Computer Standards & Interfaces 45 (2016) 26–36

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

An improved strategic information management plan for medical institutes

Tsung-Han Yang [a,1], Cheng-Yuan Ku [b,2], David C. Yen [c,*], Wen-Huai Hsieh [a,d,3]

[a] Department of Information Management, National Chung Cheng University, Chia-Yi County, Taiwan, ROC
[b] Department of Information Management and Finance, National Chiao Tung University, Hsinchu City, Taiwan, ROC
[c] School of Economics and Business, SUNY at Oneonta, Oneonta, NY 13820, United States
[d] Chang-Hua Hospital, Ministry of Health and Welfare, Executive Yuan, Chang-Hua County, Taiwan, ROC

ARTICLE INFO

Article history:
Received 8 May 2015
Accepted 1 November 2015
Available online 10 November 2015

Keywords:
Strategic information management plan
Electronic medical record
Software development
Knowledge engineering and management
Software quality engineering

ABSTRACT

The driving force behind software development of the Electronic Medical Record (EMR) has been gradually changing. Heterogeneous software requirements have emerged, so how to correctly carry out development project has become a complex task. This paper adopts the knowledge engineering and management mechanism, i.e., CommonKADS, and software quality engineering to improve existing strategic information management (SIM) plan as a design methodology to help software implementation for medical institutes. We evaluate the adopting performance by a real case that examines the maturity level of the architecture alignment between the target solution in the proposed SIM plan and the built medical system.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

There exist numerous barriers for software development in the medical sector. Medical sector is a highly professional and time-varying industry, and the types of medical resources are so complicated and difficult to manage. Besides, the technical staff in hospital is the minority and normally does not have enough knowledge and experience of implementing medical software development projects, so many medical institutes frequently cooperate with outsourcing companies. However, the implementation of software development is prone to fail, because some hospital staffs are reluctant and have difficulty to communicate with software developers about vague and dynamic software requirements due to tight schedule and knowledge gaps. Moreover, the development of medical system is often influenced by both internal and external environments. With the highly heterogeneous software requirements, how to develop a successful project of medical system is not easy. In addition, there exists the danger that hospital managers may overlook the interplay between the medical system and organizational performance, if they uncritically trust the promises made by outsourcing companies [10]. Hence, the negative consequences of software development in medical sector occur more frequently than other industries, such as gap between expectations of active users, failure to learn from past projects, and lack of integration. These results indirectly cause poor service quality, waste of medical resources, or even medical malpractice. Rahimi et al. point out that the healthcare environment needs a suitable methodology to develop and evaluate the medical systems [17,18].

Most standards in medical sector focus on specifications of electronic health records (EHR) as follows. BS EN 13606-1 is a European standard that aims to define rigorous and stable architectures for EHR. The openEHR is a health informatics specification for EHR. The Clinical Document Architecture Release 2.0 (CDA R2) addresses universal requirements for exchange and management of structured clinical documents. Even JCAHO, the American oldest and largest standards-setting and accrediting body in the medical sector, has seldom laid great stress on medical software standards. Thus, there are few approaches proposed for effectively managing and developing medical software. The BS EN 12967 presented a framework to describe system views, but it does not provide any details for system development [13].

In general, CMMI for development (CMMI-DEV) and ISO standards for software engineering are popular, because they play a vital role in integrating, regulating, and optimizing the existing practices and fundamental theories for the development of better software products. However, the adopters must follow rigorous and disciplined approaches, emphasize documentation, and take care of each detail during the software development process based on these two categories of standards. Therefore, they are too costly and cumbersome for the medical sector, because the resources and competences of hospital are usually unable to satisfy these requirements. In addition, the most important thing is that these standards lack the flexibility to handle the complicated and changing need from every medical department.

* Corresponding author at: Tel.: +1 607 436 3458; fax: +1 607 436 2543.
E-mail addresses: h730615@gmail.com (T.-H. Yang), cooper.c.y.ku@gmail.com (C.-Y. Ku), David.Yen@oneonta.edu (D.C. Yen), q0928705848@yahoo.com.tw (W.-H. Hsieh).
[1] Tel.: +886 5 2721500; fax: +886 5 2721501.
[2] Tel.: +886 3 5712121#57413.
[3] Tel.: +886 928705848.

http://dx.doi.org/10.1016/j.csi.2015.11.001
0920-5489/© 2015 Elsevier B.V. All rights reserved.

Table 1
Comparison of the modeling frameworks.

Modeling Framework	CommonKADS	PROTÉGÉ series	MIKE
Scope of modeling	CommonKADS product	Domain knowledge and problem solving method(PSM)	Reasoning behavior of experts and process of knowledge–based system design
Methodology	CommonKADS life cycle approach	Domain ontologies, domain independent methods, and mapping relations	Knowledge acquisition, design, implementation, and evaluation
Levels of specification	Conceptual, detailed, and operational (organization model, task model, agent model, knowledge model, communication model, and design model)	Domain independent, domain dependent (specification of ontological knowledge, content knowledge,and case data)	Raw, conceptual, detailed, and operational
Languages	Conceptual modeling language (CML) using Backus–Naur Form (BNF) notation	Model	Graphical representation language, NewKARL, and DesignKARL
Support	Methodological support, libraries, and automated tools	Methodological support, libraries, and automated tools	Methodological support, libraries, and automated tools
Input	Requirements, problem solving knowledge, domain knowledge, and environment	Problem solving knowledge, domain knowledge, and environment	Requirements, problem solving knowledge, domain knowledge, and environment
Output	Specification, operationalization, and documentation	Specification and operationalization	Specification, operationalization, and documentation

On the other hand, to develop the suitable software architecture [3], enhance the software quality [7], and address the above-mentioned issues, the SIM plan originally proposed to support medical software development is a candidate solution [24]. SIM describes how to organize information management, the tasks of different working groups, and the software requirements from various stakeholders. Namely, it is a blueprint for planning, directing, and monitoring the development of medical system. Furthermore, by producing the suitable information strategies based on the strategic goals of a hospital, it provides guidelines for strategic planning activities to support hospital information management which is crucial for the development of medical systems. However, SIM plan is not yet formalized or mature enough for hospitals. Aiming to improve the existing SIM plan to address software development problems, this study introduces the method of knowledge engineering and management [12] to modify the SIM plan. In this manner, our solution can answer the research question on how the proposed design framework is utilized effectively to improve the software quality of medical system and build the suitable software architecture for satisfying different software requirements of every stakeholder in the medical software development project.

The organization of this paper is as follows. Section 2 presents the related works about the SIM plan and discusses knowledge engineering and management. Section 3 precisely explains the modules of the improved SIM plan and presents the interrelationships. Sections 4 and 5 demonstrate the adopting procedure of the proposed SIM plan of a real case and discuss the results by using the concept of architecture alignment. Finally, Section 6 summarizes implications learned from this study.

2. Theoretical background

To implement the medical software development project, there should be a bridge between the project leader, users, information system staff of hospital, and outsourcing companies to generate comprehensive service level agreement. Thus, a concrete SIM plan, as a communication channel, is needed to assist in determining the project implementation methods and steps. As for the importance of SIM, Winter et al. claimed that without proper strategic planning, it would be a matter of chance if a hospital information system fulfills strategic information goals [25]. However, even if there exist some guidelines of the SIM plan [5,6], most of them are too general and vague to be applied for a specific medical institute. Besides, these guidelines are not concrete enough, so they cannot be directly adopted. Furthermore, these guidelines focus on theoretical background without the support of best practice, so they may not properly guide the inexperienced adopters, and the experiences gained from SIM plan projects cannot be stored and referenced.

Because many medical workflows are very complex, the medical system is knowledge intensive with manifold data sources. To make the SIM plan a formalized and feasible mechanism, knowledge engineering method [21] is adopted. Knowledge engineering method is used to analyze, design, and evaluate the software with resource allocation from multiple aspects. Studer et al. pointed out that the purpose of knowledge engineering is to transform the process of constructing knowledge-based systems from an art into an engineering discipline [20], so it can provide a useful framework to improve the existing SIM plan. Moreover, knowledge management assists in acquiring, creating, representing, and distributing knowledge within and between organizations [8,11]. Extending from knowledge engineering and management, some modeling frameworks have been proposed to handle the aspect of model-based approach.

CommonKADS [19] is a methodology used in the domain of knowledge engineering and management for defining the structure of expertise models. The cornerstone of CommonKADS is knowledge acquisition design system (KADS), and its primary advantage is that knowledge engineers can use a variety of models to guide the knowledge-acquisition process by refining and combining them into a fully specified model. In addition to CommonKADS, Model-based and Incremental Knowledge Engineering (MIKE) [1] emphasizes a formal and executable specification of the expertise model as a result of the knowledge acquisition phase. PROTÉGÉ series [9] exploit the notion of ontology to support users to develop an ontology-based knowledge management system.

In order to choose the most suitable modeling framework, the major characteristics of above-mentioned methods are listed in Table 1. The comparison is based on the purpose-driven framework proposed by Brazier and Wijngaards [4]. Since our purpose is to modify the existing SIM plan as a formal software development mechanism, the selecting criteria should focus on the topics related to software engineering. Therefore, it may not be appropriate to choose an ontology-related modeling framework, PROTÉGÉ series, even though it is useful for developing custom-tailored editing environments to handle semantic web applications. For levels of specification in Table 1, the CommonKADS adopts expertise models to describe enterprise status, but MIKE has no concrete method. It is not easy to use MIKE for inexperienced users. Based on these reasons, we choose CommonKADS as the proposed mechanism for knowledge engineering and management.

3. Design methodology of the improved SIM plan

This study adopts the concepts of software quality engineering [22] to integrate CommonKADS and the SIM plan for the purpose of correctly guiding the medical system development. The high-level architecture of the proposed SIM plan is carefully designed, and the relationships between the CommonKADS and the modified SIM plan are explained as follows.

The CommonKADS Methodology contains six expertise models: organization model, task model, agent model, knowledge model, communication model, and design model.

Download English Version:

https://daneshyari.com/en/article/454299

Download Persian Version:

https://daneshyari.com/article/454299

Daneshyari.com