Designing a Software Architecture for a Railway Safety Platform

Adailton Lima[1], Rodrigo Reis[1], Melina Alberio[2], Carlos Lopes[1], Cleidson de Souza[1,2]

[1]Software Engineering Laboratory (LABES) Federal University of Pará (UFPA) Belém, Pará, Brazil

[2]Vale Institute of Technology (ITV) Belém, Pará, Brazil

Abstract. *We were faced with the challenge of designing solutions to increase the safety of all people involved with two corporate railways: employees working on transportation and maintenance services and citizens who live along the railroads. Instead of focusing on different solutions, we want to design a software platform that will allow the evolution of a software ecosystem. In this paper, we present the initial design and assessment of the software architecture of this platform. We briefly report the current architecture itself, focusing on some of the design decisions we made as well as our evaluation of the ATAM method to support the design of software platforms.*

1. Introduction

A large mining company headquartered in South America funded us to develop new information and communication technologies aiming to increase the safety of employees and citizens who work or live close to the companies' railways. In addition to the risks to the company's employees and contractors, there are several cities and communities along these railways. This means that the technologies being designed in this project should not only improve the safety of company's employees, but also minimize the risks to the people who live along the railways.

The authors then faced an interesting challenge: *how to design technology to enhance the safety of company's employees, contractors and citizens who live along these railways?* The typical approach would be to identify the main risk scenarios and then design solutions that were specific to these scenarios. However, instead of doing that, we have decided to build a software platform to provide a broader set of services that can be used by different solutions for railroad safety. We then faced the problem of designing a software platform as a problem of designing a complex software system with a number of important questions related to the design of its underlying software architecture. As such, we chose to adopt a method for designing software architectures; in this case, we adopted the ATAM (Architecture Tradeoff Analysis Method) method [Kazman 2002]. This paper reports the software architecture we designed including the design decisions we have made, as well as an assessment of the ATAM as a method for designing software platforms.

2. Software Ecosystems, Platforms and their Design

According to [Bosch 2009], "a software ecosystem consists of a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs". The question then is *how to design such a software platform*? How does the underlying software architecture of this platform should look like? What are the recommendations, guidelines or methodologies that can be used to design such a software platform?

Related work to ours can be grouped in a small number of groups. In the first group, we find papers describing how organizations moved from a software product to a software (platform) ecosystem. Examples of this approach include [Costa et al. 2013] and [Bosch 2009], who discuss how to move from a software product line to a software ecosystem. Another group of related work focuses on modeling software ecosystems as a way to understand how they evolve over time (see for instance, [Monteith 2013]). Finally, the third set of related work focuses on the software architecture of software ecosystems. This includes the analysis of how extensible the APIs of these platforms are and the architectural challenges faced in software ecosystems [Bosch 2009].

Since we were not able to find an approach that could fit our particular context, we decided to adopt the ATAM method to guide the design of our software architecture. We chose ATAM because it is a fairly known method for architectural definition and assessment. In the following section we will very briefly describe our usage of the ATAM method.

3. Using the ATAM Method

As a process of "risk identification", the ATAM can be conducted in an early stage of the project, where there is no detailed information or implementation available but requirements that are used to start a discussion on the decisions about the software architecture. One of the first steps we took was to identify the stakeholders related to the railroad safety platform development. The main stakeholders are:

- **S1**: Company's information technology (IT) group responsible for the deployment of new information and communication technologies;
- **S2**: Researchers developing new information and communication safety solutions that will eventually be integrated into the software platform; and
- **S3**: Internal or external software developers that need to access platform services and data to create new safety solutions.

S1 stakeholders are aware of the current infrastructure constraints and future expansion plans. Their strategic view about the IT infrastructure is essential to understand the limits that new safety services and technologies will face. Their participation is also important to provide an overview of possible new technologies needed to upgrade the IT services of the company. The participation of S2 stakeholders is essential to identify the services, data and quality attributes that the software platform must provide. Finally, S3 are stakeholders responsible for creating new safety solutions that will exist on the platform. These solutions might be *clients* of data/services provided by other solutions or *providers* of data/services to other solutions.

Most of the identified quality attributes are related to the communications infrastructure that allow the services and data providers to use channels to provide online monitoring and notification in the railway. With these quality attributes we designed an early version of the software architecture based on a component and layered view of services, devices and applications involved in the safety platform. This step was very important to create a shared view that included all components working together, since before that, the team only had a vague notion of the required services, without knowing how to integrate them.

Following the method, we interviewed an experienced S1 stakeholder. This stakeholder is a member of a larger team that is defining new communication technologies to be acquired and deployed on one of the company's railways. His participation was important to validate the assumptions and quality attributes that should be provided by the communication services to support the actual railway safety platform. This was critical due to the geographical distance between our team and the IT infrastructure team (other S1 members). To complicate things, this railway crosses the Amazon rain forest, and no detailed information about its infrastructure was available at the beginning of the project. Through the S1 stakeholder, we confirmed the existence of maintenance shafts distributed alongside the railway. However, not all of them have energy and network connectivity to guarantee high availability, one of the initial platform requirements.

As a result of the analysis of the interviews and initial architecture design, we defined different alternative scenarios for the designed architecture. In one of them we moved distributed services to a central server. This led us to an architectural tradeoff, where we must decide between (i) the cost to create the infrastructure to support local servers spread along the railroad and (ii) the cost to communicate messages and notifications in only one central server with a potential bottleneck for system performance. This is only one example of an architectural trade-off that we faced as the result of the ATAM method. Due to space constrains we cannot report all of them.

4. Assessing the ATAM Method

According to [Taylor 2013], a successful software ecosystem is the result of good architectural styles and design decisions in order to create an open environment that supports the success criteria of the platform. This does not mean that a good architectural style is sufficient for a successful software ecosystem, but instead it is an important and necessary condition. In fact, there are social and economical factors that influence the development of software ecosystems [Barbosa 2013].

Before we conducted the design and analysis process, the stakeholders were only aware of their "safety apps", i.e., the software and/or hardware solutions they were implementing. At the time, none of them have thought about how their assumptions were affected by other components or the existing IT infrastructure. As a practical result, S2 and S3 stakeholders are now researching alternative designs to support their solutions because of the likely scenario of lack of communication with local servers.

It is important to mention that the same strength of ATAM is also a weakness. To be more specific, ATAM suggests that workshops and/or interviews should be conducted with important stakeholders, and especially software developers who will implement the architecture and, in our case, create solutions for that platform. However, one of the most important aspects of any software ecosystem is the possibility to tap into the talent of software external developers, i.e., developers who will create solutions that have not been imagined before. This means that the ATAM results can help explore the architectural space of the "known" solutions reported by the interviewed developers as well as solutions that are "similar" to these. If new solutions challenge the assumptions embedded in this architectural space, the architecture of the software platform might fail. Despite that, we still believe that ATAM provided an interesting starting point for us because it guided us through the process of designing the architecture of the software platform by documenting requirements and forcing us to think about design trade-offs.

5. Conclusions

An important aspect of any software ecosystem is its software architecture [Taylor 2013]. This architecture needs to be open and flexible to allow software developers contribute with new and innovative solutions. While the research community recognizes the importance of this software architecture, most previous work focuses on the evolving a software platform from a specific software product [Bosch 2009] [Costa 2013]. In other words, there is limited work suggesting how a software architecture should be designed when creating a new software ecosystem.

In this paper, we describe the usage of the ATAM method [Kazman 2012] to guide the definition and assessment of a software platform that is being built to increase the safety of employees and inhabitants who work and live alongside two major railways in Brazil. We also report on our evaluation of the ATAM method in this context, i.e., the advantages and weaknesses of using ATAM to design a software platform for ecosystems.

ACKNOWLEDGMENTS

We would like to thank the funding from CNPq (process number 310468/2014-0) and Chamada 59/2013 MCTI/CT-Info/CNPq (process number 440880/2013-0).

REFERENCES

Barbosa, O., Santos, R., Alves C., Werner, C., Jansen, S. “A Systematic Mapping Study on Software Ecosystems through a Three-dimensional Perspective”. In: Software Ecosystems: Analyzing and Managing Business Networks in Software Industry. Edward Elgar, Cheltenham, UK. (2013).

Bosch, J. From software product lines to software ecosystems, Proceedings of the 13th International Software Product Line Conference, 2009, San Francisco, California.

Cataldo, M. and Herbsleb. J. D. Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (ECSA '10), Carlos E. Cuesta (Ed.). ACM, New York, NY, USA, 65-72.

Costa, G., Silva, F. et al., 2013. From applications to a software ecosystem platform: an exploratory study. In *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems* (MEDES '13). ACM, New York, NY, USA, 9-16.

Kazman, R.; Klein, M.; Clements, P.; Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley. SEI Series in Software Engineering, 2002.

Kazman, R.; Michael Gagliardi, William Wood, Scaling up software architecture analysis, Journal of Systems and Software, Volume 85, Issue 7, July 2012, pp. 1511-1519.

Monteith, J. Y. and McGregor, J. D. Hadoop and its evolving ecosystem. In Proceedings of the Fifth International Workshop on Software Ecosystems, 2013.

Taylor, R. N. The Role of Architectural Styles in Successful Software Ecosystems. In: Proceeding of the 17th International Software Product Line Conference, NY, 2013.