







A Survey on the Skills, Activities and Role of the Software Architect in Brazil

Manoela R. Oliveira¹ , Felipe J. R. Vieira¹ , Sanjay Misra² ,
and Michel S. Soares¹ 

¹ Federal University of Sergipe, Av. Marechal Rondon, s/n - Jardim Rosa Elze,
São Cristóvão, Brazil

manoelareisoliveira@gmail.com, felipejrvieira@gmail.com,
michel@dcomp.ufs.br

² Covenant University, KM 10 Idiroko Rd, Ota, Nigeria
sanjay.misra@covenantuniversity.edu.ng

Abstract. Although the skills and knowledge of software architects have already been the subject of some studies in recent years, researchers and practitioners still have not come to a clear consensus about the activities that a software architect is often responsible in practice in order to be considered successful. In recent years, due to occurrence of successive changes and evolution of new technologies, the roles of the architect and even practices related to software architecture have been continuously changed in the software development life cycle. The software architect is expected to possess a diversity of skills. In addition to technical knowledge, domain knowledge and communication skills must be considered. However, there are many job offerings for this position which have in their description skills and roles totally different from the ones already known and considered essential by academic and industry studies. In order to better understand what software architects actually do in their daily activities in practice, and how this resembles or distances themselves from the skills, roles and knowledge cited in the literature as essential, in this article we have conducted a large-scale survey with 536 professionals who currently work or have worked at some period in their careers as software architects. Among the results, it is clear that the roles, responsibilities, activities and tasks performed by software/system architects are still largely unknown and diffuse in organizations, as important tasks to be performed by software architects are still not common sense in industry.

Keywords: Software architect · Industry practice · Large-scale survey research

1 Introduction

The discipline of Software Architecture emerged in the past two decades in academia [19] and more recently in software industry [4, 8, 21, 24]. A large number of patterns, frameworks, methods, tools and languages related to activities of

software architecture have been proposed within this period. Even a well-known standard, the ISO/IEC/IEEE 42010 [13], was published. This International Standard specifies the manner in which architecture descriptions of systems are organized and expressed, which confirms the importance of software architecture as a discipline of software engineering.

The role of the software architect has become increasingly common in industry, but the activities, tasks, processes and daily chores related to the software architect became a point of discussion. For instance, do software architects develop source code? How do their work relates to the work of project managers? How long does it take to someone become a (real) software architect? These questions are still subject of debate in academia and industry [1, 8] in which patterns and “antipatterns” have been suggested [11, 17].

Given the improved importance of software architecture in academic research and industrial practice it is necessary to better understand this role. In this paper we describe a survey about the daily activities performed by professionals who work as software architects in Brazil. The goal is to better understand what software architects actually do in their daily activities in practice, and how this resembles or distances themselves from the skills, roles and knowledge cited in the literature as essential.

The research question in this paper is stated as follows:

RQ - To what extent Software Architect professionals in Brazil perform software architecture activities?

Within this RQ, the idea is to describe, given a number of activities that are known as responsibilities and skills of software architects, searched in the literature, which ones are most performed by software/systems architects working in Brazil.

2 Patterns and Antipatterns of the Software Architect Job

Despite past efforts, we still have little understanding and there is no clear consensus of what software architects do in real-world settings [7]. Several authors have defined software architect patterns and anti-patterns over the past years. Koenig introduced the concept of antipattern in [15], by defining an antipattern just like a pattern, except that instead of a solution it gives something that looks superficially like a solution but is not [15].

Krutchén, in [17], defined as the main antipatterns of software architects: (i) creating a perfect architecture for the wrong system, (ii) creating a perfect architecture, but too hard to implement, (iii) architects in their ivory tower, and (iv) the absent architects. These antipatterns means, respectively, (i) a software architect who is not communicating has a greater probability of not understanding or only gradually understanding the software concerns, (ii) a software architect who does not understand the capability of the implementation team(s) that will create enormous levels of stress and frustration, and will most likely not deliver a quality product in time, (iii) architects that are isolated, not getting

enough input from the users and developers, and (iv) the software architect that comes with a complete architecture, out of the blue, which does not relate well to actual problems described in many organizations.

All these antipatterns will experience rejection, because developers must wait for architectural decisions, and no or little architecture design progress is made. Software architects are always away doing other things or fighting fires instead of taking care of architecture. The whole architecture will suffer from immaturity.

Regarding patterns, we have gathered information about definitions of duties, skills, and knowledge of the software architect. We have used definitions from the *Software Engineering Institute* [26] as a reference and performed a literature review with emphasis mainly on articles already recognized about the roles and responsibilities of the software architect, including [11, 14, 17, 20].

After we have evaluated the gathered information, we constructed the following set of activities of the software architect which are considered patterns that the software architect should follow to be considered successful:

1. Define the architecture.
2. Create documents that describe the architecture.
3. Act as a communicator of the software architecture.
4. Make sure everyone is using the proposed architecture.
5. Certify everyone is using the architecture correctly.
6. Make sure that architecture comes out in stages in a timely way so that the overall organization can make progress before it is complete.
7. Certify software and architecture are in synchronization.
8. Certify that management understands the necessary architectural details.
9. Make sure the right modeling is being done, knowing which qualities are going to be met.
10. Assist in tool and environment selection.
11. Identify and interact with stakeholders to make sure their needs are met.
12. Convince stakeholders about the value of the architectural solution.
13. Certify that architecture is not only right for operations, but also for deployment and maintenance.
14. Perform conflict resolution and analyze tradeoffs.
15. Solve technical problems.
16. Maintain and raise the morale of the architecture team.
17. Understand and plan the evolution of architecture.
18. Plan the insertion of new technologies.
19. Manage strategies for identifying and mitigating risks associated with the architecture.
20. Assist in understanding the problem domain.
21. Apply standards already used to solve past problems.
22. Keeping up-to-date with the latest trends.
23. Participate in the planning of projects.
24. Manage individuals and teams.
25. Establish architectural decisions.
26. Think about what impact the decisions have on the current architecture.

27. Act as a technology consultant.
28. Full software life cycle involvement.
29. Assisting product marketing and future product definitions.

3 The Survey

The following sections describe the methods used for conducting the survey.

The web questionnaire for this survey was designed and hosted by the online Google survey service, Google Forms. The questionnaire is composed of thirty-six questions, divided into two different parts. The first part is proposed in order to gather descriptive statistics to summarize the backgrounds of the respondents, and the second part of the survey focused on questions about the functions performed by the software architect in order to answer the research question presented in Sect. 1.

The first six questions are present in part I of the questionnaire, as follows:

1. What is your current job?
2. What is your level of formal education?
3. Number of years of experience in Information Technology related jobs.
4. Number of years in your current position.
5. How many projects are you involved simultaneously in your current job?
6. Do you work currently in which state in Brazil?

The other twenty-nine questions are relative to the twenty-nine patterns presented in Sect. 2. Each activity is measured using multiple response items scored on a 5-point Likert-type scale: 1- Never, 2- Rarely, 3- Sometimes, 4- Often, and 5- Always. The twenty-nine questions about the software architect functions were formulated as affirmative sentences, each sentence representing an item of the patterns found. All Thirty-five questions of parts I and II are mandatory, however we have created a thirty sixth optional and subjective question with the purpose of gathering comments on the routine as a software architect and suggestions for the research.

The first draft of the research instrument was tested and validated by four software architects, all with more than five years of experience in the area, in order to evaluate the understanding of the questions and suggest possible improvements.

After making adjustments from the pilot study and create the revised final version of the questionnaire, the survey was performed between August and September of 2018. Initially, a questionnaire was sent to 3187 professionals who declared their current position on the LinkedIn platform in Brazil as a software architect.

LinkedIn does not allow unlimited inbox messages to be sent to people who are not connected. Therefore, it was necessary to send invitations individually. Each invitation allowed to send a text with a maximum of 300 characters, thus it was necessary to summarize the purpose of the study and the request to participate in the research on the work of the software architect within this small amount of characters.

Then the authors opened each profile individually, clicked on “Connect” and then on “Add Note”, inserted the message requesting the answer to the questionnaire and finally clicked on “Send Invitation”.

The survey was anonymous, and participation in the survey was voluntary. All invited respondents were informed about the survey with an invitation text describing goals of the study. In addition to the 3187 invitations, the authors have had also the opportunity to submit the questionnaire to the architectural committee of one of the largest software companies in Brazil (a total of forty five members of the committee, about fifteen thousand employees working in Brazil and twenty five thousands all around the world).

Thus, we had a total of 3232 invitations and 536 responses, achieving a response rate of approximately 16.6%. Of the total of 536 responses, 72 respondents also answered the optional question.

4 Results

In this section, the results of our study are presented. We first presented the results of characterization questions. For each question, a graphic was created with the amount of answers referring to each alternative of the question.

Figure 1 depicts the level of education of the study participants. Most study participants have a MBA or other post-graduation course (42,4%). This is an important fact, as it is essential that they stay up to date with new trends and many should have sought better qualification through a professional MBA.

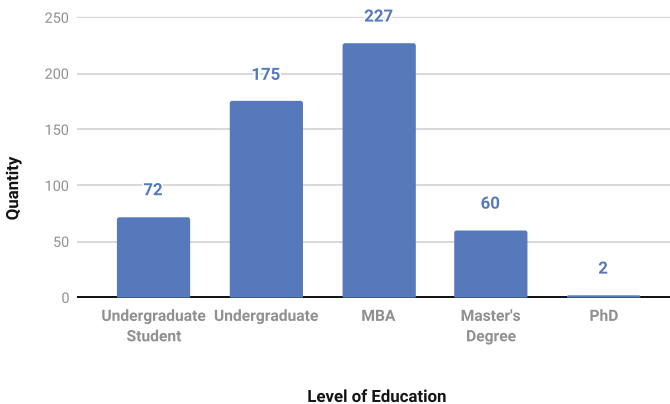


Fig. 1. Level of education

A large part of the study participants (32,6%) have at least completed an Undergraduate course, mostly in Computer Science, Information Systems or Computer Engineering, and 11,9% completed an academic Master’s degree in Computer Science. What draws attention to these numbers is the amount of

study participants who are still attending an Undergraduate course (13,4%) and the small number of people who have a PhD (0,4%), just 2.

Figure 2 depicts the number of years of IT experience before becoming a software architect. 85.3% of respondents have more than 9 years of experience and 11.8% have between 6 and 8 years, that is, 97% of respondents have more than 5 years of experience in IT. This high percentage shows that usually software architects have been working in industry for a long time and already have personal experience with IT. Of the remaining 3%, 2.6% have between 3 and 5 years and only 0.4% have less than 2 years of experience.

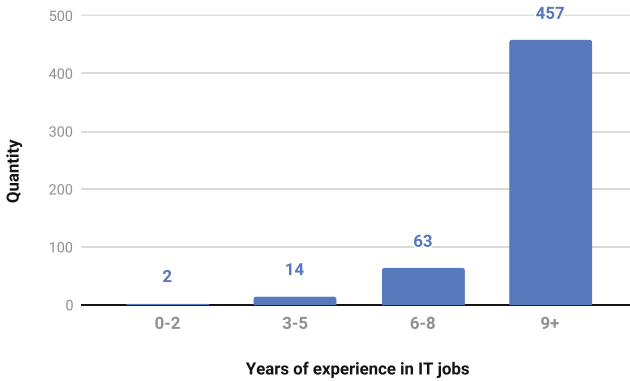


Fig. 2. Years of experience in IT jobs

Figure 3 depicts the number of years of experience in the position as software architects. All alternatives obtained a significant amount of answers. Over a quarter of the participants (26.9 %) have less than 2 years of experience, that is, they are beginning their careers as software architects. In contrast, 15.1% of the study participants are experienced architects, with more than 9 years in the position. The most expressive response, with 37.1%, was from architects with 3 to 5 years of experience, and 20.9% are architects with 6 to 8 years of experience.

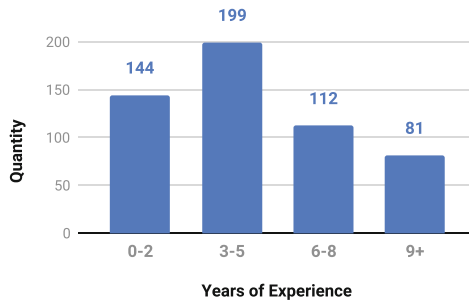


Fig. 3. Years of experience as software architect

Figure 4 depicts the current position of the participants. As suggested in [9], there may be several nomenclatures for the position of architect. It is difficult to distinguish which competencies differentiate one job from the other.

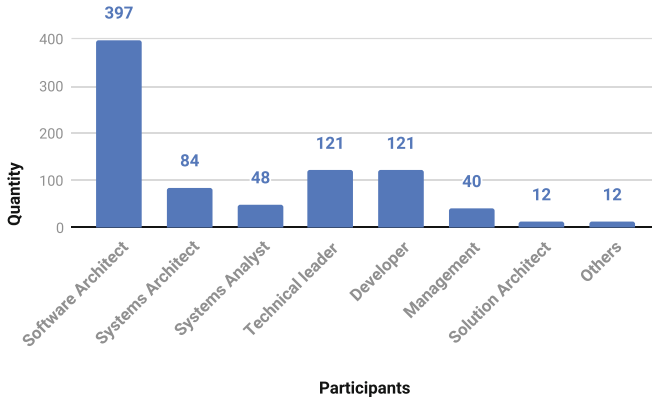


Fig. 4. Current position of participants

According to our filtering through LinkedIn, we know all participants work or have worked at some point of their career as software architects. This question aims to find out how many of them are still performing as software architects. This is a multiple choice question, allowing for more than one option, therefore the sum of percentages exceeds 100. The majority of respondents (74.1%) still works currently as software architects, 15.7% works as systems architects, 18.5% works as software developers, 22.6% works as a technical leader, and a minority works as other jobs titles.

Figure 5 depicts in how many projects the participants are involved simultaneously in their current job. For this question, all alternatives also obtained a significant amount of answers. There were three options, about a third (31,5%)

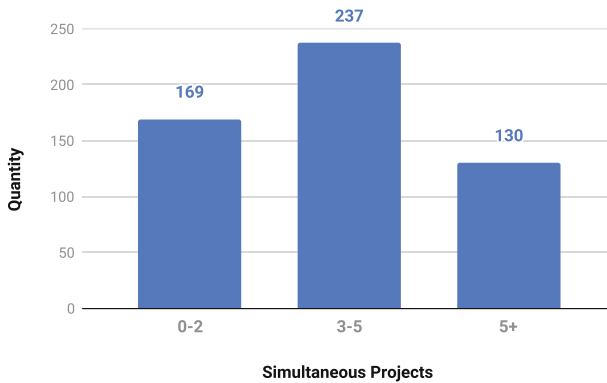


Fig. 5. Amount of projects simultaneous

of the respondents are involved with at most 2 simultaneously, it is inferred that a part of them may be working with only one project. 44.2% are involved simultaneously with 3 to 5 projects and 24.3% are involved with more than 5 projects simultaneously.

5 Discussion

The results of the research provide many possibilities of reflection. In this section, we present the answers to RQ presented in Sect. 1.

To answer the research question RQ, “To what extent Software Architect Professionals in Brazil perform software architecture activities?” we analyzed the answers and statements written by study participants.

To assist and complement the answer of research question, we present Table 1 that gathers the most performed activities by the professionals in their work. Table 1 is composed of the question number, the numbering described in Sect. 2, the percentage of participants who answered the respective question using alternative “Always”, “Often” and the sum of percentages of participants who responded to the relative sentence answering “Always” or “Often”, in descending order. In this study we considered the sum of alternatives “Always” and “Often” as the total of positive responses to the question.

As can be seen in Table 1, the five activities most executed by software architects in Brazil were: “Think about the impact that decisions have on the current architecture” (26), “Apply standards already used to solve past problems” (21), “Certify that architecture is not only right for operations, but also for deployment and maintenance” (13), “Assist in understanding the problem domain” (20) and “Solve technical problems” (15).

Keeping in mind that the software architect is required to strike a balance between technical knowledge, domain knowledge and communication skills [3, 17], this set of activities corroborates strongly with what is expected from a software architect.

With the similar intention to Table 1, we created Table 2 that gathers the less performed activities by professionals in their work. We have selected only those questions that have more than 25% of negative response rate (“Rarely” + “Never” + “Sometimes”).

The five less executed activities by software architects in Brazil were: “Assisting product marketing and future product definitions” (29), “Manage individuals and teams” (24), “Create documents that describe the architecture” (2), “Make sure that architecture comes out in stages in a timely way so that the overall organization can make progress before it is complete” (6) and “Certify that management understands the necessary architectural details” (8).

As depicted in Table 1, question 26 had the highest response rate for the “always” alternative (67.20%) and ranked first in the activity ranking, totaling 95.70% of positive responses. One of the key processes in software architecture is decision making [16, 30]. Therefore, that specific question having a high positive response rate means that software architects in Brazil are probably truly

Table 1. Total positive responses

Question	Always	Often	Always+often
26	67,20%	28,50%	95,70%
21	39,20%	50,40%	89,60%
13	51,10%	35,60%	86,70%
20	38,60%	47,60%	86,20%
15	42,70%	42,00%	84,70%
22	45,70%	38,80%	84,50%
25	32,50%	51,70%	84,20%
10	43,70%	39,70%	83,40%
17	36,90%	43,70%	80,60%
1	34,90%	45,70%	80,60%
3	38,40%	39,20%	77,60%
4	35,60%	40,90%	76,50%
7	30,60%	45,50%	76,10%
18	35,80%	38,50%	74,30%
27	32,10%	41,80%	73,90%
11	33,00%	39,20%	72,20%
5	30,00%	40,70%	70,70%
9	28,70%	41,80%	70,50%
23	29,90%	39,70%	69,60%
16	31,00%	36,60%	67,60%
19	23,50%	43,70%	67,20%
14	20,30%	43,80%	64,10%
12	24,80%	38,80%	63,60%
28	23,10%	38,20%	61,30%
8	24,10%	35,30%	59,40%
6	20,10%	38,60%	58,70%
2	19,40%	35,40%	54,80%
24	21,50%	26,70%	48,20%
29	4,30%	13,40%	17,70%

concerned about the future consequences that may arise in the development and use of the system from their architectural decisions.

Only 67,20% of the respondents have said they always or often worry about risks. According to [27], it has become the architect's role to balance factors such as risk, quality, constraints, and costs when deciding whether to use, buy, or build a component is more advantageous, then this question could have had a more expressive percentage of positive responses.

Table 2. Total negative responses

Question	Sometimes	Rarely	Never	Total negative
29	29,7%	32,5%	20,1%	82,3%
24	29,4%	16,4%	6%	51,8%
2	33,6%	10,3%	1,3%	45,2%
6	29,1%	9,3%	2,8%	41,2%
8	27,8%	9,3%	3,5%	40,6%
28	26,4%	10,6%	1,7%	38,7%
12	25,9%	7,8%	2,6%	36,3%
14	27,6%	6,2%	2,1%	35,9%
19	23,7%	7,3%	1,9%	32,9%
16	23,7%	6,9%	1,9%	32,5%
23	23,7%	6%	0,7%	30,4%
9	20,3%	7,1%	2,1%	29,5%
5	22,6%	6%	0,7%	29,3%
11	21,8%	5%	0,9%	27,7%
27	16,8%	6,7%	2,6%	26,1%
18	19,4%	5,4%	0,6%	25,04%

On sentence 20, “Assist in understanding the problem domain”, the index of positive responses was high, 86,20%. It is important that the architect understands very well the domain of the problem so that he/she chooses the architectural details that represent the best solution for each case. Within the architectural decisions theme, 84.20% of respondents affirmed in sentence 25 they always or often “Establish architectural decisions”. This is a good result, since the software architecture of a system is the result of a set of architectural decisions [28], and the architect is becoming the manager of architectural knowledge, acting as a facilitator for decision-making and ensuring that decisions are appropriate and coherent to the context of action [30].

Another question with one of the highest rates of positive responses (86,70%) was sentence 13, “Certify that architecture is not only right for operations, but also for deployment and maintenance”. This is very important, since the software architecture is constantly evolving and adapting. According to Erder and Pureur [6], architects must have a long-term vision, because the products and software survive even after the end of a project. Even after the software is deployed and being used by the client, there may still be numerous maintenance or addition of new features.

Strengthening the hypothesis about keeping the architect’s eye on the future of software, about sentence 17, “Understand and plan the evolution of architecture”, 80,60% of the respondents have chosen alternatives always or often.

89,90% of the respondents also responded positively to sentence 21, “Apply standards already used to solve past problems”. This is a great result, that almost 90% of architects already have experience and have participated in other projects previously. This is a considerable result, because since most architectural decisions made by architects are made early in the project, they use their experience and abstraction skills to get them [10].

Although most of the respondents seem to be seasoned professionals, they did not stop looking for updates, 86,20% said always or often are keeping up-to-date with the latest trends (question 22), and 74,3% of the respondents always or often plan insertion of new technologies (question 18). Even with a nearly 12% difference between the two questions, this means that even the most experienced architects are still trying to keep up with new trends.

76,50% of the respondents answered positively to sentence 4, making sure everyone is using the proposed architecture, but 18,1% mentioned that they make sure only sometimes. Consequently, almost the same percentage of positive answers from the previous question (76,10%), is sentence 7, “Certify software and architecture are in synchronization”. This similarity makes a lot of sense, whereas for both to remain always synchronized, it is necessary for the architect to constantly check whether the architecture that was proposed is actually being used.

When it comes to certifying that the proposed architecture is being used correctly (sentence 5), the result was a bit different. 70,70% of the respondents answered to carry out this activity always or often. Since the number of positive responses was lower than previous sentence “Certify software and architecture are in synchronization”, the number of negative responses also increased. 22,60% of the respondents only certify sometimes, and 6,0% answered they rarely certify the architecture is used correctly.

Still following this line of reasoning involving the actual use of the architecture in practice, on question “Make sure the correct modeling is being done, knowing which qualities will be met” (sentence 9), the result is a bit more negative, 29,5% of the respondents choose negatives responses, and 70,50% choose between always or often. However, according to Tofan *et al.* [28], in their activities, architects need to consider the functional requirements and quality attributes (or non-functional requirements) of software systems. In addition, quality attributes play an important role in the decision-making process, since architects may compromise some quality attributes through tradeoffs (e.g. security versus usability) [28].

A similar example occurs when the subject is architecture definition and documentation. Considering the sentence “I define the software architecture” (sentence 1), 80,60% of respondents chose alternatives always or often, which is a great number of positive answers, but considering the content and importance of the question, since this activity is one of the main activities, and well recognized by several authors [6,17,25]. This percentage may be a little more expressive, and it is only enough to occupy the tenth position in the ranking of questions with the highest percentages of positive answers, as presented in Table 1.

In contrast to the number of professionals who responded positively to sentence 1, the same is not true for sentence 2, “I create documents that describe the architecture”. Although documented decisions facilitate the general understanding of a system, architectural knowledge sharing, and evaluation processes [18]. One troubling result is the percentage of negative responses, 45,2% of the respondents answered negatively for sentence 2 and 54,80% choose alternatives always or often. This percentage places sentence 2 within the ranking of questions with a greater amount of negative answers, as depicted in Table 2. There may be a rationale for increasing adoption of agile methodologies with greater focus on coding and less focus on documentation.

One of possible implications is that some architects seem to work isolated. More than a half (51,8%) of respondents answered negatively to manage individuals and teams, (question 24), making this question occupy the second position of the ranking of questions with higher rate of negative answers. As already described in Sect. 1, architects working in isolation is an antipattern, architects in the ivory tower.

Probably this lack of group work influenced the high rate of negative responses to sentences such as “Perform conflict resolution and analyze trade-offs” (question 14), with 35,9%, and “Maintain and raise the morale of the architecture team” (question 16), with 32,5%. Considering this separation between the architect and the team that will develop the software, it is possible that the antipattern “creating a perfect architecture, but too hard to implement” anti-pattern is reached.

One fact that promotes reflections and draws attention is that among the questions with the highest rates of negative responses, as presented in Table 2, most are composed of tasks that require the architect to be involved with the organization or other stakeholders, although sentences such as 3, “Act as a communicator of the software architecture” and 4, “Identify and interact with stakeholders to make sure the needs are met”, have received considerable percentages of positive responses, 77,6% and 72,20% respectively.

On sentence 23, “Participate in the planning of projects”, 69.60% chose positive answers, which means that many architects (about 30%) in Brazil are not being involved in initial stages of defining the software project. In fact, software architects seem to be less involved even when it comes to business and marketing, apparently an overwhelming majority never even got involved in those steps, as shown in sentence 29, “Assisting product marketing and future product definitions”.

Sentence 28, “Full software life cycle involvement” received 38,7% of negative responses. Since software architecture is the central element throughout the software life cycle [29], it is imperative that the software architect has more active participation in all phases of the software life cycle.

It is worth mentioning two other sentences about the architect’s involvement with the organization. The first is that 40,6% of the respondents answered negatively to “Certify that management understands the necessary architectural details” (sentence 8), and the second is that 36,3% answered negatively to “Convince stakeholders about the value of the architectural solution” (sentence 12).

This impaired communication with the stakeholders can lead to several problems of understanding and acceptance of the architecture. Perhaps this lack of communication has influenced the amount of negative responses (41,2%) received by the question “Make sure that architecture comes out in stages in a timely way so that the overall organization can make progress before it is complete” (sentence 6). Without transparency and good relationship, it is very difficult for the architecture to be defined in such an efficient way.

Software architects operate in a social and organizational setting with different stakeholders and forces that influence the tasks they perform [7]. As described before, a software architect who is not communicating has a greater probability of not understanding or only gradually understanding the software concerns. Architects should ensure a coherent and sustainable architecture, for which the communication and collaboration skills become very relevant [6].

Regarding the questions about knowledge and technical skills, 73,90% answered they always or often act as a technology consultant (sentence 27), 83,4% mentioned they always or often assist in issues such as selecting environments and tools (sentence 10). The issue on solving technical problems (sentence 15) was very similar (84,70%), being one of the sentences with the highest index of positive answers.

In relation to the subjective question, we separate some significant sentences written by study participants:

“In most companies, it is not always that the architect has an active voice and/or decision-making power. Unfortunately, decisions are made on a time/delivery basis, that is, no matter how much it costs or if it will work, what is important is to deploy on time and show to stakeholders that the software product is there. It is sad when you look back and see a trail of destruction that extends until the project is canceled. And when the project is not canceled, the company coexists with the problems for decades”.

“I once heard a phrase that made me think about the job. The most important role of the software architect is to transform business requirements into technical ideas. It is not necessary for the architect to develop, but mainly it is necessary that he/she is the bridge between the product management and the technical team”.

“In Brazil, in most software companies, the role of the architect is not well defined, so he/she needs to be involved in all development processes and even often develop everything, he/she has become a senior developer model that we call full stack. This practice compromises the deadlines and quality of software”.

6 Threats to Validity

In this study, we address internal, external, construct and conclusion threats to validity.

6.1 Conclusion Validity

By making explicit the thirty-six questions used in the questionnaire and the entire protocol followed to find the architects in LinkedIn and send the questionnaires to be answered, we believe that our results are valid and can be replicated in other countries.

6.2 Construct Validity

In our revision study about the roles, skills and responsibilities of the software architect, we make efforts to gather as much information as possible. Thus, we believe the found patterns represent vastly the patterns dictated by the literature in software architecture.

6.3 External Validity

External Validity refers to the strength of generalization of the study [5]. We cannot disregard the fact that the survey was answered only by professionals through LinkedIn who are working in Brazil, thus the results cannot be easily generalized. Although this is correct, it is important to notice that a large number of these professionals work in multinational companies, besides they have had international experience (MBAs, masters or PhD studied abroad), or even working experience in other countries. Besides, the number of respondents is considerable, when compared to other surveys in Software Engineering research, as for instance, 449 in [12], 258 in [2], 268 in [22] and 99 in [23].

6.4 Internal Validity

Internal Validity refers to how much the data supports the study results. A typical mistake is the misuse of statistical analysis [5]. In this study, we used basic statistics to analyze data, so threats of internal validity are minimal.

7 Conclusion

This study analyzed the understanding of software or systems architects workers about the tasks, activities and skills of the software architect at their daily work. The research consisted of the design, application, and analysis of a survey aiming at finding the understanding of how the profession of software architect is performed by professionals in Brazil.

The vast majority of questions received a high percentage of positive responses. Although some have had a significant amount of negative responses, in the big picture the architects have actually performed much of the activities that are explained in the literature, although the authors also have some differences between what they consider as the role of the software architect, there are some activities that are basically a consensus between them.

However, we could notice, mainly from the sentences of the subjective questions that there is still a lack of understanding about the role of the software architect by the companies, who end up treating the software architect as a senior developer. This study can help the IT recruiters to find professionals who actually have the skills expected from a software architect.

As for future work, the approach used within this research can be used in other countries, or even in a number of countries. Therefore, it will be possible to compare the results between countries, but also to have a higher degree of confidence with results presented in this paper.

References

1. Barroso, A.S., da Silva, J.S.M., Soares, M.S., do Nascimento, R.P.C.: Inuence of human personality in software engineering - a systematic literature review. In: ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Porto, Portugal, 26–29 April 2017, vol. 3, pp. 53–62 (2017)
2. Besker, T., Martini, A., Bosch, J.: Impact of architectural technical debt on daily software development work – a survey of software practitioners. In: 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 278–287. IEEE (2017)
3. Correa, B.: How are architects made? *IEEE Softw.* **30**(5), 11–13 (2013)
4. Cui, X.: Retrospection and perspectives on pragmatic software architecture design: an industrial report. In: Lee, R. (ed.) *ICIS 2017. SCI*, vol. 719, pp. 13–28. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-60170-0_2
5. Easterbrook, S., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. In: Shull, F., Singer, J., Sjøberg, D.I.K. (eds.) *Guide to Advanced Empirical Software Engineering*, pp. 285–311. Springer, London (2008). https://doi.org/10.1007/978-1-84800-044-5_11
6. Erder, M., Pureur, P.: What’s the architect’s role in an agile, cloud-centric world? *IEEE Softw.* **33**(5), 30–33 (2016)
7. Galster, M., Tamburri, D.A., Kazman, R.: Towards understanding the social and organizational dimensions of software architecting. *ACM SIGSOFT Softw. Eng. Notes* **42**(3), 24–25 (2017)
8. Garlan, D.: Software architecture: a travelogue. In: *Proceedings of the on Future of Software Engineering, FOSE 2014*, pp. 29–39. ACM, New York (2014)
9. Gorton, I.: Understanding software architecture. In: Gorton, I. (ed.) *Essential Software Architecture*, pp. 1–15. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-19176-3_1
10. Hohpe, G., Ozkaya, I., Zdun, U., Zimmermann, O.: The software architect’s role in the digital age. *IEEE Softw.* **33**(6), 30–39 (2016)
11. Hoorn, J.F., Farenhorst, R., Lago, P., Vliet, H.V.: The lonesome architect. *J. Syst. Softw.* **84**(9), 1424–1435 (2011)
12. Hutchinson, J.E., Whittle, J., Rouncefield, M.: Model-driven engineering practices in industry: social, organizational and managerial factors that lead to success or failure. *Sci. Comput. Program.* **89**, 144–161 (2014)
13. ISO/IEC/IEEE: *ISO/IEC/IEEE 42010 Systems and Software Engineering-Architecture Description*, 1st edn. (2011)
14. Klein, J.: What makes an architect successful? *IEEE Softw.* **33**(1), 20–22 (2016)

15. Koenig, A.: Patterns and antipatterns. *Patterns Handb.: Tech. Strat. Appl.* **13**, 383–383 (1998)
16. Kruchten, P.: Common misconceptions about software architecture. *Rational Edge* **1**, 1998–1998 (2001)
17. Kruchten, P.: A what do software architects really do? *J. Syst. Softw.* **81**, 2413–2416 (2008)
18. Kruchten, P., Capilla, R., Dueñas, J.C.: The decision view’s role in software architecture practice. *IEEE Softw.* **26**(2), 36–42 (2009)
19. Kruchten, P., Obbink, H., Stafford, J.: The past, present, and future for software architecture. *IEEE Softw.* **23**(2), 22–30 (2006)
20. Microsoft: The Role of an Architect. *Architect. J.* (15) (2008)
21. Niu, N., Xu, L.D., Bi, Z.: Enterprise information systems architecture-analysis and evaluation. *IEEE Trans. Ind. Inf.* **9**(4), 2147–2154 (2013)
22. Rahman, A., Partho, A., Meder, D., Williams, L.: Which factors influence practitioners’ usage of build automation tools? In: *Proceedings of the 3rd International Workshop on Rapid Continuous Software Engineering*, pp. 20–26. IEEE Press (2017)
23. Raunak, M.S., Binkley, D.: Agile and other trends in software engineering. In: *2017 IEEE 28th Annual Software Technology Conference (STC)*, pp. 1–7. IEEE (2017)
24. Ribeiro, F., Rettberg, A., Pereira, C., Steinmetz, C., Soares, M.: An approach to formalization of architectural viewpoints design in real-time and embedded domain. In: *2018 IEEE 21st International Symposium on Real-Time Distributed Computing (ISORC)*, pp. 59–66, May 2018
25. SEI: Community Software Architecture Definitions (2017). <http://www.sei.cmu.edu/architecture/start/community.cfm>
26. SEI: Duties, Skills, and Knowledge of a Software Architect (2017). <https://www.sei.cmu.edu/architecture/research/previousresearch/duties.cfm>
27. Spinellis, D.: The changing role of the software architect. *IEEE Softw.* **33**(6), 4–6 (2016)
28. Tofan, D., Galster, M., Avgeriou, P., Schuitema, W.: Past and future of software architectural decisions – a systematic mapping study. *Inf. Softw. Technol.* **56**, 850–872 (2014)
29. Weinreich, R., Buchgeher, G.: Towards supporting the software architecture life cycle. *J. Syst. Soft.* **85**(3), 546–561 (2012)
30. Weinreich, R., Groher, I.: The architect’s role in practice: from decision maker to knowledge manager? *IEEE Softw.* **33**(6), 63–69 (2016)