Modeling Software Architecture Process with a Decision-Making Approach

Gilberto Pedraza-Garcia ∗†
∗Departamento de Ingeniería de Sistemas y Computación
Universidad de los Andes
Bogotá, Colombia
g.pedraza56@uniandes.edu.co
†Programa de Ingeniería de Sistemas
Universidad Piloto de Colombia
Bogotá, Colombia

Hernan Astudillo
Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile
hernan@inf.utfsm.cl

Dario Correal
Departamento de Ingeniería de Sistemas y Computación
Universidad de los Andes
Bogotá, Colombia
dcorreal@uniandes.edu.co

Abstract—The architecture of a software system is the result of balancing a set of concerns expressed by the system stakeholders through a making decision process. There are several industrial approaches to design a software architecture that identify and recognize the importance of architectural decisions and their rationale; however, these approaches are subjective and uncertain because do not provide mechanisms to systematically formalize and manage the architectural decision making process. We propose an strategy based on BPMN to describe in a clear, structured, consistent manner, and with a focus on decisions the current activities of software architecture design. To fulfill this purpose a BPMN approach is applied to specify an As-Is process and analyze, improve and transform it into a to-be process. As a result, a set of best practices applicable into BPMN activities that allow transform current activities into decision-making activities.

Index Terms—software architecture design, decision-making approach; secure software development, BPMN, applying security perspective Decision-making approach, Software architecture design, Software architecture decisions, Business Process Model and Notation (BPMN), Applying security perspective

I. INTRODUCTION

The architecture of a software system is the result of balancing a set of concerns expressed by the system stakeholders through the combination of a set of methodological, creative and collaborative activities. These activities allow the architect to consider a motivation, topic of interest or concern in the problem space, to find alternatives of solution, evaluate and choose the most convenient in the solution space. Thus, the architect develops a decision-making process which facilitates or constrains the achievement of the expected properties for the system software.

Generally, software architects follow an industrial approach for designing the software architecture, such as Attribute-Driven Design, RUP “4+1”, Architecture Centric Design Method (ACDM), Working with Stakeholders Using Viewpoints and Perspectives (V&P). Most of these industrial approaches identify and recognize the importance of architectural decisions and their rationale, however they do not provide mechanisms to systematically formalize and manage them. As a result, the resolution of a quality attribute in a software architecture is a set of subjective and uncertain decisions. The consequence is that decisions maintain intuitive and informal nature, their quality depends on the experience of the architect, the experience gained in previous projects is missed and it is running the risk of not have considered all the aspects that influence a solution.

We propose an strategy based on BPMN to describe the As-Is of the software architecture process. So, current process is described in a clear and consistent manner; besides, it allows introduce a process improvement through unnecessary steps identification, task responsibilities refining and process and technology alignment. The current activities of the process are transformed with a focus on decisions through the introduction of activities for motivating decisions, generating and evaluating solution alternatives and selecting solutions (To-Be).

In order to increase efficiency and quality of the software system, architectural teams need to understand and have a clear specification of the software architecture process. The improvement of software architecture process allows to eliminate unnecessary steps, refine task responsibilities, clarify the alignment of support technology (tools) with software architecture process, and develop a road-map for improvement.

The feasibility of this proposal is illustrated by applying the method Working with Stakeholders Using Viewpoints and Perspectives (V&P) Rozanski and Wood. The proposal is made using the standard specification Business Process Model and Notation (BPMN) that allows the graphical representation of normal architecting activities. BPMN supports process management for technical and business users and provides a technology independent, intuitive and easily notation understandable by the system stakeholders.

Section 2 introduces a background about the software ar-
architecture process, decision approach of software architecture, Bussiness Process Model and Notation (BPMN), section 3 presents the solution strategy, section 4 describes As-Is process of the study case “Applying the Security Perspective”, section 5 shows To-Be of the study case, section 6 presents the ongoing validation and section 7 concludes.

II. BACKGROUND

A. Software architecture process

Over the two last decades both academy and industry dedicated to the software have developed strategies, techniques, best practices and guidelines to the architecture design of a software system. Some approaches have been elevated to the level of methodologies for software architecture design [9]. Among the most recognized are Attribute-Driven Design [3] [4] [5], RUP "4+1" [6], Architecture Centric Design Method (ACDM) [7], Working with Stakeholders Using Viewpoints and Perspectives [8].

Although these approaches were developed in domains with several goals and vocabulary they share some commonality. For example, participation and negotiation by multiple stakeholders, the conformation of development teams for extended periods of time, solving goals and concerns [9]. Furthermore, these approaches have in common a set of activities: Understanding the problem that the architect must solve, finding candidate solutions for the problem, and evaluating and selecting the solution [1].

B. Decision approach of software architecture

Decision-making is a problem-solving process related with the selection of one satisfactory choice between several alternative possibilities applying a set of criteria. This study adopts an approach in which software architecture is designed using a making decision process (See Figure 1). That is, the information and data required to facilitate problem resolution [14].

Architectural decisions resolve particular problems that the architect find in the architecting process and involve structural changes, justifications and consequences for other decisions [15]. All decisions contains several elements of the problem and solution space. An architectural decision is motivated by an specific problem, issue or concern [16]. Solution alternatives describe particular solutions for the problem solution, its advantages and disadvantages and the outcome. The architect assesses the alternatives and selects the one that best meets the problem [15].

Design reasoning is the support for architecture decisions because it includes the reasons justifying each decision made. The focus as proposed is fit into the Bosch proposal [17], who considers architecture as the "composition of an architecture decisions set". In this way, architecture decisions and their corresponding reasoning are important knowledge elements of architecture in a software system [18]. The software architecture design reasoning allows to know and understand the stakeholders interests as considered for each decision, associated restrictions and arguments used to choose a particular one [19]. In the practice, design reasoning is often lost or vaporized resulting in trouble such as maintenance cost increase [17].

C. Business Process Model and Notation BPMN

A process is a set of tasks that interact to achieve a common purpose through the transformation of inputs into outputs and are performed by people or machines [20]. A Business Process (BP) is a set of intertwined activities that are executed in a predefined order with the aim of fulfillment a business objective [21].

Business Process Management (BPM) is a systematic approach to improve and optimize business process performance through the combination of information technologies with process and governance methodologies [20] [21]. BPM proposes a set of methods, tools, and technologies applied in activities related with design, modeling, execution, monitoring and optimization.

Business Process Modeling is "the activity of representing business processes, so that the current (As-Is) process may be analyzed and improved in future (to-Be)" [22]. BPMN (Business Process Model and Notation) defines a standardized, expressive, and understandable graphical notation for modeling of business process [23]; currently, BPMN Version 2.0 is an OMG standard.

III. SOLUTION STRATEGY

The solution proposal starts with the document review on the process, the process core elements identification, the specification of BPMN processes (As-Is), the search for opportunities to improve the process, and the improved process specification BPMN (To-Be). See Figure 2.
A. Document Review on the Process

First, the document review consists on analyzing, identifying and extracting keywords, terms, and relationships between them from the textual description of the process. We propose the application of content analysis to extract the process structure through the examination of textual documents [24].

B. Identify the Process Core Elements

Second, it is necessary to identify the core elements related with the process. To recognize this elements it is necessary to resolve the following questions:

- Why the process must be performed?. The objective of the process should be clearly stated.
- What do activities support the achievement of the objectives?. The activities fulfill a specific function in achieving the objectives.
- Who is the people that participate in the activities?. Each activity has an unique responsible actor which must be identified.
- Which activities is each actor responsible for?. Each actor has the responsibility of a set of activities to perform.
- When are these activities required?. The chronological order of the activities indicates when the activities will be carried out.
- What concerns have the stakeholders about the process?. It is necessary to state the requirements, constraints, intentions, or aspirations that each stakeholder has on the process.

C. Produce the As-Is Process in BPMN

Once we have identified the process core elements then the BPMN diagrams are elaborated. In drawing diagrams, it should consider: Ascertaining the precise boundaries, inputs and outputs of each activity, the activities described maintain a chronological order and intermediate products maintain uniqueness, all process must be embedded in the process structure.

D. Identify the Improvement Opportunities in the As-Is Process

In the opportunities identification for improvement, it is necessary to characterize the process in areas such as: the process orientation (control, inputs transformation), the way the activities are developed, the relationships and dependencies between activities among others.

Some improvements have to do with a structural formalization of activities, roles, inputs and outputs of the process. In general, The enhancements should seek to improve properties such as: definability, order, value-adding, embeddedness, and cross-functionality among others.

We propose identify the main candidate activities to transform into decisions. Generally, these activities are responsible to create or identify a new intermediate outcome of the process; besides, each intermediate outcome requires to be reviewed and approved by reviewers and stakeholders.

E. Improved Process Specification BPMN (To-Be)

The improvements identified in the previous step must be performed on the initial model (As-Is). For this the BPMN diagrams are redefined: introducing more detailed activities, extending activities, eliminating redundancies and adding new elements.

IV. STUDY CASE: ROZANSKI AND WOOD PROCESS OF "APPLYING THE SECURITY PERSPECTIVE"

A. Software architecture process of Rozanski and Wood’s method V&P

Rozanski and Wood’s method defines the software architecture as "the set of fundamental concepts or properties of the system in its environment, embodied in its elements, relationships, and the principles of its design and evolution" [8]. This method is supported on three fundamental concepts:

1) Stakeholders or people for whom the system is built with some kind of interest, need or concern in it,
2) views and viewpoints as a structuring and description mechanism based on the concerns separation principle that contain a guide of description, and
3) perspectives to capture the quality attributes that are cross-cutting to the system functional structure [8].

B. Document Review on the Process

The document review was doing on the chapter "The Perspective Catalog" and specifically on the section "The Security Perspective" [8]. The chapter document has 28 pages consisting of the following sections:
• Introduction to security.
• Applicability of security in the architectural views.
• The main concerns about security.
• Activities for application security perspective described according to Figure 3. For each activity, a notation to specify it and a set of defined tasks to perform.
• A set of architectural tactics.
• A pitfall and problem list.
• A set of checklists for requirements capture and architecture definition.

Fig. 3. Currency Process of Security Perspective Application

C. Identification of the Process Core Elements

In the process "Applying the Security Perspective", the identified core elements are summarized in Table I. Below these elements are described:

1) Process Objective: The objective is to achieve a software system ability to control, monitor and audit reliably who is authorized to perform actions on particular resources and the possibility to detect and recover from security failures [8].

2) Concerns: The concerns expressed in the documentation are described as follows:

- **Resources.** Each software system stores information and performs certain functions that are considered valuable assets [8].
- **Principal.** Software systems are used by people or entities for the purpose of performing any task. Principals represent these entities that may be people, roles, pieces of physical equipment or other system software [8].
- **Policies.** A security policy defines the controls, access, constraints and rules that describes security features of a software system [8].
- **Threats.** The threats describe the ways in which an attacker can avoid fulfilling or violating security policies [8].

- **Confidentiality.** Confidentiality is the limitation to the disclosure of information to people who have not been granted the right to know it [8].
- **Integrity.** Integrity ensures that information can not be changed without a trace and the only ones who have been changed information are the authorized principals [8].
- **Availability.** Availability is concerned with ensuring that potential attackers cannot suspend the operation of the software system by denial-of-services attacks [8].
- **Accountability.** Accountability ensures that all director actions can be traced back [8].
- **Detection and Recovery.** It has to do with the soft-
ware system ability to detect and recover from security breaches [8].

- **Security Mechanisms.** Security mechanisms are technologies, procedures and configuration settings for the implementing the selected strategy to carry out the security policies [8].

3) **Activities:** The process documentation shows Figure 4 where it is proposed the following activities:

- **Identify Sensitive Resources.** This activity allows to define the elements that must be secured [8].

- **Define Security Policy.** The security policies are the basis for designing a security system because they identify who is entrusted access to the system resources and the restrictions to the system resources, the rules of system integrity, and the accountability of the resources when they are accessed [8]. A security policy assigns a responsibility to a role on a set of resources, defines types of authorized access, sensitive operations and the associated integrity rules [8].

- **Identify Threats to the System.** One of the goals of security is to gain the trust of the stakeholders with the system software. One way to achieve this is to demonstrate the ability of the system to control the possible threats. To identify system threats is important to known potential attackers the system, its motivation, how can circumvent the security policy, its characteristics in terms of resources, commitment, level of sophistication and the consequences of that policy has been violated. Attackers can be inside and outside the organization [8].

- **Design the Security Implementation.** The purpose is to design and specify the technical security infrastructure into the architectural structures [8].

- **Assess Security Risks.** The implementation of security architecture mitigates risk but do not guarantee the elimination of risk. The end result is a balance between the mitigation costs and the possible risks of the software system. For this reason it is necessary to evaluate the risks and assess whether the security architecture design in the cost/risks balance [8].

4) **Actors:** The actors that participate in design process are:

- **Solution architect.** The solution architect outlines the architecture description of a software system and have a high-level view of the entire solution; also, he has skills to share, communicate ideas clearly with executive staff, business sponsors, and technical specialists.

- **Business architect.** The business architect knows, analyzes and advises about the structure of the business models that can be enabled by business technology.

- **Product architect.** Product architect guides the delivery of the software system to external customers. He is responsible among other aspects to: design the models of user interaction, supervise the technical integrity of the product, identify stakeholders before the first release, produce technical specifications for product development and design, lead external collaboration meetings.

- **Infrastructure architect.** Infrastructure architect is responsible of specify the hardware and software infrastructure. He describes the structure and behavior of the technology infrastructure of the system, such as: database management systems, enterprise security, enterprise messaging, and desktop tools among others. Besides, he evaluates, selects and integrates technical components to create solution designs that solve functional and non functional requirements.

- **Data architect.** Data architect is responsible to outline the information view and identify and ensure that data assets are supported by the data architecture.

- **Security Technology Expert.** This specialist has deep knowledge about software security. He is responsible for providing information about how to implement security tools and technologies and assess advantages and disadvantages of its application.

- **Reviewer.** The reviewer is responsible to evaluate the completeness, consistency and coherence of the intermediate outcomes in the process.

- **Stakeholders** Software architecture stakeholders are people, teams, or organizations with some interest in the software system development [8].


The current process is described in Figure 4.

![Fig. 4. Process of Security Perspective Application](image)

1) **Identify Sensitive Resources:** This subprocess identifies resources that can be sensitive data or system functions. To identify sensitive resources should take into account the diverse situations where it may affect the security principles (See Figure 5). The solution architect initiates "Request for sensitive resources” to the product and data architect. The
product architect "identifies sensitive functional operations" taking as input the previously security requirements identified and the functional view. Second activity is "identify sensitive data items" is performed by the data architect which he take as input previously security requirements identified and information view to identify the sensitive data items in the architecture. Later, the solution architect "consolidates sensitive data items and functional operations" resources and records all the sensitive resources.

2) Define Security policies: In this process the architectural team performs several sub-processes:

1) Identify the principal classes. In this activity, it is defined the groups for applying security policy objectives (See Figure 6). The criteria used for the principals distribution are based on the role and types of access [8]. The business architect performs the activity of identify the principal classes which takes as input the context view, the list of system users, and the previously sensitive resources identified and records the principal classes.

2) Identify the resource classes. The data and product architects classify sensitive resources using as criteria the access control type over the resource. The input of this activity is the previously sensitive resources identified and the output is the resources classes.

3) Identify the Access Control Sets. The Security expert defines the types of operations on each resource class and assigns the principal classes to access each operation. The inputs are the resources classes and the principal classes and the output is the access control set.

4) Identify the Sensitive System Operations. The solution architect performs this activity to describe other operations of the level system such as administrative functions that are not defined in sensitive resources and assign the principal classes for they. The inputs are operational and functional view, and the output is the specification of sensitive system operations.

5) Identify the Integrity requirements. In this activity the solution architect proposes integrity rules to recognize resource inconsistencies due to unauthorized changes to sensitive resources. The inputs are previously sensitive resources and sensitive system operations identified. The output is the integrity rules.

3) Identify Threats to the System: This subprocess contains several activities: The solution architect analyzes the security policies, the product and data architects analyzes sensitive resources, the solution architect identifies attackers outside the organization, the product and data architects identify attackers inside the organization. Besides, the security expert defines the types of attacks. Finally, the infrastructure architect describes the attacks results (See Figure 7).

V. PROCESS IMPROVEMENT "APPLYING THE SECURITY PERSPECTIVE"

A. Identify Improvement Opportunities about the As-Is Process

The currency process "Applying the Security Perspective" has several characteristics:

- The process shows a design approach from the problem to solution space. In other words, the process begins by identifying a security needs that are solved by a set of strategies that produce a specification in the solution space.
- An architectural team works collaboratively in the building of the several design outcomes.
A lot of activities take the results of previous activities as inputs, transform and produce a new result.

The process documentation is textual and the activities are not showed in an structured format.

The process is defined in high abstraction level that not allow to identify and assign responsibilities to the participating actors and understand the relationships between activities, actors, inputs and outputs.

The collaborative work in design teams require achieve consensus of all participants and stakeholders in the system software.

Taking clear the problems of the current process, the following process improvements are proposed:

1) **Transform a single activity into decision activity:** The candidate activities to become decision activities are those which aim describe an intermediate result in the process. This activities describes some concern as principal, sensitive resource, security policy, and threat among others. An scheme of this type of activity is described in Figure 8. This practice consists in extending a single activity with the purpose to introduce several activities that represent a decision-making process. We propose the following activities (See Figure 9):

   1) **Establish the topic motivator and objective.** In this activity, the inputs are analyzed to specify the matter of the decision. The decision starts with a motivation in the problem space that can be a concern, issue, interest or problem to solve.

   2) **Generate a set of solution alternatives.** It corresponds to candidate solutions that meet the motivators. Alternative candidates are generated from the identification of a set of possible action alternatives and the results of each alternative [25].

   3) **Evaluate the solutions alternatives.** It contains information about its advantages and disadvantages and optionally a decision structure. The advantages and disadvantages are determined by a utility function where a set of criteria assigns a value to each possible alternative [25].

   4) **Select the solution.** The alternative that best meets the problem according the utility function is chosen [15] or the reasons that led the designer to choose this alternative as a solution.

2) **Collaborative activities to Review and Approval:** In order to model collaborative activities related to the review and approval process we propose a set of activities described in Figure 10.

To ensure the process quality is necessary to incorporate activities that support it. Two types of activities are proposed: reviews on the process partial results between the architectural team members and the search for consensus and approval by the project stakeholders. The first activity has been incorporated is the "Request the architectural element review" by the member who developed the outcome. This request is sent to the designated reviewer architect who makes the corresponding review "Review the architectural element". If he find outcome remarks then suspend the activity to describe and document the remarks, the outcome responsible "Evaluate the remarks" and the process is reworked. If there are no remarks then a message is sent to the outcome stakeholders to approve. Stakeholders evaluate whether or not their concerns are met in the architecture outcome. If stakeholders find remarks then suspend the activity to record the remarks, the outcome responsible "Evaluate the remarks" and the process is reworked.
B. Improved Process Specification BPMN (To-Be)

1) Identify Sensitive Resources: This process is extended on two levels: first, to "Identify sensitive functional operations" activity have been added the following tasks (See Figure 11): The "Analyze security Requirements and concerns" activity where it is set the decision issue to should be resolved taking as input the security Requirements and Concerns. The "Identify sensitive functional operations" activity allows to describe the candidate functional operations to become sensitive. This task take as input the architectural functional view. The "Evaluate sensitive functional operations" activity compares the candidate functional operations applying one or more criteria. In the "Select sensitive functional operations" activity is prioritized and choose the final sensitive functional operations.

Second, the "identify sensitive data items" activity is extended adding the tasks: Analyze security requirements and concerns, identify sensitive data items, evaluate sensitive data items and select sensitive data items (See Figure 11).

2) Define Security policies:

1) Identify the principal classes. This activity is extended with the following activities related with decision-making approach (See Figure 12): The solution architect analyzes the current list of users, identifies the principal classes taking as input the sensitive resources and the context view. In the next activity he evaluates the principal classes identified, later he selects the principal classes. Thus the principal classes passes through the states identified, evaluated and selected forming the justification for the decision. The second extension is to add collaborative activities for review and approval. The solution architect requests a review of the principal classes selected. The review architect makes the review; if he found remarks then record them and the solution architect analyzes this remarks and restarts the process. If there are not remarks then a message is sent to the stakeholders for approving the principal classes. If stakeholders found remarks then these are recorded and the activity is suspended until they are resolved.
VI. ONGOING VALIDATION

The specification using BPMN has allowed a better understanding of the design process “Applying security perspective”. The BPMN models are validated with the support of a wiki to develop a real project related with a Tsunami Early Warning System.

VII. CONCLUSIONS AND FUTURE WORK

In this study, we presented an approach to describe an approach based on decision-making to represent a software architecture process. This approach is based on applying BPMN and a set of stages such as document review about the process, identification of process core-elements, specification of BPMN processes (As-Is), search for opportunities to improve the process, and improved process specification in BPMN (To-Be). Among the proposed improvement practices are transforming simple activities in decision-making activities where the motivating decision, the candidate alternative, and selecting of the best alternative are included. Also, the management of collaborative activities to review and approve of the resulting outcomes in the process are shown.

As future work, we want to extend the process described to support the management of the reasoning behind the design decisions by introducing knowledge management activities. Besides, it is also expected to validate the proposed solution experimentally.

ACKNOWLEDGMENTS

This work was partially supported by CCTVal (CONICYT Basal FB0821).

REFERENCES


