International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor (2018): 7.426

Evaluation of the Modifiability of an Evolution System Using the ATAM Method

Yesenia Polvo Saldaña[1], Guadalupe García Toribio[2], María Janai Sánchez Hernández[3], José Juan Hernández Mora[4], Higinio Nava Bautista[5], Julio Ariel Hurtado Alegría [6], César Alberto Collazos Ordóñez[7], Lucía Muñoz Dávila[8], Nicolás Alonzo Gutiérrez[9]

[1, 2. 3, 4] Tecnológico Nacional de México. Instituto Tecnológico de Apizaco, División de Estudios de Posgrado e Investigación Avenida Instituto Tecnológico s/n C.P. 90300, Apizaco, Tlaxcala, México.

[6, 7] Universidad del Cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Sistemas Calle 5 No. 4 – 70 Popayán, Colombia

[8, 9] Tecnológico Nacional de México. Instituto Tecnológico de Apizaco, Departamento de Centro de Cómputo. Avenida Instituto Tecnológico s/n C.P. 90300, Apizaco, Tlaxcala

Abstract: ***Software architecture is one of the main assets in development companies, for it enables the evolution of systems and the reuse of products and components. Nonetheless, some software-development companies do not have the proper definition, description, handling, or use of architecture, which leads to highly increase the maintenance cost, consequently inflicting the early malfunctioning of the product. Having said that, it is necessary to consider the architectonic evaluation in a way that allows to re-factorize the structure drawn from the risks and the satisfaction of the quality stipulations. On this article, the evaluation of the academic/administrative information system from the "Instituto Tecnológico Nacional de México, campus Apizaco," called integral services system (Sistema de Servicios Integrales) (SSI), is presented, where the Architecture Trade-Off Analysis Method (ATAM) was applied to evaluate the modifiability in three architecture types: web service, customer and programmer, identifying the possible risks in the structure that may interfere with the resulting software product, presenting the evidence and the results gathered by the evaluations.***

Keywords: Quality Attributes, Architecture, ATAM, Modifiability**.**

1. Introduction

The software architecture is a high level design that stresses structural aspects in a system formed by elements with several properties, the IEEE Std 1471-2000: defines software architecture as "the fundamental organization of a system framed in components, the relationships among them, the environment, and the principles that guide the design and evolution" [1]. Due to the complexity of the systems, there is a risk of having a bad architectonic structure, which may cause the quality requirements to be dissatisfied. That's is why it is important to have a well-defined architecture that allows the stakeholders its re-use and evolution.

Another thing to take into consideration regarding the architecture is the projects' life cycle, which are prone to experience changes to handle the costumer or the market´s needs, allowing modifications in its architecture. Since the modifiability is the capacity to enable these changes [2], which include: corrections, improvements, or software adaptations in the environment, in the requirements, and in the functional specifications [3], it makes this quality attribute to be fundamental in an architecture [4], thus, it is necessary to elaborate architectonic evaluations that allow the achievement of a real modifiability grade.

There are several ways and evaluation sceneries to make an architecture evaluation in the modifiability level. One of them is the ATAM method, that consists of nine phases in which are presented different activities to be performed by the interested personnel through the use of a utility trade.

Subsequent to the gathered characteristics, it was determined to use the architecture evaluation method on the "Sistema de Servicios Integrales" (SSI) from the Instituto Tecnológico Nacional de México, campus Apizaco, Tlaxcala, to minimize risks and to be able to take corrective actions on its architecture, which allows to evaluate the architectonic decisions against the quality attributes.

This article is organized in the following way: In Section II, the state of the art is depicted; in Section III, the evaluation method ATAM for the (SSI) architecture is depicted; in Section IV, the implementation of the evaluation is presented; and, in Section V results and conclusions are given.

2. State of the Art

An architecture evaluation is often a process that may help in different actions, such as to generate architectonic changes in the software system to improve or to solve the issues that cause a great impact on the reused projects. It is also a way to assess the quality of the technical design in a product.

The evolution of the software architecture has been widely studied in different software developmental companies, measuring several aspects such as: modularity [5], maintainability [6], reusability [7], sustainability [8], among others. However, this evaluation focuses on the measuring of the software´s architectonic system.

There are several case studies where more than one method has been applied on the architectonic evaluation in which the

structure depends on the characteristics of the system to-be evaluated, thus allowing the achievement of practical results.

A. Meiappane, B. Chithra y Prasanna Venkataesan used the software architecture analysis method (SAAM)[9], for it is an evaluation technique based on sceneries. With this method, it is proven that software systems customizable for the user needs can be acquired, the main goal of SAAM is the architectonic suitability and the risks analysis where direct and indirect sceneries are identified, as well as the amount of elements affected by the scenery [10], which is why there are five phases to be followed in table 1.

Table 1: SAAM. Method's phases

Phase	*Description*
1	Architecture analysis understandable to all those who are interested.
2	Result gathering to elicit sceneries.
3	Assessment of the direct and indirect sceneries where the cost and the exertion associated with it can be identified.
4	The disclosing of high interaction sceneries in order to separate them from the low interaction ones.
5	The sceneries are subjectively classified according to the relative importance to those who are interested.

Subsequent to the method's analysis, a case study was applied on an internet banking system, where the sceneries where gathered. Each one was assigned a load of components affected in the scenery, afterwards it was divided by the total number of components in the architecture showed on table 2 using the next formula:

$$\text{Reusability} = \frac{\text{Components affects}}{\text{Total number of components}}$$

Where the reusability grade is presented in each scenery of the system.

Table 2: Sceneries of the internet banking system

Software´s quality		*Interaction*					
Reusability	scenery	0	1	2	3	4	5
	R1 Allows those who are interest to visit the bank´s web site.	2/4	3/5	4/6	3/9	3/9	4/10
	R2 Allows registration on the banks web site.	2/4	3/5	4/6	3/9	3/9	4/10
	R3 Provides a unique user Id. and password once the user has been registered on the web site.	0/4	0/5	1/6	2/9	2/9	3/10
	R4 The user can enter and edit personal data.	0/4	0/5	1/6	2/9	2/9	3/10
	R5 Allows to check the current balance.	0/4	0/5	1/6	2/9	2/9	3/10

After getting the results of the scenery's evaluation, the reusability is introduced, allowing to measure the degree in which an architecture can adjust to the evolving changes. Nonetheless, the behavior of a quality attribute may have an impact in the performance of other attributes, and this is where the main disadvantage lies within this method, for it does not value the linkage amongst attributes.

Antti-Pekka y Simo define DASE (As the decision and evaluation of the architecture based on sceneries), with a solid approach that combines key aspects of ATAM y DCAR evaluation methods. (Decision-Centric Architecture Review), which led to a fast and precise evaluation of the architecture. [11]. The scenery analysis was taken from ATAM and the decisions of the architecture design from DCAR. The goal is to work faster, getting less people involved, and focusing on the main aspects, where it breaks down to two phases: the first one, prior to work, which collects and processes the list of decisions and sceneries; the second phase focuses on a one-day workshop related to the gathered information. After creating these methods, there were three cases verified in Finnish companies from 2015 to 2017, where the numbers of sceneries and decisions were achieved on each company shown on table 3.

Table 3: Number of decisions and sceneries on each company [11]

Types of decisions and sceneries	*A*	*B*	*C*
Identified decisions.	20	16	9
Voted decisions.	15	11	6
Documented decisions.	4	3	3
Idealized sceneries.	10	15	10
Evaluated sceneries.	10	13	8
Scenery residue grading scale.	0%	13%	20%

Subsequently, the specific findings of each case are described, taking into consideration the fact that it roughly took10 sceneries as sample for the evaluation, table 4.

Table 4: Case results using DASE method

Case	*Characteristics*	*Results*
A	Mid-size- software company. It works in solutions for entrepreneurial software. Accounting product.	The four documented decisions referred to the client data base. The sceneries focused on quality attributes such as the maintenance capability, availability and scalability of the system where impacted by the decision design in the data management.
B	Small company that develops an on demand video broadcast. Apps for different devices. Their architecture was just restructured to become more modular and flexible in order to allow same time functioning.	The three documented decisions defined the key aspects of the new complement. The architecture enabled the separation of concerns and enabled the test. The modifiability, and the tests were the main subjects for the sceneries. The mixture of the analysis from bottom up (decisions) and from top to bottom (sceneries) was appreciated and the order of the sessions was considered good.
C	Mid-sized company tied to the public sector in Finland. The evaluated product was a user authentication recently launched.	The product was developed by a single person. The three documented decisions were in regards to the use of an open source code with a framework as a base for the solution, storing settings from the customer in data bases and a specific dependence of the inherited code that added issues to the general architecture.

		The sceneries took the main role of the company that acts as the main point and center of connection for the authentication and the service. The Interoperability and the maintainability were the most important quality attributes.

By using DASE in the three cases and the results that were gathered it was determined that it's use is modest in the resources (two days per facilitator and one day per participant) with an architecture that can be successfully evaluated in a one-day workshop. The participants also thought the evaluations were useful in general terms and valued the wide perspective of the architecture that was obtained through design decisions and sceneries. To which it is determined that an architectonic evaluation helps both the system and the stakeholders to be able to guarantee a better development.

This is why an architecture evaluation is often a phase based process, where it depends on the method to be used. There are several types of methods that provide help for the evaluation, that are presented on table 5.

Table 5: Characteristics of the types of evaluation methods

Method	*Assessed Quality*	*Metrics and Tools Support*	*Process Description*	*Strengths*	*Weaknesses*	*Systems Type Applicable for*
Method SAAM (Software Architecture Analysis Method) [9].	Modifiability	Scenario classification (direct vs. indirect ones)	Reasonable	Identifying the areas of high potential complexity Open for any architectural description	Not a clear quality metric Not supported by techniques for performing the steps	All
Method ATAM (Architecture Trade-off Analysis Method) [12, 13, 14].	Multiple QAs (Modifiability)	Sensitivity Points, Tradeoff Points Supported by ATA Tool	Good	Scenarios generation based on Requirements Applicable for static and dynamic properties Quality utility tree	Requires detailed technical knowledge	All
Method DCAR (Decision-Centric Architecture Review) [15].	Multiple QAs	Scenario classifications.	Reasonable	Light evaluation method, which may be applicable after the established design of the architecture.	Identification of decisions	All
Method QuaDAI (Quality Driven Architectural Improvement) [16].	Multiple QAs	Multimode	Reasonable	Based on a multimode for the satisfaction of non-functional requirements. Relationship between transformation and quality of architecture.	Based on non-functional requirements	All
Method CBAM (Cost-Benefit Analysis Method) [17].	Costs, Benefits, and Schedule Implications	Time and Costs	Reasonable	Provide business measures for particular system changes Make explicit the uncertainty associated with the estimates.	Identifying and trading costs and benefits can be done by the participants in an open manner.	All
Method ALMA (Architecture Level Modifiability Analysis) [18, 19].	Modifiability	Impact estimation, Modifiability prediction Model.	Reasonable	Scenario generation stopping criterion.	Restricted set of case studies Concentrates on static properties.	Business Information Systems
Method PASA (Performance Assessment of Software Architecture) [20].	Performance time of a software architecture	Impact estimation, Modifiability	Reasonable	Scenario generation stopping criterion.	Restricted set of case studies Concentrates on static properties.	Business Information Systems

These are some evaluation methods that allow the identification of quality attributes in the software architecture sustaining several artifacts and tools as well as the integration of those who are interested. By analyzing the different evaluation methods, it was settled that a system must have a stablished architecture from the beginning, although, not all systems manage to meet this requirement, therefore, it helps identify the importance and the characteristics of each one of them.

The previous methods, such as ALMA and PASA, depend on the expertise of the involved parties to elaborate a feasible evaluation, whereas the CBAM method is only focused on the documentation and tendering of the architecture's requirements and risks.

For this reason, the ATAM method was selected as an evaluation method, since it meets the necessary requirements to the modifiability subject, for said method consist of an architectonic evaluation among the quality attributes, which enables the prediction on decision-making by using defined sceneries analysis by different stakeholders, with the interaction of the nine phases that are part of the method.

3. ATAM evaluation method for the SSI architecture

To fit the ATAM method to the SSI system, some adjustments are made, which are presented in nine stages grouped together in four phases [21], corresponding to the

different tasks, artifacts and roles of the evaluation method the said steps are:

Phase I: Presentation
In this stage, there are three steps to perform a formal evaluation presentation:
1) Presentation of the "ATAM" method: in this step, the leader describes to the participants the method, attempts to satisfy the expectations, and answers the questions.
2) Presentation of the business goals: motivating the effort. These should be granted within the architecture.
3) Architecture´s presentation: the architect describes architecture having in mind how it deals with the business goals that have been set.

Phase II: Investigation and analysis
This phase describes three very important steps to implement the goal of the evaluation and the analysis, which are:
1) Identification of the architectonic approaches (patterns). These elements are found by the architect, however they are not analyzed, since they just face the different system and business goals.
2) Creation of the utility trade: this stage's goal is to gather and polish the system's most important quality attributes, pinpointing these trough sceneries.
3) Analysis of the architectonic approaches: based on the results of the utility trade in the previous stage, both the stakeholders and the architect will analyze the elements in stage 4 to identify the architectonic risks, which aren't as much as architectonic risks as much as sensibility points and concessions related to said evaluation.

Phase III: Testing
It is important to take into consideration the next steps in this phase.
1) Brainstorming and sceneries' priority setting: with the inclusion of everyone involved, a wider set of sceneries is achieved, prioritizing them through voting.
2) Analysis of the architectonic approaches: It is an activity that repeats the tasks in stage 6, but in this case, it makes use of the data from stage 7. The sceneries are considered to be like test cases to confirm the analysis achieved so far.

Phase IV: Outcome report
In this final phase of evaluation, the gathered results will be rendered, all of them found in the last stage.
1) Outcome presentation: this evaluation is based upon the totality of the collected information throughout the evaluation of the ATAM method, where the findings of the stakeholders are presented and the appropriate evaluation is provided.

4. Application of the Evaluation

In the present evaluation the activities, sceneries, and artifacts are described, all conducted with the ATAM method in the SSI, where different outcomes were obtained, as well as the collaborating members, presented on table 6.

Table 6: Members who are part of the evaluation, own creation

Members	*Actions*	*Phases in which participates*
Architects	They are those responsible for generating and documenting the software architecture	All
Evaluator	To lead quality affairs.	All
Stakeholders	Administrators, programmers, users, among others.	Phase 1, 2, 3 and 4

By identifying those who are interested in the system, we can precede with the implementation with the four phases of the ATAM method for the architectonic evaluation, which are presented next:

Phase I: Presentation
The evaluation started off with the presentation of the ATAM method to the members who are part of the process, afterwards the quality attribute focused on the "modifiability" is introduced as a business goal. Subsequent to this, we get:
- Web service architecture: it requires to visualize the different parts that comprises it (figure 1).
- Client architecture: the requests to be performed on the server are displayed (figure 2).
- Architecture of the programmer: The structure of the process is represented (figure 3).

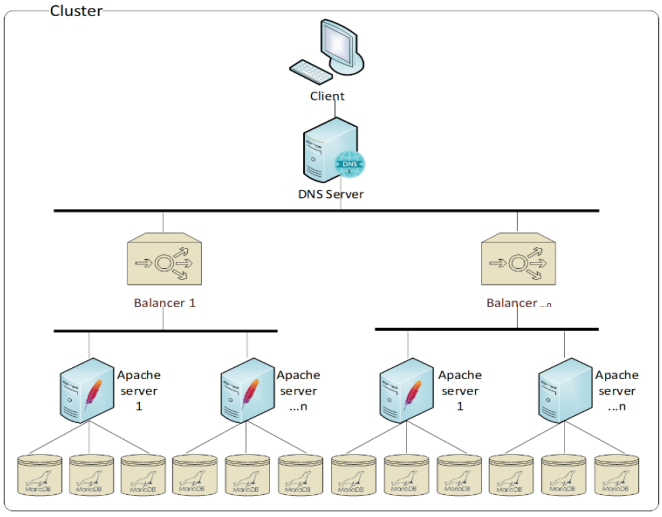

Figure 1: Architecture structure of the web service, own creation.

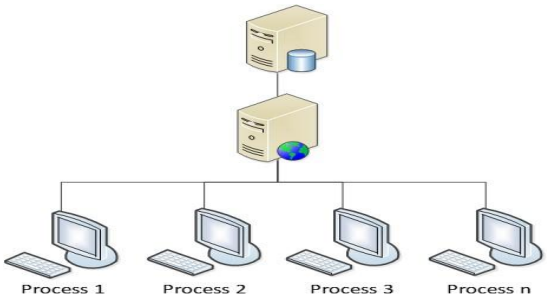

Figure 2: Client structure architecture, own creation.

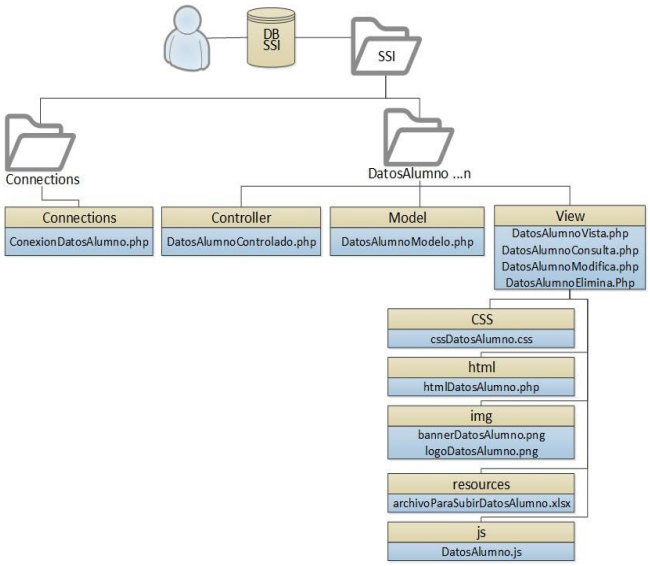

Figure 3: Programmer architecture structure, own creation.

Phase II: Investigation and analysis

Once the presentation phase is concluded, the analysis of the different architectonic patterns is carried out to be able to perform the utility trade, thus giving each scenery a value. The prioritizing level can be found in two different dimensions (the need to reach the requirement and the difficulty to meet said goal), allowing to identify the importance for the system´s success, table 7.

Table 7: Prioritization level in two dimensions, utility trade

Value	*Need to reach the requirement*	*Difficulty to reach the goal*
High	H	H
Medium	M	M
Low	L	L

Therefore, the utility trade offers a mechanism to translate the business goal into quality sceneries, in this evaluation several sceneries are obtained to evaluate focusing on the modifiability of the quality attribute where the interested parties prioritize the degree of need and difficulty, table 8.

Table 8: Utility trade for the quality attribute focused on modifiability

Name of the system	*Quality attribute*	*Scenery*	*Priority*
Sistema de servicios Integrales (SSI)	Modifiability	Scalability of the application server is web	(H, H)
		Process scalability	(H, H)
		Data base scalability	(H, H)
		Reuse of the source-code	(M, M)
		Waiting time configuration	(H, H)
		Users configuration	(H, H)
		Replication servers	(H, H)
		Data base replication	(H, H)

Based on the results from the priority settings from table 7, the architectonic risks are assessed; sensibility and balance areas as opposed to risk, to ascertain whether the architecture meets the modifiability attribute to be evaluated. Subsequent to the architectonic analysis, the next step to be followed is to associate the most important sceneries from the utility trade using the appropriate architectonic styles, for this, stakeholders and evaluators are to perform the next tasks.

- To conduct the most important sceneries using utility trades.
- Identification of each one of the components, connectors, settings and restrictions that are directly linked to the quality attribute in the most important utility trades.
- To develop and to answer a series of specific questions regarding the required attributes and the selected styles.

By the end of this stage, the evaluating team will be able to obtain the most important aspects of the architecture concerning the scenery analysis, shown on table 9.

Table 9: Scenery Analysis, own creation

Scenery analysis			
Scenery description		To improve the modifiability in the SSI in a way that makes it possible to integrate, update and delete new processes for the benefit of those interested.	
Attribute		Modifiability.	
Stimulus		To obtain a modifiable system attached to the different needs of those interested.	
Required attributes		For this, it is required: Programmers, data base administrator, system architect, evaluators, to enable the implementation of the modifiability in the system.	
Architectural decisions	*Sensibility areas*	*Risks*	*Tradeoff*
Balancers distributed system	Connection	Information losses	Availability
CRUD from web servers	Several kinds of servers	Non-important	Portability
Data base replication for other systems	Connection	Information losses	Availability
CRUD from data bases	Several kinds of data bases	Non- important	Portability
CRUD from processes	Different processes	Non-important	Portability

Phase III: testing

Subsequent to the analysis on table 9, it is considered not to carry out new sceneries, for the members of the system have been working on the architecture for a long time, consequently they know it in detail, which awarded the evaluation team the necessary sceneries for the evaluation from the start.

Taking into account the general architecture, the high priority aspects and the analyzed approaches, it was decided that the evaluated architectures have the "modifiability" quality attribute suitable and modifiable for different systems.

Phase IV: outcome report
According to the information generated by the execution of the ATAM method, a summary is carried out for the involved parties, resulting in the utility trade in table 8, the priority order according to table7, the obtained analysis, the risk groups, the sensibility areas, and the commitment areas are shown in the scenery on table 9.

5. Outcomes and Conclusions

The software architecture evaluation is a way to assess the quality attributes of a product. It is also an excellent opportunity to debate the goals. Some architecture evaluation methods are considered to be difficult to perform and expensive to use. Meanwhile, the ATAM method emerges as a solid approach that combines different phase, which makes it a swift evaluation.

ATAM is used to create and evaluate systems with preliminary definitions of the architecture, to analyze architectonic decisions in the absence of a code, proposed architectures or established systems. It is also designed to serve as a guide in the design process for the original architecture. Even though the characteristics of each one of these tasks can be very different, the method does not define alternative routes for each one of them. This is why it has been validated through its performance in different software architectures.

After analyzing and developing the different characteristics for the method, the results based on the performed evaluation are presented in the scenery analysis on table 9, of the SSI from the Instituto Tecnológico Nacional campus Apizaco, within its three architectures (web service, client and programmer), where a general reference of the results on table 10 is inferred.

Table 10: General reference of results from the SSI architectures

Architecture	*Architectural decisions*	*Results*
Web service architecture (cluster)	Balancers distributed system	A modifiability regarding the inclusion of a new balancer can be presented, the elimination of one of them without losing total information, also the updating of it´s properties.
	CRUD of web servers	It presents an adaptable modifiability to store and to transmit data, to authorize or to deny access, allowing the adaptation of additional needs to those presented at the beginning of the implementation.
	Replication of data base for other systems	Data can be modified in any replication, and then with the same synchronization to exchange the updates. The chart replication is handled for "just reading" accesses, in their modification, it will be necessary to access the data from the primary web site.
Client architecture and programmer architecture	CRUD of the data base	The information availability must be optimal, for this reason, the modifiability in which the information is found offers reliability, structure, availability and an optimal performance.
	CRUD of the processes	The cost of modifying the responsibility in a process "A" and at the same time not changing other responsibilities in different processes, reduces the cost of any modification that means changes in "A". There is also the possibility of a modification in a module and this spreading through an interface. The CRUD can be obtained from one module to another.

Due to the different results in the architectural decisions within the three architectures, it was concluded that:

Web service architecture (cluster): It presents an achievable degree of modifiability, since changes can be made. For instance: to delete, to add or to update balancers, servers and data bases without losing information and server replication.

Client and programmer architecture: They present an achievable modifiability degree, such as the structure of the data bases, which can be shared, deleted, updated, or new ones can be created, as well as the model structure, view and controller, where the modifiability emerges in the different processes, offering reliability, structure, availability, and optimal performance.

In conclusion, on the basis of the results, the main goal was achieved on this evaluation by featuring how modifiable the three architectures of the SSI can be through the main scenery and the use of architectural diagrams, which had been proved suitable with the quality attribute.

For this reason, it was established that the ATAM method is a feasible evaluation method of the SSI architectures, for they display a wide approach of the System by the different architectonic proposals, thus allowing a feasible performance for the modifiability.

References

[1] ISO/IEC/IEEE 42010. (2007). Systems and software engineering — Architecture description. Retrieved on August 30, 2018 from: http://www.iso-architecture.org/ieee-1471/defining-architecture.html.
[2] Bass, L., Clements, P. & Kazman. R. (2003). Software Architecture in Practice. (2nd edition) Addison Wesley, Reading.
[3] Bourque, P. & Fairley, R. (2014). SWEBOK - Guide to the Software Engineering Body of Knowledge, Version 3.0.IEEE Computer Society. Retrieved on August 30, 2018 from: SWEBOK: www.swebok.org
[4] ISO/IEC 9126-1 (2001, Jun.). Information technology – Software product quality. (1a Ed.) Part 1: Quality model. International Standard.
[5] Figueiredo, E., Garcia, R. & Lucena, C. J. P. (2007). On the modularity assessment of software architectures: Do my architectural concerns count?. In: First Workshop on Aspects in Architecture Descriptions.
[6] Bosch, J. & Bengtsson, P. (2001). Assessing optimal software architecture maintainability. In European

Conference on Software Maintainability and Reengineering: CSMR 01, 168–175.
[7] Lung, C., Bot, S., Kalaichelvan, K., & Kazman R. (1997). An approach to software architecture analysis for evolution and reusability. In Conference of the Centre for Advanced Studies on Collaborative Research: CASCON 97.
[8] Koziolek, H. (2011, Jun.). Sustainability evaluation of software architectures: A systematic review. In Joint ACM SIGSOFT Conference - QoSA and ACM SIGSOFT Symposium - ISARCS on Quality of Software Architectures - QoSA and Architecting Critical Systems, QoSA-ISARCS 11, 3–12.
[9] Bass, L. Clements & P. Kazman, R. (2003, Abr.). Software Architecture in Practice (2nd Edition) Pearson Education.
[10] Meiappane, A., Chithra, B. & Venkataesan, P. (2013, En.). Evaluation of Software Architecture Quality Attribute for an Internet Banking System. International Journal of Computer Applications (0975 – 8887). Volume 62– No.19.
[11] Tuovinen AP., Mäkinen S., Leppänen M., Sievi-Korte O., Lahtinen S. & Männistö T. (2017). Unwasted DASE: Lean Architecture Evaluation. In: Felderer M., Méndez Fernández D., Turhan B., Kalinowski M., Sarro F., Winkler D. (eds) Product-Focused Software Process Improvement. PROFES 2017. Lecture Notes in Computer Science, vol 10611. Springer, Cham.
[12] Kazman, R., Klein, M., Barbacci, M., Longsta, T., Lipson, H. & Carriere, J. (1998). The architecture tradeo analysis method. In: Proc. of the Fourth IEEE Int. Conf. On Engineering of Complex Computer Systems, ICECCS '98. (1998) 68-78.
[13] Kazman, R., Klein, M. & Clements, P. (2000). ATAM: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004, Carnegie Mellon Sw. Eng. Inst.
[14] Reijonen, V., Koskinen, J. & Haikala, I. (2010). Experiences from scenario-based architec-ture evaluations with ATAM. In Ali Babar, M., Gorton, I., eds. Software Archi- tecture, 4th European Conference on (ECSA 2010). Volume 6285 of Lecture Notes in Computer Science. Springer 214-229.
[15] Van Heesch, U., Eloranta, V. P., Avgeriou, P., Koskimies, K. & Harrison, N. (2014, En.). Decisión- revisiones de arquitectura céntrica. Software, IEEE 31, 69-76.
[16] Hammoudi, S., Ferreira L. P., Filipe J. & Neves, R.C. (2014). Model-Driven Engineering and Software Development: Second International Conference, MODELSWARD. Springer International Publishing. Switzerland 2014. Pag. 21.
[17] In, H., Kazman, R., & Olson, D. (2001). From Requirements Negotiation to Software Architectural Decisions. Software Engineering Institute, Carnegie Mellon University. Retrieved on September 29, 2018 from: http://www.cin.ufpe.br/~straw01/epapers/paper13.pdf
[18] Bengtsson, P., Lassing, N., Bosch, J. & Van Vliet, H. (2004). Architecture-level modifiability analysis (ALMA). Journal of Systems and Software, 69(1-2):129–147.
[19] Lassing, N., Bengtsson, P., Van Vliet, H., & Bosch. J. (2002, Mzo.). Experiences with Alma: architecture-level modifiability analysis. J. Syst. Softw., 61:47–57.
[20] Lloyd, W. G. & Smith, C. U. (2002). "PASASM: A Method for the Performance Assessment of Software Architecture." Paper presented at the Proceedings of the 3rd Workshop on Software Performance, Rome, Italy.
[21] Kazman R., Klein M. & Clements P. (2000). ATAM: Method for Architecture Evaluation. Technical report/book, Carnegie Mellon University, the Software Engineering Institute.

Author Profile

Yesenia Polvo Saldaña Engineer in information and communication technologies. Instituto Tecnológico de Apizaco, 2016. She is currently a Master Degree in computational systems in the Instituto Tecnológico de Apizaco student, and her research is related to the development and implementation of the 'Professional Residence' module in a Higher Educations Institution.

Guadalupe García Toribio Informatic engineer. Instituto Tecnológico Superior de Teziutlán, 2016. She is currently a Master Degree student in computational systems in the Instituto Tecnológico de Apizaco and her research is related to the development and implementation of the "Tutoring" module in a Higher Education Institution.

María Janaí Sánchez Hernández Informatics bachelor degree, Instituto Tecnológico de Apizaco, 2001. Master degree in computation sciences, Instituto Tecnológico de Apizaco, 2006. Professor in the systems and computation department, 2006. Coordinator of the computational systems master of the Instituto Tecnológico de Apizaco, 2015.

José Juan Hernández Mora Computer science engineer, Universidad Autónoma de Tlaxcala, 1994. Master degree in computational sciences by the Centro Nacional de Investigación y Desarrollo Tecnológico del TecNM, 2003. Research professor from the Tecnológico de Apizaco del TecNM. Professor in the master degree of computational systems in the Instituto Tecnológico de Apizaco.

Higinio Nava Bautista Computer science engineer, Universidad Autónoma de Tlaxcala, 2007. Master in software development, Instituto Universitario en Tecnologías y Humanidades, 2012. Professor in the bachelor degree on information and communication technologies engineering. Professor in the computational systems master degree from the Instituto Tecnológico de Apizaco.

Julio Ariel Hurtado Alegría Electronics and telecommunications engineer Universidad del Cauca, 1997. Network and telematic systems expert, Universidad del Cauca. Software development process expert, Universidad San Buenaventura, 2002. Associate professor Universidad del Cauca. IDIS group member since 2005. PhD. In computational sciences Universidad de Chile, 2012.

César Alberto Collazos Ordóñez Systems and computation engineer, Universidad de los Andes, 1993.PhD in Sciences Special mention Computation, Universidad de Chile, 2004. Postdoctoral stay in CARL (Collaborative Applications Research Laboratory) from Universidad de Chile, 2004 and C.H.I.C.O (Computer Human Interaction and Collaboration) from Universidad Castilla-La Mancha, Spain, 2005. Vicepresident of the Sociedad Colombiana de Computación. Coordinator IDIS group (Investigación y Desarrollo en Ingeniería del Software), Universidad del Cauca.

Lucía Muñoz Dávila Bachelor degree Informatics, Instituto Tecnológico de Apizaco, 2004. Master in computer sciences, Instituto Tecnológico de Apizaco, 2006. Professor of the systems and computation department, 2007. Computer center chief, Instituto Tecnológico de Apizaco, 2014.

Nicolás Alonzo Gutiérrez Computation engineer, Universidad Autónoma de Tlaxcala, 1993.Master in computational sciences, Instituto Tecnológico de Apizaco, 2001. Professor in the computation and systems department, 1992. Systems development computer center coordinator, Instituto Tecnológico de Apizaco, 2014.