Journal of Universal Computer Science, vol. 29, no. 12 (2023), 1510-1534
submitted: 13/4/2023, accepted: 9/8/2023, appeared: 28/12/2023

Design and Evaluation using Technology Acceptance Model of an Architecture Conceptualization Framework System based on the ISO/IEC/IEEE 42020

Valdicélio Santos
(Federal University of Sergipe, São Cristóvão, Sergipe
https://orcid.org/0000-0003-1244-4939, valdiceliomendes@gmail.com)

Michel S. Soares
(Federal University of Sergipe, São Cristóvão, Sergipe
https://orcid.org/0000-0002-7193-5087, michel@dcomp.ufs.br)

Abstract: Among the difficulties in developing software-intensive systems are the necessity of managing and controlling data that must be held for decades, as well as describing the needs and concerns of a variety of stakeholders. Therefore, one cannot neglect a good Software Engineering practice which is to develop software-intensive systems based on solid software architecture. However, the processes related to the software architecture of software-intensive systems are often considered only from a low level of abstraction. A recent architectural Standard, the ISO/IEC/IEEE 42020, defines 6 clauses for the architecture process, among them the Architecture Conceptualization process is the subject of this study. Considering that the ISO/IEC/IEEE 42020 has only recently been published, given the importance of establishing a well-defined software architecture, and considering the difficulties of understanding an architectural Standard, this work proposes a framework, and then the design and further evaluation of a web-based application to support software architects in using the activities and tasks of the Architecture Conceptualization clause based on the framework described. The ArchConcept was designed to address the high-level abstraction of the Standard ISO/IEC/IEEE 42020 and can be useful for software architects who want to follow ISO/IEC/IEEE 42020's recommendation and achieve high-quality results in their work of software architecture conceptualization. A qualitative evaluation employing a questionnaire was carried out to obtain information about the perceptions of professionals regarding the ArchConcept, according to the Technology Acceptance Model (TAM). As ArchConcept is focused on activities of Architecture Conceptualization, which is one of the early stages of a software project, the results found could be evidence of the short time dedicated to the initial phases of projects and their consequences.

Keywords: Software Architecture, ISO/IEC/IEEE 42020, Architecture Conceptualization, Technology Acceptance Model
Categories: D.2.11, H.4
DOI: 10.3897/jucs.104938

1 Introduction

A variety of processes, methods, languages, frameworks and architectures have been proposed to deal with the challenge of developing software-intensive systems. Activities related to each one of these software engineering elements have been proposed in the past decades. One cannot disregard a good Software Engineering practice which is to develop software-intensive systems based on solid software architecture. However, processes related to software architecture are still often neglected [Booch 2007, Knodel and Naab 2014, Woods 2016, Matthias et al. 2017, Dasanayake et al. 2019], even though it is well-known in industry and academia that software architecture processes are considered a critical success factor [Brown and McDermid 2007, Garlan 2014, Junior et al. 2019].

Although there are debates on the role of describing software architectures in agile software development processes [Pace et al. 2016, Hoda et al. 2018, Florian and Uludağ 2020, Rocha et. al 2023], there are important results that consider software architecture as a crucial technical component for agile development [Abrahamsson et al. 2010, Yang et al. 2016, Rocha et. al 2023], even though there is still a lack of description and analysis regarding the costs and failure stories of the combination of agile and architecture-based software development processes.

Processes related to the software architecture of software-intensive systems are often considered only from a low level of abstraction, even for a description of the software architecture. Given the importance of Software Architecture processes, a family set of architecture Standards was proposed under reference number 420X0. The ISO/IEC/IEEE 42010 Standard [ISO42010] was proposed in 2011 with the main objective of being a recommended practice for creating a description of software architecture. Recently, a new Standard, the ISO/IEC/IEEE 42020 [ISO42020], was set in a document to address a Standard for governance management, conceptualization, evaluation, and elaboration of architectures, and activities that enable these processes. Another Standard in the family, the ISO/IEC/IEEE 42030, regards means to organize and record architecture evaluations for enterprise, systems and software fields of application. ISO/IEC/IEEE 42020 is the main subject of the study presented in this article.

The importance of using International Standards such as ISO/IEC/IEEE 42020, given that it directly affects the quality and reliability of software, has been recently recognized by many authors [Santos et al. 2020, Ascher and Ehlers 2021, Quezada-Gaibor et al. 2022]. However, ISO/IEC/IEEE 42020 has only recently been considered in practice, and there is still a shortage of its actual use. Considering that ISO/IEC/IEEE 42020 has only recently been published, given the importance of establishing a well-defined software architecture, and considering the difficulties of understanding an architectural Standard, this work proposes a framework, and then the design and further evaluation of a web-based application to support software architects in using the activities and tasks of the Architecture Conceptualization clause based on the framework described.

The proposal in this article is to design and implement a web-based application to describe software architectures following the ISO/IEC/IEEE 42020 regarding the Architecture Conceptualization process, which presents high-level concepts and elements of software architecture. Given this proposal, first, we identified the main aspects for the clause Architecture Conceptualization of ISO/IEC/IEEE 42020 and then proposed a framework that provides requirements to design and develop the web-based application.

Each architectural process described in ISO/IEC/IEEE 42020 is organized in terms of purpose, desired outcomes, and a list of activities for achieving those outcomes. There are 6 architect processes in ISO/IEC/IEEE 42020: Architecture Governance, Architecture Management, Architecture Conceptualization, Architecture Evaluation, Architecture Elaboration and Architecture Enablement. This article addresses the process “Architecture Conceptualization”, given that our purpose here is to understand stakeholders’ needs, including their main concerns, requirements, purposes, processes, and objectives, before dealing with the architecture elaboration. The proposed web-based application addresses these purposes, by assisting software architects to perform their daily activities.

2 Background

Some of the basic concepts that are necessary to read this article are briefly introduced in this section.

2.1 Standard ISO/IEC/IEEE 42020

The Standard ISO/IEC/IEEE 42020 establishes a set of process descriptions for governance and management of a collection of architectures, as well as describes processes to architect entities. These processes are applicable both for a single project as well as for multiple projects and product lines. In addition, the Standard is useful throughout the lifespan of the architecture of software systems [ISO42020].

Six architecture processes are proposed in ISO/IEC/IEEE 42020, named Architecture Governance, Architecture Management, Architecture Conceptualization, Architecture Evaluation, Architecture Elaboration, and Architecture Enablement. Each one of these processes is described in terms of purpose, desired outcomes, and a list of activities and tasks for achieving those outcomes. Tasks are recommended for implementing those activities. As the focus of this article is on the Architecture Conceptualization Process, it is briefly introduced in the following subsection.

2.2 Architecture Conceptualization Process

The purpose of the Architecture Conceptualization process is to characterize the problem space and determine suitable solutions that address stakeholder concerns, achieve architecture objectives and meet relevant requirements.

Conceptualization is the process responsible for establishing and maintaining the alignment of architectures in the architecture collection with enterprise goals, policies, strategies, and related architectures. The focus is on identifying solutions, considering an emphasis on fully understanding the complete problem space. This also entails the definition and establishment of architecture objectives, as well as negotiation with key stakeholders on prioritization of their concerns [ISO42020].

Outcomes of the Architecture Conceptualization are the definition of the problem being addressed, the establishment of architectural objectives that address the key stakeholder concerns, the definition of key architecture concepts and properties, and the principles guiding its application and evolution. Besides these outcomes, the Architecture Conceptualization process addresses key tradeoffs, as well as identifies possible limitations. Finally, within this process, the candidate solutions are clearly defined and understood.

There are 10 activities related to the Architecture Conceptualization Process. Each activity is composed of several tasks that implement the activity, in a total of 100 tasks. For instance, the activity "Relate the architecture to other architectures and to relevant affected entities" is composed of 6 tasks, one of which is to "Formulate principles and precepts expected to be used during execution of the life cycle processes".

The ten activities in the Architecture Conceptualization Process are:

1. Prepare for and plan the architecture conceptualization effort.
2. Monitor, assess and control the architecture conceptualization activities.
3. Characterize problem space.

4. Establish architecture objectives and critical success criteria.
5. Synthesize potential solution(s) in the solution space.
6. Characterize solutions and the tradespace.
7. Formulate candidate architecture(s).
8. Capture architecture concepts and properties.
9. Relate the architecture to other architectures and to relevant affected entities.
10. Coordinate use of conceptualized architecture by intended users.

The idea here is to deal with high-abstraction level elements related to the first activities to establish a software architecture. In this article, all 10 activities of the Architecture Conceptualization Processes are considered.

Work products of the Architecture Conceptualization Process are an architecture conceptualization plan, an architecture conceptualization status report, a problem space definition report, architecture objectives, a quality model, and architecture views and models.

The implementation of the Conceptualization Process produces the following outcomes.

- The problem being addressed.

  The number and availability of stakeholders vary across software projects. Considering the context of the first activities of software development, for instance, those related to Requirements Engineering - which includes requirements elicitation and prioritization -, a minimum set of stakeholders consists of users and acquirers. Complex projects can impact many users and many acquirers, each one presenting different concerns. Project requirements may necessitate including two other groups as part of the minimum set of stakeholders. First, the organization, when developing, maintaining, or operating the system or software, has a legitimate interest in benefiting from the system. Second, regulatory authorities can have statutory, industry or other external requirements demanding careful analysis.

  User requirements are then transformed into system requirements for the system of interest. The consistent practice has shown that this process requires iterative and recursive steps in parallel with other life cycle processes through the system design hierarchy. The recursive application of these processes will generate lower-level system element requirements [ISO24765].

- Architecture objectives that address the key stakeholder concerns.

  First, activities are established to clarify stakeholders' needs and concerns. After that, it should be established the architecture objectives to meet requirements. Architecture entities are used to compose an architecture objective that deals with issues of the problem space, which can be used in other aspects of the solution. An example of an objective of the architecture of interest is to support the reuse of an entity across various technologies, protocols, platforms, operational venues, and market segments [ISO42020].

– The architecture's key concepts and properties.

  These concepts and properties can be expressed in the form of information and communication technology constructions and models such as information flows, control flows, data structures, operational rules, event/trace diagrams, state transition diagrams, timelines, and roadmaps. Also, they can be expressed in other forms such as risk models, financial models, economic models, simulation models, sensitivity models, queuing models (as well as other kinds of continuous and discrete event simulation models), geospatial models, management models, business models, social- and environmental-impact models, value stream models, among others.

– Key tradeoffs are understood concerning the problem being addressed and the relevant stakeholder concerns. Quality attributes consist of various system responses such as performance, security, availability, flexibility, and so forth. Almost every decision will affect more than one quality attribute and thus involves a tradeoff between quality attributes, as well as the design decisions made by stakeholders in the architecture development phase, which have far-reaching consequences [Olumofin and Misic 2007].

  Architectural design decisions that impact quality attribute interactions are classified into sensitivity points and tradeoff points. A sensitivity point applies to a decision about specific aspects of the architecture that may affect, either benefiting or impairing at least one quality attribute; a tradeoff point is a sensitivity point between two or more quality attributes that interact in opposing ways [Olumofin and Misic 2005].

  Stakeholders should be aware of the key tradeoffs to decide which of them will meet their concerns.

– Tradeoffs among architecture objectives and feasibility limitations are identified, and the architecture objectives targeted to be addressed by the architecture are clearly specified.

  The solution might not be addressing the entire problem or all aspects of the problem. Therefore, it is important to understand where the solutions fall short, and the tradeoffs that are considered when choosing among alternative solutions.

  If needs, wants or expectations drive the solutions, then negotiate with those stakeholders to determine which of the needs, wants, or expectations are to be translated into requirements on the solution and the relative priority of each.

– Candidate solutions for the problem are clearly defined and understood.

  In all steps of development, there will be a diversity of options for solutions that address the problems. These solutions must be clear to the stakeholders so that they can decide which suits their needs.

2.3 Technology Acceptance Model

Many models have been proposed to explain and predict the use of technologies such as software [Venkatesh and Bala 2008]. One of the most widely employed models for adopting and evaluating the actual use of Information Technology is the Technology Acceptance Model (TAM) [Davis 1989, Adams et al. 1992]. TAM is a technology acceptance model widely used in Information Systems, which focuses on how the perception of ease of use, perception of usefulness, and social influence affect consumer attitudes.

The TAM model suggests that, when users are presented with new technology, several factors influence their decision about how and when they will use it. Two variables were first investigated in this context: perceived usefulness, defined as the degree to which a person believes that using an IT will enhance his or her job performance, and perceived ease of use, defined as the degree to which a person believes that using a particular system will be free of effort. A third variable added later, the perceived usage [Adams et al. 1992], is defined as the degree to which the user will use the technology.

In this research, we are interested in the individual's acceptance and further use of a new web-based application for architecture conceptualization, which is a good starting point to view how this new technology might be accepted in a community. The evaluation was performed using two techniques: surveys, which were based on a questionnaire using the TAM theory, and interviews with practitioners and researchers.

3 Related works

Even before the introduction of the standard ISO/IEC/IEEE 42020, the problem of creating one first concept of software architecture was discussed in the literature, but only with a focus on parts of what is described in the ISO/IEC/IEEE 42020.

For instance, in [Pareto et al. 2012] the authors proposed a prioritization method, which combines collaborative and analytical techniques and involves several stakeholder groups to produce informed recommendations, on which areas of the architecture documentation need to be improved, and what to improve in those areas. This work addresses only one of the tasks of Activity 6, "Synthesize potential solutions in the solution space", of Clause 8 (Software Architecture Conceptualization), present in ISO/IEC/IEEE 42020 Standard.

In another work [Heesch et al. 2012], the authors propose a framework for the documentation of architectural decisions. The framework is composed of four viewpoints, related to concerns: decision details viewpoint, which mainly addresses the concerns related to the rationale that motivates decisions; viewpoint of decision's relationships, which focuses on the concerns of the relationships between decisions; viewpoint of the involvement of stakeholders in the decision, which allows explaining the relationships between stakeholders and decisions; viewpoint chronology of decisions, designed to satisfy remaining temporal concerns in the decisions. The proposed framework deals only with architectural decisions, which is one of the tasks of activity 8, "Capturing concepts and properties of architecture", of Clause 8 (Software Architecture Conceptualization).

As it has been recently released, there is a shortage of papers referring to ISO/IEC/IEEE 42020.

An architecture is presented in [Kumar et al. 2018] for a solution that allows assistance to elderly people. The work refers to the Architecture Conceptualization of the ISO/IEC/IEEE 42020 Standard, presenting the identification activities of the stakeholders, their concerns, requirements, systems' context, value proposition, systems' value proposition, systems' proposal, quality characteristics, concept maps, architectural principles, functionalities, models and visions. It is also important to note that the concept mapping of the system is not a simple task, but it can be a useful tool for the development of the system.

In [Martin 2018], the authors present an overview of the key concepts of the six processes of the ISO/IEC/IEEE 42020 Standard. The authors point out that the article describes the effort to incorporate software architecture practices within an international standard, serving as a support for the ISO/IEC/IEEE 15288 Standard [ISO15288]. The

ISO/IEC/IEEE 15288 Standard presents the design definition process for each system of interest to be developed, while the ISO/IEC/IEEE 42020 Standard shows the process of defining the architecture for a complete solution given the concerns of all stakeholders. This allows the architecture to be used for solutions of various types, such as systems of systems, collection systems and product lines. It is further stated that the architecture conceptualization process focuses on establishing the objectives of the architecture and determining the fundamental concepts and properties that best meet those objectives, given the constraints, conditions and challenges.

In their turn, [Phillips et al. 2018] wrote that ISO/IEC/IEEE 42020 Standard (under development at that time) is among the documents to manage software resiliency by improving space system resiliency because it provides software architecture design, and increases software security, thereby reducing the risk of latent software defects and vulnerabilities. A software-resilient architecture also contributes to higher assurance functionality.

More specifically, in [Diampovesa et al. 2021], the ISO/IEC/IEEE 42020 Standard is used in a case study of a Li-on battery for electrical vehicles, where the authors propose a design approach for architecture conceptualization. The concepts of problem space, knowledge space, and solutions space are meaningful in explaining the differences between modelling a system and modelling its design problem.

In another work [Blobel et, al, 2022], the ISO/IEC/IEEE 42020 was cited in a guide to finding and deploying corresponding International standards for Medicine Ecosystems.

In another case, standards ISO/IEC 18305:2016 and ISO/IEC/IEEE 42020 were used to help enhance the quality of an indoor positioning platform, as well as its integration with other systems [Gaibor 2023].

The definitions proposed in the ISO/IEC/IEEE 42020 were considered suitable for the purpose of the study [Ivanov 2022], as they take an inclusive stance on both the term enterprise, covering any relevant organisational configurations, and the term architecture, comprising both the static and the dynamic view of Enterprise Architecture in a holistic manner.

Lastly, Creff and others [Creff et al. 2020] pointed out that Model-Based Systems Engineering (MBSE) must include assistance to the system designers in identifying candidate architectures to subsequently analyze tradeoffs, and it is expensive to build analysis models to explore the space of possible solutions. Unfortunately, efforts in the Systems Engineering community generally favour solution analysis, e.g. in providing system modelling languages and tools (e.g. SysML, Capella), over exploring a set of candidate architectures (architectures synthesis), a feature needed in the early stages. They then use the ISO/IEC/IEEE 42020 Standard to explore the advantages of designing and configuring the variability problem to solve one of the problems of exploring (synthesizing) candidate architectures in Systems Engineering.

As described before, [Pareto et al. 2012] and [Heesch et al. 2012] are a sample of how the activities of conceptualization of software architectures have been performed in an isolated way if compared to the Conceptualization Process of the ISO/IEC/IEEE 42020 Standard. This article proposes to approach the 10 activities of the Conceptualization Process present in the ISO/IEC/IEEE 42020 in an unedited way, to evaluate its use and to highlight if there are advantages, according to the objectives established.

Solutions present in [Kumar et al. 2018, Martin 2018, Phillips et al. 2018] were published considering the draft version of ISO/IEC/IEEE 42020. On the other hand, other works [Creff et al. 2020, Diampovesa et al. 2021, Quezada-Gaibor et al. 2022, Blobel et, al, 2022] presented the current version of ISO/IEC/IEEE 42020, specifically addressing the architecture conceptualization process.

4 A Proposed Framework for Architecture Conceptualization

In order to create an Architecture Conceptualization, the organization should implement the relevant tasks (identified as list items under each 8.4.N activity) as appropriate to the situation, according to Clause 8 [ISO42020].

The activities adopted by the Architecture Conceptualization framework are briefly described as follows.

1. Preparing for and planning the architecture conceptualization effort

   These are the first steps towards conceptualizing the architecture, which occurs through the accomplishment of some tasks. The tasks are concerned with identifying a general area of the problem, defining purpose, objectives, and level of detail, deciding which architecture description framework will be used, the architecture strategies and approaches, developing architecture conceptualization techniques, methods, and tools, and planning the architecture conceptualization effort. All these tasks need to be approved by related stakeholders. In addition, the team responsible for dealing with each one of these tasks needs to be assigned.

2. Monitoring, assessing and controlling the architecture conceptualization activities

   The tasks of this activity consist of checking if the conceptualization effort is being accomplished according to what is expected, and likewise checking if other processes are properly using architecture conceptualization products. This is possible through metrics, which can be checklists, key performance, risk, and opportunity indicators.

   The Standard recommends assessing and controlling the architecture conceptualization effort according to the Project Assessment and Control process present in [ISO15288], whose purpose is to assess if the plans are aligned and feasible; determine the status of the project, technical and process performance; and direct execution to help ensure that the performance is according to plans and schedules, within projected budgets, to satisfy technical objectives.

3. Characterizing the problem space

   Architects need to understand the basic situation that they will have to deal with for developing and managing all the software systems for which they are responsible. Therefore, some important tasks are to identify the problem space and determine the most important requirements of the interested parties. After that, the architects need to identify the solution space, which means they have to describe all the products, frameworks, patterns, services, and technologies that will address these stakeholders' problems and needs.

   It is important to adopt a systematic approach to identify the most effective solution given several possibilities. An evaluation of alternative solutions should be provided by the architecture team, and each possible solution should be documented in such a way that important architectural decisions can be retrieved whenever it is necessary. Thus, once a decision is documented, even if in the future the decision is classified as a bad one, at least it creates knowledge about the whole situation, providing an additional experience for architects, developers and other stakeholders.

4. Establishing architecture objectives and critical success criteria

In this activity, architectural objectives are identified and defined to address stakeholders' concerns, requirements or quality attributes, which were previously identified. Critical success criteria are defined to assess whether the problem has been solved, checking whether the final objectives have been achieved. Even if the problem is not completely solved, a degree of success can be identified. It is also relevant to establish a quality model that considers the adopted quality measures as well as the relationships between them.

5. Synthesizing potential solution(s) in the solution space

   This activity proposes defining specific objectives, where necessary, to be achieved when addressing the problems and opportunities. In addition, relate these to the established architecture objectives and success criteria. Once those objectives have been part of the problem space, and are related to the architecture objectives and success criteria, it is required to assess them. On the other hand, it recommends designing a new architecture from scratch or using one previously established solution.

   It is important to store an architecture repository because the solution can be found in current or past architectures. Thus, an architecture repository also reinforces the importance of the organization of keeping an Architecture Knowledge Management. Each solution implicates characterizing strengths, weaknesses, tradeoffs, and also identifying needs, wants, and expectations for each potential solution. After that, it has to be negotiated with the stakeholders to determine which of the needs, wants or expectations are to be translated into requirements on the solution and the priority of each requirement [Capilla et al. 2016].

6. Characterize Solutions and the Tradespace

   When determining key criteria for system quality, it is commonly known that stakeholders will identify many quality characteristics expected for the final solution. Stakeholders will try to provide their feeling of what they need. For instance, an application needs to provide means to be secure, and also provide a minimum performance. Then, developers understand these restrictions, needs, and constraints, and establish non-functional requirements in structured documents. According to ISO/IEC/IEEE 29128 [ISO29128], non-functional requirements have to be written in such a way that they can be verified, i.e., there are means to identify if the expected quality characteristics of a system are fulfilled.

   Given the possible solutions, developers need to analyze tradeoffs between all possibilities. For each architectural relevant solution, an analysis should provide a full evaluation of the pros and cons of each solution. For instance, as soon as developers understand the necessity of using a framework, criteria of evaluation are established and the candidates are evaluated. Typically, quality characteristics need to be prioritized. For instance, for most software systems, as long as they do not need to provide strict timing constraints, performance is not as important as maintainability, or even it is better to improve the easiness of use instead of performance.

7. Formulating candidate architecture(s)

   After a solution is identified, and the functional and non-functional characteristics are described, it is time to formulate candidate architectures that address stakeholders' concerns according to their prioritization.

It is important to create traces between key characteristics (based on identified stakeholder concerns, relevant requirements, quality attributes, architecture objectives, and other relevant factors) and the formulated candidate architectures to show how each one addresses those key characteristics. In doing that, stakeholders will have a clear understanding of each proposed architecture and be able to make better decisions.

8. Capturing architecture concepts and properties

   Architecture Conceptualization only needs to describe the architecture to the level of specificity and granularity that is suitable for its intended users, which in many cases does not require significant elaboration [ISO42020]. The Elaboration Process deals with descriptions, views, and models that capture architectural concepts and properties.

   ISO/IEC/IEEE 42020 should be used to provide more information about these architecture description concepts. These views and models are explored further in clause 10, Architecture Elaboration process, of ISO/IEC/IEEE 42020.

9. Relating the architecture to other architectures and relevant affected entities

   This item is important to map the proposed architecture to other relevant elements such as policies, processes, logistics, and personnel. When there are relationships between architectural elements and other entities, these should be explicit. Also, these relationships must have available principles and precepts for those who will use the designed architecture. During this step, additional factors will sometimes be found that drive the architecture, which may result in changes to the architecture. In addition, this provides an opportunity to help clarify the meaning and intent of requirements and to ensure that all relevant requirements can be met by the architecture.

10. Coordinating use of conceptualized architecture by intended users

   Key stakeholders should validate the conceptualized architecture through feedback on the description, views, and models. Besides, there are other users of architecture conceptualization information who are responsible for the evaluation or elaboration of the architecture, review, management, design engineering, and so on.

   Many ways can be considered for validation. For instance, workshops in which each stakeholder can give their opinions about each relevant aspect of the architecture described so far. Each item that can be improved can be identified and documented for future reference.

5 ArchConcept: A Web-System for Architecture Conceptualization

A Software Architecture is typically developed because key people have concerns that need to be addressed by the business and Information and Technologies departments within the organization. Such people are commonly referred to as the “stakeholders” of the system. One important role of the architect is to address these concerns, by identifying and refining the requirements that the stakeholders have, developing views of the architecture that show how the concerns and the requirements are going to be addressed, and showing the tradeoffs that are going to be made in reconciling the potentially conflicting concerns of different stakeholders. Without the architecture, it is unlikely that all the concerns and requirements will be considered and met.

As described in the previous Section, the Conceptualization Architecture Process of the ISO/IEC/IEEE 42020 Standard presents many high-level abstract definitions. Due to the high amount of tasks to be performed about Architecture Conceptualization based on this Standard, the software architect has to spend time to understand what is proposed in the Standard, get stakeholders' information, and organize and gather all knowledge obtained concerning Clause 8.

Considering these needs, a web system is proposed to aid software architects in improving productivity and accomplishing a high-quality result by following the Standard's recommendation through the use of this tool. This Section describes the Architecture Description according to the ISO/IEC/IEEE 42010 Standard, the Architecture Conceptualization according to ISO/IEC/IEEE 42020 Standard and the design of the ArchConcept.

5.1 ArchConcept Requirements

The functional requirements were obtained from the main activities of the framework, which leads to capturing stakeholders' (in this study, the architect's) needs, including their main concerns and related viewpoints, problem space, architecture objectives, architecture decisions, and tradeoffs.

- RF01 - The system shall allow the software architect to register the problem.
- RF02 - The system shall allow the software architect to change the problem.
- RF03 - The system shall allow the software architect to delete the problem.
- RF04 - The system shall allow the software architect to register the stakeholders.
- RF05 - The system shall allow the software architect to change stakeholders.
- RF06 - The system shall allow the software architect to exclude stakeholders.
- RF07 - The system shall allow the software architect to register the concerns.
- RF08 - The system shall allow the software architect to change the concerns.
- RF09 - The system shall allow the software architect to delete the concerns.
- RF10 - The system shall allow the software architect to register the objectives of the architecture.
- RF11 - The system shall allow the software architect to change the objectives of the architecture.
- RF12 - The system shall allow the software architect to delete the objectives of the architecture.
- RF13 - The system shall allow the software architect to register viewpoints.
- RF14 - The system shall allow the software architect to change viewpoints.
- RF15 - The system shall allow the software architect to delete viewpoints.

- RF16 - The system shall allow the software architect to register the tradeoffs.
- RF17 - The system shall allow the software architect to change tradeoffs.
- RF18 - The system shall allow the software architect to delete tradeoffs.
- RF19 - The system shall allow the software architect to register architectural decisions.
- RF20 - The system shall allow the software architect to change architectural decisions.
- RF21 - The system shall allow the software architect to delete architectural decisions.

5.2 ArchConcept Architecture Description according to the ISO/IEC/IEEE 42010 Standard

The ISO/IEC/IEEE 42010 Standard [ISO42010] does not specify any format for recording architecture descriptions, but it describes the basic context of an architecture description. Besides, the Standard specifies the best practices for documenting enterprise, system and software architectures.

Considering the ArchConcept architecture description according to the ISO/IEC/IEEE 42010 Standard [ISO42010] presented in this work, the software architects are the stakeholders and have interests in a software tool (one or more software systems). These interests are the concerns, such as Functionality and Usage. A concern pertains to any influence on a system in its environment. Each system is situated in an environment, which is a context determining the setting and circumstances of all influences upon a software system. The ArchConcept is in the operational environment.

Software architecture is documented in multiple views to help stakeholders to better understand the software system. In addition, architecture description using multiple views allows for managing complexity and risks during software development. Four architectural views, Scenario, Business Process, Logical and Development, are presented to provide an overview of the development of the application. Each view presents different elements, improving the separation of concerns.

5.3 ArchConcept Architectural Decisions

Architectural Description is also a set of Architectural Decisions, that guide high-level project development. Among them, Tables 1, 2 and 3 present examples of Architectural Decisions for the ArchConcept application.

Description	Accessibility decision.
Constraint	Minimum requirements .
Solution	The development view of architecture should be in layers, provided by the MVC pattern
Rationale	Using the system from different devices will provide more flexibility and increase the possibility of access.

Table 1: Architectural Decision for ArchConcept application D01.

Description	Security decision.
Constraint	Do not overload the system.
Solution	Security must be addressed to protect information from unauthorized people and avoid losing information.
Rationale	Defining strategies for protecting the access of unauthorized persons and for avoiding loss of information.

Table 2: Architectural Decision for ArchConcept application D02.

Description	Platform, language, and tool preferences.
Constraint	Only open source tools should be used for development.
Solution	Define open source tools to development.
Rationale	The use of open-source development tools due to high quality and mature use and due to an experimental web application.

Table 3: Architectural Decision for ArchConcept application D03.

5.4 ArchConcept Architectural Views

The architectural views Scenario, Business Process, Logical and Development were chosen because they provide the essential views so that the application can be comprehended at a high level and communicated for design and development activities.

5.4.1 Scenarios View

The Use Case diagram of the proposed system provides a high-level view, gathering the scenarios of execution, as depicted in Figure 1.

The software architect can manage the functionalities of the system, which involves managing: Viewpoints, Concerns, Stakeholders, Problems, Objectives, Decisions and Tradeoffs. These functionalities are related to the purpose of the Architecture Conceptualization process, which is to characterize the problem space and determine suitable solutions that address stakeholder concerns and are related to the main outcomes of the Architecture Conceptualization: the definition of the problem being addressed, the establishment of architectural objectives that address the key stakeholder concerns and the definition of key architecture concepts and properties, which guide its application and evolution.

5.4.2 Business Process View

The UML Activity diagram of Figure 2 presents part of the Architecture Conceptualization process developed in the ArchConcept. As an experimental system, the only process presented is the proceedings of the architect to fill in some information about the Architecture Conceptualization, partially modelled and developed in the ArchConcept.

In Figure 2, first, the architect registers the concern, and then the stakeholders related to that concern. Following, he/she registers the viewpoint that frames the concern. The architect registers the problems, which are expressed in architectural objectives. The tradeoffs registered to characterize the architectural decisions guide the architectural objectives.

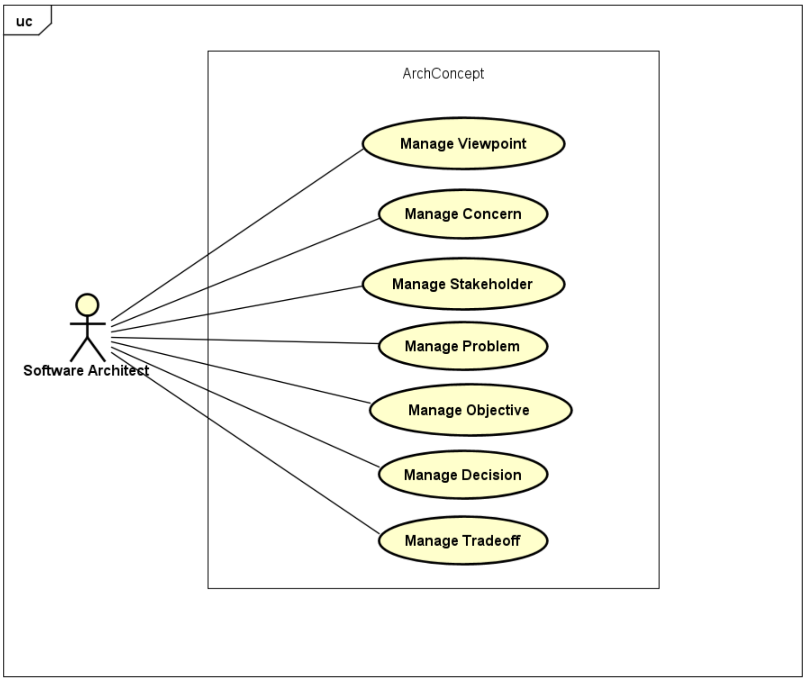

Figure 1: UML Use Case diagram of the system for architecture conceptualization.

5.4.3 Development View

Development of the ArchConcept application is based on the Model View Controller (MVC) pattern, as depicted in Fig 3. ArchConcept source code is organized into three modules that are model, view and controller. The model considers components that directly manage data, logic and rules of the application using Spring Framework and Java Persistence API (JPA). View generates output representation of the ArchConcept system using two frameworks, Thymeleaf and Materialize. Control addresses input manipulation and converts it to commands for the model or view. Control represents components in ArchConcept responsible for flow control and business rules.

The Java Spring Boot MVC, an MVC-Based Framework, is used to develop the system, along with Open Source technologies, such as Thymeleaf and JPA, for the persistence of data in the PostgreSQL database.

ArchConcept includes the use of free tools and technology as presented in Table 4.

The application was deployed to Heroku, a container-based cloud Platform as a Service (PaaS).

5.4.4 Logical View

The UML Class diagram is designed as a basis for developing the ArchConcept system. This class diagram was used to develop Java classes of the ArchConcept application. A stakeholder has an interest in concerns, which are framed by viewpoints. Also, a

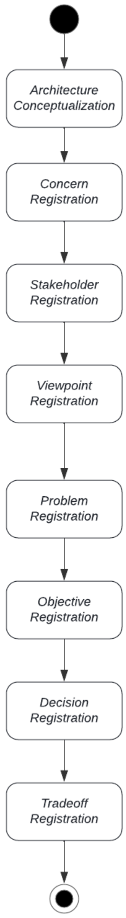

Figure 2: Architecture Conceptualization process.

Technology	**Layer**	**Role**
PostgreSQL	Data	Relational Database
Hibernate	Distribution	Object-Relational Mapping
Thymeleaf, materialize	Front End	User interface communication
Apache Maven	Deploy	IDE configuration for Deploy
Spring Tools 4 for Eclipse	Coding	IDE

Table 4: ArchConcept tools and technology.

stakeholder is related to the identification of problems, which is expressed in architectural objectives. Tradeoffs characterize the architectural decisions guided through the architectural objectives.

The Class diagram in Figure 4 presents the classes Viewpoint, Concern, Stakeholder,

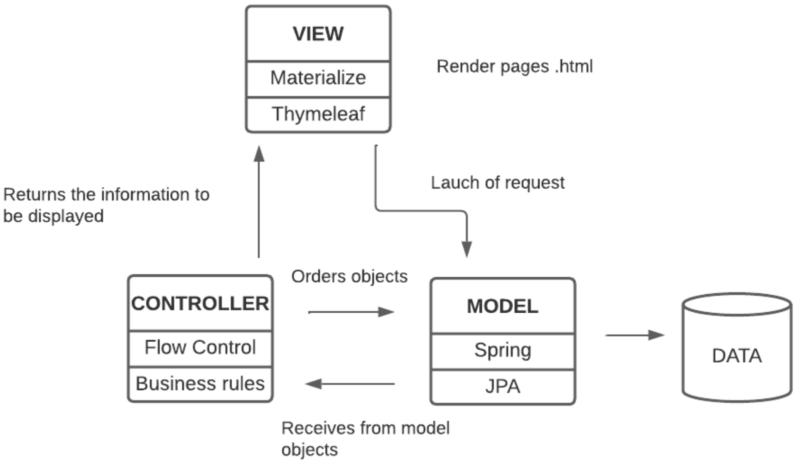

Figure 3: MVC model for application implementation.

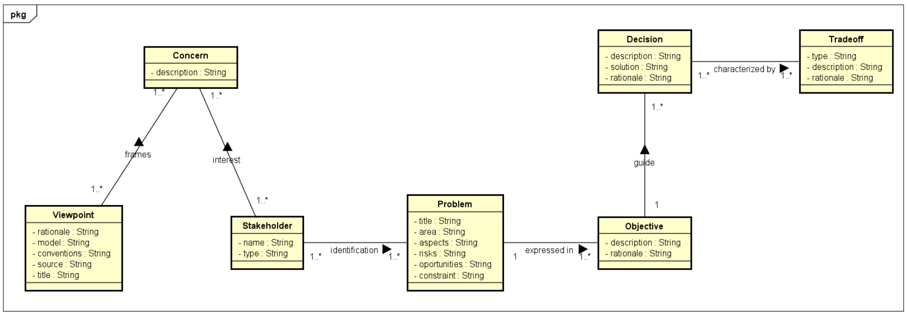

Figure 4: Class Diagram of the system for architecture conceptualization.

Name	ArchConcept Operation Viewpoint
Overview	The architecture viewpoint deals with the main stakeholder concerns related to the ArchConcept operation.
Stakeholders	Architect(s) who design and describe the architecture
Concern	Functionality Usage
Environment	Operational

Table 5: ArchConcept Operation Viewpoint.

Problem, Objective, Decision and Tradeoff, their attributes and the relationships among them.

5.5 ArchConcept Viewpoint

Architectural Description is also a set of Viewpoints that describe the system from different perspectives. Just to exemplify a Viewpoint, Table 5 presents the operation

Statement	**1**	**2**	**3**	**4**	**5**	**p**
1. Using ArchConcept allows me to better manage my work activities.		4	1	8	4	12
2. Using ArchConcept increases my performance at work.		4	5	5	3	8
3. In ArchConcept, the functionalities for registering, editing and deleting stakeholders are useful for my work.		3	4	4	6	10
4. In ArchConcept, the functionality of adding and deleting concerns is useful for my work.		3	2	7	5	12
5. In ArchConcept, the functionalities for registering, editing and deleting viewpoints are useful for my work.		3	2	7	5	12
6. In ArchConcept, the problem registration, editing and deletion functionalities are useful for my work.		2	4	6	5	11
7. In ArchConcept, the functionalities for registering, editing and deleting objectives are useful for my work.	1	2	3	6	5	11
8. In ArchConcept, the functionalities for registering, editing and deleting decisions are useful for my work.	1	2	2	6	6	12
9. In ArchConcept, the functionalities for registering, editing and deleting tradeoffs are useful for my work.	1	3	1	6	6	12
10. ArchConcept enables me to perform my tasks in less time.		5	4	7	1	8
11. Using ArchConcept increases my productivity.		5	4	6	2	8

Table 6: Perceived usefulness of ArchConcept - statements 1 to 11.

viewpoint for the ArchConcept application.

6 Evaluation of the ArchConcept using TAM

In this research, we are interested in the individual's acceptance and further use of a new web-based application for architecture conceptualization, which is a good starting point to view how this new technology might be accepted in a community. The evaluation was performed using a technique survey, which was based on a questionnaire using the TAM theory with practitioners and researchers.

As explained previously, the Technology Acceptance Model (TAM) is a model of IT adoption and use. It is adopted in this work for part of the evaluation.

6.1 The Questionnaire

The participants, researchers or practitioners, were invited according to their professional area related to Software Engineering and Software Architecture. They work as Researchers, System Analysts, University Professors, and Project Managers, on average have 9 years of experience in software development and more than 4 years of experience in Software Architecture.

After reading the user guide that is present on the initial page of the ArchConcept system, the participants used Google Forms to answer a questionnaire composed of 33 statements (11 of Perceived usefulness, 17 of Perceived ease of use, and 5 Perceived of usage of the ArchConcept system) in which they made their opinions explicit.

Statement	1	2	3	4	5	p
1. It is easy for me to register, edit or delete stakeholders.			1	5	11	16
2. It is easy for me to correlate stakeholders and concerns.		1		8	8	16
3. For me, it is easy to register, edit or delete viewpoints.			1	8	8	16
4. For me, it is easy to register a viewpoint from the concern already registered.			1	9	7	16
5. For me, it is easy to add, edit or delete problems.		1		7	9	16
6. It is easy for me to set up a problem, associating it with one or more stakeholders.			1	9	7	16
7. It is easy for me to add, edit or delete objectives.	1		2	9	5	14
8. For me, it is easy to register one or more objectives from an already registered problem.	1		4	7	5	12
9. For me, it is easy to add, edit, or delete decisions.	1	1	2	7	6	13
10. For me, it is easy to add one or more decisions from a previously registered objective.	1	2	1	7	6	13
11. It is easy for me to add, edit, or delete tradeoffs.	1	3	1	8	4	12
12. For me, it is easy to register one or more tradeoffs from a previously registered decision.	1	2	2	8	4	12
13. It is easy for me to generate architecture conceptualization reports.		1		6	10	16
14. It is easy for me to remember how to execute tasks in ArchConcept.		1	1	6	9	15
15. I can use ArchConcept without consulting the user manual.	1	3	3	7	3	10
16. Using ArchConcept makes my work easier to do.		3	3	9	2	11
17. It was easy for me to use ArchConcept.		1	1	11	4	15

Table 7: Perceived ease of use of ArchConcept - statements 1 to 17.

A 5-point Likert scale [Likert 1932] was proposed to measure the perceived attitudes of the employees by providing a range of responses to each statement. The scale ranged from (1) strongly disagree, (2) disagree, (3) neutral, (4) agree, and (5) strongly agree. For the negative statements, it was inferred from low scores that the participants are positive about a statement in the questionnaire, while for the positive statements, a high score was inferred that the participants were positive about the statement.

6.2 Questionnaire results

The questionnaire was responded to by seventeen personnel, working in the industry and as researchers/university professors. The answer regarding each question is shown in Tables 6, 7, and 8, in which "p" indicates the number of positive answers. We arbitrarily considered as positive the answers "Agree" or "Strongly Agree" (values 4 or 5). For negative sentences, we considered 1 and 2 as positive responses.

Table 6 presents answers related to statements 1 to 11 and indicates that ArchConcept presents useful functionalities, in which 72,7 % of the statements have positive answers. Statements 1, 4, 5, 8, and 9 are top-rated with 12 positive answers. 12 respondents answered that using ArchConcept allows them to better manage their work activities, and the same number of respondents would use ArchConcept functionalities like managing concerns, viewpoints, and decisions. 11 subjects answered ArchConcept is useful to

Statement	1	2	3	4	5	p
1. I am interested in knowing more about the Architecture Conceptualization process, present in the ISO/IEC/IEEE 42020 Standard.		2	2	8	5	13
2. The reports generated by ArchConcept are useful for me.	1	3	3	6	4	10
3. ArchConcept is useful for my work.		4	5	5	3	8
4. I would use at least one of ArchConcept's functionalities in my work.	1	3	1	7	5	12
5. I would use most or all of the functionality of ArchConcept in my work.	3	4		7	3	10

Table 8: Perceived usage of ArchConcept - statements 1 to 5.

manage architecture problems and objectives. 10 subjects marked that stakeholders' functionalities in the ArchConcept are useful for their work. On the other hand, only 8 answered that ArchConcept would increase their performance and productivity, and that would enable them to perform their tasks in less time.

Table 7, concerning answers related to statements 1 to 17, indicates that ArchConcept is easy to use. 100 % of the statements were positive answers. Statements 1, 2, 3, 4, 5, 6, and 13 are top-rated with 16 positive answers. For 16 respondents, it is easy to manage stakeholders, viewpoints, and problems, correlate stakeholders and concerns, register a viewpoint from the concern already registered, set up a problem, and associate it with one or more stakeholders. For 15 participants, it is easy for them to remember how to execute tasks in ArchConcept and use it. For 14 participants, it is easy to manage objectives. For 13 participants, it is easy to manage decisions and to add one or more decisions from a previously registered objective. For 12 participants, it is easy to manage tradeoffs, register one or more objectives from an already registered problem, and register one or more tradeoffs from a previously registered decision. And for 11 participants, using ArchConcept made their work easier.

Table 8, concerning the answers related to statements 1 to 5, assesses the perceived usage of the ArchConcept. 80% of the statements have positive answers. 13 respondents were interested in knowing more about the Architecture Conceptualization process, present in ISO/IEC/IEEE 42020. 12 would use at least one of ArchConcept's functionalities in their work. For 10 subjects, the reports generated by ArchConcept are useful, and they would use most or all of the functionalities of ArchConcept in their work. Only 8 answered that ArchConcept would be useful in their work.

6.3 Threats to validity

This research presents some threats to validity that range from internal, construct, statistical, and external validity. This section discusses the strategies used to manage these threats.

Internal validity.

Construct. Due to the Covid-19 pandemic, it was not possible to have face-to-face training before the use of the ArchConcept. The participants read the user guide, used the application, and answered the questionnaire. It might be some misunderstandings in this process. The number of respondents may not be enough to generalize the findings, although they are experienced practitioners and researchers.

Statistical. The objective was to find the representative power of the results, and not necessarily the statistical significance.

It should be emphasized that perceived usefulness and ease of use are people's subjective appraisals of performance and effort, respectively, and do not necessarily reflect objective reality. Though different individuals may attribute a slightly different meaning to particular statements, the goal of the multi-item approach is to reduce any extraneous effects of individual items.

In this research, we are interested in the individuals' acceptance process, which is a good starting point to view how new technology might be accepted in a community. Therefore, extensions to TAM in which the social context is taken into account, although useful, are not investigated here. In addition, the purpose is to evaluate the initial acceptance of technology, which makes TAM a suitable model.

External validity. It refers to the validity of the obtained results in other wider contexts. Since this study has not been replicated yet, we describe some criteria that can be used to identify similar contexts where our findings can be closely applied: the participants should be from areas related to Software Engineering and Software Architecture, practising work as Researchers, System Analysts, University Professors, and Project Managers, with some experience in Software Developing and Software Architecture.

Thus, we can infer that several aspects of the research can be found in other contexts, although the results cannot be generalized, which means that new experimental studies can be carried out.

7 Discussion

The Standard ISO/IEC/IEEE 42020 defines 6 clauses for the architecture process, among them the Architecture Conceptualization process which is the subject of this study. This decision was taken because our purpose here is to understand stakeholders' needs, including their main concerns, requirements, purposes, processes, and objectives, before dealing with the architecture elaboration. The proposed web-based application addresses these purposes by assisting software architects in their daily activities. Given the relevance of ISO/IEC/IEEE 42020 to providing useful and mature guidelines in terms of the Software Architecture Process, it is important that organizations consider adopting it in the near future.

Most commonly, the works presented before addressed in a partial or isolated way some of the activities and tasks of the architecture conceptualization process, such as stakeholder concerns, architectural decisions, problem space, solution space, and tradeoffs.

The lack of a system to gather and summarize all the activities of the architecture conceptualization is evident, as presented in the section Related Research. Our purpose in this work is to propose a framework and a web-based system to meet the needs of the architectural process present in the ISO/IEC/IEEE 42020 Standard for Architecture Conceptualization.

Although the architecture processes can be executed simultaneously with the interactions between them and the iteration over time, as with the Core Processes of ISO/IEC/IEEE 42020 (Architecture Conceptualization, Architecture Evaluation, and Architecture Elaboration), this article deals with Conceptualization Activities of software architecture. Following the activities and tasks in Clause 8 helps to start the architectural effort to clarify stakeholders' needs. On the other hand, some of the tasks seem to be repeated. For example, task "a) Identify the general nature of the problem area(s) that needs to be addressed" of activity "8.4.1 Prepare for and plan the architecture conceptualization effort", and task "a) Identify the potential problem area(s) that needs to be addressed" of activity "8.4.3 Characterize problem space". This is likely because the processes are interrelated, but the architecture team needs to be aware of these situations.

In addition, considering the high number of tasks (100), and their high level of abstraction, understanding and using the Architecture Conceptualization clause in practice, in industry, seems to be a challenge. Considering the novelty of the ISO/IEC/IEEE 42020 Standard, tools such as ArchConcept are the first step towards implementation of the guidelines of that Standard so that the industry can evaluate the benefits.

The Standard ISO/IEC/IEEE 42020 presents 6 Architecture processes with a high level of abstraction in many of their Activities. It offers possible integration with related Standards such as ISO/IEC/IEEE 42010, which presents a way to create an Architecture Description. ISO/IEC/IEEE 42010 proposes the system software architecture as a product composed of models, views, viewpoints, decisions, and other architectural elements, which represent important entities of software architecture, but are at a lower level of abstraction when compared to the framework proposed in this article.

For instance, task "l) Develop an architecture description consisting of relevant viewpoints, views, models, and model correspondences and express them in the specified form with a level of detail, correctness, and completeness suitable for their intended use" of activity "8.4.8 Capture architecture concepts and properties". In other words, ISO/IEC/IEEE 42020 provides processes for the application of Architecture Description defined by ISO/IEC/IEEE 42010 [ISO42020].

The conceptualization of the architecture aims to characterize the problem space and determine solutions that meet the concerns of the stakeholders, define the objectives of the architecture, and meet the relevant requirements. Taking into account these three main objectives together, it was not possible to identify studies that apply or evaluate fully the architecture conceptualization process since the release of the ISO/IEC/IEEE 42020 Standard in 2019.

8 Conclusion

A recent architectural Standard, the ISO/IEC/IEEE 42020, defines 6 clauses for the architecture process, among them the Architecture Conceptualization process is the subject of this article.

The work carried out proposed a framework and related tool to help in organizing the activities and tasks proposed in Clause 8 of ISO/IEC/IEEE 42020, leading to a mature characterization of the solution proposed by the architecture. The ArchConcept was designed to address the high-level abstraction of the Standard ISO/IEC/IEEE 42020 and can be useful for software architects who want to follow ISO/IEC/IEEE 42020's recommendation and achieve high-quality results in their work of software architecture conceptualization. In the software industry, in complex projects, usually, there would be many solutions, and the interested parties can decide on the best one according to their needs. Therefore, a software tool for architecture conceptualization is useful.

A web-based application was designed, developed, and evaluated to support software architects in using the activities and tasks of the Architecture Conceptualization clause based on the framework proposed. The Archconcept implemented activities 1, 3, 5, 6, 7 and 8 of the framework. Activities 2, 4, 9 and 10 were not contemplated as the application is experimental, covering main aspects regarding the Architecture Conceptualization and would require higher complexity.

As ArchConcept is focused on the early stages of the project (Architecture Conceptualization), the results found in this work could be evidence of the short time dedicated to the initial phase of projects and their consequences, like misunderstandings. Some respondents answered they do not perceive the usefulness of the ArchConcept system because they did not realize the importance of the Architecture Conceptualization to the system, or because there is a lack of knowledge, as 13 respondents were interested in knowing more about the Architecture Conceptualization process.

As the ISO/IEC/IEEE 42020 Standard is new and still not well known in industry and academia, the results indicated that the web-based application was not properly perceived in terms of usefulness and usage, although many respondents realized it just by reading the user guide. On the other hand, the application was considered easy for daily use.

For future works, the framework could be refined to explore more examples of the activities and tasks of the Architecture Conceptualization Process present in the ISO/IEC/IEEE 42020 standard. As the Architecture Conceptualization Process can be applied to many kinds of projects, demonstrating it through the framework could show to related personnel the applicability of adopting the ISO/IEC/IEEE 42020 standard in their teams.

References

[Abrahamsson et al. 2010] Abrahamsson, P., Babar, M. A., Kruchten, P.: "Agility and Architecture: Can They Coexist?"; IEEE Software. 27,2 (2010) 16-22

[Adams et al. 1992] Adams, D. A., Nelson, R. R., Todd, P. A..: "Perceived Usefulness, Ease of Use, and Usage of Information Technology: a Replication"; MIS Quarterly. 16,2 (1992) 227-247.

[Ascher and Ehlers 2021] Ascher, D., Ehlers, F.: "Seamless Integration of ATR Services into Situation-Adaptive Distributed Systems via Reference Modeling"; 6th Underwater Acoustics Conference and Exhibition. (2021)

[Blobel et, al, 2022] Bernd Blobel, Pekka Ruotsalainen, Mauro Giacomini: "Standards and Principles to Enable Interoperability and Integration of 5P Medicine Ecosystems". pHealth 2022: 3-19.

[Booch 2007] Booch, G.: "The Economics of Architecture-First"; IEEE Software. 24,5 (2007) 18-20.

[Brown and McDermid 2007] Brown, A. W., McDermid, J. A.: "The Art and Science of Software Architecture"; Int. J. Cooperative Inf. Syst. 16,3-4 2 (2007) 439-466

[Capilla et al. 2016] Capilla, R., Jansen, A., Tang, A., Avgeriou, P., Babar, M. A.: "10 Years of Software Architecture Knowledge Management: Practice and Future"; J. Syst. Softw. 116 (2019) 191-205.

[Creff et al. 2020] Creff, S., Le Noir, J., Lenormand, E., Madelénat, S.: "Towards Facilities for Modeling and Synthesis of Architectures for Resource Allocation Problem in Systems Engineering"; International Systems and Software Product Line Conference. A (2020) 32:1-32:11

[Dasanayake et al. 2019] Dasanayake, S., Aaramaa, S., Markkula, J., Oivo, M.: "Impact of Requirements Volatility on Software Architecture: How do Software Teams Keep up with Ever-Changing Requirements?"; J. Softw. Evol. Process. 31,6 (2019)

[Davis 1989] Davis, F. D.: "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology"; MIS Q. 13,3 (1989) 319-339.

[Diampovesa et al. 2021] Diampovesa, S., Hubert, A., Yvars, P.: "Designing Physical Systems through a Model-Based Synthesis Approach. Example of a Li-ion Battery for Electrical Vehicles"; Comput. Ind. 129 (2021)

[Florian and Uludağ 2020] Florian, M., Uludağ, O.: "Large-Scale Agile Development Patterns for Enterprise and Solution Architects"; Proceedings of the European Conference on Pattern Languages of Programs. (2020) 29:1-29:22

[Franca et al. 2017] Franca, J., De Souza, L. J., Soares, S. M.: "Development of an Electronic Health Record Application using a Multiple View Service Oriented Architecture"; 19th International Conference on Enterprise Information Systems. (2017) 308-315.

[Franca and Soares 2015] Franca, J., Soares, S. M.: "SOAQM: Quality model for SOA applications based on ISO 25010"; 19th International Conference on Enterprise Information Systems, Proceedings. 2 (2015) 60-70

[Gaibor 2023] Gaibor, D.P.Q.: "Cloud-based Indoor Positioning Platform for Context-adaptivity in GNSS-denied Scenarios". Ph.D Thesis. Tampere University, 2023.

[Garlan 2014] Garlan, D.: "Software Architecture: A Travelogue"; Proceedings of the on Future of Software Engineering. (2014) 29-39

[Heesch et al. 2012] Heesch, U., Avgeriou, P., Hilliard, R.: "A Documentation Framework for Architecture Decisions"; J. Syst. Softw. 85,4 (2012) 795-820.

[Hoda et al. 2018] Hoda, R., Salleh, N., Grundy, J. C.: "The Rise and Evolution of Agile Software Development"; The Rise and Evolution of Agile Software Development. 35,5 (2018) 58-63.

[IEEE1220] IEEE, IEEE 1220:2005 IEEE Standard for Application and Management of the Systems Engineering Process, IEEE Standard for Application and Management of the Systems Engineering Process, (Set 2005), 1-97.

[ISO15288] ISO/IEC/IEEE, ISO/IEC/IEEE International Standard - Systems and software engineering – System Life cycle processes, 15288:2015(E), (May 2015), 1-118.

[ISO24765] ISO/IEC/IEEE, ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary, 24765:2017(E), (Aug 2017), 1-541.

[ISO29128] ISO/IEC/IEEE, Information Technology — Security techniques — Verification of cryptographic protocols, 2011.

[ISO29148] ISO/IEC/IEEE, ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering, 29148:2018(E), (Nov 2018), 1-104.

[ISO42010] ISO/IEC/IEEE, Systems and Software Engineering – Architecture Description, ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), (Dec 2011), 1-46.

[ISO42020] ISO/IEC/IEEE, ISO/IEC/IEEE International Standard - Software, systems and enterprise – Architecture processes, 42020:2019(E), (Jul 2019), 1-126.

[Ivanov 2022] Ivanov, Vladislav.: "Exploring the scope of Enterprise Architecture thinking: Perspectives from Finnish organisations". Ph.D Thesis. University of Jyväskylä, 2022.

[Junior et al. 2019] Junior, A. A. da C., Misra, S., Soares, M. S.: "A Systematic Mapping Study on Software Architectures Description Based on ISO/IEC/IEEE 42010: 2011"; Computational Science and Its Applications - ICCSA 2019 - 19th International Conference. 11623 (2019) 17-30.

[Knodel and Naab 2014] Knodel, J., Naab, M.: "Software Architecture Evaluation in Practice: Retrospective on More Than 50 Architecture Evaluations in Industry"; 2014 IEEE/IFIP Conference on Software Architecture. (2014) 115-124

[Kumar et al. 2018] Kumar, A., Lokku, D., Natarajan, S., Nori, K.: "Architecting Disruptive Digital Product-Service Systems"; INCOSE International Symposium. (Jul 2018) 1280-1295.

[Likert 1932] Likert, R.: "A Technique for the Measurement of Attitudes"; Archives of Psychology. (1932) 1-55

[Losavio et al. 2015] Losavio, F., Ordaz, O., Esteller, V.: "Quality-Based Bottom-Up Design of Reference Architecture Spplied to Healthcare Integrated Information Systems"; 9th IEEE International Conference on Research Challenges in Information Science. (2015) 70-75.

[Majumder et al. 2017] Majumder, S., Aghayi, E., Noferesti, M., Memarzadeh-Tehran, H., Mondal, T., Pang, Z., Deen, M. J.: "Smart Homes for Elderly Healthcare - Recent Advances and Research Challenges"; Sensors. 17,11 (2017) 2496.

[Martin 2018] Martin, J. N.: "Overview of an Emerging Standard on Architecture Processes - ISO/IEC/IEEE 42020"; Annual IEEE International Systems Conference. (Apr 2018) 1-8.

[Matthias et al. 2017] Matthias, G., Tamburri, D., Kazman, R.: "Towards Understanding the Social and Organizational Dimensions of Software Architecting"; SIGSOFT Softw. Eng. Notes. 42,3 (2017) 24-25.

[Mclain et al. 2016] Mclain, K. B., O'Hara-Leslie, E. K., Wade, A.: "Foundations for Assisting in Home Care"; Open Suny Textbooks. (2016).

[Michel S. Soares 2010] Soares, M. S.: "Architecture-Driven Integration of Modeling Languages for the Design of Software-Intensive Systems"; Delft University of Technology, Netherlands. (2010)

[Nakagawa et al. 2011] Nakagawa, E. Y., Antonino, P. O., Becker, M.: "Reference Architecture and Product Line Architecture: A Subtle But Critical Difference"; Software Architecture - 5th European Conference. 6903 (2011) 207-211.

[Northrop99 1999] Northrop, L.: "A Framework for Software Product Line Practice"; Object-Oriented Technology, ECOOP'99 Workshop Reader, Proceedings. 1743 (1999) 365-366

[Olumofin and Misic 2005] Olumofin, F. G., Misic, V. B.: "Extending the ATAM Architecture Evaluation to Product Line Architectures"; 5th Working IEEE/IFIP Conference on Software Architecture. (2005) 45-56.

[Olumofin and Misic 2007] Olumofin, F. G., Misic, V. B.: "A Holistic Architecture Assessment Method for Software Product Lines"; Inf. Softw. Technol. 49,4 (2007) 309-323.

[Pace et al. 2016] Pace, J. A. D., Villavicencio, C., Schiaffino, S. N., Nicoletti, M., Ceferino, H. V.: "Producing Just Enough Documentation: An Optimization Approach Applied to the Software Architecture Domain"; J. Data Semant. 5,1 (2016) 37-53.

[Pareto et al. 2012] Pareto, L., Sandberg, A. B., Eriksson, P. S., Ehnebom, S.: "Collaborative Prioritization of Architectural Concerns"; J. Syst. Softw. 85,9 (2012) 1971-1994.

[Phillips et al. 2018] Phillips, D. M., Mazzuchi, T. A., Sarkani, S.: "An Architecture, System Engineering, and Acquisition Approach for Space System Software Resiliency"; Inf. Softw. Technol. 94 (2018) 150-164

[Quezada-Gaibor et al. 2022] Quezada-Gaibor, D., Torres-Sospedra, J., Nurmi, J., Koucheryavy, Y., Huerta, J.: "Cloud Platforms for Context-Adaptive Positioning and Localisation in GNSS-Denied Scenarios - A Systematic Review"; IISE Transactions on Sensors. 22,11 (2022) 110

[Rocha et. al 2023] Rocha F.G, Misra S, Soares M.S. "Guidelines for Future Agile Methodologies and Architecture Reconciliation for Software-Intensive Systems". Electronics. 2023; 12(7):1582.

[Rodriguez 2018] Rodriguez, L. M. G.: "A Reference Architecture for Healthcare Supportive Home Systems from a Systems-of-Systems Perspective"; University of Southern Brittany, France. (2018).

[Santos et al. 2020] Santos, V. M., Misra, S., Soares, M. S.: "Architecture Conceptualization for Health Information Systems Using ISO/IEC/IEEE 42020"; Computational Science and Its Applications - ICCSA 2020 - 20th International Conference. 12254 (2020) 398-411.

[Teixeira et al. 2019] Teixeira, J. G., Pinho, N. F., Patrício, Lia.: "Bringing Service Design to the Development of Health Information Systems: The case of the Portuguese National Electronic Health Record"; Int. J. Medical Informatics 132 (2019).

[Venkatesh and Bala 2008] Venkatesh, V., Bala, H.: "Technology Acceptance Model 3 and a Research Agenda on Interventions"; Decision Sciences. 2, 39 (2008) 273-315

[Wang et al. 2017] Wang, K., Zwart, C., Wellnitz, C., Wu Jing Li, T.: "Integration of multiple health information systems for quality improvement of radiologic care"; IISE Transactions on Healthcare Systems Engineering. (2016) 169-180

[Woods 2016] Woods, E.: "Software Architecture in a Changing World"; IEEE Software. 33 (2016) 94-97

[Yang et al. 2016] Yang, C., Liang, P., Avgeriou, P.: "A Systematic Mapping Study on the Combination of Software Architecture and Agile Development"; Journal of Systems and Software. 111 (2016) 157-184.