Requirement Engineering Education in the UK, an Empirical Study

Russell Lock
Department of Computer Science, Loughborough University, Loughborough, United Kingdom
r.lock@lboro.ac.uk

Abstract

The quality of software is critically dependent on the quality of Requirements Engineering activities undertaken during software development. This paper outlines a survey of Higher Education (HE) institutions in the UK undertaken to determine the nature of the topics covered relating to Requirement Engineering, and the extent to which such topics are practically taught and assessed. Very few surveys of Requirement Engineering within HE have been conducted, and, to the authors knowledge this will be the first significant one published which focussing on HE in the UK. The paper concludes that a number of key issues exist in the UK provision for teaching in this area, which impact on the ability of industry to leverage the skills gained by students whilst studying at university.

Keywords: Requirement Engineering, Survey, Higher Education

1.0 Introduction

The principle objective of the research is to investigate the state of Requirement Engineering within Higher Education (HE) for Computer Science in the UK, in order to determine to what extent it supports the needs of graduate employers engaged in in RE activities. In order to achieve this, a survey was developed for HE to determine whether RE was explicitly taught as a topic and, if so, the subjects it included, the depth of coverage, how it was taught, and how it was assessed.

The structure of the paper is as follows. Previous research relating to HE surveys and curriculum design will be explored, along with an examination of current best practice for the teaching of Requirements Engineering in Higher Education, from a pedagogical perspective, and in terms of expected content and depth. The methodology for the survey will then be discussed. The key results from the survey are then examined, followed by discussion on the limitations of the research. Finally, conclusions are drawn on the research conducted.

2.0 Background

Theory on the Diffusion of Innovations [6] emphasises the importance of five key characteristics in relation to the rate of uptake for technology: relative advantage, compatibility, complexity, trialability and observability.

current software engineering jobs. The study highlighted key areas of shortfall for teaching in Computer Science, namely: testing, quality assurance, requirements gathering and analysis, project management, user interface design and

rather than prescriptive subject specifications, including the Bologna Process [18], ECTS (European Credit Transfer and Accumulation project) [19]

general SE texts. Therefore the list was supplemented by the RE topics covered by two longstanding SE textbooks, Sommerville [23] and Pressman [24].

3.0Survey Methodology

The survey data (the questions for which can be found in Appendix A), was gathered electronically, and was distributed by the CPHC (College Professors and Heads of Computing). The survey had a return rate of 43 respondents from a total of 104 potential universities. In all cases the departments represented were either Computer Science, Computing, Informati

Figure 1: Responses by League Position as determined by the Guardian 2016 subject rankings

In terms of teaching approach within HE, the survey was designed to determine the mode of delivery for topics so, for example whether a given topic was being practiced within lectures, through coursework etc, or simply summatively assessed through assessment mechanisms such as exams. The survey results were processed through the IBM SPSS analytics suite, and, due to the monotonic, ordinal nature of the questions Spearman correlations were used for the correlations reported in the following section.

4.0Key Survey Results

The following subsections explore the results of the data analysis relating to five key themes which emerged from the results.

4.1 Dedicated Requirement Engineering Modules

Respondents were asked whether their degree scheme had a dedicated Requirement Engineering Module rather than teaching the subject within Software Engineering content, or taught in a decentralised manner throughout their programmes. 15 (58%) of the respondents reported dedicated modules for this topic. It is possible that the dedicated RE modules are more likely to be used within Software Engineering degree schemes (given their bias towards material in that sub-discipline), however a test for correlation between degree scheme and the use of dedicated RE modules showed a significant unexpected negative correlation between them (-0.52) in the data set. Information was collected within the survey relating to module titles under which RE material was taught, and while not complete in reporting (this question was optional), this showed that RE is more commonly taught under the module heading of Development, implying the bundling of Requirements-Design-Implementation within Software Engineering degree schemes rather than isolated standalone modules.

4.2 Hours of Tuition

Respondents were also asked how many hours of tuition were given for RE topics. The results showed significantly varying amounts of tuition ranging from 3 hours to 120 hours during a degree scheme with an average of 35 hours, which appears to be a slight improvement on the evidence gathered by the 1995 study by Macaulay [14].

learning for students on a given course, as could the techniques discussed/applied/assessed under these four headings.

On cursory inspection there may

theories put forward in HE [27], especially for approaches that rely on direct observation of real world issues not evident from existing documentation/models.

The concepts of functional and non-functional requirements are the only topics universally taught (these are explicitly listed in the 2007 QAA Subject Specific Benchmark for Computing) with UML, use cases and user requirements having near complete coverage across institutions. However, the complexity of UML raises concerns regarding depth of coverage when considering the average number of hours put to RE topics as a whole as discussed earlier. With the exception of UML, the Model Based Software Engineering (MBSE) approaches coverage was noticeably poorer, including approaches such as GORE. SysML in particular fared badly (69% of courses had no content on this), industry take-up of SysML has also been limited compared to UML. Further investigation into the data showed that SysML was not being used as a substitute for UML. Unless the teaching of MBSE broadens beyond UML the next generation of RE practitioners may have to rely on androgogical activities, self-interest / internal / external courses etc upon completing their on-boarding processes in order to leverage these approaches going forward. Unfortunately the shortfall in MBSE related teaching currently coincides with a global recession that has seen industry training budgets reduced or even removed unless mandated.

24 Roger P (2014) Software Engineering: A Practitioner's Approach: Roger Pressman, Bruce Maxim: 9780078022128: Amazon.com: Books, 8th ed. McGraw-Hill Education
25 Schumann H, Wendel H, Braukhane A, Berres A, Gerndt A, Schreiber A (2010) Concurrent systems engineering in aerospace From excel-based to model driven design. Proc. 8th Conf. Syst. Eng. Res.
26 Elton L (2010) Research and Teaching: Conditions for a positive link. Teach. High. Educ.
27 Ramsden P (2003) Learning to Teach in Higher Education. Routledge
28 Israilidis J, Lock R, Cooke L (2012) Ignorance management: An alternative perspective on knowledge management in multinational organisations. In: Proc. Eur. Conf. Knowl. Manag. ECKM. pp 493–501
29 Daneva M (1999) Measuring reuse of SAP requirements. In: Proc. 1999 Symp. Softw. reusability - SSR '99. ACM Press, New York, New York, USA, pp 141–150
30 Nuseibeh B, Easterbrook S (2000) Requirements engineering. In: Proc. Conf. Futur. Softw. Eng. - ICSE '00. ACM Press, New York, New York, USA, pp 35–46
31 Frakes WB (ed) (2000) Software Reuse: Advances in Software Reusability. doi: 10.1007/b75206