A Requirements-to-Implementation Mapping Tool for Requirements Traceability

Jorge Esparteiro Garcia[1,3*], Ana C. R. Paiva[2,3]
[1] Polytechnic Institute of Viana do Castelo, Praca General Barbosa, Viana do Castelo, Portugal.
[2] INESC TEC, Rua Dr. Roberto Frias, Porto, Portugal.
[3] Faculty of Engineering, University of Porto, Rua Dr. Roberto, Frias, Porto, Portugal.

* Corresponding author. Tel.: +351 258809610; email: jorgegarcia@esce.ipvc.pt
Manuscript submitted October 5, 2015; accepted December 5, 2015.
doi: 10.17706/jsw.11.2.193-200

Abstract: Software quality is a major concern in software engineering, particularly when we are dealing with software services that must be available 24 hours a day, please the customer and be kept permanently. In software engineering, requirements management is one of the most important tasks that can influence the success of a software project. Maintain traceability information is a fundamental aspect in software engineering and can facilitate the process of software development and maintenance. This information can be used to support various activities such as analyzing the impact of changes and software verification and validation. However, traceability techniques are mainly used between requirements and software test cases. This paper presents a prototype of a tool that supports the mapping of functional requirements with the pages and HTML elements of a web site. This tool helps maintaining requirements traceability information updated and, ultimately, increasing the efficiency of requirements change management, which may contribute to the overall quality of the software service.

Key words: software requirements specification, requirements traceability, requirements management.

1. Introduction

Requirements Engineering (RE) is related to the process of eliciting requirements and needs and developing them into a detailed, agreed requirements documented and specified in such a way that they can serve as the basis for all other development activities [1]. It is estimated that 56 percent of the discovered errors typically found in a project are related to the requirements specification [2]. Reducing this type of errors is the single most effective action developers can take to improve project outcomes.

Traditionally, most RE tasks, including the requirements elicitation, are carried out in the beginning of the system development lifecycle [3]. However, when the software application is a service provided continually over time, as are web applications and services, requirements may change or new requirements may come up for several reasons, for instance, new laws, new needs, etc. So, being able to maintain requirements updated over time may be a challenge.

Keep traceability information between software artifacts and requirements may be useful to evaluate the impact of change requests and help maintenance activities. Software traceability has been increasingly recognized as an important quality of a well-engineered software system [4]. It is defined by the Center of Excellence for Software and Systems Traceability (CoEST) as *the ability to interrelate any uniquely identifiable software engineering artifact to any other, maintain required links over time, and use the resulting* *network to answer questions of both the software product and its development process* [5]. Specifically, requirements traceability has been demonstrated to be an important contribute to organizations that make proper use of traceability techniques [6].

Despite the benefits resulting from the use of requirements traceability techniques, the poor tool support is perhaps the biggest challenge to its implementation [6], [7]. Furthermore, the use of requirements traceability tools is just used for about 50% of the software engineering industry [8]. One of the main reasons for this low rate is because, in the existing requirement management tools, it exists poor support for traceability and tools are inadequate for the needs of the software engineering industry [9]. Kannenberg and Saiedian [6] states that creating cost-effective requirements traceability tools would serve to greatly improve the practice of traceability in the software engineering industry. In addition, Di and Zhang [10] stated that the recovering traceability links between requirement and other artifacts is not yet well investigated and there is a stringent need for an automatic tracing method to trace high-level requirements to other software artifacts which are expressed in natural language and may evolve autonomously. Moreover, these techniques are used mainly between requirements and software test cases and the majority of these tools are unable to automatically generate and maintain traceability relationships [11].

Thus, the goal of this work is to develop a new approach and tool for requirements management based on traceability. The tool maps the functional requirements with the implemented functionalities (pages and HTML elements) of the website under analysis. The mapping information will help increasing the efficiency of the requirements change management process, based on requirements traceability, which may contribute to the overall quality of the service provided.

The remainder of this paper is organized as follows: next Section presents an overview of requirements engineering and the requirements traceability task with the existing tools. Section 3 gives an overview of our approach and presents the developed tool to help the process of requirements traceability. Related work is discussed in Section 4 and conclusions and future directions are presented in Section 5.

2. Background

Requirements Engineering (RE), also called requirements analysis, is the process of determining the necessary conditions for a new or modified software product. Zave [12] defines RE as the branch of software engineering concerned with the real-world goals for functions and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.

RE is an iterative process in which activities are interleaved. The processes used for RE depend on the application field and on the organization developing the requirements. However, there are some general activities [13] common to all processes: elicitation and specification; validation; and management.

The requirements elicitation, sometimes also called requirements discovery, is the process of discovering, reviewing, documenting, and understanding the stakeholder’s (e.g., end-users, managers, domain experts) needs and constraints for the system. Requirements specification is the process of documenting the stakeholder's needs and constraints clearly and precisely. It is necessary to describe fully what the application will do and how it is expected to perform. Requirements validation is an iterative process of ensuring that the system requirements are complete, correct, consistent, clear, feasible, necessary, prioritized, unambiguous and verifiable. There are some validation techniques like requirement reviews, prototyping and test-case generation. Requirements Management (RM) can be described as the process of managing changing requirements during the requirements engineering process and system development [14].

During the development of a system, and also after deployment, new requirements emerge and it is necessary to keep track of them. In the last years, RM is increasingly recognized as crucial [15], due to the needing of writing requirements readable and traceable, in order to manage their evolution over time.

There are three areas related to requirements management [16]: change management; requirements attributes; requirements traceability. Change management describes the procedures and tasks needed to be followed when analyzing a requirements change. Sommerville [14] proposed a workflow process to change management. It starts with problem analysis during which problems are discussed and changes are proposed. Then, changes and their impact are analyzed. Finally, the changes are implemented and documents are modified to reflect the changes. In the last years, RM is increasingly recognized as crucial [15], due to the needing of writing requirements readable and traceable, in order to manage their evolution over time. The requirements evolution consists in the changes made to the requirements after the initial deploy of the requirements specification document [17]. A software requirements specification evolve when a system adjustment is performed, its behavior changes and it is necessary to change some of the requirements initially specified.

Software traceability is recognized as a critical success factor in software development [18] and has been recognized as an important quality of a well-engineered software system [19]. Requirements traceability has been demonstrated to provide many benefits to organizations that make proper use of traceability techniques [6]. Requirements traceability is the ability to describe and follow the life of a requirement, in both a forward and backward direction [19]. The objective of forward traceability is to ensure that each requirement is implemented in the product and that each requirement is thoroughly tested [20]. Backward traceability can verify that the requirements have been kept current with the design, code, and tests [20]. The most used method for tracing requirements to their outcomes is a Requirements Traceability Matrix (RTM).

A RTM is defined as a table that illustrates logical links between individual functional requirements and other system artifacts [21]. However, manual traceability methods may not be feasible because traceability links that need to be captured grows exponentially with the size and complexity of the software system [22].

Requirements Traceability is also an important methodology during the maintenance phase because the initially defined requirements often change during the lifecycle of the project and it is very important to assess the impact of those changes. Traceability allows to determine what requirement, test cases or other artifacts need to be changed and can also determinate the costs and risks associated with that change. Sherba [23] added another question that a traceability approach should also answer: how traceability relations are going to be viewed and queried.

Traceability analysis is related to the process of tracking forward or backward through a network of interrelationships between components of a system and its documentation (Table 1). The approach proposed in this paper addresses these questions to support efficiently the requirements management activity.

Table 1. An example of a Requirements Traceability Matrix [6]

System Requirement	Software Requirement	Design Element	Code Module	Test Case
A005-00150-80-0505	005-00150-80-0112	Airspeed Calculation	Calculate_airspeed()	Tc_103.doc
A005-00150-80-0506	005-00150-80-0234	Airspeed Display	Display_airspeed()	Tc_125.doc

3. Our Approach

This section describes the proposed approach and tool for managing software requirements of a web application. The main idea is to trace functional requirements with the implementation artifacts like webpages and HTML elements and controls. Identifiers for each traceability element such as requirement identifiers, web page URL and HTML element identifiers are stored within the database.

Fig. 1 shows a diagram of the proposed solution. The tool runs through a bookmarklet and runs on any web browser. This allows the traceability tool to work with any available website. To develop a tool for the proposed traceability approach we needed to divide the project in three different tasks:

- Identify the information to be captured and traced
- Identify how this information is captured and managed in the database
- Create the tool itself with the different traceability views and functionalities to analyze the traced requirements with the implementation artifacts.

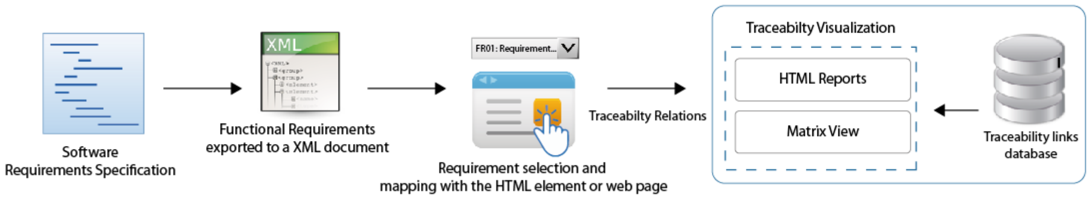

Software Requirements Specification
XML
Functional Requirements exported to a XML document
FR01: Requirement...
Requirement selection and mapping with the HTML element or web page
Traceabilty Relations
Traceabilty Visualization
HTML Reports
Matrix View
Traceability links database

Fig. 1. Functional requirements mapped with the HTML elements and web pages.

The first task toward the development of the tool was to identify the information that will be traced. Since we are analyzing the functional requirements of a web application, the traceability information necessary is the set of functional requirements, and the webpages and HTML elements and controls.

The next task was to identify how the information of the functional requirements and webpages would be captured. Firstly, to fulfill this task, the functional requirements of the website under analysis are mapped with the web pages and HTML elements present in the website.

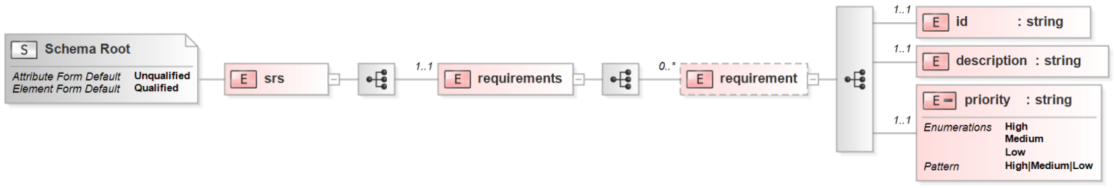
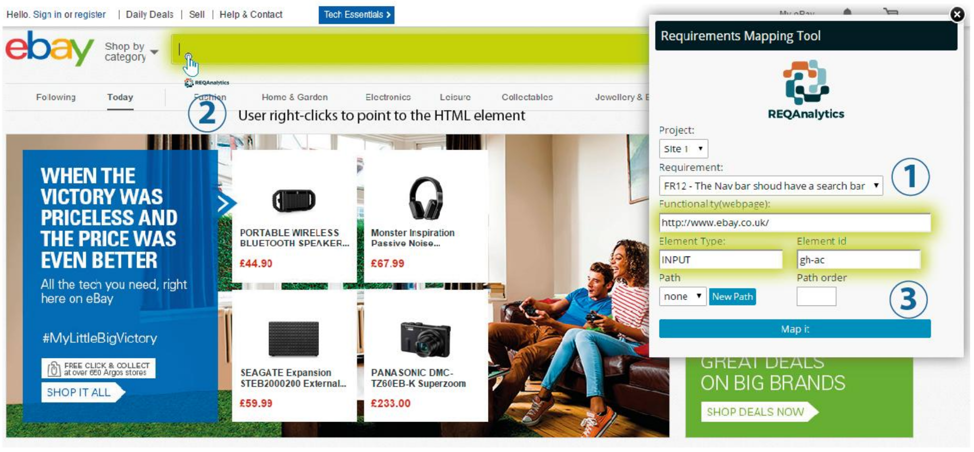


Fig. 2. XSD Schema Diagram of the XML file necessary to import the functional requirements.

The mapping may be established between a single requirement and a web page, but it can also be the relation between a sequence of different webpages or HTML/elements and a requirement. With this traceability tool it is possible to map a requirement to multiple implementation artifacts and an artifact to multiple requirements. To perform this mapping, an XML document is previously generated with the functional requirements defined in the requirements specification of the website. This XML Schema was created during this project and intends to ease the task of describing the functional requirements. This XML document must comply with the definitions of the XML Schema created for this purpose. The XML Schema Diagram is shown in Figure 2. We have chosen XML on this approach due to the fact that XML is the most widely used data-interchange language among different tools and applications. Nevertheless, this could be done with another data-interchange format like JSON. Then, this XML document with the functional requirements is imported to the system to be mapped. This mapping is established manually using a high-level mapping web tool included in the system. As Hayes and Dekk [24] stated, the human analyst is required as an active participant in the traceability process.

This web tool can work with any web application or website since it was developed inside a bookmarklet that works within any web browser. To use this bookmarklet, the user that wants to perform the mapping has to access a specific link where the tools is located or just can access it using a previous saved bookmark. To establish the mapping between requirements and the web elements, i.e. to create the traceability links the human interaction is needed.


Hello. Sign in or register | Daily Deals | Sell | Help & Contact
Tech Essentials
ebay
Shop by category
Following
Today
Fashion
Home & Garden
Electronics
Leisure
Collectables
Jewellery &
2
User right-clicks to point to the HTML element
WHEN THE VICTORY WAS PRICELESS AND THE PRICE WAS EVEN BETTER
All the tech you need, right here on eBay
#MyLittleBigVictory
SHOP IT ALL
PORTABLE WIRELESS BLUETOOTH SPEAKER...
£44.99
Monster Inspiration Passive Noise...
£67.99
SEAGATE Expansion STEB2000200 External...
£59.99
PANASONIC DMC-TZ60EB-K Superzoom
£233.00
GREAT DEALS ON BIG BRANDS
SHOP DEALS NOW
Requirements Mapping Tool
REQAnalytics
1
3
New Path
Map it

Project: Site 1

Requirement: FR12 - The Nav bar shoud have a search bar

Functionality(webpage): http://www.ebay.co.uk/

Element Type: INPUT

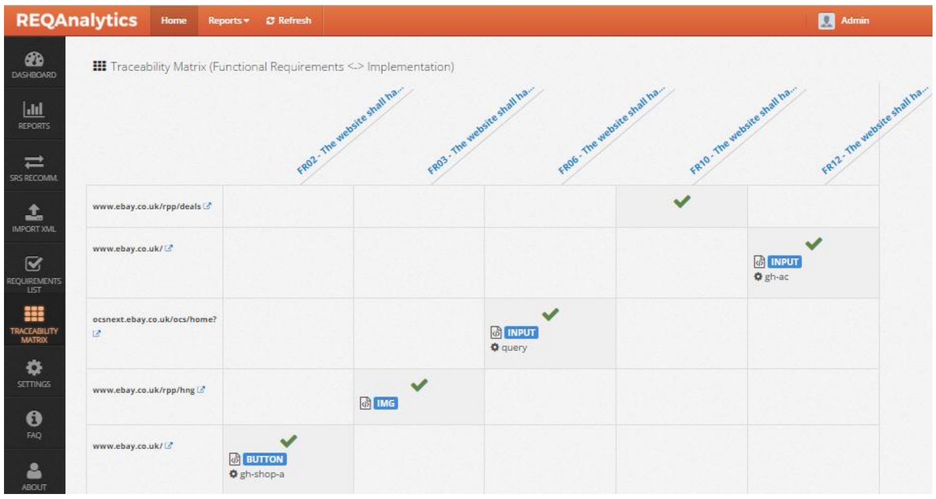
Element Id: gh-ac

Path: none

Path order:

Fig. 3. Mapping tool inside a bookmarklet.

A screenshot of this tool is shown in Fig. 3. The user must select (1) a requirement in a check box (where are all the functional requirements previously imported from the XML document), and point / click (2) on the page and/or HTML element that is related with this requirement. In the case of a requirement related to a sequence of webpages or HTML elements, the user has to select all that artifacts and map it with the requirement. Finally, by clicking on the button "Map it" (3) the tool will save the traceability link between the requirement and the webpage and / or HTML element.



Traceability Matrix (Functional Requirements <-> Implementation)

	FR02 - The website shall ha...	FR03 - The website shall ha...	FR06 - The website shall ha...	FR10 - The website shall ha...	FR12 - The website shall ha...
www.ebay.co.uk/rpp/deals				✓	
www.ebay.co.uk/					✓ INPUT gh-ac
ocsnext.ebay.co.uk/ocs/home?			✓ INPUT query		
www.ebay.co.uk/rpp/hng		✓ IMG			
www.ebay.co.uk/	✓ BUTTON gh-shop-a				

Fig. 4. Traceability matrix mapping requirements with the implementation of a website

The association between the requirements and the web pages/HTML elements is stored in a MySQL database.

Finally, the last task was to create the tool itself with the different traceability visualizations (e.g., Figure 4) and functionalities to analyze the traced requirements with the implementation artifacts. With the information of the traceability links and the functional requirements stored in the database, the tool will provide several reports of traceability visualization like a traceability matrix and lists with the tracing links between the functional requirements and the implementation artifacts.

This tool will be integrated in a recommender system under development called REQAnalytics that will use the information of this traceability links to assist in the task of requirements management in order to possible improve the quality of web applications. With the traceability information between functional requirements and implementation artifacts, this system is able to:

- List the requirements analyzed with the respective mapping with the page and/or HTML element.
- Generate different traceability views of the traceability links.
- Help in the task of requirements management, which contributes to the quality of the website.

4. Related Work

Poor tool support is perhaps the biggest challenge to the implementation of traceability [6].

In a recent study, Li and Walid [25] found that matrices and graphs were preferred to support management tasks, while hyperlinks were preferred to support implementation and testing tasks. Matrices were found appropriate to gain an overview of the traceability, while graphs were found suited to navigating the resulting traces.

There are several research and commercial tools, such as DOORS and Rational RequisitePro by IBM, and CaliberRM by Borland available that provide capabilities for documenting requirements, managing their changes and support traceability between artifacts. In other traceability approaches, there exit some work related to different tools and methodologies for requirements traceability. Saiedian [26] stated that existing traceability tools focus primarily on requirements traceability or traceability among the various artifacts of a software product, however, there is still an open issue in end-to-end traceability. For instance, Sherba and Anderson [23] proposed a traceability, which has an evolution service that analyzes the changes to a set of relationships over time. It analyzes existing links in different versions without evolving the links themselves. Huffman Hayes et al. [27] developed a traceability tool, called RETRO (Requirements Tracing On-target [20]), to trace requirements. RETRO implements both VSM and LSI for determining requirements similarity. Egyed and Grunbacher [28] presented a tool-supported technique easing trace acquisition by generating trace information automatically. Neumuller and Grunbacher [29] developed a traceability environment and introduced in a very small software company where they have found that comparably simple automation techniques are surprisingly effective. Cuddeback et al. [30] presented a framework for the study of analyst interaction with artifacts generated automatically during the tracing.

Gotel and Finkelstein [19] detected that traceability methods were preferred in the industry due to shortcomings in available traceability tools. However, this problem still exists today because manual traceability methods are still preferred by a significant percentage of software organizations [6].

Some research has also been done to investigate the methods to recovery traceability links between design and implementation. For instance, Egyed and Grunbacher [28] proposed a method to recover traceability links between requirements and Java programs by monitoring the Java programs to record the usage of the program classes when scenarios are executed. Despite this, the majority of the related techniques are used mainly between requirements and software test cases and the majority of these tools are unable to automatically generate and maintain traceability relationships [11].

5. Conclusions

In this paper, we presented a tool to map the functional requirements of a software requirements specification of a website with its functionalities, through its webpages and HTML elements. This tool allows to identify traceability links between the functional requirements and the developed features of the website.

The presented methodology for requirements traceability and software requirements management can be applied in an evolutionary context that support and manage functional requirements during software lifetime. Moreover, these techniques are used commonly between requirements and software test cases and the majority of these tools are unable to automatically generate and maintain traceability relationships.

In this paper we demonstrated that creating traceability links between functional requirements and implementation artifacts like web pages and HTML elements may provide a valuable help to the requirements maintenance of a website, since existing traceability methods like traceability matrices are also prone to errors and are vulnerable to changes in the system.

This paper also presents a high-level mapping tool that through a web-based application, maps the functional requirements of the software requirements specification with the implemented functionalities (pages and HTML elements) of the website that contributes for diminishing the problem of poor tool support that is referred in several papers [6] [7]. Although we believe that our approach can be successfully applied to requirements traceability between functional requirements and implemented artifacts, is our intention to include this approach in a project that will enable a deeper analysis for example in the navigation paths and in the workflows of the website.

References

[1] Phol, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer Publishing Company, Incorporated.

[2] Hooks, I. F., & Farry, K. A. (2001). *Customer-Centered Products: Creating Successful Products through Smart Requirements Management.*

[3] Royce, W. (1990). TRW's Ada process model for incremental development of large software systems. 2-11.

[4] Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. *IEEE Trans. Softw. Eng., 27(1)*, 58–93.

[5] CoEST: Center of excellence for software traceability. Retrieved, from http://www.coest.org.

[6] Kannenberg, A., & Saiedian, D. H. (2009). Why software requirements traceability remains a challenge. *CrossTalk Magazine — The Journal of Defense Software Engineering*.

[7] Williams, C. L. (2011). A many-to-many (M:N) relation model to improve requirements traceability. Austin, Texas, U.S.A.

[8] Lempia, D., & Miller, S. (2006). Requirements engineering management, *Proceedings of the National Software and Complex Electronic Hardware Standardization Conference*.

[9] Spanoudakis, G. P. K., Zisman, A., & Pérez-miñana, E. (2004). Rule-based generation of requirements traceability relations.

[10] Di, F., & Zhang, M. (2009). An improving approach for recovering requirements-to-design traceability links. *Proceedings of the 2009 International Conference on Computational Intelligence and Software Engineering* (pp. 1-6).

[11] Hayes, J. H., Dekhtyar, A., & Osborne, J. (2003). Improving requirements tracing via information retrieval. *Journal of Lightwave Technology*, 138–147.

[12] Zave, P. (1997). Classification of research efforts in requirements engineering. *ACM Comput. Surv., 29(4)*, 315–321.

[13] Lowe, D. (1999). Hypermedia and the web: An engineering approach.

[14] ISommerville , I. (2004). *Software Engineering*. Addison Wesley.

[15] Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: A roadmap. *Proceedings of the Conference on the Future of Software Engineering* (pp. 35 –46).

[16] Sommerville, I., & Kotonya, G. (1998). Requirements engineering: Processes and techniques.
[17] Anton, A. I., & Potts, C. (2001). Functional paleontology: System evolution as the user sees it. *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001* (pp. 421–430).
[18] Dömges, R., & Pohl, K. (1998). Adapting traceability environments to project-specific needs, *Commun. ACM*, *41(12)*, 54–62.
[19] Gotel, O. C. Z., & Finkelstein, C. W. (1994). An analysis of the requirements traceability problem. *Proceedings of IEEE International Conference on Requirements Engineering* (pp. 94–101).
[20] Westfall, L. (2006). Bidirectional requirements traceability.
[21] Wiegers, K. E. (2003). *Software Requirements*, Microsoft Press.
[22] Cleland-Huang, J., Chang, C. K., & Christensen, M. (2003). Event-based traceability for managing evolutionary change. *IEEE Trans. Softw. Eng.*, *29(9)*, 796–810.
[23] Sherba, S. A., & Anderson, K. M. (2003). A framework for managing traceability relationships between requirements and architectures.
[24] Hayes, J. H., & Dekhtyar, A. (2005). Humans in the traceability loop. *Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering* (p. 20).
[25] Li, Y., & Maalej, W. (2012). *Requirements Engineering: Foundation for Software Quality*. Heidelberg, Berlin: Springer.
[26] Saiedian, H., Kannenberg, A., & Morozov, S. (2011). A streamlined, cost-effective database approach to manage requirements traceability. *Softw. Qual. J.*, *21(1)*, 23–38.
[27] Hayes, J. H., Dekhtyar, A., & Sundaram, S. K. (2006). Advancing candidate link generation for requirements tracing: the study of methods. *IEEE Trans. Softw. Eng.*, *32(1)*, 4–19.
[28] Egyed, A., & Grunbacher, P. (2002). Automating requirements traceability: Beyond the record and replay paradigm, *Proceedings of the 17th IEEE International Conference on Automated Software Engineering* (pp. 163–171).
[29] Neumuller, C., & Grunbacher, P. (2006). Automating software traceability in very small companies: A case study and lessons learne. *Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering* (pp. 145–156).
[30] Cuddeback, D., Dekhtyar, A., & Hayes, J. (2010). Automated requirements traceability: The study of human analysts. *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference* (pp. 231–240).

Jorge Esparteiro Garcia is an assistant lecturer at Polytechnic Institute of Viana do Castelo (IPVC) since 2005/2006. He received his bachelor's degree in computer science in 2004 from Faculty of Sciences of University of Porto (FCUP) and the master degree in informatics engineering in 2007 from the Faculty of Engineering of University of Porto (FEUP). He is a PhD student at FEUP in informatics engineering. His research interests are in the area of software engineering, web mining and requirements management.

Ana C. R. Paiva is an assistant professor at the Informatics Engineering Department of the Faculty of Engineering of University of Porto (FEUP) where she works since 1999. She teaches subjects like software testing, formal methods and software engineering, among others. Her PhD thesis is entitled "Automated specification based testing of graphical user interfaces". She is a researcher at INESC TEC (www.inesctec.pt) and coordinator of the Software Engineering Research Group (softeng.fe.up.pt/SERGUP) which gathers researchers and post graduate students with common interests in software engineering. She is a member of the PSTQB (www.pstqb.pt) and member of several ISTQB (www.istqb.org) Working Groups.