**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

This chapter summarizes the evaluation of the literature relevant to the Online Complain System. It examines, compares and relates different concepts, approaches, methods and techniques relevant to the project. This involves assessment of the work done so far on similar systems. Some relevant existing technologies relating to the development of Complaint Management System are discussed.

**2.2 Review of Related Methodologies**

A software development methodology is a collection of procedures, techniques, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system (Simaanya, 2014). It is the framework that is used to structure, plan, and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. According to Manjunatha, (2015), these development models or methodologies are the various processes that are selected for the development of a software development project, depending on the project’s aims and goals. The models specify the various stages of the process and the order in which they are carried out.

There are many development life cycle models or methodologies that have been developed and used in order to achieve deferent required objectives. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.

Popular models used for software system development includ**e waterfall** model, **incremental** model, **iterative** model and **V-Shaped** model. In waterfall model, each development phase must be completed fully before the next phase can begin – phases do not overlap in waterfall model. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. This model is simple and easy to understand and use. It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process. Phases are processed and completed one at a time. Phases do not overlap. Waterfall model works well for smaller projects where requirements are very well understood. However, this model is not good for complex and object-oriented projects. It is also a poor model for ongoing projects.

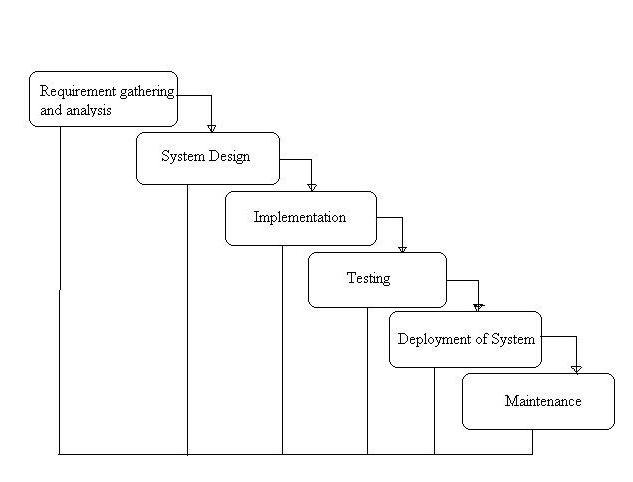


Figure 2.1: Water fall Model

V-Shaped model means Verification and Validation model. Just like the [waterfall model](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/), the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins.  Testing of the product is planned in parallel with a corresponding phase of development in **V-Shaped model**. It is simple, easy to use and works well for small projects where requirements are easily understood. Testing activities like planning, [test designing](http://istqbexamcertification.com/what-is-test-design-or-how-to-specify-test-cases/) happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model. Nonetheless, the V-Shaped model is very rigid and less flexible – If any changes happen in midway, then the test documents along with requirement documents has to be updated.



Figure 2.2: V-Shaped Model

Iterative [life cycle model](http://istqbexamcertification.com/what-are-the-software-development-models/) does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model. Iterative model hence, involves a step by step, repetitive process of software development. It is a continuous model that seeks to improve the software after every stage of iteration, thus it allows the defects to be tracked at early stages. This avoids the downward flow of the defects. Costly system architecture or design issues may, however, arise because not all requirements are gathered up front for the entire lifecycle when using this type of model

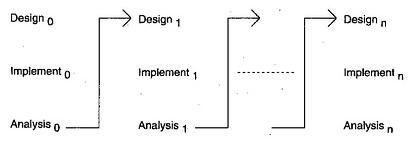
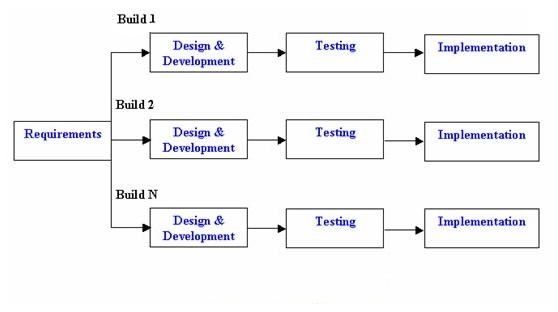


Figure 2.3: Iterative Mode

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a [“multi-waterfall” cycle](http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/).  Cycles are divided up into smaller, more easily managed modules.  Each module passes through the requirements, design, implementation and [testing](http://istqbexamcertification.com/what-is-a-software-testing/) phases. A working version of software is produced during the first module, so you have working software early on during the [software life cycle](http://istqbexamcertification.com/what-are-the-software-development-life-cycle-phases/). Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

Figure 2.4: Incremental Model

**2.3 Review of Related Works**

Osman and his co-worker proposed Online Complaint Management System, Online Complaint Management System provides an online way of solving the problems faced by the public by saving time and eradicate corruption. The objective of the complaints management system is to make complaints easier to coordinate, monitor, track and resolve, and to provide company with an effective tool to identify and target problem areas, monitor complaints handling performance and make business improvements.

Esraa and her co-workers proposed A Model for Customer Complaint Management System using SOA. They describe the Complaint Management System oriented by Web-application which will be used by Citizens in order to make complaints about their dissatisfaction on provided services. Their system will be able to handle complaints by recording and giving feedback for each raised complaint. Results of the study can be a good reference to find out users needs from e- complaint and the handling process of this complaint in the body of any organization.

Rokhmawati and Pradana proposed an E-complaint at GraPARI Telkomsel Malang, Indonesia. They are interested in making an e-complaint to be easier in handling customer complaints. Acceptance of customer complaints through My GraPARI machine or through the call center is considered not meet the needs, given the limited number of My GraPARI machines available only one unit in the city of Malang. As for the handling of complaints that require further handling, call

center is considered not able to handle maximally because every process of handling must be informed directly to the customer. They gave a solution of information system development as one form of application of the concept of

Customer Relationship Management (CRM) Operational intended to receive complaints from customers. The method used in system development is using waterfall model which includes several stages of development, including: needs analysis, design, implementation and testing. System testing is done with several types of testing, including: black-box, compatibility, web service testing and system evaluation using the utility system approach. From the overall test results can be concluded that the features on the system have been running with a valid and can run on several types of browsers that have been tested.

**2.4 Review of Development Tools and Technologies**

The following development tools and programming languages were used in the design of web-based software applications, and are considered relevant in the development of web-based applications similar to the one developed in the course of this project.

**Front-End Technologies:**

Front end is a term used to characterize program interfaces and services relative to the initial user of these interfaces and services. It usually refers to the client side of an application. A front end application is one that users interact with directly. It its everything involved with what the user sees, including design and some languages like the ones briefly described below:

**HTML (*HyperText Markup Language*)**

HTML is the standard markup language for creating web pages and web applications. HTML documents are readable to internet browsers and can be accessed by multiple users on the internet.

**CSS (*Cascading Style Sheets*)**

CSS is a style sheet language used to describe presentation and layout of HTML tags. CSS is used to enable separation of document contents from document presentation. This refers to the separation of document presentation aspects such as colors, layouts and fonts from the actual document content. CSS helps us achieve layout design and control much easier (Simaanya, 2014).

**A new version of CSS, CSS3 is used in the design of this system.**

**Bootstrap**

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Bootstrap concerns itself with front-end development only.

**JavaScript**

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities – It is complimentary to and integrated with Java.

**JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.**

**JQuery**

JQuery is a fast, small, and feature-rich JavaScript library with a combination of versatility and extensibility. It makes things like HTML document traversal and manipulation, event handling and animation, much simpler with an easy-to-use API that works across a multitude ofbrowser (JQuery, n.d.).

**Back-End Technologies:**

**PHP (*Hypertext Preprocessor*)**

PHP is an open source, server-side, HTML-embedded web-scripting language used to create web applications in combination with a web server such as Apache. PHP enables you to embed code fragments in normal HTML pages — code that is interpreted as your pages are served up to users. PHP also serves as a “glue” language, making it easy to connect your web pages to server-side databases (Suehring, Converse & Park, 2009).

**PHP is free, it is open source, it is full featured, it is cross-platform, it is stable, it is fast, it is clearly designed, and most importantly for this project, it is easy to learn.**

**MySQL (*SQL* – *Structured Query Language*)**

MySQL is a database management system (DBMS) for relational databases (therefore, MySQL is a Relational DBMS). A database, in the simplest terms, is a collection of interrelated data, be it text, numbers, or binary files that are stored and kept organized by the DBMS (Ullman, 2008).It is therefore, a storage place for data. The user runs an application that accesses data from the database and presents it to the user in an understandable format (Soumya, 2007).

**Sublime Text**

Sublime Text is a cross-platform source code editor with Python application programming interface (API). It supports many programming languages and markups languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses (PixelsTech, 2016).

**WampServer**

Is a Windows web development environment that allows you to create dynamic Web applications wi2th Apache, PHP, and MySQL. WampServer automatically installs everything you need to intuitively develop Web applications (Sourceforge, 2016).