**Software Project Management Plan**

**Web Application of Commerce Bank**

10/10/2021

**Team Members**

Bollimuntha Jahnavi Sri Kavya

Jayadeep Reddy M

Sai Akhil Palagati

Yeshwanth Varma

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| V1.0 | 10/10/2021 | Initial release |
|  |  |  |

**Document Storage**

This document is stored in the project’s SVN repository at: http://company.com/svn/project-name/docs/spmp.doc.

**Document Owner**

Jahnavi Sri Kavya Bollimuntha is responsible for developing and maintaining this document.**Table of Contents**

[1 Overview 4](#_Toc199052960)

[1.1 Purpose and Scope 4](#_Toc199052961)

[1.2 Goals and Objectives 4](#_Toc199052962)

[1.3 Project Deliverables 5](#_Toc199052963)

[1.4 Assumptions and Constraints 5](#_Toc199052964)

[1.5 Schedule and Budget Summary 6](#_Toc199052965)

[1.6 Success Criteria 6](#_Toc199052966)

[1.7 Definitions 6](#_Toc199052967)

[1.8 Evolution of the Project Plan 6](#_Toc199052968)

[2 Startup Plan 6](#_Toc199052969)

[2.1 Team Organization 6](#_Toc199052970)

[2.2 Project Communications 7](#_Toc199052971)

[2.3 Technical Process 7](#_Toc199052972)

[2.4 Tools 7](#_Toc199052973)

[3 Work Plan 7](#_Toc199052974)

[3.1 Activities and Tasks 7](#_Toc199052975)

[3.2 Release Plan 8](#_Toc199052976)

[3.3 Iteration Plans 8](#_Toc199052977)

[3.4 Budget 8](#_Toc199052978)

[4 Control Plan 8](#_Toc199052979)

[4.1 Monitoring and Control 8](#_Toc199052980)

[4.2 Metrics Collection 8](#_Toc199052981)

[5 Supporting Process Plans 8](#_Toc199052982)

[5.1 Risk Management Plan 8](#_Toc199052983)

[5.2 Configuration Management Plan 8](#_Toc199052984)

[5.3 Verification and Validation Plan 9](#_Toc199052985)

# Overview

## Purpose and Scope

This section provides an executive overview of the project. It explains why the project is being initiated and what can and cannot be expected from project. It may also include any background or contextual information necessary for understanding the project.

The purpose for the project explains the problem or opportunity the project will address. The statement of purpose isn’t a statement of what you are doing (“we plan to automate billing”), but rather why you are doing it (“The purpose of this project is to streamline billing in order to save time, money and resources.”).

Project scope defines the boundaries of the project—what will and won’t be included in the project. Defining project scope helps set expectations regarding what can be expected from the project. The scope definition may also play a role in evaluating requests for changes or new features. Project plans and estimates are based on the scope definition. A request for a change that is outside the current scope of the project can’t be accepted without a change in project scope.

*Partial Example*

A user’s guide and a system installation guide will be provided. One-on-one personalized training isn’t one of the project deliverables.

## Goals and Objectives

Goals and objectives define expected project outcomes. Goals are broad and inspirational. Objectives are narrow and measurable.

Project goals generally related project outcomes to business objectives (reduced cost, increased revenue, improved quality, etc).

A well-worded objective is SMART: Specific, Measurable, Attainable/Achievable, Realistic and Time-bound.

*Partial Example*

Project goals:

1. Establish a mobile porthole for real estate data.
2. Create a mobile porthole that appeals to homebuyers as well as home sellers.

Project objectives:

1. Create a database on homes offered for sale in a local area.
2. Create client software that allows access to the database from mobile devices.
3. Create an interface that home sellers and their agents can use to enter and update home data.
4. Extend mobile access to real estate data in the multiple listing service (MLS).

## Project Deliverables

This section lists the outputs of the project that are delivered to the customer.

*Partial Example*

The following items will be delivered to the customer on or before 1/1/2008:

1. Source code for both the client and server portions of the system.
2. User’s Guide
3. System Administrators Manual
4. Test Plan
5. System test Cases
6. Suite of regression tests
7. Data conversion program for migrating existing data to new database format.

## Assumptions and Constraints

Assumptions are conditions, usually outside the control of the project team, that are taken for granted. Project plans (i.e. estimates) typically depend on certain assumptions being true. Assumptions that turn out to be false, may jeopardize project success. In order to reduce project risk, the project manager may elect to validate certain assumptions as part of the risk management process.

This is also a good place to document verbal promises or assurances given to you.

Constraints are limits or restrictions on freedom. Projects may have technical as well as non-technical constraints. Priorities for schedule and budget can impose non-technical constraints on a project. Restrictions on programming language or delivery platform are examples of technical constraints that limit design and implementation options.

*Partial Example*

Assumptions:

1. The location API works on the test hardware.
2. A senior architect will be assigned to the project during the first 4 weeks.
3. The Unix server and RAID controller can be purchased and delivered by 7/1/2008.
4. Facilities will provide private office space for 3 outside contractors for the duration of the project.

Constraints:

1. The software must run on a Windows Mobile 6 device.
2. The database must be open source.
3. The software must be ready by 1/1/2008.

Note, the following is not a reasonable assumption for inclusion in this section: “We assume that our group has the necessary skills and knowledge needed to complete the project.” This might be something you are taking for granted, but it is not something worth documenting in the project plan. The assumptions you want to list here are those that are outside your control. Once the development team is established, it is their responsibility to possess or develop the skills and knowledge needed to complete the project. If there is a concern that the existing team doesn’t have the skills and knowledge needed to complete the project successfully, add it as a risk and develop a plan for mitigating the risk.

## Schedule and Budget Summary

The schedule summary shows start and end dates for high-level activities ending in major milestones or deliverables. Milestones are major events in the project life cycle that are used to measure progress.

A Gantt chart is an excellent tool for visualizing the start and stop dates of major scheduled activities.

The budget summary shows total project cost, possibly broken down into separate categories for such things as salaries, equipment, travel, overhead, etc.

## Success Criteria

Success criteria spell out what has to happen before the project can be considered a success. Having explicit success criteria serve two purposes. First, during a project success criteria help to focus attention on what is important. Second, at the conclusion of a project (project closure) success criteria are used to assess whether or not the goals and objectives of the project have been achieved.

To be effective in both of these endeavors, success criteria must be defined in a way that is both quantifiable and verifiable.

For more advise on how to define the success criteria for a project, I recommend: *Success Criteria Breed Success*, by Karl Wiegers. It is available on the web.

*Partial Example*

* Total project cost does not exceed 20% of the post-requirements phase estimate.
* All high-priority use cases in the requirements specification are delivered before May 15.

## Definitions

This section should define potentially unfamiliar or ambiguous words, acronyms and abbreviations.

## Evolution of the Project Plan

This section describes plans for updating the project plan throughout the project.

*Partial Example*

Before the start of an iteration, the project plan will be updated to include a schedule of detailed tasks for the upcoming iteration. At the conclusion of an iteration, the project plan will be updated to include the actual effort for each completed task.

Risk mitigation efforts will be evaluated at the start of each iteration. Severe risks will be analyzed and added to the project plan as soon as they materialize.

# Startup Plan

## Team Organization

This section explains project roles and the authorities and responsibilities associated with these roles. Lines of communication, authority and reporting relationships are often shown with an org chart. If development team is known, actual names can be associated with roles.

*Partial Example*

Project Manager: The project manager is responsible for creating the project plan (with input from those doing the work), managing risks, running the weekly team meeting and providing monthly status reports to senior management.

Programmers (3): Programmers are primary responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings.

Build Coordinator: The build coordinator is responsible for setting up, running and distributing the results of the nightly build.

## Project Communications

This section contains the project communications plan. The communications plan describes how information is gathered and distributed.

## Technical Process

This section describes the software development methodology or conventions the team agrees to live by. When following an organization standard process, this section will refer to the standard process and state any deviations that are planned for this project. In the absence of an organization standard process, this section will define planned phases, entry and exit criteria for each phase, major milestones, workflows, and other aspects of the proposed development process.

## Tools

This section specifies the development tools the team will be using to perform their work.

*Partial Example*

* Programming Language – Java
* Version Control – source code and written artifacts will be stored in a Subversion repository.
* Defect tracking – defects and issues will be tracked using Bugzilla.
* Build tools – local and main builds will be done using CodeMake.
* Automated testing – unit tests will be implemented with the JUnit testing framework.

# Work Plan

## Activities and Tasks

A work breakdown structure is an excellent tool for identifying a complete list of tasks.

Depending on the needs of the project, some or all of the following attributes will be recorded for each task:

* Task name
* Task Description
* Owner
* Effort estimate
* Actual effort
* Planned start and stop dates
* Actual start and stop dates
* Dependencies among other tasks

## Release Plan

For day-to-day project management the release and iteration plans (described in the next section) are probably the two most important project management artifacts.

The release plan lists expected completion dates for major milestones and delivery dates of key work products. The project’s technical development process to a certain extent will dictate the choice and timing of milestones and deliverables. For example, projects following the Rational Unified Process will have four major milestones: life-cycle objectives, life-cycle architecture, initial operation capability, and product release.

## Iteration Plans

An iteration plan is a short-term fine-grained plan that shows the tasks to be completed during an iteration.

## Budget

The project budget is the projected cost of the project as a function of time. At any point in the project you should be able to say how much money has already been spent and estimate of the amount of money needed to complete the project.

# Control Plan

## Monitoring and Control

Include in this section plans and procedures for tracking progress and controlling performance. Included here will be the approximate dates of technical as well as managerial reviews. Typically each major milestone or project phase will end in a review.

For projects that don’t have a separate communication plan, this section may also include information on the timing and content of status reports and other project review documentation.

*Partial Example*

Weekly – Team meeting. Project participants report status, progress and potential problems.

3/1/2008 – Critical Design Review. Formal inspection of product architecture.

5/15/2008 – Executive Review. The project manager presents current project status to project sponsor and senior executives.

## Project Measurements

Product and process measures support project management and estimation by analogy. At the beginning of a project, estimates are made for product size, project cost and delivery dates. During a project, progress is tracked with measures of actual effort, integrated lines of code and actual expenditures. Keeping track of estimates and actuals during a project helps to calibrate whatever technique is being used to make estimates. Storing project performance data on completed projects provides a rich source of data for estimating future projects.

*Example*

|  |  |  |
| --- | --- | --- |
| **Phase** | **Measurement** | **Source** |
| Release Planning | Record effort estimates for product features | Mgr |
| Iteration Planning | Record effort estimates for scheduled tasks  Update effort estimates for product features  Update estimated dates in release plan | Mgr |
| Iteration Closeout | Record actual effort for scheduled tasks  Record actual effort for product features  Record LOC count for modules written | Mgr/Pgr |
| System Test | Record the rate at which errors are found. | QA |
| Project Closeout | Archive project performance data in process database. (See process database definition for a list of measures to record.) | Mgr |
| Ongoing | Record defects found from integration testing through first year of release.  Assign each defect to one of the following categories: blocker, critical, major, minor or trivial. Keep track of the state of each defect: open, assigned, fixed, closed. | Mgr/Pgr/QA |

# Supporting Process Plans

## Risk Management Plan

Identify technical and managerial risks. Prioritize risks. Consider the probability of each risk turning into a problem and the likely consequences. For the highest priority risks, what actions will be taken to minimize the probability of the risk turning into a problem and the resulting consequences? What are the contingency plans for selected risks that do become a problem? Identify processes for monitoring risks and updating the risk management plan.

## Configuration Management Plan

Configuration management plans for this document and other baselined work products including review procedures and change management procedures.

*Partial Example*

1. All work products will be stored in a centralized CVS repository running on a central server.
2. The naming convention for documents will be: NNN-VVV.suffix where NNN is a mnemonic that reflects the function of the document, VVV is a 3 digit version number, and 'suffix' is the standard/normal suffix for the document type. For example, the second version of the requirements document created as a Microsoft Word document might be labeled: REQ-002.doc.
3. All project (work products) items (documents, source code, test cases, program data, test data, etc) will be stored in the CVS repository but not all will be under change control (subject to formal change control procedures.) Only the system requirements, project plan and source code will be baselined and under configuration control.
4. Items that are subject to change control will be considered baselined after a group review at the end of the life cycle phase during which they are created. Baselined here means that the product has undergone a formal review and can only be changed through the prescribed change control procedures.
5. The change control procedure once a product is baselined is: (1) anyone wanting to make a change to a baselined item sends an email to the rest of the group describing the change, reason for the change, expected impact, and timeline for integrating the change. (2) if no one responds to the group within 2 days with a reason for why the change request shouldn't be permitted, it will be considered accepted and the person proposing the change may proceed with the change. If anyone does object to the change, the reason for objecting will be discussed at a meeting where everyone is invited to attend and voice their opinion. At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.
6. Including a change history with all documents is encouraged but only required for baselined documents. The change history should be at the front of the work item and include: (1) the name of the person making the change, (2) brief description of what has changed, (3) reason for the change, and (4) the date the change was integrated.

## Verification and Validation Plan

The verification and validation plan defines what actions are being taken to assure the quality of the development process and resulting software products.

*Partial Example*

The Verification and Validation plan is specified as a separate documented located in the version control system at: http://company.com/svn/project-name/docs/VandVPlan.doc

## Product Acceptance Plan

The product acceptance plan defines what is acceptable in terms of product quality and product functionality. Acceptance criteria should be objective and measurable. Note product success is one aspect of project success. Teams wanting to establish a clear understanding of what will be considered acceptable project performance may want to define a more general plan for project success that includes quantitative goals for delivery date, cost, etc.

*Partial Example*

The Verification and Valida