3.0 METHODOLOGY

3.1 Introduction

Methodology is defined as a set of procedures which is used in the software development process. In software engineering and project management, methodology offers a codified set of recommended practices, accompanied by training materials, formal educational programs, worksheets, and diagramming tools. The existence of the system development process is no guarantee that the system will be delivered on time, fully match with the user’s requirement, the long-term quality assurance and some others limitation. Therefore, a methodology is important in providing a systematic, disciplined, quantifiable approach to development, operation and maintenance of the system. Methodology can be defined in system development life cycle (SDLC) such as waterfall model, incremental process model, evolutionary process model and etc. SDLC is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

3.2 System Development Process

System development process is composed of many activities such as requirements analysis, specification, modeling, construction, and testing, documentation, training, and support, and maintenance.

3.2.1 Requirements Analysis

Requirement analysis helps the developer to better understand the problem that will be solving to produce a finished product which meets customer needs. By extracting the requirements of desired software product is the first task in software development process. Initial communication between users and developers are important in order to elicit requirements from user at the same time to recognize incomplete, ambiguous or contradictory requirements.

3.2.2 Specification

Specification is the task of precisely describing the software to be a written document, a set of graphical models, a formal mathematical model, a collection of usage scenarios and a prototype. In practice, most successful specifications are written to understand and fine-tune applications that were already well-developed. Specifications are most important for external interfaces that must remain stable.
3.2.3 Modeling

Modeling process is the activity that involved the creation of models to illustrate the system's requirements and the design that can be achieved for this project. The architecture of a software system is an abstract representation of that system which is concerned with making sure the software system will meet the requirements of the product, as well as ensuring that future requirements can be addressed. The architecture step also addresses interfaces between the software system and other software products, as well as the underlying hardware or the host operating system.

3.2.4 Construction and Testing

The activity in this construction phase combines code generation and the testing in order to find out the errors in the code. The testing activities include unit test, stress test, and acceptance testing. Testing is important to ensure the system is reliable, consistent, correctness, and complete.

3.2.5 Documentation

An important task is documenting the internal design of software for the purpose of future maintenance and enhancement. Documentation is the most important for external interfaces.

3.2.6 Software Training and Support

Software training and support is part of the deployment phase which needs to be taken into consideration. Basically, people are occasionally resistant to change and avoid venturing into an unfamiliar area. Therefore, it is very important to have training classes for the most enthusiastic software users, shifting the training towards the neutral users intermixed with the avid supporters, and finally incorporate the rest of the organization into adopting the new software.

3.2.7 Maintenance

Maintaining and enhancing software to cope with newly discovered problems or new requirements can take far more time and effort of development of the software. Software engineering work in maintenance is responsible on fixing bugs as well as extending systems to do new things, which in many ways can be considered new work. This is not an easy task to handle maintenance work in software development industry.

3.3 Software Process Model

Software process model is the definition and documentation of process activities to be performed, their orders, and the expected deliverables. It is essential for disciplined, repeatable,
and systematic software development life cycle. This model defines the different factors such as domain, quality assurance, and complexity with the aim to deliver ideal product to end users. The process model also includes the elaboration of process definition or process programming. Waterfall model, incremental model, prototyping model, spiral model are best known process model in software engineering.

3.4 The Chosen Methodology

Throughout the system development process, the waterfall with prototyping model is chosen to implement in this project. This process model is using the characteristics and strengths of prototyping to enhance the waterfall model.

3.4.1 Waterfall Model

Waterfall model also called classic life cycle that suggests a systematic sequential approach to software development. Its linear fashion that leads to blocking states, team members must wait for other members to complete dependent tasks.

The sequence of activities in waterfall model is following the basic sequence of system development process activities including communication, planning, modeling, construction and deployment.

Advantages of using waterfall model
i. It is useful for requirements of a problem that are well defined.
ii. The work process that is to proceed to completion in a linear sequence.
iii. Documentation driven model.

Problems/Challenges associated with the waterfall model

i. Real projects rarely follow the sequential flow that the model proposes.
ii. At the beginning of most projects there is often a great deal of uncertainty about requirements and goals, and it is therefore difficult for customers to identify these criteria on a detailed level. The model does not accommodate this natural uncertainty very well.
iii. Developing a system using the Waterfall Model can be a long, painstaking process that does not yield a working version of the system until late in the process.
iv. The Waterfall model assumes that once you are done with a phase, all issues covered in that phase are closed and cannot be reopened.

3.4.2 Prototyping Model

Prototyping model is suitable in situations where the developer may be unsure of the efficiency of an algorithm, adaptability of an OS, or the form of the human-machine interaction. It is also useful whenever customer is clueless about the details of the
A Tool to Represent 3D Object as Bounding Volume

October 20, 2009

project. This model provides a better understanding for developer and customer of what is built when requirements are fuzzy.

**Advantage of using prototyping model**

i. Prototyping iteration is planned quickly and quick design occurs.
ii. Focuses on a representation of those aspects of the software that will be visible to the customer or end-users.
iii. Prototype is deployed and then evaluated by the customer or end-users.

**Problems/Challenges associated with the prototyping model**

i. Customer sees what appears to be a working version of the software
ii. Inappropriate OS or programming language may be used simply because it’s available and known
iii. An inefficient algorithm may be implemented simply to demonstrate capability.

3.4.3 Waterfall with Prototyping Model

The reason for choosing waterfall with prototyping model:

1. System visibility

The strength of prototyping is implemented to this process model that is to make the system visible to developer and user. Developer can construct a prototype to have a better view of the system.

2. System well structured
By using this methodology, the system will be well structured as the process will proceed to completion in a linear manner.

3. System is predictable

The developer is able to predict the performance and functionality of the system being developed. The developer can construct a prototype and have an overview of the system requirement. Prediction of the system’s functionalities can be recognized before entering to the construction phase of the process model. If the section is not meeting the requirement, an immediate modification will be carried out.

4. Documentation driven model

Documentation for every activity in this process model is well prepared before proceeds to the next steps. Documentation is an important work product that helps to record the overall specification of the system development process.

5. Ease of use

This is a model that is suitable for this project because every step in the process model is well defined. Besides, all the requirements for this CV database system have been identified. Therefore, it will be easy for developer to follow the sequence of development.

3.5 Requirement Gathering Approach

Requirement gathering is an activity to gather information from the end-user by analyzing user behavior and understand their information needs towards the system that is being developed. This phase is actually a base for the whole development effort. Requirement gathering is involving a series of fact-finding techniques that help to identify the user needs. For instance, the techniques included sampling of existing documentation, forms, and databases, research and site visits, observation of the environment, questionnaires, interview, prototyping and joint requirement planning. The chosen techniques for this project are conducting a research and site visits. Literature review is based on the research on journal articles and existing systems. It is a good source of information that provides idea on how other people have solved similar problems of the project.