CIS ~~572 Object Oriented System Design~~ 376 Software Engineering Two

ChocoAn Software Project Management Plan ( SPMP )

Version - 0~~3~~4

Team : ~~She-Wolverines~~ The Best and Brightest



Submitted to : Prof. Thomas Steiner

## 

**Software Project Management Plan Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **VersionNumber** | **Date** | **Revision Author** | **Description of Revision** |
| 1 | 05/28/2019 | She-Wolverine | Initial SPMP |
| 2 | 06/11/2019 | She-Wolverine | Revised : Task network diagram , Major software functions and LOC and Reconciled Estimates |
| 3 | 07/07/2019 | Second Run | Revising SPMP |
| 4 | 8/21/2019 | Final Run | Revising SPMP |

# Table of Contents

[**1.0 Introduction**](#_heading=h.1fob9te) **4**

1.1 [Problem statement](#_heading=h.3znysh7) 4

[1.2 Project scope](#_heading=h.2et92p0) 5

[1.3 Major software functions](#_heading=h.tyjcwt) [6](#_heading=h.tyjcwt)

[1.4 Performance/Behavior constraints](#_heading=h.3dy6vkm) 7

[1.5 Management and technical constraints](#_heading=h.1t3h5sf) 7

[**2.0 Project Estimates**](#_heading=h.4d34og8) **6**

[2.1 Historical data used for estimates](#_heading=h.2s8eyo1) 6

[2.2 Initial Estimate](#_heading=h.17dp8vu) [7](#_heading=h.17dp8vu)

[2.3 Estimation techniques applied and results](#_heading=h.26in1rg) [11](#_heading=h.26in1rg)

[2.4 Reconciled Estimate](#_heading=h.lnxbz9) [13](#_heading=h.lnxbz9)

[2.5 Project Resources](#_heading=h.35nkun2) [13](#_heading=h.35nkun2)

[**3.0 Risk Management**](#_heading=h.1ksv4uv) **[13](#_heading=h.1ksv4uv)**

[3.1 Project Risk Table](#_heading=h.44sinio) [14](#_heading=h.44sinio)

[3.2 Overview of Risk Mitigation, Monitoring, Management](#_heading=h.3j2qqm3) [16](#_heading=h.3j2qqm3)

[**4.0 Project Schedule**](#_heading=h.1y810tw) **[17](#_heading=h.1y810tw)**

[4.1 Project task set](#_heading=h.4i7ojhp) [17](#_heading=h.4i7ojhp)

[4.2 Task network](#_heading=h.2xcytpi) [19](#_heading=h.2xcytpi)

[4.3 Timeline chart](#_heading=h.1ci93xb) 20

[**5.0 Staff Organization**](#_heading=h.3whwml4) **[24](#_heading=h.3whwml4)**

[5.1 Team structure](#_heading=h.2bn6wsx) [24](#_heading=h.2bn6wsx)

[5.2 Management reporting and communication](#_heading=h.qsh70q) [24](#_heading=h.qsh70q)

[**6.0 Tracking and Control Mechanisms**](#_heading=h.3as4poj) **[24](#_heading=h.3as4poj)**

[6.1 Quality assurance and control](#_heading=h.1pxezwc) [24](#_heading=h.1pxezwc)

[6.2 Change management and control](#_heading=h.49x2ik5) [25](#_heading=h.49x2ik5)

[6.3 Tools](#_heading=h.2p2csry) [26](#_heading=h.2p2csry)

[**7.0 Appendix**](#_heading=h.147n2zr) **26**

# 1.0 Introduction

This section provides an overview of the software engineering project. This document serves as the project plan for the ChocoAn DataCenter Software development.

## 1.1 Problem statement

Chocoholics Anonymous is an organization to help individuals who are addicted to chocolate and need professional help to fight the addiction. Chocoholics Anonymous recruited professional healthcare providers like dieticians, internists and exercise specialists. All these professionals can work together and address the chocolate addiction issue faced by a member.

To make the functioning of the business easier, ChocAn has a terminal which allows providers to look up for the services that they can provide and ~~also~~ their costs are displayed on the screen. Providers can guide the members on which service would better suit them according to the kind of addiction they have. Members can swipe the card or enter their member number to receive the services. After providing the services to the members, providers can again swipe the members card, enter member details, provider details, service details and bill the ~~c~~ChocAn. Providers can gather weekly bills and ChocAn will make payment to the provider based on the services and number of services that each provider had served for a given week. Member will receive details about the weekly services they received and ~~also~~ the total cost of the services rendered. Similarly, the provider will receive the reports about weekly services and weekly fee details. ChocAn provides flexibility to modify the members/provider details and ~~also~~ would delete them from the system if they are no longer part of ChocAn. Managers can view the accounts Payable. ChocAn has third party vendors like Bank and ACME which handles Money transfers to Providers, ~~and~~ handles the membership fees, and statuses respectively. Additionally, ChocAn can verify the member details from the ChocAn Data center before providing services.

## 1.2 Project scope

**1.2.1 Inclusions**

**Inputs:**

* Healthcare Professional (Provider) - Unique provider number, name, address, account number.
* ChocAn Member - Unique member number, name , address , status
* Service Record - provider number ,member number, service code, service fees, date\_of\_service

**Outputs:**

* Provider Directory - list of service name, service code and service fees.
* Provider’s weekly report - provider name and number , List of attended members and services provided, total fees.
* Member’s weekly report - member name and number , List of services availed including the provider name, total fees.
* ~~Summary~~ Manager report - List of all the providers to be paid that week, the no. of consultations and total fee for the week.

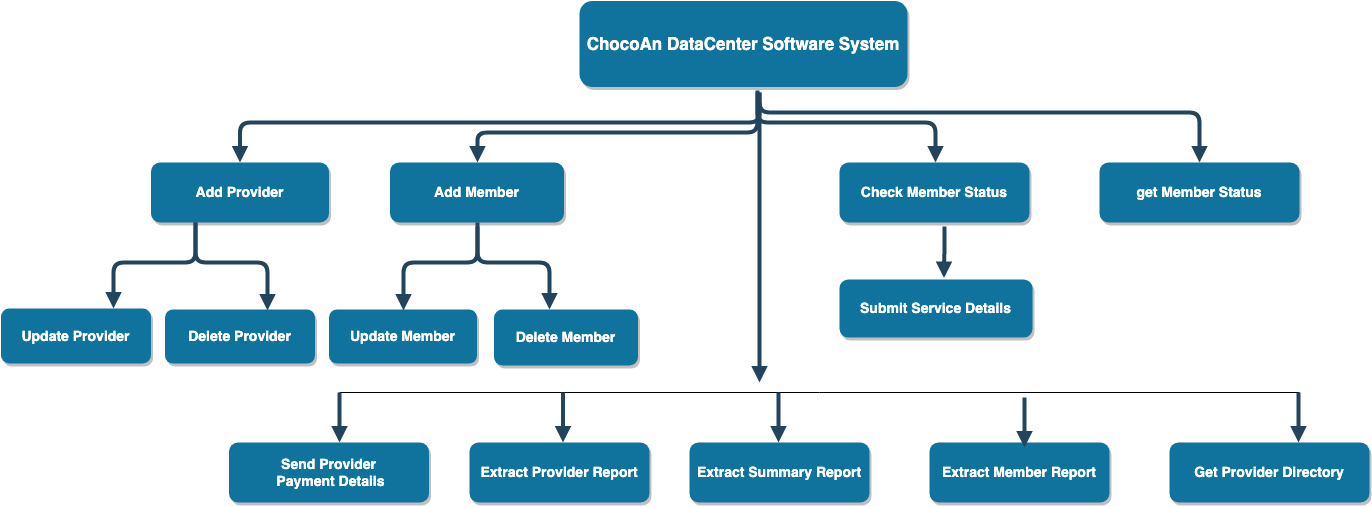
**Functionality:**

* ChocAn Manager - Initiate report generation.
* ChocAn Operator - Add, Update and Delete Member and Provider information.
* Acme Accounting Services(AAS) - update member status based on the payment history.
* EFT Banking - process providers weekly payments via EFT transfer.
* Provider - request for provider directory
* Provider - Submits service record indicating the service details

**1.2.2 Exclusions**

* As agreed we are not developing the simulation for this project.
* Development of the terminal is out of scope.
* Development of the Acme Accounting Services software for payment processing is out of scope.
* Implementation of the EFT components are out of scope.

## 1.3 Major software functions



## 1.4 Performance/Behavior constraints

* Batch job for generating the weekly reports runs every friday at midnight.
* Batch job for Acme Accounting Services runs everyday at 9 PM to update the ChocoAn DataCenter with the members status updates.
* System is to be available every day from 3 AM to 9 PM.

## 1.5 Management and technical constraints

**1.5.1 Management constraints**

The final product must be delivered by 08/22/2019.

Maximum resources working on the project is 4 people.

**1.5.2 Technical constraints**

None.

# 2.0 Project Estimates

This section provides cost, effort and time estimates for the projects

## 2.1 Historical data used for estimates

The below estimates are done based on prior project work experience and the term projects. A project with 20,000 lines of code and 20 modules had taken around 960 hours to complete the tasks. Also, a project having a duration of 3 months took 480 hours , was estimated based on 5,000 lines of code with 10 modules that are referred for calculating the estimates.

## 2.2 Initial Estimate

**2.2.1 SPMP Completion Estimate**

Initial SPMP completion will take around 24 hours of initial work in total to complete. Considering 6 hours of buffer time for any additional updates on the SPMP, total 30 hours are required.

**2.2.2 Overall project estimate**

**2.2.2.1 Line-of Code Estimate**

The estimates for lines of code are done based on each module of the project as below:

1. Modify member information
2. Modify Health Care Provider
3. Check member status
4. Report Generation
5. Saving the record or transaction
6. Modify Dietitians in system
7. Modify Internists in system
8. Modify Exercise Experts in system

Approximately 8000 lines of code with 1000 lines of code per module.

We will do 40 lines of code per day.

8000/ 40 = 200 days

200 days / 4 (no of persons) = 50 days

50 x 8 = 400 hours.

**2.2.2.2 Function Estimate**

|  |  |  |
| --- | --- | --- |
| **Function** | **Complexity** | **Weight** |
| modifyHealthCareService | Medium | 6 |
| modifyMemberData | Medium | 6 |
| checkMemberStatus | Low | 2 |
| addServiceRecord | Medium | 6 |
| generateProviderReport | Low | 2 |
| generateMemberReport | Low | 2 |
| generateSummaryReport | Low | 2 |
| getMemberData | Low | 2 |
| getDataForProvider | Medium | 4 |
| generateProviderDirectory | Medium | 5 |
| runBatchWeekly | Medium | 4 |
| runBatchEveryDay | Medium | 4 |
| modifyDietitianData | Medium | 6 |
| modifyInternistData | Medium | 6 |
| modifyExerciseExpertsData | Medium | 6 |
| **Total** |  | **63** |

#### 

Function points total = 63

63 Function points x 8 hours = 504 Hours.

63 Function Points / 4 people = 15.75 Function Points per person

**2.2.2.3 Tasks Estimate**

|  |  |
| --- | --- |
| **Task** | **Estimates (in hours)** |
| **Problem Definition** | 55 |
| **Requirement Gathering** | 84 |
| **Design** | 340 |
| **Development** | 140 |
| **Testing** | 185 |
| **Deployment** | 48 |
| **Total** | 852 |

**2.2.2.4 Total overall project time estimate in hours of effort**

After reviewing the complete SPMP document with all the team members, we agreed on to go with the task estimates. Therefore the total number of hours is 852.

## 2.3 Estimation techniques applied and results

A description of each estimation technique and the resultant estimates are presented here.

**2.3.1 Estimation technique 1 – lines of code**

We reviewed the complete SPMP document and based on that added ~~2~~ 1 more modules for the lines of code:

1. Modify member information
2. Modify Health Care Provider
3. Check member status
4. Report Generation
5. Saving the record or transaction
6. Process Payment
7. Provider Services
8. Modify Provider information

Total lines of code is ~~7~~ 8000 as 1000 lines of codes per module.

**2.3.2 Estimate for technique 1 – lines of code**

We will do ~~20~~ 40 lines of code per day.

~~7~~8000/~~20~~ 40 = ~~350~~ 200 days

~~350~~ 200 days / 4 (no of persons) = ~~88~~ 50 days

~~88~~ 50 x 8 = ~~704~~ 400 hours.

**2.3.3 Estimation technique 2 – function points**

After reviewing the complete SPMP document, we agreed on the below functions:

|  |  |  |
| --- | --- | --- |
| **Function** | **Complexity** | **Weight** |
| Add Provider | Medium | 4 |
| Update Provider | Medium | 4 |
| Delete Provider | Low | 2 |
| Add Member | Medium | 4 |
| Update Member | Medium | 4 |
| Delete Member | Low | 2 |
| Extract Provider Report | Medium | 5 |
| Extract Member Report | Medium | 5 |
| Extract Summary Report | Medium | 5 |
| Check Member Status | Medium | 4 |
| Get Member Status | Medium | 6 |
| Submit Service Details | High | 8 |
| Get Provider Directory | Medium | 4 |
| Send Provider Payment Details | High | 8 |
| Total |  | 65 |

Function points total = 65

65 Function points x 8 hours = 520 hours.

65 Function Points /4 people = 16.25 Function Points per person

**2.3.4 Estimation technique 3 – process/task**

It will be the same as the task estimate as we did the task estimates after doing the function points and lines of codes estimate.

**2.3.5 Estimate for technique 3 – process/task**

Total hours for the task estimate are 852 hours.

## 2.4 Reconciled Estimate

Taking an average of the above three methods.

~~704~~ 400 + 520+ 852 /3 = ~~692~~ 591 hours

The total estimates for the tasks are 692 hours.

## 2.5 Project Resources

**Human Resources:**

* The team size is 4 members.
* All the estimates for effort have been calculated for a 4-person team, with a Team Lead and 3 supporting members.

# 3.0 Risk Management

This section discusses project risks and the approach to managing them.

## 3.1 Project Risk Table

|  |  |
| --- | --- |
| **LEVELS** | |
| Low | 1 |
| Medium | 2 |
| High | 3 |
| Critical | 4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RISK** | **PROBABILITY** | **DEGREE OF IMPACT** | **IMPACT OF RISK** | **MITIGATION PLAN** | **CONTINGENCY PLAN** |
| Changes in the requirements or incorrect requirements | 1 | 3 | It will lead to a lot of rework and induce more cost to make the changes. It will also lead to a low quality product. | We should be in touch with the client and business team and have an open communication with them to ensure we have correct requirements | We should document the requirements properly and get them signed off by the client and business team. |
| Change in the Technology | 2 | 3 | We need to make changes in the software as the technology changes | Look for how long it can work unaffected by the change in technology | We should keep track the technology and maintain our software accordingly |
| Unavailability of Provider | 2 | 3 | Members will be unattended, delay in providing services and chaos in managing the members | Members should be asked to see other provider in the absence of certain provider | Provider should notify the manager beforehand if he is going to be unavailable and members should be managed accordingly |
| Improper working of software | 1 | 4 | All the process will be either delayed or hindered | Find where is the problem | Fix the problem as soon as possible |
| Developers Unavailability | 2 | 3 | Delay in Deliverables | Backup plan for task completion is required | Division of tasks within the available resources |
| Lack of coverage in testing | 2 | 3 | End users would experience defects that may result in program not running to highest expectations | Have test cases prepared prior to testing changes by developers | Receive proper Acceptance Criteria during meetings. |
| Team does not understand the scope of the task | 3 | 4 | Lack of understanding would have the customer return the product. If there are any requirements missing from the provided list, it would push back the release. | Prior to accepting tasks during the development cycle, verify Developers and QA know scope of the project | During meetings, ask questions and if one thing is unsure, bring it up and continue with the meeting once everyone understands the scope of the task. |
| Releasing the product with new defects not found during final testing | 2 | 3 | QA is not running complete regression and sanity testing prior to a release | Have a checklist of what existing functionality needs to be tested in the current release | Use an external program that keeps track of test cases that have been tested and provide details such as screen shots of test being successful where applicable |

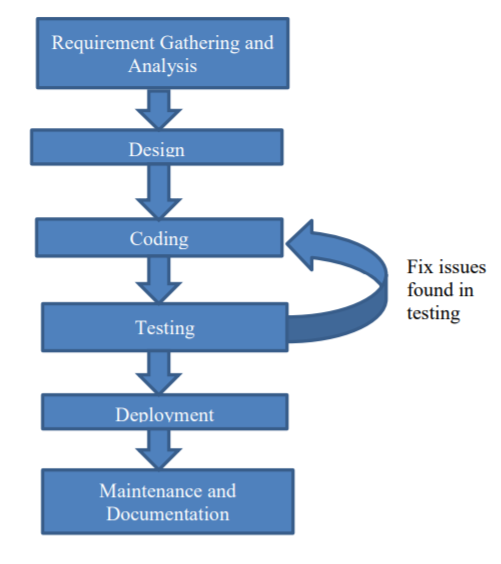
## 3.2 Overview of Risk Mitigation, Monitoring, Management

* Risk Factor= (Probability) x (Degree of Impact)
* If the Risk factor>4, we will implement the mitigation plan.
* We will monitor for the risks side by side.
* We’ll try to prevent the risk as far as possible or have a mitigation plan in hand.
* We’ll try to concentrate firstly on the higher level risks.
* The team will communicate regularly to predict new risks and plans to mitigate and prevent them.
* The risk table will be updated as more risks are monitored.
* Team meetings will be conducted regularly to manage the risks, discuss new risks and plans to handle them.

# 4.0 Project Schedule

Project schedule for this project is explained below. Schedule is set according to different phases in the project development task.

## 4.1 Project task set

“Waterfall Life Cycle Model” is used as a process model in this project. Framework activities and task set selected for the project are presented as below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Task** | **Project Sub-Task** | **Task Description** | **Hours to Complete** |
| Project Planning | Project Planning | Plan project tasks as per the expected project outcome. | 48 |
| Requirements | Requirements Gathering | Gather and understand requirements from client. Give valuable input to client if required based on domain knowledge. | 24 |
| Requirements Analysis & Review | Analyze requirements and come up with agreement with client to accept those requirements. If changes required in them, update requirements with client acceptance. | 60 |
| Design | High Level Design | Perform high level design to decide modules and their internal communication. Decide the integration strategy for submodules. | 100 |
| Low Level Design | Design low level approach to decide coding techniques, coding standards, flow charts, inter process communications, code logic and code management. | 148 |
| Design Reviews | Perform Code design reviews with peers or Code inspections with professor/ clients and update design plan accordingly. | 74 |
| Development | Coding | Develop software with proper coding standards. Make sure all promised features are included in software with correct software stability. | 50 |
| Coding reviews and code inspections | Perform peer code reviews to confirm code is written correctly. Change code or code logic as per the review / inspection comments. | 72 |
| Testing | Functional and Non- Functional testing | Develop test plans to Perform functional and non-function testing on developed code and report defects. | 48 |
| User acceptance testing | Perform user acceptance testing to make sure user gets look, feel and all features promised in a software. | 48 |
| Defect fixing | Work on Defects found during testing phase and fix then in code. Perform retesting and regression testing to make sure defect is fixed and no other module is broken due to fixes in code. | 72 |
| Deployment and documentation | Deployment | Deploy software to client or professor with live demo and presentation. | 24 |
| Documentation | Provide documentation to client or professor to understand software better | 24 |
| Update of Document | Document Correction | Revise current documentation and validate all information is correct and accurate. | 40 |
| Validation of Document | Approval of all team members of any revised document that has occured. | 20 |
| **Total** | | | 852 |

## 4.2 Task network

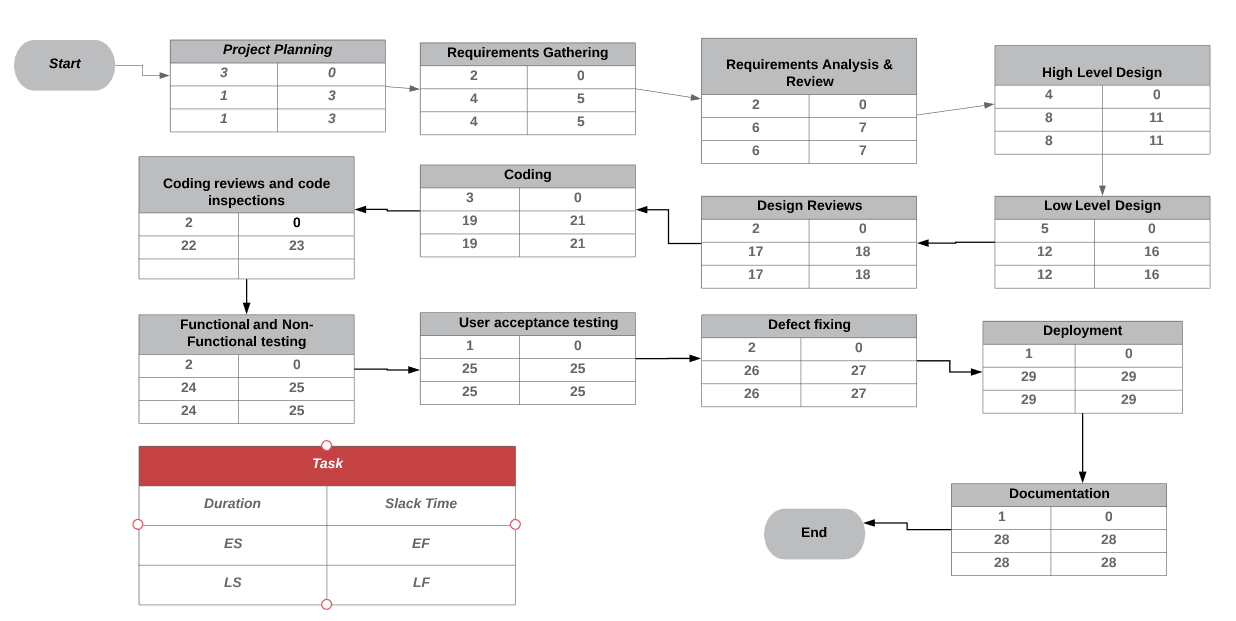
Project tasks and their dependencies are noted as shown below.

|  |  |
| --- | --- |
| Dependencies | Project Tasks |
|  |  |

## 4.3 Timeline chart

A project timeline chart is shown below in Table format and Gantt chart format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Task** | **Project Sub-Task** | **Time to complete in Hours** | **Team members** | **Time to complete in Days** | **Time to complete tasks at team level** |
| Project Planning | Project Planning | 48 | 5 | 3 | 5/13 – 5/15 |
| Requirements | Requirements Gathering | 24 | 5 | 2 | 5/16 – 5/17 |
| Requirements Analysis & Review | 60 | 5 | 2 | 5/18 – 5/19 |
| Design | High Level Design | 100 | 5 | 4 | 5/20 – 5/23 |
| Low Level Design | 148 | 4 | 5 | 7/19 – 7/23 |
| Design Reviews | 74 | 4 | 9 | 7/23 – 7/25 |
| Development | Coding | 50 | 4 | 6 | 7/26 – 7/31 |
| Coding reviews and code  inspections | 72 | 4 | 4 | 8/1 – 8/4 |
| Testing | Functional and Non-  Functional testing | 48 | 3 | 4 | 8/5 – 8/8 |
| User acceptance testing | 48 | 2 | 3 | 8/9 – 8/11 |
| Defect fixing | 72 | 4 | 3 | 8/12 – 8/14 |
| Deployment  and  documentation | Deployment | 24 | 4 | 2 | 8/15 – 8/16 |
| Documentation | 24 | 4 | 2 | 8/17 – 8/19 |
| Update of Document | Document Correction | 40 | 4 | 2 | 7/3 - 7/14 |
| Validation of Document | 20 | 4 | 3 | 7/15 - 7/18 |





# 5.0 Staff Organization

## 5.1 Team structure

The team consists of 4individuals including the Team Lead.

## 5.2 Management reporting and Communication

* All the work done by the team should be reported to Team Lead on a daily basis.
* Team Lead should maintain the track of all the work and report to the client on a weekly basis.
* There should be stand-up call set three times a week to know the status of the tasks done and assign new tasks.
* The team members can set up for a team meeting whenever required, to share ideas and improve communication between them.
* In absence of a team lead, a shadow person should be allocated to report the status.
* Include the team lead, project manager and client in the emails, so that they get updated with the status of the project.

# 6.0 Tracking and Control Mechanisms

Techniques to be used for project tracking and control are identified.

## 6.1 Quality assurance and control

QA Team will perform below activities:

* Create Test Strategy and Test Plan which will contain the schedule and number of resources working on it.
* Create and execute test cases
* Report any defect from the test execution in the defect tracking sheet. Test Team will assign a priority to the defect
* Create Traceability matrix to ensure the requirements coverage
* Perform Unit testing, Integration testing and Acceptance testing
* All the Test documents will be reviewed and verified by supervisors
* All the testcases will be reviewed by at least two different team members.
* Test Sign off will not be provided until all the defects are fixed and quality is achieved.
* Testing team has to ensure on time delivery and communicate any unexpected issues or delays to the client.
* Entry and Exit criteria will be mentioned in the test plan or strategy

## 6.2 Change management and control

It is possible that we can have changes in the requirements of the system design, functionality of the system, new regulations have to be implemented etc., In such cases we can follow the change management techniques or approaches and effectively integrate the new changes into the design, development and Test Plan. The effort and time required to implement the changes depends on the new requirement and also existing project plan.

If there is any change in business requirements/critical user journey then we will conduct an impact assessment affecting the development and Test deliverables. The results of this assessment will contribute to the decision making of a change request. However Project members will not own the change request process but only will support in decision making on a need basis. All the change requests would be recorded and managed in a tool provided by client.

The impacted project deliverables will be updated with the change and reviewed with project stakeholders and will be modified for approved changes

Factors which play a vital role on change assessment impact are:

1. Project Deliverables (Development & Testing)
2. Timelines (Development & Testing)
3. Resource Load (Development & Testing)
4. Effort & Costs (Development & Testing)

Change management leading practices that can be followed in the project are:

1. Policies, procedures and standards
2. Request For Change (RFC)
3. Processes for Approval
4. Managing the Deployment etc.

A revision history will be maintained along with the version control for all the changes that are done during the course of the entire project including test and development.

A log history will be used for tracking & measuring purposes for any changes that may occur in the artifacts due to the change arising out of a business requirement or scope addition or de-scoping of the requirement

## 6.3 Tools

Version control system to track changes in the development processes will be used. Each change or development will be tracked with the name, date and what changes and developments are made to the system so far. Examples of version control can be Git, CVS etc., additional Softwares and hardwares and any external tools required for the project will also be included.

HP Quality Center will be used for Test Management & Defect Management which actually provides the features required for configuration management that includes – Version Control, Log History with the Resource Name, Date & Time Stamp of the change done, Contact History, Review Comments Capture Date etc.

# 7.0 Appendix

None.