

SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

CIS 375 - Steiner

Last Updated: 6/11/18

Version 1.1

Team "2"007 Britney Spears:

***Joshua Attard, Veeram Hirekhan, Hassan Mehdi,
Maram Mohammed, Allison Ramasami, Srinivas Simhan***

Table of Contents

1.0 Introduction	4
1.1 Problem statement	4
1.2 Project scope	4
1.2.1 Inclusions	4
1.2.2 Exclusions	5
1.3 Major software functions	5
1.4 Performance/Behavior constraints	6
1.5 Management and technical constraints	6
1.5.1 Management constraints	6
1.5.2 Technical constraints	6
2.0 Project Estimates	8
2.1 Historical data used for estimates	8
2.2 Initial Estimate	8
2.2.1 SPMP Completion Estimate	8
2.2.2 Overall project estimate	8
2.2.2.1 Line-of Code Estimate	8
2.2.2.2 Function Estimate	8
2.2.2.3 Tasks Estimate	8
2.2.2.4 Total overall project time estimate in hours of effort	9
2.3 Estimation techniques applied and results	9
2.3.1 Estimation technique 1 – lines of code	9

2.3.2 Estimate for technique 1 – lines of code	9
2.3.3 Estimation technique 2 – function points	9
2.3.4 Estimate for technique 2 – function points	12
2.3.5 Estimation technique 3 – process/task	12
2.3.6 Estimate for technique 3 – process/task	14
2.4 Reconciled Estimate	14
2.5 Project Resources	14
3.0 Risk Management	15
3.1 Project Risk Table	15
4.0 Project Schedule	17
4.1 Project was task set	17
4.2 Task network	17
4.3 Timeline chart	22
5.0 Staff Organization	23
5.1 Team structure	23
5.2 Management reporting and communication	23
6.0 Tracking and Control Mechanisms	24
6.1 Quality assurance and control	24
6.2 Change management and control	24
6.3 Tools	24
7.0 Appendix	25

1.0 Introduction

This is the Software Project Management Plan (SPMP) for Team "2"007 Britney Spears. This project is being done for Software Engineering I (CIS 375 - T. Steiner), and is being undertaken by Joshua Attard, Veeram Hirekhan, Hassan Mehdi, Maram Mohammed, Allison Ramasami, and Srinivas Simhan. The project involves automating Chocoholics Anonymous data processing and storage to speed up their current manual processes.

1.1 Problem statement

We are creating a database to store member, transaction, and provider information, so that the database can be automatically managed and updated. The information can be accessed at any time by the ChocAn Data Center, the Provider Directory, the member/provider terminals at the ChocAn facility, and Acme Accounting Services.

1.2 Project scope

1.2.1 Inclusions

The data store includes:

- Customer Information
 - Name, ID number, Address, Phone Number, Email Address
- Employee Information
 - Name, ID number, Email Address
- Service Information
 - Service Code, Provider ID Number, Provider Name, Provider Phone Number, Processed Status

Software Description:

- Inputs
 - Member, provider, and employee information in a centralized data store
- Processing
 - Allow operators to add, remove, or modify the customer, service, and employee information through an interface.
 - Allow providers for ChocAn to enter service information remotely through a ChocAn terminal.

- Allow providers for ChocAn to request a provider directory, which contains a list of all services and service codes.
- Generate reports telling members what services they received along with processing status, providers what services they have billed to ChocAn, and managers who has billed them with services that week along with any services that have not been processed yet. These reports are sent via email to the appropriate recipients.
- Outputs
 - Reports for members, providers, and managers.
 - Provider directory.

1.2.2 Exclusions

None.

1.3 Major software functions

- Link database that allows terminal to have access to customer and provider information.
 - Member name, Member #, Member Address
- Store provider/service data in database.
 - Service Name, Service Code, Fee
- Provide menu for operators to select whether they would
 - Like to add, delete, or modify database records for providers or members.
 - Allow operators to search via each column in the database.
 - Actually make changes to the records.
- Provide menu for providers to either order a provider directory or bill a service to ChocAn.
 - Create a provider directory via information from the database.
 - Allow the provider to enter the information to bill the service, and update the database accordingly.
 - Information entered includes Provider Name, Member Information, Comments, Date of Provided Service, Service Code, Member Name, Member #
 - Provider directory contains Service Name, Service Code, Fee for all services.
- Generate reports for service providers, members, and managers.
 - Email reports to members, providers, and managers.

- Member report: member name, number, address, city, state, zip code, and a list of dates of services, provider names, service names for every service the member received through ChocAn.
- Provider report: provider name, number, address, city, state, zip code, and a list of dates provided and ordered, member name and number, service code, fee for all , total services provided and total fee.

New Functionality:

1. Reports are actually emailed to their respective recipients every Friday instead of just being an output.
2. Services now have a processing status that is edited by Acme. Managers' reports now contain outstanding services that have not been processed yet for any time period, and members' and providers' reports also contain the processing status of each service. When a service has been processed, the provider and member involved in the service receive an email notifying them of the change.
3. All data in the data store will be stored in an online database. All programs will access this if they need to do anything with the data store.

1.4 Performance/Behavior constraints

- Reports have to be sent on time and to the correct email address
- Database has to be accessible 24/7

1.5 Management and technical constraints

1.5.1 Management constraints

- Team members have jobs and other classes they have to allocate time towards.
- Project has to be completed by August 20th of 2018

1.5.2 Technical constraints

- Team Members have limited experience with databases
- Will not be able to calculate this until we start on the project
- Limited experience as we are college students

2.0 Project Estimates

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

2.1 Historical data used for estimates

- Prior assignments done by team members used to estimate time
- Prior assignments done by team members used to estimate lines of code and number of services processed per week
- Prior work projects done by team members to estimate quality
- Percentage of employee time spent working each day
- Number of services processed per week

2.2 Initial Estimate

Based solely on 1.0 and 2.1 details provide.

2.2.1 SPMP Completion Estimate

15 hours (collectively as a team)

2.2.2 Overall project estimate

2.2.2.1 Line-of Code Estimate

2000 LoC

2.2.2.2 Function Estimate

10 function points

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

- Paperwork and documentation (12 hours)
- Create database and backend (12 hours)
- Create front end for service providers to enter and retrieve data and interact with database (40 hours)
- Link software to generate reports on services and customers (20 hours)
- Design terminals (100 hours)

- Create software to bridge terminals with data center (20 hours)

2.2.2.4 Total overall project time estimate in hours of effort

- Total = Tasks Estimate
- 204 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

To estimate the amount of lines of code, we have to use the formula.

$$LOC_{total} = FP * LOC$$

Component	FP	SLOC	TLOC	KLOC
Database	43	110	4730	4.73

2.3.2 Estimate for technique 1 – lines of code

Estimate in hours generated for lines of code technique is calculated with the following formula: Total = KLOC / Average(SLOC), where SLOC is the estimate LOC completed per person.

$$Total = 4730/65$$

$$Total = 73 \text{ hours}$$

2.3.3 Estimation technique 2 – function points

Function point estimation for External Inputs
--

Provide menu for operators to select whether they would like to add, delete, or modify database records.
--

Store provider/service data in database.
--

Function point estimation for External Outputs

Allow user to enter provider number, then cross-reference with database.
--

Function point estimation for External Inquiries

Link database that allows terminal to have access to customer and provider information.

Provide menu for providers to either order a provider directory or bill a service to ChocAn.
--

Display menu for operators to modify data.
--

Function point estimation for External Interface Files

Integrate queries into program to obtain information.

Design SQL query to add the entered service to database.

Function point estimation for Internal Logical Files

Generate reports for service providers, members, and managers.

Automatically run the program on a given interval.

Component:	Complexity		
	Low	Average	High
EL	3	4	6
EO	4	5	7
EQ	3	5	6
EIF	5	7	10
ILF	7	10	15

Component:	How Many	Complexity
EL	2	3

EO	1	4
EQ	3	3
EIF	2	5
ILF	2	7

$2*3+1*4+3*3+2*5+2*7= 43$ **function points**

2.3.4 Estimate for technique 2 – function points

One function point is roughly 8 hours of work, so the project estimate is $43*8=344$ **hours** of work.

2.3.5 Estimation technique 3 – process/task

Task Name	Duration
Communication	50 hours
Determine project scope	7 hours
Inform stakeholders	15 hours
Identify requirements	24 hours
Identifying communication with Team Members	4 hours
Planning	179 hours

Identifying resources	100 hours
Identifying risks	4 hours
Construct Timeline	36 hours
Rough draft models	39 hours
Design	95 hours
Looking at specifications of where everything should be	25 hours
constructing the design	70 hours
Coding	350 hours
Develop code	230 hours
Debugging code	90 hours
Reviewing functionality	30 hours
Testing	270 hours
Developing unit test for functions	100 hours
Integration testing	80 hours
Manual testing	90 hours
Deployment	72 hours
Determine final deployment strategy	24 hours
Develop deployment methodology	24 hours

Deploy/validate software	24 hours
--------------------------	----------

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

2.3.6 Estimate for technique 3 – process/task

Estimate in hours generated for process/task technique is 1016 hours.

2.4 Reconciled Estimate

Combining 2.3.2(73 hours), 2.3.4(344 hours), and 2.3.6(1016 hours), our final average estimate is 478 hours. Utilizing the task estimates from these sections, 2.3.4 is probably the most accurate estimate due to understanding what needs to be done and how much time each task can possibly take in order to finish the project. Based off of the function points and its multiplier, we can assume this is the best estimate due to it having a conversion from function points to hours, therefore telling us how long this could possibly take to finish.

The final cost of the project is estimated at \$576, this is based off of the lines of code, software being used, equipment for the software, time taken to finish the project and personal cost. Since our presentation for our project is on August 13th, we have approximately 61 days to finish the complete the project.

2.5 Project Resources

Resources needed for this project will be team members (Josh, Allison, Srini, Hass, and Veeram) who will be doing their assigned parts of the project. We will be needing a couple of computers that has the software Visual Studio installed and functioning. We also use google inorder to make share documentation with the entire group via google hangouts, and google docs. We have a app on our phones/computers called GroupMe so if we need to connect with the group about an important subject then we can talk about it at a later time.

3.0 Risk Management

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

3.1 Project Risk Table

Risk	Probability of Risk H/M/L *	Potential Impact of Risk C/H/M/L *	Risk Mitigation Strategy and Contingent Action(s)
Team member dropping course	Low	High	Mitigation: Communicate to team and reassign tasks and/or convince members not to drop Contingency: Redistribute immediate responsibilities to another team member until next team meeting
Team member gets sick/injured	Medium	High	Mitigation: Communicate with the team and reassign tasks, therefore making it easier on the team member to do quality work while recovering Contingency: Reevaluate responsibilities to the remaining team members until the current sick/injured team member is mostly healthy.
Team member doesn't do what they were assigned to do	Low	High	Mitigation: Communicate with the team and reassign tasks. Assign tasks and finish them ahead of time so failure to complete something has less of an impact on the project. Contingency: Reevaluate responsibilities to the remaining team members, and have a discussion with the team member that couldn't complete their work.
SQL Server takes longer than expected to integrate into C++	Medium	High	Mitigation: Thorough research into SQL Server and the API responsible for communicating with it (SQLAPI++) early into the project. Contingency: Drop the SQL database entirely and store/work with the database data as .csv files.

* Critical /High / Medium / Low

3.2 Overview of Risk Mitigation, Monitoring, Management

Risks will be discussed formally at every Sunday meeting. Each of the risks will be re-evaluated for the probability and impact of that risk happening at this point in time. If the probability and impact are deemed to be sufficiently high by the group, the mitigation plan will be put into place. If the risk actually happens and is now a problem, the news of this will be communicated through GroupMe so the contingency plan can be put into place as soon as possible. If a mitigation plan or contingency plan is put into place, the status of the risk/problem will be evaluated daily through GroupMe (are we on-track despite the risk/problem, do we need to put more effort into resolving it, etc.) until the risk/problem is resolved.

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.

Task Set:

- A. Design full relational database
- B. Implement the database in SQL Server
- C. Allow user to enter provider number, then cross-reference with database.
- D. Create interface to allow provider to enter service information.
- E. Design SQL query to add the entered service to database.
- Z. Add database functionality to provider program
- F. Determine file type and format for downloadable provider directory.
- G. Design SQL query to obtain data for provider directory.
- H. Integrate into C++ program to let user download a provider directory.
- I. Display menu for operators to modify data.
- J. Have C++ pull from SQL for the operator program.
- K. Determine general report file type and formatting.
- L. Determine SQL queries needed to generate reports.
- M. Code C++ program to generate reports.
- N. Integrate queries into program to obtain information.
- O. Automatically run the program on a given interval.

We will be using an agile process model.

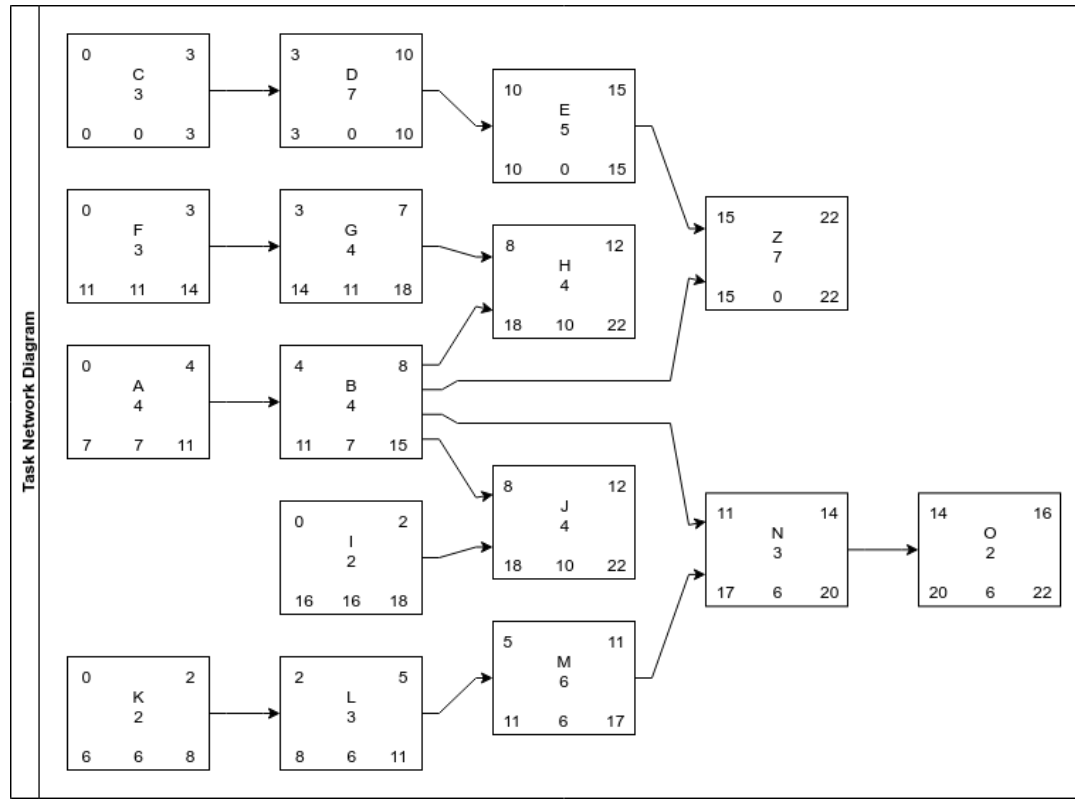
4.2 Task network

Project tasks and their dependencies are noted in this diagrammatic form.

Task	Duration (estimate) (days)	Predecessor Tasks	Successor Tasks	Start Date (ES)	End Date (EF)	Slack Time	Person Responsible (one person)	% Complete
A: Design full relational database	4	--	B	0	4	7	Allison Ramasami	0
B: Implement the database in SQL Server	4	A	Z, H, J, N	4	8	7	Hassan Mehdi	0
C: Allow user to enter provider number, then cross-reference with database.	3	--	D	0	3	0	Josh Attard	0
D: Create interface to allow provider to enter service information.	7	C	E	3	10	0	Veeram Hirekhan	0
E: Design SQL query to add the entered service to database.	5	D	Z	10	15	0	Maram Mohammed	0

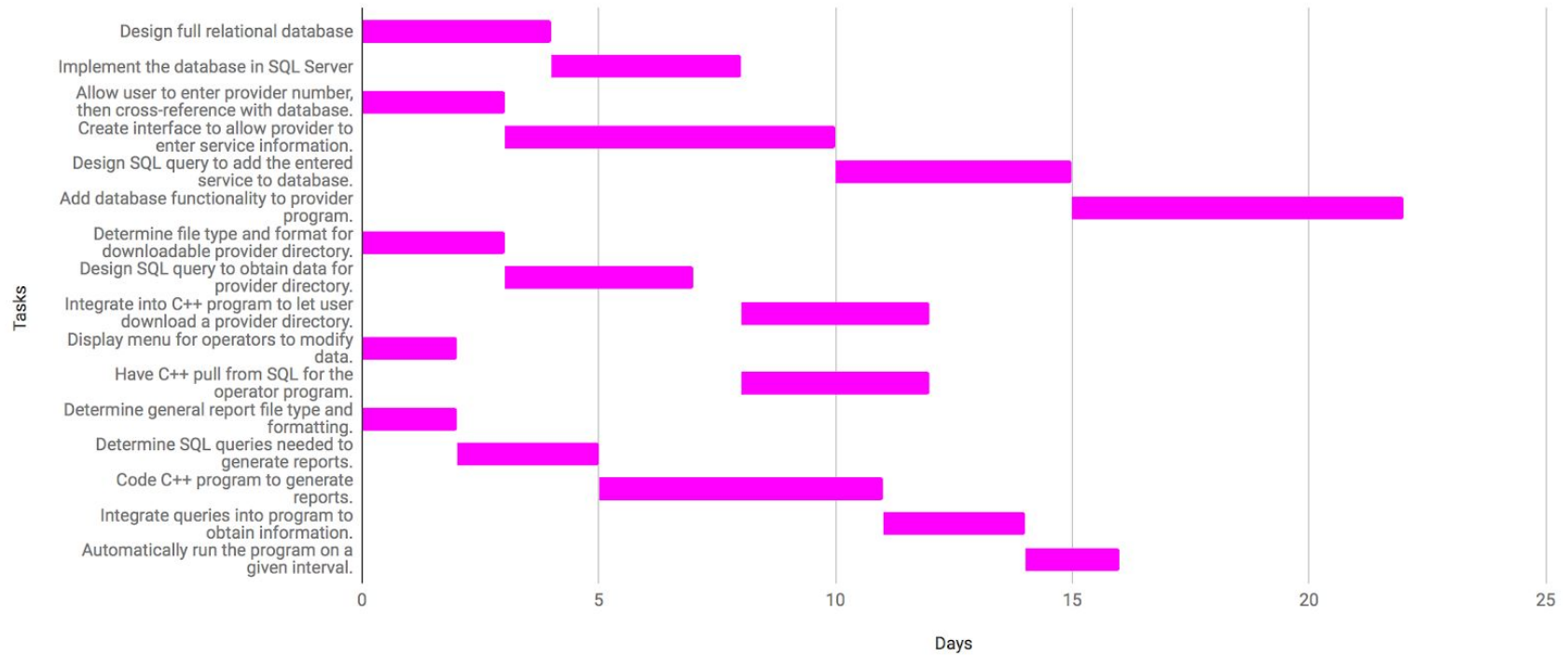
Z: Add database functionality to provider program	7	B, E	--	15	22	0	Veeram Hirekhan	
F: Determine file type and format for downloadable provider directory.	3	--	G	0	3	11	Josh Attard	0
G: Design SQL query to obtain data for provider directory.	4	F	--	3	7	11	Srinivas Simhan	0
H: Integrate into C++ program to let user download a provider directory.	4	B, G	--	8	12	10	Allison Ramasami	0
I: Display menu for operators to modify data.	2	--	J	0	2	16	Maram Mohammed	0
J: Have C++ pull from SQL for the operator program.	4	B, I	--	8	12	10	Maram Mohammed	0

K: Determine general report file type and formatting.	2	--	L	0	2	6	Hassan Mehdi	0
L: Determine SQL queries needed to generate reports.	3	K	M	2	5	6	Allison Ramasami	0
M: Code C++ program to generate reports.	6	L	N	5	11	6	Josh Attard	0
N: Integrate queries into program to obtain information.	3	B, M	O	11	14	6	Veeram Hirekhan	0
O: Automatically run the program on a given interval.	2	N	--	14	16	6	Srinivas Simhan	0



4.3 Timeline chart

Timeline



5.0 Staff Organization

5.1 Team structure

- Team Lead
 - Organizes team meetings
 - Writes-up weekly reports and submits them
 - Consolidates all due reports and submits them
- Team Members
 - Actively edit and review any items related to the project

5.2 Management reporting and communication

- Weekly Sunday meetings, either on google hangouts or in person. Depending on the workload additional meetings held during the weekdays, usually on google hangouts
- Google Drive Folder that gives every member of the team access to each document
- Group chat using the app GroupMe that keeps the team in constant communication

6.0 Tracking and Control Mechanisms

6.1 Quality assurance and control

During weekly meetings each recently worked on item will be reviewed and discussed by everyone on the team. Before each item is finalized, the team member informs the group chat that the item is completed and ready for review. Every other team member then reviews the item and gives their approval to finalize the item or discusses anything they found problematic with the rest of the team. Each stage of software development will undergo testing, and newly discovered risks will be re-evaluated and factor into our planning.

6.2 Change management and control

Any possible change to the project or item will be reviewed by a majority of the team. Changes to the project cost, timeline, or resources will be documented and presented to stakeholders before finalized. Once changes are approved, the documents become current.

6.3 Tools

The team will track each document throughout the project. The Google Drive Folder will have each past version of documents as well as the current version, and each team member will be able to edit each document.

7.0 Appendix

Supplementary information is provided here.