Software Engineering Lab Report 1

Virtual Clinic

Samyak Jain - 16CO254

Mishal Shah - 16CO125

January 23, 2018

Process Model

The model chosen by us is “Waterfall model with Back-flow”. Waterfall model is a linear sequential model and given the fact that our requirements are clear, concise and unambiguous, it fits in really well with our problem description:

Justification:

Main reasons for Choosing “Waterfall model with Back-flow”

1. Waterfall model is simple, easy to use and it will help us in “understanding” (living) each phase of the Software Development Life Cycle really well i.e. It clearly divides the problem into various independent stages. In this model phases are processed and completed one at a time. Phases do not overlap apart from a back-flow to correct small mistakes, errors or omissions.
2. Our project is short - small in size and scope, Waterfall model works well for such projects.
3. Waterfall method works well when the requirements are very well understood, clear and stable. Our project is exactly that -high level requirements are very well known, clear and fixed. We have 5 stakeholders – Doctors, Patients, Chemists and Labs apart from System Administration. Our Requirements are not ambiguous. The purpose and requirements for each of user is clearly understood and thus can proceed with the waterfall model.
4. Backflow which is a slight modification of the waterfall model gives us the advantage of visiting the previous stage and correct the deliverables of prior phases/software if required.

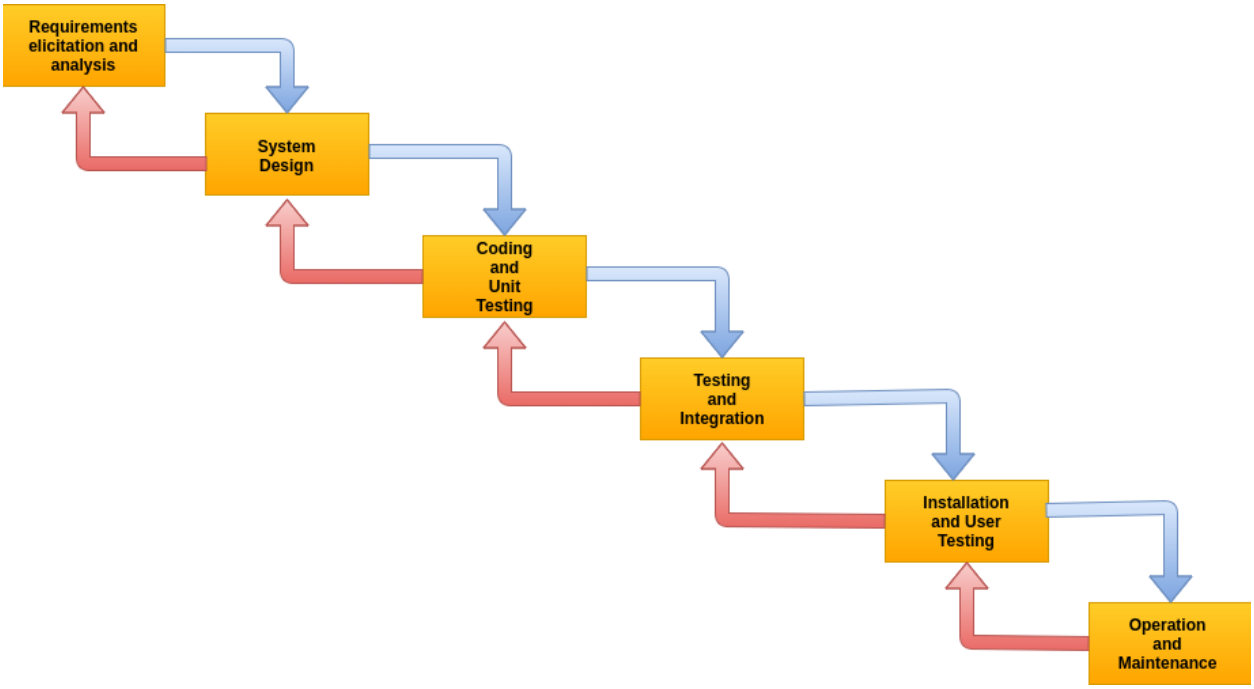

Figure 1: Software Lifecycle Development Process

Explanation:

1) Requirement elicitation and analysis

Work Done: In this stage all the requirements of the software are clearly understood from the business users. i.e. requirements are gathered from them. E.g. the roles and activities of all users – System Admin, Doctors, Patients, Chemist and Labs are understood and documented. We will understand – What should the system do? Why? Where? When? Who (Roles)? We will also try to understand non-functional requirements – number of concurrent users, response times etc. All the requirements stated (both functional and non-functional) are thoroughly analyzed .It is determined whether the stated requirements are clear, complete, consistent and unambiguous, and any apparent conflicts are resolved. We will define the Business Rules for each of the entities (ERD) i.e. Doctors, Patients, Chemists and Labs. We will also define the core assumptions and

constraints and refine the Data Flow Diagrams and produce User Interface Specifications. All of this will be documented in FRS and SRS.

Document(s) Generated: Two detailed document i.e. FRS – Functional Requirement Specifications and SRS-software requirement specifications are produced and are validated (reviewed) by the business user(s).

2) System design

Work Done: In this phase, we will design the application, network, databases, user interfaces, and system interfaces. Basically, we will transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented vide set of programs. This phase shall define the “How” part of the system. If needed, we will revisit the prior phase (Requirements Elicitation and Analysis)

Document Generated: A detailed design document (System Design Document) is generated stating clearly the design to be followed to implement the given problem as per the requirements. The document generated is verified against the SRS produced during the requirement phase. Any changes or constraints related to requirements are understood and dealt with. (feedback). We will also create a traceability matrix to ensure that Design is complete.

3) Coding and unit testing

Work Done: During Coding and Unit Testing, we will create the actual product i.e. write/code the set of programs to render business functionality as per design. We will also implement the database structure as defined in the System Design document. The output of the coding phase is the source

code for the software and the database design that acts as input to the testing and integration phase.

Further, all individual programs shall be tested for Correctness and Completeness (boundary testing, functional testing etc.). The actual product i.e. programs are constructed and verified on the basis of the requirements stated earlier and other constraints that arise during the design. It is matched with design and if problems arise then design itself is visited again and changes are made to design them it is implemented again. (feedback). Bugs found in unit testing are fixed and Unit Testing is repeated till such time that all Unit Testing bugs are fixed.

Document Generated: Unit Tested Source Code, Databases

4) Testing and Integration

Work Done: In this phase all the individual software programs shall be combined and tested as a group. E.g. Doctors defined in Doctors database should be available for Patients to Consult. The combined set of programs shall be tested together with associated control data, usage procedures, and operating procedures, to determine whether they are fit for use. This phase is also known as System Testing. If problems are found in testing and integration then again the Coding and Unit Testing phase is visited and rewrite of code needs to be done in order to make the software functional. Post that Unit Testing is performed again, before re-commencing Testing

We shall also prepare an Installation and User Manual for the system

Document Generated: Integrated and Tested Code, User Manual, Installation Manual

5) Installation and User Testing

Work Done: In this phase, we shall installing the software on user machines. For this Installation manual prepared in the prior phase is used. If required, we may need to do some post-installation configurations at user end. Software shall be tested for portability and adaptability and integration related issues are solved during installation. Finally the software is tested by (business) user for fitness for use.

If the software fails at customer site then the root causes are identified and the project would be rolled back (backflow) to the source of defects i.e. Requirements, Design or Implementation

Documents Generated: System Installed with Production Databases and Tested at Customer Site.

6) Operation and maintenance

Work Done: In this phase, the software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems. If the software fails at customer site at run-time then the root causes are identified and the software is fixed – breakdown maintenance

Documents Generated: Maintenance Reports (periodically), Also, updated/ Revised documents of previous phases as System undergoes User and/or Technology Changes

Tools

Tools required for drawing the following diagrams:

- Context diagram: Lucid chart
- Data Flow Diagram: Lucid chart
- ER diagram: ERD+

Project management tools to be used:

Achievo: is a Web-based project management and tracking tool for small- to medium-sized companies.

Sr No	SDLC stage	Tool Name	About the Tool
1.	Requirement elicitation and analysis	OSRMT- Open Source Requirement Management Tool	Requirements management tool designed to achieve full SDLC traceability for features, requirements, design, implementation and testing. UI for requirements derivation, version control, attributes etc.
		ProR	ProR is a part of the Eclipse RMF project and is built for extensibility.

2.	System Design	ArgoUML	ArgoUML is a program that allows users to create complex and professional diagrams. It can be employed for statistical purposes as well as for common business needs.
3.	Coding and unit testing	Backend: Django 2.0	Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
		Database: MYSQL	MySQL is an open-source relational database management system(RDBMS).
		Frontend: HTML, JavaScript, jQuery, AngularJS	HTML: is the standard mark-up language for creating web pages and web applications JavaScript: As a multi-paradigm language, JavaScript supports event driven functional, and imperative (including object-oriented and prototype based) programming style.

			jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML AngularJS: The AngularJS framework works by first reading the HTML page, which has additional custom tag attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables
		Styling: CSS, Bootstrap	Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language. Bootstrap is a free and open-source front-end web framework for designing websites and web applications.
		IDE: PyCharm (student version)	It provides code analysis, a graphical debugger, an integrated unit tester, integration with version-control systems (VCSes), and supports web development with Django.
		Version Control:	GIT is a version-control system for

		Git, GitHub	tracking changes in computer-files and coordinating work on those files among multiple people.
		Text Editor: - Sublime	Sublime Text is a cross-platform source code editor with a Python Application programming interface(API)
		Usage of system library for unit testing: Django- unittest JavaScript/jQuery: assert	The preferred way to write tests in Django is using the unittest module built in to the Python standard library.
4.	Testing and integration	AngularJS: karma, jasmine, protractor JS	Karma is a JavaScript command line tool that can be used to spawn a web server which loads your application's source code and executes your tests. Jasmine is a behaviour driven development framework for JavaScript that has become the most popular choice for testing AngularJS applications protractor JS: Protractor is an end-to-end test framework for Angular and AngularJS applications. Protractor runs tests against your application running on a real

			browser, interacting with it as a user would.
		Travis CI	Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted on GitHub.
5.	Server hosting/ Installation (user testing)	Heroku If bugs are found again, coding and unit testing tools will be used along with latest version of software from Git.	Heroku is a cloud Platform as a Service (PaaS) supporting several programming languages. that is used as a web application deployment model.
6.	Operation and Maintenanc e	Latest version from Git, GitHub will be used along with coding and unit testing tools.	

Summary

For developing Virtual Clinic, our software engineering project, we have chosen the “Waterfall model with backflow” as the process model. The main reason for choosing this model is that it is simple, easy to use and it clearly divides each phase of SDLC. Our project’s high level requirements are very well known, clear and fixed hence waterfall with backflow suits it quite well. We will be doing this project in 6 phases namely requirement elicitation and analysis, system design, coding and unit testing, testing and integration, installation and unit testing and operations and maintenance. Key Deliverables for each phases are SRS & FRS, design document, unit tested source code, integration and system tested code and user manual/installation manual. Finally we aim to install and test the system at

customer site and provide breakdown/ corrective maintenance as needed. We will be using various open source tools to accomplish the project like ProR, ArgoUML, Django 2.0, Travis CI in various different phases of the life cycle model. We plan to complete this project in 12 week' timeframe using the waterfall with back flow model and various open source tools available.