**CS 631-01**

**ASSIGNMENT - 02**

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

**INDEX Pg.no**

1. Overview 4

1.1. Project Summary 4

1.1.1. Purpose, Scope, and Objectives 4

1.1.2. Assumptions and Constraints 4

1.1.3. Project Deliverables 5

1.1.4. Schedule 5

2. Reference Materials 5

3. Definitions and Acronyms 6

4. Project Organization 6

4.1. External Interfaces 6

4.2. Internal Structure 6

4.3. Roles and Responsibilities 7

5. Managerial Process Plans 8

5.1. Start-Up Plan 8

5.1.1. Estimation Plan 8

5.1.2. Staffing Plan 9

5.2. Work Plan 10

5.2.1. Work activities 10

5.2.2. Schedule Allocation 12

5.2.3. Resource Allocation 13

5.3. Control Plan 14

5.3.1. Quality Control Plan 14

5.3.2. Metrics Collection Plan 15

5.4. Risk Management Plan 16

6. Supporting Process Plan 17

6.1. Testing Plan 17

6.2. Documentation Plan 24

6.3. Quality Assurance Plan 24

6.4. Process Improvement Plan 26

**1. Overview**

Chocoholics Anonymous (ChocAn) organization develops data processing software to provide consultation to the people who addicted to chocolate. The organization provides treatment and exercise to bring the addicted people out of that habit, and to make them healthier in life.

* 1. **Project Summary**
     1. **Purpose, Scope, and Objectives**:

**Purpose: -**To develop the data processing software for ChocAn organization which maintains the provider and consumer details.

**Scope: -**  Only the members with plastic card can access the service.

**Objective: -** The main objective is to automate the data processing. Health care providers will interact with system via terminal to provide service to the member and to provide the cost of service. It generates the report which is sent over the mail to both provider (list of members to whom service is provided) and member (service offered).

**1.1.2. Assumptions and Constraints**:

Assumptions and Constraints are given below:

* If the project will be completed within 4 months and 8 days (Approximately).
* Assuming the KDSI value as 10 in COCOMO model.
* Software must be delivered on time.
* The budget constraints must meet.
* The software must be user friendly.
* Delivered software must meet the specification as per the business need.
* The software must pass the acceptance testing done in client workspace.
* Software must be working in any environment and it must adapt to other company.
* Software must be flexible to add the additional functionality when it is needed in future.

**1.1.3. Project Deliverables:**

A completed project with source code and documentation must be delivered on February 28, 2019. After delivering the software, presentation will be given on the software with actual data.

**1.1.4. Schedule:**

By using COCOMO model in estimation plan, it will take 5 months to develop the software. Duration for each workflow is given below:

* Requirement workflow (3 weeks, 5 person)
* Design workflow (4 weeks, 6 person)
* Implementation and coding workflow (5 weeks, 11 person)
* Testing workflow (6 weeks, 12 person)
* Deployment and maintenance (2 days, 6 person)

**2. Reference Materials**

* Referred from textbook Object-Oriented and classical software engineering

(8thedition) by Stephen.

* Online websites used for reference:
* <https://www.guru99.com/black-box-testing.html>
* <https://www.guru99.com/white-box-testing.html>
* <https://diceus.com/top-7-software-quality-metrics-matter/>
* <https://en.wikipedia.org/wiki/PDCA>

**3. Definitions and Acronyms**

**ChocAn:** ChocAn is an organization which is dedicated to chocolate addicted people to provide medical treatment with health care professionals.

**COCOMO (Constructive Cost model):** This model is used for cost estimation analysis and it was developed by Barry W. Boehm.

**Walkthrough:** It is a form of software peer review in which designer or programmer leads members of the development team and to other team members go through the software product.

**Inspection:** It is the most formal review type. The goal of the inspection is to identify the defects.

**4. Project Organization**

**4.1. External Interfaces:**

Team members who involved externally in the project. It includes client, Project Managers.

**4.2. Internal Structure:**

The team members who are involved in the development of software project internally. It includes developer, tester and supporting team.

**4.3. Roles and Responsibilities:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **Teams/Team Member** | **Role** | **Responsibilities** |
| **1.** | ***X team (4 members)*** | Requirement Engineer | Communicates with client to get the requirement clearly and will analysis the requirement. Acts as intermediate between client and development team. |
| **2.** | ***Y team (6 members)*** | Designer | Will do the architectural design to the software. |
| **3.** | ***Z member*** | Development Lead | Lead will ensure to manage the project and make sure of the task done by the team member. And lead will send the report to client on daily basis. |
| **4.** | ***P team (10 Members)*** | Coder | Coder will be able to understand design documentation and understand development environment. Must write the code in such a way it builds the software based on the architecture design. |
| **5.** | ***Q member*** | Quality Assurance | QA will set the standard for quality. And will ensure whether testing and development team follows this standard for producing high quality product. |
| **6.** | ***R member*** | Teste Lead | Test lead will be responsible for planning, managing and deploying test activities. |
| **7.** | ***S team (11 members)*** | Tester | Tester will do testing on the developed product and will find the defects if any bugs in product. Finally, will provide the quality software with error free bugs. |
| **8.** | ***T team (6 members)*** | Support team (Development people) | Will set the environment for deployment and will support post- delivery maintenance. |

**5. Managerial Process Plans**

**5.1. Start-Up Plan**

**5.1.1. Estimation Plan:**

COCOMO model is used to provide the estimation plan for the project. Estimation will be tracked and later it will be modified based on the project progress.

By assuming that the product is built in organic mode because of small scale project and considering 1000 lines of code (KDSI). By using this assumption nominal effort is calculated below:

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

= 3.2\*(10) ^1.05

= 36 person-months (Approximately)

and 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

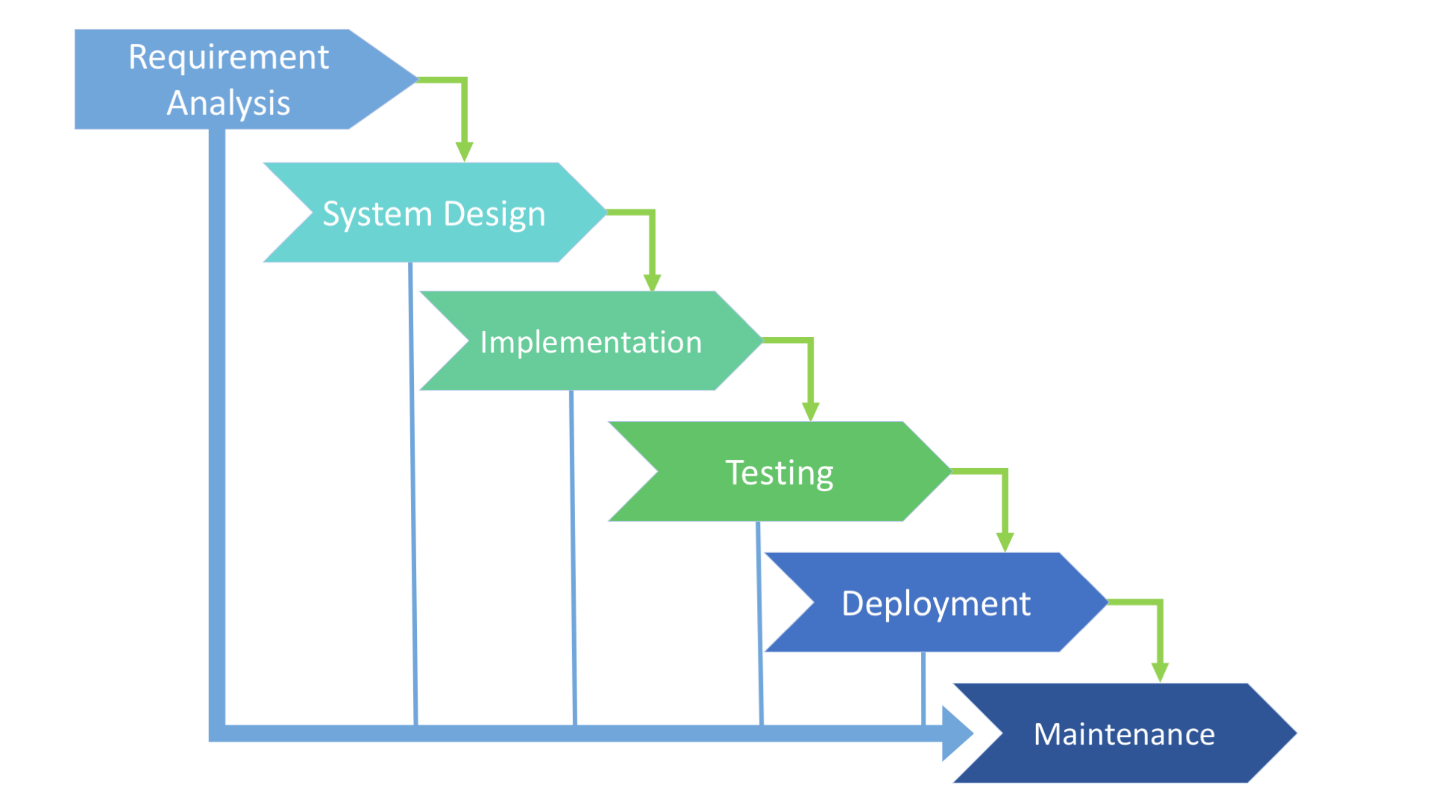
**5.1.2 Staffing Plan:**

Based on COCOMO model totally 47 team members are needed to build the software. The team members are classified into different teams and it’s shown in the form of pie chart.

**5.2. Work Plan**

**5.2.1. Work activities:**

The below steps are followed to do work activities for this project:



Since it’s small scale project and requirements are very clear and specific. The best life cycle model to develop the software will be Waterfall model.

The below tabular form shows the duration of each phase:

|  |  |
| --- | --- |
| **Phases** | **Duration of each phase** |
| Requirement Analysis phase | For collecting the client requirement clearly and analyzing the collected requirement it will take hardly 3 weeks. |
| Design Phase | For doing the architectural design in which product is broken down into modules and for each module must do detailed design. It will take around 4 weeks to complete. |
| Implementation Phase | For coding and unit testing it will take 5 weeks to finish it. |
| Testing Phase | To do quality testing it will take 6 weeks of time to complete it. |
| Deployment Phase | For deploying the product in client environment or to set up the environment in client workspace it will mostly take 2 days’ time. |
| Maintenance phase | Once product is deployed, if there is any minor defect then maintenance will be supported by development team on the same day |

The following diagram depicts the pictorial representation of time required for doing each phase.

**5.2.2. Schedule Allocation:**

Once requirements are clearly identified and documented, then only design phase will be implemented. After doing the detailed design the coder can perform the coding task. Once coding is done then testing phase will be started. Once tester provides quality testing with no defect then only the software will be deployed. So, each phase is dependent on preceding phase. Below diagram depicts the information about preceding phase dependences.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Testing |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  | | |  |  |
| Design |  |  |  | |  |  |  |  |  |
| Requirement Analysis |  | |  |  |  |  |  |  |  |

**5.2.3 Resource Allocation:**

The below tabular form and chart will depict the information about resource allocation to the project and time duration required for each team.

**Tabular Form:**

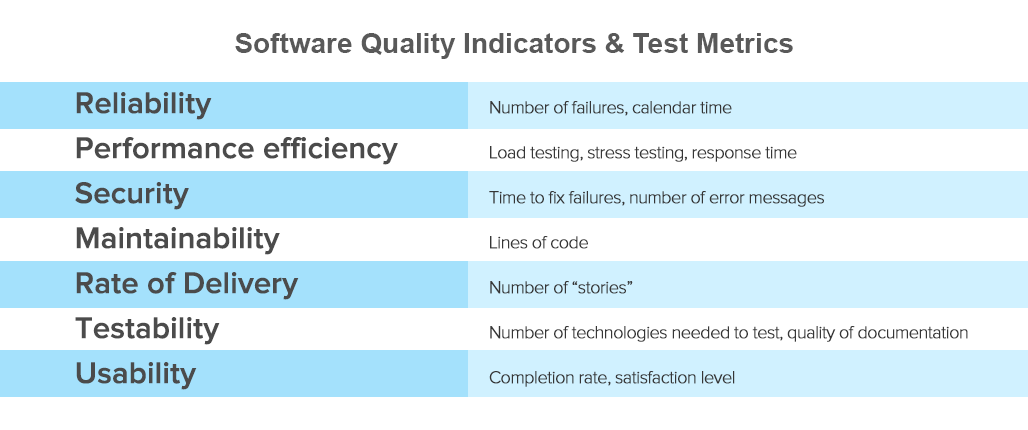
|  |  |  |  |
| --- | --- | --- | --- |
| **PHASES** | **NUMBER OF WORKING DAYS** | **Resource Allocated team** | **Number of Resource ( Person ) Allocated** |
| Requirement Analysis Phase | 3 weeks | Requirement Engineer | 5 |
| Design Phase | 4 weeks | Designer | 6 |
| Implementation | 5 weeks | Coder | 11 |
| Testing | 6 weeks | Testing team | 12 |
| Deployment and post -delivery maintenance | 2 days | Development team | 6 |

**Chart:**

**5.3. Control Plan**

**5.3.1. Quality Control Plan:**

The below diagram shows the way in which quality is measured and how it can be controlled.



**5.3.2**. **Metrics Collection Plan**

The list of metrics collected is as follows:

* Number of classes and interfaces
* Complexity of software
* Functional point analysis
* Algorithm analysis
* Number of bugs

These metrics are collected from requirement document and it’s stored in hard disk.

**Metrics in each phase:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metrics** | | | | | | |
| **Phase** |  | **Number of classes and interface** | **Software Complexity** | **Functional Point Analysis** | **Algorithm Analysis** | **Number of bugs** |
| *Requirement Analysis* | No | Yes | Yes | No | No |
| *Design* | Yes | Yes | Yes | Yes | No |
| *Implementation* | Yes | Yes | Yes | Yes | Yes |
| *Testing* | Yes | Yes | Yes | Yes | Yes |

**5.4. Risk Management Plan**

The risk factors included are given below:

**No existing product:** There is no existing product with which the new product can be compared. Accordingly, it will not be possible to run the product in parallel with an existing one. Therefore, the product should be subjected to extensive testing.

**Inexperienced Client:** The client is not aware of the computer. So, must pay the special attention in requirement workflow and must communicate with client to develop user friendly product.

**Fault in design:** Because of the major design fault testing must be performed carefully during the design work flow. Also, each team member must test their own code.

**Hardware Failure:** In case of any hardware failure the other machinery must be spare for the replacement of that broken machine.

**6. Supporting Process Plan**

**6.1. Testing Plan**:

Testing plans are of two types

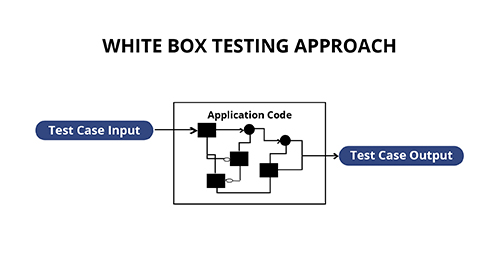
1. Execution based testing and

2. Non-Execution based testing

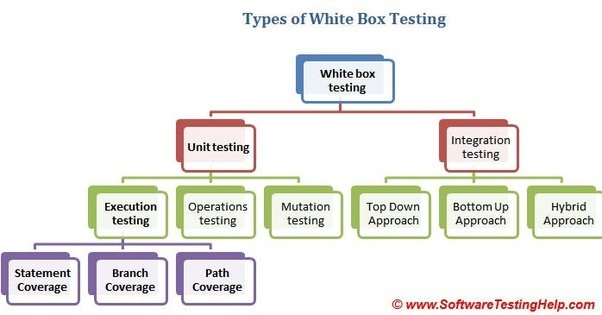
**1. Execution based testing:** The modules are tested against each test cases. There are two types of execution-based testing:

1. White box testing
2. Black box testing

**1. White box testing**: White box testing will test the internal structure and working of a product.



**Types of White box testing:**



**Unit testing:**

It is performed by developer and individual module is tested by developer who build the code for that module. Unit testing is further classified into execution testing, operation testing and mutation testing.

**Execution testing:**

Testing the expected code with actual results which is done by the developer. It is divided in to three parts:

***Statement Coverage:***

Every statement in the code tested at least once during the testing.

***Branch Coverage:***

It checks every possible path of the software application.

***Path Coverage:***

It refers to designing test case such that all independent paths in a program are tested at least once.

**Integration testing:**

This is done by testing team, in which entire product functionality is tested. It includes top down, bottom up and hybrid approach.

**Black box testing:**

It focusses only on input and output of the software system without any knowledge of the internal structure of the software. Black box testing is done by testing team.



**Types of Black Box Testing:** Black Box testing are of several types, but the following are the main ones.

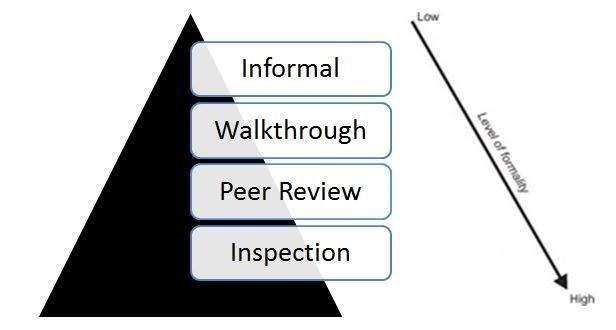
* ***Functional Testing:*** This is used for the functional requirements of a system, this is done by the software testers.
* ***Non-functional testing:*** This is not related to any of the testing of a specific functionality, but non-functional requirements like performance, scalability and usability are tested by testing team.
* ***Regression testing:*** This testing is done after fixing the codes, upgrades or any of the other system maintenance which is to check the new code has not affected the existing code. This testing is done by testing team.

**Black box testing strategy:**

* ***Equivalence Class Testing*:** This is used to minimize the number of possible test cases for an optimum level while the maintenance is reasonable for the test coverage.
* ***Boundary Value Testing:*** Boundary value testing is focused on the boundary values. This technique is to determine whether a certain range of values will be acceptable or not.
* ***Decision Table Testing:*** Decision table keeps the causes and its effects in a matrix. There is a unique combination in each column.

**Non-Execution based testing:**

Testing the software without running the test cases. It is also known as static testing. Two types of non-execution-based testing is Walkthrough and Inspections.



**Walkthrough:**

There are two ways to conduct Walkthrough. The first is participant driven. Participating member will present the list of items which is unclear and items they think incorrect. The representative of the analysis team will respond to the query and clarifies what is unclear to reviewer.

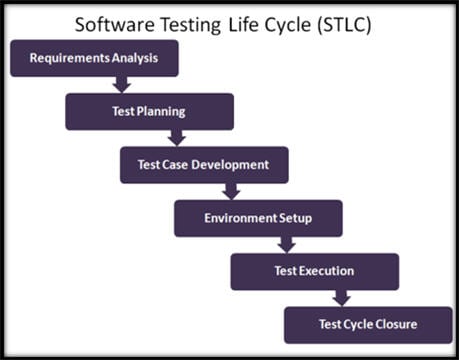
The second way of conducting review is document driven. A person will be responsible for the document either individually or part of the team. Walks the participants through this document with the reviewers interrupting either with the prepared comments or comments triggered from the presentation.

**Inspections:**

Inspection goes in depth beyond the walkthrough and has five normal steps:

* An overview of the document to be inspected by one of the representative in team and must produce the document which will be reviewed by other participants.
* In preparation, participants will understand the document in detail and list the fault types identified in recent inspections.
* As with walkthrough, the purpose is to find and document the faults, not to correct them. Within one day the leader of the inspection team must produce a written report of the inspection.
* In the rework, the individual responsible for the document resolves all faults and problem noted in the written report.
* In the follow-up, the moderator must ensure that every issue raised has been solved satisfactorily, by fixing the document or clarifying items incorrectly flagged as faults.

Software Testing Life Cycle:



* **Requirement Analysis:**

Testing team performs requirements analysis and the team will take approximately 1 week for this activity.

* **Test Planning:**

Testing plan will be drawn by testing team and it takes around 1 week to complete it.

* **Test Case Development:**

Testing team will write test cases for each functionality separately and also have the integrated test cases and test case will have all the combinations including negative test cases. This cycle takes long time almost 3 weeks.

* **Environment Setup and Test Execution:**

Test case have to be written properly. Based on the test cases testing team will begin executing this test cases. Test case execution required 1 week time to execute all the test cases.

* **Test cycle closure:**

After completing the test case execution. If there is no defect found in testing phase, then testing cycle will be closed. If defect exist in the product, then defect cycle management process will be started and it will be continued until the product becomes error-free.

**Testing Estimation Plan:**

**How much testing is needed:**

The above testing methods must be done by both development and testing team until the acceptance testing is succeed and when it met the client requirement.

**6.2. Documentation Plan**

Documentation plan includes source code, user manuals and requirement document:

**Source code:**

The code that has been building to make the software work as per the functionality. Developer will do documentation for source code.

**Requirement Document:**

Requirement engineer will collect the requirement from client and convert those requirement in document format.

**User manuals:**

Developer will provide user manuals to client. It includes how to operate the system and how to use it.

**Defect tracking document:**

This document is generated by testing team, it has defect log that exist in system.

**6.3. Quality Assurance Plan**

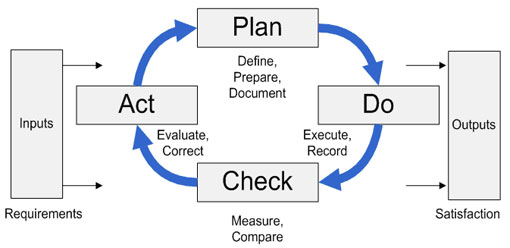
Quality assurance plan is a documentation which will be constructed by the project team and meant to ensure the final products are of the better quality. This contains a set of documented activities which is to ensure that client is satisfied with the product which the company delivers. The quality assurance plan has the below documentation activities:

* Quality assurance plan defines quality objectives- Six sigma uses predefined algorithm for determining quality objectives and level of quality is defined.
* Defining roles and responsibilities: After defining and stating quality objectives, the next step is to define roles and responsibilities of team members. Each role has to be assigned to team member.

**6.4. Process Improvement Plan**

Process improvement is a commitment given to the customer. The Plan-Do-Check-Act model is a cyclic process which is used to improve the process continuously.

The Plan-Do-Check-Act model has four stages: Plan, DO, Check, and Act, this has explained below in detail.



**Plan:** It will identify key business process that will produce a desire outcome. Understanding the business process required to run the business and increase the performance sets which is the base for process improvement. Once foundation is set and key process are in place it is easier to identify the problems within a process and to create a suitable process plan to deal with it.

**Do:** Key business process needed constant evaluation to eliminate defects and to maximize the performance. It is mandatory for business organization to understand the problem and analyze the root cause to develop a process plant to resolve systemic issues and to improve the process. Testing is the part of “Do” phase to provide effective solution to the process issue.

**Check:** The effectiveness of key processes depends on the effectiveness of process plan that has been developed to improve. Without knowing the effectiveness of corrective action and results, it will be impossible to determine the process capabilities and improvements.

**Act**: Adjusting the process by incorporating corrective actions. It will allow the organization to re-deploy the new processes with the workforce.