The Journal of Systems and Software xxx (2010) xxx–xxx

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 xxx

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'

Table 1
Summary of existing evidence on OTS component selection.

Paper	Research Agenda	Findings related to OTS selection
Torchiano and Morisio (2004)	An interview study of OTS usage in IT companies in 2002	EV1. Developers seldom use formal selection procedures. EV2. Architecture is more important than requirements for product selection.[a]
Li et al. (2008, 2009)	Series of empirical studies focused on process improvement and risk management in OBSD (from 2003 to 2005)	EV3. Integrators select OTS components informally. They rarely use formal selection procedures. EV4. There is no specific phase of the development process in which integrators choose OTS components. EV5. Involving clients in OTS component decisions is rare and sometimes unfeasible.
Keil and Tiwana (2005)	A web-based survey on attributes of enterprise COTS that organizations value most in 2005	EV6. Prioritization of some attributes, according to their importance, by practitioners using COTS components: Functionality, Reliability, Cost, Ease of customization, and Ease of use.
Chen et al. (2008)	A web-based survey on software development practices using OSS in the Chinese software industry in 2007	EV7. Searching OSS was mainly based on Google rather than other portals such as SourceForge. EV8. No formal methods were used to select. EV9. Chinese integrators ranked requirements compliance as the most important criteria to compare OSS components, while licensing price and technical support from the OSS community were regarded as the least important criteria to evaluate OSS. EV10. Familiarity was mainly used for evaluating and deciding components.
Umarji et al. (2008)	A web-based survey aimed to collect the ways in which integrators search for components on the Internet in order to categorize their motivations, in 2008.	EV11. Integrators mainly browse the Internet to find components and reference examples that help them to understand component integration details.
Land et al. (2009)	A web-based survey aimed to gather information about how software reuse is performed in practice	EV12. Integrators evaluate components insufficiently and use test cases and prototyping for evaluation.

Please note that EV1, EV3 and EV8 refer to the same finding.

[a] Contradicted by Li et al. (2009) and Chen et al. (2008).

OSS nature, they share the same fundamental selection principles as those for COTS. Such evaluation criteria are further explored by, for instance (Cruz et al., 2006), the QualOSS Model Framework (Ciokolwski and Soto, 2008), and the QualiPSo model of OSS trustworthiness (Del Bianco et al., 2009). Comprehensive surveys may be found in (Merilinna and Matinlassi, 2006; Mohamed et al., 2007; Mahmood et al., 2007; Land et al., 2008; Jadhav and Sonar, 2009). However, regardless of the kind of components these proposals mainly address (COTS or OSS), they mostly focus on the evaluation criteria and decision-making phases, setting aside the practical problem of how to search for and locate components and suitable information about them (Land et al., 2009; Jadhav and Sonar, 2009). As a result, there is no consensus on the applicability of these proposals in industrial practice.

2.2. Body of evidence

A usual problem of the Software Engineering (SE) discipline comes from the lack of empirical evidence to support research hypotheses and the subsequent evaluation of proposed solutions (Kitchenham et al., 2004; Basili and Elbaum, 2006). The OBSD area is not an exception, as only a slight extent of the literature refers to empirical studies. This subsection summarizes representative studies that offer some kind of evidence about OTS component selection practices.

Most of the literature related to OBSD refer to academic proposals supported by proof of concepts or toy cases just to demonstrate the feasibility of the proposals (see, the systematic review performed by Jadhav and Sonar (2009)). The limited number of empirical studies usually refers to single case studies or experience reports that provide limited information about OTS selection practices (e.g., Jensen, 2003; Anderson, 2004; Carvallo et al., 2007). Besides, these reports are from contexts that are scarcely described and are hardly to be generalized to commercial sectors; e.g., avionics and departments of defense (Majchrowski and Deprez, 2008). For example, the study presented by Jensen (2003) discusses some lessons learned using post-mortem analysis of avionics software development. Regarding component selection, his conclusion was limited to emphasize that it was a key and critical process, requiring proper preparation and diligence in order to achieve a successful software project. While these papers are valuable for understanding the importance of the OTS component selection activities, they are not particularly relevant to the specific problems practitioners face every day, and provide little concrete advice related to OTS components' selection.

Only a few studies have conducted more large-scale field studies representing industrial sectors (e.g., Land et al., 2009; Li et al., 2009; Chen et al., 2008; Keil and Tiwana, 2005; Torchiano and Morisio, 2004). However, all of these studies have had a broader scope than component selection, and their findings related to component selection are quite limited. Below there is a brief overview of these studies, and Table 1 summarizes the actual evidence related to OTS components' selection supported by these available studies.

Torchiano and Morisio (2004) performed a qualitative study on OTS usage in 7 IT companies in 2002. The study identified six "theses" on OTS component usage. Two of them involved the selection stage, and are stated as EV1 and EV2 in Table 1. From 2003 to 2005, Li et al. (2009) performed a series of empirical studies aimed to test and clarify the theses stated by Torchiano and Morisio (2004). Li et al. (2009) focused on process improvement and risk management issues, and performed a qualitative pre-study followed by a larger quantitative study in 127 companies and a follow-up study (Li et al., 2008). Results from these studies contradicted Torchiano and Morisio's thesis T2, and offered 10 facts characterizing the state of the practice of OBSD. Three of these facts were directly related to component selection, and are stated as EV3, EV4 and EV5 in Table 1. Keil and Tiwana (2005) used a set of evaluation criteria commonly mentioned in the literature as relevant for selecting COTS components, and asked information systems managers how they evaluate these attributes of COTS software. It resulted in the following prioritization of the attributes: Functionality, Reliability, Cost, Ease of customization, and Ease of use, which together are listed as EV6 in Table 1. Although valuable, this result left many questions unanswered (e.g., Why were these criteria important? In which context were they important?).

In addition, from August 2005 to November 2006, Chen et al. (2008) performed a study to investigate the major challenges facing the Chinese software industry using OSS components. Regarding component selection, they concluded that four factors were important, which are denoted as EV7, EV8, EV9 and EV10 in Table 1. Their results also allowed the conclusion to be drawn that requirement compliance was more important than architecture compliance. It is relevant to mention that this study focused only on OSS components, and was moreover restricted to the Chinese industry. Keeping these limitations in mind, the same authors therefore declared that there may be significant variations with respect to the Occidental industry.

In 2008, with the aim of eliciting a set of scenarios that may inform the design of new features for component searching tools, Umarji et al. (2008) conducted a web-based survey for collecting the ways in which integrators searched for components in order to categorize the integrators' motivations. Although the reported results are still very preliminary, they stated one motivation, which is denoted as EV11 in Table 1. Finally, Land et al. (2009) carried out a web-based survey to gather information about how software reuse is performed in practice. Given the general nature of this survey, the findings regarding component selection were limited to some observations, which are denoted as EV12 in Table 1.

In general, it can be observed that what we really know about the industrial practice of OTS component selection is quite limited. Furthermore, academic researchers often hold false assumptions about how components are used in the industry (Li et al., 2009). For instance, most of the component selection proposals related in Section 2.1.2 assume an "ideal" situation where the components are suitably arranged, documented and reside in a common place. However, this is far from reality (Ulkuniemi and Seppänen, 2004; Boeg, 2006). Therefore, the adoption of research proposals by industry is hardly motivated, as industry does not see its practices identified in the literature and there is, in addition, a lack of information about the effectiveness of these proposals in real environments (Land et al., 2008).

The maturation, acceptance, and adoption of good SE ideas depend on many factors; one of these factors is the availability of evidence (Erdogmus, 2010). The aim of the Evidence-Based Software Engineering (EBSE) paradigm (Kitchenham et al., 2004) is to encompass these factors. Thus, based on this paradigm, we claim that it is vital to investigate the actual OTS component selection industrial practices in order to foster the necessary alignment and synergy between research and industry to envisage effective solutions.

2.3. Objectives of this study

In line with the general observation that the lack of industrial uptake of some methods proposed in the SE discipline comes from the lack of empirical evidence to support research hypotheses and the subsequent evaluation of proposed solutions (Seaman, 1999; Kitchenham et al., 2004; Basili and Elbaum, 2006), our overall objective is to provide a solid step forward in understanding the current industrial OTS selection practices. This objective was broken down into three more concrete research questions, related to the three OTS selection activities mentioned above:

RQ1. How do integrators identify OTS components?
RQ2. How do integrators evaluate OTS components?
RQ3. How do integrators choose OTS components?

The research questions mostly focus on understanding: (1) the processes and (2) the resources used by integrators to perform identification, evaluation and choosing (RQ1, RQ2 and RQ3 respectively). RQ2 also considers the kind of factors or evaluation criteria mainly used to evaluate and choose components, and RQ3 considers the stages of the software development process where the selection activities are performed. Even though there is some evidence related to OTS selection (as shown in the previous section), it was quite limited and dated from some years ago. Therefore, our intention was not only to increase our knowledge and understanding about industrial OTS selection practices, but also to provide up-to-date information to situate the former evidence.

3. Research method

As the nature of our inquiry was clearly exploratory, we decided to carry out the study using a qualitative research approach based on semi-structured interviews to collect data directly from industrial practitioners (Robson, 2002). Qualitative studies have been claimed to be necessary to complement quantitative ones, given that qualitative knowledge is an essential prerequisite for the generation and testing of hypotheses and for interpreting the results of such tests (Seaman, 1999). The interview guide was carefully designed following the guidelines stated in (Oates, 2006) and previous experience performing international surveys (Conradi et al., 2005). In general, the guide mostly focused on a single finished project that the respondents were familiar with, and on a single component used in that project. Considering a single project instead of many projects allowed us a better interpretation and assessment of contextual information. It would otherwise have been very difficult to interpret certain decisions or influential factors related to the nature of the projects. For the same reason, we also decided to focus several of the questions on a single component used in that project. The project and the component(s) were chosen by the interviewee without any intervention from us. In addition to this particularization of the inquiries, we also added follow-up questions (such as: *Is this typically how you do this? If not, how do you usually do it?*) in order to identify and understand potential representative practices, as suggested by (Lutters and Seaman, 2007; Patton, 2002). It allowed a richer vision of the processes, resources and criteria used to select components in the whole of each organization. The interview guide used in the study may be consulted in Appendix A.

3.1. Sampling

The target population was practitioners in charge of performing component selection activities. Participating organizations were chosen from our direct or indirect industrial collaboration network. We aimed for a maximum variation approach, so that organizations covered as many different characteristics as possible with respect to size, application domain, and business area. The only requirement for companies to participate was that they had undergone a finished project that implied component selection. We received 23 respondents in total from Spain, Norway and Luxembourg. Interviews were mainly performed in the mother tongue of the respondents and face-to-face in their working place, by one or two researchers of the team. Interviews lasted around 1 h each and were recorded for subsequent analysis.

Participating organizations. 20 organizations participated in the study. In 3 of these organizations we could interview more than 1 integrator. Table 2 provides an overview of these organizations and relates their corresponding respondents. The organizations covered a varied spectrum regarding kind, ownership, business areas, and size of the local staff. 9 organizations were standalone (i.e., financially independent from another parent organization), and 11 were subsidiary (i.e., independent units but mainly

Table 2
Overview of organizations participating in the study.

Id organization	Id respondent	Kind	Ownership	Main business area	Staff
A	**A**	Standalone	Private	SCC	≈20
B	**B**	Subsidiary	Private	SCC	≈200
C	**C**	Standalone	Private	SCC	11
D	**D**	Standalone	Private	SCC	63
E	**E**	Standalone	Private	SCC	21
F	**F**	Standalone	Private	ITD	10
G	**G**	Standalone	Private	SCC	≈200
H	**H**	Standalone	Private	SCC	≈4
I	**I**	Subsidiary	Public	ITD	15
J	**J**	Standalone	Private	SCC (mainly Business Intelligence)	≈200
K	**K1, K2**	Subsidiary	Public	ITD	[a]
L	**L**	Subsidiary	Private	SH	[a]
M	**M**	Subsidiary	Private	SCC	≈150
N	**N**	Subsidiary	Private	SCC	80
O	**O1, O2**	Subsidiary	Public/Private	SCC	160
P	**P**	Standalone	Private	SCC (mainly Geographic Information Systems)	40
Q	**Q1, Q2**	Subsidiary	Private	SCC	900
R	**R**	Subsidiary	Public/Private	SCC	60
S	**S**	Subsidiary	Private	SH	35,000
T	**T**	Subsidiary	Public/Private	ESSS (Business Applications)	4

SCC, Software Consultancy; ESSS. Expert Support for Selecting Software Solutions; SH, Software House; ITD, IT Department.

[a] Staff was geographically distributed so the respondent did not know the size.

owned/financed by a parent organization). 15 were from the private sector, 2 from the public sector, and 3 were held by both public and private capital. 14 of the companies were software consultancy companies (SCC) that perform software development tasks for different clients as their primary business; 3 were IT departments (ITD) in public or tertiary organizations that usually perform or outsource some software development tasks for covering the internal demands of the organization; 2 companies were software houses (SH) that develop and commercialize specific proprietary solutions. 1 organization provides expert support for selecting software (ESSS) solutions based on their clients' requirements. However, this organization does not perform any software development tasks. In addition, 3 organizations explicitly stated that their business area was oriented towards a specific domain.

Respondents. The respondents had different positions in the organizations, and actively participated in (or were in charge of) the component selection tasks in at least the project they based their answers on. Most respondents had an education background related to computer science (only one did not have an academic background related to computer science, but rather biology). 14 of the respondents had a bachelor's degree, 8 had a master's degree and 1 had a Ph.D. degree. The respondents had between 2 and 35 years of experience in software development.

Projects and components. As explained above, each interviewee was asked to talk about a single finished project. The resulting set of projects was very diverse, and used a variety of components that ranged from libraries and APIs to more complex solutions. In addition, some questions from the interview guide also required the interviewee to refer to a specific component used in the project.

Table 3 summarizes the projects and the total effort (in person/hour) spent on each project. In addition it gives some examples of the components used, and specifies the components that were chosen by the respondents to base their answers on. In some cases, the specific name of the chosen component is not revealed as this was explicitly required by some respondents.

All projects in Table 3 involved software development tasks, except for the one approached by ***T***. This project exclusively focused on selecting components on behalf of one of the company's clients. As a result, the project's output was a document assessing and ranking candidate components that covered the client requirements.

3.2. Data analysis

Interviews were prepared for analysis by the manual transcription of audio records to text documents, and were translated to English so that the whole research team could assess and discuss the data. We used content analysis (Krippendorff, 1980) as a basis for performing the assessment of the collected data, and generating categories by grouping sentences of phrases that described the same idea, action or property. We tried to be exhaustive with the categories, in order to include as much detail provided by the respondents as possible. The assessment of categories was enriched by the information obtained from further questions (such as, "Is this typically how you do this?") that helped to identify and understand practices not used in the particular projects approached, but which might be representative in the organizations. In this way, we got a broader understanding of the OTS selection practices in each organization. We also generated frequencies of codes as an indicator of popular and unpopular practices. In addition, associations of codes were useful to further understand the data and reveal aspects of potential importance. Appendix B presents the association of codes corresponding to the data reported here. Appendix B and the narrative way we report the results attempt to provide detailed evidence that may promote further understanding in the reader when confronted by other situations not considered by the research question reported in this paper. Preliminary results from the first 8 interviews were presented in (Ayala et al., 2009), while the paper presented here provides complete and detailed results from all interviews.

4. Results

This section presents the results of the study. They are grouped in 3 subsections according to the research questions introduced above. Results are described in terms of the categories or codes generated from the data analysis. Tables are used to show the category codes, the description of each category, some representative quotes, and the frequency of answers belonging to each category. By the qualitative nature of our study, the tables are complemented with narrative descriptions of the results and discussions of the result's representativeness based on the answers given to further questions such as "Is this representative of how you perform this?". Appendix B also provides further information about categories and

Table 3
Overview of projects and OTS components approached by the respondents.

Respondent	OTS-based system	Total effort	Some COTS used	Some OSS used	Component approached
A	Web application for managing a student database	480	Oracle	Apache libraries, and Spring framework	Spring web service project
B	Web application for trade-off foundations for an Internet bank	17,520	Websphere application server	Spring framework, and miscellaneous common projects	Spring framework
C	Windows application for assisting dyslectic people in typing/writing	29,200		LGPL libraries for OpenOffice	Libraries for Open Office
D	A content management system covering the requirements of most of their clients	21,900		Miscellaneous such as Lucene, Jlog, Spring, Hibernate, readers for text, pdf, Word	Java Script components
E	A system that records work hours, scheduling and appointments	14,600		Posgress, Hibernate, Spring, Java Server Faces (JSF), RichPhase, Jasper reports, Facelet, Tomhawk	RichPhase
F	Adding statistics and graphical features to an existing system	60		Google Charts API	GoogleCharts
G	Corporative website for a company that sells items on the Internet	640		Mambo, OSCommerce, and miscellaneous add-ons	Mambo
H	Web tool for personal data management	1920		Miscellaneous from the AJAX environment.	Hibernate
I	Development and maintenance of a resource management system	5840		Miscellaneous Php, MySQL, and Apache components	MySQL
J	Business intelligence system for the management of marketing campaigns	[a]	Business Objects, SQL Server 2005, .Net components		SQL Server 2005
K1	Framework for J2EE development	[a]		J2EE-based components (e.g., Spring, and Hibernate)	J2EE-based components
K2	System for processing and managing emergency calls	47,520	.Net components and Oracle		Oracle
L	Updating of a record management system	12,000	IBM products such as Content Manager, Petition Server	Some miscellaneous components such as Formatting Objects Processor (FOP)	IBM Content Manager
M	A collaborative virtual campus for a business school	5832	SharePoint, Windows Server, SQL server		SharePoint
N	Web application for management of incidences	[a]		Miscellaneous Debian, MySQL 5, and Apache 2 components	Debian
O1	Migration of a proprietary content management system to an OSS-based solution	2280		Plone	Plone
O2	Migration from LotusNotes to an OSS-based solution	11,520		OpenCoreBusiness	OpenCoreBusiness
P	Web queries for visualizing geographical information	6000	OracleEnterprise solutions for GIS	Miscellaneous components from the J2EE framework	J2EE-based components
Q1	A web geographic information system to identify the availability of telecommunication services by area	[a]	ArgiServer	Miscellaneous components from Googlemaps, Google earth and Hibernate	GoogleMaps
Q2	A content management system with diverse features	1200	-	Miscellaneous Java components	Java components
R	A web project for people collaboration	33,620		Miscellaneous MySQL, Java, Php or Perl components	Java components
S	Data warehouse commercial system	[a]	Commodities	[a]	[a]
T	Selection of a suitable solution for the management, sales, accounting and logistics of a chain of shops at the airport	[a]	Some commercial solutions were evaluated	Some OSS solutions were evaluated	A proprietary solution finally chosen by the client

[a] Respondent did not know or asked to keep this information confidential.

their relationships. Interpretation and discussion of the findings according to the research question are tackled in Section 5.

4.1. RQ1. Industrial OTS component identification practices

This section summarizes the results related to the identification of OTS components. Data regarding the process used by industry to identify components is presented in Section 4.1.1, while further details about the resources used to deal with such identification are presented in Section 4.1.2.

4.1.1. Processes for identifying OTS components

The answers to the question about how the chosen component was identified were categorized as shown in Table 4. These categories are mutually exclusive to denote the specific way in which each of the projects dealt with the component searching task.

In the searching phase, no company used any established procedure or guidelines to drive component identification. Instead, it was previous experience and previous awareness of the components (SerA) that were the most influential factors determining the way integrators identified components. In addition, most of the responses that fall into this category coincided on their answer to the question “Is this representative of how you identify components?”, as they stated that this was a usual practice when they were familiar with the domain: “*As we are experts in this domain, we should know the components and the providers that are available in the market. If those that we know do not cover what the client wants, we have to search for it [the suitable component], but we usually know where to search*” (**J**).

Please cite this article in press as: Ayala, C., et al., Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. J. Syst. Software (2010), doi:10.1016/j.jss.2010.10.019

Table 4
Categories of component identification.

Code	Category description	Representative quotes	Freq.
SerA	**Direct component identification**. Respondents had used or heard about the component before, and were able to find the component directly.	*"In this project, we knew* [sic] *components that exist in this domain. We are very familiar with this domain."* (**J**)	12
SerB	**Internet browsing**. Respondents were not familiar with any candidate component and used Internet searches and Internet browsing to find a component.	*"We did not have so much experience with this domain, so we basically browsed the Internet."* (**C**)	6
SerC	**Hire an expert company**. Respondents were not familiar with the domain and hired an expert company to perform the task	*"We contracted an expert company to perform the elicitation of requirements and the identification of candidate components to cover them. But we closely participated with them throughout the process."* (**O2**) *"We contracted a consultant company to do the task. They were the ones that did it; we did not participate on this."* (**K2**)	3[a]
SerD	**No search**. Respondents stated that the component to be used was decided in advance by the boss or by the client. So, no search was performed.	*"The company decided what we were going to use before we came into the project. So, I was not part of any identification in this project."* (**B**)	2
			23

[a] Two of the answers are from the same company.

6 of the respondents said that they identified their component(s) using keywords to browse the Internet (SerB). When we inquired, "Is this representative of how you identify components?", most of the respondents emphasized that it might be considered representative only in cases when they have neither previous experience with the related domain nor internal experience in the company: *"Yes, we typically browse the Internet using keywords when* [*we*] *do not have* [*an*] *idea on where to find the components"* (**G**).

Other organizations hired an expert company for doing the tasks related to identifying components (SerC). We distinguished between two different kinds of involvement: (1) organizations that have a close participation with the hired company to perform the searching tasks and (2) organizations that totally delegate the task to the hired company. In the first case, the respondents were able to provide details of the process, while in the second case the respondents did not know how the searching tasks were performed. When we inquired further, "Is this representative of how you identify components?", respondents emphasized that it was mostly critical projects that received this treatment: *"No, no, we do not do it for all projects of the company, only the ones that are critical for the organization"* (**O1**). It is important to remark that two of the answers that fall in this category were from the same company.

Finally, 2 respondents recognized that no search tasks were performed in the project as the component was decided in advance by the client or by the boss (SerD). Regarding whether this was a representative practice in the company or not, respondents said that it depended on the client requirements and/or strategic relationship with component providers. *"It is generally our company or the client that has an agreement with a provider and then you must use what . . . was already decided"* (**R**).

4.1.2. Resources used to search for components

The answers categorized here include not only details related to the specific projects addressed by the respondents, but also the general resources used in their usual selection practices. Categories used here are not mutually exclusive as interviewers mentioned more than one resource used to perform this task. Table 5 shows the categories related to the resources used to identify components.

The results show that sharing experiences has the most significant relevance for identifying components (Res-A). Most respondents highlighted the influence of informal information and channels (such as asking opinions about the component from colleagues, attending conferences, and professional networks).

In general, all respondents commented that the Internet was "the way" to be aware of trends, the availability of components in the marketplace, and even to share their experiences. Respondents mostly used domain-specific websites they knew to keep themselves updated about technological trends (Res-B). Some examples of these are: TheServerSide, ExpertsExchange, StackOverFlow. These websites commonly provide a way to interact with other integrators in the domain, and share experiences of components and other marketplace trends. Furthermore, it was observed that the component searching process is becoming a gradual activity rather than being totally on-demand. Internet search engines such as Google (Res-C) were vital to browsing the Internet.

Other resources such as written material, e.g., books or specialized reports (Res-D), and hiring specialized consultant companies (Res-E) have not been explicitly evidenced by any other empirical study. The responses also show that the clients may be used as a potential source of component identification (Res-F), as they often are familiar with the domain and can suggest some initial component alternatives.

The awareness of, and relationship with, component providers (Res-G) to gather component information was not as popular as expected. Surprisingly, only 1 interviewee stated the use of freely available repositories, such as SourceForge, to search for components (Res-H).

Summarizing, we may say that the influence of experience led to two different scenarios in the component searching phase: (a) skipping the searching stage and only considering a known component (even if it is not the best one to cover the requirements) or (b) performing the search in a more predictive way by consulting experience networks, or domain-specific communities (i.e., integrators that are familiar with the domain usually know where to search or ask when they are looking for components). In contrast, when integrators did not have previous experience, they mainly used two different practices: (a) using Google for browsing the Internet or (b) hiring consultancy companies in the case of critical projects.

4.2. RQ2. Industrial OTS component evaluation practices

4.2.1. Processes to evaluate components

As shown in Table 6, the most popular and representative process was to informally evaluate components. Most of the respondents mentioned that they did not use nor know of any formal procedure or method to drive the evaluation and decision of components (Eval-A). Instead, they proceeded informally, often without even documenting the information on the various components for their subsequent comparison. Respondents agreed that the criticality of the component within the system to be built and previous experience with the component have a direct influence on the evaluation process: *"there are a lot of variances in this process given the importance of the component to be selected or if you have used it before"* (**B**). In general, components that do not play a critical

Table 5
Categories of resources used in component selection.

Code	Description	Representative quotes	Freq.
Res-A	**People's experiences**: The respondents gathered experiences from people inside the company, and from external contacts in meetings, conferences, and diverse professional networks.	*"We are very influenced by . . . word of mouth"* (**B**) *"We always try to ask other public administrations how they solved similar problems to have a clearer idea of the component we need to identify"* (**K2**)	12
Res-B	**Domain specific web sites**: The respondents continuously monitored domain-specific websites.	*"We identify components by . . . rumors in forums and portals that are recognized in our environment and that we usually follow"* (**D**)	11
Res-C	**Internet search engines**: The respondents browsed the Internet using search engines (all respondents mentioned Google).	*"When we have to identify components, it is mostly Google that works. Especially if we do not know the component's domain, otherwise we use other resources that do not retrieve as many hits as Google does"* (**I**)	11
Res-D	**Written material**: The respondents used books, magazines or specialized reports to keep themselves updated about new technologies and available components.	*"We usually buy books as they provide some suggestions about technologies"* (**D**). *"Our decisions are typically influenced by benchmarks done by companies such as Gartner or Forrester"* (**S**)	5
Res-E	**Consultancy companies**: The respondent found components through hiring specialized consultancy companies.	*"These kinds of consultant companies have extensive experience in the domains they work on. As a result, they know the products and trends of that area; they are familiar with the pros and cons of these products; so it is easier for them to evaluate and suggest components. They do it quicker and their results are more reliable than our results when we are not familiar with the area"* (**O1**)	5
Res-F	**The client**: The respondents stated that clients were a source of knowledge for identifying components.	*"Sometimes the client knows the domain better than you and suggests some candidate components"* (**R**)	2
Res-G	**Providers**: The respondents said that the relationship with/awareness of providers was useful to identify components.	*"We are used to having relationships with providers in the domain, so they are continuously offering information about trends of their products"* (**K2**)	2
Res-H	**Component repositories**: The respondents used component repositories, such as SourceForge, to identify components.	*"I use to search on SourceForge"* (**Q2**)	1

role on the system tend to be more informally evaluated. Sometimes the evaluation and decision is just based on the awareness of positive opinions about the component from the experiences of internal or external people to the organization. On the contrary, when the criticality of the component is high, integrators tend to invest more time and resources for evaluating the candidate components. When we inquired if this could be considered a representative process for all company projects, all respondents agreed on that. Some respondents also justified this as: *"We have several projects and several selection processes, each one being different: the evaluation criteria changes with each project and with each client as well as the process. So it is impossible to try to have an established procedure"* (**P**).

In some cases, the influence of previous experience is such that the process of evaluating candidate components was skipped (Eval-B). Respondents said that personal experience and the experience of others play a crucial role in the whole component selection activity, especially to face time-to-market demands. One of the respondents said, "*We do not usually have time; you always have to do it faster. So, we need to capitalize our previous knowledge and at the same time . . . avoid risks of introducing new and unknown components"* (**P**). When we inquired if this was a representative practice for other projects of the organization, the respondents emphasized that it depended on the criticality of the component and the project.

Hiring external consultants to drive the evaluation process was an emerging practice. It covers two categories corresponding to the involvement of the respondent with the hired consultant. Respondents belonging to Eval-C closely participated with the hired consultant to perform the evaluation tasks, while Eval-D respondents did not, so they could not provide further details of the evaluation procedure. In the case of Eval-C, the respondents applied the OSMM method (Golden, 2004) to drive the evaluation of components in the company. When we inquired if this was a representative practice in the company, they stated that they only

Table 6
Characterization of answers regarding processes to evaluate components.

Code	Description	Representative quotes	Freq.
Eval-A	**Informal procedures**. Respondents neither used nor knew of any formal procedure or method for selecting components.	*"We do not follow any precise methodology; it was totally ad-hoc"* (**L**)	16
Eval-B	**No evaluation**. Respondents recognized that no evaluation of candidates was performed as they directly chose one component they had used before.	*"In this case, I knew the components and there were not too many options (or they do more or less the same) so, we directly used this one"* (**K1**)	2
Eval-C	**Hire support to perform the task**. The respondents emphasized that a company was hired to lead the application of the Open Source Maturity Model (OSMM) for evaluating candidate components.	*"We had support from a company to apply the OSMM. The process evaluation was: we made a table with the requirements based on interviews with stakeholders (but we did not do a very exhaustive study) and we assigned weights to the requirements. Then, we assessed each candidate based on this table. Afterwards we did proof of concepts or prototypes for the 3 final candidates"* (**O1**)	2[a]
Eval-D	**Outsource the task**. A company was hired to perform the evaluation of candidate components (the procedure used was unknown to the respondent).	*"This was a very important and critical project. Since we did not have much experience on this domain, we hired an expert company to suggest the best component for our needs"* (**K2**)	1
Eval-E	**Established evaluation procedures**. The respondent used established procedures based on the Kano model to evaluate candidate components.	*"For evaluating the components in this project, we applied the specific checklist we have. It is based on the Kano model"* (**S**)	1
Eval-F	**Tool supported evaluation**. The company developed a tool to partially support the candidate components' evaluation.	*"We have supporting tools that help us in some evaluation tasks. We used to work specifically in the business application domain, and with the years we have developed a tool that helps us to rank ERP components based on the values of a set of attributes we consider important"* (**T**)	1
	Total		23

[a] These two respondents are from the same organization.

Please cite this article in press as: Ayala, C., et al., Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. J. Syst. Software (2010), doi:10.1016/j.jss.2010.10.019

Table 7
Characterization of answers regarding how component information was acquired and assessed.

Code	Description	Representative quotes	Freq.
Info-A	**Straightforward testing of trivial functionality**. Respondents just tested some basic functionality of candidate components to have an overview of whether the component might work as expected.	*"We downloaded a demo for testing the component"* (**Q2**)	9
Info-B	**Building a prototype**. Respondents spent some time and effort building a prototype to check the behavior of the component in the expected environment.	*"It was a proof of concept (i.e., a simple prototype) to test the component in the expected environment"* (**H**)	6
Info-C	**Asking for people's experiences**. Respondents based their evaluation on previous experiences, mainly of people inside the company.	*"We basically searched for internal experiences in the company or . . . other contacts in communities. As they gave positive comments about the component we went for it"* (**L**)	6
Info-D	**Just reading functional information of the component**. Respondent does not even test the component functionality but only checks the technical documentation available.	*"We checked the technical documentation of the component and ask[ed] in forums for more technical and detailed information"* (**M**)	1
Info-E	**Hiring a consultant**. Respondent did not know how components were assessed as the task was hired and the process was unknown	*"It was an outsourced company that did it"* (**K2**)	1
	Total		23

hire companies to support the evaluation tasks for critical projects. Regarding the application of the OSMM methodology for evaluating components, they highlighted that they were trained on how to apply the method, but they do not apply it on a daily basis: *"We do not use the complete OSMM methodology in the day to day projects. We sometimes informally use a very small set of the evaluation criteria that the methodology suggests (e.g., the maturity of the OSS community or things like that) but never all. We could not do it as it implies a lot of time to find all the information, so for non-critical components we mostly base our decisions only on recommendations from our development team"* (O2). Similarly, Eval-D respondents stated that hiring a consultant company to drive the evaluation tasks was not a representative practice for all projects of the company, but a controlled strategy to deal with the risks of selecting components in critical projects.

Only one of the approached companies stated that they follow an established procedure to evaluate the candidate components (Eval-E). The respondent explained that the company follows strict procedures to procure components as they need to ensure a safe commercialization environment for the resulting product. The procedure used was envisaged by the company and is based on the Kano model (Kano et al., 1984). This model is a theory of product development and customer satisfaction that uses five categories to represent product attributes that are important for customers. He agreed that this was a representative practice of the company. *"Yes. We usually follow the classical practices of product management discipline. We have established procedures to decide whether to buy a third party component or to develop the components ourselves. We have an established template based on the Kano model where we have to fill in the component's information"* (**S**).

Another respondent (**T**) emphasized that even when they do not follow established procedures, they have developed a kind of spreadsheet tool that helps them to assign weights to some criteria for ranking candidate components in the Enterprise Resource Planning (ERP) domain (Eval-F). The respondent also agreed that this was a representative evaluation practice for the projects in the company.

4.2.2. Resources used to evaluate OTS components

In general, all respondents mentioned that they search for component information mainly by consulting both the provider and the resources listed in Table 5. To gain a further understanding as to how evaluation was performed, we asked the respondents to further describe how the information was used. Table 7 summarizes the findings.

As can be observed, the most popular means of component evaluation was the straightforward testing of very basic component functionality (Info-A). Regarding the representativeness of this practice for other projects in the company, most of the respondents stated that this was not representative as it depended on several factors, such as the availability of the component to be tested (i.e., OSS components are able to be downloaded, but in the case of COTS, there is not always a demo to try on), and the criticality of the evaluated component on the system to be built (i.e., critical components tend to be tested more consciously).

Other respondents stated that they built a prototype to check if the component behaved as expected (Info-B). These prototypes ranged from straightforward prototypes to more formal ones that required a significant effort to set up a suitable testing infrastructure. Respondents said it was not a representative practice for other projects in the company. Rather, it is mainly done when the component is critical and/or they are using the components for the first time.

In line with the previous comment, several respondents said that the component was mainly evaluated based on comments from other developers in the company or other external people they trusted (Info-C). Respondents mentioned that this practice may be considered usual only when the component to be selected does not play a critical role in the entire system.

One respondent mentioned that the evaluation of the component was based on reading the component documentation from the provider without fully testing the component in advance (Info-D). The respondent emphasized that this was done as they trusted the provider information and support (they previously had a close relationship with this provider). Regarding the representativeness of this practice in the company, he said that this was not representative as in most cases component documentation is not of good quality and/or relationships with providers are not so close. In addition, several respondents commented that most of the component information available was targeting end users rather than integrators. This finding also coincides with (Jansen et al., 2008).

Finally, one respondent did not provide details of the evaluation activity as his company outsourced the evaluation task to another organization.

4.2.3. Evaluation criteria

Although several respondents stated that there was not a formal list of established criteria for evaluating components, all of them mentioned some factors that were taken into account for choosing the component. Table 8 categorizes the answers from the respondents. Most categories in Table 8 include criteria that entail several interrelated factors.

Results show that satisfying functional requirements from the clients (Crit-A) and ensuring the technological stability and evolution/maintenance of the component by the provider (Crit-B)

Table 8
Characterization of answers regarding component evaluation.

Code	Description	Representative quotes	Freq
Crit-A	Compliance with client's functional requirements	*"We check that the component fits the functional requirements"* (**A**)	18
Crit-B	To ensure technological stability and evolution of the component and the provider	*"You always try to ensure technological stability, so you try to see if the component will be maintained by the provider. In other words if the provider will evolve the component"* (**D**)	18
Crit-C	Evidence of successful component usage	*"If we see the component successfully running in others' environments it is a very influential factor for us, especially if we see or know that the component is being used by a big and well-known company"* (**D**)	11
Crit-D	Availability of support	*"You need to make sure that you can get skilled people during the integration and during the whole system lifecycle"* (**B**)	10
Crit-E	Previous experience with the component	*"We very often fall back on what we have used or know from before. We favor the components we have experience with"* (**A**)	10
Crit-F	Ease of component integration	*"It should be relatively easy to start using the component, and it should enable you to reach your goal faster"* (**A**)	8
Crit-G	Performance and scalability of the component	*"Performance is important"* (**P**)	8
Crit-H	Licensing terms	*"We checked the kind of licenses and we normally discard some components because of this"* (**R**)	6
Crit-I	Price	*"Price is very important for us as we are a small company"* (**H**)	4
Crit-J	Quality of documentation	*"We check, among other things, the documentation from the provider"* (**D**)	3
Crit-K	Source code availability	*"Source code availability is important because we prefer to have ... control over the source code to avoid the feeling/fear of things going wrong without us knowing why and what ... is causing it"* (**C**)	3

were the two most popular criteria used to evaluate components. Regarding the latter, respondents stated such diverse aspects as: *"It is important to assure that the component can be maintained over time by its provider. So we check the activity of the community: how often they release, how many errors are reported and fixed, and things like that...* (**D**), *"We check that there is an updated web"* (**H**), *"We ... trust ... the reputation of the provider"* (**F**). It is important to mention that throughout the interviews we observed that several respondents did not explicitly mention functional requirements as they assumed it was not a comparison factor among candidate components that equally covered functionality. Instead, they assumed that functionality was an implicit factor considered when identifying components. Therefore, we may argue that although Crit-A and Crit-B have the same number of respondents, Crit-A could have higher relevance.

Several respondents considered that having real evidence of the successful usage of the component by other companies (Crit-C) was a very influential factor. *"We always check several factors, but the most important one is to see that the component is being used by other companies. This makes you trust ... the component. I would say that selection is sometimes a matter of trust of components and the providers"* (**R**). *"We usually try to find another company that is using the component and ask them for their experience"* (**K2**).

Examples of factors related to the availability of component support from the integration stage to the entire system lifecycle (Crit-D) were, "*It is important to ensure the availability of companies that may provide support in case of integration issues or any other issue*" (**K1**) or *"we make sure that we have programmers that can modify the tool (if OSS) or to reserve a budget for support from the provider (if COTS)"* (**N**).

Factors related to previous experiences with component and knowledge reuse (Crit-E) were also considered important. "*We choose components that have the same programming language we know*" (**H**). "*We try to select components that are very used in our client's environment, even if we know they are not the best ones in the market. But in this way we can reuse our knowledge and satisfy our clients' requirements*" (**Q1**).

Eight respondents mentioned the ease of integrating the component (Crit-F) and performance and scalability of the component itself (Crit-G) as factors that have a relative importance when evaluating the components. *"Non-functional information as performance is very important, but due to time-to-market pressures, we do not have enough time to verify it"* (**R**).

Factors such as licensing terms (Crit-H), price (Crit-I), quality of documentation (Crit-J), and source code availability (Crit-K) were controversial as some respondents considered these categories quite important while others explicitly said that these factors were not relevant at all. Some examples are: *"Licenses are not a problem if you develop custom software, I mean, that you will not commercialize the resulting software in multiple copies"* (**H**), *"We did not care much about price (as long as it is not extremely expensive). I would say that sometimes the price does not really matter depending on the project"* (**C**), *and "We do not usually look at the code that much. If we have to modify a lot of code, we just discard the component and consider developing the functionality ourselves. We almost never touch the code"* (**Q2**).

4.3. RQ3. Industrial OTS component choosing practices

This section describes the procedures followed to choose components and the development phase where components are selected.

4.3.1. Processes for choosing components

To understand how the final decision was taken, we asked the respondents to describe some details about which procedure was performed to make the final decision. Table 9 summarizes the two self-explanatory categories found.

The most typical situation was that the development team or the boss decided the component(s) to be selected (Dec-A). The results from the question, "Is this typically how you do this?", show that respondents considered a typical situation in most projects to be one in which clients are usually not aware of the internal implementation of the resulting system. "*Usually, the clients do not even know which components are being integrated."* (***G***). On the other hand, in some projects, the final decision was taken by the client (Dec-B). Respondents highlighted that in some projects it was the client who had to finally decide the component to be chosen. This was especially true for companies such as ***T*** whose business model is based on providing component evaluation surveys where it is always the client who has the final decision.

It is important to highlight that in both cases, respondents recognized that the decisions were greatly influenced by some strategic business issues such as established relationships with providers, previous knowledge or experience with a candidate component,

Please cite this article in press as: Ayala, C., et al., Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. J. Syst. Software (2010), doi:10.1016/j.jss.2010.10.019

Table 9
Characterization of answers related to the final decision.

Code	Description	Quotes	Freq.
Dec-A	Decision was made by the project team	*"There are some bosses higher up from the development team who had to give the final "go""* (B) *"It was a kind of workshop where we made some simple examples (i.e., prototypes) and then the people from the project team choose the one that they would like to use"* (**D**)	18
Dec-B	Decision was made by the client (based on our suggestions)	*"We showed . . . the clients some simple prototypes with the alternatives and suggested the best option. As they agreed we went for this component"* (**P**) *"We deliver the documentation about the evaluation to the client and it is always the client who decides. But generally, the decision we suggest is the one taken"* (**T**)	5
	Total		23

technologies, or programming languages that the teams already master.

4.3.2. When in the development process were the components chosen?

Results show that in all assessed cases, respondents said that critical components were chosen at very early stages of the development process (i.e., at the contractual phase), and that it was a representative practice in the company. Their main motivations were:

(a) to reduce the potential risks at late stages of the project when project restrictions and risks are higher. *"We avoid selecting components at late stages. In such a case, you do not really evaluate that the component is the best option for the project, but use the first component at hand in order to solve the situation. I mean, you mostly try to solve the problem instead of finding a suitable component, therefore this may be so risky"* (**B**).

(b) to avoid any kind of contractual breach with clients. *"You must perform component selection at a very early phase because your client does not want surprises. You cannot tell to your client that you do not know which components will be used because it mainly implies budget risks. So, before sending the formal proposal/contract to the client, you must know what component you will use. We know that this is probably not the best way to proceed, but this is how the business works"* (**P**).

5. Discussion of main findings

The previous section aimed to present a comprehensive view of the data gathered from the interviews in order to provide the reader with rich data for their own assessment. This section aims at discussing the most important findings and observations from the data with respect to the stated research questions and the existing body of research and evidence.

5.1. Processes to select components

5.1.1. Continuous monitoring of the marketplace is becoming a usual and recognized practice inside companies

Our results show that continuous monitoring of the marketplace is becoming a usual practice among integrators to keep themselves updated about components, technologies and trends (even before they have a specific need). Therefore, the search practice is often becoming a continuous monitoring activity rather than being on a project demand basis. Some respondents also stated that their companies had recently set up a dedicated department or person for performing this task. The latter has research and practical implications. On the one hand, it implies a restructuring of the tasks and responsibilities of the software development team, and on the other hand it has increased the need of enabling intra-organizational channels of communication for interacting/informing results.

5.1.2. The use of informal procedures to search for, evaluate and choose components was the most popular way of selecting components

In line with the claim that component selection methods proposed in the literature mostly focus on the component evaluation phase, setting aside the problem of identifying components and related information (Bertoa et al., 2006); we found that the component searching phase was informally performed and mainly influenced by experience in all the companies (as explained below). Regarding component evaluation, it was very interesting to see that in contrast to previous studies that stated that companies neither used nor knew of any formalized methods to select components (Land et al., 2009; Li et al., 2009; Chen et al., 2008; Torchiano and Morisio, 2004); our results suggest that there is an *incipient interest or awareness of some component selection methods*, as some companies intend to apply evaluation methods and tools for supporting evaluation, mainly when using OSS components. Further observations from these companies are:

- Company ***S*** used established procedures based on the Kano model (Kano et al., 1984). Some relevant observations regarding this company were: ***S*** is a big and very well established company that traditionally develops and commercializes the same software products (including updated versions of their products). In addition, the domain approached by the company does not vary either, and they have established agreements with a specific set of component vendors. As a result, the company selection processes are quite stable, making it feasible and valuable to reuse knowledge, and to follow repeatable procedures and guidelines for selecting components.
- Company ***O*** tried to use the OSMM methodology (Golden, 2004). This was mainly motivated by their need to succeed and justify their decision on selecting OSS components over proprietary solutions, and their aim to extend their business model to offer services around OSS-based solutions. It is important to emphasize that the methodology was not applied as is (rather, a very lightweight version of it was employed), mainly because of the problem of getting suitable component information to cover the vast set of evaluation criteria proposed by the methodology. Our observation was that following a method was more sensible when diving into a new domain (as the domain of OSS components is for several organizations).
- Company ***T*** provided support to select OTS components. As knowledge management is vital for offering expert services to their clients, they therefore have the need to develop a software tool to store component information and enable knowledge reuse about components. In this context, we think that incipient organizations providing OTS selection support like ***T*** is, may potentially embrace knowledge reuse strategies and/or tools for ensuring the management of their OTS component related knowledge as this gives them a competitive advantage.

Our assessment and comparison of the context of these companies with the companies that performed component selection in a fully informal way, led us to suggest some factors that might have positively influenced the incipient adoption of formalized evaluation methods, such as stability of domains approached by the companies that valued the reuse of knowledge and the establishment of repeatable procedures or tools; and the increasing adoption of OSS components over proprietary solutions. Regardless of the time and cost restrictions usually associated with these projects, the factors that made the establishment of procedures for selecting components difficult were: the diversity of clients, domains and projects that some companies had to approach, as well as the opportunistic more than systematic way of reusing components. It seems that the more opportunistic the reuse of components is the less motivation there is to use formalized methods.

5.1.3. Previous experience with and criticality of the component in the whole system were the most influential factors leading the way companies selected components

The crucial role of previous experience is given by the fact that companies need to face to-time-to market restrictions, capitalize the knowledge gained when mastering a component as well as to avoid the risk of introducing new components (i.e., component that has not been previously used) in the software development tasks. This is especially true when the component to be integrated plays a critical role in the system to be built. Most integrators defined "criticality of the component" as the over-cost and problems related with changing the component at later stages. At this respect, some respondents said. *"Component evaluation is greatly subjective and mostly based on the experience. Of course we look at the component website, and other portals with forums or comparative tables, etc. but the experience is more important"* (**G**). Another respondent said. "*We do not often test the components throughout; the reason is that we do not have time, you always have to do it faster. Sometimes it is just good feeling or asking others for experience"* (**D**). Furthermore, most of the companies consider component selection as: *"an investment as we have to build prototypes to show to our clients. So, each selection process implies a risk and at the same time is part of the knowledge investment of the company"* (**P**).

While the value of experience is important, considering it as the most influential factor for selecting components is at the same time hampering the fully exploitation of the potential benefits of the OTS marketplace. We observed that several companies used the experience as the first risk reduction strategy when using OTS components. So, they deterred the use of potential components just because they did not have previous experience with these components. It is a fact that, in order to remain competitive, software-intensive organizations should optimally manage the potential risks and rewards of the OTS marketplace (Gartner, 2008). Therefore, it is crucial to establish synergy among integrators, and component providers to provide and enable other resources that better reflect integrators needs and lead integrators to face the risks of using OTS in other ways and not just based on their experience (see Section 5.2).

5.1.4. OTS components are mostly selected at early stages of OBSD

All respondents from our study agreed that the component selection took place at a very early stage of the development process (i.e., at the contractual phase). This result notably differs from the result offered by (Li et al., 2009), where 68% of integrators selected components at early stages, 13% did it at late stages and 3% did not know when components were selected. Regardless of the time difference between the consecution of both studies, and the potential change of the industrial practices during this period, one possible explanation to such variation may be that most of the component selection projects approached by Li et al. (2009) were internal projects where the company did not have contractual constraints. However, most of the projects approached by our study were external and constrained by several contract rules and budget. Therefore, it seems that there is a relationship between the kind of projects (i.e., internal or external), the relevance of the chosen components in the system (i.e., if they play a critical role in the system) and the stages where components are selected. Internal projects seem to have certain flexibility on the stage where selection is performed, while external projects must state this as a contractual agreement at the beginning of the project.

5.1.5. OTS component decisions are mainly taken by the development team

Research on requirement negotiations often assumes that a client will be interested in, and be capable of, discussing OTS component selection issues (Finkelstein et al., 1996). However, in practice this is usually not true. In line with the evidence EV5 from (Li et al., 2009), we found that it was mainly the integrators' organization (i.e., the software project development team) that typically decides OTS components, whereas clients usually only care about the final products.

5.2. Resources used to select components

This subsection describes our findings regarding the resources used to identify, evaluate and choose components.

5.2.1. Integrators mostly use knowledge and experience-sharing mechanisms

While most of current research usually assumes that component providers' portals (Bertoa et al., 2006; Simmons and Dillon, 2006), repositories (Wanyama and Far, 2006) and search engines (Cechich et al., 2006) are the primary ways in which integrators identify components and obtain information about them; the results from our study show that integrators hardly agreed on the use of these resources in practice. Instead, integrators found it more practical to use resources that promoted experience and knowledge sharing mainly on the Internet, for instance domain-specific websites that offer forums to interchange ideas and solutions to common problems (e.g., TheServerSide or ExpertsExchange). The direct interaction with colleagues and professional networks (e.g., asking for comments about a component from a colleague, or attending specialized conferences or workshops) gave valuable results as well. To the best of our knowledge, the exploitation of this social interaction for supporting the OTS component selection has not received great attention yet. There is a demanding need to effectively deal with the inherent subjectivity of this kind of information. Reputation mechanisms as used in other business domains as ebay.com could be really valuable to deal with the subjectivity of diverse opinions.

5.2.2. Integrators typically use Google as a search engine to identify new components and information about them

Regarding search engines, integrators neither mentioned nor used any other search engine than Google to identify components and information about them. This observation fully agreed with (Chen et al., 2008), so it might confirm that the Western industry also uses Google as the Chinese software industry does. Our results related to this specific topic may provide qualitative information to better understand how integrators search for OTS components and information about them. This may be an input for researchers aimed to improve searching functionalities, as (Umarji et al., 2008; Gallardo-Valencia and Sim, 2009).

Please cite this article in press as: Ayala, C., et al., Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. J. Syst. Software (2010), doi:10.1016/j.jss.2010.10.019

5.2.3. *Integrators hardly use repositories to identify components*

Surprisingly, despite the crucial need to have and improve suitable component repositories to find components, which has been greatly discussed in the literature (Wanyama and Far, 2006; Clark et al., 2004), our data shows that current component repositories such as Sourceforge were hardly used by the respondents. A respondent said, "*I know SourceForge, but I do not like it. In portals like this it is really difficult to navigate and find relevant components. It is better to find components in a specialized portal (i.e., domain-specific) or by Google and then go to SourceForge just for downloading it*" (**F**). One of the main problems with the use of repositories seems to be the difficulty of browsing them. Part of the problem and the challenge is that they index components under a single classification, but no classification is correct under all circumstances (Bowker and Leigh Star, 2000). It is impossible in principle to identify all possible relevant and future features of the OTS marketplace, so more flexible classification mechanisms are required to make the repositories useful for industrial practice.

5.2.4. *Hiring specialized companies to select components was used as a risk reduction strategy*

This was an interesting finding as no previous studies had reported on this practice. Some of the approached projects stated that they had hired an expert company for performing the selection tasks. Other respondents agreed that this was a resource also used by their companies in other projects. In addition, all these respondents agreed that this was an effective strategy for dealing with the risks of component selection in critical projects. This finding complements the list of risk reduction strategies when using OTS components found by (Li et al., 2008).

5.2.5. *There seems to be a potential market niche for component selection support*

In line with the previous finding, and based on the data gathered, we observed that hiring support for selecting components was an activity much in demand for critical projects in almost all organizations. Thus, it seems to be a potential market niche for companies aiming to provide support for selecting components. Company ***T*** is a good example of an expert company that offers dedicated support for selecting business application components.

5.2.6. *Some resources used are overlooked by the literature*

Our results show that apart from the resources usually discussed in the literature for searching for components (i.e., providers' portals, search engines and repositories), some additional resources were mentioned by the respondents. These resources included written material (i.e., books, magazines or specialized reports) and the exploitation of the client's knowledge on the domain. Currently, these resources have been practically overlooked by the literature. From the practitioners' point of view, it could be helpful to have further evidence and insights about how such resources may be exploited, and how they may potentially benefit the selection processes.

5.3. *Evaluation criteria used*

Being aware that no universal criteria list can be established, our aim is to provide qualitative information about the investigated projects (see Table A1 in Appendix B) so that the reader can to some extend understand the context in which the projects occurred, and their potential relationship with the criteria mentioned by the respondents (see Table 8). We think that this list may help researchers to identify some common criteria that were used in our approached projects.

5.3.1. *The list of evaluation criteria is neither formally established nor documented.*

Although there is great body of research mainly focused on the establishment of evaluation criteria for assessing candidate components (see Section 2), our results show that the fact that component information must be searched and processed (see Table 7) makes it impractical for integrators to use the proposed evaluation criteria in terms of time and cost associated to the projects. In almost all cases there was not a formal list of established criteria for evaluating the components. The criteria mentioned by the respondents were quite informal, and their use depended on the availability of the information covering the criteria. As a result, metrics over the criteria were not used either.

5.3.2. *Compliance with functional requirements and certainty of the future maintenance by the provider were the most important evaluation criteria*

Although the crucial role of requirement compliance on component evaluation was also stated by previous studies (Li et al., 2009; Chen et al., 2008), our results show that the certainty of the future maintenance of the component by the provider also played a crucial role.

In addition, although our results seem to support the findings from Li et al. (2009) and Chen et al. (2008), and disagree with Torchiano and Morisio (2004) in the claim that architectural compliance was more important for component selection than requirement compliance, the qualitative data we gathered led us to suggest a possible explanation to this fact.

According to the architectural level, Torchiano and Morisio (2004) observed two kinds of projects: (1) projects where integrators first identify the component(s) providing the system's core functionalities and then adopt its architecture and (2) projects where the architecture was already decided and thus, the integrators just look for components that may cover such constraints (i.e., requirements). By assessing our respondents' explanations we realized that most of the approached projects fall into this second category. Even if it was not intentional, the majority of our sample projects represent web information system applications where architectural issues were implicit requirements in the OTS component selection process. Analyzing the data available from (Li et al., 2008; Chen et al., 2008), it seems that their sample projects also fall into the second category. However, most of the projects assessed by Torchiano and Morisio fall into the first category as they referred to coarse-grained applications that played primary roles on diverse architectural layers, e.g., selection of complete Operating Systems such as QNX or FreeBSD; and Database systems such as MySQL or Oracle DB. Therefore, this might explain the potential differences between these studies. As a result, we could suggest that depending on the architectural implications of the component to be selected, the architectural issues may become an explicit or implicit part of the requirements to be covered by the component.

5.3.3. *Relative importance of non-functional criteria*

Although several works in the literature have emphasized the relevance of non-functional requirements in component selection (see Jadhav and Sonar, 2009), our results show that in our approached projects, they were not at the top of the evaluation criteria lists. Instead, respondents said that factors such as the performance of the component and its ease of integration were considered only as relatively important, especially because they are difficult to obtain: "*It is very complicated to find information about non-functional requirements so we end up in the experience (that's why experience is key) and to see if others have succeeded in using the component*" (**I**).

On the other hand, the list of evaluation criteria presented by (Keil and Tiwana, 2005) differs with our results on the importance of some of the criteria. For instance, the criteria ease of use has the lowest importance in Keil's study, while on the contrary it was of considerable importance in our study (see Crit-F in Table 8). Since Keil and Tiwana do not provide further information about their project's context but the fact that they approached COTS components, we cannot further assess other potential sources of variability. At this respect, we could say that given the inherent context-related nature of the evaluation criteria, our results provide a more extensive and detailed list that was elicited from the integrators. In addition, we elicited the rationale and the context to understand when and why these criteria are used (see Table A1 in Appendix B).

5.3.4. *Value of real evidence of the component usage is usually underestimated*

Our results show that real evidence from other developers or companies, illustrating that the component has been successfully adopted in their contexts, played a crucial role in component evaluation. However, this is typically underestimated by the literature and component providers that do not explicitly provide such information. We think that the results of our study may contribute to increasing the awareness of real industrial needs, and motivate practitioners and researchers to focus their efforts on the value of experience and problem reports that show how a certain product was adopted in a different context.

5.3.5. *Potential difference between Oriental and western software industry on the relevance of evaluation criteria*

Availability of support was an important factor to evaluate. Surprisingly, this importance fully contradicts the results from (Chen et al., 2008) that stated that it was the least important criteria for evaluating OSS components in the Chinese industry. Such a difference may suggest potential differences between Western and Oriental software industries.

6. Limitations of the study

Like most studies in Software Engineering, our study faces some reliability threats. This section discusses these threats in terms of construct, internal, and external validity, as suggested by (Robson, 2002; Wohlin et al., 2000). It furthermore emphasizes the corresponding strategies used to deal with these threats.

Regarding construct validity, this study was supported by 2 main principles: rigorous planning of the study, and the establishment of protocols for data collection and data analysis. This was especially important as the research involved several researchers and participants from different countries. In addition, the interview guide used as an instrument to gather data, was carefully designed and piloted with 6 academic and industrial people (these interviews were discarded for the real study) in order to improve its understandability. As a result, some changes in the interviews were done to enhance the elicitation process. For instance, some vocabulary was defined at the beginning of the interview guide to homogenize concepts.

Regarding internal validity, we tried hard to envisage and harmonize the data gathering and the subsequent data analysis strategies. With respect to the data gathering strategy, we took relevant decisions for approaching a further understanding of the OTS selection industrial contexts. One of the main relevant decisions was to focus most of the questions of the interview guide on a single component selection project and a component from that project. In this way, we could further inquire and analyze specific contexts that generated a particular decision. This enhanced the value of our analysis and observations, as it allowed for the understanding of the rationale behind OTS selection decisions and the organizational factors. We are aware that some possible biases may be related to this strategy, for instance the fact that some time has passed since the project was completed, so it could be difficult for the respondents to remember some project details. To reduce the possible side effects of this, we detected sensitive questions in the interview guide (when piloting the guide) and improved the questions that involved sensitive information. In addition, we sent the interview guide in advance to the respondents so that they could be informed of the kind of questions to be asked. As a result, when performing the study, we rarely experienced respondents having difficulty remembering project details.

Another factor raised was that the projects were selected by the participants. They may have selected the most successful project to base their answers on, although we asked them to use the most familiar one. We dealt with this possible bias by explaining to the respondents that our study was not focused on analyzing "wrong practices" but on knowing "how it is done in industrial practice". In addition, there is always the possibility that the respondent forgets something (usually something that he/she does not consider important) or does not explicitly state it when he/she is asked about it (Seaman, 1999). To reduce this issue, the research team discussed some potential things that might be omitted by the respondents, and we paid particular attention to this during the interviews in order to ask for clarifications if necessary. Even so, we experienced this case when asking about which was the most important evaluation criteria to evaluate components. 18 respondents explicitly mentioned that "compliance with clients' functional requirements" (Crit-A, see Table 8) was the most important factor. However, we realized that even if the other respondents did not explicitly mention it in their responses, they assumed that it was an obvious factor that did not need mentioning. Despite this, the results obtained by the study were still not greatly affected by this as Crit-A ended up at the top of the list of the criteria.

With respect to the data analysis strategy, recording all interviews (and later on transcribing them) contributed to a better understanding and assessment of the data gathered. The generated categories were analyzed, discussed and reviewed by all researchers of the team to ensure their accuracy, understanding and agreement. Furthermore, categories were checked with respect to the data gathered in order to confirm that none of the stated categories refuted any of the conclusions, and that the variability factors were well understood by the research team. Furthermore, we triangulated responses, especially in the case of respondents from the same company, so that we strengthened the correct understanding of the results.

Regarding external validity, it is important to highlight that qualitative studies, such as the one we performed, rarely attempt to make universal generalizations beyond the studied setting. Instead, they are more concerned with characterizing, explaining and understanding the phenomena under the contexts of study (Robson, 2002). To strengthen the external validity, we addressed several topics in our study. Some of the most relevant ones are listed. First, the companies in this study were selected by a strategy combining convenience and maximum variation sampling from 3 different countries (Spain, Norway and Luxembourg). The use of this convenience sampling approach reflects the difficulty in gaining industrial participation in these kinds of surveys. We tried to mitigate any possible bias traditionally related to convenience sampling (Robson, 2002) by combining a maximum variation sampling, so that the approached organizations covered different characteristics regarding size, application domain, and business area. Second, another factor strengthening the external validity was that we had no control over the projects and components chosen by the respondents. Third, the approached projects and OTS components used were of different size and types, and the interviewees had different backgrounds (see Table 2). Nevertheless, most of the resulting sampling companies were developing web applications, and the approached projects did not cover domains such as real time or

Table 10
Summary of previous evidence on OTS component selection.

OTS selection activities	Existing evidence (see Table 1)	Comparison of results	Some further findings from this study
Searching/locating components	EV7 (Chen et al., 2008)	√	Integrators hardly use repositories to identify components. Identification of components and information about them is becoming a continuous monitoring activity rather than being on a project demand basis.
	EV11 (Umarji et al., 2008)	√	We provide further qualitative data about the resources that are used by integrators to search for OTS components.
Evaluating components	EV1 (Torchiano and Morisio, 2004)	√	Incipient adoption (but still quite limited) of some component selection methods.
	EV3 (Li et al., 2009)		Qualitative data to understand the adoption contexts.
	EV8 (Chen et al., 2008)		Some companies are starting to familiarize themselves with existing formal procedures, but they do not regularly apply them (mainly because of the problem of getting suitable component information that cover the suggested evaluation criteria).
	EV2 (Torchiano and Morisio, 2004)[a]	±	Depending on the architectural implications of the OTS component to be selected, architectural issues may become an implicit or explicit part of the requirements to be covered by the component. The qualitative data we gathered helped to understand the potential differences among the results from (Li et al., 2009; Chen et al., 2008) and (Torchiano and Morisio, 2004).
	EV6 (Keil and Tiwana, 2005)	±	Some variations exist with regard to the prioritization of the criteria. For instance, compliance with client requirements was the most important factor (coinciding with Chen et al., 2008; Keil and Tiwana, 2005). However, availability of support was an important factor in our study, while Chen et al., found that it was among the least important ones.
	EV9 (Chen et al., 2008)		In addition, we found that some factors considered by integrators in the industry have been greatly overlooked by previous research
	EV10 (Chen et al., 2008)	√	Experience and knowledge sharing led the way companies select components.
	EV12 (Land et al., 2009)	√	Other evaluation practices were identified and contextualized.
Choosing components	EV4 (Li et al., 2009)	×	Integrators mostly chose components at the contractual agreement stage of the development process.
	EV5 (Li et al., 2009)	√	OTS component decisions are mainly taken by the development team instead of by the client. In both cases, the decisions are greatly influenced by some strategic business issues.

Legend: confirm (√), partially support or help to understand (±), or refuse (×) previous evidence.

[a] Contradicted by Li et al. (2009) and Chen et al. (2008), see explanation in Section 5.3.

life critical requirements. We are aware that both factors may have an impact on how components are selected, and so we highlight that our findings might be considered more relevant for the web information systems and non-critical domains.

On the other hand, given the contextual and non-deterministic nature of OTS component selection, the results presented here should be interpreted with caution. For instance, the resulting list of evaluation criteria presented in Table 8 should be interpreted in the context of the approached projects. Such a list informs about the factors that were taken into account during OTS component evaluation, but it does not imply that the factors that ended up at the top of the list were the ones that dominated the final decision (see Section 4.3.1). We tried to offer contextual details of the OTS selection practices in the sampling companies, so that researchers and practitioners may further understand them. Furthermore, we emphasize that our findings should not be taken as assertions but also as potential hypotheses that need to be further validated.

7. Conclusions and future work

This qualitative study presents our results related to the exploration of industrial OTS component selection practices in 20 organizations from Spain, Norway and Luxembourg. The main findings of the study reveal some practices that have not been previously reported that are becoming part of software development, as well as potential market niches for software-intensive companies. In addition, the results emphasize issues that have received considerable attention by researchers but not by industry, and vice versa. Furthermore, when comparing our results with some previous empirical works that provided some evidence about OTS component selection (e.g., Torchiano and Morisio, 2004; Keil and Tiwana, 2005; Chen et al., 2008; Li et al., 2009; Land et al., 2009), we found that some of the problems still remain, but some variations seem to have occurred. Based on the findings and discussions presented in the paper, Table 10 summarizes the extent to which the results confirm (√), partially support or help to understand (±), or refuse (×) these previous evidences (summarized in Table 1).

The results of this work may provide a broad understanding of industrial OTS selection practices and have positive implication for research and practice.

- For researchers, they may envisage their solutions, mainly considering the factors that are really used in industrial practice, and to identify new research challenges and aspects that have been overlooked by the literature. We furthermore advise researchers to be cautious when basing their research on uncertain "assumptions" that might not correspond to reality. Researchers should try to report their experiences and the evidence that shows their successful and failed OTS component selection practices, since evidence from other contexts is used as an important part of the identification and evaluation of components. This evidence should be made available and be integrated into the proposed tools and methods.

- For software-intensive organizations that perform OTS component selection practices, the results presented help to increase their awareness of the implication of factors such as experience and previous knowledge in the whole component selection process. This then enables them to consider allowing their employees to spend more time in familiarizing themselves with their domains, and to foster experience-sharing not only inside the company but also externally. Furthermore, some organizations may even consider offering consultancy services to other organizations for selecting components, as there seems to be a market niche for that.
- For component providers that may also use the information about how components are selected, which resources are usually used, and the evaluation criteria that are important for integrators. This will help them to better address their product improvement and marketing strategies.

To conclude, we remark that while our findings should be further validated, they represent an initial step forward in maturing the OTS component marketplace. We hope that our study might motivate researchers and practitioners to envisage more effective actions to improve the state of the practice; and thereby contribute to an optimal management of the potential risks and rewards of using OTS components. In particular, we expect to motivate researchers to perform more qualitative research that allows for a better understanding of real world practices.

We see this study as an initial step to drive our future studies on the right path. Thus, the results obtained in this study will direct our future research in this area. We are especially interested in extending the scope of this study to other domains in order to validate and improve our results. In addition, we aimed to further inquiry what are the main industrial problems in order to arrange a more consolidated research agenda based on real industrial needs.

Acknowledgements

We thank all people that participated in piloting an early version of the interview guide and the interview participants who took time from their workdays to participate in our interviews. Claudia Ayala's work was mostly carried out during the tenure of an ERCIM Alain Bensoussan Postdoctoral Fellowship. This work has been partially supported by the Spanish project ref. TIN2007-64753.

Appendix A. Interview guide

This appendix shows the interview guide used to perform the interviews. It contained a definitions section used to explain some terms to the respondents, followed by sections related to the topics addressed by this paper.

A.1. *Definitions*

Term	Definition
Activity	A life-cycle phase during software development.
Adoption	The process of selecting and integrating OTS components into an OTS based system which is maintained.
Community	The group of users and developers around an OTS component. This includes "traditional" OSS communities but also communities around other software products and online forums and user groups with a general interest in software development.
Component	It is a commercially available or open source piece of software that other software projects can reuse and integrate into their own products.
Component provider	Makes, maintains and publishes new components for others to use
COTS	Commercial-Off-The-Shelf: Software for sale at a certain price and terms. Hard to influence further development.
OSS	Open Source Software: software with a license granting users the freedom to run, study, modify and redistribute the software. Redistribution of derivative products must sometimes be done with the same/original license.
OTS component	Off-The-Shelf component: COTS or OSS components.
Internet resource	An Internet resource is here defined as a web portal, software repository, the web page of a software provider or some other web page containing information or resources relevant to software development.
Project	Set of activities and their associated resources to produce a system according to certain requirements or goals.
Selection	Identification, evaluation and choice—here of OTS components.
System	Here, a software system. Results of a software development project, excluding code and documentation.
Trust	Assured reliance that a product's or provider's properties are as announced.
Personal network	Colleagues, friends, customers, competitors or other people you know and interact with.

A.2. *Questions guide*

The projects we want to investigate in this interview are **typical completed software development projects** using **one or more OTS components**. If you have experience with several such projects, please **select the project** that you are **most familiar with,** and **base your answers on the system developed in that project**.
About The System

1. What kind of functionality is provided by the system and who are its users?
2. Which COTS and OSS components did you integrate into the system and what were their main characteristics? Could you please give an overview of the architecture of the system?

A.3. *General selection practices*

A.3.1. *Identification*

Identification/searching is the first part of the selection process. The following questions are aimed at understanding how you searched for and identified components in this project.

3. Which activities were performed to identify this component in this project?
If the component was identified before, could you please detail how you identified it for the first time?
4. Which resources did you consult when searching for this component?
Please list the resources you interacted with or used in the selection of components.
Is this typically how you identify components? What has the strongest influence on the identification of OTS components?

Table A1
Relationship among categories.

Respondent	Search (definition of categories is in Table 4)	Resources (definition of categories is in Table 5)	Evaluation (definition of categories is in Table 6)	Information (definition of categories is in Table 7)	Decision (definition of categories is in Table 9)
A	SerA	Res-B, Res-A	Eval-A	Info-A	Dec-A
B	SerD	Res-G, Res-B, Res-A	Eval-B	Info-B	Dec-A
C	SerB	Res-C, Res-B, Res-D, Res-E	Eval-A	Info-A	Dec-A
D	SerB	Res-C, Res-A	Eval-A	Info-A	Dec-A
E	SerA	Res-A, Res-D, Res-B	Eval-A	Info-A	Dec-A
F	SerA	Res-A, Res-C	Eval-A	Info-A	Dec-A
G	SerB	Res-C, Res-B	Eval-A	Info-C	Dec-A
H	SerA	Res-B	Eval-A	Info-B	Dec-A
I	SerA	Res-A, Res-C, Res-D	Eval-A	Info-A	Dec-A
J	SerA	Res-D, Res-A	Eval-A	Info-C	Dec-B
K1	SerA	Res-B, Res-C, Res-A	Eval-B	Info-C	Dec-A
K2	SerC	Res-E	Eval-D	Info-E	Dec-A
L	SerB	Res-C, Res-B, Res-A	Eval-A	Info-B	Dec-A
M	SerA	Res-B	Eval-A	Info-D	Dec-A
N	SerB	Res-C, Res-A	Eval-A	Info-B	Dec-B
O1	SerC	Res-E	Eval-C	Info-C	Dec-A
O2	SerC	Res-E	Eval-C	Info-C	Dec-A
P	SerA	Res-B, Res-F	Eval-A	Info-B	Dec-B
Q1	SerA	Res-B, Res-E, Res-D	Eval-A	Info-A	Dec-A
Q2	SerB	Res-A, Res-H, Res-C	Eval-A	Info-A	Dec-A
R	SerD	Res-F	Eval-A	Info-A	Dec-B
S	SerA	Res-C, Res-A	Eval-E	Info-B	Dec-A
T	SerA	Res-C, Res-A, Res-G	Eval-F	Info-C	Dec-B

A.4. Evaluation and choice

Evaluation and choice use are the second and third activities of the selection process. The following questions are aimed at understanding how you evaluated the candidate components and chose the component you integrated.

5. Which activities were performed to evaluate and choose this component?
6. Which evaluation criteria were taken into account to evaluate and choose this component?
7. Why were these evaluation criteria important and which was/were the most important one(s)?
8. Which resources and information sources were consulted to evaluate components? Please list the resources you interacted with or used.
9. Is this typically how you evaluate and choose components? What has the strongest influence on the evaluation of OTS components?

Appendix B. Table of code relationships

This appendix contains a summary of the categories of the answers from each respondent. Through this paper, we have discussed and based our findings on this data. By providing Table A1, we enable the reader to verify the discussions and findings from the paper and assess other potential relationships that are not related with the research questions addressed by this paper. The first column shows the respondent code and the subsequent columns show the category codes (introduced and detailed in Section 5) that each respondent answered.

References

Aguirre, J., 2005. IPSComp—Intelligent Portal for Searching Components. Master Thesis. Vrije Universiteit Brussel—Belgium in collaboration with Ecole des Mines de Nantes, France.

Anderson, W.B., 2004. COTS selection and adoption in a small business environment: how do you downsize the process? In: Proceedings of International Conference on COTS-Based Software Systems (ICCBSS), p. 216.

Ankolekar, A., Herbsleb, J., Sycara, K., 2003. Addressing challenges to open source collaboration with semantic web. In: Proceedings of 3rd Workshop on Open Source Software Engineering, the 25th International Conference on Software Engineering (ICSE), Portland, OR, pp. 9–14.

Ayala, C., Hauge, Ø., Conradi, R., Franch, X., Li, J., Sandanger-Velle, K., 2009. Challenges of the open source component marketplace in the industry. In: Proceedings of the 5th IFIP Working Group 2.13 International Conference on Open Source Systems (OSS2009)—Open Source Ecosystems: Diverse Communities. Springer, Skövde, Sweden, pp. 213–224.

Basili, V.R., Elbaum, S., 2006. Empirically driven SE research: state of the art and required maturity. In: Proceedings of the International Conference on Software Engineering (ICSE'06), p. p. 32.

Bertoa, M., Troya, J.M., Vallecillo, A., 2006. A Survey on the quality information provided by software component vendors. Journal of Systems and Software 79, 427–439.

Birkmeier, D., Overhage, S., 2009. On component identification approaches—classification state of the art, and comparison. In: Proceedings of the International Conference on Component-Based Software Engineering, Lecture Notes in Computer Science, pp. 1–18.

Boeg, J., 2006. Certifying software component attributes. IEEE Software 23 (3), 74–81.

Bowker, G.C., Leigh Star, S., 2000. Sorting Things Out Classification and Its Consequences. University of California, San Diego.

Carvallo, J.P., Franch, X., Quer, C., 2007. Determining criteria for selecting software components: lessons learned. IEEE Software 24 (3), 84–94.

Cechich, A., Requile-Romanczuk, A., Aguirre, J., Luzuriaga, J.M., 2006. Trends on COTS component identification. In: Proceedings of the Fifth International Conference on Commercial-Off-The-Shelf (COTS)-Based Software Systems (ICCBSS'06). IEEE Computer Society, pp. 90–99.

Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., Liu, C., 2008. An empirical study on software development with open source components in the Chinese software industry. Software Process: Improvement and Practice 13 (1), 89–100.

Ciokolwski, M., Soto, M., 2008. Towards a comprehensive approach for assessing open source projects. In: Proceedings of the International Conference on Software Process and Product Measurement. Springer, pp. 316–330.

Clark, J., Clarke, C., De Panfilis, S., Granatella, G., Predonzani, P., Sillitti, A., Succi, G., Vernazza, T., 2004. Selecting components in large COTS repositories. Journal of Systems and Software 73 (2), 323–331.

Conradi, R., Li, J., Slyngstad, O.P.N., Kampenes, V.B., Bunse, C., Morisio, M., Torchiano, M., 2005. Reflections on conducting an international survey on software engineering. In: Proceedings of the International Symposium on Empirical Software Engineering (ISESE'05). IEEE CS Press, pp. 214–223.

Cruz, D., Wieland, T., Ziegler, A., 2006. Evaluation criteria for free/open source software products based on project analysis. Software Process: Improvement and Practice 11 (2), 107–122.

Del Bianco, V., Lavazza, L., Morasca, S., Taibi, D., 2009. Quality of open source software: the QualiPSo Trustworthiness Model. In: Proceedings of the 5th IFIP Working Group 2.13 International Conference on Open Source Systems (OSS2009)—Open Source Ecosystems: Diverse Communities. Springer, Skövde, Sweden, pp. 199–212.

Erdogmus, H., 2010. How important is evidence really? IEEE Software 27 (3), 2–5.

Finkelstein, A., Spanoudakis, G., Ryan, M., 1996. Software package requirements & procurement. In: Proceedings of the International Workshop on Software Specification and Design (IWSSD). IEEE Computer Society Press, pp. 141–145.

Frakes, W.B., Kang, K., 2005. Software reuse research: status and future. IEEE Transactions on Software Engineering 31 (5), 529–536.

Gallardo-Valencia, R.E., Sim, S.E., 2009. Internet-scale code search. In: Proceedings of the Workshop on Search-Driven Development-Users, Infrastructure, Tools

Please cite this article in press as: Ayala, C., et al., Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. J. Syst. Software (2010), doi:10.1016/j.jss.2010.10.019

and Evaluation (ICSE-SUITE'09). IEEE Computer Society, Vancouver, Canada, pp. 49–52.

Gartner, 2008. The Evolving Open-source Software Model. Predicts from December 2008.

Glass, R.L., 2004. Matching methodology to problem domain. Communications of the ACM 47 (5), 19–21.

Golden, B., 2004. Succeeding with Open Source. Addison-Wesley Professional.

Jadhav, A.S., Sonar, R.M., 2009. Evaluating and selecting software packages: a review. Information and Software Technology 51 (3), 555–563.

Jansen, S., Brinkkemper, S., Hunink, I., Demir, C., 2008. Pragmatic and opportunistic reuse in two innovative start-up companies. IEEE Software, Special Issue on Opportunistic Software Systems Development 25 (6), 42–49.

Jensen, R.W., 2003. Lessons Learned From Another Failed Software Contract. Crosstalk, The Journal of Defense Software Engineering, September 2003.

Kano, N., Nobuhiku, S., Fumio, T., Shinichi, T., 1984. Attractive quality and must-be quality. Journal of the Japanese Society for Quality Control 14 (2), 39–48.

Keil, M., Tiwana, A., 2005. Beyond cost: the drivers of COTS application value. IEEE Software 22 (3), 64–69.

Kitchenham, B.A., Dyba, T., Jorgensen, M., 2004. Evidence-based software engineering. In: Finkelstein, A., Estublier, J., Rosenblum, D. (Eds.), Proceedings of the 26th International Conference on Software Engineering (ICSE 2004). IEEE Computer Society, Edinburgh, Scotland, 23rd–28th May, pp. 273–281.

Krippendorff, A., 1980. Content Analysis. Sage Publications, London.

Land, R., Blankers, L., Chaudron, M., Crnkovic, I., 2008. COTS selection best practices in literature and in industry. In: Proceedings of the 10th International Conference on Software Reuse (ICSR'08). Springer, Beijing, China, pp. 100–111.

Land, R., Sundmark, D., Lüders, F., Krasteva, I., Causevic, A., 2009. Reuse with software components—a survey of industrial state of practice. In: Proceedings of the International Conference on Software Reuse (ICSR'09), pp. 150–159.

Li, J., Conradi, R., Slyngstad, O.P.N., Torchiano, M., Morisio, M., Bunse, C., 2008. A state-of-the-practice survey of risk management in development with off-the-shelf software components. IEEE Transactions on Software Engineering 34 (2), 271–286.

Li, J., Conradi, R., Bunse, C., Torchiano, M., Slyngstad, O.P.N., Morisio, M., 2009. Development with off-the-shelf components: 10 facts. IEEE Software 26 (2), 80–87.

Lutters, W.G., Seaman, C., 2007. Revealing actual documentation usage in software maintenance. Information and Software Technology 49 (6), 576–587.

Mahmood, S., Lai, R., Kim, Y.S., 2007. Survey of component-based software development. IET Software 1 (2), 57–66.

Majchrowski, M., Deprez, J.C., 2008. An operational approach for selecting open source components in a software development project. In: Proceedings of the 15th European Conference on Software Process Improvement. Springer, pp. 176–188.

Merilinna, J., Matinlassi, M., 2006. State of the art and practice of open-source component integration. In: Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications. IEEE Computer Society, pp. 170–177.

Mohamed, A., Ruhe, G., Eberlein, A., 2007. COTS selection: past present and future. In: Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBSĩ07), pp. 103–114.

Morisio, M. (Ed.), 2006. Proceedings of the 9th International Conference on Software Reuse, (ICSR'06). Lecture Notes in Computer Science.

Morisio, M., Seaman, C.B., Basili, V.R., Parra, A.T., Kraft, S.E., Condon, S.E., 2002. COTS-based software development: processes and open issues. Journal of Systems and Software 61 (3), 189–199.

NCube, C., Oberndorf, P., Kark, A.W., 2008. Opportunistic software systems development: making systems from what's available. IEEE Software 25 (6), 38–41.

Oates, B.J., 2006. Researching Information Systems and Computing. Sage Publications, London.

Openbrr, 2005. Business Readiness Rating for Open Source A Proposed Open Standard to Facilitate Assessment and Adoption of Open Source Software, http://www.openbrr.org/wiki/images/d/da/BRR_whitepaper_2005RFC1.pdf, Request For Comments, 2005.

Patton, M.Q., 2002. Qualitative Research & Evaluation Methods, 3rd ed. Sage Publications, California.

Robson, C., 2002. Real World Research: A Resource for Social Scientists and Practitioner-researchers, 2nd ed. Blackwell Publishers Inc.

Seacord, C., Hissam, A., Wallnau, K., 1998. Agora: a search engine for software components. Internet Computing 2 (6), 62–70.

Seaman, C., 1999. Methods in empirical studies of software engineering. IEEE Transactions on Software Engineering 25 (4), 557–572.

Semeteys, R., Pilot, O., Baudrillard, L., Le Bouder, G., Pinkhardt, W., 2006. Method for Qualification and Selection of Open Source software (QSOS) version 1.6, Technical report, Atos Origin.

Simmons, G.L., Dillon, T.S., 2006. Towards an ontology for open source software development. In: Damiani, E., Fitzgeralg, B., Scacchi, W., Scotto, M., Succi, G. (Eds.), IFIP International Federation for Information Processing, Volume 203 Open Source Systems. Springer, Boston, pp. 65–75.

Torchiano, M., Morisio, M., 2004. Overlooked aspects of COTS-based development. IEEE Software 21 (2), 88–93.

Ulkuniemi, P., Seppänen, V., 2004. COTS component acquisition in an emerging market. IEEE Software 21 (6), 76–82.

Umarji, M., Elliott-Sim, S., Lopes, C., 2008. Archetypal Internet-Scale Source Code Searching. In Open Source Development Communities and Quality Working Group 2. 3 on Open Source Software. IFIP International Federation for Information Processing. Springer, pp. 257–263.

Wang, Z., Xu, X., Zhan, D., 2005. A survey of business component identification methods and related techniques. International Journal of Information Technology 2 (4), 230–238.

Wanyama, T., Far, B.H., 2006. Repositories for COTS selection. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE'06), pp. 2416–2419.

Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A., 2000. Experimentation in Software Engineering—An Introduction. Kluwer Academic Publishers.

Yanes, N., Sassi, S.B., Jilani, L., 2006. MoReCOTS: a specialized search engine for COTS components on the web. In: Proceedings of the International Conference on COTS-Based Software Systems (ICCBSS'06). IEEE Computer Society, pp. 109–115.

Claudia Ayala received the PhD degree in Informatics from the Technical University of Catalunya (UPC), Barcelona, Spain, where she is currently a lecturer. She was a postdoctoral ERCIM research fellow at the Norwegian University of Science and Technology (NTNU), Norway. Her current research interests include empirical software engineering, OTS-based development, requirements engineering and software quality.

Oyvind Hauge received his PhD degree in computer science from the Norwegian University of Science and Technology. He is currently a consultant at CapGemini, Trondheim. His research interests are empirical software engineering, OTS-development and OSS adoption.

Reidar Conradi received the MS and PhD degrees from the Norwegian University of Science and Technology (NTNU), Trondheim, in 1970 and 1976, respectively. He has been at NTNU since 1975. He is now a professor in the Department of Computer and Information Science (IDI), NTNU. He was a visiting scientist at the Fraunhofer Center for Experimental Software Engineering in Maryland and at the Politecnico di Milano in 1999 and 2000. His current research interests lie in software quality, software process improvement, version models, software evolution, open source software, and associated empirical studies. He is a member of the IEEE.

Xavier Franch received his PhD in informatics from the Technical University of Catalunya (UPC), where he is an associate professor. His main interests include requirements engineering, COTS-based development, and software quality.

Jingyue Li received the PhD degree in computer science from the Norwegian University of Science and Technology (NTNU) in 2006. He is presently a research scientist at NTNU. He worked at IBM China Ltd., before he came to NTNU. His research interests include software engineering, OTS-based development, project risk management, and aspect-oriented programming.

Please cite this article in press as: Ayala, C., et al., Selection of third party software in Off-The-Shelf-based software development—An interview study with industrial practitioners. J. Syst. Software (2010), doi:10.1016/j.jss.2010.10.019