Chapter 3  Methodology

3.1  Introduction

Bennett et al (2002) defined that a methodology is a set of general principles that guide a practitioner to the choice of particular method suited to a specific task or project. In the domain of information system, a methodology consists of an approach to software development, a set of techniques and notations that support the approach, a life cycle model to structure the development process, and a unified set of procedures and philosophy.

3.2  System Development Methodology

As shown in Figure 3.1, a modified waterfall development model that complemented by use case driven object-oriented development techniques has been chosen by the researcher as the development methodology for the proposed Online Colour Communication Web services for Textile Industry.

Figure 3.1: Waterfall development model applied for Web services development
The project kicked-off by conducting literature review on the current practices adopted by most of the companies involving in the colour communication process of textile industry, current existing electronic colour communication applications in the industry, Web services technology, as well as the potential technologies that would be used to develop this project.

The domain knowledge and the research materials collected by the researcher at the stage of literature review serve as the input for the following stage, namely system analysis. In this stage, use case driven object-oriented technique will be used to elicit the system requirements.

During system analysis, the requirements are captured and modelled in the form of use case diagrams which in the later stage of system analysis, each use case will be further translated into several class diagrams that are able to realize the requirements captured by the use case itself.

The next stage of the development process is system design. Again, the deliverables produced by the researcher at the previous stage will be used as the input for designing and refining the system. The class diagrams modelled by the researcher based on use case driven object-oriented technique will be further refined with object attributes and methods. Database design that includes the effort of creating database tables, defining table attributes and performing normalization will also be conducted during the stage of system design.

System implementation takes place after the system design stage. It involves coding and development efforts which translate the design of the system into concrete outcome. The final stage of the development is deployment and testing. The complete developed
system will be deployed on the server. Various types of tests will be conducted in order to verify and validate the correctness of the system.

3.3 Justification for Choosing Waterfall Development Model

The justification for the researcher to choose waterfall model as the system development methodology are as follows:--:

i. It encourages proper planning before starting any phase in the development process.

ii. Misinterpretation of the system may surface early and could be rectified before the system become too complicated to make any changes or enhancement on it.

iii. It presents a very high level view of what goes along throughout the entire development cycle, and it suggests to the researcher the sequence of activities he would expect to encounter.

iv. Its simplicity makes it relatively easy to be explained to various project stakeholders.

3.4 Use Case Driven Object-oriented Development Technique

Cook and Daniels (1994) pointed out that object-oriented methodology is an approach based on the development of self-contained modules or objects that can be easily replaced, modified and reused. The development techniques applied under this approach encourage the system stakeholders to view the world as a system of corporative and collaborating objects. This is very different from the traditional development techniques where the development philosophy is based on functions and procedure calls.

The use case driven object-oriented approach provides a mechanism for mapping from real world problems to abstraction from which software can be developed effectively (Bennett et al, 2002). It provides conceptual structures that help to deal with modelling complex during its incremental evolution along the development life cycle.
3.5 Justification of Choosing Use Case Driven Object-oriented Technique

The following is the justification why the researcher has chosen use case driven object-oriented development technique to complement the development life cycle based on waterfall model:

**i. To cater the technical nature of the system developed by using object-oriented programming language**

As this project will be developed under waterfall development process model that was originated and evolved under the requirements of traditional structural development approach, it is lacking of the development nature that could cater the technical needs of the system developed under object-oriented programming language. Therefore, the object oriented techniques had been chosen by the researcher to complement the shortcomings of traditional waterfall development model.

**ii. It provides traceability for user requirements across all development phases**

Use case driven object-oriented analysis and design technique provides traceability for user requirements right from requirement gathering phase until system testing phase. Under this development technique, initially, the requirements are being elicited and modelled from use case diagrams, followed by use case realization activities that translate the use case modelling into collaboration diagrams and class diagrams. During the design stage and implementation stage, the analysis class diagrams and the associations among the classes will be transformed into programming codes. When the researcher finally comes to the system testing stage, he could easily use the information gathered during the use case modelling stage as the input to for the required validation and verification testing.
3.6 Unified Modelling Language

Unified Modelling Language is the powerful modelling language that has been widely applied in the object-oriented development techniques. According to Scott (2001), Unified Modelling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artefacts of a software intensive system. It helps the participants in a software development efforts build models that will enable them to visualize the system, specify the structure and behaviour of the system, construct the system, and document the decisions made along the way.

By using UML modelling techniques, every complex system is best approached through a small set of nearly independent views of a model as single view is not sufficient to visualize the overall aspects of a system (Booch and Eykholt, 1998).

Bahrami (1999) summarized the nine mostly used UML graphical diagrams into four categories as the following. These graphical diagrams are very useful in modeling the static structure and dynamic behaviour of the system.

1. Class diagram (Static)
2. Use case diagram
3. Behaviour diagram (dynamic)
   3.1 Interaction diagram
   3.1.1 Sequence diagram
   3.1.2 Collaboration diagram
   3.2 State chart diagram
3.3 Activity diagram
4. Implementation diagram
   4.1 Component diagram
   4.2 Deployment diagram
3.6.1 Use Case Diagram

Use case modeling is one of the UML modeling technique that use by the researcher to capture the goals of the users and the responsibility of the system to the users. In other words, use cases are scenarios for understanding system requirements. (Bahrami, 1999) Every use case is a sequence of actions that an actor performs within a system to achieve a particular goal (Scott, 2001). Use case modeling is being applied in this project to capture functional requirements of the system.

3.6.2 Sequence Diagram

UML Sequence Diagrams are an easy and intuitive way of describing the behaviour of a system by viewing the interaction between the system and its environment. Sequence Diagrams are interaction diagrams which capture the behaviour of a single use case, besides showing the pattern of interaction among objects. (Bahrami, 1999) A sequence diagram shows an interaction arranged in a time sequence. Sequence diagram is being applied in this project to capture the behavioural aspects of the proposed system.

3.6.3 Collaboration Diagram

Collaboration diagram is another type of UML interaction diagram in which describes and models the interactions and associations among classes. These interactions are modelled as exchanges of messages between classes through their associations (Albir, 1998). Bahrami (1999) pointed out a major difference between sequence diagram and collaboration diagram as in a sequence diagram: the arrows which indicate the messages sent within the given use case normally proceed in one direction across the page to reflect the sequence of message order sending over timeline; but in collaboration diagram, the sequence is indicating by numbering the messages.
3.6.4 Class Diagram

UML class diagram is the main static analysis diagram that shows the static structure of the model. A class is drawn as a rectangle with three components separated by horizontal lines. The top compartment holds the class name; the middle compartment holds the attributes of the class; while the bottom compartment holds a list of operations. (Bahrami, 1999)

3.6.5 Use Case Realization

In the early stage of system analysis, use cases are used to build an initial model based on users' requirements for a new system (Bennett et al, 2002).

Since the use cases are rather high level and concentrate on a user-centred view of the system, it needs to be translated into analysis diagrams, which serves as the primary models for describing the internal structure and behaviour of the proposed system. (Bennett et al, 2002)

For this purpose, the use cases will be translated into analysis class diagrams that will also form a basis for the later design class diagrams, from which the executable code will be developed.

The activity that used to translate use cases into class diagrams is known as use case realization, which uses collaboration diagrams to help with transition from use cases to class diagrams.

The notations used by the researcher during the process of modelling class diagrams help to analyse in greater detail the model of requirements that is initially expressed as use cases. The use of class diagram related notation include elaborating the class
definitions by using attributes and operations, and analysing the logical structure and relationship between classes by using associations.

3.7 Research Techniques

Various research techniques were being used in this research in order to gather necessary information which serves as the building blocks for subsequent analysis, design and development efforts:

These techniques are as follows:

i. Study of existing systems

A few existing systems were analyzed in order to gain ideas on the features to be included in the proposed project. The review done on the existing systems helped the researcher to compare and evaluate their respective strengths and weaknesses with the proposed architecture.

ii. Review of technologies

The review on technologies such as Web services, Apache AXIS framework, Java programming language, Apache Tomcat Web server and so on gave the researcher better understanding on the strengths and constraints of the technologies to be used to develop the proposed system. It helps the researcher in determining the requirements that could be included into the system based on the technologies supported.

iii. Discussion

Useful advice and ideas were given by the supervisor throughout the entire project life cycle. The exchange of ideas between the supervisor and the researcher enabled the researcher to overcome several bottlenecks encountered during the development process.
iv. Library Research and Internet Research

Library provides a lot of reading materials that are useful for the research of this project. Books, journals, dissertations in the main library of the University of Malaya provide valuable information for analysing, designing, developing and evaluating the proposed system.

Besides, there are many informative articles related to the business domain of the proposed project available in the Internet. White papers from various vendors and technology service providers are often free for download. The use of Internet search engines such as Google and Yahoo often equipped the researcher with vital information which could not be found in printed materials.

3.8 Summary

Waterfall development model had been chosen as the process model in this project because it encourages proper planning before starting any phase in the development process. However, waterfall model which evolved under the requirements of traditional structural development approach is lacking of the development nature that could cater the technical needs of the system developed by using object-oriented programming language. Therefore, use case driven object-oriented techniques had been used by the researcher to complement the shortcomings of traditional waterfall development model. By using various types of UML diagrams, researcher will be able to capture and model the requirements as well as translate the solutions from one form to another in the nature of object-oriented approach.