A New Proposed Software Engineering Methodologyfor Healthcare Applications Development

Abdullah Al-Dahmash, Samir El-Masri

Department of Information Systems, College of Computers and Information Sciences
King Saud University, Riyadh, KSA

***Abstract**: The health Information system is widely expected to increase patient safety and reduce medical errors. Design and implementation of a healthcare system need to consider the software engineering principles and methodologies. In this paper, a literature review has been carried out for healthcare systems implementation. The paper also discusses three common software methodologies and their uses in the healthcare domain. A Software Engineering Methodology for Healthcare Applications development (SEMHTA) will be proposed. SEMHTA has been introduced in this paper to solve the healthcare software development challenges. This paper also provides a comparison between the most common software engineering methodologies and the new proposed one.*

***Keywords:**Healthcare system, Software Engineering Methodology, Healthcare software implementation.*

I. Introduction

The aim of the healthcare systems is to achieve the best possible support for patient care and to provide the optimal medical care. The healthcare systems should support good health, fair financial contribution, and excellent services. Healthcare industry has rapidly grown in the last decade, especially in developing countries. Medical software implementation represents one of the major future challenges in healthcare industry. Using healthcare information systems will increase the number of health services, improve the quality of care and reduce the medical errors. The use of information systems in the healthcare industry is dramatically increasing. A successfully implementation of healthcare software in healthcare organization appears to be a difficult task. Software engineering for healthcare systems is an emerging field for software developers and IT specialists. Software engineering methodologies are being used in healthcare the same way used in other industries. The nature of systems and industry such as healthcare systems should be considered in the software-development methodology. A healthcare system has properties and features that need to be addressed, as they are different from other systems. For these reasons, we are proposing a new software engineering methodology for healthcare applications SEMHTA. This methodology is supposed to solve the software-development issues and challenges in the healthcare domain. The main reason for developing a new software engineering methodology for healthcare such as SEMHTA is the need for building a reliable and secure healthcare system with a high performance.

II. Literature Review

In general, software engineering involves the analysis, design, implementation, testing, maintenance and documentation of software systems. Since the concept of software engineering as a discipline was known more than three decades ago, there have been different definitions of software engineering. The Software Engineering Institute at Carnegie Mellon University gives the following definition: software engineering is that form of engineering that applies the principles of computer science to achieving cost-effective solutions to software problems [1]. The IEEE Computer Society in a joint effort with the ACM agreed on the following definition: The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [2]. Software engineering methodology can be defined as a group of techniques and tools used in the development of software. Methodology should describe each phase of the software-development life cycle. Software methodology normally does not specify the details. These details usually left to the organization’s needs and requirements. The major difference between a methodology and a framework is that a framework should be used at a more abstract level [3]. In the last two decades healthcare applications and the use of information systems in health industry have grown rapidly. The idea of healthcare today is much different from what it was 10 years ago. Healthcare is now more than simply medical care and treatment. The goal of healthcare is to provide more efficient services to enjoy a healthy society. Therefore, healthcare information system is a comprehensive, integrated and designed to manage the medical and administrative aspects of a medical organization [4]. The principal goal of IT application in healthcare is the quality improvement of medical care. Therefore, the healthcare system has to be characterized by a high reliable system [5]. Electronic medical information systems have been identified as an enable to keep healthcare systems sustainable, reliable and safety. A 2005 RAND study estimates that the U.S.A. could save $81 billion annually and help to improve the healthcare industry through the adoption of high quality electronic medical information systems [6]. The definition of "medical software" must consider the new aspects of management and maintenance of the software engineering. Moreover, the technological development in the healthcare field has made medical software more invasive and more maintainable [7]. Healthcare software systems should be designed under several aspects that may not be realized by the software engineers. Successful implementation requires a partnership between healthcare professionals and software engineers. Healthcare professionals and software engineers can have different expectations and understanding of the system. Healthcare professionals have a little understanding of software and may not be motivated to use it. Software engineers need to be aware of the nature and requirements of the medical software [8].

III. RESEARCH QUESTION

In order to develop a new healthcare application, there should be a software engineering methodology followed to produce the system. There are a lot of software engineering methodologies available. What are the advantages and disadvantages of each methodology? Do those methodologies support non-functional requirements for healthcare projects such as reliability, performance, security and availability? What is the new proposed methodology? These are the questions that will be discussed in this paper.

IV. Most Common Software Engineering Methodologies

In this section, the most common software engineering methodologies and their use in healthcare projects will be discussed:

Waterfall Model: The waterfall model is a sequential design in which deployment seen as a waterfall through the stages of planning, analysis, design, implementation and testing. Issues such as time-consuming and unclear requirements at the early stages are seen as obstacles for the waterfall model. In response to these issues and other's issues of the waterfall model many modified waterfall models have been introduced [9].

Iterative Model: In this model, the overall life cycle is composed from several iterations. The iteration is consisted from a couple of activities: planning, analysis, design, implementation and testing. At the end of the iteration there should be an iteration delivery which is in most cases the iteration release. Most iteration releases are internal. The final iteration is the completed product [10]. Figure [1] describes the iterative model.

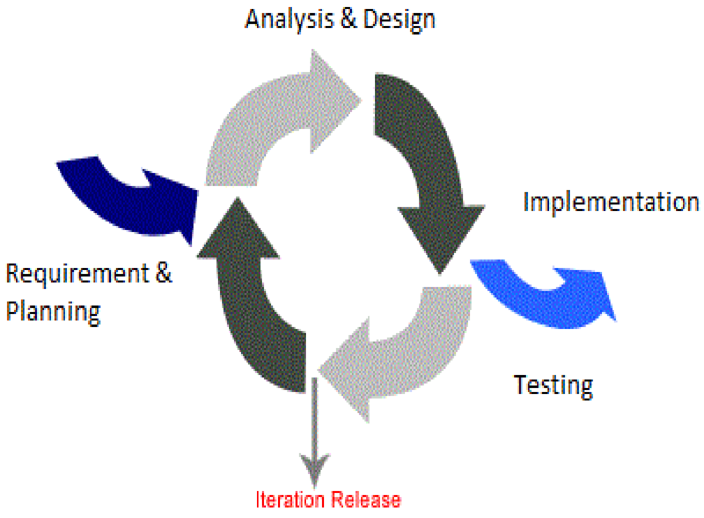


Figure1: The Iterative Model

Agile Model: The agile model defines a set of principles and practices, which aims to minimize the documentation activities. The main goal of an agile model is to do the development activities with high professionalism and less efforts and documents. The major advantage of using an agile model is the development time factor. The disadvantage of this approach is its needs of high knowledge and experienced team [11].

V. Healthcare Software Issues

Health care is a complex socio-technical system. There is an opinion that information technology can induce errors, instead of correcting them [12]. There are several issues need to be considered in dealing with healthcare systems. Issues such as reliability, security and performance are the main issues in this field.

Performance: performance is a non-functional requirement which is a very important requirement for any software. From a healthcare software perspective, the performance issue can be seen in form of [13]:

Performance issue related to the data: there are two important issues with the data in health-care projects, which are data accuracy and data availability. Performance issue related to healthcare staff: performance issues from health-care staff’s viewpoints: appropriateness of information, the communication via the healthcare system and integrity of the system.

Performance issue related to patients: at the end, there is a relationship between the healthcare system and the patients. The healthcare system provides services to the patient. Performance issues from patient’s perspective: reduce the waiting time, privacy and increase of satisfaction.

Security: Security's requirements describe which security mechanisms to use. Security's requirements focus on what it should be achieved – not how [14]. Health care system is not like other IT systems, it deals with patient’s data and other confident data. So security requirement is one of the most challenges that facing software engineers when they try to develop a healthcare software. Security's requirement of healthcare software should be focused on areas such as patient's data and medical record data protection.

Reliability*:* reliability analysis of a healthcare system is similar with reliability engineering in other projects. There are a lot of methodologies to analyze the project reliability such as software reliability analysis, hardware reliability analysis and human reliability analysis (HRA). In [15] the healthcare system includes four components hardware, software, human factor and organization factor. To analyze the reliability of a healthcare system should throw a measure of performance of each one from these components.

Availability: the availability of a hardware/software is the percentage of time when the system is up and running [16]. In healthcare software, availability is a critical characteristic. In healthcare, all medical systems should be 24/7 for an emergency event and for a surgery. So the software methodology for a healthcare project needs to consider the availability of the system as a demand and as a challenge too.

VI. Software Engineering Methodology For Healthcare Applications (Semhta)

The new proposed methodology should solve issues faced by software engineers and healthcare specialists whendealing with developing a new system. SEMHTA will focus on the healthcare system properties and nature. The new methodology considers the healthcare background of the end user and the impact on the development activities. SEMHTA is built on four main phases: Planning and Analysis, Design and Build, Implementation, Validation and delivery. Each phase contains several activities, and each phase is interconnected with other phases. SEMHTA actually try to merge the software engineering life cycle with the project management activities. SEMHTA framework is described in figure [2].

Planning and Analysis Phase

The Planning Phase is often the most challenging phase to build healthcare applications. Planning phase involves the creationof a set of plans to help guide technical and healthcare teams through the execution of the system. The purpose of the Planning Phase is to plan all project processes and activities. In addition, to build channels between the healthcare specialists and the software-development team in order to improve the communication between them.

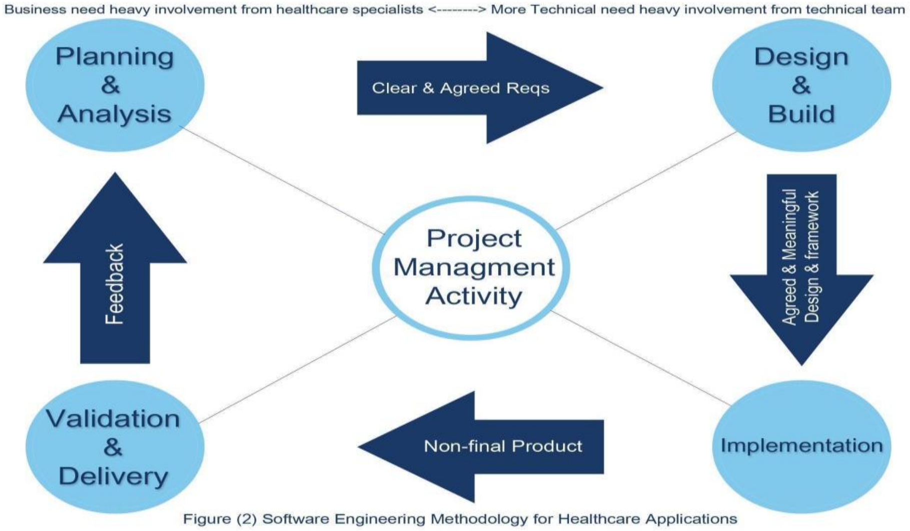
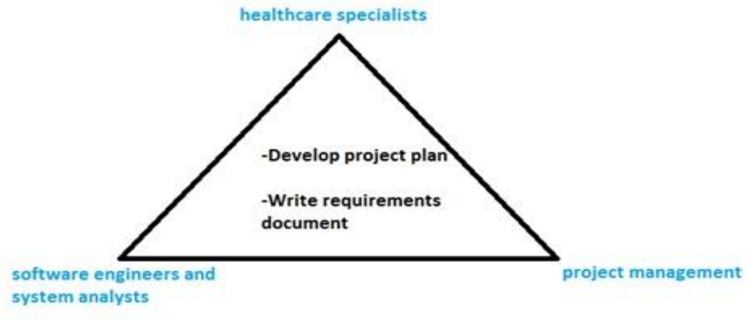

Business need heavy involvement from healthcare specialists <--------> More Technical need heavy involvement from technical team
Planning & Analysis
Clear & Agreed Reqs
Design & Build
Feedback
Project Managment Activity
Agreed & Meaningful Design & framework
Validation & Delivery
Non-final Product
Implementation

Figure (2) Software Engineering Methodology for Healthcare Applications

There must be a common understanding of the system among all healthcare staff and project development team. The deliverable result at the end of the planning phase is a complete project plan document. During the analysis phase, the main activity is to gather hospital's requirements and environmental considerations. The analysis phase defines the requirements of the system, independent of how these requirements will be implemented. Many techniques are used to gather and manage system requirements. In SEMHTA, the workshops' technique is the one used for manage the requirements. The requirements workshop involves the users from different background such as physicians, informaticists, top management and even some hospital's patients. The deliverable result at the end of this phase is a detailed and analysed requirements document. The requirement document does not specify the architectural or implementation details, but specifies information at the higher level of description. Requirements document should be written in simple words understandable by all healthcare and business users. Involvement of the healthcare specialist is a key factor to build a requirements document that reflects the essentially demand of the system. Involvement of the end user in all requirements identification activities will guarantee a high reliable system and perfect operational solution. Planning and analysis activities are described in figure [3].


healthcare specialists
-Develop project plan
-Write requirements document
software engineers and system analysts
project management

Figure[3] Planning and Analysis phase in SEMHTA

Design and Build Phase

In the design phase, the architecture of the healthcare system is established. This phase starts with the requirements document delivered by the planning and analysis phase. The architecture defines components, interfaces and behaviors of the healthcare system. The deliverable result at the end of the design phase is the design document. The design document describes a technical plan to implement the requirements. Details on computer programming languages and environments, machines, packages, memory size, platform and many other technical details are established and agreed on at the end of the design phase in SEMHTA. The requirements document must guide all processes and activities in the design phase. SEMHTA suggests that there should be a significant participation from the healthcare specialists in defining the design document.

Implementation phase

In the implementation phase, the development team builds the components that identified in the design phase. In SEMHTA, with a given design document from the design and build phase and the requirements document from the planning and analysis phase, the development team should implement correctly what it has been requested. The implementation phase of SEMHTA deals with issues of quality, performance, libraries and debugging. The deliverable result at the end of the implementation phase is the product itself.

Validation and Delivery phase

In many software engineering methodologies, the validation phase is a separate phase which is performed by a different team after the completion of the implementation. Unfortunately delegating validation and testing to another team leads to many errors and mistakes undiscovered “No one knows the system more than the development team". SEMHTA suggests that the testing and validating activities should be performed by a joint team. The joint team members are development team, testers, healthcare specialists and end users. This will lead to a secure and available healthcare system. Again, SEMHTA suggests that all validation activities should be driven by the requirements document. In case of any problem in the product, more clarifications, and feedbacks should go back to the analysis phase.

SEMHTA and Project Management Activities

It is very clear that SEMHTA makes a bridge between software engineering and project management activities. In the planning and analysis phase, the concept of building the project plan is a good example of this relationship. In SEMHTA, in order to develop software, there should be a heavy involvement from healthcare specialists. Furthermore, an important factor that will lead this development to a success is to apply all project management concepts and practices into the software engineering methodology.

VII. Comparisons Of Software Engineering Methodologies

In table 1, there is a comparison between the software engineering methodologies that have been described early with our new proposed SEMHTA. The comparison is based on the healthcare system issues such as reliability, performance, security and availability. This comparison is only valid for healthcare applications.

Table 1. Comparisons of Software Engineering Methodologies

Methodology Issue	Waterfall	Incremental	Agile	SEMHTA
Reliability	Low	High	Low	High
Performance	Moderate	High	High	High
Security	High	Low	Low	High
Availability	Moderate	Moderate	Moderate	Moderate

VIII. Conclusion And Future Work

In this paper, a new software engineering methodology SEMHTA has been proposed. SEMHTA methodology is composed from four main phases; planning and analysis, design and build, implementation and validation and delivery. The new proposed methodology specially built for healthcare applications and systems. A brief review has been done on the common software engineering methodologies. The paper has discussed some software-development issues when dealing with healthcare industry. The paper provides a comparison across the common software methodologies with SEMHTA showing the strengths of the new proposed methodology over others. The comparison between the methodologies focuses on the healthcare's software-development issues. Future research activities may deal with areas to improve SEMHTA efficiency, build SEMHTA architecture model and study the impact of SEMHTA on a healthcare project. Future research in this area can be on fields such as study nature and characteristics of healthcare systems, the relationship between healthcare specialists and software engineers within a healthcare project life cycle and many other areas.

References

[1] Ford, G., 1990 SEI Report on Undergraduate Software Engineering Education, CMU Technical Report no. CMUlSEI-90-TR-003, 1990.

[2] IEEE. IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12, 1990.

[3] Mnkandla, E., About Software Engineering Frameworks and Methodologies, IEEE, 2009.

[4] Key, W. and Liu, Z., Software Engineering in Public Health: Opportunities and Challenges, International Conference on Computer Distributed Control and Intelligent Environmental Monitoring, 2010.

[5] Zaitseva, E. and Levashenko, V., Reliability Analysis of Healthcare System, the Federated Conference on Computer Science and Information Systems, 2011, pp. 169–175.

[6] Weber-Jahnke, J. and Price, M., Engineering Medical Information Systems: Architecture, Data and Usability & Security, 29th International Conference on Software Engineering, 2007.

[7] R. Miniati, F. Dori, E. Iadanza and M. F. Medici, Medical Software Management:A failure analysis approach for maintenance and safety plan, 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina, 2010.

[8] Ried, L. and et.all, Design and Implementation of a Hospital Quality Assurance Program (H-QAP), IEEE, 2011.

[9] Birrell, N., Ould, M., A Practical Handbook for Software Development, Cambridge University Press, USA, 2003, pp.16-21.

[10] Larman, C., Agile and Iterative Development: A Manger 'sGuide,Addison-Wesley, USA, 2004, pp. 9-12.

[11] Ambler, W., Agile Model Driven Development (AMDD), XOOTIC MAGAZINE, 2007.

[12] Aarts, J., Towards safe electronic health records: A socio-technical perspective and the need for incident reporting, Health PolicyandTechnology, 2010, pp.8-15.

[13] Ozdemir, D. and Gozlu, S., *Investigation of Performance Criteria for Health Information Systems*, *PICMET 2008 Proceedings*, 2008.

[14] *Jensen, J., Tondel, I. and Anderson, H., Reusable Security Requirements for Healthcare Applications*, *International Conference on Availability, Reliability and Security*, 2009.

[15] Zaitseva, E., Reliability Analysis Methods for Healthcare system, *3rd IntConf on Human System Interaction*, Poland, 2010, pp.211-216.

[16] Candea, G. and FOX, A., Designing for High Availability and Measurability, *1st Workshop on Evaluating and Architecting System Dependability (EASY)*, 2001.