Martínez-Fernández *et al. Journal of Software Engineering Research and Developments* 2014, **2**:7
www.jserd.com/content/2/1/7

Journal of Software Engineering Research and Development
a SpringerOpen Journal

RESEARCH Open Access

Towards guidelines for building a business case and gathering evidence of software reference architectures in industry

Silverio Martínez-Fernández[1]*, Claudia P Ayala[1], Xavier Franch[1], Helena Martins Marques[2] and David Ameller[1]

*Correspondence:
smartinez@essi.upc.edu
[1] GESSI Research Group, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, 08034 Barcelona, Spain
Full list of author information is available at the end of the article

Abstract

Background: Software reference architectures are becoming widely adopted by organizations that need to support the design and maintenance of software applications of a shared domain. For organizations that plan to adopt this architecture-centric approach, it becomes fundamental to know the return on investment and to understand how software reference architectures are designed, maintained, and used. Unfortunately, there is little evidence-based support to help organizations with these challenges.

Methods: We have conducted action research in an industry-academia collaboration between the GESSI research group and *everis*, a multinational IT consulting firm based in Spain.

Results: The results from such collaboration are being packaged in order to create guidelines that could be used in similar contexts as the one of *everis*. The main result of this paper is the construction of empirically-grounded guidelines that support organizations to decide on the adoption of software reference architectures and to gather evidence to improve RA-related practices.

Conclusions: The created guidelines could be used by other organizations outside of our industry-academia collaboration. With this goal in mind, we describe the guidelines in detail for their use.

Keywords: Software architecture; Reference architecture; Empirical software engineering; Business case; Cost-benefit analysis; Reference architecture context; Industry-academia collaboration

1 Background

Nowadays, the size and complexity of information systems, together with critical time-to-market needs, demand new software engineering approaches to design Software Architectures (SA) (Nakagawa et al. 2011). One of these approaches is the use of software Reference Architectures (RA) that allows to systematically reuse knowledge and components when developing a concrete SA (Cloutier et al. 2010; Galster and Avgeriou 2011). Hence, RAs provide guidance for the development and evolution of a class of software systems in a cost-effective manner (Martínez-Fernández et al. 2013b).

© 2014 Martínez-Fernández et al.; licensee Springer. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

As defined by Bass et al. (2003), a Reference Model (RM) is “a division of functionality together with data flow between the pieces” and an RA is “a reference model mapped onto software elements and the data flows between them”. It is important to denote that this RA definition already emphasizes two fundamental assets of an RA: the RM that defines its fundamental properties, and the software elements that makes the RM operational by its implementation (Bass et al. 2003; Galster and Avgeriou 2011).

A more detailed definition of RAs is given by Nakagawa et al. (2011). They define an RA as an architecture that encompasses the knowledge about how to design SAs of systems of a given application or technological domain. Therefore, an RA must address: business rules; architectural styles that address quality attributes; best practices of software development such as architectural decisions, domain constraints, legislation, and standards; and the software elements that support development of systems for that domain. All of this must be supported by a unified, unambiguous, and widely understood domain terminology (Nakagawa et al. 2011).

In this paper, we consider that an RA may have the elements that these two RA definitions state. We show the relationships among RM, RM-based RA and RA-based concrete SA in Figure 1. Throughout the paper, we use the term RM to refer to industry-specific RM, although RMs can also be defined by other agents such as research centers or nonprofit organizations. Also, we use the term RA to refer to RM-based RA and SA to refer to RA-based concrete SA. Angelov et al. have identified the generic nature of RAs as the main feature that distinguishes them from concrete SAs (Angelov et al. 2012). Every application has its own and unique SA, which is derived from an RA. This is possible because RAs are abstract enough to allow their usage in differing contexts (Angelov et al. 2012).

The motives behind RAs are: to facilitate reuse, and thereby harvest potential savings through reduced cycle times, cost, risk and increased quality (Cloutier et al. 2010); to help with the evolution of a set of systems that stem from the same RA (Galster and Avgeriou 2011); and to ensure standardization and interoperability (Angelov et al. 2012). Due to this, RAs are becoming a key asset of organizations (Cloutier et al. 2010).

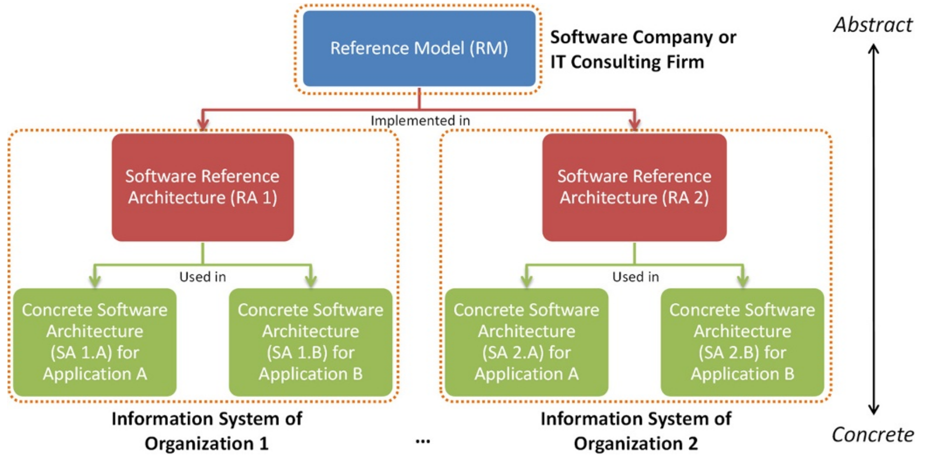
Reference Model (RM)
Software Company or IT Consulting Firm
Abstract
Implemented in
Software Reference Architecture (RA 1)
Software Reference Architecture (RA 2)
Used in
Used in
Concrete Software Architecture (SA 1.A) for Application A
Concrete Software Architecture (SA 1.B) for Application B
Concrete Software Architecture (SA 2.A) for Application A
Concrete Software Architecture (SA 2.B) for Application B
Information System of Organization 1
...
Information System of Organization 2
Concrete

Figure 1 Relationships among RM, RA and SA. A reference model (RM) could be the baseline of many software Reference Architectures (RA). Likewise, RAs serve as a reference for the concrete Software Architecture (SA) of the applications of an information system. These three artifacts go from a high level of abstraction to a low level of abstraction.

1.1 Research problem

Although the adoption of an RA might have plenty of benefits for an organization, it also implies several challenges, such as the need for an initial investment (Martínez-Fernández et al. 2013b) and getting real evidence for driving its design and use (Angelov et al. 2013). Currently, organizations have little support for dealing with these two challenges.

First, there is a shortage of economic models to precisely evaluate the benefit of RAs in order to make informed decisions about their adoption in an organization (Martínez-Fernández et al. 2013b). Organizations with a wide portfolio of applications, which may consider adopting an existing or new RA to create and maintain such applications, lack an approach to know whether it is worth for them to invest on the adoption of an RA. This situation could be addressed by making a business case with the help of an economic model that perform cost-benefit analysis about the adoption of an RA (Reifer 2002).

Second, there is also a shortage of experience reports disseminating the context of RAs in industry. For instance, a recent literature review about evidence in software architecture, in which only two papers were about RAs, shows that there is limited knowledge about RAs (Qureshi et al. 2013). As a result, practitioners usually find the current literature about RA scarce and abstract (Angelov et al. 2012), limiting the industrial uptake of research results in the field. In this context, we argue that in order to enable practitioners to fully exploit the benefits of RA adoption and usage, the research community must clarify the context of RA in practice. We propose conducting empirical studies to accumulate real evidence for understanding the context of RAs from essential types of stakeholders. Such evidence aims to help practitioners to better understand RA-related practices and, then, to identify the current challenges to improve these practices in their organization. Throughout the paper, we use the term "RA-related practices" to refer to common practices in RA projects, such as defining the goals of an RA, RA design, RA review, RA use and so on.

1.2 Research goals and contribution

In this context, the goal of this research is to support organizations (i.e., software companies, information technology consulting firms) to deal with the following Research Questions (RQ):

- RQ 1: Is it worth for an organization to invest on the adoption of an RA?
- RQ 2: How can an organization get corporate evidence that is useful for RA-related practices?

In addition, we aim for a third research question:

- RQ 3: How the results from RQ1 and RQ2 could be articulated to provide prescriptive support utilities (i.e., guidelines and artifacts) to support organizations in such endeavors?

These support utilities aim to allow organizations to set up and carry out empirical studies aimed to extract evidence to support RQ1 and RQ2.

1.3 Paper structure

The paper is structured as follows. In Section 2, we explain the research methodology used. In Section 3, we report the details of how RQ 1 was approached and its results. In

Section 4, we report the details of how RQ 2 was approached and its results. In Section 5, we report how the results from RQ 1 and RQ 2 have been packaged to answer the RQ 3, and we present the guidelines for building a business case and gathering evidence of RAs and their context of use. Finally, in Section 6, we end up with conclusions and future work.

The work presented in this paper is a follow-up of the work presented in (Martínez-Fernández et al. 2013a). It substantially extends such work by:

- Providing further details of the research methodology applied (see Section 2).
- Adding results obtained in RQ 1 (see Section 3): providing new results about the application of an economic model to build a business case of an RA (see new Section 3.2).
- Making more narrow the scope of RQ 2 (see Section 4), updating the set of criteria to understand and evaluate RAs (see Section 4.1), and providing new results about why RAs are used in organizations (see new Section 4.2.1).
- Providing details of the formative and summative stages of the guidelines obtained from the RQ 1 and RQ 2 results (see Section 5). To analyze if the proposed guidelines could be used by other organizations, we also add the context of RA projects as reported by other researchers and practitioners (see new Section 5.1.1).

2 Methods

This section explains the research methodology used, and the context of this research. The research is being done under an industry-academia collaboration (Martínez-Fernández and Marques 2014) between the GESSI research group and *everis*, a multinational IT consulting firm based in Spain.

The architecture group of *everis* faced the two problems stated in Section 1.1. The main motivation of *everis* for conducting this research is twofold:

- strategic/organizational: providing quantitative evidence to its clients about the potential economic benefits of applying an RA;
- technical: identifying strengths and weaknesses of RA-related practices in RA projects in order to disseminate them and, if necessary, to improve them.

2.1 Action research

This research consists of an ongoing action research initiative among the GESSI research group and *everis*. Action research is "learning by doing" - a group of people identify a problem, do something to resolve it, see how successful their efforts are, and if not satisfied, try again (Brydon-Miller et al. 2003). The action research cycle consists of: 1) diagnosis of a problem, 2) examination of options to solve the problem, 3) selection of options and execution, 4) analysis of the results, and, 5) repetition for improvement.

However, the idea behind this research is not only to help *everis* to cope with RQ 1 and RQ 2. With the goal of RQ 3 in mind, we also aim to package the solution in such a way that it can be easily used by other organizations dealing with similar problems as *everis*. The guidelines that will be proposed in this paper are shaped throughout our involvement with *everis* to cope with the RQ 1 and the RQ 2.

2.2 Context of RAs in everis

We analyze the context of RA projects from our experience with *everis*. We focus on the case in which *everis* has designed an RA with the purpose of deriving concrete SAs for each application of a client organization. This usually happens when *everis* is regularly contracted to create or maintain information systems in client organizations. Each information system of the client organization is built upon the RA and includes many software applications (see Figure 2). RAs enable reuse of architectural knowledge and software components (normally associated to particular technologies) for the design of concrete SAs in client organizations. Therefore, RAs provide a baseline that facilitates standardization and interoperability as well as the attainment of business goals during applications' development and maintenance.

Besides, a special characteristic of *everis* is its previous experience in multiple RA projects. This experience allows *everis* to build a more abstract industry RM. This RM includes best practices from previous successful experiences, which serve as a reference for new RAs that inherit a certain level of quality.

The context of *everis* is very similar to other IT consulting firms. As a recent Gartner report shows, IT consulting firms "leverages industry-specific or industry reference models to accelerate client delivery and ensure quality and consistency across client engagements" (Brand 2014). However, "clients must ensure that generic industry or reference models [...] are sufficiently customized and tailored to enable their unique business

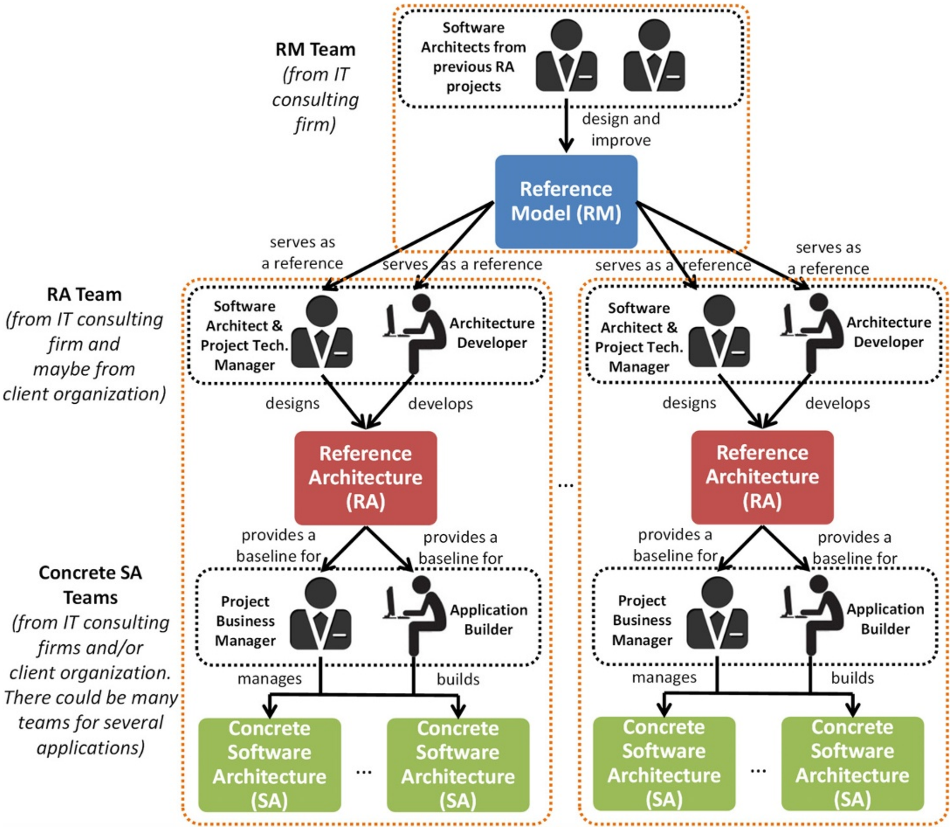
RM Team
(from IT consulting firm)
Software Architects from previous RA projects
design and improve
Reference Model (RM)
serves as a reference
serves as a reference
serves as a reference
serves as a reference
RA Team
(from IT consulting firm and maybe from client organization)
Software Architect & Project Tech. Manager
Architecture Developer
designs
develops
Reference Architecture (RA)
...
provides a baseline for
provides a baseline for
Concrete SA Teams
(from IT consulting firms and/or client organization. There could be many teams for several applications)
Project Business Manager
Application Builder
manages
builds
Concrete Software Architecture (SA)
...
Concrete Software Architecture (SA)
Software Architect & Project Tech. Manager
Architecture Developer
designs
develops
Reference Architecture (RA)
provides a baseline for
provides a baseline for
Project Business Manager
Application Builder
manages
builds
Concrete Software Architecture (SA)
...
Concrete Software Architecture (SA)

Figure 2 Relevant stakeholders for RA analysis. The figure explains which types of stakeholders and their roles in each type of project (RM, RA and SA projects).

capabilities and environments" (Brand 2014), so that the RM does not stifle competitive advantage of the RA.

Next subsections respectively show the types of projects that can be found on the *everis* context and the stakeholders that participate in each project.

Types of projects There are three types of projects with different targets (Figure 2).

1. RM projects.
2. RA projects.
3. SA projects.

Stakeholders for RA analysis Stakeholders need to be clearly defined for RA assessment purposes (Angelov et al. 2008). The people involved in an RA assessment are the evaluation team, which conducts the empirical studies of the guidelines, and stakeholders from architectural projects. In the three types of projects defined above performed by IT consulting firms, we consider the following five stakeholders essential for RA assessment: project business manager, project technological manager, software architect, architecture developer, and application builder. Each of these stakeholders has a vested interest in different architectural aspects, which are important to analyze and reason about the appropriateness and the quality of the three types of projects (Gallagher 2000). However, there could be more people involved in an architectural evaluation, as Clements et al. indicate in (Clements et al. 2001). As a consequence, although this context is generic for IT consulting firms, projects' stakeholders may vary between firms. Below, we describe to which type of project essential stakeholders belong and their interests.

RM project It is composed of software architects from the IT consulting firm that worked in previous successful RA projects. They are specialized in architectural knowledge management. Their goal is to gather the best practices from previous RA projects' experiences in order to design and/or improve the corporate RM.

RA projects RA projects involve people from the IT consulting firm and likely from the client organization. Their members (project technological managers, software architects and architecture developers) are specialized in architectural design and have a medium knowledge of the organization business domain.

Project technological managers from the IT consulting firm are responsible for meeting schedule and interface with the project business managers from the client organization.

Software architects (also called as RA managers) usually come from the IT consulting firm, although it may happen that the client organization has software architects in which organization's managers rely on. In the latter case, software architects from both sides cooperatively work to figure out a solution to accomplish the desired quality attributes and architecturally-significant requirements.

Architecture developers come from the IT consulting firm and are responsible for coding, maintaining, integrating, testing and documenting RA software components.

SA projects Enterprise application projects can involve people from the client organization and/or subcontracted IT consulting firms (which may even be different than the RM owner) whose members are usually very familiar with the specific organization domain. The participation of the client organization in RA and SA projects is one possible strategy for ensuring the continuity of their information systems without having much dependency on subcontracted IT consulting firms.

Project business managers (i.e., customer) come from client organizations. They have the power to speak authoritatively for the project, and to manage resources. Their aim is to provide their organization with useful applications that meet the market expectations on time.

Application builders take the RA reusable components and instantiate them to build an application.

3 Building the business case of RAs in everis

This section details our action research initiative with regard to RQ 1, which is intended to quantify the benefits and costs of RAs through a business case. "A business case is a tool that helps you make business decisions by predicting how they will affect your organization. Initially, the decision will be a go/no-go for pursuing a new business opportunity or approach" (Northrop and Clements 2014). A useful economic function for business cases is the Return On Investment (ROI). The ROI is a "measure of how much profit an investment earns computed by dividing net income by the assets used to generate it" (Reifer 2002). Although the ROI is the most popular function in business cases, there are many others (Ali et al. 2009). For instance, Net Present Value (NPV) estimation with discounted cash flow is mostly used to address the time value of money, and the Internal Rate of Return (IRR) compares the profitability of investments. Sarmad Ali et al. summarize the economic functions used for software product lines (Ali et al. 2009).

Figure 3 shows the action research cycles conducted in order to answer RQ 1.

In the first cycle of the RQ 1, we diagnosed the problem of the lack of approaches to justify the investment on RAs to *everis*' clients in monetary terms. As a consequence, we

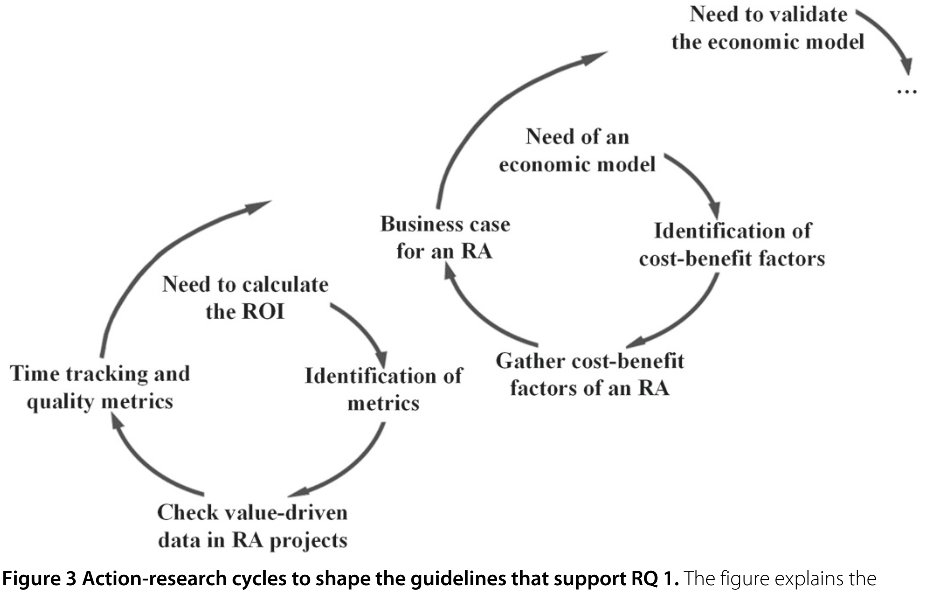
Need to validate the economic model
...
Need of an economic model
Business case for an RA
Identification of cost-benefit factors
Need to calculate the ROI
Time tracking and quality metrics
Identification of metrics
Gather cost-benefit factors of an RA
Check value-driven data in RA projects

Figure 3 Action-research cycles to shape the guidelines that support RQ 1. The figure explains the formative stage to create the checklist of value-driven data in RA projects and the REARM economic model.

designed online questionnaires to ask stakeholders the metrics available in RA projects and conducted them in several RA projects. As a result, we observed that effort metrics could be derived from time tracking practices and that cost-benefit factors could be computed by using reuse-based metrics from the source code of the RA projects. The results from the online questionnaires to study the metrics available in RA projects are reported in Section 3.1.

In the beginning of the second cycle of action research, we diagnosed the need of having an economic model that would use the available data. Then, we identified economic functions meaningful to *everis* to justify RA investments. We also identified cost-benefits factors from the literature that can be calculated from the available metrics. Later, we computed these cost-benefit factors for a real RA project. At the end of this second cycle, we could build the business case for that RA project. In Section 3.2, we report an application of REARM, which is the economic model that we created.

Currently, we are at the beginning of a third cycle in which we diagnosed the problem of validating the economic model in another RA project.

3.1 First cycle: survey to check existing value-driven data in RA projects

In this survey, a sample of 5 *everis*' RA projects and 5 SA projects were selected on the basis of their suitability and feasibility to contact at least with one person that participated in the projects.

The main perceived economic benefits on the use of RAs are the cost savings in the development and maintenance of systems due to the reuse of software elements, and the adoption of best practices of software development that increase the productivity of developers (Martínez-Fernández et al. 2013c). To quantify these cost savings, we used online questionnaires to ask project technical managers and application builders about existing information from past projects. An excerpt of the questions of this survey is shown in Table 1, whereas Additional file 1 includes all the questions. When the client organization has no experience in RAs, these data need to be estimated, which could be potentially error-prone.

3.1.1 Results: costs and benefits metrics for RAs

In this section we describe the information that was available in order to calculate the costs and benefits of adopting an RA. We divide existing information in two categories: effort and software metrics. First, the invested effort from the tracked activities allows the calculation of the costs of the project. Second, software metrics help to analyze the benefits that can be found in the source code.

Effort metrics to calculate projects' costs In RA projects, 4 out of 5 client organizations tracked development efforts, while maintenance effort was tracked in all 5. In SA projects, 4 out of 5 client organizations tracked development and maintenance effort.

The development effort is the total amount of hours invested in the development of the RA and the SAs of applications. It could be extracted from the spent time for each development's activity of the projects. The maintenance effort is the total amount of hours invested in the maintenance of the RA and the SAs of applications. Maintenance activities include changes, incidences, support and queries.

Table 1 An excerpt of the template survey for software architects to check existing value-driven data in RA projects

Type of question	Question
Questions about data available for the RA.	Is it possible to access to a code quality management tool (e.g., Sonar) used in the RA project? If not, is it possible to install Sonar to take metrics from the source code of the RA?
	Would it be possible to estimate the degree of reuse in each of the modules of the RA with respect to the RM? (optional)
	Did the RA stakeholders track the time spent on each task in the development of the RA? If so, with which granularity?
	Is there data about personnel and time invested in maintaining the RA? If so, with which granularity?
	Would it be possible to give an estimate of the additional training time that an application builder needs to use the RA?
	Would it be possible to specify a standard hourly rate of performing tasks on the RA?
Questions about data available for the applications based on the RA.	How many applications are (or will be) based on the RA? Make a list of the applications with a contact person.
	Do you have an overview of the development of applications based on the RA? If so, go on. If not, we will contact the person who you indicated in the previous answer.
	For which of the above applications is it possible to know which RA modules have been reused and the degree of reuse?
	Is additional effort needed to reuse RA modules?
	Can you estimate how long it would take to develop RA modules instead of reusing them?
	Is it possible to access to a code quality management tool (e.g., Sonar) used for the RA-based applications? If not, is it possible to install Sonar to take metrics from the source code of the applications?
	Please, indicate the generic characteristics (e.g., reuse percentage of RA modules, size of applications) of three ideal types of RA-based applications with low, medium and high complexity.
	Did the RA stakeholders track the time spent on each task in the development of the RA-based applications? If so, with which granularity?
	Is there data about personnel and time invested in maintaining the RA-based applications? If so, with which granularity?
	Would it be possible to specify a standard hourly rate of performing tasks on the RA-based applications?
	Have you done any comparison between the costs and the benefits between RA-based applications and ad-hoc applications?
	Currently, are there indicators or metrics to evaluate the improvement of the quality attributes in the applications because of the RA usage?
Questions for adding comments and propose metrics to calculate the ROI of the RA.	In addition to the information discussed above, do you think that there is other available information to evaluate how the RA affects the applications development?
	Do you think that other metric, not mentioned above, could be useful to calculate the ROI of building applications based on the RA?
	Our economic model to calculate the ROI of an RA is based on the reuse and maintenance of code. Do you think there are other quality attributes or important factors for evaluating an RA?
	Before sending the survey, would you like to add any comments that may help to understand the context of your answers?

Software metrics to calculate benefits in reuse and maintainability Source code in RA and SA projects was obviously available in all projects. However, due to confidentiality issues with client organizations, it is not always allowed to access source code.

The analysis of the code from RA and SA projects allow quantifying the size of these projects in terms of LOC or function points (e.g., number of methods). Having calculated the project costs as indicated above, we can calculate the average cost of a LOC or a function point. Since the cost of applications' development and maintenance is lower because of the reuse of RA modules, we can calculate the benefits of RA by estimating the benefits of reusing them. Poulin defines a model for measuring the benefits of software reuse (Poulin 1997). Maintenance savings due to a modular design could be calculated with design structured matrices (MacCormack et al. 2012). For a detailed explanation about how such metrics can be used in a cost-benefit analysis, the reader is referred to (Martínez-Fernández et al. 2013b).

3.1.2 Lessons learned

Improvements in the quality attributes of an RA (e.g., reuse, maintainability, security) are extremely difficult to evaluate in an analytic and quantitative fashion contrary to the efficacy of the business (e.g., sales) (Carriere et al. 2010). This is because software development is a naturally low-validity environment and reliable expert intuition can only be acquired in a high-validity environment (Erdogmus and Favaro 2012). In order to evaluate RAs based on an economics-driven approach, software development needs to move to a high-validity environment. The good news is that it could be done with the help of good practices like time tracking, continuous feedback, test-driven development and continuous integration. In order to get the metrics defined above, tools such as JIRA (Atlassian 2014) and Redmine (Redmine 2014) allow managing the tasks and their invested time, general software metrics (like LOC) and percentages of tests and rules compliance can be calculated by Sonar (SonarSource 2014) and (Jenkins 2014). We think that adopting good and repeatable practices to collect data is the basis for moving software development to a high-validity environment and consequently being able of performing an accurate cost-benefit analysis.

3.2 Second cycle: a case study to apply an economic model to calculate the ROI of adopting an RA

In this case study, a sample of 1 *everis*' client organization RA project in a public administration and 1 SA project was selected. These projects were selected because the public administration that adopted the RA was interested in the study results. Besides, by the time we conducted the study, *everis*' started the aforementioned SA project, being highly feasible to collect quantitative data. Although we were aware of other SA projects with participants that do not belong to *everis*, it was not possible to contact with them.

Results from the previous survey (see Section 3.1) revealed that the data available in order to calculate costs and benefits were effort and software metrics. We collected these metrics, which are presented in (Martínez-Fernández et al. 2013b), from two types of tools. In order to collect data, JIRA (Atlassian 2014) was used to collect the invested effort from training, development and maintenance activities. Keeping track of activities is common in practice for project management and auditing. In addition to that, Sonar (SonarSource 2014) was used to gather software metrics to analyze the benefits that can be found in the source code. Sonar offers tool support for obtaining general software metrics such as LOC, dependencies between modules, technical debt, and percentages of tests and rules compliance.

3.2.1 Results

After retrospectively collecting the data of the past RA and SA projects, we analyzed which cost-benefit factors of adopting an RA could be calculated and envisaged the REARM economic model (Martínez-Fernández et al. 2013b) in order to calculate the ROI of the RA investment. By the time we conducted the study, the public organization had already: (1) adopted an RA designed by *everis*, (2) created an application using the RA, and (3) fixed errors discovered in the RA software elements that were reused by the application.

We calculated a three-year ROI of 42% with a payback period of 16,5 months and 7 applications in the public organization (see Figure 4).

Moreover, applications are introduced into the market earlier from the seventh month on. This is due to the effort avoidance of reusing the RA in the development of new applications.

The results of this study are further reported in (Martínez-Fernández et al. 2013b).

3.2.2 Lessons learned

As Clement et al. point out in their book (Clements and Northrop 2002), experts agree that the number of applications you need to build for a reuse-based architecture-centric approach to pay off is between 2 and 3. We also agree with that statement. In our study it turned out to be 7 because the application from which we collected data was small and only 20% of the RA was reused. However, in applications in which this percentage is higher, the benefit from the RA is greater. If we perform sensitivity analysis to REARM, with a higher reuse of the RA in applications (for instance higher than 70%, which is likely in medium to large applications), the RA pays off with 2 applications.

4 Gathering evidence of RAs in everis

This section details our action research initiative with regard to the RQ 2. Figure 5 shows the action research cycles conducted in order to answer RQ 2.

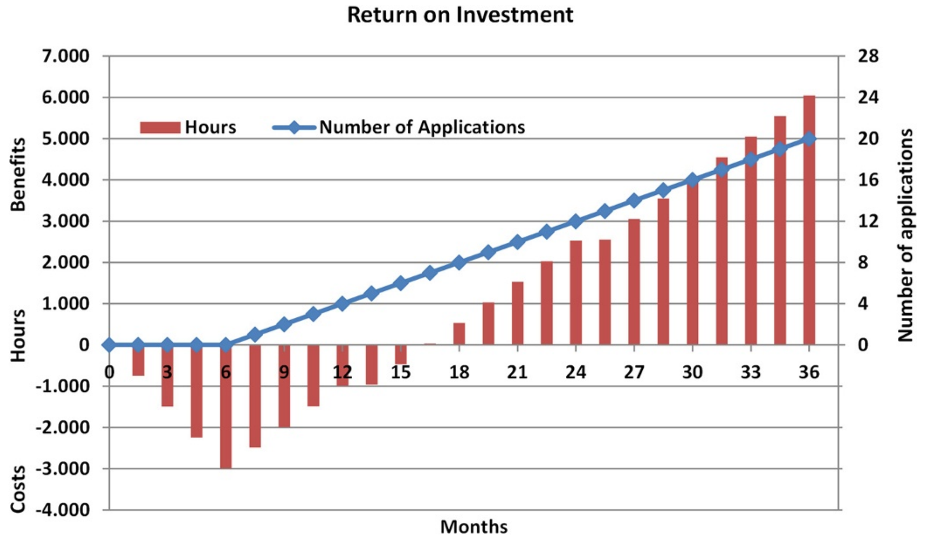

Figure 4 ROI of developing and maintaining RA-based applications versus stand-alone fashion in a public administration. The public administration of this study will realize a three-year ROI of 42% with a payback period of 16,5 months and 7 applications.

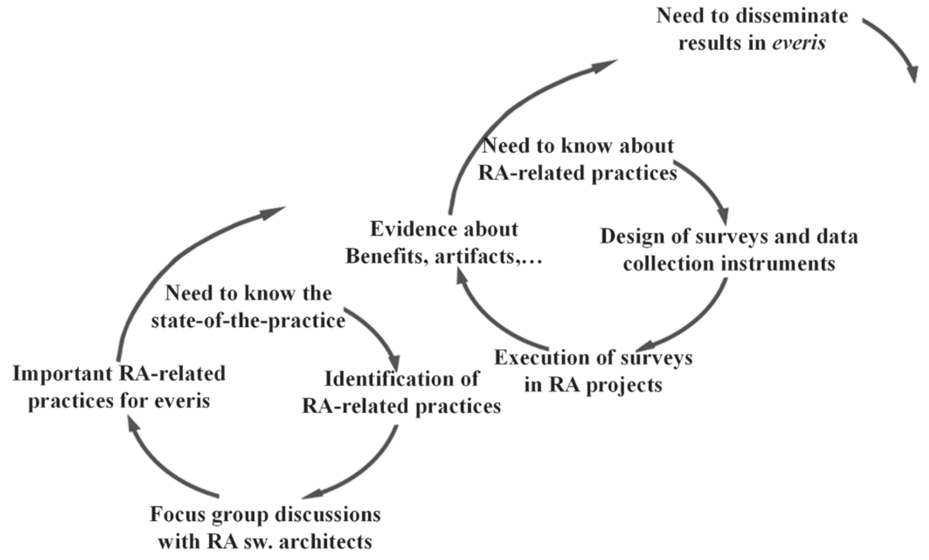
Need to disseminate results in *everis*
Need to know about RA-related practices
Evidence about Benefits, artifacts,…
Design of surveys and data collection instruments
Need to know the state-of-the-practice
Execution of surveys in RA projects
Important RA-related practices for everis
Identification of RA-related practices
Focus group discussions with RA sw. architects

Figure 5 Action-research cycles to shape the guidelines that support RQ 2. The figure explains the formative stage to create the template survey that aims to gather relevant evidence from RA projects.

In the first cycle of the RQ 2, we diagnosed the need of knowing about the state of past and present RA projects in *everis* in order to reuse architectural knowledge and improve RA-related practices in prospective RA projects. As a consequence, we planned to identify a set of criteria about RAs that are relevant for practitioners. Finally, we identified five aspects that indicate what evidence to gather in order to support RA-related practices, which are presented in Section 4.1.

In the second cycle of this action research, we planned to gather evidence about these aspects. We designed surveys to gather mostly qualitative evidence about such aspects. Then, we executed these surveys in a case study with several RA projects in *everis*. As a consequence, we obtained results about why *everis*' clients adopted RAs, and the benefits and drawbacks of RAs. These results are reported in Section 4.2.

Currently, we are at the beginning of a third cycle of the RQ 2 in which we diagnosed the problem of disseminating these results in *everis*.

4.1 First cycle: identifying practical review criteria for RAs

In order to gather evidence of RAs, it becomes necessary to previously identify the aspects that are relevant for practitioners about RAs. In this section we identify an initial set of criteria, which might be further refined after gathering evidence from RAs. To do so, we analyzed the key criteria mentioned by *everis*' software architects during the meetings and focus group discussions of our action research initiative, and we also studied the literature. We prioritized the *everis*' vision to make a more practitioner-oriented set of criteria.

Although a commonly accepted set of criteria to evaluate RAs does not exist (Angelov et al. 2008; Gallagher 2000; Graaf et al. 2005), it has been claimed that RAs have to be evaluated for the same aspects as SAs (Angelov et al. 2008). For this reason, we started by analyzing some available works on SA evaluation (Bass et al. 2008; Clements et al. 2001; Falessi et al. 2010). However, existing evaluation methods for SAs are not directly applicable to RAs or other architecture-centric approaches such as product lines architectures because they do not cover their generic nature (Angelov et al. 2008). The development of a family of software systems has some characteristics that distinguish it from the development of software systems (Montagud et al. 2011). Therefore, existing evidences for product line architectures can be also used to evaluate RAs, namely: generic characteristics such as "variability, reusability, commonality, and compositionality" (Montagud et al. 2011); the propagation of architectural decisions while reusing common assets (Montagud et al. 2011); and lower development costs with respect to developing systems individually (Ali et al. 2009; Linden et al. 2004; Montagud et al. 2011).

Due to these reasons, the authors of the paper and two more software architects from *everis* elaborated further this analysis considering the specific characteristics of RAs as described in (Angelov et al. 2008, 2012; Galster and Avgeriou 2011; Graaf et al. 2005; Nakagawa et al. 2011), commonalities with other architecture-centric approaches such as product line architectures (Ali et al. 2009; Deelstra et al. 2005; Linden et al. 2004; Montagud et al. 2011), and experience from *everis*. The resulting aspects for understanding and evaluating RAs are detailed below and summarized in Table 2.

Aspect 1 refers to the need of determining the context and classifying an RA. As Angelov et al. point out (Angelov et al. 2012), there are five types of RAs depending on their characteristics. Among the most important characteristics are (Angelov et al. 2012):

- the organization(s) that will use the RA (e.g., a *single organization* or *multiple organizations* that share a domain),
- who defines the RA (e.g., *software companies* as IT consulting firms, *software groups* from the organization that use the RA, and so on),
- the origin and motivation of the RA (e.g., *preliminary* when the RA solves a new problem or *classical* when it is based on previous experiences),
- the goal of the RA and the domain of the RA-based applications (e.g., *standardization* of SAs or *facilitation* of the design of SAs),
- and the RA elements it may include (e.g., *components and connectors*, *policies and guidelines*, and so on).

The classification of an RA is vital to better understand its limits, to ensure its congruency, and to facilitate its evaluation.

Table 2 Summary of relevant aspects for software reference architecture understanding and evaluation as seen by practitioners

Aspect	Description of the RA Aspect
1	Overview and classification of an RA
2	Requirements and quality attributes analysis
3	Architectural knowledge and decisions
4	Supportive technologies
5	Business qualities and architecture competence

Aspect 2 consists of the quality attributes targeted by an RA. The achievement of quality attributes is in fact an SA most compelling reason for existence (Bass et al. 2003). However, SAs and RAs do not strictly determine all of an application's qualities. One example is usability: "whether the user sees red or blue backgrounds, a radio button or a dialog box" (Clements et al. 2001) is not determined by an SA or an RA. A list of quality attributes that lie squarely in the realm of SAs is defined by the architecture trade-off analysis method (Clements et al. 2001): performance, reliability, availability, security, modifiability, portability, functionality, variability, subsetability and conceptual integrity. For instance, variability shows how well an RA could be expanded or modified to produce new SAs of applications. Besides, an RA could address more architectural qualities than an SA (e.g., variability, reusability, commonality, compositionality, and applicability) (Angelov et al. 2008; Montagud et al. 2011). Quality attributes analysis should be wider for RAs in this sense.

Aspect 3 comprises architectural decisions. Many prominent researchers (Angelov et al. 2008; Falessi et al. 2010) highlight the importance of architectural decisions for the SA design process and the architectural evaluation. For RAs, architectural decisions are even more important than in a single software system since, owing to systematic reuse, an inadequate design decision could be propagated to several software systems (Montagud et al. 2011).

Aspect 4 consists of the supportive technologies such as methods, techniques and tools (Falessi et al. 2010; Nakagawa et al. 2011) that aim to improve the RA design process and support the use of the RA during the development of applications. Moreover, this aspect is very important for practitioners, since they are interested in knowing the latest versions of technologies and tools used in the RA projects, and providing application builders with tools that improve their productivity.

Aspect 5 refers to business qualities of an RA. SAs also address business qualities (Angelov et al. 2008) (e.g., cost, time-to-market) that are business goals, i.e. the objectives of an organization that affect their competence (Bass et al. 2008). These business qualities are even more important in the context of families of applications, such as RAs: "the main arguments for introducing software product family engineering are to increase productivity, improve predictability, decrease the time to market, and increase quality (dependability)" (Linden et al. 2004).

We recommend gathering evidence about these five aspects, which are summarized in Table 2, in order to improve RA-related practices. Next, Section 4.2 explains how we are gathering evidence about these aspects. Currently, there are no guidelines to support the gathering of these aspects altogether. This has motivated our work.

4.2 Second cycle: surveys to gather evidence to improve RA-related practices

In these surveys, the target populations were RA projects and SA projects executed by *everis*. A sample of 9 representative *everis*' projects in client organizations were selected. All these projects were from Europe (7 from Spain).

In order to collect data, semi-structured interviews were used for project technological managers, software architects, and client's project business managers. The reason of using interviews is that these roles have higher knowledge than the other roles about the architectural aspects of the Table 2, or another perspective in the case of client's project business managers, so we wanted to collect as much information as possible from them. Prior to the interviews, questionnaires were delivered to collect personal information about the interviewee and to inform him/her about the interview. On the other hand, online questionnaires were used for architecture developers and application builders, since most of their questions are about supportive technologies and their responses can be previously listed, simplifying the data collection process.

To perform data analysis, the research team held several discussion meetings during and after data collection, and established specific protocols and templates for data analysis. In the case of the interviews, we used an Excel-based template to organize each participant's answer to each question using tables. For doing this, we used the interview transcripts and individual notes taken by different researchers during the interviews. Then, we analyzed the data from two perspectives that lead us to a better understanding and interpretation of the results. First, we analyzed the answers at the project level, in order to understand the specific context of each project and the perspective of its stakeholders. Second, we analyzed the answers at whole by assessing all participants' answers related to each question, in order to categorize their answers using template tables that described (in each column): the name of the category, a detailed description of the cases covered by the category, the participant, and the explicit sentences that support the category. Such categories were then further discussed and analyzed by the entire team in order to better analyze the evidence and improve our understanding until reaching an agreement. As a result, some categories were split, modified, discarded or added to ensure that all answers were well-represented.

The complete version of the surveys protocol, which contains interview guides and questionnaires, is available at the Additional file 2. An excerpt of the questions for software architects of this survey, which only includes the questions used for the results described in the next two subsections, is shown in Table 3.

4.2.1 Results: motivations to use RAs for designing SAs of applications

In this subsection, we present results about the Aspect 1, which was defined in Section 4.1. Such results answer why RAs were adapted for creating SAs of *everis* client organizations' applications. Next, we report the motives that trigger the origin behind each RA project. These results can help organizations to analyze if they need an RA. Table 4 shows the characteristics and objectives of each RA project.

The motivations why *everis*' client organizations adopted an RA are shown below. We report between brackets the identifier of the client organization that indicated such motivation (see Table 4).

- 5 out of 9 projects (B, C, D, F, H) reported the update of technologies to develop applications since they were obsolete or application maintenance was costly.
- 4 out of 9 projects (A, B, G, H) mentioned the need to homogenize the development of similar applications and identify their common elements to foster reuse.
- 4 out of 9 projects (C, D, F, G) aimed to simplify application development (e.g., use of widely-known technologies) and improve productivity of application builders in order to hire profiles less specialized and reduce development time.

- 2 out of 9 projects (C, I) needed to improve business processes of the organization because of organizational changes or applications misalignment with business needs.
- 2 out of 9 projects (C, E) were started because the client organization had difficulties in developing applications without the help of a software vendor.
- 2 out of 9 projects (A, B) mentioned the need to support and enable application development in any platform (e.g., web, smartphone, POS terminal, ATM).
- 2 out of 9 projects (B, D) stated the need to migrate functionality from legacy systems to new systems (also known as "downsizing") to reduce maintenance costs.
- 1 out of 9 projects (A) reported the lack of products on the market adapted to its needs and business processes.

4.2.2 Results: strengths and weaknesses of RAs

In this section we present preliminary results about the Aspect 5 defined in Section 4.1. Such results answer how the adoption of RAs provide observable benefits to the different involved actors in RA projects in *everis*. Below, the resulting benefits and aspects to consider are reported, and it is indicated between brackets the identifier of the client organization that mentioned such benefits (see Table 4).

Benefits in *everis*' RA projects We grouped benefits in four categories. Below each category, explicit sentences from software architects are also indicated.

Table 3 An excerpt of the template survey for software architects to gather evidence about Aspects 1 and Aspect 5 of RA projects

Type of question	Question
Questions about the classification of the RA.	What do you understand by RA?
	What was the objective of the project?
	Which was the client organization's problem that motivated the project?
	What was the relationship with the client organization during the design and maintenance of the RA?
Questions about the business qualities of the RA for the client organization.	What benefits does the client organization experience from adopting your RA? And developers from using it?
	Which problems does the client organization experience from adopting your RA? And developers from using it?
	How the training for the client organization was conducted in order to them use your RA?
	Did the use of your RA cause any organizational change in the client organization?
	How does the use of your RA reduce the time-to-market of RA-based applications in the client organization?
	What types of non-functional requirements are reinforced because of using your RA in applications?
	To sum up, what conclusions do you draw from the facilities provided by your RA for the client organization?
Questions about the business qualities of the RA for the IT consulting firm (optional).	How is your RA based on an RM and any other existing architectural knowledge and software components in your company?
	What do you think should be replaced, included or updated in prospective versions of the RM?
	To sum up, what conclusions do you draw from the facilities provided by the RM for the IT consulting firm?

- 7 out of 9 projects (A, B, C, E, F, G, I) stated "reduction of the development time and faster delivery of applications".
  - An RA allows starting developing applications since the first day by following architectural decisions already taken.
  - An RA decreases the development time of applications since the RA's modules that implement needed functionality are reused in the application.
- 7 out of 9 projects (A, B, C, D, E, H, I) mentioned "increased productivity of application builders".
  - An RA facilitates material and tools for the development, testing and documentation of applications, and for training application builders.
  - An RA generates or automatizes the creation of code in the applications.
  - An RA indicates the guidelines to be followed by the application builders.
  - An RA reduces the complexity of applications' developments because part of the functionality is already resolved in the RA.

Table 4 Overview and characteristics of selected *everis*' RA projects

Id.	Domain	RA objective	Examples of applications	RA project type
A	Industry	Web-based applications to allow vendors updating information about clients in a department store.	Application that allows vendors to manage client information.	Design
B	Banking	Multi-platform applications that are fast, satisfy practices of the market and support transaction processing.	Application for online banking. There are approximately 300 applications.	Design, Evolution
C	Banking	Multi-platform applications of a bank. To improve the productivity of applications builders and to facilitate the development of applications.	Application to integrate some existing applications with the same look and feel. Application to improve the business process of depositing money. There are approximately 10 applications.	Design, Evolution
D	Insurance	Applications that satisfy internal request for proposals.	Applications to hire insurance policy (home insurance, life insurance...).	Design, Evolution
E	Public sector	Java web applications, with flexible front-end, integration and batch processes.	Application to support billing and technical services.	Design, Evolution
F	Public sector	Web-based applications for the different departments of a public administration.	Application to deliver forms and pdfs. There are around 50 applications of different departments of a public administration.	Design, Evolution
G	Public sector	Applications with enhanced reuse and reduced development costs.	Application for enterprise resource planning and customer relationship management.	Design
H	Insurance	Applications integrated with services of an insurance company.	Application to manage health care and payments.	Design
I	Public sector	Applications that include the business processes of a utility organization.	Application for urban management reports.	Design

  - An RA facilitates the configuration of its modules and the integration with legacy systems or external systems.

- 6 out of 9 projects (A, B, C, D, E, G) stated "cost savings in the maintenance of the applications".

  - An RA increases the control over applications through their homogeneity.
  - An RA maintains only once reused services by all applications.
  - An RA allows adding or changing functionalities by means of a modular design.
  - An RA establishes long term support standards and "de facto" technologies.

- 4 out of 9 projects (C, F, G, I) mentioned "increased quality of the enterprise applications".

  - An RA helps to accomplish business needs by improving key quality attributes.
  - An RA helps to improve the business processes of an organization.
  - An RA reuses architectural knowledge of previous successful experiences.

Aspects to consider that eventually could become risks in *everis*' RA projects

- 5 out of 9 projects (B, C, E, F, H) considered "additional learning curve". An RA implies an additional training for their own tools and modules, even if its technologies are standard or "de facto" already known by the application builder.
- 4 out of 9 projects (B, C, D, E) stated "dependency on the RA". Applications depend on the reused modules of the RA. If it is necessary to make changes in a reused module of the RA or to add a new functionality, application builders have to wait for the architecture developers to include it in the RA for all the applications.
- 2 out of 9 projects (A, D) considered "limited flexibility of the applications". The use of an RA implies following its guidelines during the application development and adopting its architectural design. If business needs require a different type of application, the RA would limit the flexibility of that application.

The results of this study are further reported in (Martínez-Fernández et al. 2013c).

4.2.3 Lessons learned

During the pilot of the survey, we learnt the following lessons about its design:

- The same term could have slightly different meaning in the academia and in the industry (for instance, the term "enterprise architecture" is sometimes used in the industry to mean "RA for a single organization").
- The software development methodology used during the development of applications is seldom prescribed by the RA, so we removed it as relevant aspect for RAs (it was an aspect in previous versions).
- Contacting stakeholders from client organizations was harder than contacting interviewees from *everis.* This is mainly because it was *everis* who requested the study, so they had a clear interest on it.

5 Results and discussion

In this section, we present how the results from RQ1 and RQ2 have been articulated into guidelines to support organizations building a business case and gathering evidence of RAs. The guidelines have been incrementally constructed based on our action research approach with *everis*. For its shaping and validation, the construction of the guidelines is respectively divided in two stages: the formative and the summative stages.

5.1 Formative stage: shaping the guidelines

In the formative stage, the goal is to incrementally shape the guidelines from the feedback of the action research. In order to analyze whether other organizations deal with similar problems as *everis*, we highlight the similarities of RAs designed by *everis* with other RA contexts that have been reported in the literature and by practitioners. After this analysis, we will only package into the guidelines the material that could be used under other organizations context.

5.1.1 Similar contexts of RAs in practice

The results from our action research in *everis* are particular to the context described in Section 2.2. IT consulting firms, such as Accenture (Brand 2014) and Capgemini (Herold and Mair 2013) also fit into such context (i.e., they use industry-specific RM, and they carry out the three types of projects described there). However, to properly create the guidelines for other RA contexts, it is vital to first characterize RA projects conducted by other companies.

Architecture-centric approaches to develop families of software applications are not new. Deelstra et al. give a classification of these approaches with respect to the level of reuse (Deelstra et al. 2005): 1) standardized infrastructure; 2) platform: 3) software product line, and 4) configurable product family. In this classification, RAs can be positioned as standardized infrastructures or platforms (Graaf et al. 2005), whereas software product lines and configurable product families are based on Product Lines Architectures (PLA). Several authors has stated that RAs are more generic than PLAs (Martínez-Fernández et al. 2013b). Next, we classify RAs in the industry under the two former categories.

On the one hand, standardized infrastructures have been used by public administrations in Germany (Herold and Mair 2013), in the Netherlands (Galster et al. 2013), and in Spain (García-Alonso et al. 2010). These RAs provide software assets as inspiration for the design of applications, but little domain engineering effort is performed (i.e., little domain specific functionality is included in the RA). They are popular in public administration because there is a need to cover multiple organizations from different business domains (i.e., ministries or departments of the government) and little common functionality exist. Also, the high distribution of development teams implies that these RAs play only an informative or instructive role rather than regulative.

On the other hand, platforms additionally "require a certain amount of domain engineering to create and maintain the assets that implement the common functionality" (Deelstra et al. 2005). There are several business domains that have used this type of RAs.

In the space domain, the NASA has detected that "many Earth science data system components and architectural patterns are reconstructed for each mission" (Mattmann and Downs 2010). In order to reuse these assets in new systems, they created the NASA's ESDS RA (NASA 2012).

In the banking domain, RAs are usually used to integrate legacy systems and migrated service-oriented systems that contain the business logic. The common scenario is that these RAs provide common services that then may be reused in the different front-ends or channels (e.g., desktop applications, web client applications, mobile applications, ATMs and so on). An example is Credit Suisse (Murer and Hagen 2013).

The most mature domain for RAs may be the embedded systems domain. For instance, Océ, one of the world's leading copier manufacturers, uses an RA to derive an SA for engines incorporated in a specific series of Océ printers (Graaf et al. 2005). Besides, in the automotive industry, car manufacturers such as Volvo (Eklund et al. 2005) and BMW (Reichart and Haneberg 2004) started to use RAs to develop the software of electronic/engine control unit based on basic software components that were unique to them. As a further step, AUTOSAR has become popular later because it standardizes basic software components for many car manufacturers, suppliers and other related companies (AUTOSAR 2014). This enables the reuse of software developed by original equipment manufacturers in multiple car manufacturers. This has led to software ecosystems that are characterized by a network of developers rather than a single organization providing the final product to the customer (Eklund and Bosch 2014).

To sum up, we can conclude that core idea of using an RA for the development and maintenance of a family of software products is common in all these contexts. However, RM are only commonly used in IT consulting firms. As a consequence, the study of RM during the application of the guidelines will be optional so that it can be used in these other contexts.

It is also important to note that all RAs described in this section are based on practical experience in the industry. The guidelines will target this type of RA, also known as *classical* (Angelov et al. 2012). Conversely, the guidelines would be hardly applicable in *preliminary* RAs, i.e., those that are "defined when the technology, software solutions, or algorithms demanded for its application do not yet exist in practice by the time of its design" (Angelov et al. 2012).

5.1.2 Packaging the guidelines

everis' results are being suitably packaged with the aim of being applied in prospective RA projects and also in similar organizations. Figure 6 summarizes the guidelines for building a business case and gathering evidence of RAs in industry.

First of all, organizations that may want to use these guidelines need to fit into the context depicted in Section 5.1.1. This means that they need to design an RA based on practical experience, and to use such RA for the development and maintenance of a family of applications in industry. This is because the input for using the guidelines is evidence from real RA projects.

The guidelines support organizations:

1. To decide whether to invest on an RA by providing them with:
    - A checklist to analyze existing value-driven data in RA projects.
    - An economic model that uses such value-driven data to calculate the ROI of adopting an RA.

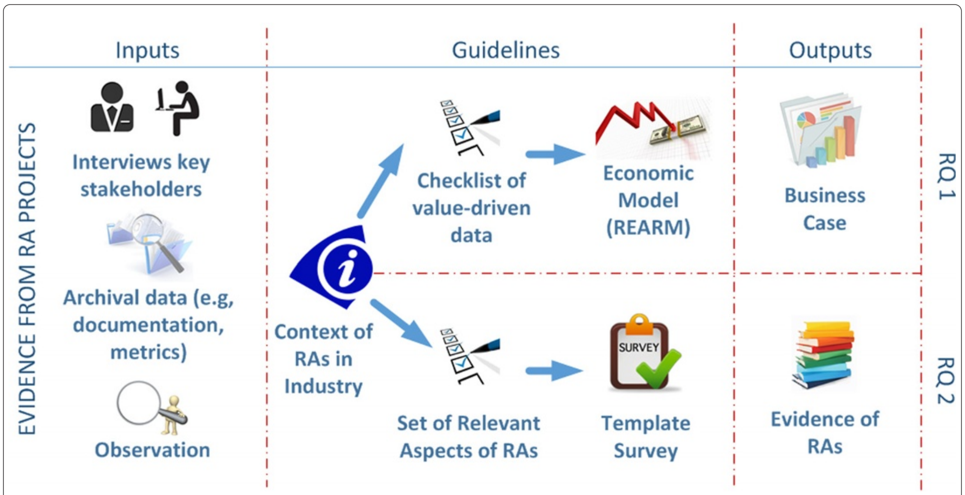
Inputs
Guidelines
Outputs
EVIDENCE FROM RA PROJECTS
Interviews key stakeholders
Archival data (e.g, documentation, metrics)
Observation
Context of RAs in Industry
Checklist of value-driven data
Economic Model (REARM)
Business Case
RQ 1
Set of Relevant Aspects of RAs
SURVEY
Template Survey
Evidence of RAs
RQ 2

Figure 6 Empirical studies of the guidelines for building a business case and gathering evidence of RAs in industry. The guidelines are composed of the context of RAs in industry, and four empirical studies. The four empirical studies are used as follows. To answer RQ 1, there are two studies: a survey to check existing value-driven data in organizations and a case study to calculate the ROI of adopting an RA. To answer RQ 2, there are two studies: a focus group to determine the set of criteria about RAs important for an organizations and a template survey to gather evidence about the previously identified relevant aspects of RAs.

2. To improve RA-related practices based on corporate evidence by providing them with:
   - A practitioner-oriented set of criteria to understand and evaluate an RA.
   - Templates of interview guides and online questionnaires to gather relevant evidence.

On the one hand, to analyze whether it is worth to invest on an RA, a checklist of value-driven data that an organization might have is facilitated to check if the provided economic model provided can be executed. On the other hand, to improve RA-related practices with evidence, a set of relevant aspects for RAs is facilitated to check which ones are important for an organization and then use the provided template surveys to gather evidence.

To gather evidence with the help of the aforementioned artifacts, empirical studies can be set up and carried out. Several data collection techniques exist (Lethbridge et al. 2005). We propose surveys that use interviews and questionnaires, case studies that require documentation analysis and metrics gathering, and focus groups. Below, we explain the dimension, context, objective and method applied in each empirical study of the guidelines. To see the threats to validity in *everis*, the reader is referred to (Martínez-Fernández 2013).

I) A survey to check existing value-driven data in RA projects

Dimensions RQ 1.

Context Typically, organizations do not have resources to compare the real cost of creating applications with and without an RA and historical data may be scarce. Thus, alternatives should be considered.

Objective The objective of this survey is to identify the quantitative information that can commonly be retrieved in RA projects in order to quantitatively calculate the costs and benefits of adopting an RA in an organization. This is an initial step to create repeatable techniques for performing a cost-benefit analysis.

Method Exploratory surveys with personalized questionnaires applied to relevant stakeholders (e.g., manager, architect, developer) to find out the quantitative data that has been collected in RA projects and SA projects. An example of conducting this empirical study and its approach for data collection is described in Section 3.1.

II) A case study to apply REARM to calculate the ROI of adopting an RA

Dimensions RQ 1.

Context Before deciding to launch an RA, organizations need to analyze whether to undertake or not the investment. Offering organizations an economic model that is based on former RA projects data can help them to make more informed decisions.

Objective The objective is to analyze whether it is worth investing on an RA with the help of an economic model, in order to improve the communication among architects and management, and to improve their decisions.

Method A case study that applies an economic model to calculate the ROI of adopting an RA. Depending on the maturity of the organization, two approaches can be applied. If the organization does not have experience with an RA, the economic model should be fed with estimated data. Nevertheless, when the organization already has experience with RAs (i.e., the case of IT consulting firms), real data can be gathered by means of an exploratory quantitative post-mortem analysis. Then, the economic model quantifies the potential advantages and limitations of using an RA. Some related works explain how to calculate the ROI of a product (Forrester 2014), software reuse (Frakes and Terry 1996; Poulin 1997), and software product lines (Ali et al. 2009). We suggest the use of REARM, which is an economic model specific for RAs presented in (Martínez-Fernández et al. (Martínez-Fernández et al. 2013b)). An example of conducting this empirical study and its approach for data collection is described in Section 3.2.

III) A focus group to study the relevant criteria of RAs for an organization

Dimensions RQ 2.

Context Typically, organizations drive the design and use of RAs in an unsystematic manner (Eklund et al. 2005). To drive RA-related practices based on evidence, it becomes fundamental to identify the relevant aspects of RAs as seen by practitioners.

Objective The objective of this study is to identify the aspects that are important for each organization in order to support RA-related practices.

Method A focus group with relevant stakeholders (e.g., manager, architect, developer) to find out which aspects of RAs are important to them. A focus group is considered a proven and tested technique to obtain the perception of a group of selected people on a defined area of interest (Lethbridge et al. 2005). An example of conducting this empirical study and its approach for data collection is described in Section 4.1.

IV) Surveys to gather evidence to improve RA-related practices

Dimensions RQ 2.

Context To reuse architectural knowledge and improve RA-related practices in prospective RA projects, organizations need to understand RA's characteristics, as well as its potential benefits and limitations. Gathering evidence from previous RA projects is a feasible way to start gaining such an understanding.

Objective The purpose of the survey is to understand the impact of using RAs for designing the SAs of the applications of an information system of a client organization. This is a descriptive survey that measures what occurred while using RAs rather than why. The following questions are important for organizations in order to understand relevant Aspects 1 to 5 of RAs (defined in Table 2):

1. Why is an RA adapted for creating SAs of the organizations' applications? What type of RA is being designed and used in the organization?
2. What is the state of practice in requirement engineering for RA projects in the organization?
3. What is the state of practice in architectural design for RA projects in the organization?
4. Which tools and technologies are currently being used in RAs projects by the organization?
5. How does the adoption of RAs provide observable benefits to the different actors involved in RA projects in the organization?

Method Exploratory surveys with personalized questionnaires applied to relevant stakeholders (e.g., architects, developers) to gather their perceptions and needs. An example of conducting this empirical study and its approach for data collection is described in Section 4.2.

Finally, the output for using guidelines for RQ 1 is a business case that evaluates whether it is worth or not to invest on an RA. On the other hand, the output of gathering evidence about Aspects 1 to 5 of Table 2 is a corporate knowledge base about these aspects. It is important to note that, although it is not strictly necessary, it is recommended to gather Aspects 1 to 5 for building the business case. In other words, guidelines are complementary and support each other (e.g., results from a preceding study can be used to corroborate or further develop other results). For instance, in our industry-academia collaboration, the qualitative results about the benefits and drawbacks of RAs supported the unquantifiable benefits, and uncertainties and risk of the business case.

5.2 Summative stage: validating the guidelines

The summative stage will take place once the guidelines will have been adequately shaped and improved. The primary role of this stage will be to validate the version of the guidelines after applying it in *everis* with more practitioners. This evaluation consists of the use of the guidelines in other organizations to design and conduct empirical studies. Organizations analyzing whether to make the strategic move to RA adoption and organizations that face the design and use of RAs based on evidence will benefit from these guidelines.

6 Conclusions

Conducting empirical studies is becoming one of the main sources of communication between practitioners and the academia. We are conducting an action research initiative in *everis*. This has allowed gathering industrial evidence:

- To build a business case of adopting an RA in a public administration of Spain, which is an *everis*' client (Martínez-Fernández et al. 2013b). The feedback from this study improved the checklist of value-driven data and the REARM economic model. We learned the importance of good practices like time tracking, continuous feedback, test-driven development and continuous integration in order to quantitatively evaluate RAs. We performed a cost-benefit analysis of an RA adoption in a public administration and it showed that the RA pays off after creating 7 small applications. With medium to large applications, this number could be reduced to 2 applications.
- To create an evidence-based information about RAs in *everis*. It includes evidence about: (a) the motivations of *everis*' client organizations to adopt RAs; (b) benefits and drawbacks of using RAs (Martínez-Fernández et al. 2013c); and (c) artifacts of RAs (Martínez-Fernández et al. 2014). The feedback from these studies improved the set of relevant criteria of RAs, and our template survey to gather evidence about RAs. Among the main benefits, stakeholders indicated that the adoption of an RA bring cost savings in the development and maintenance of applications and improved productivity during the design of concrete SAs.

These results have been evaluated by the *everis*' managers involved in the research. Besides, for their widespread use in *everis*, they have been published in form of proposed solutions, internal reports, executive summaries, and pilots run in real projects (Martínez-Fernández and Marques 2014).

The main contribution of this work is the formulation of guidelines for conducting empirical studies to support building a business case and gathering evidence of RAs. These guidelines consist of stating the context of RAs in industry and their stakeholders, and an assortment of four complementary empirical studies that allow understanding and evaluating RAs. The created guidelines could be used by other organizations outside of our industry-academia collaboration. With this goal in mind, we describe the guidelines in detail for their use. However, practitioners should be aware that conducting empirical studies is time consuming and they should dedicate proper effort.

Future work spreads into two directions. In terms of shaping the guidelines (the formative stage), we need to validate REARM in another *everis*' RA project and to analyze the rest of qualitative data from the survey to gather evidence about RAs. With respect to validation (the summative stage), we plan to apply the guidelines in other organizations if possible.

Additional files

Additional file 1: List of questions of the survey to check existing value-driven data in RA projects (in Spanish).

Additional file 2: Protocol of surveys to gather evidence and to create a corporate knowledge base about RAs.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Every author was important for the completion of this article, however, their previous activities are also important to reach the research results, then, these activities are listed in this section. SMF, CA and XF envisaged the guidelines and the protocol of the studies. HM was responsible of looking for RA projects at *everis*, contacting the participants of the studies to schedule meetings with SMF and DA and provide more information when necessary. The guidelines have been shaped in periodic meeting at *everis* among SMF, CA, XF and HM. SMF and DA performed the face-to-face interviews to the survey participants. All authors read and approved the final manuscript.

Acknowledgements

This work has been supported by "Cátedra *everis*" and the Spanish project TIN2010-19130-C02-01. We would also like to thank all participants of the surveys for their kind cooperation.

Author details

[1]GESSI Research Group, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, 08034 Barcelona, Spain. [2]Everis, Diagonal, 605, 08028 Barcelona, Spain.

Received: 15 November 2013 Accepted: 13 May 2014
Published: 21 August 2014

References

Ali MS, Babar MA, Schmid K (2009) A comparative survey of economic models for software product lines. In: Software Engineering and Advanced Applications, 2009. SEAA'09. 35th Euromicro Conference on. IEEE, pp 275–278. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5349963&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5349963.

Angelov S, Trienekens JJ, Grefen P (2008) Towards a method for the evaluation of reference architectures: Experiences from a case. In: Software Architecture. Springer, Berlin Heidelberg, pp 225–240. http://link.springer.com/chapter/10.1007%2F978-3-540-88030-1_17.

Angelov S, Grefen P, Greefhorst D (2012) A framework for analysis and design of software reference architectures. Inform Software Tech 54(4):417–431

Angelov S, Trienekens J, Kusters R (2013) Software reference architectures-exploring their usage and design in practice. In: Software Architecture. Springer, Berlin Heidelberg, pp 17–24. http://link.springer.com/chapter/10.1007%2F978-3-642-39031-9_2.

Atlassian (2014) Jira: Issue and project tracking software. https://www.atlassian.com/es/software/jira. Accessed 12 Feb 2014

AUTOSAR (2014) Automotive open system architecture (autosar). http://www.autosar.org/. Accessed 12 Feb 2014

Bass L, Clements P, Kazman R (2003) Software architecture in practice. Addison-Wesley Professional. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=30659.

Bass L, Clements P, Kazman R, Klein M (2008) Evaluating the software architecture competence of organizations. In: Software Architecture, 2008. WICSA 2008. Seventh Working IEEE/IFIP Conference on, IEEE, pp 249–252. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4459163&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F4459128%2F4459129%2F04459163.

Brand S (2014) Accenture has business-outcome-driven ea capabilities, but doesn't automatically begin with this approach. https://www.gartner.com/doc/2651816/accenture-businessoutcomedriven-ea-capabilities-doesnt. Accessed 25 Feb 2014

Brydon-Miller M, Greenwood D, Maguire P (2003) Why action research? Action Res 1(1):9–28. doi:10.1177/14767503030011002, http://arj.sagepub.com/content/1/1/9.short.

Carriere J, Kazman R, Ozkaya I (2010) A cost-benefit framework for making architectural decisions in a business context. In: Software Engineering, 2010, ACM/IEEE 32nd International Conference on, IEEE, vol 2, pp 149–157. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6062148&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6062148.

Clements P, Northrop L (2002) Software product lines: practices and patterns. Addison-Wesley Professional. http://www.addison-wesley.de/9780201703320.html.

Clements P, Kazman R, Klein M (2001) Evaluating software architectures: methods and case studies. Addison-Wesley Professional. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=30698.

Cloutier R, Muller G, Verma D, Nilchiani R, Hole E, Bone M (2010) The concept of reference architectures. Syst Eng 13(1):14–27

Deelstra S, Sinnema M, Bosch J (2005) Product derivation in software product families: a case study. J Syst Soft 74(2):173–194. doi:10.1016/j.jss.2003.11.012, http://www.sciencedirect.com/science/article/pii/S0164121203003121.

Eklund U, Bosch J (2014) Architecture for embedded open software ecosystems. J Syst Softw. doi:10.1016/j.jss.2014.01.009, http://dx.doi.org/10.1016/j.jss.2014.01.009, http://www.sciencedirect.com/science/article/pii/S0164121214000211.

Eklund U, Askerdal Ö, Granholm J, Alminger A, Axelsson J (2005) Experience of introducing reference architectures in the development of automotive electronic systems. In: ACM SIGSOFT Software Engineering Notes, vol 30. ACM, New York, pp 1–6

Erdogmus H, Favaro J (2012) The value proposition for agility–a dual perspective. http://www.infoq.com/presentations/Agility-Value. Accessed 3 Mar 2014

Falessi D, Babar MA, Cantone G, Kruchten P (2010) Applying empirical software engineering to software architecture: challenges and lessons learned. Empir Software Eng 15(3):250–276

Forrester (2014) Forrester's total economic impact (tei). http://www.forrester.com/marketing/product/consulting/tei.html. Accessed 12 Feb 2014

Frakes W, Terry C (1996) Software reuse: metrics and models. ACM Comput Surv (CSUR) 28(2):415–435

Gallagher BP (2000) Using the architecture tradeoff analysis method to evaluate a reference architecture: A case study. Tech. rep., Software Engineering Institute, CMU/SEI-2000-TN-007

Galster M, Avgeriou P (2011) Empirically-grounded reference architectures: a proposal In: Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality of software architectures–QoSA and architecting critical systems–ISARCS. ACM, New York, pp 153–158

Galster M, Avgeriou P, Tofan D (2013) Constraints for the design of variability-intensive service-oriented reference architectures–an industrial case study. Inf Softw Technol 2(55):428–441. http://dx.doi.org/10.1016/j.infsof.2012.09.011

García-Alonso J, Olmeda JB, Murillo JM (2010) Java para aplicaciones corporativas de la administración. In: Teniente E, Abrahão S (eds) JISBD. IBERGARCETA Pub. S.L. 2010, pp 263–266. http://www.informatik.uni-trier.de/~ley/db/conf/jisbd/jisbd2010.html#Garcia-AlonsoOM10

Graaf B, Van Dijk H, van Deursen A (2005) Evaluating an embedded software reference architecture. industrial experience report In: Software Maintenance and Reengineering, 2005. CSMR 2005. Ninth European Conference on, IEEE, pp 354–363. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1402153.

Herold S, Mair M (2013) Checking conformance with reference architectures: a case study In: Enterprise Distributed Object Computing Conference (EDOC)2013 17th IEEE International. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6658265.

Jenkins (2014) An extendable open source continuous integration server. http://jenkins-ci.org/. Accessed 12 Feb 2014

Lethbridge T, Sim S, Singer J (2005) Studying software engineers: data collection techniques for software field studies. Empir Software Eng:311–341. http://link.springer.com/article/10.1007/s10664-005-1290-x.

Northrop LM, Clements PC (2014) A framework for software product line practice, version 5.0. http://www.sei.cmu.edu/productlines/frame_report/index.html. Accessed 1 Mar 2014

Linden FVD, Bosch J, Kamsties E (2004) Software product family evaluation In: Software Product Lines. Springer, Berlin Heidelberg, pp 110–129. http://link.springer.com/chapter/10.1007%2F978-3-540-24667-1_27.

MacCormack A, Baldwin C, Rusnak J (2012) Exploring the duality between product and organizational architectures: a test of the mirroring hypothesis. Res Pol 41(8):1309–1324

Martínez-Fernández S (2013) Towards supporting the adoption of software reference architectures: an empirically-grounded framework In: Proceedings 11th International Doctoral Symposium on Empirical Software Engineering (IDoESE) at the Empirical Software Engineering International Week; Baltimore, Maryland, USA. available at http://umbc.edu/eseiw2013/idoese/pdf/eseiw2013_IDoESE_180.pdf. Accessed 12 Feb 2014

Martínez-Fernández S, Marques H (2014) Practical experiences in designing and conducting empirical studies in industry-academia collaboration In: Conducting Empirical Studies in Industry (CESI), 2014 2nd International Workshop on. ACM, New York, pp 15–20. http://doi.acm.org/10.1145/2593690.2593696.

Martínez-Fernández S, Ayala C, Franch X, Ameller D (2013a) A framework for software reference architecture analysis and review In: Memorias del X Workshop Latinoamericano de Ingeniería en Software Experimental ESELAW2013; Montevideo, Uruguay. available at: http://cibse2013.ort.edu.uy/pdf/ESELAW-ISBN-978-9974-8379-3-5.pdf. Accessed 12 Feb 2014

Martínez-Fernández S, Ayala C, Franch X, Marques H (2013b) Rearm: a reuse-based economic model for software reference architectures. In: Favaro J, Morisio M (eds) Safe and Secure Software Reuse, Lecture Notes in Computer Science, vol 7925, pp 97–112. doi:10.1007/978-3-642-38977-1_7

Martínez-Fernández S, Ayala C, Franch X, Martins Marques H (2013c) Benefits and drawbacks of reference architectures. In: Drira K (ed) Software Architecture, Lecture Notes in Computer Science, vol 7957, pp 307–310. doi:10.1007/978-3-642-39031-9_26

Martínez-Fernández S, Ayala C, Franch X, Marques H (2014) Artifacts of software reference architectures: a case study In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. doi:10.1145/2601248.2601282

Mattmann C, Downs R (2010) Reuse of software assets for the NASA Earth science decadal survey missions. Geoscience and Remote Sensing Symposium (IGARSS)2010 IEEE International, vol 1. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5653018.

Montagud S, Abrahão S, Insfran E (2011) A systematic review of quality attributes and measures for software product lines. Software Qual J 20:425–486. doi:10.1007/s11219-011-9146-7, http://link.springer.com/10.1007/s11219-011-9146-7.

Murer S, Hagen C (2013) 15 Years of Service Oriented Architecture at Credit Suisse:1–30. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6654116. Accessed 25 Feb 2014

Nakagawa EY, Antonino PO, Becker M (2011) Reference architecture and product line architecture: a subtle but critical difference In: Software Architecture. Springer, Berlin Heidelberg, pp 207–211. http://link.springer.com/chapter/10.1007%2F978-3-642-23798-0_22.

NASA (2012) Esds reference architecture working group: Esds reference architecture for the decadal survey era. https://earthdata.nasa.gov/sites/default/files/field/document/ESDS%20Reference%20Architecture%20v1.1.pdf. Accessed 12 Feb 2014

Poulin JS (1997) Measuring software reuse: principles, practices, and economic models. Addison-Wesley Longman Publishing Co., Inc., Boston

Qureshi N, Usman M, Ikram N (2013) Evidence in software architecture, a systematic literature review In: Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering. ACM, New York, pp 97–106

Redmine (2014) A flexible project management web application. http://www.redmine.org/. Accessed 12 Feb 2014

Reichart G, Haneberg M (2004) Key drivers for a future system architecture in vehicles In: Convergence International Congress & Exposition On Transportation Electronics. SAE International. http://papers.sae.org/2004-21-0025/.

Reifer DJ (2002) Making the software business case: improvement by the numbers. Pearson Education. http://books.google.es/books?id=W61QAAAAMAAJ.

SonarSource (2014) Continuous inspection of code quality. http://www.sonarsource.com/. Accessed 12 Feb 2014

doi:10.1186/s40411-014-0007-5
Cite this article as: Martínez-Fernández *et al.*: **Towards guidelines for building a business case and gathering evidence of software reference architectures in industry.** *Journal of Software Engineering Research and Development* 2014 **2**:7.

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Immediate publication on acceptance
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com