

Architecture Conceptualization for Health Information Systems Using ISO/IEC/IEEE 42020

Valdicélio Mendes Santos[1(✉)], Sanjay Misra[2(✉)], and Michel S. Soares[1(✉)]

[1] Federal University of Sergipe, São Cristóãvo, Brazil
valdiceliomendes@gmail.com, michel@dcomp.ufs.br
[2] Center of ICT/ICE Research, Covenant University, Ota, Nigeria
sanjay.misra@covenantuniversity.edu.ng

Abstract. Health Information Systems (HIS) are complex systems which present many difficulties in their development. From the software engineering point of view, among the difficulties for HIS development are the necessity of managing and controlling data that must be held for decades, even considering the evolution of technology in the following years, as well as the necessity of cooperating with legacy systems and describing the needs and concerns of a variety of stakeholders. Considering the domain, HIS deal with human life, and errors during software development, management and operation can be catastrophic. These concerns are relevant from the software architecture point of view. Therefore, developing HIS based on a solid software architecture is a success factor that cannot be neglected. However, the processes related to the software architecture of HIS are often considered only from low level of abstraction, even for software architecture description. The ISO/IEC/IEEE 42020 defines 6 clauses for architecture process, among them the Architecture Conceptualization process is the subject of this paper. Given the importance of establishing a well-defined software architecture, and considering the difficulties of understanding an architectural standard, and also considering that ISO/IEC/IEEE 42020 has only recently been published, we propose a framework for using the Architecture Conceptualization clause, which leverages high level concepts and elements of software architecture. A case study on a HIS is described.

Keywords: Architecture conceptualization · ISO/IEC/IEEE 42020 · Health information system

1 Introduction

A variety of processes, methods, languages, frameworks and architectures have been proposed in order to deal with the challenge of developing complex software-intensive systems. Activities related to each one of these software engineering

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 398–411, 2020.
https://doi.org/10.1007/978-3-030-58817-5_30

elements have been proposed in the past decades. However, processes related to software architecture are often neglected [2,9,10,16], even though it is well-known in industry and academia that software architecture processes are considered a critical success factor [5], and even for agile software development, software architecture processes and activities have been considered important [1].

Given the importance of Software Architecture processes, a family set of architecture standards was proposed under reference number 420X0. For instance, ISO/IEC/IEEE 42010 [6] 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 [8], was set in a document with the purpose of addressing 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, is about means to organize and record architecture evaluations for enterprise, systems and software fields of application. ISO/IEC/IEEE 42020 is the main subject of study in this paper.

Each architecture 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.

The proposal in this paper is to describe a software architecture for a Health Information System in accordance with the ISO/IEC/IEEE 42020 regarding the Architecture Conceptualization process, which leverages high level concepts and elements of software architecture. Given this proposal, we identified the main aspects for the clause Architecture Conceptualization of ISO/IEC/IEEE 42020 and described a simple framework for its use in practice. This article addresses the process Architecture Conceptualization for a Health Information Systems product line, given that our purpose here is to understand stakeholders needs, including their main concerns, requirements, purposes, processes and objectives, before dealing with the architecture description using, for instance, ISO/IEC/IEEE 42010. Further processes are planned for future research.

2 Background

Some basic concepts are necessary in order to read this paper, which are briefly introduced in this section.

2.1 Health Information System

Health Information Systems (HIS) can help in a variety of operations that are needed in every healthcare organization, such as data acquisition and presentation, communication and integration of information, surveillance, information storage and retrieval, data analysis, decision support and education [14].

In a broad view, HIS also includes systems that handle data related to the activities of caregivers and healthcare organizations. As an integrated effort, they can be leveraged to improve patient outcomes, improve research and influence decision-making. A variety of health stakeholders present demands for HIS, including patients, clinicians and public health officials. These stakeholders are responsible for collecting data and compiling it in a way that can be used to improve healthcare decisions and benefit everyone.

HIS are most commonly composed of software and system elements, as well as sensors, actuators, and medical machines. Examples of constituents systems elements of a HIS are wearable biomedical sensors, activity detection systems, sleep monitoring systems, environment monitoring systems, home security systems, energy management systems, home automation systems, and companion robots [12]. Integrating the many software systems and components of HIS has been subject of research, as existing HIS systems often do not contemplate inter operation with other systems running inside home provided by different companies. For instance, Losavio et al. [11] established guidelines to integrate heterogeneous, independent, and distributed Healthcare Information Systems, allowing the creation of software products for those systems. Considering the radiology domain, with many HIS to be integrated, the authors of article [15] developed a novel technology that integrates multiple HISs.

For the healthcare domain, there are guidelines to construct HIS that provide support to chronic disease management of patients at home. As HIS deal with human life, they have to be designed to achieve non-functional requirements including interoperability, security, safety, performance and reliability, which confirms the necessity of software architecture.

2.2 Standard ISO/IEC/IEEE 42020

Standard ISO/IEC/IEEE 42020 establishes a set of process descriptions for governance and management of a collection of architectures, as well as describe 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 life span of an architecture of software systems [8].

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 in this paper is on the Architecture Conceptualization Process, it is briefly introduced in the following subsection.

2.3 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 of establishing and maintaining alignment of architectures in the architecture collection with enterprise goals, policies and strategies and with related architectures. The focus is on identifying solutions, with 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 [8].

Outcomes of the Architecture Conceptualization are the definition of the problem being addressed, establishment of architectural objectives that address the key stakeholder concerns, definition of key architecture's 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 a number of tasks which 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 them is to "Formulate principles and precepts expected to be used during execution of the life cycle processes".

The ten activities in 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.

Due to shortage of space, and also because the idea here is to deal with high abstraction level elements related to first activities in order to establish a software architecture, in this paper only activities 1, 3, 4, 6, 8 and 10 are considered, and the other activities are subject of future works.

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.

Implementation of Conceptualization Process produces the following outcomes.

- The problem being addressed.

Stakeholders vary across projects when considered in the context of first activities of software development, as for instance, those related to Requirements Engineering, which includes requirements elicitation and prioritization. A minimum set of stakeholders consists of users and acquirers, who may not be the same, and even the number of users is most often higher than acquirers.

Complex projects can impact many users and many acquirers, each with different concerns. Project requirements can necessitate to include 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. 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 [7].

- Architecture objectives that address the key stakeholder concerns.

First, activities are established to clarify stakeholders needs and concerns. After that, it should be established architecture objectives to meet those planned 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. A generic example of an objective to take advantage of an opportunity is for the architecture of interest to support reuse of an entity across various technologies, protocols, platforms, operational venues, and market segments [8].

- The architecture's key concepts and properties.

These concepts and properties could 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 could 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 with respect to the problem being addressed and the relevant stakeholder concerns.

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 to be considered when choosing among alternative solutions.

If needs, wants or expectations drive proposed solutions, then negotiations should be necessary with those stakeholders to determine which of the needs, wants or expectations are to be translated into requirements, as well as the relative priority of each one.

3 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 8.4.N activity) as appropriate to the situation, according to Clause 8 of ISO/IEC/IEEE 42020 [8].

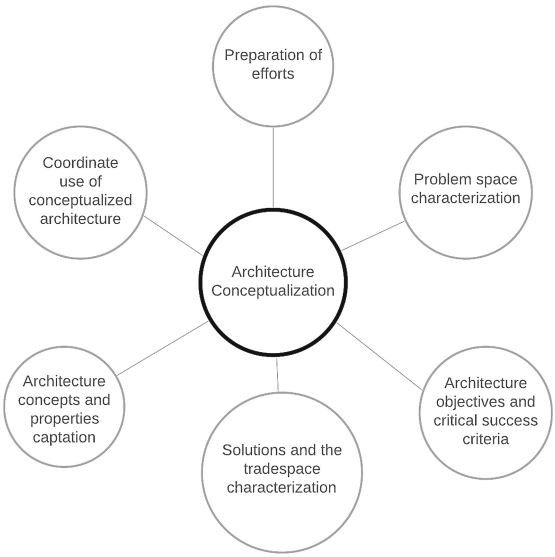

Fig. 1. Activities for architecture conceptualization framework

Figure 1 depicts the activities adopted by the Architecture Conceptualization framework, which are briefly described as follows.

Figure 2 depicts, as a work product, an architecture conceptualization plan proposed by the framework.

3.1 Preparing for and Planning the Architecture Conceptualization Effort

These are the first steps towards conceptualizing the software architecture, which occurs through accomplishment of some tasks. These tasks are concerned with

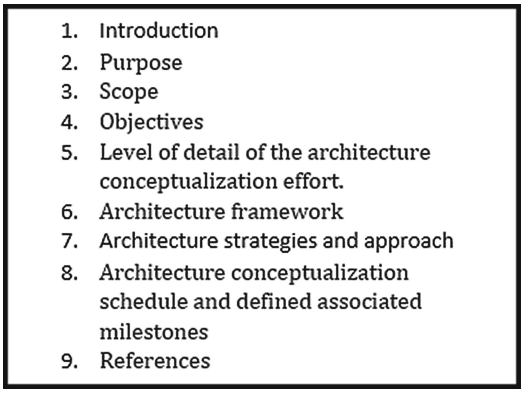

Fig. 2. An architecture conceptualization plan outline.

identifying the 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, which means that they are responsible for the decisions. In the future, once a previous decision is questioned by a stakeholder, then the rationale for each decision can be retrieved. In addition, the team responsible to deal with each one of these tasks need to be assigned.

3.2 Characterizing the Problem Space

Software architects need to understand the basic situation that they will have to deal with for developing and managing all the software systems which they are responsible. Therefore, some important tasks are to identify the problem space and determining 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 stakeholder problems and needs.

It is important to adopt a systematic approach to identify the most effective solution given a number of possibilities. An evaluation of alternative solutions should be provided by developers and 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 is created knowledge about the whole situation, providing additional experience for architects, developers and other stakeholders.

3.3 Establishing Architecture Objectives and Critical Success Criteria

In this activity, architectural objectives are identified and defined to address stakeholders' concerns, requirements, risks, constraints 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 relationships between them.

Relevant scenarios are identified for each problem and opportunity regarding boundary conditions, root causes and drives. It is necessary to check if there are gaps, shortfalls of current or planned solution in dealing with respective problems and opportunities. After all these tasks, architectural objectives are identified and defined to address stakeholders' concerns, requirements or quality attributes. Critical success criteria are defined to assess whether the problem has been solved, checking whether the final objectives have been achieved.

3.4 Characterize Solutions and the Tradespace

When determining key criteria for system quality, it is commonly known that stakeholders will identify a number of 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, 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 analyse tradeoffs between all possibilities. For each architectural relevant solution, an analysis should provide a full evaluation of 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 easiness of using instead of performance.

3.5 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 [8]. The Elaboration Process deals with description, views and models that captures architecture concepts and properties.

This activity is related to capture architecture concepts in terms of views and models. This is possible through identification of relevant stakeholders, key aspects, and definition of the purpose, scope, breadth and depth of the architecture views and models. Nevertheless, 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 [8].

This activity also provides registry of architectural concepts, rationales, properties, decisions, guidelines and characteristics. Achievement of architecture and solution characteristics is provided from the identified processes, activities and tasks in this stage. Besides, the architecture components, their interdependence and interactions are captured.

The ISO/IEC/IEEE 42010 [6] Standard should be used to describe relevant viewpoints, views and models. These artifacts are explored further in clause 10, Architecture Elaboration process of standard ISO/IEC/IEEE 42020.

3.6 Coordinating Use of Conceptualized Architecture by Intended Users

Key stakeholders should validate the conceptualized architecture through feedback of the description, views and models. Besides, there are other users of architecture conceptualization information who are responsible for evaluation or description of the architecture, as well as reviewing, managing, and designing the software architecture.

There are many ways that can be considered for validation. For instance, workshops in which each stakeholder can express its opinions about relevant aspects of the architecture described so far. Other possibilities are evaluation of technical manuals by stakeholders. Each item that can be improved is identified and documented for future reference, which improves architectural knowledge for the organization, and also improves software development processes.

After all the efforts to accomplish the previous tasks and activities, it is equally important to communicate the work products to the interest parties. Key stakeholders should validate the conceptualized architecture through feedback of the description, views and models. Besides, there are other users of architecture conceptualization information who are responsible for evaluation or elaboration of the architecture, review, management, design engineering, and so on.

Each item that can be improved is identified and documented for future reference. It is also required to monitor the use of architecture conceptualization information with the interested parties to make sure that all stakeholders have received it and are registering feedback on the contents of architecture work products. These feedback should be incorporated into architectural work products such as views, models and further descriptions.

When the conceptualized architecture is not clear, mature or complete enough to the stakeholders, it could be necessary to review current architectural products, and maybe refine or prepare a more complete set of architecture views and models.

4 An Example of Architecture Conceptualization for HIS

Work products of the Architecture Conceptualization Process are an architecture conceptualization plan, architecture conceptualization status report, a problem space definition report, architecture objectives, a quality model, and architecture views and models. From all of these work products, status report is not considered in this paper, as it refers to Activity Monitor, which is not addressed in this paper.

Regarding item 3.1 and relating to the architecture conceptualization plan in Fig. 2, the general nature of the problem area is a Health Information System product line capable of dealing with interoperability between legacy systems. The purpose is to define a conceptualization architecture which meet requirements of key stakeholders and capture early decisions about high-level design. These stakeholders have a special role to play in the architecture development effort because they will deal with the resulting solution for years to come.

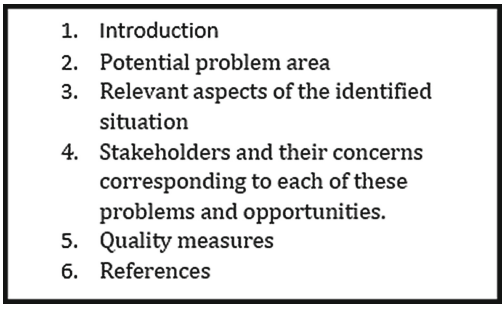

Fig. 3. A problem space definition report outline.

Concerning item 3.2, characterization of the problem space, the problem space definition report is outlined in Fig. 3. Patients may need reminders or help to take their medications, and often need assistance with ambulation (walking) and transferring from a bed to a chair or wheelchair, or getting in and out of the shower. Many patients have adaptive equipment, such as walkers, wheelchairs, canes, and prosthetic devices, which assist them in moving around their home [13].

Existing HIS are sometimes proprietary, monolithic, present high coupling between modules, and expensive solutions for patients and their relatives. Most of such systems do not consider their inter-operation with existing, distributed, and external systems, such as Electronic Health Records (EHR), Patient Health Records (PHR), emergency systems (e.g., ambulances, fire departments), and other HIS.

For the healthcare domain, there are guidelines to construct HIS that provide support to chronic disease management of patients at home. As HIS deal with human life, they have to be designed considering important non-functional

requirements including interoperability, security, safety, performance and reliability.

Patients are key stakeholders of HIS. An example of key patient user requirements, described at high level of abstraction, are as follows:

- Patients shall have all the relevant information needed to allow the management of their own health and their interaction with the health system.
- Patients shall be recognised when they access the system and can quickly see relevant health details.
- Patients shall access their own health records and maintain a health diary.
- Patients shall have high levels of trust in the security and confidentiality of services they are using.
- Patients shall have the ability to access services while traveling and on the move.

Care Providers are also key stakeholders. An example of key user requirements, described at high level of abstraction, are as follows:

- Care providers shall have the ability to constantly monitor and interact with patients despite distance and mobility of either party.
- The healthcare delivery environment shall be safe and supportive of patient-care.
- Care providers shall access information from other systems to support their decision.
- Care providers shall be able to interact with patients regardless their electronic devices.

Considering these two important stakeholders, related concerns for each one is as follows:

- Patients: Affordability, Availability, Dependability, Reliability and Usability.
- Care providers: Availability, Flexibility, Maintainability, Reliability, Usability, Resilience.

According to item 3.3 and as a work product, some architecture objectives and architectural significant requirements are described as follows.

- Domain requirements. The reference architecture must enable the development of HIS that, remotely, estimate and provide the patient physical status at any time, e.g., informing the status of patient through the analysis of his/her physiological functions and body systems (e.g., cardiovascular, nervous, respiratory) signs and symptoms.
- Interoperability and integration requirements. The architecture must enable the development of HIS that allow interoperable communication between legacy systems.
- Reliability requirements. The reference architecture must enable the development of HIS that offer fault-tolerant mechanisms for constituent systems interactions.

Another work product is a quality model. SOA Quality Model (SOAQM) [3,4] will be adopted. This model is based on ISO/IEC/IEEE 25010 and has quality attributes that are considered relevant during development of SOA applications. According to HIS objectives, the following characteristics of SOAQM will be measured:

- Compatibility.
- Usability.
- Reliability.
- Security.

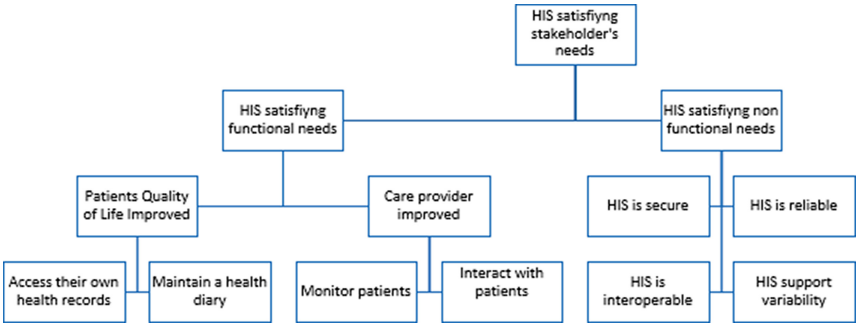

Fig. 4. High level mission view of HIS.

Concerning item 3.5 and the architecture views and models as a work product, Fig. 4 describes a high level mission view of HIS.

Regarding item 3.6, it is available to users an online form to register feedback on the architecture and on the contents of the architectural work products.

5 Discussion and Conclusion

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 document deals with Conceptualization Activities of a software architecture for a HIS product line. Following the activities and tasks in Clause 8 is helpful for starting the architectural effort.

However, 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 the 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, which means that the software 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, this is subject for future work.

ISO/IEC/IEEE 42020 presents 6 Architecture processes with 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 architecture elements, representing important entities of a software architecture, but are at a lower level of abstraction when compared to the framework proposed in this paper. For instance, task "l) Develop an architecture description consisting of relevant viewpoints, views, models, 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 application of Architecture Description defined by ISO/IEC/IEEE 42010 [8].

The work carried out concluded the activities and tasks proposed by Clause 8 of ISO/IEC/IEEE 42020, leading to a mature characterization of the solution proposed by the architecture. In a real situation, there would be many solutions, and the interested parties could decide the best one according to their needs.

A more complete form of architecture description, using more specific elements including views and models is subject of the Architecture Elaboration process, as presented in the ISO/IEC/IEEE 42020 standard [8]. Activities 1, 3, 4, 6, 8 and 10 from Architecture Conceptualization are considered in this paper, and the other activities (2, 5, 7 and 9) are subject of future works.

Acknowledgments. This study was financed in part by FAPITEC and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

1. Abrahamsson, P., Babar, M.A., Kruchten, P.: Agility and architecture: can they coexist? IEEE Softw. **27**(2), 16–22 (2010)
2. 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), 1–21 (2019)
3. França, J., Soares, M.S.: SOAQM: quality model for SOA applications based on ISO 25010. In: ICEIS 2017–Proceedings of the 19th International Conference on Enterprise Information Systems, vol. 2, April 2015

4. França, J., de Souza Lima, J., Soares, M.S.: Development of an electronic health record application using a multiple view service oriented architecture. In: ICEIS 2017–Proceedings of the 19th International Conference on Enterprise Information Systems, pp. 308–315, January 2017
5. Garlan, D.: Software architecture: a travelogue. In: Herbsleb, J.D., Dwyer, M.B. (eds.) Proceedings of the on Future of Software Engineering, FOSE 2014, Hyderabad, India, 31 May–7 June 2014, pp. 29–39. ACM (2014)
6. 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), pp. 1–46, December 2011
7. ISO/IEC/IEEE: ISO/IEC/IEEE International Standard - Systems and software engineering-vocabulary. 24765:2017(E), pp. 1–541, August 2017
8. ISO/IEC/IEEE: ISO/IEC/IEEE International Standard - Software, systems and enterprise - Architecture processes. 42020:2019(E), pp. 1–126, July 2019
9. Júnior, A.A.C., Misra, S., Soares, M.S.: A systematic mapping study on software architectures description based on ISO/IEC/IEEE 42010:2011. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11623, pp. 17–30. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24308-1_2
10. Júnior, A.A.C., Misra, S., Soares, M.S.: ArchCaMO - a maturity model for software architecture description based on ISO/IEC/IEEE 42010:2011. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11623, pp. 31–42. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24308-1_3
11. Losavio, F., Ordaz, O., Esteller, V.: Quality-based bottom-up design of reference architecture spplied to healthcare integrated information systems. In: 9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015, Athens, Greece, 13–15 May 2015, pp. 70–75. IEEE (2015)
12. Majumder, S., et al.: Smart homes for elderly healthcare - recent advances and research challenges. Sensors **17**(11), 2496 (2017)
13. Mclain, K.B., O'Hara-Leslie, E.K., Wade, A.C.: Foundations for Assisting in Home Care. Open Suny Textbooks (2016). https://milnepublishing.geneseo.edu/home-health-aide
14. Teixeira, J.G., de Pinho, N.F., Patrício, L.: Bringing service design to the development of health information systems: the case of the Portuguese national electronic health record. Int. J. Med. Inform. **132**, 813–817 (2019)
15. Wang, K., Wellnitz, C., Zwart, C., Li, J., Wu, T.: Integration of multiple health information systems for quality improvement of radiologic care. IISE Trans. Healthc. Syst. Eng. **7**(3), 169–180 (2016)
16. Woods, E.: Software architecture in a changing world. IEEE Softw. **33**, 94–97 (2016)