International Journal of Engineering & Technology, 7 (3.20) (2018) 235-245

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Proper Software Engineering Process in Developing an Integrated Telehealth System

Mohd Khanapi Abd Ghani[1], Mustafa Musa Jaber[2,3], Salama A Mostafa[4], Aida Mustapha[4], Mazin Abed Mohamed[1,5], Hannani Aman[4]

[1] *Biomedical Computing and Engineering Technologies Applied Research Group, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia*
[2] *Faculty of Civil and Environmental Engineering, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia*
[3] *Nabu Research Academy, Baghdad, Iraq*
[4] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia*
[5] *Planning and Follow Up Department, University Headquarter, University of Anbar, Anbar, Iraq*
**Corresponding Author Email: khanapi@utem.edu.my*

Abstract

Software development project becomes difficult because of the complexity in the business requirements, rigid framework and unpredictable performance. These cause difficulties to deliver the software on time, to maintain it and to adapt to new requirements. These scenarios had been faced in developing the healthcare information system which typically complex and comprehensive. Due to its complexities, the development of healthcare information system and other related healthcare applications typically disparate and less integrated. This led to maintenance and system integration issues that recently struggling by most healthcare providers around the world to resolve it by putting effort to link all existing disparate system into integrated one. This was due to the fact that a proper software engineering process was given less attention at the beginning of the health ICT project. This paper proposed, a systematic software engineering process based on customized Rational Unified Process for managing the development of integrated telehealth system. This paper also proposed literate modeling approach for performing system analysis and design model for users' accessibility and comprehensibility.

Keywords: *Malaysian Public Healthcare Facility, ICT initiatives, Issues and Challenges, Lifetime Health Record*

1. Introduction

Managing software development projects is more difficult in a complex business environment, rigid design approach and improper software development process. This results in late delivery, complicated systems maintenance and inability to adapt to new requirements. The healthcare sector is one of the complex environments and there are many activities that need to be managed and countless decisions to be made [1]. That is why healthcare information system (HIS) is built and a proper software engineering process is required for managing the development process of the HIS applications (such as hospital information management system (HIMS), telemedicine, telehealth and e-health). The development of HIS applications normally disparately and relatively poor integrated. This led to system integration issues that recently struggling by most countries to resolve it by putting effort to link all existing disparate system into integrated one. This was due to the fact that the development of the system at the beginning of the software development project was not given attention on the proper software engineering process and the use of appropriate modelling approach.

This paper proposes a proper software engineering process and visual modelling tool based on customized RUP and literate modelling approach. One of the significant benefits of these approaches was that the complexity of telehealth system can be addressed by developing the system incrementally and the system maintenance could be addressed accordingly. On the other hand, the literate modelling could improve the communication between technical team and domain team (users such as doctors, nurses and medical assistant) during system modelling that help the users ease to understand the proposed model of system requirements and design – which normally presented in technical language such as UML notation. Before describing the proposed approach, this paper presents some of the problems that are related to software development project in healthcare sector. Then, the HIS and its problems are discussed. Thereafter, the proposed software engineering process and modelling approach will be described in detail. Finally, the paper ends with the summary and conclusion of the research work.

2. Health Information System: the Problem and its Context

2.1. Types of Healthcare Application

Health information system (HIS) was built to mitigate the complexity and the increases of business functions of health system. The aim of the HIS was to contribute to a high-quality and efficient patient care – including medical and nursing care and, the administrative and management tasks needed to support such care [2]. The common healthcare applications include

hospital information and management system (clinical information system, patient management system, radiology information system, pharmacy information system, laboratory information system, admit-discharge-transfer information system, billing and human resources information system), telemedicine system, telehealth system and e-health system.

Hospital information and management system (HIMS) provide supports to the healthcare providers in managing its day-to-day business functions including administrative and clinical functions. The HIMS is the central information system in the hospital and it integrates all subsystems such as clinical information system, patient management system, billing and functions unit in order to provide healthcare services to the patient efficiently [3].

Telemedicine and telehealth system are new paradigm of health information system in which the systems capable to provide healthcare services and support to the patient at distance regardless of time and locations [4]. Telemedicine and telehealth system were developed from the emergence of medical knowledge and ICT (internet). Medical knowledge provides health information and ICT act as enabler to access, display and transport the information across the healthcare providers and share the patient medical history among healthcare professionals and researchers [5].

E-health system combines and links all type of health information system in a single network and the stakeholders could access and share the health information seamless and borderless. The e-health system aims to implement paper-less concept where the administrative and clinical functions would be processed electronically [6]. Looking back at the health information system literature described above, it has been noted that the health information system is complex and the development of this system should be planned and executed systematically. A systematic and proper architecture framework should be exercised and a proper software engineering process is crucial to be practiced in developing the health information system (telehealth system).

2.2. Software Development Issues in Healthcare Domain

Healthcare services delivery could be described as an information intensive industry. In contrast the banking industry is transaction intensive. Both are able to benefit greatly from the use of computing, information and communications technology [7]. However, the healthcare industry is still one step behind in this regard. Healthcare system provides plenty of services according to disciplines or specialties. Different discipline will be managed by different department and manager. For example, a hospital could consist of outpatient department, orthopedic department, pediatric department, obstetrics and gynecology and etc. The different specialty and management of disciplines could lead to difficulties in unifying the user requirements. Hence, the acceptance of the requirements would be delayed and the requirements might keep changes if a proper software engineering process does not implement during the development of the healthcare information system.

In fact, the problem becomes worst when the healthcare information system is plans to be implemented nationwide. With regards to Malaysian context of healthcare facilities setup, there were four levels of healthcare system nationwide. The healthcare system supported by different level of health network that broken down to primary care (health centre and polyclinic), secondary care (district hospital), tertiary care (state general hospital) and national tertiary care (reference centre) [8]. The multilevel organization and administration of healthcare facilities across country would be the big challenges in handling the user requirements and designing the integrated telehealth system solution. In addition to the above issues, the healthcare industry data takes many form including numbers, text, images, coded data, signals and sound [1]. In order to manage these varieties of functionalities and data, it requires a proper modeling and significant policies to protect patient medical records by unauthorized people. Again, a proper and a flexible approach or methodology should be implemented and practiced in the design process of healthcare information system in general and integrated telehealth system in particular.

3. Case Study: the Development of Malaysian Integrated Telehealth System

This project in our case study - Malaysian integrated telehealth system was conceived to fulfill the vision of the Government of Malaysia and specifically, the Ministry of Health Malaysia as regards strengthening and supporting the delivery of the envisaged future healthcare system for the country whilst at the same time optimizing use of the present one. By definition, telehealth system refers to the provision of healthcare and health-related services using telecommunications, information and multimedia technologies to link participants in the healthcare system [9]. It is in fact one of the strongest driving forces of the telehealth initiative that it offers vast opportunities for linkage of healthcare consumers, healthcare providers, levels of care, facilities and services in a continuous, integrated and seamless system that has not only the participation but also the interests of all stakeholders [8]. Telehealth will enable a more person-focused, integrated and productive healthcare system. The telehealth initiative will fully exploit and harness ICT in order to enable the transition of the healthcare system from one that is relatively fragmented into an integrated one.

Originally, the Malaysian integrated telehealth system consists of four components [8], namely:

- Mass Customized Personalized Health Information and Education (MCPHIE)
- Continuing Medical Education (CME)
- Teleconsultation (TC)
- Lifetime Health Plan (LHP)

Although, the recent structure of telehealth system was reorganized, the missions, goals and functionalities of the system are still based on the original components described above. Figure 1 illustrates the interplay amongst components within the telehealth applications; the LHP (clinical support system - CSS, lifetime health record - LHR, personalized lifetime health plan - PLHP and group data services - GDS) system forms the backbone for integration into the other three applications (CME, MCPHIE and TS) to form an integrated solution.

Figure 1: The applications within the integrated telehealth project [9]

The development of the four components of telehealth involved different domains in the project. These domains were content, application and infrastructure. All these domains have their own structure and processes before they will integrate into integrated workable application. The aim for the integrated, scalability and maintainability require flexible management tool to manage the software development process of the telehealth system. In order to draw the system development process involved in the development of the telehealth project, the development activities involved in the three domains will be described in the next section. Perhaps, this will give a clear picture on what processes involved in the entire project and provide evidence that the project was complex and crucial to have best practices software engineering process.

3.1. Applications Development Scenario

The software development for telehealth project involved four main components – LHP, CME, MCPHIE and TS. Every component has their own organization structure and lead by the software development manager. The development of the modules and subsystems of the four telehealth components progresses concurrently. Each component has to fulfil their user requirements, system architectural design, perform system testing and acquire user acceptance independently. The completed modules and subsystem for all four components will be integrated incrementally for accessing and sharing the telehealth services seamlessly across healthcare service levels. All of the above applications used store-and-forward and internet technologies as a telecommunication medium for delivering the telehealth services. The virtual private network (VPN) is used to link all referral hospitals and health centers. The spectrum VPN bandwidth requirements ranged from 2 mbps from the hospital to the data centre to 64 kbps from the health clinics to the hospital and 100 mbps for the local area network (LAN) of the hospital [5].

The medical records are hosted in two areas: one centralized database residing at the telehealth data centre and another that is linked to the many distributed databases at the various hospitals. All distributed databases contain detailed medical records that are captured online and in real time via web-based clinical information system (CIS) application at every individual health centre. These distributed databases are then linked up to the centralized data centre whereby any information residing at the various distributed data centers may be retrieved by an attending healthcare provider or the patient as required. Going back to the basics of the implementation approach to integrated telehealth and the objectives of quality health care, affordability and accessibility, the telehealth project opted for an n-tier, open system architecture which provides scalability, robustness, access anytime and anywhere and interface with existing legacy systems. The aim was to make the project affordable with low initial capital investment. Unfortunately, the downside of this choice was in performance issues and thus a compromise had to be negotiated, whereby a modified n-tier system was designed and installed to suit performance requirements [10]. This modified n-tier system addresses the issues of cost (and therefore affordability), scalability and generally, supports the continuum of care premise of the integrated telehealth project. It has to be noted that the modification (software maintenance) of telehealth module and subsystem for accommodating any requirements has not been easily done/modified if a proper software development process does not practiced in the project. The software artifact such as design document and software version are crucial components need to be maintained and controlled effectively for avoiding software defects and mistakes. The Malaysian integrated telehealth system was developed based on Rational Unified Process (RUP) with the support of the Unified Modelling language (UML) tool for analysis and design modelling.

However, this approach has little understanding from the project team especially the users who purely have different background and knowledge in system development process and the design model using UML. The word "based on RUP" methodology was too ambitious and generic because RUP provide complete, comprehensive and generic processes to develop and manage software development project [11]. The project team should choose a direction and customized the relevant processes and deliverables of RUP that appropriate to the telehealth project. The people, social-culture, management style and project nature are crucial factors to be considered before adopting the appropriate methodology in the software project.

3.2. Contents Development Scenario

In the integrated telehealth project, the content domain development team has had to develop a vast scope of functionalities in order to serve various stakeholders in all the applications. These stakeholders comprise various categories of healthcare professionals providing medical and healthcare, patients, educators in the medical disciplines and health governance managers as well as other industrial stakeholders of the health service industry. At the beginning of the telehealth project, even the word "content" was alien and awkward to define. Despite not having a clearly defined concept of the word, our partners in the project proceeded to establish 'content working groups' consisting mainly of eager but inexperienced (in e-learning philosophies, concepts and skill sets) writers. A series of 'content creation' workshops to come up with innovative content to serve both the care providers as well as the patients were put into motion. However, the lack of standardized approach and lack of clarity regarding the shape of contents to be delivered, as well as some measure of imprecise requirements, hampered the content output. As with all ICT ventures, where the requirement specifications are crucial to the end product design, the information design of content became the debate of all involved. Designs and concepts in various innovative forms mushroomed in all the various groups and each was unaware of the others' progress and evolution until two years into the development phase when the project was finally initiated officially [12]. During these developing years, there was a dichotomy in the development of content (the content working groups plodding in one direction and the steering group in the other) and as with all great stories, the twain never met until then. The sequence of events after the two groups met could make a great public relations exercise in its own right, when great minds have to come to a definitive agreement.

What was content development in the telehealth project all about? [12] further explained the aims of the content development of the telehealth project. The primary aims of contents in the different projects of integrated telehealth project are as follows:

MCPHIE contents are aimed at giving customized and personalized health information and education to MCPHIE web-users to empower them to maintain the highest status of health possible. These contents are customized to their lifestyle and risk categories based on the users' own data input, thereby leaving the onus of accuracy to the users. These services are available throughout the nation and thus make telehealth equitable and accessible to all. PLHP contents are selected by healthcare providers from customized MCPHIE contents based on age, gender and diagnosis. On top of that, further personalization of information relevant to the personal disposition and circumstance of the patient, are also added on, as deemed appropriate by the healthcare provider.

JIT (just-in-time) CME is aimed at giving valuable relevant information, based on clinical practice guidelines (CPGs) and

other scientifically proven evidence, to healthcare providers at the point of care. The JIT CME is expected to enhance learning, since it is executed mostly at the workplace, during the doctor-patient encounter, and because it is a specific response to a specific situation for a highly specified patient at that very specific point in time of the care process, is a very effective learning tool indeed. CME has a vast bank of content to serve both formal and non-formal continuing medical education catering to all levels of health professionals throughout the nation. The primary requirements of access, management and delivery of content therefore, are that they have to be integrated, timely and efficiently distributed.

3.3. The Lessons from the Case Study

Looking back at the two scenario of telehealth development, it was summarized as following: First, the integrated telehealth system is a kind of project that designed tailored to specific requirement of Malaysian need and perspective. The custom development is the most suitable development strategy to develop the integrated telehealth system. However, a systematic processes, approach and tools have to implement at the early stage of the system development. Second, the telehealth project involved from various teams, stakeholders and users. A proper management tool is critical to be implemented for managing the system development process of the project. Third, the telehealth system requirements are vast and interdependent by the project processes and applications, which needed these requirements to be managed effectively (might be frozen for the next process to be developed so that the end product could be delivered on time). It was noted that, the telehealth project would be evolved incrementally and developed gradually towards the end of the final product. A proper software engineering process is crucial to be implemented and practiced in such software project like telehealth system. A systematic processes and flow of managing the software development process of telehealth system would produce quality, reliable and scalable system that would impact in improving the healthcare services to the consumers.

4. System Development Approach Analysis

4.1. Visual Modeling vs. Structured Modeling

Unified Modeling Language (UML) and other visual models capture important information about the business and business systems in a brief and elegant manner [13]. However, these information is understand and accessible only to software development team or those who understand the visual syntax and semantics of the visual modelling language. For non-technical users (such as healthcare professionals and health workers), they faced significant difficulty to understand and appreciate the model. In most businesses, the people who know UML are in the minority and so, in a way, the UML model encrypts important business information so that it is accessible only to those few who are “in the know” [14]. For that reason, the project team especially at the domain site have had issues in understanding the notation of the model used in UML. Among the issues are.

- **Accessibility**

Accessibility is the capability to access the information contained in a UML model [14]. The users may also need to know how to work with the modelling tool for accessing the information embedded in the model. The cut and paste of diagram from the modelling tool into the reports does not a practical way to maintain the design artefacts. These make the model of limited practical use.

- **Comprehensibility**

Comprehensibility is the ability to understand the semantics of the model [14]. Due to lack of accessibility to the UML model, the users have little understood and difficulty to obtain the business value from the models. In facts, they may find it difficult where to start either when reading the report or reading the model in a modelling tool. Hence, the important business requirements always watch out by the users and make argument after the system design or system test.

4.2. Iterative Vs. Structured Approach

The iterative process of the RUP has given invaluable improvement in handling complex system and constantly changing of the requirements [11]. For example, the clinical information system (CIS) might change over time once the patient management system (PMS), laboratory information system (LIS), radiology information system (RIS) and pharmacy information system (PIS) were integrate one to another. Iterative approach mitigates risk in early stage of the development of CIS module by going through the identified sequence of activities multiple times and revisiting each of the key activities in a planned manner. The first iterative of CIS might include the basic interface requirements (such as retrieve patient demographic from PMS, send investigation order to LIS and RIS and prescribe drugs to PIS) to/from other support modules and ends with an executable release. As such, the problems and unseen requirements will be revealed at the first attempts of the system implementation.

On the other hand, a structured sequential application development methodology (SSADM) is still used in their software projects. This methodology has often been criticized for lack of modularity, reusability, maintainability and compositionality. SSADM has been used extensively in the past include in the development of legacy hospital information system. This approach works well for CIS standalone projects for small clinics usage. It also suitable for projects where the requirements are stable and relatively fixed, the problem domain is well understood and the solution has been proven on similar projects in the past [15]. Once, the health organization is grown with multilevel of healthcare services nationwide for providing and ensuring the continuum of care to the patients, the requirement and the business process become complex. The SSADM or waterfall approach cannot be utilized for handling constantly changing of requirements, complex system and cutting-edge of technology. These would happen at the development of the integrated and distributed telehealth system. It is impossible to develop a complete integrated system by segmenting them into sequential phases (for example, requirements, analysis, design, implementation and test). Longer time is required to implement the system and any defects or discrepancy of requirements can only be detected at the end of the phases. At that time, the requirements and the technology may absolute and the user would have other fantastic requirements that may hamper the existing design.

Although the sequential approach difficult to utilize in developing the complex system that is constantly changing in requirements, the representation form of its modelling can easily understood and access by the users. The context diagram (CD), data flow diagram (DFD), structure chart (SC), entity relationship diagram (ERD) and program specification (PS) easily understand by the users and commonly used in any legacy software project of healthcare system [12]. From the discussion above, both approaches (iterative and structured) have pros and cons and it looks that the combination of modelling notation in designing the framework can be considered. However, inconsistency used of modelling tools would increase the maintenance efforts in the software project. For example, the training effort, software license maintenance and technical support. The next section will propose

the suitable approach used in the system development of telehealth system.

5. Recommendations

The previous section discusses the strengths and the weaknesses of both, iterative and structured approaches. The business process of healthcare system is complex and requires flexible approach to manage the development of its applications [1]. After considering the capability and the flexibility of both approaches in managing the software development process, it was suggested that the development approach for the development of the integrated telehealth project is based on the customized RUP. The literate modelling will be utilized as modelling tool for comprehensibility and accessibility.

5.1. Implementing Software Engineering Practices

The proposed software engineering practice was designed based on the critical areas that have to be systemized in managing the system development project of the telehealth project.

5.1.1. Establish Software Development Process Framework

The RUP is an iterative and incremental software development process. RUP approach proposes that software is developed over a number of phases (inception, elaboration, construction and transition) and each of which consists one or more iterations [16]. The phases are executing by the activities logically according to the disciplines (business modelling, requirements, analysis and design, implementation, test, deployment). The above list of RUP phases, disciplines and activities may feel overwhelmed. In the development of the telehealth system project, not all of the artefacts have to produce in a consistent form and it can be customized according to the need and nature of the system or project. The most critical and crucial activities involved during the development of the telehealth system are business modelling, requirement modelling and analysis and design modelling. These activities require extensive of users' involvement for agreeing and accepting the proposed to be system. The system design starts by defining goals and constraints based on the research questions and stakeholder requirements. The design process then move into an identification of architectural risk areas that demand primary attention. In facts, the RUP approach enables us to focus on critical functions of the system that can be developed iteratively and incrementally.

By the way of example and in the case of telehealth applications, the first iterative of the software/system development might include fundamental and critical functions (such as 'register patient use case', 'maintain LHR use case', 'get LHR use case', 'do consultation use case' and 'maintain patient master index use case') of clinical workflow to upkeep the patient health record continuously. This first iterative would be used as prototype system and validity testing of the proposed framework. Figure 2 depicts the proposed iterative plan of proposed system development framework from requirements to implementation of the telehealth system project.

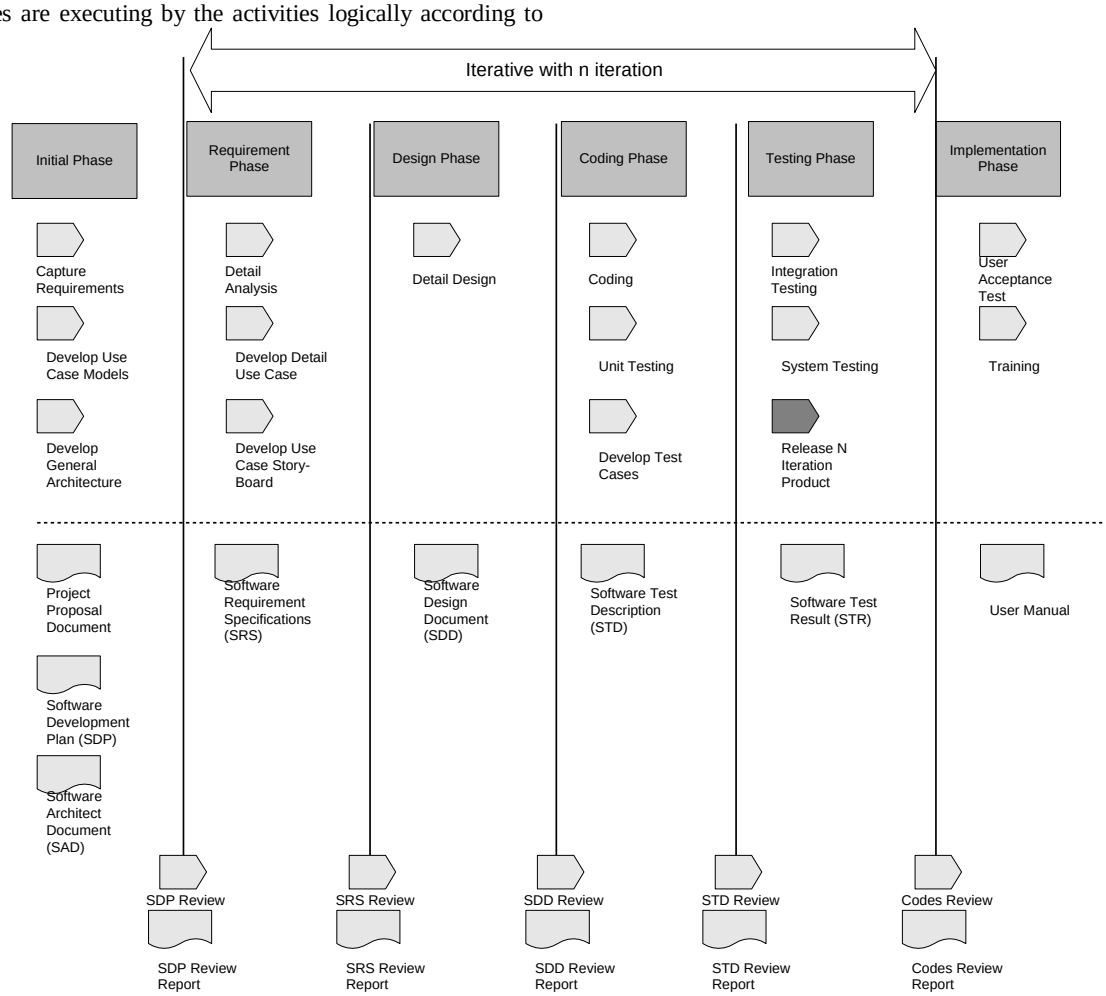

Figure 2: Proposed software development process for Malaysian integrated telehealth applications

5.1.2. Establish Software Engineering Environment

The software engineering environment identifies and describes the plans for establishing and maintaining the resources (software, firmware and hardware) necessary to perform the software engineering activities of telehealth system project. This includes the development team, tools, documents, methods and procedures used during the execution of the

software development process of telehealth project. The engineering environment was

divided into three: First, system development environment. This is where the telehealth applications requirements were analysed, designed, constructed and tested. The software designer, domains expert and users work together towards deployment of the telehalth system. The development environment was provided and equipped with standard software items (see table 1), hardware and firmware items (see table 2) and database environment setup (see table 3).

Table 1 Software items, compilers and operating system

Item	Description	Version
Java 2 Enterprise Edition (J2EE)	Java 2 platform	JDK 1.5
Microsoft Windows XP (workstation)	Operating System	V.4.0
Iplanet	Web Server/Appl. Server	Sp2 and SP3
Web Browser	Web Browser Application	IE 5.5 and above
Oracle 8i	RDBMS	8.1.7
Edit Plus	Java Development Tools	
SOLARIS SPARC II	Unix Operating System	
Microsoft Access	Client RDBMS for data reference	v.7.0
Microsoft Visual Basic Development Tool	IDE tools to design the user interface/template	Enterprise v.6.0

Table 2: Hardware and firmware items

Item	Description	Total
Database Server	- SUN Solaris/U10 450Mhz - 256MB SDRAM	1
Web Application Server	- SUN Solaris/E250 450Mhz - 256MB SDRAM	1

The second engineering environment to be setting up is stagging environment. This is where the tested application system (beta) from the development environment was transferred before the application system can be deployed for actual live in the production environment. The beta version of telehealth system will be tested by the implementation team and the representatives form key users from various healthcare facilitiey centres. The actual environment of the implementation and workflow's scenario of healthcare services were tested and validated.

Table 3 Database profile for development environment

Database profile	Naming convention
Database Instance	LHP01D
Tablespace	TS_CSS
	TS_LHR
	TS_PLHP
	TS_GDS
Table Owner	LHP_OWNER_D
	LHP_OWNER_T
	LHP_OWNER_U

Finally, the production environment of telehealth system was deployed and configured. This is where the software items, hardware and firmware items and, database items for production system were installed for accomodating the processing requirement of the actual telehealth transactions. The setup components (software, hardware, firmware and database) should be similar with the setup of the development environment and stagging environment. By standardising the software and hardware type and version, the system maintenance could be simplified and the software developer could spend more time on developing the business functions of telehealth services rather than spend time to test the system portability between development environment and production environment. Figure 3 depicts the proposed development environment for telehealth system project.

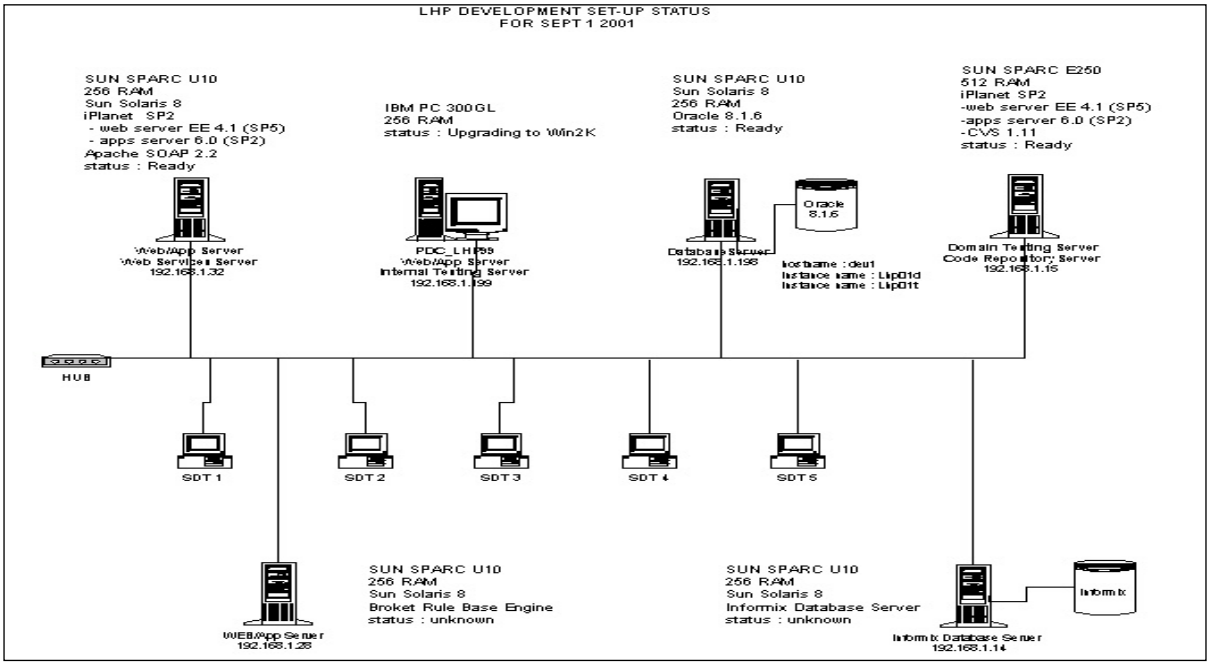

Figure 3: The development environment setup of telehealth project

5.1.3. Establish Software Configuration Management

Managing changes in software development is a universal problem and routine tasks for software developers in managing software development life cycle. Business requirement is dynamic and on the whole, change is good for improving system stability and maturity. However, developers need some control over their development environment. A development workspace is where developers put artifacts that they need to document, design, code, build, and test [17]. These artifacts such as design document, source files, libraries, and configuration files can be shared (in whole or in part) or private to each developer. In a shared workspace, any change instantly affects other developers using the workspace and the quality of the final system. Normally, it is good to be in synch among team members, but most people need a window of time to work free of change. Giving developers each a private workspace, where they can integrate changes appropriately and build the whole system if needed. The dynamic processes, activities, deliverables and changes involved in the life cycle of software development require systematic configuration management for supporting the succession of the software project. Based on my experience in managing software projects, there are three types of configuration items critical to be setting up and established. These include organizational structure for configuration management, configuration identification and configuration control.

The organisational structure describes the organisation responsible and the resources necessary for configuration management. Configuration management is achieved by controlling the configuration identification numbering of each deliverables excluding the source codes versioning. A member of the group is appointed as the configuration manager (CM) [18,19]. The configuration manager is a permanent post (ideal situation) to keep the whole implementation of the project properly. But, for telehealth project, one of the system analyst is appointed to lead the team of CM. The structure of the team is given in the Figure 4 below.

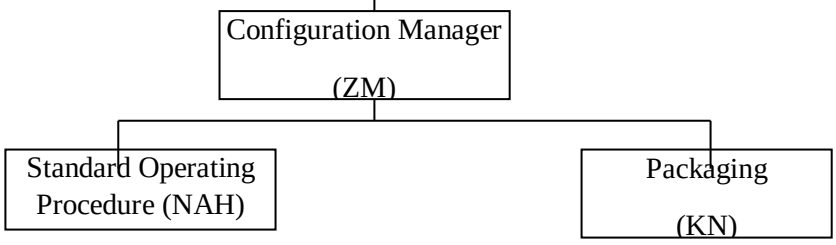

Figure 4: Organizational Structure - Configuration Management

The configuration manager lead by ZM and the team (NAH and KN) will manage the development environment using concurrent versions system (CVS) commands in DOS at client's PC and Unix commands operating system at server side. The configuration manager and the team is responsible for the followings:

- Creating the software development library (SDL)
- Managing the entire development space
- Archiving the SDL and burnt into CD/diskette once the project is completed
- Assisting other members of the team in the use of the CVS
- Facilities to manage the configuration
- Software packaging
- Requesting for software configuration control board (SCCB) meeting.

On the other hand, configuration identification is necessary to ensure all software products can be put under configuration control. All software products are specifically and consistency identified by configuration identification (CI) type, numbering, name, association, computer software configuration item (CSCI)[20,21], version and release date. By way of example, the identification of the CSCI to be developed is based on the following syntax (refer Figure 5).

System Configuration Identification: **LHP-XXXX-NN-SS**

Where

LHP	=	Project Name – Lifetime Health Plan
XXXX	=	Year number [Year number for this project is 2001]
NN	=	Sub system code [e.g. REG = 02]
SS	=	System Number [System Number for this project is 02]

Therefore, the example of complete System Number of this project is: **LHP-2001-02-02**

Figure 5: The example of system configuration identification for telehealth system

Another example of configuration identification proposed for telehealth project is the numbering scheme for the product version[22,23]. The syntax for the product version identification is as following method:
[Version].[Release].[Builds]
Where:
Version

- System version are usually identified by major version numbers, eg. 1.x, 2.x and so on.
- A Configuration Manager should ensure that, for any version of a system, there is and identifiable feature list (under configuration control) that the version implement.

Releases

- Releases to represent reliability improvements that do not affect the specification, except where clarifications are implemented.
- Releases usually fix bugs, but do not implement new requirements.
- Two releases of the same version are expected to behave the same, except for fixed for bugs fixes.
- System releases are designated by minor release numbers, e.g. 1.1.x, 1.2.x, and so on.
- A system should have a convention for which numeric level identifies a release as opposed to a version, and the convention should be strictly adhered to.

Builds

- A build is an instance of the system, integrated for the purpose of testing of a release.
- Several builds may occur before the build is deemed stable enough to be released.
- System build are designated by minor build numbers, e.g. 1.1.1, 1.1.2, and so on.

As we already known that the software development life cycle involved a series of processes and disciplines for the entire lifetime of the application software. That is why the telehealth project is critical to have a proper configuration control to manage the life cycle of its development [24,25]. The configuration control provides a detailed description of the procedures to be used in controlling changes to and maintaining the developmental configuration(s) and internally controlled documentation. A flow chart of the configuration control, which describes the abnormalities, modification, submission, examination, approval and disapproval, is presented in Figure 6.

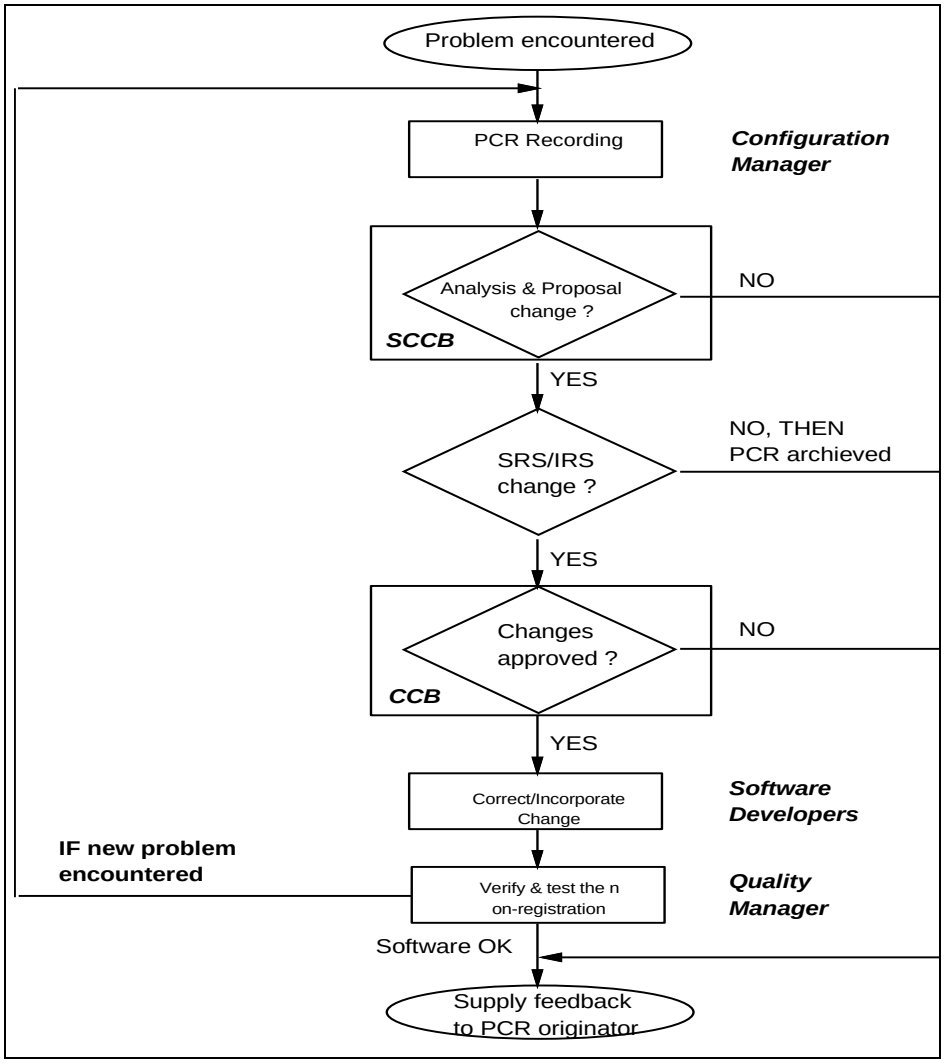

Figure 6: The configuration control workflow for telehealth project

The review procedures will be conducted by the following boards – configuration control board (CCB) and software configuration control board (SCCB). The CCB will meet periodically as per directive of the Software Development Manager. It is chaired by the Software Development Manager. The board establishes the baselines and has the power to decide on the evolution of the baselines. The CCB has the authority at the business level. On the other hand, the SCCB will meet periodically as per directive of the Team Leader or upon request by the Configuration Manager [26]. It is chaired by Project Leader. The board establishes the baseline and has the power to decide on the evolution of the configuration of the software being developed. The SCCB has the authority at the software development level. The storage, handling and delivery of project media are also taking care by the configuration personnel. All the documents identified in the Allocated Baseline are prepared using Microsoft Word application in a specific directory readable to the team. Once the documents are approved, they will be archived in a specific directory decided by the CM.

The documentation associated with the software will also delivered to the customer in hard copy. The CM will create a Referential Directory to be referred by all developers in the group. The source codes will also be delivered in hard copy. The executable modules representing the part of the system or final product will be made available to the Project Manager in a specific directory identified by the CM. At the completion of the project, the CM will also manage storage of the complete SDL tree in a compress mode (tape archived). Except for this storage version, all other items related directly to the software development must be destroyed.

It is the responsibility of the CM to ensure the configuration status accounting of all the PCRs. These include:

- Keeping an up-to-date information on the number of PCRs made, rejected, waiting or approval;
- How many of the approved requests have been solved;
- Minutes of CCB/SCCB meetings.

The CM is also responsible to update the list of documents associated with software developed by the team members. It is also the responsibility of the CM to prepare other documents requested by the Client (if any), for example the VDD (Version Description Document).

5.2. Implementing Literate Modelling Tool

Literate modeling [14] is a combination of UML notation with narrative explanation. The semantics of the model has an ability to understand by the users, non-technical person and domain expert (such as healthcare professionals and medical assistants). Based on experiential evidence done by [3] on accessibility and comprehensibility of different groups of people to various types of UML model (see Figure 7), a use case specification, use case diagram and activity diagram are the most active (comprehensibility rate between zero (worst) and seven (excellent)) UML artefact that access and comprehend by the non-technical groups (non-technical manager, user and domain expert). These artefacts provide comprehensive business knowledge

capturing and presentation that almost the time make used by the non-technical group of people. For that reason, the literate modelling is very helpful to be used in analysis and design of the proposed framework.

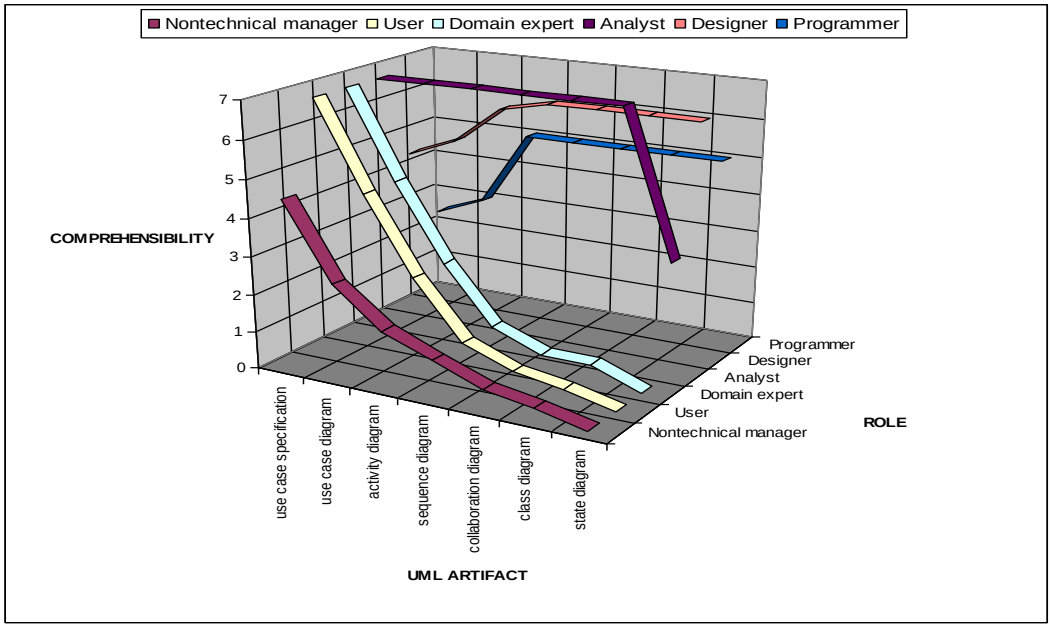

Figure 7: UML comprehensibility [13]

The used of notation not restricted to UML format but, it might include other form of representation that comprehend by all group of people. For example, the graphic diagram and flow chart are simple representation form commonly used in describing the workflow of particular business. Among diagrams that would be used in the design of the telehealth system are depict in table 4.

Table 4: Modelling diagram used in telehealth project

Diagram name	Description
Use case diagram	Is use to capture the requirements and communication tool between system and users
Activity diagram	Is use to describe the activity and flow of the business
Interaction diagram	Is use to show the dynamic interaction and process among object in one or more use cases
Package diagram	Is use to describe the high level structure of the proposed framework
Deployment diagram	Is use to depict the actual implementation of the applications, hardware and network infrastructure at user site.
Flow chart	Is use to describe the workflow of particular business services
Graphic diagram	Is use to overview the overall business process

6. Discussion and conclusions

Health sector or system is typically complex and involved human intervention in providing its services. However, with the convergence of medical knowledge and ICT, the complexity could be deduced and managed effectively through the use of computer system. In fact, the electronic medical records generated from the health information system could be beneficial to policy maker and researchers for creating health plan and policy. On the other hand, the development of health information system (integrated telehealth system) could be more complex than managing the health system if a proper software development process is not given attention during the development process of that system. The health information system is about computerizing the healthcare business use cases (business processes) that add value to its users (health workers), who are represented as actors of the system. This shows that the development of the integrated telehealth system involved various group of expertise include technology/software team, healthcare domain experts and administrators. We suggest that these different aspects of expertise, roles and responsibilities can be combined in a single framework - software engineering process framework - to facilitate the comprehensive development of integrated telehealth system and other types of health information system. In addition, the software team should able to customize the modelling technique and notation language that able to understand by non-technical persons like healthcare domain experts and administrators. By doing this, the system output would have fewer mistakes, provide functionalities that match with users requirements and deduce the system maintenance in future.

To conclude this paper, we observe that the system development of integrated telehealth is not easy. The process for integrating various solutions of telehealth components for achieving integrated system is not just another project. It is a complex processes and the project should consider deploying a systematic software engineering process for managing the complexity. It is up to the organization to choose an appropriate methodology into its software project. The most important in this aspect is that a proper software engineering practiced should be deployed (although start with simple processes) for the quality and reliability of the telehealth application.

Acknowledgments

This work is supported by Universiti Tun Hussein Onn Malaysia under the Tier 1 Grant Scheme Vot U894 and Universiti Teknikal Malaysia Melaka.

References

[1] Coiera, E., 2015. Guide to health informatics. CRC press.

[2] M. M. Jaber, M. K. A. Ghani, N. Suryana, M. A. Mohammed, and T. Abbas, “Flexible Data Warehouse Parameters: Toward Building an Integrated Architecture,” Int. J. Comput. Theory Eng., vol. 7, no. 5, pp. 349–353, 2015.

[3] M. B. Alazzam, A. S. H. Basari, A. S. Sibghatullah, M. R. Ramli, M. M. Jaber, and M. H. Naim, "Pilot study of EHRs acceptance in Jordan hospitals by UTAUT2," J. Theor. Appl. Inf. Technol., 2016.

[4] M. K. A. Ghani, and M. M. Jaber, "The Effect of Patient Privacy on Telemedicine Implementation in Developing Countries: Iraq Case Study," Res. J. Appl. Sci. Eng. Technol., vol. 11, no. 11, 2015.

[5] Mohan, J. and Yaacob, R.R.R., 2004. The Malaysian Telehealth Flagship Application: a national approach to health data protection and utilisation and consumer rights. International Journal of Medical Informatics, 73(3), pp.217-227.

[6] Holliday, I. and Tam, W.K., 2004. E-health in the East Asian tigers. International Journal of Medical Informatics, 73(11-12), pp.759-769.

[7] Hersh, W.R., 2002. Medical informatics: improving health care through information. Jama, 288(16), pp.1955-1958.

[8] Suleiman, A.B., 2001. The untapped potential of telehealth. International journal of medical informatics, 61(2-3), pp.103-112.

[9] Ministry of Health Malaysia (1997), Telemedicine Blueprint : Telemedicine Flagship Application. Copyright 1997 Government of Malaysia.

[10] Ahmed, K.Z. and Umrysh, C.E., 2002. Developing enterprise Java applications with J2EE and UML. Addison-Wesley.

[11] Pollice, G., Augustine, L. and Lowe, C., 2004. Software development for small teams: a RUP-centric approach. Addison-Wesley Professional.

[12] M. K. A. Ghani, M. M. Jaber, and N. Suryana, "Barriers Faces Telemedicine Implementation in the Developing Countries : Toward Building Iraqi Telemedicine Framework," ARPN J. Eng. Appl. Sci., vol. 10, no. 4, pp. 1562–1567, 2015.

[13] Scott, K. and Fowler, M., 2017. UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language. Addison Wesley.

[14] M. Hassan, A. Fuad, M. A. Mohammed, and M. M. Jaber, "Follow up System for Directorate of Scholarship and Cultural Relations in Iraq," in International Conference on Computer, Communication, and Control Technology, 2014, no. I4ct, pp. 182–187.

[15] Seetharaman, A., Saucier, D. and Lamperillo, G., Sony Corp and Sony Pictures Entertainment Inc, 2014. Software design and development in a service oriented environment. U.S. Patent 8,887,130.

[16] DeVilliers, D.J., 2002. Introducing the RUP into an Organization. The Rational Edge, 6.

[17] Berczuk, S., 2003. Pragmatic software configuration management. IEEE Software, 20(2), pp.15-17.

[18] Mostafa, S. A., Mustapha, A., Mohammed, M. A., Ahmad, M. S., & Mahmoud, M. A. (2018). A fuzzy logic control in adjustable autonomy of a multi-agent system for an automated elderly movement monitoring application. International journal of medical informatics, 112, 173-184.

[19] Mohammed, M.A., Ghani, M.K.A., Hamed, R.I., Mostafa, S.A., Ibrahim, D.A., Jameel, H.K. and Alallah, A.H., 2017. Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. Journal of Computational Science, 21, pp.232-240.

[20] Mostafa, S. A., Mustapha, A., Khaleefah, S. H., Ahmad, M. S., & Mohammed, M. A. (2018, February). Evaluating the Performance of Three Classification Methods in Diagnosis of Parkinson's Disease. In International Conference on Soft Computing and Data Mining (pp. 43-52). Springer, Cham.

[21] Mohammed, M.A., Ghani, M.K.A., Hamed, R.I. and Ibrahim, D.A., 2017. Review on Nasopharyngeal Carcinoma: Concepts, methods of analysis, segmentation, classification, prediction and impact: A review of the research literature. Journal of Computational Science, 21, pp.283-298.

[22] Mostafa, S. A., Darman, R., Khaleefah, S. H., Mustapha, A., Abdullah, N., & Hafit, H. (2018, June). A General Framework for Formulating Adjustable Autonomy of Multi-agent Systems by Fuzzy Logic. In KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications (pp. 23-33). Springer, Cham.

[23] [23] Mohammed, M.A., Ghani, M.K.A., Hamed, R.I. and Ibrahim, D.A., 2017. Analysis of an electronic methods for nasopharyngeal carcinoma: Prevalence, diagnosis, challenges and technologies. Journal of Computational Science, 21, pp.241-254.

[24] Ghani, M.K.A., Mohammed, M.A., Ibrahim, M.S., Mostafa, S.A. And Ibrahim, D.A., 2017. Implementing an Efficient Expert System for Services Center Management by Fuzzy Logic Controller. Journal of Theoretical & Applied Information Technology, 95(13).

[25] M.A. Mohammed et al., Neural network and multi-fractal dimension features for breast cancer classification from ultrasound images, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.01.033.

[26] M.A. Mohammed, M.K.A. Ghani, N. Arunkumar, R.I. Hamed, M.K. Abdullah, M.A. Burhanuddin, A real time computer aided object detection of nasopharyngeal carcinoma using genetic algorithm and artificial neural network based on Haar feature fear, Future Generation Computer Systems (2018), https://doi.org/10.1016/j.future.2018.07.022.