***SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)***

**1.0 Introduction**

This section provides an overview of the software engineering project.

* 1. **Problem statement**

The purpose of the new hotel booking system is to replace the old and outdated hotel booking system. The old system is outdated, slow, inefficient and the employees have to put the customers on hold. The employees also have to work with multiple screens to pull up customer information about the reservation in order to cancel or rebook.

The new system will allow for a smooth experience for both the customers and employees. It will allow for both customers and employees the ability to book, modify, & cancel reservations efficiently while allowing for implementation of a rewards system and outside bookings, giving customers more flexibility and value to their stays. ~~The new system will allow for outside bookings and allow the implementation of a reward system~~. Also employees will be able to view metrics of various customers and manage the various administrative functions of the hotel

**1.2 Project scope**

**1.2.1 Inclusions**

Inputs

* Login credentials
* Booking Dates, Times
* Hotel Location
* Customer Information (e.g. name, phone, address)
* Reservation number
* Room Preferences

Processing functionality

* Verifying login credentials
* Checking room availability
  + adding to waitlist
* Processing bookings
* Processing payment
* Calculating Customer Rewards
* Calculating metrics (customer tracker)

Outputs

* ~~Metrics: popular booked locations, revenue, rooms most booked~~
* Customer reward points
* History of Customer activity
  + Login/access account
  + check in/check out
  + reward points gained and spent
* Employees review reservations and/or cancellations
* Response to third party website
* Summary Reports
  + Popular booked locations
  + Revenue vs Expenses
  + Rooms most booked
  + Occupancy Rate

**1.2.2 Exclusions**

* n/a

**1.3 Major software functions**

* Access the system
  + Create a new user account / ID
  + ~~Store username in database~~
  + ~~Store password in a secure manner in database~~
  + ~~Give users a unique ID to track throughout the system~~
  + ~~Check if credentials are correct~~
  + Login/Log-out the user
  + Check account information (separate views for users and admins)
* Check-in Check-out
  + When a customer arrives or leaves check in or out
* Reservation
  + Display Rooms and check room availability
  + Create, make changes to, or cancel reservations
  + Create a waitlist if all the rooms are reserved
* Rewards
  + History of Rewards activity
  + Manage and view rewards
  + view and book upgrades using rewards obtained based on type of room and length of stay
* Hotel Administration
  + Manage the major physical functions of a Hotel
  + Track metrics/Summary Reports (for bookings and number of visits to the site)
  + ~~Use the user’s unique ID~~
    - Most booked Rooms
    - Most booked hotels
    - Track number of reward points
    - Most visited Customers.
    - Revenue from each stay

**1.4 Performance/Behavior constraints**

* ~~Performance:~~
  + ~~Load times for any page of information should not take any longer than five (5) seconds~~
* ~~Behavior:~~
  + ~~Guest users should not be able to manage other guest users, only administrators should have this privilege~~
  + ~~Guests should not be able to view other guests’ metrics or bookings~~

**1.5 Management and technical constraints**

**1.5.1 Management constraints**

* Drop dead delivery date: August 19, 2020
* Lots of other courses and work. (Time management)
* Collaborating remotely
* Limited Meet time as a group

**1.5.2 Technical constraints**

N/A

**2.0 Project Estimates**

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

**2.1 Historical data used for estimates**

Describes the historical data that is relevant to the estimates presented. This is, what historical work can be used as the basis for your estimations (e.g. prior classwork projects, work projects)

* Peter
  + Deployment of upgraded OS projects
  + Created and developed a full game of euchre, including bots using C++
  + developed a machine learning algorithm to predict forum posts using C++
  + Developed Yahtzee game using C#
  + created database management system using winforms in Visual Studio
* Monier
  + Endurance Run Management Projects (Bosch)
  + Mitsubishi ER project testing Gen5 Radar (Bosch)
  + Developed business plan for QuickPark parking indication system
* Khalid
  + P703 cluster HMI project C++ (Denso)
  + Built an 8-bit Microprocessor using VHDL. (School)
  + Developing a security system using facial recognition and Vitis AI Software.
* Jeffrey
  + Health Decisions, Inc
    - Created a customer-facing web portal, upgraded and maintained existing databases. Developed in C#, MS SQL, and created HTML / CSHTML pages with custom CSS styling
  + Autodesk, Inc.
    - Developing for desktop Data Management and Project Lifecycle application (Autodesk Vault), working with C# and C++ with Common Language Infrastructure (CLI) to interop between them, as well as MS SQL

**2.2 Initial Estimate**

Based solely on 1.0 and 2.1 details provided.

**2.2.1 SPMP Completion Estimate**

* 30 Hours

**2.2.2 Overall project estimate**

**2.2.2.1 Line-of Code Estimate**

* 3500 lines of code

**2.2.2.2 Function Estimate**

* 15

**2.2.2.3 Tasks Estimate**

What are high level tasks to complete (this is not a project plan) but tasks to think about for this estimate

* Filling out Software project management plan
* Completing and constructing a complete ER diagram
  + identifying the attributes associated with each use case
* Use case summary
  + identifying use cases and the flows associated with each use case
* Use Case Document specification
  + describing what each of the flows are doing
  + listing the actors involved in each use case
* Analysis/Requirements gathering
* Building and testing software

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

1000 hours

 **2.3 Estimation techniques applied and results**

A description of each estimation technique and the resultant estimates are presented here. **DO NOT COMPLETE THIS SECTION UNTIL REMAINDER OF SPMP IS COMPLETED AND REVIEWED!**

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

estimated total lines of code/ estimated number of hours = lines of code per hour

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

3000 estimate lines of code

1000 estimated hours for the entire group

3000/1000 = 3 lines of code per hour

**2.3.3 Estimation technique 2 – function points**

|  |  |  |
| --- | --- | --- |
| Category | How Many? | Complexity (1-10) |
| External inputs (EI) | Customer:   1. Booking Dates 2. Hotel Location 3. Room 4. user name 5. password 6. name 7. address 8. phone # 9. # of guests   Employee:   1. Booking Info 2. username 3. password 4. cancel   User:   1. Is Employee?   Total: 14 | 3 |
| External outputs (EO) | Customer:   1. Availability, 2. Booking # 3. Name 4. Booking Dates 5. Hotel room 6. Activity logging   Hotel:   1. Availability 2. Metrics 3. Reservations 4. Booking# 5. Hotel 6. Location of customer 7. Summary Reports 8. Most visited customers 9. Reward Points   Total: 15 | 9 |
| External Inquires (EQ) | 1. Customer username 2. Customer address 3. Customer phone # 4. Booking Date 5. Booking # 6. Hotel Room # 7. # of times customer logged in 8. # of times a room was booked 9. Username taken? 10. Is user admin? 11. Reward Points | 8 |
| Internal Logical Files (ILF) | Hotels   1. Flat file of hotels, ratings, locations 2. Table of customer and employee activity tracking/logging 3. Table of users: Customer & Employee information 4. Table of rewards types and package types    1. History of customer rewards 5. Table of reservations 6. Table of rooms of each hotel location | 5 |

**2.3.4 Estimate for technique 2 – function points**

14\*3+15\*9+11\*8+ 7\*5 = 42+135+88+35= 300 function points

4 hours per point

300\*4 = 1200 hours

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

Tables or equations associated with lines of code estimation technique m are presented. Re-estimate effort based on actual tasks defined in completed SPMP.

1. Project Approval Process
   1. Create Project Charter Task
      1. Have every team member sign off on charter
   2. Create SPMP
      1. Continuously update relevant parts of the SPMP
   3. Create Use Case Summary
      1. Define the use cases
      2. Define the flows within each of the use cases
   4. Create Use Cases
      1. List the order of the flows in a use case
      2. develop an activity diagram and identify the the conditions under which the use case will happen
      3. Explain the various actors involved in a use case
2. Analysis Artifacts Task
   1. Create DFD Diagrams
   2. Create State Diagrams
   3. Create Communication Diagrams
   4. Create Robustness Diagrams
   5. Create Traceability Matrix
   6. Create Basic UI elements
   7. Create Testing Strategies
3. Complete Building Application Task
4. Testing Process
   1. Complete Unit Testing Task
   2. Complete System Testing Task
5. Release Process
   1. QA Task
   2. Presentation Task

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

1500

**2.4 Reconciled Estimate**

For final time we took an average of the times calculated in the three different techniques, giving an estimated time for the project duration

Regarding cost, we took the salary of a software engineer on our team, multiplied by the number of team members, times the estimated hours for the project

Final Time: 1233.33 hours

Final Cost: $15 / hr \* 4 resources \* 1233.33 hours = $73,999.80

**2.5 Project Resources**

* Wednesday Meetings with Client
* Textbook, Powerpoint slides, and lectures
  + Notes taken and shared with the group
* Team members
  + Khalid, Peter, Monier, Jeffrey
* IDE/Compiler/Libraries
  + Visual Studio 2017 - 2019 IDE
  + Standard C++ libraries
* Data storage
  + SQL Databases
* Laptop/Personal Device
  + OS: MacOS, Windows 10
* Phones
  + Communicate with team members about assignments and weekly group meetings
  + Notify of any significant changes and/ or additions to the tasks

**3.0 Risk Management**

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

**3.1 Project Risk Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Mitigation** | **Contingency Plan** |
| Time management | ~~Medium~~ High | High | Split up the work, weekly check ins, work consistently | Focus on most important aspects of the project |
| Functionality | Low | High | Thorough testing as we go through. | Try to get the main components to work first. (Prioritize) |
| Losing a team member temporarily | ~~Low~~ Medium- High | Medium-High | To know the status of our team members and using github and centralized notes to have access to their work. Take notes for absent team members if necessary. | Temporarily reassigning work as needed depending on the length of absence of team member |
| Need to re-work milestones | High | High | Plan on doing and assigning extra work to team members to fix the issues with the previous milestone. Make sure previous versions are kept intact . Talk and meet with team members and Professor as necessary to eliminate confusion | Prioritize the documents that are due first and those that have the most impact to the project. Ask for an extension for the assignment |
| Having a few days until Milestone is due | Medium | High | Don’t procrastinate. | Have team members work individually and IM for required details, so we’re not dependent on others when unnecessary. |
| Team Member Lost | Medium | High | Keep written records of role responsibilities and knowledge gained during time in team for future references. Keep public notes if possible. | Transfer knowledge and role responsibilities by means of keeping written record, so new member can be up-to-speed as quickly as possible. |

**3.2 Overview of Risk Mitigation, Monitoring, Management**

How will Risk Mitigation, Monitoring and Management be handled. Communication plan, when to apply the mitigation strategy, etc.

* Apply mitigation strategy as soon as risk appears more apparent
* Groupchats-notify if going to be absent or falling behind in project work
* Github- back our code and other documents
* Centralized notes and documents - ensure we each have access to project-critical information even if other members are unavailable
* Weekly Meetings- check in on project work status of each team member

**4.0 Project Schedule**

This section presents an overview of project tasks and the output of a project scheduling tool.

**4.1 Project task set**

The process model, framework activities and task set that have been selected for the project are presented in this section.

Process Model: Waterfall

**Recurring Tasks**

* Weekly Team meetings
  + Complete tasks for various milestones and discuss future plans for the project
* Weekly status report
  + Relay back to Professor Steiner what is going on with the group and the status of the various milestones and assignments
  + Identify updated risks to tasks and list what is to be done to remedy the situation
* Coordinating weekly meetings
  + Rotating who is leading each meeting

**Framework activities and Tasks:**

Pre-Project work

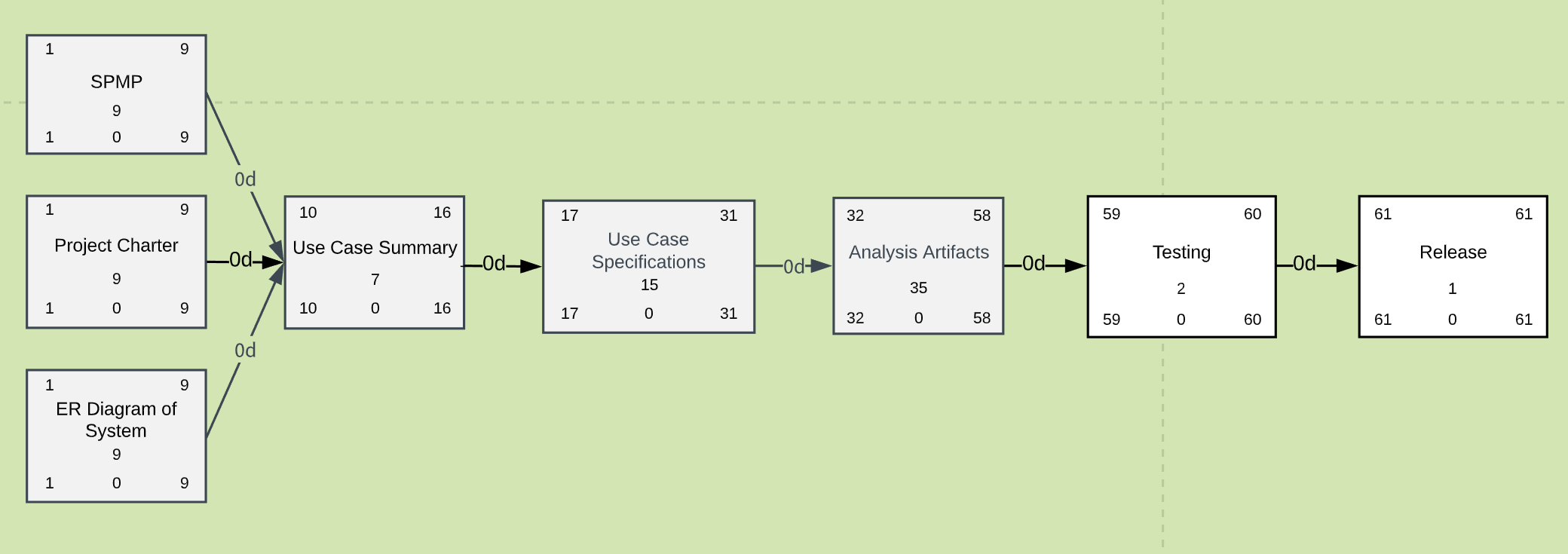
* Sign off on project charter
* Fill out and determine the tasks and timeline on SPMP
  + Introduction
  + Project Estimation
  + Risk Management
  + Project Schedule
  + Staff Organization
  + Tracking and Control Mechanisms
* Develop and create the Complete ERD
  + identify all the entities and independently determine if each one has a relationship to the other

Use Cases

* Create Use Case Summary
  + Develop and create a use case diagram to represent broad use cases that entail many types of flows and the primary and secondary actors interacting with the system
  + List all the use cases and the flows that fall under each use case but do not go into detail about the flows
* Create Use Cases Specification Documents
  + Determine the Trigger
  + List and detail the Pre and Post Conditions
  + Detail the flows and activities associated with a particular use case
  + List and determine the Main and alternate flows of the use case
  + Create swimlane diagrams

* Create Analysis Artifacts
  + Determine communication between the use cases for all data diagrams
  + Fill in details for Robustness Diagrams
  + Use Robustness Diagrams as base for Class and Communication diagrams
  + Create State Diagrams and DFD Diagrams
  + Create Traceability Matrix and User Interfaces
* Design
* Testing System
* Release To Customer

**4.2 Task network**

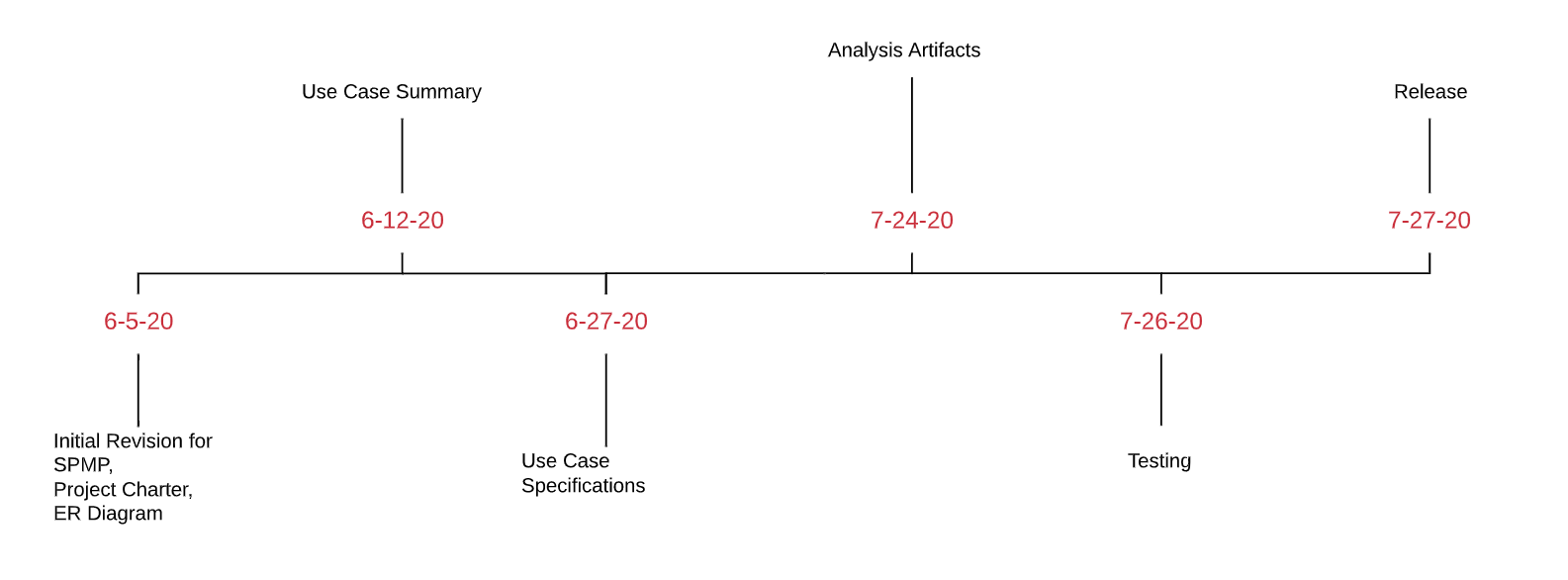


Calculation: Tasks start date: 5-27-2020, Tasks Presentation Date: 7-27-2020, Duration: 61 days

JF 6/21/2020: Since our due-dates for Use Cases have been shifted, let’s update the Timeline and Task Network charts to reflect this. Since only the Use Cases have been pushed back, those are the only durations that’ll change right now.

**4.3 Timeline chart**

A project timeline chart is presented. This may include a timeline for the entire project or for each staff member.



**5.0 Staff Organization**

The manner in which staff are organized and the mechanisms for reporting are noted.

**5.1 Team structure**

* Team Lead - Peter
  + Reports back to Professor on how the team is doing and managing
  + Assigns necessary work to self and individual team members for completion
* Software Developers - Khalid, Monier, Jeffrey, Peter
* Meeting leaders will rotate each time

**5.2 Management reporting and communication**

Mechanisms for progress reporting and inter/intra team communication are identified.

* Group chats
* Weekly Team meetings
  + Discuss and design artifacts for the upcoming tasks due for the milestones
  + Assign to team members artifacts to work on and report back to team after finishing it for review
* Class meetings
  + View and watch lecture recordings to understand the tasks and pieces of the milestones
* Meetings with Client (Professor Tom Steiner)
  + every wednesday morning to discuss business requirements and use case specifications of the hotel management system

**6.0 Tracking and Control Mechanisms**

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

**6.1 Quality assurance and control**

An overview of SQA activities is provided. This is an outline at this point and will be used to create a detailed plan later in the project.

* Unit Testing
* Testing individual functionality
* System-Wide Testing
  + ~~User Accounts~~ Employee/Customer individual account & functionality testing
  + Administration

**6.2 Change management and control**

An overview of SCM activities is provided on how changes will be handled – communication of a change, how decisions made of approval, defer or reject change request.

* Identify SCI (Software Configuration Item-Artifacts) objects
  + Basic Objects
  + Aggregate Objects
* Change Control
  + Evaluating the new added business, technical requirements, potential side effects, configuration items, and overall impact of new changes that come in and managing them effectively to determine if they shoud be incorporated.
* Version Control
  + Managing various versions of a particular artifact, making sure that it is traceable to the very beginning and to the final version
* Impact Management
  + Understanding interdependencies and how they related to the timeline, quality, and cost of the project
* Configuration Audit
  + Determine consistency and accuracy of the SCI’s prior to proceeding to the next milestone
* Status Reporting
  + Keeping track of each of the SCI- What happened? Who did it? When did it happen? What else will be affected?
  + Report and document changes to SCI

Baselining: Once a particular task has been submitted to canvas, that version of the document becomes the baseline and cannot be changed without the approval of other team members.

* Google Drive
  + Version tracking via google docs, sheets, and slides
* Github
  + Create branches for each team member
  + Review and test branches for pieces of functionality prior to merging with master branch
    - Unit testing of functionality and functions

**6.3 Tools**

What tools will be used to control access and versioning of artifacts?

* Github
  + Source Code, Unit testing, and System testing
* Google Drive
  + Artifacts
    - Pre-Project
      * SPMP
      * Entity-Relationship Diagram
      * Project Charter
    - Use Cases/ Business Requirements
      * Use Case Summary
      * Use Case Specification documents
    - General
      * Time tracking
      * Meeting Notes from bi-weekly meetings
      * Notes of meetings with Client for business requirements
  + Folders containing old and new versions of documents labeling the version of the group of documents

**7.0 Appendix**

Supplementary information is provided here.

**Ver 2. (on separate document- ERD version 2)**