## Lab 3 Training Part

## Software Quality Plan and Control

## Team Members:


- **Tanya Agarwal || Id: 110060426**
- **Arshdeep Kaur || Id: 110031302**
- **Anubha Sharma || Id: 110037181**
- **Roxy Leonard Palma || Id: 110037154**
- **Avinash Gupta || Id: 110060197**
- **Harjinder Gill || Id: 110030918**
- **Shafkat Waheed || Id: 110060302**
- **Priyam Dua || Id: 110059536**

# Project Description:

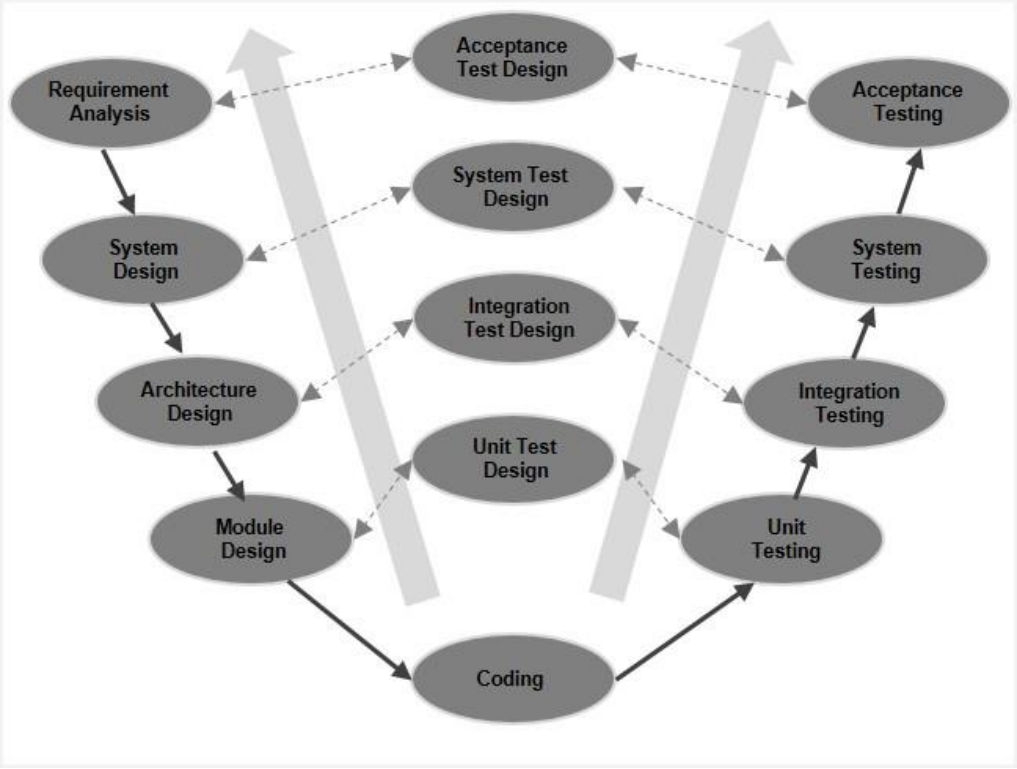
A web search engine that uses and implements various Data structures, searching and sorting algorithms to search, sort and store HTML pages containing the information about various universities and colleges in Canada i.e. variety of programs offered, detailed information about the subjects, the fees structure of the institution and many more.

## 1. Software Development Life Cycle:

V-Model also referred to as Verification and Validation Model will be used for the development of a web search engine to search, sort and store information about various universities and colleges by implementing Data structures, sorting algorithms.

### 1.1 V-Model Development Methodology

The V-Model specifies a series of linear stages that occurs across the life cycle, one at a time, until the project is completely developed. 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 the V-model. The following figure describes the different phases in a V-Model of the SDLC.



There are several verification phases in V-Model, each of these is explained below:

- Requirement Analysis: During the initial phase, system requirements and analysis are performed to determine the feature set of needs of users. As described in the V-Model during each design stage, the corresponding tests are also designed to be implemented during the testing stages. During the requirement analysis, acceptance tests are designed that will be implemented later.
- System Design: This phase is issued to generate a specification document that will outline data layers, business logic, and so on. System tests are also designed for later use.
- Architecture Design: In this phase, the specifications are drawn up that provide the detail of how the application will link all its various components referring to high-level design. Integration tests are also developed during this time.
- Module Design: In this phase specifications for low-level design is drawn, including how all functional, coded business logic will be implemented. Unit tests are also created for later use.
- Coding/Implementation: In this phase, the actual coding and implementation occur. All the previously generated designs and specifications are converted into coded functional systems.
- Unit Testing: The process moves back up the far side of the V-Model with inverse testing. This phase eliminates the vast majority of potential bugs and issues. Therefore, it is the lengthiest testing phase of the life cycle.
- Integration Testing: Test cases that were developed during the architecture design phase are executed here to ensure that the system functions across all components and integrations.
- System Testing: The test cases created during the system design phase are executed, largely focusing on performance and regression testing.
- Acceptance Testing: Acceptance test cases are implemented to ensure that the system is functional in a live environment with actual data, ready for deployment.

1.2 Why V-Model is chosen?

After researching all the development lifecycle models, the following points define why V-Model will assist in achieving our final goal:

- The project length and scope are well defined, and the documentation & design specifications are clear. Therefore, the V-Model will be suited due to its rigid nature and linear design, implementation, and testing phases.
- The V-model is ideal for time management. With fairly clear and well-understood requirements defined by the customers and stakeholders, it is easy to create a timeline for the entire development life cycle.

## 2. Roles and Responsibilities:

### 2.1 Project Manager:

The project manager will be responsible for coordinating with various team members and the resources to complete the project on time. The project manager needs to be up-to-date on different approaches to system development but not likely to do any programming themselves.

### 2.2 System Analyst:

The system analyst will be the one who analyzes the requirement of the proposed system and will ensure that requirements are conceived and documented properly. The role of system analyst starts during the Requirement Analysis phase.

### 2.3 Solution Architect:

The solution architect is responsible for providing the details such as how the application will link all its various components referring to high-level design. The role of solution architect starts during the Architecture Design phase.

### 2.4 Programmers:

A programmer is responsible for defining the low-level design and coded business logic that needs to be implemented. Design and specification are converted into coded functional systems by the programmers. The role of a programmer starts during Module Design and Implementation phase. They also perform unit testing as they have a better understanding of the code that they wrote.

### 2.5 Testers:

- Integration Testing is mostly done by testers when two modules are integrated, to test the behavior and functionality of both modules after integration.
- System Testing is conducted by the testers to evaluate the system's compliance with its specified requirement.
- Acceptance Testing is usually done by the customer or a client for the verification of the behavior of a product.

| **Life Cycle Phases** | **Roles** |
|---|---|
| Requirement Analysis | System Analyst |
| System Design | System Design Analyst |
| Architecture Design | Solution Architect |
| Module Design, Implementation, and Unit Testing | Programmer and Tester |
| Integration Testing, System Testing, and Acceptance Testing | Tester |

## 3. Tools and Technologies used:

| | |
|---|---|
| Back-end | Spring Boot, REST |
| Database | MySQL |
| Version Control | GIT |
| Languages | Java, JavaScript, and SQL |
| Deployment tool | Azure |
| Testing frameworks | Mockito, JUnit |

## 4. Java Coding Guidelines:

- Explicitly initialize all variables.
- Use StringBuffer instead of String class for frequent concatenations.
- Catch blocks should not be empty.
- Method name should begin with a small letter, use camel casing for readability.
- There should not be any unused class members.

## 5. Roles and Responsibilities:

| | |
|---|---|
| Tanya Agarwal | System Analyst |
| Anubha Sharma | Programmer and Tester |
| Priyam Dua | Tester |
| Harjinder Gill | System Design Analyst |
| Roxy Leonard Palma | System Design Analyst |
| Shafkat Waheed | Tester |
| Arshdeep Kaur | Programmer and Tester |
| Avinash Gupta | Solution Architect |

## 6. References:


- https://airbrake.io/blog/sdlc/v-model
- V-model (Software Engineering) - javatpoint
- https://study.com/