The Journal of Systems and Software 85 (2012) 2171–2181

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A systematic literature review of stakeholder identification methods in requirements elicitation

Carla Pacheco, Ivan Garcia*

Postgraduate Department, Technological University of the Mixtec Region, Mexico

ARTICLE INFO

Article history:
Received 2 August 2011
Received in revised form 27 April 2012
Accepted 30 April 2012
Available online 8 May 2012

Keywords:
Systematic review
Requirements engineering
Stakeholder identification
Requirements elicitation
Software engineering

ABSTRACT

This paper presents a systematic review of relevant published studies related to topics in Requirements Engineering, specifically, concerning stakeholder identification methods in requirements elicitation, dated from 1984 to 2011. Addressing four specific research questions, this systematic literature review shows the following evidence gathered from these studies: current status of stakeholder identification in software requirement elicitation, the best practices recommended for its performance, consequences of incorrect identification in requirements quality, and, aspects which need to be improved. Our findings suggest that the analyzed approaches still have serious limitations in terms of covering all aspects of stakeholder identification as an important part of requirements elicitation. However, through correctly identifying and understanding the stakeholders, it is possible to develop high quality software.

1. Introduction

The Requirements Engineering (RE) area is an essential part of any software development project that specifies, analyzes, and defines the product goal, functionality, and limitations of the final product (IEEE, 1998; Wiegers, 2003; Hofmann and Lehner, 2001). The fact that software requirements have a significant impact on final software product quality implies that it is reasonably well documented (Liscomb, 2003; Standish 2009; IEEE, 2004; SEI, 2006).

Usually RE can be described as a common series of stages including elicitation, analysis, specification, validation, and management (Pressman, 2005; Sommerville and Sawyer, 1997). In addition, three of the most important categories of problems affecting the correctness of software requirements are defined in the literature: gaining, comprehension and volatility (Loucopulos and Karakostas, 1995; Sommerville and Sawyer, 1997; Kotonya and Sommerville, 2000).

Our research is focused on the first of these stages, namely: requirements elicitation. Requirements elicitation is recognized as one of the most critical activities of software development (Davis et al., 2006); poor execution of elicitation will almost certainly guarantee that the final project is a complete failure. Since project failures are so uncontrolled (Standish, 2009), it is quite likely that improving how the industry performs elicitation would have a dramatic effect on the success record of the industry (Hofmann and Lehner, 2001). Improving requirements elicitation requires us to first understand the stakeholder identification phase (Nuseibeh and Easterbrook, 2000). In the case of requirements elicitation activities – in which the problem to be solved is identified – the most important thing is that the stakeholders be correctly identified (SEI, 1992). Relationships and ways of communicating between the development team and the customer are established at this time (ISO, 2004; Sommerville, 2002). Despite its importance, the identification of stakeholders, including the identification of their needs and expectations, is poorly achieved in software projects (Sommerville, 2002; Pressman, 2005). One probable cause is that this process is mistakenly viewed as a self-evident task in which direct users, clients and the development team are the only stakeholders. It could also be due to the fact that the identification area can be eliminated or substituted by opinions or knowledge obtained from other more accessible sources of information. In the short term, this would create less conflicts of interest resulting from different points of view (Smith, 2000).

Theoretical and empirical approaches are now being undertaken more often to investigate a widening range of phenomena in Software Engineering (SE) specifically in requirements elicitation as part of RE. As each approach is certainly limited in scope, researchers need to be able to rigorously and systematically locate, assess and aggregate outcomes from all significant empirical and theoretical studies related to a particular topic of interest, in order to provide an objective summary of the relevant evidence. This need has been addressed through the application process of a Systematic Literature Review (SLR).

* Corresponding author.
E-mail addresses: leninca@mixteco.utm.mx (C. Pacheco), ivan@mixteco.utm.mx (I. Garcia).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.04.075

In previous work, we performed an empirical study to identify how the stakeholder identification process can affect requirements quality and, as a consequence, the developed software quality. This study, presented in Pacheco and Tovar, proposed a categorization of stakeholder identification methods in requirements elicitation. Our work in this paper focused on three issues that had not been examined in earlier work. First, although stakeholder identification methods were identified we needed to determine and provide evidence about effective practices recommended to use them. Secondly, once those effective practices were exposed we needed to determine the consequences of incorrectly performing the stakeholder identification methods; and thirdly, with the collected data of previous issues, we summarized what aspects of stakeholder identification needed to be improved.

Consequently, in this paper, we present and discuss our experiences of applying the systematic literature review in order to gather and evaluate available evidence pertaining to Stakeholder Identification (SI) in requirements elicitation.

This paper is organized as follows: Section 2 presents other approaches related to SLR in requirements elicitation; Section 3 describes the method used for our SLR, reporting on the quality of the papers included in this section; Section 4 reports on the results of our synthesis of identified topics based on our four research questions; Section 5 presents some of the limitations of this study; in Section 6, we give suggestions for further research; and finally in Section 7, we present our conclusions.

2. Related work

There is still no substantial research related to SLR in requirements specification and elicitation techniques, and above all, in stakeholder identification, as seen below.

In 2006, Davis et al. (2006) reported a systematic review of empirical studies concerning the effectiveness of elicitation techniques, and the subsequent aggregation of empirical evidence gathered from those studies. These are the most significant results that were obtained: (1) interviews, preferentially structured, appear to be one of the most effective elicitation techniques; (2) many techniques often cited in the literature, like card sorting, ranking or thinking aloud, tend to be less effective than interviews; (3) analyst experience does not appear to be a relevant factor; and (4) the studies conducted have not found significant positive effects for the use of intermediate representations during elicitation.

Cheng and Atlee reviewed RE research and identified "future" research directions suggested by emerging software needs. This research examined technologies developed to address specific requirements tasks, such as elicitation, modeling, and analysis. Such a review enabled authors to identify mature areas of research, as well as areas that warrant further investigation. Next, they reviewed several strategies for performing and extending RE research results, to help delineate the scope of future research directions (Cheng and Atlee, 2007).

Davis et al. (2006) proposed recommendations based on the previous systematic review, which was updated and expanded with 13 new empirical studies and more than 60 new empirical results, to present some recommendations about the situations in which elicitation techniques are useful (Dieste et al., 2008).

Nicolás and Toval presented a systematic review of the literature related to the generation of textual requirements specifications from software engineering models. According to the results obtained, the benefits of both lists of textual requirements (usually written in natural language) and software engineering models (usually specified in graphical form) – can be brought together by combining the two approaches in the specification of system and software requirements documents (Nicolás and Toval, 2009).

Condori-Fernandez et al. described an empirical mapping study, which was designed to identify what aspects of software requirement specifications were empirically evaluated, in which context, and by which research method. On the basis of 46 identified and categorized primary studies, authors found that understandability was the most commonly evaluated aspect of SRS; experiments were the most commonly used research method, and that the academic environment was where most empirical evaluation takes place (Condori-Fernandez et al., 2009).

Dieste and Juristo presented the results of a systematic review of 564 empirical studies on elicitation techniques and aggregated these results to gather empirically grounded evidence. They selected and extracted data from 26 of those publications (containing 30 empirical studies) to provide a set of elicitation applicability guidelines based on the gathered pieces of knowledge. Their general finding is that interviews are the most effective of all of the tested elicitation techniques (although they are possibly less efficient in some domains than other techniques, like laddering or sorting techniques). Likewise, the authors do not recommend the use of introspective techniques (i.e., protocol analysis) because they fared worse than all of the other techniques in all of the tested dimensions (effectiveness, efficiency, and completeness) (Dieste and Juristo, 2011).

In summary, these studies only cover some aspects of requirements elicitation; however none of them analyze stakeholder identification evidence, despite this being a crucial part within requirements elicitation.

3. Research method

A SLR is "a means of evaluating and interpreting all available research relevant to a particular research question or topic area or phenomenon of interest" (Kitchenham, 2004). The research papers summarized in the review are referred to as primary studies, while the review itself is a secondary study. The accumulation of evidence through secondary studies can be very valuable in offering new insights or in identifying where an issue might be clarified by additional primary studies.

A SLR examines and interprets all available research relevant to a particular question or topic area. It aims to present an evaluation of the literature relative to researching a topic by using a rigorous and auditable methodology summary (Beecham et al., 2007).

So, due to the impact of SE and RE on software quality, as mentioned briefly in Section 1, we conducted a systematic review to see how SI is performed and how it can be improved.

We followed guidelines derived from those used by medical research, adapted and applied by Kitchenham (2004) and Kitchenham et al. (2004) to reflect the specific problems of SE and RE research (i.e. Beecham et al., 2007; Brereton et al., 2007).

In accordance to Kitchenham (2004) and Kitchenham et al. (2004), we took the following steps.

3.1. Identify the need for a systematic literature review

RE is a discipline that arose when it became evident that the quality of requirements specification was the key factor in preventing, with the least possible cost, many of the causes leading to software failure (Raghu, 1995). Thus, efforts in this direction employed at an early stage of a project have great repercussions, and are also more profitable than other efforts carried out afterwards. The problem of the "software crisis" has, therefore, to a great degree shifted to requirements. But, is there some aspect within the requirements area that deserves to be given particular treatment? Based on our previous line of thinking, this aspect should be connected to one of the initial activities of RE; that is,