CORE
Metadata, citation and similar papers at core.ac.uk
Provided by WestminsterResearch

WestminsterResearch
http://www.wmin.ac.uk/westminsterresearch

Learning how to discover requirements.

Ljerka Beus-Dukic[1]
Ian Alexander[2]

[1] School of Informatics, University of Westminster
[2] Scenario Plus Ltd, London, UK

Copyright © [2008] IEEE. Reprinted from the 3rd International Workshop on Requirements Engineering Education and Training. IEEE, pp. 12-14. ISBN 9781424440863.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Westminster's products or services. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.
Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of the University of Westminster Eprints (http://www.wmin.ac.uk/westminsterresearch).

In case of abuse or copyright appearing without permission e-mail wattsn@wmin.ac.uk.

Learning How To Discover Requirements

Ljerka Beus-Dukic
University of Westminster, London, UK
L.Beus-Dukic@westminster.ac.uk

Ian Alexander
Scenario Plus Ltd, London, UK
iany@scenarioplus.org.uk

Abstract

Requirements Engineering (RE) is traditionally taught in academia using an RE process which starts from a well-defined problem. Our approach focuses on the early stages of requirements discovery where students have to learn both about the application domain and about what stakeholders feel is the problem. The approach comprises all the basic elements of requirements, and ways to discover them using many small discovery cycles.

In this position paper we outline the rationale for our approach and reflect on our initial experiences with teaching undergraduate RE module using this approach.

1. Introduction

Traditionally, RE is taught using the development life-cycle as its focus. Students are taught how requirements are elicited, analyzed and negotiated, specified, and managed over the duration of a software development project. They are normally presented with a set of various methods and techniques which they can use in commonly recognised processes of Elicitation, Analysis, Negotiation, and Management [6, 8, 10].

A typical premise is that we know what the problem is and requirements are sitting somewhere waiting to be elicited, organized and managed. Unfortunately, well-defined problems do not easily come by both in industry and academia. Our students become noticeably aware of this fact for the first time when faced with the task to define a topic for their final year project. What they are not aware of yet is that this situation is similar to what they will encounter in industry as graduates.

There are plenty of textbooks [2, 6, 8, 10] and tools designed to help organise and manage requirements once you have them, but coverage of the critical early stages of requirements discovery is patchy. As Kotonya and Sommerville put it in their textbook [6]: "*Structured methods of requirements analysis ... are not particularly useful for the early stages of analysis where the application domain, the problem, and the organisational requirements must be understood.*"

How to go from a situation where nobody knows what problem ought to be solved, to a position where everybody agrees what the problem to be solved is?

1.1 Requirement Elements

It is not possible to start writing formal requirements until we know who the stakeholders are, what their goals are, what the context is, and so on. These things are not "requirements" in the narrow sense of verifiable contractual statements which appear in requirements specification, but defining them takes us to the point where we know what problem we want to solve is. Goals, scenarios and so on contribute to forming individual requirements which we call "requirement elements". "The requirement" in the broad sense means a network of inter-related requirement elements – a requirement that satisfies a goal, is justified in a rationale model, is using terms defined in the project dictionary, etc. This is a richer structure than an old-fashioned list of statements, and it fulfils its purposes better.

Our approach therefore focuses on different ways of discovering these requirement elements in several discovery contexts (Table 1). Requirements are human needs and come from people (individuals, groups). Things (requirements prototyping, reverse engineering from existing products, requirements reuse) can help to discover requirements; however, the candidate requirements discovered from these things need to be validated with people.

Different projects vary considerably in the importance of the different requirement elements. In practice, therefore, projects sometimes entirely omit some elements. For example, a project with a simple stakeholder structure might omit Stakeholder analysis.

2008 Requirements Engineering Education and Training (REET'08)
978-0-7695-3628-6/08 $25.00 © 2008 IEEE

Authorized licensed use limited to: University of Westminster. Downloaded on June 9, 2009 at 06:14 from IEEE Xplore. Restrictions apply.

The resulting lack could be compensated for by additional explanation of stakeholder issues in Scenarios, Definitions, or Rationale.

Table 1. Requirement elements and discovery contexts

Discovery contexts / ***Requirement Elements***	From Individuals	From Groups	From Things
Stakeholders			
Goals			
Context, Interfaces, Scope			
Scenarios			
Qualities and Constraints			
Rationale			
Definitions			
Measurements			
Priorities			

1.2 Discovery Cycle

A generic discovery process can be drawn as a simple inquiry cycle [3] (Figure 1). An inquiry cycle is a cycle of activities, carried out by a team, to enable them to inquire effectively what should be developed as a product. What all inquiry cycles have in common is a period of action followed by a period of reflection [7]. Both are necessary: action to get on with the work; reflection to consider whether the work is complete, or going in the right direction. It is no good waiting until the end of the requirements phase before trying to validate each finely polished requirement. You need to do that right away, checking each little discovery as you go along. So there is not one big requirements process, but many little inquiry cycles.

2. RE module contents

A Requirements Engineering module has been taught at the University of Westminster since 2001. It is a core module for the final year Bachelor of Science Software Engineering students, and an optional module for students of other computing courses over a 12-week semester. Until 2007, the content of the module was largely based on traditional RE process and textbook [6]. However, this year we have adopted our new approach. We believe it is better because it is a richer, more realistic model of what projects need to do, and how they vary. Not everyone needs scenarios or goals or stakeholder analysis, but many do.

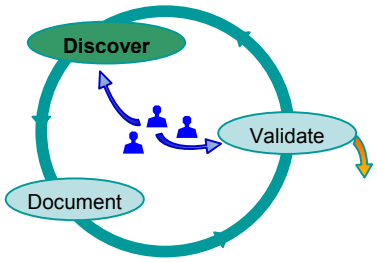

Figure 1. Inquiry cycle for creating requirements

2.1 Teaching approach

The teaching of the RE module is done using a traditional method of lectures and tutorials in blocks of four hours a week. The material provided in lectures [4] covers majority of requirements elements and discovery contexts (Table 2) and is based on our forthcoming textbook [1]. The main purpose of the lectures is to provide sufficient theoretical knowledge before students can start learning by doing in tutorials.

Table 2. Lecture plan

Week	*Lecture*
1	Introduction
2	Goals
3	Stakeholders
4	Context, interfaces, scope
5	Requirements from individuals
6	Requirements from groups
7	Functionality
8	Qualities and constraints
9	Prioritisation
10	Requirements from things
11	Tools
12	Revision

Student practice requirements discovery in weekly two-hour tutorials. Depending on the type of inquiry, they work in teams of two to five members. Students are encouraged to communicate with each other and the teacher. Interaction within the teams involves

2008 Requirements Engineering Education and Training (REET'08)
978-0-7695-3628-6/08 $25.00 © 2008 IEEE

Authorized licensed use limited to: University of Westminster. Downloaded on June 9, 2009 at 06:14 from IEEE Xplore. Restrictions apply.

adopting various roles [11] appropriate to the application domain under discussion and type of inquiry. The progress of each team's inquiry is closely monitored by the teacher and any current problem is discussed with the affected team. Feedback on teams' findings is provided by the teacher at the end of tutorial.

3. Discovering requirements for a final year project

The student assessment on the RE module is entirely coursework based. There are four coursework assignments: 1) Individual report on early requirements analysis (20% of the overall mark); 2) Group report on use case modeling (20%) [5]; 3) Individual report on a specific topic from RE chosen from a set of given topics (30%); 4) Individual report on project requirements (30%).

In their final assignment students were asked to:

- Write a brief problem statement for their project.
- Identify all stakeholders and their roles, and document these using an Onion model.
- Document models of key aspects of their project:
    - The required functionality (Use Case Diagram)
    - The structure of application domain information (Domain Model)
- Write a list of all functional requirements and indicate their priorities.

Each part of this assignment was assessed for: adequacy, correctness, coverage, readability, and consistency.

The rationale behind this assignment was to assess students' ability to apply what they have learnt in the RE module on their own final year project, which is the first complete software project they have to do. It was possible to devise such an assignment because the students are taught the RE module in the first semester of their final year when they have already selected a topic for their final year project. Majority of students invent a topic; very few select the one proposed by supervisors.

This was in effect a bespoke assignment. Although the assignment specification was the same for all students, the provided answers were unique for each student as they were based on requirements for students' own individual projects. There were 55 students in this year's cohort which provided us with enough evidence about the effectiveness of our new approach to teaching RE. For example, stakeholder analysis is one aspect of the new approach which was enthusiastically taken up by students and majority applied it well. Similarly, most of the students produced a prioritised list of valid functional requirements at the appropriate level. However, there is room for improvement as a good number of students still find writing problem statements difficult.

4. Conclusion

This was the first year we employed this more realistic and pragmatic approach to teaching RE using combination of requirements elements and discovery contexts. The achieved results and positive student feedback encourage our hope that this approach will help turning ill-defined problems to well-defined ones.

5. References

[1] Alexander, I. F., and L. Beus-Dukic, *Discovering Requirements: How to Specify Products and Services*, Wiley, 2009.

[2] Alexander I. F., and R. Stevens, *Writing Better Requirements*, Addison-Wesley, 2002.

[3] Alexander, I. F., *Requirements Engineering as a Co-operative Inquiry*, Requirements Engineering 3:130-137, 1998.

[4] Beus-Dukic, L. and I. F. Alexander, *Requirements Engineering*, Lecture Slides, University of Westminster, 2007.

[5] Beus-Dukic, L. and C. Myers, *Use and Abuse Cases*, 1st International Workshop on Requirements Education and Training, Paris, France, 2005.

[6] Kotonya, G., and I. Sommerville, *Requirements Engineering: Processes and Techniques*, Wiley, 1998.

[7] Heron, J. *Co-operative Inquiry, Research into the Human Condition*, Sage, 1996.

[8] Leffingwell, D., and D. Widrig, *Managing Software Requirements: A Unified Approach*, Addison-Wesley, 2000.

[9] Nuseibeh, B., and S. Easterbrook., *Requirements Engineering: A Roadmap*, In: *The Future of Software Engineering*, 22nd International Conference on Software Engineering, ACM, Limerick, Ireland, 2000.

[10] Robertson S.and J. Robertson, Mastering the Requirements Process, Addison-Wesley, 1999.

[11] Zowghi, D., and S. Paryani, *Teaching Requirements Engineering through Role Playing: Lessons Learnt*, Proc. of the 11th IEEE International Requirements Engineering Conference, 2003.

2008 Requirements Engineering Education and Training (REET'08)
978-0-7695-3628-6/08 $25.00 © 2008 IEEE

Authorized licensed use limited to: University of Westminster. Downloaded on June 9, 2009 at 06:14 from IEEE Xplore. Restrictions apply.