussion. This is vital as we are trying to encourage a two way conversation, where both Scientists and members of the general public can raise concerns.

Lastly, there must be some method of recording which topics are discussed most frequently, and most fervently. There must be a way of collecting and presenting the metadata that is accessible to appropriate stakeholders.

Considerations:

The Application must:

Accessibility: The application must be accessible by both members of the public and scientists from around the world. It is, therefore, likely that the application will have an internet frontage.

Access and Accounts: There will be an array of various users. The user groups that should be considered are; guests, registered users, scientists, administrators/moderators. There should be a level of segmentation also, so that private conversations, or discussions can be sectioned off into layered access, for example; scientists only, or scientists and administrators.

Create Topics or Threads: All user accounts should be able to create discussions ('threads') freely, with the exception of guests.

Search Engine: It should be possible for all user accounts to search through topics to find those that are of interest to them. This should probably be conducted through the use of keywords. A browse function should be available also, so that users who are not necessarily sure about what topics they possess a strong opinion on can simply look through a list.

Collect Metadata/Provide Analytics: In order to be able to actually provide some overview of the debate as it happens. This facility would perhaps record thread names, key word tags, and provide some metrics around views or posts. The metrics would have to be tied to a method to gauge, or provide insight in to, public interest.

Usage suggestions, and Guide Documentation: There should be a usage manual available which explains the applications intended purpose. This manual should be segmented into relevant information for each level of user. It should explain how to use the application, some best practices, and what to expect from the applications function. It may also go as far as to explain some basics about interaction between users.

The Application should:

Away from Application Contact: While being able to record threads and post in them should be a primary concern for members of the public, the ability to receive updates on threads of interest from either participation or in some sort of aggregated overview newsletter would be desirable.

Subscriptions: Another feature that would be handy would be subscriptions. Subscriptions could take many forms however the most likely forms are; subscription to a particular thread, or to a range of topics predefined by the user. However, it may also be possible to introduce subscription to a scientist or group of scientist, for example UCLA Berkley – Department of Environmental Science, Policy, and Management.

Terms and Conditions: There should be a method where-by users are required to accept usage agreement before creating a thread, or posting to one. It would contain the rules around usage of the application, as well as consequences for failure to abide by them.

System Request Document:

Date: 6th September 2015

Project Sponsor:

Global Consortium of Environmental scientists

Business Need:

Environmental issues in today's climate are of growing concern to every sector of society. Awareness of issues are more important to the general members of the public than ever before, therefore it is of growing importance that members of the public have greater accessibility to environmental scientists and their research. And vice versa that environmental scientists have a channel of communication with the public to ensure their research is connected with the needs and priorities. Your team will create an application that satisfies these requirements.

Business Requirements/Functionality:

Users will need access to an open forum with web functionality that will allow environmental scientists to make postings about research progress and provide easier access for members of the public to this information as well as provide an opportunity to debate topics with other users.

Business Value/Expected Value:

Tangible: Increase in advertising revenue.

Intangible: Improved public confidence and improved communication between public and environmental scientists, improved communication between environmental scientists across a number of regions, possibility of additional funding for research projects,

Special Issues or Constraints:

Level of privilege for each member must be decided upon including an administrator and moderators for the forum.

Recommended System Development Methodology:

Discussion:

There are a number of different types of development methodologies. These methodologies include structured development, rapid application development, and agile development.

Structured development relies on 'a formal step-by-step approach to the software development lifecycle' (Dennis et al., 2015, p. 7). This moves in a logical progression from one phase to the next. Members of this family that are commonly encountered include Waterfall Development, and Parallel Development methodologies.

Rapid Application Development (RAD) is a methodology for managing the software development lifecycle. It 'attempts to address the problem of long delays between the analysis phase and the delivery of the system' (Dennis et al., 2015, p. 8). This family of methodologies attempts to get the software in to the hands of users as quickly as possible so that, in turn, the users can be in a position to better offer feedback for the next iteration of development. Typically RAD relies on an array of developmental tools. A major drawback is managing user's expectations as the development process produces results so quickly (Dennis et al., 2015, p. 9). Members of this family that are commonly encountered are; Phased Development, Prototyping, Throw-away Prototyping.

The last family of development methodologies that I will include is Agile Development. Agile development places emphasis on rapid, iterative, development cycles. These cycles involve delivery of software and receiving feedback in a constant manner. This feedback is typically given in a face-to-face meeting (Dennis et al., 2015, p. 12). Agile methodologies are typically used in conjunction with object orientated principles. Agile methodologies do have some criticisms; they rely on co-location of developers and users, they required a large amount of management (which inherently contradicts Agile principles), they produce little or no documentation, and lastly there is some doubt as to whether or not they can produce large pieces of mission critical software. Members of this family that are commonly encountered are; Extreme Programming and Scrum.

Having considered the available brief for the system that we are required to build, I feel that an Agile method of development is inappropriate for this particular piece of software. I feel this because we are unsure as the geographical location of our clients. This precludes the ability to have regular, face to face meetings. While it may be possible to still organise a remote meeting of some sort, it would decrease the effectiveness of the Agile methodology.

Similarly, I feel that a Structured Development methodology would also be inappropriate. It is simply not feasible to develop a system which is centred on users' interactions in a linear manner without the chance for iterative reviews.

Recommendation:

I feel that the most appropriate development methodology for this system is an Evolutionary Prototyping approach. Prototyping is part of the Rapid Application Development Family, and relies heavily on short iterative cycles of creation of functional software and user feedback.

Because the brief is rather scant, there is no relevant technical or user documentation, Prototyping allows us to quickly establish and refine user requirements by presenting workable pieces of software, and receive feedback. By doing so it is possible to mitigate, or reduce, risk. Because the iterative cycles are relatively short, and culminate in a workable feature being reviewed, there is less chance of producing features other than what the stakeholders agree is ideal.

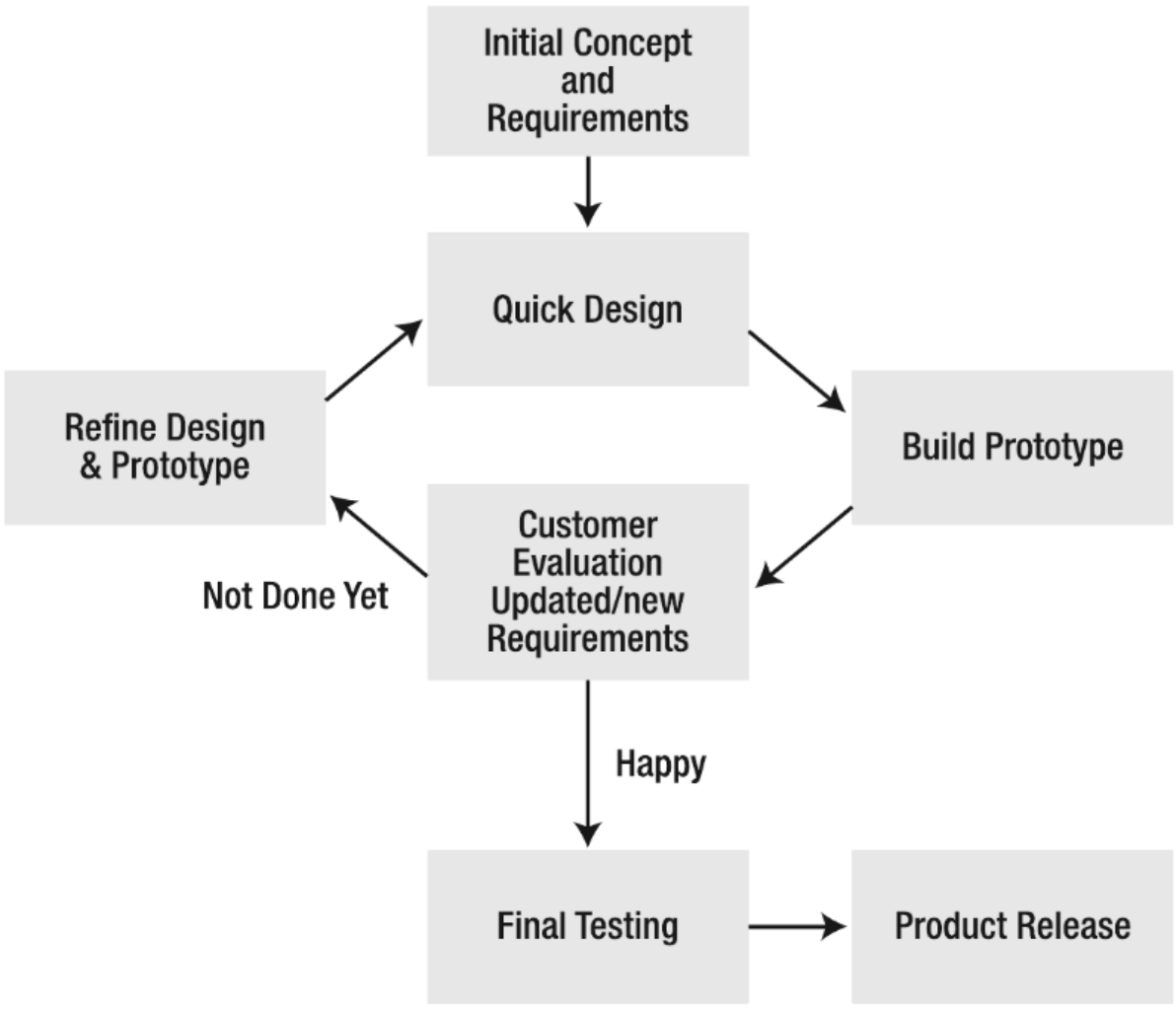


Figure 1 The Prototyping Life-cycle (Nurch, 2006)

I feel that also, given that there is a strong possibility that a large proportion of the features which may be required in the software can be repurchased, this method of development would complement modifying off-the-shelf software. Required features could be addressed individually or

in family groupings. Each additional feature (or family of) added to the core package could be introduced as a prototype, allowing a selected group of users to provide feedback on it.

This process could be repeated over and over until the application is finished. By doing this, and utilising an object oriented approach, each prototype would be clean and contain a single functionality allowing developers to focus down on each problem, and correct it according to the feedback.

Lastly, the users and stakeholders become much more engaged with the development process. By seeing how the software will function interest in the project is maintained, and feedback is encouraged.

There are some disadvantages to this methodology. As it is exploratory in nature, it will require strong management to prevent scope creep. As the brief is open, it is unknown exactly what features may be required. In turn, the number of iterative cycles is also unknown. This means that scheduling and planning becomes difficult. Likewise, budgets are vague initially.

Because each prototype evolves from the previous; there is a strong chance that the application may devolve in to a frankensteinian collection of quick-fixes hacked together in order to rapidly address stakeholder and user feedback. Code maintenance in the form of constant refactoring and version management become crucial as iterations occur.

Lastly, RAD is typically used in a time sensitive situation. The brief specifies no time limit, however it is reasonable to assume that the project must have an end date, and that development should occur in a timely fashion.

Acquisition Considerations:

While examining the brief and drafting my series of considerations I realised that the type of software that we have been asked to produce is almost identical, in terms of functional requirements, to a piece of forum software.

As there are already a large range of forum software packages available it would be worth the projects time to conduct a study, or review, of packages to see whether any of them would fit their requirements.

As discussed previously, I feel that the most appropriate software development methodology in this instance in a member of the Rapid Application Development family; Evolutionary Prototyping. This methodology could easily be adjusted to incorporate either commercial off-the-shelf (COTS) or bespoke software. It is also flexible enough to manage a mixture of the two.

COTS Vs Bespoke:

A lot can be written on the topic of Commercial off-the-Shelf Software Vs a Custom or Bespoke option. This is a debate that is both on going, and large. I will not spend an inordinate amount of time discussing it here, however I feel that a brief overview is appropriate.

Commercial off the shelf software has an array of advantages and disadvantages:

Summarised from Build vs. Buy: How to Know When You.... and Pros and Cons of Choosing Off-the-Shelf Software Products

COMMERCIAL OFF THE SHELF (COTS)

ADVANTAGES	Disadvantages
Cost effective - the majority of the development has already been done.	*Little competitive advantage* -Rival businesses can easily purchase the same piece of software.
Feature rich - typically contains a wide variety of functions	*Mismatch between requirements and the functionality* - Of the software. If your requirements fall outside the parameters of the makers intended usage, you may not be able to accurately address your business needs and compromises will have to be made.
Time tested - many pieces of cots have been in use for some time, and accordingly bugs have been fixed.	*There is a dependence on an external* supplier - for many aspects of the project; implementation, testing, and enhancements.
Well supported - cots packages tend to be well supported by the producers.	
Documentation - cots packages also tend to have suitable amounts of support documentation.	

Custom Software has advantages also:

- A custom or bespoke piece of software will always meet the exact requirements stipulated by stake holders.
- Integration is smoother with current practices as they will have been reviewed as part of the creation process.
- Custom software is appropriately scaled to the requirements of the scenario it is attempting to match.
- Support can be managed by the business themselves. This reduces the reliance on external organisations, and controls the type of support provided.

Recommendations:

Having listed the advantages of each method of acquisition, it is time to consider a recommendation.

Firstly, I feel that it would be both prudent and advisable to research, after following a requirement gathering plan, Off-The-Shelf alternatives. It may very well be the case that an already existing piece of software is able to provide a large amount of the required functionality.

Secondly, when researching options the ease with which the software can be modified should be taken in to account also. Unless the software provides exact functionality there may be a requirement to modify it.

Lastly, if an existing piece of software is selected to either fulfil the requirements or as a basis for further development, it should be presented as the first prototype. This is in keeping with the ideal of prototyping, and will increase the speed with which the developers can refine the projects goals.

Description of Required Team:

The Team in General

Evolutionary Prototyping is a difficult process to manage. Because it creates working software regularly and demonstrates it to stakeholders, it can create unrealistic expectations and schedules. It is recommended that team sizes be kept small, with members who are able to perform more than a single role. It is important that the team is well managed. It is expected that the team be able to communicate and cooperate effectively. These teams, in the Rapid Application Development context, are often referred to as Skilled Workers with Advanced Tools or SWAT teams (Nurch, 2001).

Key Members of the Team Include

What follows is a series of descriptions of appropriate team roles provided by Nurch R (2001) in his work 'Project Management: Best practices for IT Professionals". A summary of the roles the author recommends for this type of methodology is as follows.

Management Sponsor (or Sponsor): The project sponsor is effectively a member of business management, or someone who is closely linked to the end-user community. They will act as an interface between the project team and stakeholders.

A RAD Facilitator: This is the person that is responsible for planning, executing and managing the project. It is critical that the right facilitator is chosen. They should be a respected and skilful leader with a good reputation. They must have a thorough understanding of RAD principals. Other names used for this role outside of the RAD context are; project manager, DP manager, or End User Manager.

Scribe: The scribe has a particularly important role in the RAD team. They are responsible for documenting the RAD sessions. It is important the decisions, ideas, and contributions be recorded. The reasons why are just as important as the events. It is not necessary that the scribe be an IT person, however it helps. Lastly, the scribe is responsible for distribution of documentation and records at the end of a RAD session.

Information Specialist: Are systems analysts who are used to interpret the end users' needs and transform them in to a design or prototype. They can advise end users on new technology that may help the implementation of the new application. They should be good listeners and demonstrate empathy.

Specialists: Specialists have a specialist area of knowledge that is necessary to help complete the project. An example of this would be; a Database Designer, a Network specialist, a System programmer, or even a Business Process Designer.

In the Context of our Project

It is highly likely that we will require someone who is able to fulfil each of these roles. Whether or not a single person carries out a single role, or team members are chosen who can fulfil multiple roles is open to discussion.

RAD Facilitator is required to manage the project and ensure that it is productive.

As there is a discussion between Off-the-Shelf and Custom software, it is highly likely that we will have to have an initial discussion which includes a Sponsor about how to fulfil initial expectations. These discussions will probably also involve an analyst of some sort in the form an Information Specialist. They will also be required to provide information and advice on Off-the-shelf Vs Custom software, as well as any other relevant information.

It is likely, as we are building or deploying a web based application that specialists in that area will be required. These specialists will probably have to demonstrate a range of skills from;

- Creation and editing of a web page.
- Web server establishment.
- Server-side scripting.
- Database design and implementation.
- Security.

The number of specialists required may be small however, calling upon each member of the team to demonstrate key skills in more than one area.

Technical Feasibility Analysis:

In this section we will examine the technical requirements of the project, and its associated feasibility. Dennis, Wixom and Tegarden describe a technical feasibility study as a document that examines "the extent to which the system can be successfully designed, developed, and installed by the IT group" (Dennis et al., 2015, p. 45). This document is important to the decision making process as it highlights any risk associated with the development of the application.

General Technical Feasibility:

As we think about the requirements of the solution that are known to date we realise that as a web application it will require the implementation of some relatively well known technologies. These technologies will probably include:

- A web server of some sort that provides a frontage to interface with web browsers. For example; Apache.
- Associated trappings of a Web Based application like a domain name, a connection to the internet and hosting, SSL Certification, security measures etc.
- A scripted server of some sort. This could be written in one of many languages, such as PHP. This will act as the actual application and as an intermediate between the web frontage and a database.
- A Database program of some sort to manage the information attached to the application. This Database should employ some form of open-source relational database management system. For example; MySQL.

As the technologies required to produce the application are relatively well known we can be relatively confident that experienced specialists can be found to help. To this extent we are confident that the desired Application is technically possible.

Familiarity:

Development of web based communication tools is a well-worn path. The ability to create discussions of a range of sizes pre-dates popular usage of the Internet and the World Wide Web. It is therefore relatively safe to assume that the development team are familiar with the type of software solutions that are currently available, or that can be built.

Project Size:

At this early stage it is difficult to gauge the size of the project. This is all highly speculative, and is dependant.

Compatibility Problems and Integration:

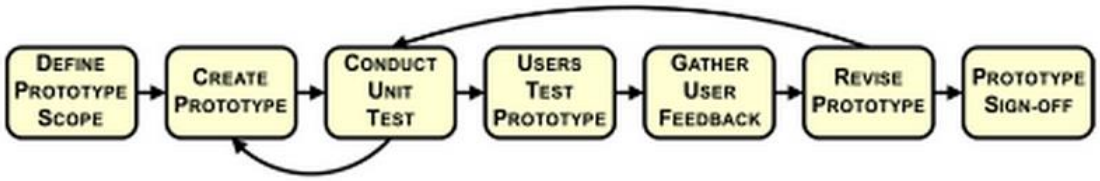
There are no current business processes in place, so integration with current practices and applications is not really a consideration. There are also no other systems that we are aware of with which integration requires special consideration.

Requirements Gathering Plan:

"Apart from a few fortuitous accidents, no product has ever succeeded without prior understanding of its requirements." (Robertson & Robertson, 2006, pp. 1). Therefore it is important that for any project we show due diligence in gathering requirements so we can maximize the success of the project. It is a fact requirements come from people not from systems therefore the basis for our requirements gathering plan must be to talk to people more specifically the people who hold a stake in our project. "All the key stakeholders (the people who can affect the system or who will be affected by the system) must be included in the requirements gathering process" (Dennis et al, 2013, pp.95). When looking at our system request we can easily identify that the major stakeholders of our project are the end users, those who will be using the final product. Therefore we have to understand what uses and needs they have for the system to make sure we can tailor the system directly to their needs.

- Workshop Prototyping

With our chosen methodology the most common way to elicit requirements is to build a prototype of the proposed system a workshop is then organised with members of each stakeholder group. For this particular system a draft version of the final product can be used as the prototype. Additionally due to the proposed software being an internet forum of which a number of comparable internet forum based software already exists therefore one of these similar pieces of software could be used in the workshop. Users then try out this simple prototype and through this interaction it can elicit users into giving further requirements information or possibly changing their mind about existing requirements. In addition to prototyping it is possible to conduct a brainstorm during the workshop to gather further ideas from attendees. After the workshop has concluded an additional survey can be sent out to attendees, this can be beneficial to gain further information from attendees who wish to make suggestions anonymously. From here amendments to the requirements document can be made as the prototype continues to evolve towards a final product (Evolutionary Prototyping, n.d). This technique of requirements gathering holds a number of advantages as requirements can be created and refined much more rapidly than various other techniques. Additionally "active user involvement means that the new system has a better chance of meeting user needs, eases implementation of the new system, and can reduce training costs" (Turban, Rainer & Potter, 2004, pp. 474). This is another one of the main advantages of this methodology as it will give users hands-on experience with the product throughout development and while it is assumed the forum will have additional users in addition to the particular users who are participating in the prototyping it can still be a good gauge of how users adapt to using the system as it gives end users some experience in using a sample of the final system which can be beneficial once the system goes live and management has some evidence that the system is easy to use.



Business Intelligence Guidebook: From Data Integration to Analytics, Rick Sherman, 2015, pp. 369

The above diagram shows the prototyping project progression. It shows how the end users will participate in testing the prototype and then give feedback. From this feedback adjustments will be made to requirements and the prototype will be revised these steps can be repeated therefore with this methodology of software development requirements are very flexible to change.

- Interviewing

"Interviewing users is the most commonly employed approach to requirements gathering" (S, Robertson. J, Robertson, 2006, pp. 104). Obviously when we want to know an individual's needs we would just ask them, therefore interviewing is the most common approach. Our main stakeholders for this project are the end users of the product therefore our interviewees will be environmental scientists and general members of the public who show interest in environmental issues. Although interviewing is one of the most common approaches for gathering requirements it is not without faults and many factors must be taken into consideration. The success of the interview really hinges on the ability of the interviewer, the interviewees and that the right questions are asked to the appropriate interviewee. For this project we have two different groups of stakeholders therefore we will have a different set of questions for each and hope to understand each perspective. A set of questions that are both closed ended and open ended questions, with additional probing questions asked if the interviewer feels that can get more information out of the interviewee.

- Study Analogous Systems

"The starting point for many projects is often a similar or an existing system" (Requirements, 2012). For our system we have found that the commonly used message board forum is appropriate for what the general consortium of environmental scientists. Currently there are numerous pieces of forum software that exist which already feature a large number of the features the users will require. Here we could explain to users what the other systems are used for and how it could adapt to their specific needs.

Specific Requirements for the System

- Create levels of accounts – A level of authorisation for different types of membership. For example Administrator and Moderator privileges, different capabilities between users (Environmental Scientists and Members of the Public. Also a function where a new member is required approval by the online forum administrator.
- Create and reply to threads – The ability for users to create a thread on certain subjects. With other users having the ability to reply to threads. This is the primary requirement of the system as the project brief was to create a tool for Environmental Scientists to communicate and discuss their research with general members of the public.
- Private messaging segmentation – A private messaging segment of the system where users can engage in private discussions amongst each other.
- Search function – A search function that users can search through threads using topics, keywords or post history by specific members
- Web frontage – The forum should have a web frontage.
- Accessible from tablet, mobile and computers – The devices in which people access the web with are becoming increasingly diverse therefore it is important that the system should be accessible from various devices.

Stakeholder Engagement Plan:

"In order to design stakeholder engagement processes that work, engagement owners need a clear understanding of who the relevant stakeholders are and how and why they might want to engage" (Accountability, 2015, pp.23). Therefore it is important to profile and gauge the stake each stakeholder has and how we can engage them in the analysis, modelling and design of the system. We have already established our development methodology which is Evolutionary Prototyping under the Rapid Application Design methodology tree. "The common practice in prototyping and evolutionary process, apart from the iteration is the involvement of users and stakeholders throughout the development process as part of the test and evaluation activity" (Hall & Fernandez-Ramil, 2007, pp.200) therefore not only are stakeholders involved in each iteration of the prototype they are commonly involved in the test, evaluation and final implementation of the system. We can certainly conclude in this methodology engagement of stakeholders is of greater emphasis than traditional SDLC as end users (or in our case a sample group of the end users) will be included throughout the development process and their engagement is imperative with the success of the project. This is one of the main advantages of this methodology it will give users hands-on experience with the product throughout development and while it is assumed the forum will have additional users in addition to the users who participate in the prototyping.

Stakeholder	**Interest**	**Engagement**	Importance
Global Consortium of Environmental Scientists	The project sponsor the global consortium wish that this project will provide a tool for greater communication between environmental scientists and the community.	Once the prototypes have evolved through workshops and user feedback has been collated leading to the system requirements have been fully refined. The requirements specification can be submitted to the 'Project Sponsor' as defined in the system request as the Global Consortium of Environmental Scientists. From here they will review the final requirements documents and give their approval to move ahead with testing, integration and implementation of the final system	High
Environmental Scientists	This project will provide environmental scientists with a tool allowing them communication with members of the public and access to the public's thoughts to their research.	Our development methodology very much includes end users of the system in its development. Attendance or participation in workshops is imperative to the systems success. The feedback given will help to	Medium

		refine requirements of the system and can inform analysts of any changes they wish to be made	
World Bank	Their stake is providing funds for the development of the system.		Low
Members of the Public	For members of the public who are found to be interested in societal environmental issues, the system will provide a new tool for to gather information and communicate with environmental scientists about issues that interest them.	Along with the environmental scientists these members of the public will be the main end users of the system. Same as the environmental scientists their attendance or participation in prototype workshops is necessary for the systems success. Participation in surveys post workshop will help to provide further feedback, refining and adding to requirements which will help the prototype continue to evolve towards the final product.	Medium

Numerous considerations must be made when engaging with stakeholders. Planning and organisation of the workshop are paramount and also further considerations must be made for each attendee to ensure the time and cost of the workshop are optimised. Analysts should have some understanding of the profiles of each stakeholder they engage to avoid risks such as “conflict between participating stakeholders, unwillingness to engage, participation fatigue” (Accountability, 2015, pp. 34) and a range of other risks.

Documents and Diagrams Suite

Using the requirements gathering techniques described in the requirements gathering section our requirements definition has been created and what this system is required to do has been determined. From here the information we have gathered can be presented in a series of diagrams and documents. The accepted standard notation in the industry is known as the Unified Modified Language (UML) and "almost all object-oriented development projects today use these models to document and organize the requirements obtained during the analysis workflow." (Dennis et al, 2015, pp. 119)

- Requirements Specification Document

Throughout our prototyping process we must document the requirements elicited from the stakeholders, we do this throughout the process as with our methodology requirements are often changed and refined. The output of this requirements gathering will be in the form of a requirements specification document. The requirements specification "defines or specifies the system that needs to be produced in order to fulfil the stated requirements" (Hall & Fernandez-Ramil, 2007, pp.166). Our requirements specification will include a concept map using a specific CASE tool software. This will allow us to easily define and display the non-functional and functional requirements of the system. In addition to this a document listing the requirement grouped under the type of requirement.

- Use Case Diagrams

A use case diagram is defined as a diagram that "illustrates in a very simple way the main functions of the system and the different kinds of users that will interact with it" (Dennis et al., 2015, pp. 121). This diagram is utilised to show at a very high level a functionality of a system to either a user or system benefits, this user or system external to the subject is described in the use case diagram as an 'actor'. We can determine which the major use cases for our system are by reviewing the requirements specification document. An example of a use case diagram specific to our project would be a diagram showing the process of creation, searching and replying to threads on the forum. Here the actors would be users of the forum either a member of the public or an environmental scientist. So the use case diagram would show the user creating a subject thread and another user replying to the topic.

- Activity Diagrams

"Activity diagrams portray the primary activities and the relationships among the activities in a process" (Dennis et al., 2015, pp.131). It basically shows in greater detail than a use case diagram the logical flow of a business process. An example for our project would be the process of creating a new membership, the diagram starts with an initial node and would show the workflow of the process like this: the database would check if the specific username is unique, if the username is already in use it will redirect the user to create a different username when confirmed the user will get a message saying a verification email has been sent, the system will then send an automated email for verification to ensure the user is human and in fact legitimate. The user will click the verification link thus creating the membership and the end of the process which in the diagram will be displayed as a decision node.

Bibliography:

Hall, P., & Fernandez-Ramil, J. (2007). *Managing the Software Enterprise: Engineering and Information Systems in Context* (1st ed.) Boston, MA: Cengage Learning EMEA

Robertson, J., & Robertson, S. (2013). *Mastering The Requirements Process: Getting Requirements Right* (3rd ed.) New Jersey, NJ: Pearson Education

Accountability. (2015). *AA1000 Stakeholder Engagement Standard: Revision for Public Content.* Retrieved from http://www.accountability.org/images/content/7/9/792/AA1000SES%20Revision%20for%20Public%20Comment-%20June%202015.pdf

Guideline: Requirements Gathering Techniques. (n.d). Retrieved from http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/guidelines/req_gathering_techniques_8CB8E44C.html

Evolutionary Prototyping. (n.d). Retrieved from http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg3.htm

Dalcher, D., & Brodie, L. (2007). *Successful IT projects*. London: Thomson Learning.

Dennis, A., & Wixom, B. (2012). *System analysis and design* (5th ed.). Hoboken, NJ: John Wiley.

Dooley, J. (2011). *Software development and professional practice*. New York, N.Y.: Apress.

Langer, A. (2012). *Guide to software development designing and managing the life cycle*. London: Springer.

Li, M. (2005). *Unifying the software process spectrum International Software Process Workshop, SPW 2005, Beijing, China, May 25-27, 2005 : Revised selected papers*. Berlin: Springer.

Lucia, A. (2013). *Software engineering International Summer Schools, ISSSE 2009-2011, Salerno, Italy. Revised tutorial lectures*. Berlin: Springer.

Murch, R. (2001). *Project management: Best practices for IT professionals*. Upper Saddle River, NJ: Prentice Hall.

Pros and Cons of Choosing Off-the-Shelf Software Products 1. (n.d.). Retrieved September 6, 2015, from http://blog.aciron.com/pros-and-cons-of-choosing-off-the-shelf-software-products/

Putnam, L., & Myers, W. (n.d.). *Five Core Metrics: The Intelligence Behind Successful Software Management*.

Rapid Application Development. (n.d.). Retrieved September 6, 2015, from http://www.blueink.biz/RapidApplicationDevelopment.aspx

System Request Template. (n.d.). Retrieved September 6, 2015, from http://www.wiley.com/college/info/dennis241008/resources/Resources/proj2.htm - See more at: http://reffor.us/index.php#sthash.H8pjD9ch.dpuf

Tilloo, R. (n.d.). RAD Model In Software Engineering. Retrieved September 6, 2015, from http://www.technotrice.com/rad-model-software-engineering/