

Conducting Empirical Studies on Reference Architectures in IT Consulting Firms

Silverio Martínez-Fernández¹, David Ameller¹,
Claudia Ayala¹, Xavier Franch¹, Xavier Terradellas²

¹ Software Engineering for Information Systems Group
Universitat Politècnica de Catalunya (GESSI-UPC)
Barcelona, Spain
{smartinez, dameller, cayala, franch}@essi.upc.edu

² Architecture Centre of Excellence
Everis (ARCHEX)
Barcelona, Spain
xavier.terradas.fernandez@everis.com

April 2012

Report ESSI-TR-12-2

Departament d'Enginyeria de Serveis i Sistemes d'Informació

Conducting Empirical Studies on Reference Architectures in IT Consulting Firms

Silverio Martínez-Fernández, David Ameller, Claudia Ayala, Xavier Franch

Software Engineering for Information Systems Group
Universitat Politècnica de Catalunya (GESSI-UPC)
Barcelona, Spain
{smartinez, dameller, cayala, franch}@essi.upc.edu

Xavier Terradellas

Architecture Centre of Excellence
Everis (ARCHEX)
Barcelona, Spain
xavier.terradas.fernandez@everis.com

Abstract— Tight time-to-market needs pushes IT consulting firms (ITCFs) to continuously look for techniques to improve their IT services in general, and the design of software architectures in particular. The use of reference architectures allows ITCFs reusing architectural knowledge and components in a systematic way. In return, ITCFs face the need to assess these reference architectures in order to ensure their quality, return on investment and incremental improvement. Little support exists to help ITCFs to face this challenge. In this work-in-progress paper we present an empirical framework aimed to assess ITCFs' reference architectures and their use in IT projects by harvesting relevant evidence from the wide spectrum of involved stakeholders. We are currently applying this framework in an ITCF and we report the issues found so far.

Keywords— *Software architecture, reference architecture, empirical software engineering.*

I. INTRODUCTION

Nowadays, the size and complexity of information systems (IS), together with critical time-to-market needs, demand new software engineering approaches to design software architectures (SA) [13]. One of these approaches is the use of reference architectures (RA) that allows to systematically reuse knowledge and components when developing a concrete SA [5][11].

As defined in [13], an RA “encompasses the knowledge about how to design concrete architectures of systems of a given application [or technological] domain; therefore, it must address the business rules, architectural styles [...], best practices of software development [...], and the software elements that support development of systems for that domain”.

Due to their reusable nature, RAs are becoming a key asset of information technology consulting firms (ITCFs). Therefore, their exhaustive assessment (e.g., in terms of quality, cost and time reduction) becomes necessary. The goal of this paper is to present an empirical framework aimed to assess the RAs used by ITCFs in their IT projects executed in client organizations. This framework could be used by ITCFs to drive improvements on their RAs.

The paper is structured as follows. In Section 2 we present the context of our proposal. In Section 3 we describe the fundamental aspects of RAs that are suggested to be assessed. In Section 4 we describe the empirical studies that compose the framework. In Section 5 we present the ongoing application of the framework in the context of an ITCF. In Section 6 we end up with conclusions and future work.

II. CONTEXT OF IT CONSULTING FIRMS

We are interested in the case in which an ITCF has designed an RA with the purpose of deriving SAs for client organizations. This usually happens when the ITCF is regularly contracted to create or maintain ISs in client organizations. Each IS is built upon the derived SA (we call it *RA-based SA*) and includes many enterprise applications implemented on top of this SA (*SA-based enterprise applications*), see Fig. 1.

The use of RAs allows ITCFs to reuse their architectural knowledge and software components (normally associated to particular technologies) for the design of RA-based SAs in client organizations. Thus, a good RA guarantees a certain level of quality for each RA-based SA. Resulting RA-based SAs provide a baseline that facilitates standardization and interoperability as well as the attainment of business goals during enterprise applications' development and maintenance.

In the scenario depicted in Fig.1, there are three kinds of projects with different targets: 1) RA projects; 2) RA-based SA projects; 3) SA-based enterprise application projects. Each kind of project has its own stakeholders who need to be clearly defined for assessment purposes [1]. RA projects are run exclusively by an ITCF team, specialized in architectural knowledge management. RA-based SA projects involve one ITCF team and likely another team from the client organization; their members are specialised in architectural design and have relevant knowledge of the organisation business domain. Finally, SA-based enterprise application projects can involve teams from the client organization and/or subcontracted ITCFs (which may even be different than the RA owner) whose members are usually very familiar with the specific organisation domain. The participation of the client organization in these two last types of projects is one possible strategy for ensuring the continuity of their ISs without having much dependency on the ITCF.

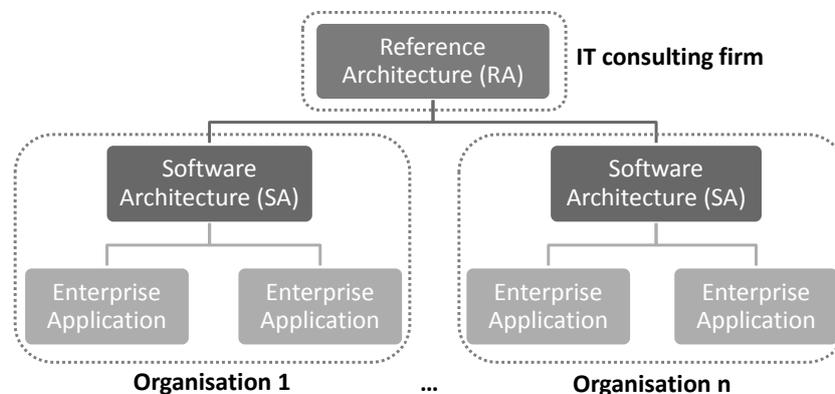


Figure 1. Relationship among RAs, SAs and enterprise applications.

III. RELEVANT ASPECTS OF REFERENCE ARCHITECTURES

In this section we identify important aspects to assess RAs. In [1], Angelov et al. state that SAs and RAs have to be assessed for the same aspects. For this reason, we started by analysing some available works on SA assessment [3][7]. However, due to the generic nature of RAs, some of the aspects found for SA assessment were not directly applicable to RA assessment. Therefore, we elaborated further this analysis considering both the specific characteristics of RAs as described in [1][10][11][13] and our own experience in the field. The resulting aspects for assessing RA are detailed below and summarized in Table I.

Aspect 1 refers to the need of having an *overview of the RA*. It includes an analysis of its generic functionalities, its domain [1], its origin and motivation, its correctness and utility, and its support for efficient adaptation and instantiation [11].

Falesi et al. [7] and other studies such as [10] highlight the importance of *requirements analysis and quality attributes*, as well as *decision-making and architectural evaluation* for the SA design process. These two aspects should also be considered for the RA assessment because, as we said, SAs and RAs have to be assessed for the same aspects [1]. Thus, we considered them as Aspects 2 and 3 respectively. However, since an RA has to address more architectural qualities than an SA (e.g., applicability) [1], this analysis could be wider for RAs in this sense.

SAs also address *business qualities* [1] (e.g., cost, time-to-market) that are business goals that affect their *competence* [3]. It is also applicable to RAs, so it is considered as Aspect 4.

To improve the SA design process, there also exist supportive technologies such as *methods*, and *techniques and tools* [7][13]. Thus, it is not only important for an RA to collect data to assess its design process, but also its supportive technologies, which are assessed by Aspects 5 and 6.

As stated in [7], a crucial aspect to define the goodness of a SA is related to *the Return on Investment (ROI)*. The optimal set of architectural decisions is usually the one that maximizes the ROI. Aspect 7 is intended to quantify benefits and costs of RAs to calculate their ROI.

We recommend gathering evidence about all these aspects, which are summarised in Table I, while assessing an RA. Existing methods for SA assessment have been previously applied for RA assessment, such as in [1] and [10]. However, up to our knowledge none of them cover all the aspects of Table I. Hence, new approaches to assess RAs considering these aspects altogether are required. This has motivated our work.

TABLE I. SUMMARY OF RELEVANT ASPECTS FOR RA ASSESSMENT

Aspect	Description of the Architectural Aspect
1	Overview: functionalities [1], origin, utility and adaptation [11]
2	Requirements analysis [7], also called quality attributes [1][10]
3	Architectural knowledge and decisions [7][10][13]
4	Business qualities [1] and architecture competence [3]
5	Software development methodology [7][13]
6	Technologies and tools [7][13]
7	Benefits and costs metrics to derive SAs from RAs [7]

IV. AN EMPIRICAL FRAMEWORK TO ASSESS REFERENCE ARCHITECTURES

In this section, we present the ongoing version of our empirical framework to assess RAs (considering the relevant architectural aspects presented in Table I) in the context described in Section II. The framework aims to serve as a point of reference for practitioners (mainly ITCFs) that need to assess their RAs, who can apply it by themselves or with the support of a research team, as in our case (see Section V).

The pillars of the framework are the guidelines for conducting empirical studies in software engineering recommended by Wohlin et al. [17]. These guidelines suggest envisaging activities to understand, evaluate and improve the main object of a study, which is RAs in our case. Owing to it is impossible to start improving directly in most cases [17], the current version of the framework has mainly addressed the activities of *understanding* and *evaluating* RAs.

On the one hand, in order to *understand* the ITCF's RA setting and how well such RA is working, the framework suggest three different and complementary types of empirical studies. First, qualitative surveys and case studies aimed to gather information related to the Aspects 1 to 6 (as defined in Section III). Second, a quantitative post-mortem analysis to target the collection of metrics related to the Aspect 7. These studies gather information not only from RA projects, but also from RA-based SA projects as they are a direct outcome of the RA usage, and from SA-based enterprise application projects as they are a direct outcome of the RA-based SA usage. This allows analysing the RA's suitability for producing the RA-based SAs for the ITCF's client organizations as well as the detection of improvement opportunities.

On the other hand, in order to *evaluate* the ITCF's RA, there exist some evaluation methods in the literature (e.g., [1], [10] and the last step of [11]). These evaluation methods mainly gather information from RA projects, mostly covering Aspects 1 to 6. In our framework, we propose to apply one of these methods to evaluate the quality of an RA.

Below, the empirical studies that have been envisaged for our framework are explained. We use the same structure as in [7]: context and motivation, objectives, method and expected results. Table II summarises the studies that compose the framework and gives some guidelines to support their conduction.

A. Qualitative surveys to understand the current situation

Context: Before deciding to launch an RA-based SA project (or improving an RA), it is needed to understand RA's characteristics, as well as its potential benefits and limitations. Assessing previous RA-based SA projects is a feasible way to start gaining such an understanding.

Objective: To understand the impact of using an RA in RA-based SA projects in the client organisations.

Method: Exploratory surveys with personalised questionnaires applied to relevant stakeholders (e.g., leader, architect, developer) to gather their perceptions and needs.

Expected results: To get an understanding of the impact and suitability of the RA for the elaboration of RA-based SA projects. Improvement insights can also be identified from different stakeholders.

B. *Quantitative post-mortem analysis to calculate ROI*

Context: Before investing in an RA-based SA, project business leaders from prospective client organisations need to analyse whether undertaking or not the investment. Offering them evidence from former projects can help them to make more informed decisions.

Objective: To assess whether it is worth investing in an RA-based SA.

Method: An exploratory quantitative post-mortem analysis. It is aimed to quantify the potential advantages and limitations of using an RA-based SA. Results from the qualitative survey detailed above are an important input for the design of this study. Some examples can be found at [9] [16].

Expected results: A quantitative report that supports business leaders to make informed decisions.

C. *Case studies to seek an explanation of the situation*

Context: Potential actions need to be envisaged after identifying potential problems, improvement opportunities and ROI information from the previous empirical studies.

Objective: To formulate hypotheses from the two previous studies and run case studies to seek an explanation for the intended hypothesis.

Method: Explanatory case study design to seek and explanation of previously identified situations.

Expected results: Feedback and interpretation with respect to previously identified situations from a case study in which the RA has been used.

D. *RA Evaluation to prove its effectiveness*

Context: A positive evaluation of the RA would prove its effectiveness and quality. Hence, an RA must be evaluated to justify its use. The three previous studies offer potential insights for RA improvements for that cases in which the quality of the RA is not sufficient.

Objective: To evaluate the RA.

Method: An existing empirical method to evaluate RAs such as [1], [10] and the last step of [11].

Expected results: An evaluation of the RA to analyse its effectiveness and to determine which improvements should be incorporated in the RA.

E. *Summary of the empirical studies of our framework*

Table II summarises the characteristics of the empirical studies of our framework. The first column shows in which step of the guidelines of [17] the study applies. The second column indicates its type. The third column points out the stakeholders that are involved in the study. Characteristics of the empirical methodological approach are in the fourth column. It comprises three characteristics [17]: the purpose of the study, which could be exploratory, descriptive, explanatory or improving; the collected data that may be quantitative or qualitative; and the research design that may be fixed, semi-fixed or flexible. The fifth column describes the main goal of each study. The sixth column emphasizes the aspects from Table I that should be covered by the study, which depends on its data collected (i.e., qualitative or quantitative). In the last columns, existing guidelines (both general and specific for software engineering) to conduct the studies are recommended.

It is important to note that the empirical studies suggested by our framework are complementary and support each other. Our framework benefits from this combination of studies. For instance, collecting data from different studies allows triangulation (i.e., data validation). Also, results from a preceding empirical study can be used to corroborate or develop further these results (e.g., using an explanatory case study to find out why the results from an exploratory survey are as they are). For this reason, the suggested studies have been designed to be conducted sequentially.

TABLE II. EMPIRICAL STUDIES OF THE FRAMEWORK TO ASSESS REFERENCE ARCHITECTURES

Step	Type of study	Study's stakeholders	Methodology characteristics	Goal	Aspects targeted	Guidelines	
						General	For SE
Understand	Survey	Several RA-based SA teams, several SA-based enterprise application teams	Exploratory	To understand the scenario from each stakeholder's perspective	1, 2, 3, 4, 5, 6	[12] [14]	[4] [17]
			Qualitative				
			Semi-fixed				
	Post-mortem analysis	The RA team, several RA-based SA teams	Exploratory	To calculate the ROI for RA-based SAs	7	[12] [14]	[6] [9] [16] [17]
			Quantitative				
			Fixed				
	Case study	An RA-based SA team	Explanatory	To seek an explanation of the situation found out in the two previous studies	1, 2, 3, 4, 5, 6	[12] [14]	[15] [17]
			Qualitative				
			Flexible				
Evaluate	RA Evaluation	The RA team	Improving	To evaluate the RA quality	1, 2, 3, 4, 5, 6	n/a	[1] [7] [10] [11]
			Qualitative				
			Fixed				

V. INSTANTIATION OF THE FRAMEWORK: THE EVERIS CASE

The presented empirical framework is currently being applied at the Architecture Centre of Excellence of Everis, a multinational ITCF. The main motivation of Everis for conducting the empirical studies is twofold: 1) technical: identifying strengths and weaknesses for their RA; 2) strategic: providing evidence to their clients about the potential benefits of applying their RA. Everis fits into the context described in Section II, e.g., they carry out the three types of projects described there.

Following the criteria found in [1], Everis' RA can be seen as a Practice RA, since it is defined from the accumulation of practical knowledge. According to the classification of [2], it is also a classical, facilitation RA for multiple organisations designed by a software organisation in cooperation with user organisations. It is classical because its creation is based on experiences, and its aim is to facilitate guidelines for the design of systems, specifically for the IS domain. Fattah presents in [8] another classification scheme that would consider it as an enterprise RA because it is "a blueprint for the Solution Architecture [RA-based SA] of a number of potential projects [SA-based enterprise applications projects] within an organisation that embodies [...] principles, policies, standards and guidelines".

All the studies presented in this paper are planned to be conducted to assess Everis' RA. The survey protocol has already been designed and reviewed. On the other hand, the post-mortem analysis to calculate the RA's ROI is currently under design. The roles of the different stakeholders and an excerpt of the survey protocol are shown below. The complete version of the survey protocol is available at <http://www.essi.upc.edu/~gessi/ecs12-survey-protocol.pdf>.

A. Mapping between studies and stakeholders

As it was already said, stakeholders need to be clearly defined for RA assessment purposes [1]. In Everis' projects, there are four kinds of stakeholders in both ITCF and client organisation teams: project business leader, project technological leader, software architect and developer. Each of these stakeholders has a vested interest in different architectural aspects, which are important to analyse and reason about the appropriateness and the quality of the three kinds of projects [10]. Table III shows how the roles are covered by the different studies in the Everis case.

B. The survey protocol of the Everis case

1) Sampling. The target population of this survey are RA-based SA projects and SA-based enterprise application projects. A representative sample of these projects in several client

organisations has been selected. Table III indicates with an ‘S’ the roles that will be interviewed in each project.

2) Approach for data collection. On the one hand, semi-structured interviews will be used for Project Technological Leaders and Software Architects, and Client’s Project Business Leaders. The reason of using interviews is that these roles have higher knowledge than the other roles about the architectural aspects of the Table I, or another perspective in the case of Client’s Project Business Leaders, so we want to collect as much information as possible from them. Prior to the interviews, questionnaires might be delivered to collect personal information about the interviewee and to inform him/her about the interview. On the other hand, online questionnaires will be used for RA-based SA Developers and SA-based enterprise application Developers, since most of their questions are about supportive technologies and their responses can be previously listed, simplifying the data collection process.

TABLE III. STAKEHOLDERS OF THE EVERIS CASE

Project ^a	ITCF Team				Client Organization Team			
	<i>PBL</i>	<i>PTL</i>	<i>Arc</i>	<i>Dev</i>	<i>PBL</i>	<i>PTL</i>	<i>Arc</i>	<i>Dev</i>
RA	E	E	R, E	E	n/a	n/a	n/a	n/a
SA	C	S, C	S, R, C	S, C	S, C	C	C	C
Application	n/a	n/a	n/a	S	n/a	n/a	n/a	n/a

a. Legend: Project Business Leader (PBL), Project Technological Leader (PTL), Software Architect (Arc), Developer (Dev), Survey (S), ROI study (R), Case study (C), and RA evaluation (E).

VI. CONCLUSIONS

Driving empirical studies is becoming one of the main sources of communication between practitioners and the academia. The main contribution of this work-in-progress paper intends to be the formulation of a framework to conduct empirical studies for assessing RAs. It consists of a list of relevant aspects for RAs assessment, and an assortment of four complementary empirical studies that allow assessing these aspects. The framework can be adapted to the specific context of ITCFs. Consequently, practitioners that apply the framework in their ITCFs, either by themselves or through collaboration with researchers, could benefit from a common reference framework to assess RAs.

Future work spreads into two directions. In terms of validation, we are conducting the Everis case using our framework, getting feedback for assessing its effectiveness. With respect to this first version of the framework, we aim to extend it considering Wohlin’s improvement step (see Section IV) in order to build preliminary guidelines for improving RAs in ITCFs.

ACKNOWLEDGMENTS

This work has been supported by the “Cátedra Everis” project. We would also like to thank Matthias Galster for his comments and suggestions.

REFERENCES

- [1] S. Angelov, J.J.M. Trienekens, P. Grefen, “Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case,” ECSA 2008.
- [2] S. Angelov, P. Grefen, D. Greefhorst, “A Framework for Analysis and Design of Software Reference Architectures,” Information and Software Technology, 54(4), 2012.
- [3] L. Bass, P. Clements, R. Kazman, M. Klein, “Evaluating the Software Architecture Competence of Organisations,” ECSA 2008.
- [4] M. Ciolkowski, O. Laitenberger, S. Vegas, S. Biffel, “Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering,” ESERNET 2003.
- [5] R. Cloutier et al., “The Concept of Reference Architectures,” Systems Engineering, 13(1), 2010.
- [6] B. Collier, T. De Marco, P. Fearey, “A Defined Process for Project Post Mortem Review,” Software IEEE, 13(4), 1996.

- [7] D. Falessi, M. Ali Babar, G. Cantone, P. Kruchten, "Applying Empirical Software Engineering to Software Architecture: Challenges and Lessons Learned," *Empirical Software Engineering*, 15(3), 2010.
- [8] A. Fattah, "Enterprise Reference Architecture," *Via Nova Architectura*, 2009.
- [9] Forrester Research, "Forrester's Total Economic Impact (TEI)", available online at: www.forrester.com/TEI, last accessed: Dec 2011.
- [10] B.P. Gallaguer, "Using the Architecture Tradeoff Analysis MethodSM to Evaluate a Reference Architecture: A Case Study," DTIC Document, 2000.
- [11] M. Galster, P. Avgeriou, "Empirically-grounded Reference Architectures: A Proposal," *QoSA+ISARCS*, 2011.
- [12] D.E. Gray, "Doing Research in the Real World," SAGE Publications Ltd, 2009.
- [13] E. Y. Nakagawa, P. O. Antonino, M. Becker, "Reference Architecture and Product Line Architecture: A Subtle But Critical Difference," *ECSA 2011*.
- [14] C. Robson, "Real World Research: A Resource for Social Scientists and Practitioner-Researchers," Blackwell Publishers, 2002.
- [15] P. Runeson, M. Höst, "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," *ESEM 2008*.
- [16] R. van Solingen, "Measuring the ROI of software process improvement," *Software IEEE*, 21(3), 2004.
- [17] C. Wohlin, M. Höst, K. Henningsson, "Empirical Research Methods in Software Engineering," *ESERNET 2003*.