**CS631-01 Advanced Software Engineering** – **Assignment 2**

Name**: KAVERI HARI CHARAN YADAV**

S0 ID**:** **S01996273**

**Project:** **Software Project Management Plan (SPMP) for Chocoholics Anonymous**

**1 Overview**

This document serves as the project plan for the Chocoholics Anonymous software development. Chocoholics Anonymous (ChocAn) is an organization dedicated to helping people addicted to chocolate and provide treatments with health care professionals.

**1.1 Project Summary**

**1.1.1 Purpose, Scope, and Objectives**

**Purpose: -** To develop data processing for chocAn organization.

**Scope: -** Only authenticated users and providers can access the services from the ChocAn.

**Objective: -** To receive health care services from ChocAn members need to pay a monthly fee to ChocAn.

**1.1.2 Assumptions and Constraints**

* The product must be delivered on time.
* Need to meet the specifications of the given product.
* The product need to complete within the estimated budget.
* Need to be flexible to adapt new features when added to the existing features.

**1.1.3 Project Deliverables**

The product will be delivered with the user manual and the presentation will be given on the product usage.

**1.1.4 Schedule**

To develop this project it may take around three to four months by using COCOMO model. The time takes for each phase is given below

* Requirements workflow
* Analysis workflow
* Design workflow
* Implementation workflow
* Testing workflow

**2 Reference Materials**

Reference textbook - Object Oriented and Classical Software Engineering by Stephen R. Schach – 8th edition.

Reference Textbook – Software Project Management, A real world guide to success by Joel Henry.

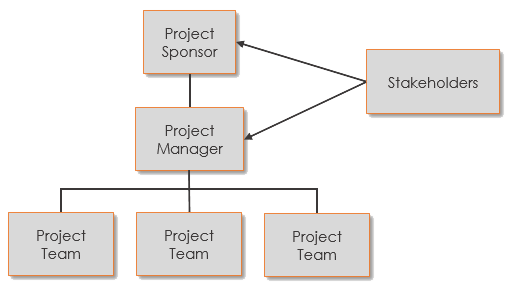
<http://www.projectengineer.net/the-project-organization-chart/>

**3 Definitions and Acronyms**

ChocAn- Chocoholics Anonymous

COCOMO Constructive Cost Model

**4 Project Organization**



The project organization contain external interface and the internal structure.

**External interface:** The project member will communicate with client and with the members of the organization to get the requirements clearly.

**Internal structure**: It defines the structure of the organization. It may contain different teams within the organization. One team do analysis, testing, implementation, deployment.

The figure represents the basic organization structure

|  |  |
| --- | --- |
| **Project organization** | **Description** |
| **Project Sponsor** | This is the person to whom the project’s [deliverables](http://www.projectengineer.net/25-example-project-deliverables/) are delivered.  They are [one level](http://www.projectengineer.net/the-project-sponsor/) above the project manager.  They are not directly involved with the project’s day to day [execution](http://www.projectengineer.net/process-groups/project-execution/), but they usually have some form of responsibility for the success and/or failure of the project. |
| Project Manager | The [project manager](http://www.projectengineer.net/how-to-become-a-project-manager/) is the person that handles the day to day administration of the project and project team and is ultimately accountable for the project’s success.  Their job is to ensure the project deliverables are produced on time, on budget, on quality, and with all stakeholders are satisfied. |
| Project Team | This group carries out the project work.  They produce the project’s [deliverables](http://www.projectengineer.net/make-all-project-deliverables-count/) and perform the tasks necessary to finish the project.  Although they are normally in a position to take responsibility for their work, it is project manager who ultimately must take responsibility for the success or failure of the project. The project team consist of requirements engineer, architect, lead engineer, programmer, tester, maintenance engineer, team leader, and tool expert.   |  |  |  | | --- | --- | --- | | **Team Member** | **Roles** | **Responsibilities** | | T1 | Requirement Engineer | The main of the requirement engineer is to gather requirements from the users and do analysis, validation and managing of the requirements. | | T2 | Architect Engineer | The architect engineer will do the architectural design of the software. | | T3 | Development Lead | The development lead maintains and monitors the team work and sends the daily analysis report to the client. | | T4 | Programmer | Programmer need to understand the design of the documentation and development environment. | | T5 | Testing | The tester will execute the test cases and will identify the bugs and sends the bug report to the test lead. | | T6 | Quality Assurance | Quality assurance will set the standard for quality and ensures whether the testing team is following those standards for the producing the quality product. | |
| Stakeholders | Every project, by definition, has someone who has an interest in its outcome.  As a minimum the project sponsor could be considered a stakeholder because they are expecting the deliverables at a certain time, budget, and quality level. Stakeholders come in many forms: **Clients, customers, and end users, Investors.** |

**5 Managerial Process Plans**

**5.1 Start-Up Plan**

**5.1.1 Estimation plan**

By using COCOMO model we can provide estimation plan for the project.

Assume that the product is built in organic mode because project is small scale project and consider 1000 lines of code (KDSI). The nominal effort is calculated as

Nominal Effort=3.2\*(KDSI) ^ 1.05 person months

=3.2\*(10) ^ 1.05

=36 person months approximately

and also assuming the following cost drivers; very high software reliability, nominal database size, low product complexity, nominal programming language experience and low use of software tools.

Total effort= Nominal effect \* Cost drivers

= 36\*1.40\*1.00\*0.85\*1.00\*1.10

= 47 person-months

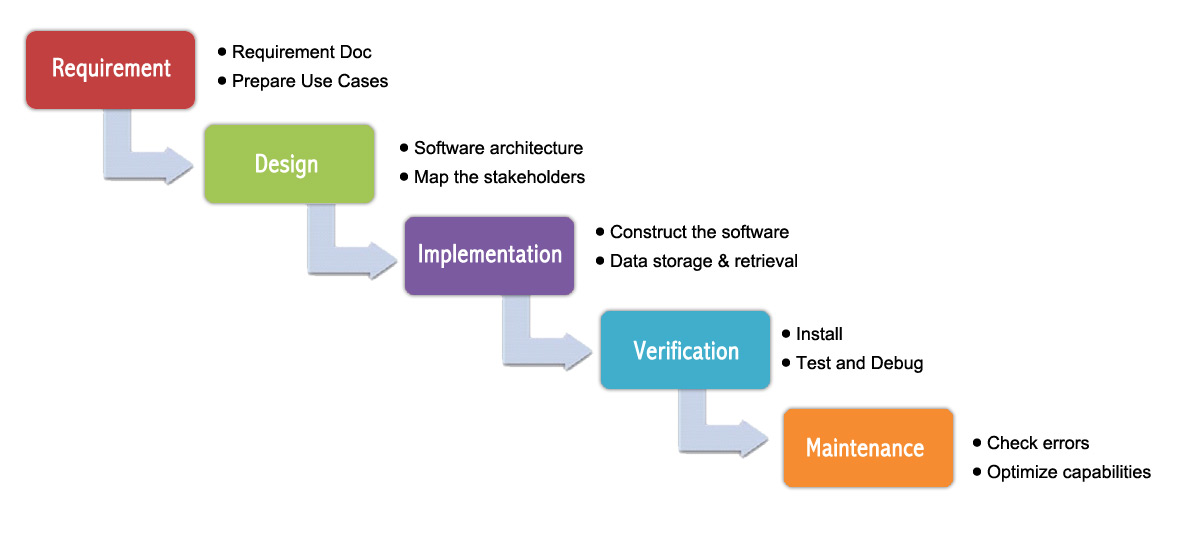
**5.1.2 Staffing Plan:**

Based on COCOMO model totally 47 team members are needed to build the software. The team members are classified into different teams.

**5.2 Work Plan**

**5.2.1 Work activities:**

For this product the requirements are clear and specific , the waterfall model life cycle is used to develop the software.



The table below describes the time duration for each phase

**5.2.2 Schedule Allocation:**

The project begins when the requirements are clearly understood and gathered. Once clear with the requirements design phase begins where entire design of the product is describe after this phase only development and implementation phase is performed. After the product is developed with the given specifications the testing process is performed to check the user acceptance testing and after the deployment if any updates or changes made to the product can be performed by maintenance team.

|  |  |
| --- | --- |
| **Phase** | **Duration of each phase** |
| **Requirement phase** | To collect the requirements and understand it may require 15 days |
| **Design phase** | Designing may take 20 days |
| **Implementation phase** | Constructing the software may take 30 days |
| **Verification phase** | Testing and debugging the product may take 40 days |
| **Maintenance phase** | Want make any enhancements or upadates after deployment of the product it may require 10 days |

**5.2.3 Resource Allocation:**

Resource allocation is the process of assigning and scheduling available resourcesin the most effective and economical manner.

|  |  |  |
| --- | --- | --- |
| **Work Flow** | **Team members** | **Working days** |
| Requirement phase | Requirement Engineer | 15 |
| Design phase | Architect Engineer | 20 |
| Implementation phase | Programmer | 30 |
| Verification phase | Tester | 40 |
| Maintenance phase | Developers | 10 |

**5.3 Control Plan**

**5.3.1 Quality control plan**

Quality is the degree to which any product or service possesses a desired combination of attributes, to satisfy the stated and implied needs.

The following factors will influence the quality

* Reusability
* Adaptability
* Testability

**5.3.6** **Metrics Collection Plan**

The below the metrics are collected from the client requirement to produce the quality product.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metrics | | | | | | |
| Phase |  | Simplicity | Quality | Process improvement | User Sentiment | Productivity |
| Requirement Analysis | Yes | Yes | No | Yes | No |
| Design | Yes | Yes | Yes | No | No |
| Implementation | Yes | Yes | Yes | No | Yes |
| Testing | Yes | Yes | Yes | Yes | Yes |

**5.4 Risk Management Plan**

The purpose of risk management is to identify potential problems before they occur so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.

A risk management plan is a document that a [project manager](https://en.wikipedia.org/wiki/Project_manager) prepares to foresee risks, estimate impacts, and define responses to issues. It also contains a [risk assessment matrix](https://en.wikipedia.org/wiki/Risk_matrix) .

The risk for any project may be backup failure, hardware failure.

**6 Supporting Process Plan**

**6.1 Testing Plan**

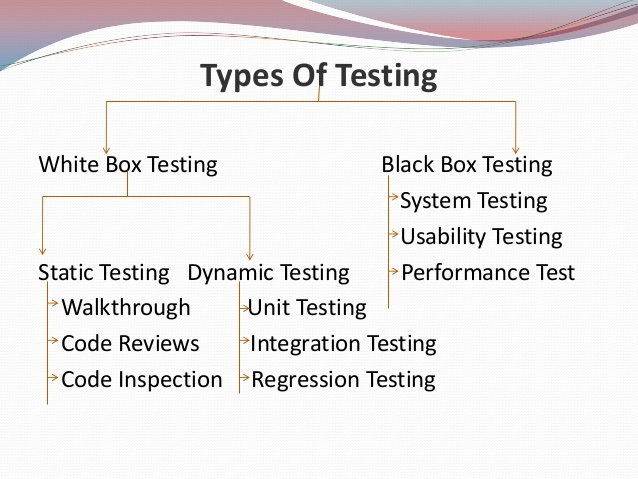
**Testing plans are classified as two types:**

1. **Execution Based Testing**
2. **Non-Execution Based Testing**
3. **Execution Based Testing: This phase is again divided into two types:**
4. **White Box Testing**
5. **Black Box Testing:**

**a)White Box Testing: - This** is also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing. It is a [software testing method](http://softwaretestingfundamentals.com/software-testing-methods/), in which the internal structure or design or implementation is being tested is known to the tester.

White Box Testing Levels are:

* **Unit Testing:** White-box testing is done in unit-testing in order for early detection of defects and also rectify those defects as soon as possible and also helps to prevent the occurrence of the defects in future.
* **Integration Testing:** White-box testing at this level is written to test the interactions between the interfaces with each other.
* **Regression Testing:** White-box testing during regression testing is the use of recycled white-box test cases of unit and integration testing levels.



1. **Black Box Testing:**

* Black box testing is the [Software testing method](https://www.softwaretestingclass.com/what-is-software-testing/) which tests the software without knowing the internal code structure or program.
* Basically software under test is called as “Black-Box”, we are treating this as black box and we test the software without knowing the internal software structure.

1. **Non-Execution Based Testing:** Non-execution based testing means the module is always verified by a team.

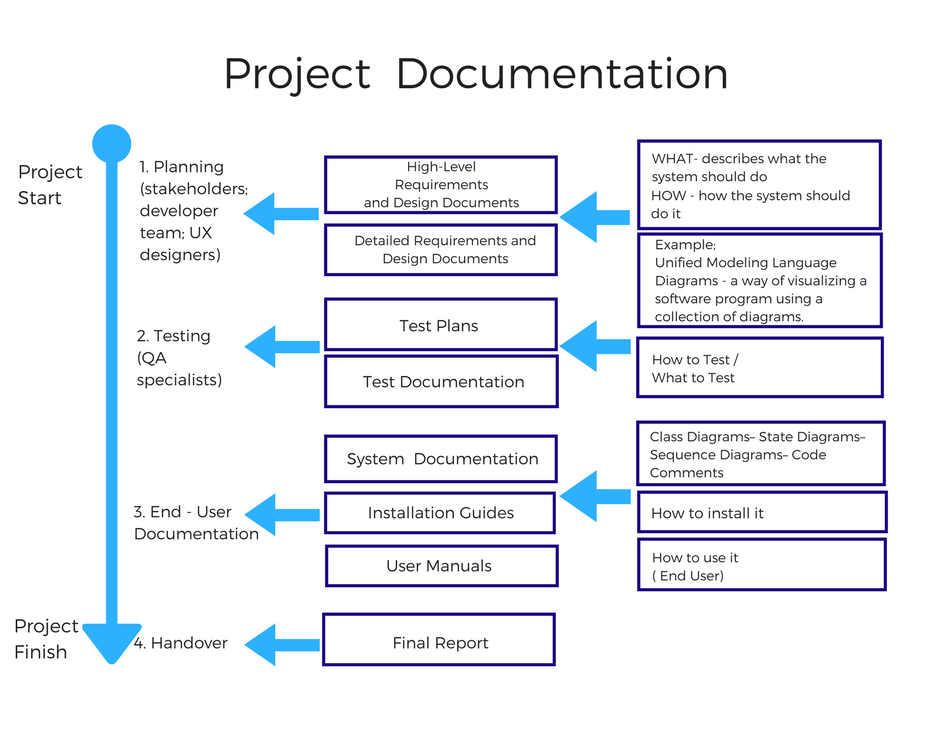
The non-execution based testing depends on fault detection strategy. This is further classified as, Documentation Plan

* In this project how much testing is required?

Until the user acceptance is achieved.

**6.2 Documentation Plan**

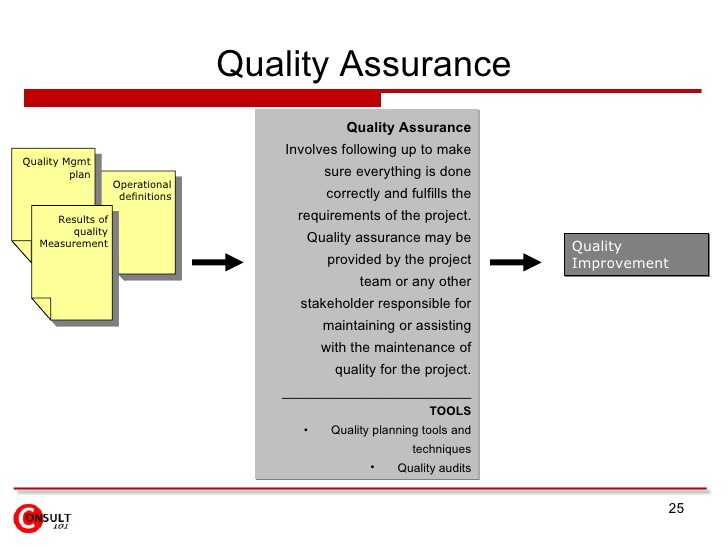
The project documentation plan contains the following

****

**6.3 Quality Assurance Plan**

A quality assurance plan is a document, that is constructed by the project team, in order to ensure whether the final product is having the maximum quality or not.

* A quality assurance plan contains a set of documented activities which are meant to ensure that customers are satisfied with the goods or services that are provided by a company

****

**6.4 Process Improvement Plan**

Software process improvement is usually one of the most important areas an organization considers when improving the overall performance of its business processes and practices. Capability Maturity Model Integration (CMMI) is a process level improvement training and appraisal program

