Naaman

Project Plan

# Introduction

This software development plan is essentially a project timetable. It provides the start and finish dates for each phase of the software project, as well as the procedures necessary to accomplish each phase of work. This plan consists of the information about the project organization, project practices and measurements, project deployment and project milestones and objectives.

Project Organization part describe the team members, their roles in the project and also their responsibilities. The project organization is a framework that makes it easier to plan and execute project operations. Its major goal is to establish an environment that encourages team members to engage with each other with the least amount of disruptions, overlaps, and conflict as possible. The effectiveness of the plans is measured using project practices and measurements. Cost and schedule variables are examples of project output variables, while scope, budget, schedule, and safety performance are examples of project outcome measures. The deployment part contains deployment strategies and procedures for changing or upgrading an application's operating instance. The last part of this project plan is the project milestones and objectives. This part defined the moment in the life cycle of a project that is used to track progress toward the end objective. Milestones are used in project management to signify the start or conclusion of a project, external evaluations or input, budget checks, the submission of a significant deliverable, and so on.

# Project organization

|  |  |  |
| --- | --- | --- |
| Team member | Project Role | Responsibility |
| Sobana Handi Achini Thisarangi De Silva | Completer | Pay attention to details.  Make sure that the job is completed before the deadline |
| Dehemi Vihara Dissanayake Liyanage | Resource investigator | Explore opportunities and contacts.  Negotiate for resources |
| Gaury Chethana Thanthirigama | Shaper | Drive team forward.  Determine the most effective methods for solving issues. |
| Temuulen Tsengel | Coordinator | Clarify goals, promote decision making.  Understand the importance of each team member's contribution. |
| Micheal Weisang | Leader | Provide support and encourage to the corporation.  Managing the day-to-day activities of the team |

# Project practices and measurements

Management practices

* Define roles and responsibilities of team members - Roles of team members has been well defined for productivity.
* Define meeting types - Meetings are very important and should be defined in detail (meeting participants, context, average duration etc.).
* Define documentation strategy – The application requires a documentation strategy (backlogs, lessons learned info, bug items etc.) for consistency, understandability, and more successful software creation.
* Define and track software quality measurements - Some measurement standards should be chosen to monitor code quality and growth based on the attributes of the program type.
* Keep track of issues and requirements – Requirement management define the software's scope and provide a traceable foundation for functional testing. Those issues/requirements are better kept and managed with tools that offer greater functionality than text editing.

Technical practices

* Create a Minimum Viable Product (MVP) - This will aid in the rapid and low-cost deployment of applications.
* Keep the code as simple and consistent as possible - When it comes time to update, simplicity and a consistent coding style will make things a lot easier for the development team.
* Test continuously - Continuous end-to-end testing will increase the trust in the quality of your code.
* Use a variety of resources to double-check the code - This helps developers detect issues early in the development process and allows them to learn from one another and improve their coding abilities.
* Make realistic time and budget projections - Stress is exacerbated by short deadlines and limited funds. On the other hand, too much time may encourage developers to postpone, and too large a budget may lead to waste.

Measurements

* Quality of the code - For premium software, bug-free and semantically accurate code is critical. Quantitative and qualitative metrics can be used to classify code quality requirements.
* Performance - Software security metrics assess a software program's intrinsic security and verify that no unauthorized alterations are made to the product before it is delivered to the customer.
* Usability - Because all software products are designed for end users, usability and user-friendliness are key quality indicators.

# Deployment

* Recreate - This method results in service downtime that is dependent on the application's shutdown and boot times. Advantages of this process are it will be easier to setup and application state will be renewed.
* Ramped - The ramping deployment technique is gradually deploying a version of an application by replacing instances one by one until all of them have been deployed. To enhance the deployment time, we can change the following parameters: parallelism, max surge, max unavailable

Advantages are the limited number of users have been given access to the new version, easy to keep track of the mistake rate and performance and Quick reversal.

* A/B testing - A/B testing deployments include sending a portion of people to a new feature under controlled settings. Advantages of this process are several versions can be running at the same time and can get complete control over the flow of traffic.

# Project milestones and objectives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | **Phase** | **Iteration** | **Dates** | **Primary objectives** (risks and use case scenarios) |
| ITC303 – Software Development Project 1 | Inception Phase | I-1 | 02/08 – 15/08 | Establish Vision  Establish Initial Use Case Model  Complete Preliminary Non-functional Requirement Analysis  Identify/Document Candidate Architectures  Establish Version Control |
| I-2 | 16/08 – 29/08 | Establish Risk List  Complete Full Description for Critical Core Risky Difficult (CCRD)Use Case  Implement Technical Competency Demonstrator  Create Test Plan  Establish Initial Project Plan  Deliver Life Cycle Objectives Milestone (LCOM)  Complete Inception Phase Project Assessment |
| Elaboration Phase | E-1 | 30/08 – 12/09  (Session Break) | Mitigate Highest Priority Risk(s)  Implement Highest Priority Architectural Element(s) to Support CCRD Use Case  Complete Development Testing for Highest Priority Architectural Element(s) |
| E-2 | 06/09 – 19/09 | Mitigate 2nd Highest Priority Risk(s)  Implement 2nd Highest Priority Architectural Element(s) to Support CCRD Use Case  Complete Development and Integration Testing for 2nd Highest Priority Architectural Element(s) |
| E-3 | 20/09 – 3/10 | Mitigate 3rd Highest Priority Risk(s)  Implement 3rd Highest Priority Architectural Element(s) to Support CCRD Use Case  Complete Development and Integration Testing for 3rd Highest Priority Architectural Element(s)  Deploy Executable Architecture in Trial Environment  Complete Internal User Acceptance Testing for CCRD Use Case in Trial Environment |
| E-4 | 4/10 – 17/10 | Contingency  Deliver Life Cycle Architecture Milestone (LCAM)  Complete Elaboration Phase Project Assessment |
| Mid-year Semester Break | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mid-year Semester Break | | | | |
| ITC309 – Software Development Project 2 | Construction Phase | C-1 | 15/11 – 21/11 | Implement 2nd Highest Priority Use Case(s)  Complete Development and Integration Testing for 2nd Highest Priority Use Case(s)  Complete Internal User Acceptance Testing for 2nd Highest Priority Use Case(s) |
| C-2 | 22/11 – 6/12 | Implement 3rd Highest Priority Use Case(s)  Complete Development and Integration Testing for 3rd Highest Priority Use Case(s)  Complete Internal User Acceptance Testing for 3rd Highest Priority Use Case(s) |
| C-3 | 6/12 – 19/12 | Implement 4th Highest Priority Use Case(s)  Complete Development and Integration Testing for 4th Highest Priority Use Case(s)  Complete Internal User Acceptance Testing for 4th Highest Priority Use Case(s) |
| C-4 | 20/12 – 02/01  (Session Break) | Contingency  Deliver Initial Operation Capability Milestone (IOCM)  Complete Construction Phase Project Assessment |
| Transition Phase | T-1 | 3/01 – 16/01 | Deploy Application in Trial Environment  Complete 1st Round External User Acceptance Testing  Resolve Any Identified Issues |
| T-2 | 17/01 – 30/01 | Complete 2nd Round External User Acceptance Testing  Resolve Any Identified Issues |
| T-3 | 31/01 – 13/02 | Contingency  Deliver Product Release Milestone (PRM)  Complete Final Project Assessment |