

Software Engineering Lab Report I

Automated Lab Program Evaluator

Project Members

1. Ashwin Joisa : 16CO104
2. Praveen Raj : 16CO115

January 23, 2018

1. Process Model

Classic Waterfall Model

1.1. Justification

The requirements of this problem are reasonably well understood, and moreover they are unlikely to change. This project tries to bring about well defined adaptations or enhancements to the existing system of evaluating lab programs. The work flows from communication through deployment in a reasonably linear fashion.

1.2. Explanation

1.2.1 Feasibility Study

The main aim of feasibility study is to determine whether it would be financially and technically feasible to develop the product.

- At first we will try to have a rough understanding of what is required to be done by taking into account the client's requirements. We then study different input data that can be feed to the system and types of output data to be produced by the system. We study what kind of processing is needed to be done on these data and they look at the various constraints on the behavior of the system.
- After we have an overall understanding of the problem we investigate the different solutions that are possible. Then we examine each of the solutions in terms of what kind of resources required, what would be the development time for each solution.
- Based on this analysis we pick the best solution and determine whether the solution is feasible technically.

1.2.2 Requirements Analysis and Specification

The aim of the requirements analysis and specification phase is to understand the exact requirements of the customer and to document them properly. This stage consists of two distinct activities, namely:

- Requirements gathering and analysis
- Requirements specification

In this stage we will gather all relevant information from the customer regarding the product to be developed. This is done to clearly understand the customer requirements so that incompleteness and inconsistencies are removed. After all ambiguities, inconsistencies, and incompleteness have been resolved and all the requirements properly understood, we start the requirements specification activity. During this activity, the user requirements are systematically organized into a Software Requirements Specification (SRS) document. The customer requirements identified during the requirements gathering and analysis activity are organized into a SRS document. The important components of this document are functional requirements, the nonfunctional requirements, and the goals of implementation.

1.2.3 Design

The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. Two distinctly different approaches are available: the traditional design approach and the object-oriented design approach. We will be following the **Object-oriented design approach**. In this technique, various objects that occur in the problem domain and the solution domain are first identified, and the different relationships that exist among these objects are identified. The object structure is further refined to obtain the detailed design.

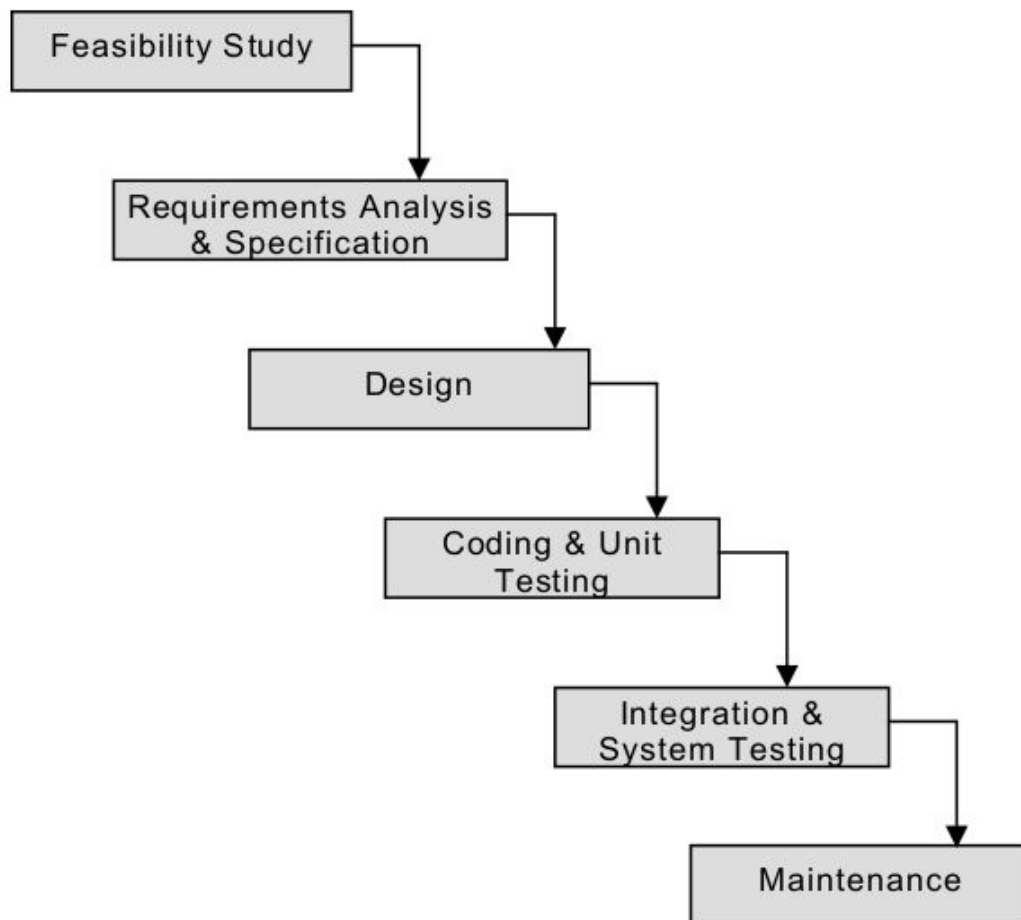


Fig: The Classic Waterfall Model

1.2.4 Coding and Unit Testing

The purpose of the coding and unit testing phase (sometimes called the implementation phase) of software development is to translate the software design into source code. We will implement each component of the design as a program module. The end-product of this phase is a set of program modules that have been individually tested. During this phase, we will unit test each module to determine the correct working of all the individual modules. It involves testing each module in isolation as this is the most efficient way to debug the errors identified at this stage.

1.2.5 Integration and System Testing

Integration of different modules will be undertaken once the coding and unit testing is done. During the integration and system testing phase, we will integrate the modules in a planned manner. During each integration step, the partially integrated system will be tested and set of previously planned modules are added to it. Finally, when all the modules have been successfully integrated and tested, system testing will be carried out. The goal of system testing is to ensure that the developed system conforms to its requirements laid out in the SRS document.

1.2.6 Maintenance

Maintenance of a typical software product requires much more than the effort necessary to develop the product itself. Maintenance involves performing any one or more of the following three kinds of activities:

- Correcting errors that were not discovered during the product development phase.
- Improving the implementation of the system, and enhancing the functionalities of the system according to the customer's requirements.
- Porting the software to work in a new environment.

2. Tools

S.No	Stage	Tool Name	About the Tool
1	Feasibility Study	LibrePlan	LibrePlan is a collaborative tool to plan, monitor and control projects. With LibrePlan we identify the tasks which make up a project and estimate the hours needed to accomplish them. You do it by creating hierarchical structures (WBS) with containers and leafs in a <i>graphical and powerful way</i> . LibrePlan helps you in this field providing a <i>Monte Carlo simulation</i> which estimates the probability density function of the project duration.
2	Requirements Analysis and Specification	rmToo	<i>rmToo</i> is a free and open source requirements management tool. The power of <i>rmToo</i> lies in the fact that the development environment can handle the input and output files – there is no need for a special tool set environment.
3	Design	LucidChart Drawing tool	LucidChart is an open source online drawing tool. This tool is going to be used for drawing Context Diagram, Data-Flow Diagrams and UML Diagrams.
		ERD Plus.	ERDPlus is an open source online drawing tool used for making the Entity-Relationship Diagrams.

4	Coding and Unit Testing	Integrated Development Environments: C++ IDE, a package in, Atom Text Editor.	Atom is a text editor that's modern, approachable, yet hackable. The C++IDE package in Atom is used to customise the editor to suit development in C++.
		Database Tools: MySQL	MySQL is an open-source, SQL based relational database management system with reliability, a very good performance and ease-of-use.
		Build Tools: GNU g++ compiler. GNU gcc compiler.	The GNU compilers used here are used for building the project source code, while using them during runtime too.
		CppUnit	CppUnit is a unit testing framework module for the C++ programming language. It allows unit-testing of C sources as well as C++ with minimal source modification.
5	Integration and System Testing	Travis CI	Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub.
6	Maintenance	GitHub	GitHub is an open source tool used for collaborative development and to efficiently handle version control tasks.

3. Summary

This report is a consolidated document of our approach to the problem of automating the process of evaluating lab programs, previously done manually. Here we have described the process model, that we have chosen for the software production process (ie. the classic waterfall model). An overview of each stage along with how we will be carrying out the required steps for each stage has also been added. The necessary tools that we will be utilising for effective implementation of our software to automate the process of evaluating lab programs have also been described.