The Journal of Systems and Software 84 (2011) 620–637

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners

Claudia Ayala [a,*], Øyvind Hauge [b], Reidar Conradi [b], Xavier Franch [a], Jingyue Li [c]

[a] Technical University of Catalunya, UPC Campus Nord-Omega, Barcelona 08034, Spain
[b] Norwegian University of Science and Technology. NTNU Campus IDI-Gløshaugen, Trondheim 7491, Norway
[c] DNV Research & Innovation, Veritasveien 1, 1363 Høvik, Norway

ARTICLE INFO

Article history:
Received 27 May 2010
Received in revised form 1 October 2010
Accepted 14 October 2010
Available online 18 November 2010

Keywords:
Software engineering
Software reuse
Empirical study
Off-The-Shelf-based software development
Component selection

ABSTRACT

The success of software development using third party components highly depends on the ability to select a suitable component for the intended application. The evidence shows that there is limited knowledge about current industrial OTS selection practices. As a result, there is often a gap between theory and practice, and the proposed methods for supporting selection are rarely adopted in the industrial practice. This paper's goal is to investigate the actual industrial practice of component selection in order to provide an initial empirical basis that allows the reconciliation of research and industrial endeavors. The study consisted of semi-structured interviews with 23 employees from 20 different software-intensive companies that mostly develop web information system applications. It provides qualitative information that help to further understand these practices, and emphasize some aspects that have been overlooked by researchers. For instance, although the literature claims that component repositories are important for locating reusable components; these are hardly used in industrial practice. Instead, other resources that have not received considerable attention are used with this aim. Practices and potential market niches for software-intensive companies have been also identified. The results are valuable from both the research and the industrial perspectives as they provide a basis for formulating well-substantiated hypotheses and more effective improvement strategies.

1. Introduction

Nowadays, the approach of building software systems by reusing third party software as Off-The-Shelf (OTS) components has been recognized as a crucial success factor for the software industry. We refer to the definition of an OTS component as stated by Torchiano and Morisio (2004) that defines it as: *"a commercially available or open source piece of software that other software projects can reuse and integrate into their own products"*. This definition includes components/services acquired by a fee (known as Commercial-Off-The-Shelf software; COTS) or from Open Source communities (known as Open Source Software; OSS). OTS-Based Software Development (OBSD) allows companies to achieve better quality, and faster technology adoption and innovation, while reducing development costs and time-to-market (Jansen et al., 2008; NCube et al., 2008). The potential advantages of this technology have led to an increasing availability of OTS components in a wide variety of application areas. Hence, it has been claimed that "*It is becoming not only impractical, but also virtually impossible for mainstream IT organizations to ignore the growing presence of third party software in major segments of the IT industry. The failure to optimally manage the potential risks and rewards of using this software will put IT organizations at an increasingly serious risk in coming years*" (Gartner, 2008).

The success of OBSD greatly depends on the ability of the integrators to select the most suitable component(s) to be integrated (Boeg, 2006). However, although there has been a great body of research on component selection, the evidence shows that there is a limited knowledge about current industrial OTS selection practices. As a result, there is often a gap between theory and practice, and the proposed methods are hardly used in the industrial practice (Torchiano and Morisio, 2004; Li et al., 2009; Jadhav and Sonar, 2009). As a consequence, software companies are still facing OTS component selection under considerable risk and uncertainty (Boeg, 2006; Jansen et al., 2008; Birkmeier and Overhage, 2009). Researchers from the Empirical Software Engineering (ESE) community have claimed that to mitigate this lack of industrial uptake, researchers must become aware and more precise about their proposed approaches' assumptions, contexts and limitations (Glass, 2004; Kitchenham et al., 2004; Basili and Elbaum, 2006; Erdogmus,

* Corresponding author. Tel.: +34 934137890; fax: +34 934137833.
E-mail addresses: cayala@essi.upc.edu (C. Ayala), hauge@idi.ntnu.no (Ø. Hauge), conradi@idi.ntnu.no (R. Conradi), franch@essi.upc.edu (X. Franch), jingyue.li@dnv.com (J. Li).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.10.019

2010). It is therefore clear that to improve OTS component selection practices; the research community must understand what the actual industrial OTS selection practices are in order to envisage more realistic and effective solutions (Glass, 2004).

In this context, focusing on the perspective of the component integrators (i.e., the person(s) in charge of selecting and integrating components), we performed a qualitative survey based on semi-structured interviews with 23 component integrators from 20 software-intensive organizations in Spain, Norway and Luxembourg. The main goal of this study is exploring and describing up-to-date industrial OTS selection practices, as an initial step towards the alignment of research endeavors with real industrial needs. We think that results from this work may help maturing the OTS component marketplace, as researchers and practitioners may use the evidence provided by this paper to understand the practical challenges of OTS component selection, and properly align their efforts for facing them. In particular, researchers may use the evidence presented in this paper to identify and align new research questions, generate and test hypotheses, and interpret the results of such tests. Likewise, practitioners and diverse actors related to the OTS component marketplace (e.g., component providers, components intermediaries, and providers of services around components) may use this paper to identify and understand other OTS selection practices and to envisage strategic actions for improvement.

The remainder of this paper is organized as follows: Section 2 provides a brief background to OTS component selection and an overview of the empirical evidence available. In addition, we present the objectives of this study. Section 3 discusses the methodological approach followed to perform the study and introduces the details of organizations, individuals and projects approached by the study. Section 4 presents the results obtained from the study, while Section 5 provides an in-depth discussion of findings. Threats to validity are presented in Section 6. Section 7 summarizes the conclusions and future work.

2. Background

Systematic software reuse is an engineering strategy proposed to increase productivity and software quality, and lead to economic benefit (Morisio, 2006). Although software reuse has been an active research arena for more than a decade, the special nature of OTS components has motivated particular research lines addressing reuse of OTS components (Morisio, 2006). In this section we give a brief background on OTS component selection, and summarize the body of evidence that exists in the area based on published surveys.

2.1. State-of-the-art component selection

OTS component selection is widely recognized as an interrelated process that plays a central role in overall OBSD (Morisio et al., 2002; Mahmood et al., 2007). Roughly speaking, component selection can be viewed as consisting of three activities that are usually staged (Finkelstein et al., 1996; Mohamed et al., 2007; Land et al., 2008):

(a) *Identification of candidate components.* It is aimed to locate one or more candidate components that may cover the system requirements (while avoiding non-relevant components) and to acquire information that makes their evaluation and comparison feasible.
(b) *Evaluating components with respect to the expected requirements.* This activity's aim is to assess to what extent the candidate component(s) covers/cover the system requirements.
(c) *Choosing suitable component alternative(s).* This refers to the comparison of the candidate components to choose the one(s) that best fits/fit the stated requirements.

The Internet is a vital part of component selection (Clark et al., 2004; Wanyama and Far, 2006; Mahmood et al., 2007; Umarji et al., 2008) as it constitutes the virtual place where components are mainly searched for and provided. It is called the OTS marketplace, and also includes the exchange interactions between integrators (i.e., component reusers) and component providers, as well as the actions of other actors that facilitate or promote such transactions, e.g., intermediaries, and marketing channels (Ayala et al., 2009). The OTS marketplace is characterized by the uncontrolled growth of component offerings and demands, new versions of existing components, and the lack of standards describing these components. It has been recognized that the existence of the OTS marketplace has introduced new and profound challenges to the software reuse arena (Morisio, 2006). To respond to these challenges, an extensive body of research has been put forward.

2.1.1. Identification of OTS components

Searching for reusable components was traditionally supported by centralized component repository systems with specific classification and searching mechanisms (Frakes and Kang, 2005). However, the special nature of the OTS marketplace has shifted this focus to a global reuse approach (Morisio, 2006). Main efforts to support OTS component searching have been devoted to classification structures and specialized search engines (see Birkmeier and Overhage, 2009; for a survey). On the one hand, several works have been proposed to categorize OTS components' attributes. On the other hand, automatic or semi-automatic search engines using different technologies have been proposed for finding and identifying OTS-related hits, relying on some available component catalogues. Representative examples are: Google's specialized free code search (GoogleCodeSearch) addressed to find open source code on the Internet, and academic tools such as Agora (Seacord et al., 1998), IPSCom -Intelligent Portal for Searching Components (Aguirre, 2005), or MoReCOTS (Yanes et al., 2006). In addition, the use of global ontologies (Simmons and Dillon, 2006; Cechich et al., 2006) or the Semantic Web (Ankolekar et al., 2003) has also been proposed to deal with the lack of homogeneous descriptions of components. However, none of these mechanisms and tools have been feasibly implemented or adopted in industrial practice (Cechich et al., 2006; Birkmeier and Overhage, 2009). Furthermore, component searching has been stated as a complex and immature arena that actually requires different common efforts from very diverse areas such as software reuse, code search, information retrieval, and program comprehension (Wang et al., 2005; Gallardo-Valencia and Sim, 2009; Birkmeier and Overhage, 2009).

2.1.2. Evaluating and choosing OTS components

In recent years there has been a plethora of proposals aimed to support component evaluation and decision making. These proposals range from suggesting sets of evaluation criteria and changes to the software development processes, to proposing novel technologies emerging from other areas such as decision support systems, method engineering, strategic contracting and procurement, simulation and formal reasoning. Early proposals mainly focused on COTS components, but in the last years the potential benefits of OSS are gaining considerable attention. Several proposals and large scale research projects focus on OSS selection particularities. Some of the first examples are the Open Source Maturity Model (OSMM; Golden, 2004), Open Business Readiness Rating (OpenBRR; Openbrr, 2005), and the Qualification and Selection of Open Source software (QSOS; Semeteys et al., 2006). Besides suggesting a number of new evaluation criteria that reflect the components'