REQAnalytics: A Recommender System for Requirements Maintenance

Jorge Esparteiro Garcia¹,³,* and Ana C. R. Paiva²,³,α

¹Polytechnic Institute of Viana do Castelo, 4900-347 Viana do Castelo, Portugal
²INESC TEC, ³Faculty of Engineering, University of Porto,
Rua Dr. Roberto Frias, 4200-65 Porto, Portugal
*jorgegarcia@esce.ipvc.pt, apaiva@fe.up.pt

Abstract

In the context of SaaS, where the change requests can be frequent, there is the need for a systematic requirements management process so as to maintain requirements updated and ease the management of changes required to improve the service to provide.

Changes to perform need to be prioritized and their impact on the system should be assessed. The extraction and analysis of the use of the services provided through the web and their relationship to the requirements can help identify improvements and help keep the service useful for longer period of time.

This paper presents REQAnalytics, a recommender system that collects information on the usage of a web service, relates that information back to the requirements, and generates reports with recommendations and change suggestions that can increase the quality of that service. The proposed approach aims to provide reports of the analysis made in a language closer to the business where, for example, it indicates new workflows and navigation paths, identifies the features that can be removed and presents the relationship between requirements and the proposed changes helping to maintain the software requirements specification updated and useful.

Keywords: Software Requirements Specification, Requirements Management, Web Usage Mining

1. Introduction

Requirements engineering is related to the process of eliciting individual stakeholder requirements and needs and developing them into detailed, agreed requirements documented and specified in such a way that they can serve as the basis for all other system development activities [1]. Traditionally, it is carried out in the beginning of the system development lifecycle [2].

However, when the software application is a service provided continually along time, requirements may change or new requirements may come up for several reasons, for instance, new laws and new needs. This is called a Software as a Service (SaaS) which is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. SaaS are a type of software that typically must be available 24 hours a day and should satisfy user needs. They may be the core of a company and their quality should be a major concern.

It is common sense in the field of software development that uncontrolled and outdated software requirements specification leads to many project failures. So, in order to increase the lifetime of a service, it is important to deal carefully with requirements management issues to maintain requirements updated and to prioritize change requests. Otherwise, the
requirements’ documents get outdated becoming useless. In this context, management of change requests and the analysis of their impact may be complex and extremely difficult.

The information about the way a SaaS is used may be helpful to deal with the adaptations and updates needed during its lifetime. Information like number of visits, duration of visit, page views, click path and exit rate, can be used for different purposes as assessment of quality of web-products (i.e., websites, web videos) [3] or to statistical analysis of website data usage [4]. This set of data can be collected through web analytics tools and can be helpful to maintain the requirements, particularly with regard to traceability, extensions and prioritization that, ultimately, can help in the improvement and maintenance of the web systems.

Currently, web analytics tools are able to collect diverse data about the usage of a website but they only generate reports with navigation statistics, duration of navigation on the website and other metrics that are mostly often used simplistically to see which content has more adherence by users [5]. Either, web analytics has focused on analysis and reporting of business metrics of interest mainly to marketers [6].

Despite of continuous research in the field of Web Analytics, there are still some challenges which researchers need to work upon. The analysis and research done till this moment show that some of the open challenges of this field are:

- There are several web analytics tools with large volumes of data, but they do not give any recommendations to the improvement of the website based on the data collected.
- Stakeholders are often skeptical regarding a new form of automated tool support [7]. Therefore, high quality recommendations should be presented to answer the stakeholder’s doubts.
- No focus on the improvement of the software requirements specification (SRS).
- The current web analytics tools have a number of weaknesses. For instance, they do not analyze the service based on different kinds (roles) of users; do not analyze typical navigational paths (which may be useful to define or improve workflows); do not produce reports based on the requirements and, therefore, in a language closer to the business.

The potential of the analysis of usage data is yet to be explored [8]. An analysis directed to the improvement is not currently done and this data is disregarded for the development and improvement of the quality of a web application. Web Analytics tools available [3] do not provide functionalities that enable a more intelligent analysis to suggest improvements to the website, such as suggest new workflows, identify and remove unused features or present more readable and legible reports. In addition, nowadays, the Requirements Management of a website is only done to solve detected problems, satisfy or correct the failures of the software requirements specification [9] not focused in its improvement, and therefore in the improvement of the quality of the website.

Besides analytics tools, there are Recommendation Systems that have been widely used in e-commerce websites to provide user personalization [10] like product, content or service recommendations. Recommendation Systems provide information items estimated to be also valuable for a software engineering task in a given context [11]. Recommender Systems for Software Engineering (RSSE) is a novel approach to support developers in decision making. RSSE can help developers to find alternative decisions in a wide range of software engineering tasks from reusing code to writing effective bug reports. The overall goal is to provide the right information, at the right time, to the right person. This would allow requirements engineers to spend their limited time on more important aspects of the project [11].

On this paper, we propose REQAnalytics, a recommender system that collects data about the usage of a website and suggests changes to the requirements specification.
The remainder of this paper is organized as follows: Section 2 discusses the related work. Section 3 describes the approach of the recommender system developed to assist the requirements maintenance, and Section 4 presents some results achieved with this approach in a case study based on a newspaper website. Finally, Section 5 concludes the paper and sets future directions.

2. Related Work

Research in the field of Requirements Management based on data collected of the usage of website is few [12], or not exactly related with the research proposed on this research work. For instance, Gao [13] proposed a solution where the evolution of software requirements models is based on the feedback collected. Ghezzi [14] presented an approach that automates the acquisition of user-interaction requirements through web logs and analyses them by means of probabilistic model checking to identify navigation anomalies and emerging users behaviors.

In a survey related to software requirements specification in model-driven development, Valderas [15] demonstrated that few of the existing approaches are specifically defined for the specification of web application requirements. Furthermore, once requirements are specified, there is little support for allowing the systematic or automatic derivation of the conceptual model that properly satisfies the software requirements specification.

Currently there are several research works in the field of recommender systems for software engineering (RSSE) [16]. They are mainly focused on the coding phase where the developed tools give recommendations to assist developers on several programming tasks like, suggesting code reuse opportunities [17] or support software requirements elicitation [18]. For instance, Maalej [19] proposed a continuous and context-aware approach for communicating user input to engineering team.

In other related work, Danylenko an Lowe [20] suggest recommendation systems based on context-aware composition to enable a system designer to postpone and automate decisions regarding efficiency non-functional requirements, such as performance. Though, this approach focus on handling efficiency non-functional requirements during the software development process and is carried out before deploy of the system.

A recommendation-based approach to requirements reuse is presented by Dumitru [21]. The proposed recommendation approach is content-based filtering, where a vector of keywords (derived from the description of the new software project) is matched with the keywords extracted from requirements artifacts from the repository of already completed software projects.

A recent recommender system, INTELLIREQ [22] is based on different recommendation approaches that support stakeholders in requirements-related activities such as definition, quality assurance, reuse, and release planning. However, on this approach, the recommender system is used just through the elicitation phase.

The available RSSE [16, 23] differ from our proposal since the data collected to generate recommendations is not based on the data on the use of the website. In addition, our approach differs from all the related works described, because we have developed a recommender system (REQAnalytics) that supports the evolution of the website and allows the requirements maintenance throughout its lifecycle.

Using the data gathered from a web analytics tool, REQAnalytics generates recommendations to the software requirements specification and to the website itself. These recommendations also allow to analyze typical paths taken by users (which may be useful to define or improve workflows) and suggest changes to the functional requirements and to the website in a language closer to the business.
3. The REQAnalytics System

This section describes REQAnalytics, the proposed recommender system. Figure 1 shows an overview of the developed system.

This system is designed to be used through a web browser. Developed in PHP and using a MySQL database as support, REQAnalytics analyzes the web usage and navigation performed along the pages of a web site to generate recommendations to improve the quality of the software requirements specification.

To generate these recommendations, we developed a web based mapping tool included in REQAnalytics that allows to map the functional requirements of the website with its functionalities. To collect the web data usage from the website, REQAnalytics use a web analytics tool, OWA - Open Web Analytics, that allows to gather this kind of data and save it to a database.

![Diagram of REQAnalytics System](image)

**Figure 1. An Overview of the REQAnalytics Recommender System**

The REQAnalytics system is divided in four phases:

- Requirements mapping – map the functional requirements with the functionalities (pages and HTML elements) of the website.
- Collect Web usage data – use of the web analytics tool OWA for collecting web usage data (pages viewed, clicked web elements, click paths, session duration, entry pages, exit pages)
- Analysis of the data collected – the data provided by OWA is analyzed and intersected with the mapping information defined on the first phase.
- Generation of recommendation report – a high level recommendation report is generated with possible improvements of the requirements specification and of the website itself

3.1. Mapping of Requirements

At the first phase, the functional requirements of the website under analysis are mapped with the web pages and HTML elements present in the website.

To perform this mapping, an XML document is previously generated with the functional requirements defined in the requirements specification of the website. Then,
This XML document with the functional requirements is imported to the REQAnalytics to be mapped. This mapping is established using a high-level mapping web tool included in the REQAnalytics system.

To establish the mapping between requirements and the web elements, the user must select a requirement in a check box (where are all the functional requirements previously imported through the XML document), and point/click on the page and/or HTML element that is related with this requirement. The mapping process is similar to the one described in [24].

This mapping information is stored in a MySQL database, which stores the association between the requirements and the web pages/HTML elements. This information allows to analyze what are the HTML elements clicked, the URL of the pages visited and what is the requirement that is associated with this information.

This mapping tool allows to establish simple relations/mappings between requirements and web elements and more complex relations between requirements and a navigation path (i.e., a sequence of actions to perform on the website) to support the mapping of requirements that are associated with more than one webpage or element.

3.2. Collect Web Usage Data

The purpose of this phase is to collect the web usage data available using a web analytics tool. The analytics tool used in this system as referred before is Open Web Analytics. It will collect several data like, pages visited, DOM elements clicked, click paths, entry pages, exit pages and duration of session. This data will be stored in a support database for further analysis.

3.3. Analysis of the data collected

REQAnalytics lists and analyses the web usage data collected by the web analytics tool, with the mapping information stored in the first phase. The aim of this analysis is to identify possible improvements on the website.

For this purpose, the tool analyzes the paths/traces performed by users while visiting the website in order to identify, among others, possible shortest paths, workflow changes (which are the most and least used requirements), the need to create new requirements not present in the previous requirements specification, the need to change the priority of the requirements and to detect patterns among navigation paths.

3.4. Generation of recommendation report

After the analysis made in the previous phase, it is generated a detailed report with several recommendations for improvement of the website under analysis. This recommendation report is built in a language closer to the business (than the language used by existing web analytics tools).

The recommendations presented in the report show the functional requirements associated with the page and/or HTML elements and propose the following recommendations:

1. Show the most and least used website functionalities, mapped with the respective requirement
2. Creation of new requirements
3. Eliminate requirements whose functionality are not used
4. Change the priority of the functional requirements
5. Detect the most used navigation paths
6. Determine shorter paths for specific functionalities
4. Case Study

This section presents the results of the application of REQAnalytics in a real website: Cidade Tomar, a regional newspaper of Portugal.

The main objectives defined for this case of study are:

- Analyze the data collected from the usage of the website and produce recommendations to the software requirements specification and to the website itself
- Identify common navigation paths to different sessions of the website
- Analyze the priorities of the functional requirements of the website

4.1. Implementation

As described in Section 3, the first phase of this process is to identify the functional requirements of the website requirements specification document that will be used by the analysis of the recommender system engine.

Table 1 describes the functional requirements that were identified and used for analysis in this case study:

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF01</td>
<td>The website must have a menu with all the categories of the news</td>
<td>High</td>
</tr>
<tr>
<td>RF02</td>
<td>The first web page must have a section with the latest news</td>
<td>High</td>
</tr>
<tr>
<td>RF03</td>
<td>There must exist a news search box in all web pages</td>
<td>High</td>
</tr>
<tr>
<td>RF07</td>
<td>Top stories on the front page must have an image</td>
<td>High</td>
</tr>
<tr>
<td>RF10</td>
<td>The website must have a Home button redirecting to the first page</td>
<td>Low</td>
</tr>
</tbody>
</table>

The functional requirements described in Table 1 were firstly exported to a XML document with the format accepted by REQAnalytics. Then the mapping relating each requirement to the webpage and/or website HTML element that implements it was established. This mapping was built using the web based mapping tool included in our system.

The web usage data for this case study was collected using the Open Web Analytics (OWA).

The time period used in this study was 2 weeks (14 days) before the recommender system analysis was carried out.

4.2. Results

After this data has been analyzed, REQAnalytics generated a set of recommendations to the priority of the functional requirements (shown in Figure 3) and detected the most used navigation path (shown in Figure 4). Here are some of the observations/recommendations generated by REQAnalytics based on the analyzed data that may help the software engineer to improve the software requirements specification of the website.
• List of the requirements analyzed with its mapping to the page and/or HTML element. This mapping allows to identify if all requirements are implemented on the website. Figure 2 shows the list of mapped requirements.

![List Of Functional Requirements](image)

**Figure 2. Screenshot of REQAnalytics with the List of the Requirements Analyzed with the Respective Mapping**

• The most visited page was http://www.cidadetomar.pt/tema.php?id=1 with 4198 clicks in the period of analysis. This page maps to the requirement with the id “RF10: The website must have a Home button redirecting to the first page”. Therefore, REQAnalytics recommends to increase the priority of the requirement RF10 to high (Figure 3). REQAnalytics also recommends to maintain the priority of the other functional requirements under analysis.

![Screenshot of REQAnalytics with the Requirements Priority Recommendations](image)

**Figure 3. Screenshot of REQAnalytics with the Requirements Priority Recommendations**

• The most traversed path along the website in shown in Figure 4. It was noticed that after visiting Page 2, Page 3 or Page 4 (pages of the news stories), user returns always back to the homepage to read another news. Therefore, this could be a reason for the user to leave the website before it is desired, due to the excessive number of clicks the user has to perform to read the news of the website. Based on this observation, REQAnalytics recommends to redesign the navigation path as shown in Figure 5.
Figure 4. Most Used Navigation Path

After implementing the changes to the possible navigation paths through the site (replacement of paths in Figure 4 by paths in Figure 5), we analyzed the possible improvements in the usage of the website over a 2 week time period. Before the changes were implemented, the average number of pages per visit was 1.7. After the changes, that number increased to 2.11.

Figure 5. Navigation Path Redesigned

Figure 6 shows the results obtained for the period before the analysis carried out by REQAnalytics and the results obtained after the changes have been implemented to the website. This corresponds to an increase of almost 25% in number of pages per visit, which for a newspaper website is a substantial value as it may correspond to an increase of their revenue from advertising.

Summarizing, considering this case study, the observations/recommendations given by REQAnalytics were:

- Increase the priority of functional requirement RF10 from low to high since it was the most used functionality of the website and to maintain the priority of the other functional requirements under analysis.
- Redefine the most used navigation path;
- Identification of the most used pages.functionalities of the website;
- List of the requirements analyzed with the respective mapping with the page and/or HTML element.
The recommendations generated by REQAnalytics are presented in a language closer to the business and with a kind of analysis that existing analytics tools do not offer as they usually only generate reports with the navigation statistics. These results suggest that recommender systems for software engineering can be used in a meaningful way to help the requirements maintenance during the life cycle of a website where the requirements are constantly changing and evolving and therefore help improve quality of the website.

![Figure 6. Pages per Visit Without and After Recommendations Applied Using the Same Period of Time, 14 Days](image)

5. Conclusions and Future Work

This paper presents a recommender system, REQAnalytics that maps requirements with the functionalities of a website that implements them, and relates this information with the web usage data, in order to generate recommendations to the website under analysis.

The proposed approach is illustrated in a case study that examines the website of an online newspaper. This analysis resulted in recommendations in terms of changing or maintaining the priority of a requirement, detection of most used navigation paths and identification of shorter paths for a given functionality provided by the website.

We demonstrated that recommendations generated by REQAnalytics, like the detection of most used navigation paths, significantly improved the number of pages per visit in 25%. The results show that the analysis of the use of a website provide recommendations that allow the website to meet the expectations of its customers and users.

Furthermore, this approach when compared to existing web analytics and other related tools, presents more readable and understandable reports for users and stakeholders, enabling a more intelligent analysis to suggest improvements to the website, as well as suggest new workflows and identify and remove requirements that are not used.
This approach also helps in the task of requirements management, which contributes to the quality of the web service itself, however additional work is needed to compare these results to those obtainable using other type of recommendations.

As future work, we intend to give support for other types of automated recommendations, e.g., suggest the creation of new features, change the location of some HTML elements based on the mapping information, generate test cases based on the requirements and use of user feedback to allow a direct analysis. We also aim to extend REQAnalytics in order to be possible to work with different web analytics tools since each of these tools can provide different kinds of data and metrics and because each website may use a different web analytics tool.

Finally, an interesting improvement to REQAnalytics would be to automatically apply the suggested recommendations to the software specification requirements.

References


Authors

Jorge Esparteiro Garcia is 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 (www.fe.up.pt/~apaiva) is 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 Working Groups of the ISTQB (www.istqb.org).