Towards Supporting the Adoption of Software Reference Architectures: An Empirically-Grounded Framework

Silverio Martínez-Fernández
Universitat Politècnica de Catalunya
Jordi Girona, 1-3
08034, Barcelona (Spain)
+34 93 4137174
smartinez@essi.upc.edu

ABSTRACT

A Software Reference Architecture (SRA) allows organizations to reuse architectural knowledge and software components in a systematic way and, therefore, to reduce costs. SRAs mainly appear in organizations in which the multiplicity of software systems (i.e., software systems developed at multiple locations, by multiple vendors and across multiple organizations) triggers a need for life-cycle support for all systems. Thus, SRAs are very attractive when organizations become large and distributed in order to develop new systems or new versions of systems. In return, organizations face the need to analyze the return-on-investment (ROI) in adopting SRAs, and to review these SRAs in order to ensure their quality and incremental improvement.

The goal of this research is to envisage an empirically-grounded framework that supports organizations to decide on the adoption of SRAs and its subsequent design and suitability for the organization purposes. It helps organizations to harvest and arrange relevant evidence from the wide spectrum of involved stakeholders and available information and documentation in SRA projects. Such a framework is being shaped through an action-research approach between our research group and *everis*, an IT consulting firm.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *cost estimation, life cycle, productivity*

D.2.11 [**Software Engineering**]: Software Architectures – *domain-specific architectures*

Keywords

Software reference architecture, empirical software engineering

1. INTRODUCTION

1.1 Background

Every software system has a [concrete] software architecture [8]: "the software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both" [8].

Nowadays, the size and complexity of software systems, together with critical time-to-market needs, demand new software engineering approaches to design software architectures [26]. One of these approaches is the use of a Software Reference Architecture (SRA) that allows to systematically reuse knowledge and elements when developing a concrete software architecture [11] [15]. A SRA becomes, then, the baseline for many software systems, as depicted in Figure 1.

IDoESE'13 Baltimore, Maryland USA. Copyright owned by author.

The purpose of SRAs is to serve as guidance for the development, standardization, and evolution of diverse software systems [26]. This is possible because SRAs are abstract enough to allow its usage in differing contexts [2].

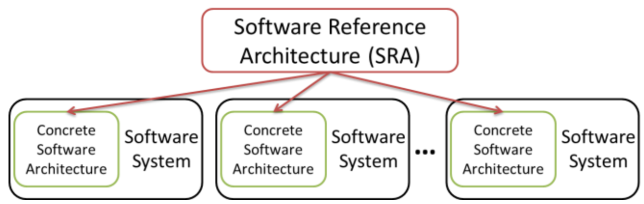

Figure 1. SRAs are the baseline of many software systems.

1.2 Motivation

It has been claimed that SRAs "reduce the complexity of hardware and software architecture by systematically reducing environmental diversity [...], enables greatly increased speed and reduce operational expenses as well as quality improvements due to lowered complexity, greater investment and greater reuse" [34]. Thus, "IT organizations that lack architecture and configuration standards [...] have higher costs and less agility that those with enforced standards" [34].

According to their expected benefits, SRAs have become widely studied and used by researchers and practitioners. There are many examples of SRAs of different types. On the one hand, there are SRAs that target a *technological* domain (also called platform-oriented SRAs [27]) such as The Open Group Standard for SOA Reference Architecture, which is a blueprint that provides guidelines and options for making architecture, design, and implementation decisions when adopting a service-oriented approach to information technology [38]. There are also SRAs from academia to solve well-known technological problems (e.g., software testing tools [29]).

On the other hand, there is another type of SRAs that focus on a specific *business* domain. These SRAs can either target many organizations (whose software systems share the business domain), or target a specific single organization (which aims to standardize or facilitate the development and maintenance of its own software systems). An example of a SRA that targets many organizations is AUTOSAR [5], which focus on the automotive domain and is being used for many car manufacturers, and suppliers in order to standardize the software in modern vehicles.

An example of SRAs for a single organization is the SRA for NASA's Earth Science Data Systems, which facilitates and homogenizes the development of this type of systems [28]. In this thesis work, we pay special attention to this last type of SRAs, those that are designed for and adopted in a single organization. They appear in organizations where the multiplicity of software systems (i.e., systems developed at multiple locations, by multiple vendors and across multiple organizations) triggers a need for life-cycle support for all systems [11]. In this context, SRAs are very attractive when organizations become large and distributed [25] in order to develop new systems or new versions of systems.

1.3 Problem Statement

Despite SRAs are considered a highly relevant strategic asset by industry and academia, there are still some gaps that hamper its successful adoption. While comprehensive methods have been defined for the analysis of cost and benefits (e.g., CBAM [8]) and the review (e.g., ATAM [10]) of (concrete) software architectures, SRAs have received relatively less attention in literature. The reason for this can be probably traced in an assumption that these methods for software architectures are directly applicable to SRAs. But in practice, as Angelov et al. point out [4], practitioners face additional difficulties in working with SRAs. It is due to the specific features of SRAs with respect to software architectures [3], such as the need of a high up-front investment, their generic nature and high level of abstraction, the wide group of stakeholders that they involve, or the risks from the instantiation in the organization's portfolio of software systems. This situation triggers specific questions that have not been addressed yet. Specifically, two major issues that need further research have been identified:

1. Lack of support for deciding on the adoption of SRA. There is a shortage of approaches to precisely evaluate the benefit of architecture projects [9] in order to take informed decisions about adopting a SRA in an organization. Thus, managers and executives lack of support to analyze whether it is worth to invest on the adoption of a SRA in their organization and to calculate its return-on-investment (ROI).
2. Lack of support for SRA design and review. On the one hand, in spite of research on the elements that should compose a SRA [27] and methods about how to design them [15], there is little evidence and support about how they are actually put forward in practice. On the other hand, although there are several evaluation methods for SRA review [3][14][17], they have been hardly applied in the industrial practice. One potential reason could be that the organizations find them expensive to apply and hard to be customized and selected for their specific needs and practices [7]; especially because there is no support for identifying the real factors that might apply in their specific organizations (e.g., some methods suggest to consider the important quality attributes for the organization, but which are such attributes for the organization?). Therefore, in such situations, practitioners face difficulties that jeopardize the success of the SRA project.

In this context, we greatly believe that the availability of evidence about cost and benefits of real SRA acquisition programs and SRA design and review industrial experiences would serve as a basis for articulating a framework to support organizations and practitioners to face both problems.

The paper is structured as follows. Section 2 describes previous work to cope with these two issues. Section 3 shows the goal of this thesis. Section 4 presents the empirically-grounded framework that aims to accomplish such goal. Section 5 explains the action-research initiative in which the framework is being shaped and validated, and Section 6 the threats to validity. Section 7 shows the current status of this research and future work. Finally, Section 8 exposes the points in which advice would be more valuable.

2. RELATED WORK

This section presents firstly related work for calculating the ROI of architecture-centric approaches; and secondly related work about reviewing SRAs.

2.1 SRA Economics

Although there is a lack of research in evaluating the economic viability of SRA adoption, there is a strong base of research in related areas that could be adapted with this purpose. One the one hand, given the reusable nature of Software Product Lines (SPL), the economic models that have been proposed in this area for identifying its costs and benefits can be adapted to SRA economics. However, our analysis of these models led us to identify that they mainly fall short in:

- Validation in industry. "Very few [economic models for product line architecture] actually have been used as a basis for further development or adopted in industry" [20]. Thus, "there is a clear need for many more empirical studies to validate existing models" [1].
- Easy adoption of models in industry by identifying realistic metrics to collect and report. "It is difficult for the practitioners to evaluate the usability and usefulness of a proposed solution [economic models for product line architecture] for application in industry" [20]. No guidelines exist to fully operationalize the models in practice [33].

On the other hand, there is also related work in other areas. For instance, economics-driven software architecture analysis methods (e.g., CBAM [8]). However, existing proposals do not specifically aim at making an investment analysis of the adoption of an architecture-centric program. SRA adoption is actually a sub area inside their generic decision-making context. Furthermore, other works have addressed aspects as the quantification of the benefit from architecture projects that improve quality attributes [9], and the uncertainties associated with early lifecycle cost estimation [13]. At a lower level, generic software metrics like design structure matrix (DSM), could also be adequate for calculating the cost and benefit factors of SRA adoption and make more complete models. As a result, the intended thesis uses and tailors knowledge in these mentioned areas for its own purposes.

2.2 Review of SRAs

The software architecture of a software system is an early result of the development cycle that helps to identify and address important quality aspects, such as system's performance, security, reliability, and maintainability, before the implementation of the system is started [10]. Therefore, it is a strategic approach for reducing complexity, costs and risks. As a result, to ensure architectural quality is an elusive goal. Thus, the practice of evaluating software architectures (i.e., the process of evaluating whether suitable aspects have been addressed) has matured, with well-known methods such as ATAM [10], which helps stakeholders understand the consequences of architectural decisions with respect to the software system's quality attribute requirements.

Existing methods for software architecture evaluation have been previously applied for SRAs, such as in [3][14][17]. However, "the software engineering community rarely adopts the methods and techniques available to support disciplined architecture review processes" [7]. Four possible reasons for this are:

- As Ali Babar et al. point out, we think that "there remains a need for systematically accumulating and widely disseminating evidence about the factors that may influence the selection and use of different methods, techniques, and tools for architecture evaluation" [6].
- Evaluation methods do not include evidence about relevant aspects of SRAs, such as which quality attributes are relevant in this type of projects.
- Evaluation teams need to have the vision from all stakeholders (e.g., project managers, software architects, developers, etc.). This is not always supported by evaluation methods, leading to problems while conducting architectural reviews. 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 reference architecture [14].
- There is a lack of recent research that proposes means to evaluate reference architectures [26].

These four issues can be addressed by collecting evidence about the factors that influence the selection of evaluation methods, studying the important review criteria and the interest of essential stakeholders. For this reason, more and more empirical studies to support SRAs theory are starting to be conducted, as [4][16][21].

2.3 What is it needed?

This state-of-the-art drove us to:

- The formulation of an economic model for SRAs built upon:
    - Cost and benefit factors from product line architecture models that are easy-to-apply by the industry. The goal is to provide guidelines to fully operationalize the model in practice.
    - The gap of SRA economics inside the software architecture decision-making context.
    - Generic software metrics that can quantify new cost and benefit factors.
- The study and collection of evidence about relevant aspects to support the design of SRA and the use of architecture evaluation methods. In [21], we identified six qualitative relevant aspects (overview, requirements, architectural decisions, business qualities, methodology and technologies) which we take as a primary input for their further refinement based on the evidence from organizations.

3. RESEARCH OBJECTIVES

This section exposes the goal of this thesis: to support organizations to decide on the adoption of SRAs and its subsequent design and suitability for the organization purposes. As stated above, such a goal will be dealt with by exploiting real evidence. It is divided into two Research Questions (RQ), which respectively deal with the two problems stated in Section 1.3.

3.1 Research Question 1

The creation and maintenance of complex software systems involves making a series of business-critical architecture design decisions. Imagine that you are the CIO of an organization with a wide portfolio of software systems. You have read about the expected benefits that a SRA may bring to your organization, e.g., standardization of concrete software architecture of systems, greater reuse, shorter time-to-market, reduced costs, reduced risk, support for system development at multiple locations, by multiple vendors and across multiple organizations, and so on. Therefore, you are considering adopting an existing or new SRA to create and maintain your organization's software systems. However, how do you know if it is worth for your organization to invest on the adoption of a SRA? This question could be answered by making a business case with the help of an economic model for SRAs. In the SRA context, an economic model is needed to help making business cases. An economic model should take into account costs, benefits, risks, and schedule implications. An economic model to perform cost-benefit analysis on the adoption of SRA is a key asset for optimizing architectural decision-making.

Reifer defines a business case as the "materials prepared for decisions makers to show that the idea being considered is a good one and that the numbers that surround it make financial sense" [31]. That is, business cases enable to justify investments in technology. Spending in the adoption of a SRA without a previous and trustworthy analysis seems to be reckless and can lead to a disaster. This triggers the statement of the RQ 1:

RQ 1. How can organizations be supported to quantitatively analyze the up-front investment on the adoption of a SRA?

The objective of the RQ 1 is to provide guidelines to support organizations to quantitatively analyze if it is worth to adopt a SRA. Such an objective consists of constructing an economic model that supports the decision of adopting a SRA. This analysis optimizes the decision-making process when studying whether to make the strategic move to SRA in an organization.

This research question is divided into four sub-research questions. The motivation of each sub-research question is as follows. First of all it is needed to understand the context in which SRAs are adopted (RQ 1.1). Second, to identify the data that can be easily collected in the industry to quantify the costs and benefits of SRAs (RQ 1.2). Third, to define the actual costs and benefits implied by SRA adoption (RQ 1.3). Finally, to make the business case for the adoption of SRAs (RQ 1.4).

- RQ 1.1: How are SRAs used in practice?
- RQ 1.2: Which available data do organizations have to quantitatively calculate the costs and benefits of adopting a SRA in an organization?
- RQ 1.3: Which are the cost and benefit factors of acquiring a SRA in an organization?
- RQ 1.4: How is it possible to calculate the ROI of the adoption of a SRA in an organization?

3.2 Research Question 2

Introducing a SRA into an organization not only involves making a decision considering the aforementioned productivity issues, but also involves the analysis of risks, non-risks, benefits and trade-offs. Whereas productivity is actually measured in terms of effort/cost and economic benefits, architectural quality is usually estimated in relation to eliciting implicit and explicit requirements of the different stakeholders affected by the development of the system. Nevertheless, both views are necessary to achieve a comprehensive analysis of the system.

To help organizations to cope with architectural quality during the design and review processes of SRAs, we propose to accumulate real evidence about the relevant aspects for SRAs from key stakeholders (e.g., which quality attributes they mainly enforce). This triggers the statement of the RQ 2:

RQ 2. How can organizations be supported to deal with architectural quality from its own industrial evidence?

The objective of the RQ 2 is to support practitioners to deal with architectural quality by providing them a way to gather its own industrial evidence about the aspects that matter for their SRA. Such gathered aspects are aimed to be used to feed any existing architectural evaluation method. Specifically, this research question is divided into five sub-research questions. These RQs address relevant aspects for the design and review of SRAs: business qualities (RQ 2.1), elements (RQ 2.2), requirements (RQ 2.3), architectural decisions (RQ 2.4), and supportive technologies (RQ 2.5) respectively.

- RQ 2.1: How different stakeholders perceive the potential benefits and drawbacks of SRAs?
- RQ 2.2: Which are the elements that compose a SRA and what is their potential reuse across domains?
- RQ 2.3: Which quality attributes does a SRA enforce?
- RQ 2.4: How are architectural decisions taken and documented in SRA projects?
- RQ 2.5: Which supportive technologies (i.e., methodologies, tools) are currently being used in SRA projects?

The next section presents an empirically-grounded framework that aims to provide means to answer these RQs.

4. RESEARCH APPROACH

Empirical research is a way of gaining knowledge by means of direct and indirect observation or experience. One of the objectives of Empirical Software Engineering is to gather and utilize evidence to advance software engineering methods, processes, techniques, and tools. This thesis proposal fosters the conduction of empirical studies as a way to incrementally build up SRA theory. The next section describes the expected contribution of this thesis: a Framework for SRA Analysis and Review.

4.1 An Empirically-Grounded Framework

To accomplish the goal of this research, we plan to devise a framework by providing procedural guidelines for setting up and carrying out empirical studies. The framework is composed of an assortment of empirical studies that would help organizations to deal with RQ1 and RQ2. Each empirical study fits into one of the three steps for empirical research suggested by Wohlin et al. [39]: *understand*, *evaluate* and *improve*. The main idea is that it is in most cases impossible to start improving directly and that empirical studies can be complementary and support each other (e.g., results from a preceding study can be used to corroborate or develop further these results).

The framework explicitly deals with the understanding and evaluation steps. The improving step is achieved by iteratively applying the evaluation step and considering the lessons learned. The studies should be conducted sequentially.

Figure 2 describes the studies that compose the framework. The rows indicate the step in which the study is being applied whereas the columns show the RQ that the study approaches.

As shown by Figure 2, the framework is composed of four studies, two for each RQ:

- RQ 1 is supported by two studies:
  - A Survey to check existing value-driven data in organizations. It aims to provide support or guidelines to check existing value-driven data in the organization in order to perform a quantitative evaluation.
  - An Economic model to calculate the ROI of adopting a SRA. It aims to provide an economic model to calculate the ROI of adopting a SRA.
- RQ 2 is supported by two studies:
  - A Survey to understand the impact of using a SRA. It aims to provide support or guidelines to understand the impact of using a SRA in the organization in order to perform a qualitative evaluation.
  - An architectural evaluation method specific for SRA. The above survey helps to provide support for the selection of an architectural evaluation method for SRA and easy its conduction with information from key stakeholders. As explained in Section 2.2, currently, there exist evaluation methods. Therefore, the framework just enables practitioners for the smooth application of such existing methods.

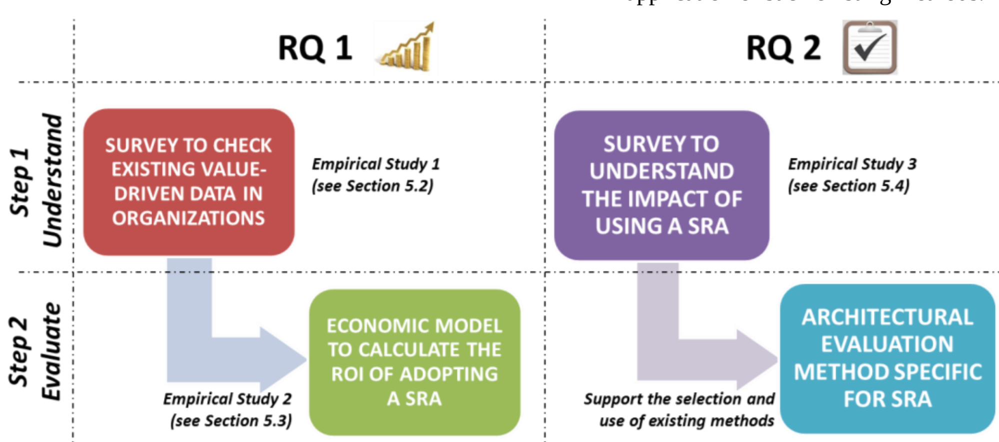

Figure 2. Empirically-grounded framework to support organizations on SRA adoption, design and review.

5. ACTION-RESEARCH PROJECT

This research has its origin in an ongoing action-research initiative among our research group and *everis*, a consulting company. Action research is "learning by doing" - a group of people identify a problem, do something to resolve it, see how successful their efforts were, and if not satisfied, try again [18]. As a result, the aforementioned framework is being devised by applying the action-research cycle in *everis*: 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) repeat for improvement.

As a consulting company, *everis* offers solutions for big businesses (e.g., banks, insurance companies, public administration, utilities, and industrial organizations) that need to manage a wide portfolio of software systems that share a specific-domain. Given the complexity of the resulting information systems, which integrate bespoke applications with commercial packages, these systems need high-quality software architecture. This is the service that organizations hire to *everis*. The solution provided by *everis* is based on the adoption of a SRA in the organization, from which concrete software architectures are derived and used in a wide spectrum of software systems.

In this context, *everis* commissioned our research group two main tasks (respectively aligned with our RQs):

- to calculate the ROI that organizations get after adopting a SRA.
- and to gather evidence to support the design and review of SRAs for their clients.

Precisely, the architecture group of *everis* experienced difficulties to cope with the two issues that we are coping with in the RQs. As a consequence, we are applying the studies envisaged in RQ 1 in order to calculate the ROI derived from SRAs that *everis* created (or plan to create) for organizations. On the other hand, to support them to achieve architectural quality, we plan to conduct the studies of RQ 2. To do so, it is necessary to contact SRA's stakeholders [22]. In *everis*, three essential roles are distinguished: *software architects* that cooperatively work to figure out a SRA to accomplish the desired quality attributes and architecturally-significant requirements of the client organization; *architecture developers* that are responsible for coding, maintaining, integrating, testing and documenting the SRA's software components; and *application builders* that take reusable components from the SRA and instantiate them to build concrete software architectures for software systems.

5.1 Framework Shaping and Validation

The expected result of this thesis is the empirically-grounded framework aforementioned (see Figure 2). The framework will be incrementally constructed based on the action-research approach in *everis*. *everis'* results will be suitably packaged with the aim of being applied in similar organizations. Furthermore, for its shaping and validation, the research is divided in two stages: the formative and the summative stages.

During the formative stage, we will conduct empirical studies 1, 2 and 3 (detailed below) in *everis*. As their conduction advances, their feedback will contribute to incrementally shape and package obtained results.

The summative stage will take place once the framework has been adequately improved, shaped and packaged. The primary role of this stage will be to validate the final version of the framework with practitioners.

The next three sub-sections respectively describe how the studies of the framework have been designed and are being conducted in the action-research initiative with *everis*.

5.2 Empirical Study 1

Objectives of this study. The objective of this survey is to identify the quantitative information that can be retrieved from past projects in order to feed the economic model (see Empirical Study 2 below). The main perceived economic benefits on the use of SRAs are the cost avoidance 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. The economic model needs this data to define and calibrate the parameters to calculate the ROI of adopting an SRA in an organization.

Method. Exploratory surveys with personalized questionnaires applied to relevant stakeholders to find out the quantitative data that has been collected in SRA projects and application projects.

Sampling. A sample of 5 *everis*' SRA projects and 5 applications built upon such SRAs have been selected, and their respective software architects and application builders.

Approach for data collection. We use online questionnaires to ask software architects and application builders about existing information in past projects for calculating cost avoidance from SRA reuse in applications. The questionnaire is composed of yes-no questions asking if specific metrics are available for SRA projects. Questionnaires enable the addition of comments and metrics if desired by the interviewees.

Data Analysis Methods and Techniques. The data analysis consists of counting in how many projects a specific metric is available.

Further details of this approach have been reported in [22].

5.3 Empirical Study 2

Objectives of this study. To remain competitive, organizations are challenged to make informed and feasible value-driven design decisions. However, there is a lack of support for evaluating the economic impact of these decisions with regard to SRAs. This damages the communication among architects and management, which can result in poor decisions. This empirical study analyze whether it is worth investing in a SRA with the help of an economic model.

Method. A case study in which REARM [23], which is an economic model for SRA adoption, is applied.

Sampling. A sample of 2 *everis*' SRA projects and 2 applications built upon such SRAs have been selected.

Approach for data collection. Results from the Empirical Study 1 revealed that the data available in order to calculate costs and benefits are effort and software metrics [22]. We collect these metrics, which are presented in [23], from two types of tools.

On the one hand, a time tracking tool (e.g., JIRA [19], Redmine [30]) to collect the invested effort from training, development and maintenance activities. Keeping track of activities is common in practice for project management and auditing.

On the other hand, tools that calculate software metrics to analyze the benefits that can be found in the source code. For instance, SonarSource [37] offers tool support for obtaining general software metrics such as LOC, dependencies between modules, technical debt, and percentages of tests and rules compliance.

Data Analysis Methods and Techniques. For analyzing the output of the economic model we apply analysis techniques for business case, such as breakeven analysis, cost-benefit analysis, payback analysis and sensitivity analysis [31].

We have conducted this empirical study in a public administration. Results have been published in [23].

5.4 Empirical Study 3

Objectives of this survey. The purpose of this survey is to understand the impact of using SRAs for designing the concrete software architecture of the applications of a client organization. This is a descriptive survey that measures what occurred while using SRAs rather than why. With this survey we want to incrementally increase the evidence about key aspects that really matter to the SRA being analyzed. These key aspects correspond to the five sub-RQs of RQ 2.

Method. Descriptive surveys with personalized questionnaires applied to relevant stakeholders to gather their perceptions and needs.

Sampling. The target populations of this survey are SRA projects and SRA-based applications executed by *everis*. A sample of 9 representative *everis*' SRA projects in client organizations was selected. In these projects we plan to contact three essential stakeholders: software architects, SRA developers and application builders. All these projects were from Europe (seven from Spain).

Approach for data collection. On the one hand, semi-structured interviews are used for software architects. The reason of using interviews is that these roles have higher knowledge than the other roles about SRA, so we want to collect as much information as possible from them. Prior to the interviews, questionnaires are delivered to collect personal information about the interviewee and to inform him/her about the interview. On the other hand, online questionnaires are used for SRA 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.

The complete version of the protocol and the questionnaires is available at http://www.essi.upc.edu/~gessi/papers/eselaw13-survey-protocol.pdf.

Data Analysis Methods and Techniques. To perform data analysis, we apply qualitative analysis methods [35]. Our research team held several discussion meetings during and after data collection. For avoiding bias, the *everis*' team did not participated on these meetings (we contacted them just for serving as intermediaries for approaching to the respondents whenever we needed clarifications). We incrementally processed the manual transcriptions of all interviews and automatically got the data from the online questionnaires. We based our analysis on grounded theory techniques [12] such as constant comparison and cross-case analysis [24]. These techniques are well-fitting in situations where the researcher does not want to use pre-conceived ideas, and instead is driven by the desire to capture all facets of the collected data and to allow the propositions to emerge from the data.

The data analysis consisted of two steps. First, analysis is driven by coding pieces of data as the constant comparison method requires [35]. We first read the interview transcripts and the data from the questionnaires and attach a coding word to a portion of the text – a phrase or a paragraph. The codes are selected to reflect the meaning of the respective portion of the interview text to a specific RQ. Second, we perform cross-case analysis [24] to see the different views from multiple stakeholders over the answers with the same code. We cluster all pieces of text that related to the same code to analyze it in a consistent and systematic way.

Finally, to interpret the results, we plan to hold a meeting with the *everis* team in order to discuss and improve our understanding of the results.

By the time of writing this paper, preliminary results about the RQ 2.1 have been reported in [21].

6. THREATS TO VALIDITY

This section discusses possible threats to validity of the design of the framework in the action-research project with *everis*. It is presented in terms of construct, internal and external validity as well as reliability, as proposed in [32][40]. It also emphasizes the mitigation actions used.

6.1 Construct Validity

To strengthen this aspect we have performed a rigorous planning of the study and established a solid protocol for data collection and data analysis. First of all, to start devising the framework, it became necessary to previously identify relevant aspects to assess SRAs. However, a commonly accepted set of criteria to assess SRAs does not exist [22]. Thus, we identified important aspects to assess SRAs out of practice and out of the literature in [22]. The framework envisages these aspects as a primary input for their further refinement during its formative and summative stages.

The close involvement of the *everis* team in the research planning and design is being vital to the suitable construction and development of the data collection instruments (i.e., the economic model's metrics, the interview guides and the questionnaires). In addition, these instruments have been piloted and enhanced to ensure their effectiveness. Given the involvement of the *everis* team on the study, we were aware of the importance of including specific mitigation actions for evaluation apprehension by ensuring the aggregated presentation of the responses and their confidentiality.

6.2 Internal Validity

One of the main relevant decisions that directly affected the sampling approach is that we decided to first choose *everis*' projects and then participants that covered the roles we were looking for. In this way, we ensured that each participant would focus his/her answers on the context of the defined project. This would allow a better interpretation and assessment of contextual information. It would otherwise have been very difficult to interpret certain SRA influential factors related to the nature of the projects. We are aware that some possible biases may be related to this strategy, for instance that the *everis*' team chooses the most successful projects as sampling. To minimize this, we explained them the importance of having a representative sampling of the projects they perform in order to obtain reliable data.

Regarding the individuals, there is always the possibility that they forget something or do not explicitly state it when they are asked about it. To reduce this risk: 1) in the case of the interviews, we discussed some potential topics that might be omitted by the respondents, and paid particular attention to them during the interviews in order to ask for clarifications if necessary; 2) in the case of the online questionnaires, we designed them in such a way that the respondent could add additional comments and has to answer all the corresponding questions while he/she could complete the questionnaire at any time, so it gives them the possibility of consulting registries and documentation in case he/she needs to remember something; 3) in all cases we performed triangulation by adding questions aimed to confirm the correctness of the answers.

We put forward several mitigation strategies. First, recording and transcribing all interviews contributed to a better understanding and assessment of the data gathered. Second, to reduce the potential researcher bias, several meetings were held among the researchers in order to discuss the results. Third, although the access to the tools, source code and documentation is provided by the *everis*' team, the collection of metrics and data from them is done by the researchers.

6.3 External Validity

As it was mentioned in Section 1.2, SRAs are widely recognized in the industry, and other organizations present a very similar context to *everis*. As a consequence, we think that it could be possible to observe similar experiences in projects and companies with similar contexts. As Seddon et al. suggests: "if the forces within an organization that drove observed behavior are likely to exist in other organizations, it is likely that those other organizations, too, will exhibit similar behavior" [36].

As future work, as part of the summative stage, we plan to replicate the framework in similar contexts to *everis*. We present the results from the studies with a detailed explanation of SRA projects' context as well as the methods and materials used. This is essential to allow the replication of the empirical studies of the framework for other SRA vendors and acquisition companies with similar contexts in order to corroborate the results and being able to generalize the results.

Moreover, we acknowledge that several other factors may influence SRA projects and their stakeholder's perceptions (as organizational processes and policies, resources, cultural issues, etc.). We, therefore plan to design the framework in a way that other factors can be included at the convenience of the organization that is applying it.

6.4 Reliability

In order to strengthen this aspect we addressed the validity of the study. Besides the strategies above, we maintained a detailed protocol, conducted the survey tasks (data collection and analysis) by at least two researchers, spent sufficient time with the study, and gave sufficient concern to the analysis of all responses. Such analysis was subsequently discussed with the *everis* team to improve the understanding and contextualization of the conclusions. Finally, in case of the existence of related theory papers or empirical studies, we study how the results from our action-research approach support or refute previous hypothesis and add new empirical-based propositions.

7. CURRENT STATUS AND FUTURE OF THIS RESEARCH

So far we have completed the following activities of the formative stage to shape the framework:

- Preliminary design of the framework [22] based on the state-of-the-art and the state-of-the-practice.
- Conduction of the Empirical Study 1 [22].
- Conduction of the Empirical Study 2 [23].
- Execution of the Empirical Study 3 and analysis of one group of questions (related to benefits and drawbacks [21]).

We plan in the immediate future to carry out the analysis of the remaining groups of questions from the Empirical Study 3.

Finally, we will package the final research results and validate them in the summative stage of the framework. This evaluation consists of the replication of the Empirical Studies 1, 2 and 3.

Organizations analyzing whether to make the strategic move to SRA adoption and organizations that face the design and review of SRAs will benefit from this framework.

8. ISSUES OF DISSERTATION

There are several issues of which I would like to get feedback at the doctoral symposium, although any feedback would be welcome:

- Are there any other relevant aspects to SRAs that could be addressed in future studies?
- Are there any more ways to improve the data collection and analysis?
- How to ensure external validity for similar contexts? How is it possible to generalize/package the results without bias?

9. ACKNOWLEDGMENTS

This work has been supported by "Cátedra everis" and the Spanish project TIN2010-19130-C02-00.

10. REFERENCES

[1] Ali, M.S., Babar, M.A., Schmid, K.: A comparative survey of economic models for software product lines. In: Software Engineering and Advanced Applications, 2009. SEAA'09. pp. 275–278. IEEE (2009)

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

[3] Angelov, S., Trienekens, J.J., Grefen, P.: Towards a method for the evaluation of reference architectures: Experiences from a case. In: ECSA, pp. 225–240. Springer (2008)

[4] Angelov, S., Trienekens, J.J., Kusters, R.: Software reference architectures - exploring their usage and design in practice. In: Software Architecture. Springer (2013)

[5] AUTOSAR: http://www.autosar.org/ (2013) [Online; accessed 10-Jul-2013]

[6] Babar, M.A., Bass, L., Gorton, I.: Factors influencing industrial practices of software architecture evaluation: an empirical investigation. In: QoSA, pp. 90–107. Springer (2007)

[7] Babar, M.A., Gorton, I.: Software architecture review: The state of practice. Computer 42(7), 26–32 (2009)

[8] Bass, L., Clements, P., Kazman, R.: Software architecture in practice, 3rd Edition. Addison-Wesley Professional (2012)

[9] Carriere, J., Kazman, R., Ozkaya, I.: A cost-benefit framework for making architectural decisions in a business context. In: Software Engineering, 2010 ACM/IEEE 32nd International Conference on. vol. 2, pp. 149–157. IEEE (2010)

[10] Clements, P., Kazman, R., Klein, M.: Evaluating software architectures. Addison-Wesley Reading (2001)

[11] Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The concept of reference architectures. Systems Engineering 13(1), 14–27 (2010)

[12] Corbin, J., & Strauss, A. (Eds.). (2008). Basics of qualitative research: Techniques and procedures for developing grounded theory. Sage.

[13] Ferguson, R. et al: "Quantifying uncertainty in early lifecycle cost estimation (QUELCE)." SEI Technical Report, paper 706 (2011).

[14] Gallagher, B.P.: Using the architecture tradeoff analysis method to evaluate a reference architecture: A case study. Tech. rep., DTIC Document (2000)

[15] Galster, M., Avgeriou, P.: Empirically-grounded reference architectures: a proposal. In: Proceedings of QoSA. pp. 153–158. ACM (2011)

[16] Galster, M., Avgeriou, P., Tofan, D.: Constraints for the design of variability-intensive service-oriented reference architectures–an industrial case study. Information and Software Technology (2012)

[17] Graaf, B., Van Dijk, H., van Deursen, A.: Evaluating an embedded software reference architecture -industrial experience report-. In: CSMR 2005. Ninth European Conference on. pp. 354–363. IEEE (2005)

[18] Glass, R. L. (1994). The software-research crisis. Software, IEEE, 11(6), 42-47.

[19] JIRA. http://www.atlassian.com/es/software/jira (2013) [Online; accessed 14-Jul-2013]

[20] Khurum, M., Gorschek, T., Petersson, K.: Systematic review of solutions proposed for product line economics. Strategic Decision Support for Software Intensive Product Management p. 64 (2008)

[21] Martínez-Fernández, S., Ayala, C., Franch, X., Martins, H.: Benefits and drawbacks of reference architectures. In: 7th European Conference on Software Architecture (ECSA) (2013)

[22] Martínez-Fernández, S., Ayala, C., Franch, X., Martins, H.: A framework for software reference architecture analysis and review. In: 10th Workshop on Experimental Software Engineering (ESELAW) (2013)

[23] Martínez-Fernández, S., Ayala, C., Franch, X., Martins, H.: REARM: A reuse-based economic model for software reference architectures. In: 13th International Conference on Software Reuse (ICSR) (2013)

[24] Miles, M. B., & Huberman, A. M. (1994). Qualitative data analysis: An expanded sourcebook. Sage.

[25] Muller, G., van de Laar, P.: Researching reference architectures. In: Views on Evolvability of Embedded Systems, pp. 107–119. Springer (2011)

[26] Nakagawa, E.Y., Antonino, P.O., Becker, M.: Reference architecture and product line architecture: A subtle but critical difference. In: Software Architecture, pp. 207–211. Springer (2011)

[27] Nakagawa, E.Y., Oquendo, F., Becker, M.: RAModel: A reference model for reference architectures. In: WICSA. pp. 297–301. IEEE (2012)

[28] NASA: ESDS Reference Architecture for the Decadal Survey Era. http://earthdata.nasa.gov/sites/default/files/field/document/ESDS%20Reference%20Architecture%20v1.1.pdf (2012). [Online; accessed;10-Jul-2013]

[29] Oliveira, L.B.R., Nakagawa, E.Y.: A service-oriented reference architecture for software testing tools. In: Software Architecture, pp. 405–421. Springer (2011)

[30] Redmine. http://www.redmine.org/ (2013) [Online; accessed 14-Jul-2013]

[31] Reifer, D.J.: Making the software business case: Improvement by the numbers. Addison-Wesley Professional (2001)

[32] Robson, C. (2002). Real world research: A resource for social scientists and practitioner-researchers (Vol. 2). Oxford: Blackwell.

[33] Schmid, K.: An initial model of product line economics. In: Software Product-Family Engineering, pp. 38–50. Springer (2002)

[34] Scott, D.: Gartner hype cycle for real-time infrastructure. http://www.gartner.com (2012), [Online; accessed 1-Mar-2013]

[35] Seaman, C.B.: Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering, vol. 25(4), pp. 557-572, (1999).

[36] Seddon, P., Scheepers, P.: Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples. EJIS, pp. 1-16, (2011).

[37] Sonar Source. http://www.sonarsource.com/ (2013) [Online; accessed 14-Jul-2013]

[38] The Open Group: SOA Reference Architecture. https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?publicationid=12490 (2013).

[39] Wohlin, C., Höst, M., Henningsson, K.: Empirical research methods in software engineering. In: Empirical Methods and Studies in Software Engineering, pp. 7–23. Springer (2003)

[40] Yin, R. K. (2009). Case study research: Design and methods (Vol. 5). Sage.