Contents lists available at ScienceDirect

Information and Software Technology

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

Knowledge management initiatives in software testing: A mapping study

Érica Ferreira de Souza [a,*], Ricardo de Almeida Falbo [b], Nandamudi L. Vijaykumar [a]

[a] National Institute for Space Research (INPE), Av. dos Astronautas, 1758, 12227-010 São José dos Campos, SP, Brazil
[b] Federal University of Espírito Santo (UFES), Av. Fernando Ferrari, 514, 29075-910 Vitória, ES, Brazil

ARTICLE INFO

Article history:
Received 7 December 2013
Received in revised form 21 May 2014
Accepted 22 May 2014
Available online xxxx

Keywords:
Software testing
Knowledge management
Mapping study

ABSTRACT

Context: Software testing is a knowledge intensive process, and, thus, Knowledge Management (KM) principles and techniques should be applied to manage software testing knowledge.
Objective: This study conducts a survey on existing research on KM initiatives in software testing, in order to identify the state of the art in the area as well as the future research. Aspects such as purposes, types of knowledge, technologies and research type are investigated.
Method: The mapping study was performed by searching seven electronic databases. We considered studies published until December 2013. The initial resulting set was comprised of 562 studies. From this set, a total of 13 studies were selected. For these 13, we performed snowballing and direct search to publications of researchers and research groups that accomplished these studies.
Results: From the mapping study, we identified 15 studies addressing KM initiatives in software testing that have been reviewed in order to extract relevant information on a set of research questions.
Conclusions: Although only a few studies were found that addressed KM initiatives in software testing, the mapping shows an increasing interest in the topic in the recent years. Reuse of test cases is the perspective that has received more attention. From the KM point of view, most of the studies discuss aspects related to providing automated support for managing testing knowledge by means of a KM system. Moreover, as a main conclusion, the results show that KM is pointed out as an important strategy for increasing test effectiveness, as well as for improving the selection and application of suited techniques, methods and test cases. On the other hand, inadequacy of existing KM systems appears as the most cited problem related to applying KM in software testing.

1. Introduction

Software development is an error prone task. To achieve quality software products, it is essential to perform Verification & Validation (V&V) activities throughout the software development process. Verification and Validation (V&V) activities intend to ensure, respectively, that a software product is being built in conformance with its specification, and that it satisfies its intended use and the user needs [12]. V&V activities can be static or dynamic. Static V&V activities are typically done by means of technical reviews and inspections, and they do not require code execution. Dynamic V&V activities, in turn, involve code execution, and are done by means of testing [10,25]. Thus, Software Testing consists of dynamic V&V of the behavior of a program on a finite set of test cases, against the expected behavior [12].

Advances in technology and the emergence of increasingly complex and critical applications require using testing strategies, in order to achieve high quality and reliability software products [3]. Currently, software testing is considered as a process consisting of activities, techniques, resources and tools [25,26]. During software testing, a significant amount of information is generated. In fact, software testing is a knowledge intensive process, and thus it is important to provide computerized support for tasks of acquiring, processing, analyzing and disseminating testing knowledge for reuse [3]. In this context, testing knowledge should be captured and represented in an affordable and manageable way, and therefore, could make use of principles of knowledge management.

According to O'Leary [29], Knowledge Management (KM) formally manages knowledge resources in order to facilitate access and reuse, typically by using advanced Information Technology (IT), playing a major supporting role in KM. IT-supported KM

[*] Corresponding author. Tel.: +55 1232086549.
E-mail addresses: erica.souza@lac.inpe.br (É.F. de Souza), falbo@inf.ufes.br (R.d.A. Falbo), vijay.nl@inpe.br (N.L. Vijaykumar).

http://dx.doi.org/10.1016/j.infsof.2014.05.016
0950-5849/© 2014 Elsevier B.V. All rights reserved.

solutions are built around an organizational structure that integrates informal, semiformal, and formal knowledge to facilitate its access, sharing, and reuse [34]. The main goal of KM is to promote knowledge storage and sharing, as well as the emergence of new knowledge [30].

The Software Engineering community has recognized the need for managing knowledge and that it could learn much from the KM community [11]. Software development is a quickly changing, knowledge-intensive business, involving many people working in different phases and activities [24]. Knowledge in Software Engineering is diverse and organizations have problems in identifying its content, location, and use. An improved use of this knowledge is the basic motivation and driver for KM in Software Engineering. As a consequence, KM in Software Engineering has been subject of deeper analyses, such as those conducted by Rus and Lindvall [24], and by Bjørnson and Dingsøyr [11].

As a sub-area of Software Engineering, Software Testing also presents the same features. Knowledge can be applied to different testing tasks and purposes [15]. Knowledge on application domain and on testing techniques, as well as personal experiences, can be used to guide test case design, and to recognize failures. In an exploratory approach for software testing, where test cases are not defined in advance in an established test plan, but are dynamically designed, executed, and modified, tester knowledge is crucial. In this case, knowledge together with the observed actual behavior of the tested system can be used to create new, better tests during exploratory testing [15].

Given the great importance of knowledge for software testing, and the potential benefits of managing this knowledge, this paper aims to identify the state of the art on KM initiatives in Software Testing, by means of a mapping study. A mapping study provides a broad overview of a research area in order to determine whether there is research evidence on a particular topic. Results of a mapping study help identifying gaps in order to suggest future research and provide a direction to appropriately position new research activities [18,19,32].

The mapping study presented in this paper is an extension of an initial study we performed to identify publications discussing principles of KM applied to software testing, which can be referred to in [35]. In this initial study, we searched studies published until January 2013, and investigated the following aspects: study distribution over the years, purposes of employing KM in software testing, types of knowledge items typically managed in the context of software testing, supporting technologies used, and benefits and problems reported on implementing KM initiatives in software testing. In the extension presented in this paper, we include studies published until December 2013, and we change the search string to incorporate other terms that are also related to KM, leading to new relevant studies. Moreover, new studies were selected by means of snowballing the primary study references, as well as by directly searching publications from researchers and research groups of the studies previously selected. Snowballing is a process that checks if the selected studies cite other relevant studies, retrieve those studies, and continue this process until no more relevant studies are found [13]. Finally, we also enlarge the scope of our investigation, by considering other aspects not previously addressed in the initial study, namely: source of the publication, research focus from the testing perspective, research focus from the KM perspective, and type of the research performed.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of KM and software testing, as well as related research. Section 3 discusses the research method applied to perform the mapping study. Results are presented in Section 4. Section 5 discusses the results, their implications, and limitations. Finally, Section 6 concludes the paper and presents directions for future work.

2. Background

In this section, we briefly present the main concepts related to the topics addressed in this paper, namely: Knowledge Management and Software Testing. Moreover, we briefly discuss related research, i.e. secondary studies that are related to these topics.

2.1. Knowledge management (KM)

Knowledge is one of the most valuable assets for most organizations. There are two main types of knowledge [27]: tacit and explicit. Nonaka and Takeuchi [27], use the tacit-explicit distinction to differentiate unarticulated and articulated stocks of knowledge [37]. Tacit knowledge is the subjective and experience-based knowledge that cannot be documented, and typically remains only in people's minds. This type of knowledge depends on personal experience and involves intangible factors such as beliefs, perspectives, values and intuition [27]. Tacit knowledge covers knowledge that is unarticulated and associated to the senses, movement skills, physical experiences, intuition, or implicit rules of thumb. Even if we try hard, this type of knowledge cannot be fully articulated [28]. Explicit knowledge, in turn, represents the objective and rational knowledge that can be documented, and, thus can be accessed by multiple individuals [27]. Explicit knowledge can be uttered and captured in drawings and writing, and can be easily used and shared. The concept of "knowledge conversion" explains how tacit and explicit knowledge interact along a continuum [28].

In the context of software testing, KM can be used to capture knowledge and experience generated during the testing process. However, usually, this knowledge has been stored on paper or in people's minds. When a problem arises, we look for experts across our working environment, relying on people we know, or we look for documents. Unfortunately, paper has limited accessibility and it is difficult to update [29]. On the other hand, in a large organization, it can be difficult to locate who knows a certain subject, and knowledge in people's minds (tacit knowledge) is lost when individuals leave the organization. Therefore, testing knowledge has to be systematically collected, stored in an organizational repository, and shared across the organization. In other words, KM is, indeed, necessary.

KM can be viewed as the development and leveraging of organizational knowledge to increase organization's value [40]. Organizational knowledge creation aims at making available and amplifying knowledge created by individuals as well as crystallizing and connecting it to an organization's knowledge system [28]. KM entails formally managing knowledge resources in order to facilitate access and reuse of knowledge, typically by using advanced information technology [29].

Information technology plays a major supporting role in KM [34]. A wide range of technologies have been used in the development of KM systems, such as databases, data mining, intranets and internet, intelligent information retrieval, intelligent agents, case-based reasoning, yellow pages, ontologies, visualization models, and groupware [21,29].

2.2. Software testing

Software Testing consists of dynamic verification and validation of the behavior of a program on a finite set of test cases, against the expected behavior. Software Testing should be supported by a well defined and controlled testing process [7]. Testing process consists of several activities, typically including [2,6,25,26]: Test Planning, Test Case Design, Test Implementation, Test Execution and Test Result Analysis. First, testing should be planned. Key aspects of test planning include, among others, defining the test environment for