

# iBudget

## Software Project Management Plan

Approvals: L. Assayah LA date: 2/13/12  
C. Leung CL date: 2/13/12  
Q. Pham QCP date: 2/13/12  
V. Velev VV date: 2/13/12  
J. Reimels JR date: 2/13/12

### Revision History

Date	Author	Version	Reason
2/12/12	V.Velev	1.0	First Draft
2/12/12	Q.Pharm	1.0	Draft Schedule
2/12/12	V.Velev	1.1	Added Budget and Resource Allocation section.
2/12/12	V.Dineva	1.1	Risk Management Contribution
2/12/12	Q.Pharm	1.1	Risk Management Contribution
2/13/12	Q.Pharm	1.2	Add TOC and Reference Section. Modify Schedule.

## Table of Contents

1. Introduction .....	3
1.1. Project overview .....	3
1.2. Project deliverables .....	3
1.3. Evolution of the SPMP .....	3
1.4. Project deliverables .....	3
1.5. Acronyms.....	3
2. Project Organization.....	3
2.1. Process Model .....	3
2.2. Organizational Structure .....	4
2.3. Project Responsibilities.....	4
3. Management Process .....	5
3.1. Management objectives and priorities .....	5
3.2. Assumptions, Dependencies and Constraints .....	5
None. ....	5
3.3. Risk management.....	5
3.4. Monitoring and controlling mechanism .....	6
3.5. Staffing Plan .....	6
4. Technical Process .....	7
4.1. Methods, tools and techniques .....	7
4.2. Software Documentation .....	7
5. Resources and schedule .....	7
5.1. Resources .....	7
5.2. Budget and resource allocation .....	7
5.3. Schedule.....	8
6. References.....	10

# 1. Introduction

## 1.1. Project overview

iBudget is planned as a personal financial management software tool that produces as an output a thorough and simple analysis of one's financial statements. The user should be able to use iBudget to easily and efficiently attain an overview of his or her finances as well as create a budget.

## 1.2. Project deliverables

At the end of the school semester, the team shall deliver a working software program as well as all the necessary documentation.

## 1.3. Evolution of the SPMP

This document shall be maintained on a weekly basis by the project leader. It is subject to configuration management by means of the SCMP.

## 1.4. Project deliverables

Source code for the *iBudget* project and all necessary documentation.

## 1.5. Acronyms

QA – quality assurance  
SCMP – Software Configuration Management Plan  
SPMP – Software Project Management Plan  
SRS – Software Requirements Specification  
SDD – Software Design Document  
STP – Software Test Plan

# 2. Project Organization

## 2.1. Process Model

The project will be executed using a Waterfall with Feedback model. This sequential approach will consist of the following phases: Inception, Requirements, Implementation, Testing, and Maintenance. Each phase will allow the team to go back to a previous phase (e.g. go back to Requirements from Implementation to redefine the project.)

## 2.2. Organizational Structure

The team will consist of members with designated roles. The roles are team leader, the configuration management leader, the quality assurance leader, the requirements management leader, the design leader, and the implementation leader. These roles are shown in figure 2.2.

### iBudget Project Organization

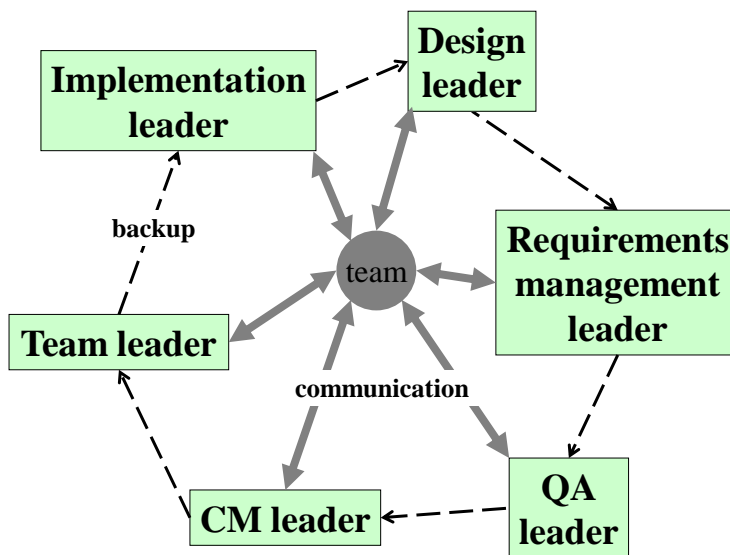


Figure 2.2: iBudget Project Organization

## 2.3. Project Responsibilities

The responsibilities of the participants in the project are shown in table 2.3.

Member	Team Leader	CM Leader	QA Leader	Requirements Management Leader	Design Leader	Implementation Leader
Document Responsibility	SPMP	SCMP	SQAP STP	SRS	SDD	Code Base

Table 2.3: Project Responsibilities

Being responsible for the document includes the following:

- Making sure the document is completed in time.
- Keeping the document up-to-date throughout the project life cycle

### 3. Management Process

#### 3.1. Management objectives and priorities

The highest management priority shall be the completion of the project on time before the end of the school semester. This also includes that the project be on schedule. The second priority shall be meeting a degree of quality as specified by the quality parameters. The third priority shall be to include as many features as were planned throughout the project lifecycle.

#### 3.2. Assumptions, Dependencies and Constraints

None.

#### 3.3. Risk management

#	Risk Title	Likelihood to Occur 1-10	Impact 1-10	Retirement Cost 1-10	Priority	Retirement or Mitigation Plan	Responsible Engineer	Target Completion Date
1	Requirements Inflation	3	8	2	48	Add additional features depending on time.	All	04/02
2	Poor Productivity	7	10	10	40	Keep developers engaged and motivated.	Project Leader	04/30
3	Shorfalls in 3 <sup>rd</sup> party software	4	3	6	336	Analyze software capabilities in advance.	TBD	02/30
4	Insufficient time	8	9	5	30	Follow schedule.	Project Leader	04/30
5	Unforeseen software defects	7	7	5	45	Allocate enough time for defect resolution.	All	04/16

Risk #1: Overpromising and underdelivering on specified requirements will result in the failure of the project. The team has decided on a basic feature set and will add additional functionality depending on time left in the semester.

Risk#2: Time constraints and other personal/professional responsibilities can result in poor productivity. The team leader is responsible for keeping other team members engaged.

Risk#3: Functionality of 3<sup>rd</sup> party tools might not suffice. The team should analyze each tool in advance.

Risk#4: The team is required to finish the project by the end of the school semester. The team leader is responsible for making sure that all deadlines are met.

Risk#5: Software defects are inevitable. All developers should allocate enough time to debug their code and resolve any uncovered defects.

### 3.4. Monitoring and controlling mechanism

There will be a weekly project meeting on Mondays from 5PM to 6PM. The team will discuss all issues pertaining to the project at that time. Further discussion will take place via email or telephone. In case the need arises for addition meetings, the team will schedule one or more on a week-by-week basis depending on the availability of all team members.

Each team member's progress will be tracked by the team leader. Each member is responsible to submitting a weekly report to the team leader, indicating the tasks he or she has worked on throughout the week as well as the tasks planned for the following week. The team member must also specify an estimated time needed to complete planned tasks as well as the actual time spent on already completed tasks.

### 3.5. Staffing Plan

Team member roles are specified as in Table 3.5.

<u>Name</u>	<u>Team Leader</u>	<u>CM Leader</u>	<u>QA Leader</u>	<u>Requirement Management Leader</u>	<u>Design Leader</u>	<u>Implementation Leader</u>
Vladimir Velez	X					
Jonathan Reimels		X				
Vanya Dineva			X			
Laurene Assayah				X		

Churk Leung					X	
Quan Pham						X

Table 3.5: iBudget Staffing Plan

## 4. Technical Process

The SRS will describe the technical process in detail. This section describes aspect of the process which are not explicitly stated in the SRS.

### 4.1. Methods, tools and techniques

The iBudget project will use Apache server, MySQL, and will be implemented in PHP.

### 4.2. Software Documentation

Refer to the SQAP.

## 5. Resources and schedule

### 5.1. Resources

TBD

### 5.2. Budget and resource allocation

Estimate before beginning requirement analysis.

Estimate LOC:

$$\begin{aligned}
 FP &= (\sum UFP_i) * GCF \\
 UFP_i &= 3 (EI) + 4 (EO) + 3 (EIN) + 7 (ILF) + 5 (ELF) = 22 \\
 GCF &= 0.65 + 0.01 * 42 = 1.07 \\
 FP &= 23.54
 \end{aligned}$$

$$LOC = 23.54 * (600 LOC/FP) = 14 KLOC$$






















COCOMOI:

$$\begin{aligned}
 \text{Effort Applied (E)} &= 3.0 * (14)^{1.12} = 57 \text{ man-months} \\
 \text{Development Time (D)} &= 2.5 * (57)^{0.38} = 11 \text{ months} \\
 \text{People required (P)} &= 57/11 = 6
 \end{aligned}$$

## 5.3. Schedule

ID		Task Name	Duration	Start	Finish
1	✓	<b>Scope</b>	<b>6 days</b>	<b>Mon 1/30/12</b>	<b>Mon 2/6/12</b>
2	✓	Define Project Scope	1 hr	Mon 1/30/12	Mon 1/30/12
3	✓	Define Development Tools and Hosting Environment	1 hr	Mon 1/30/12	Mon 1/30/12
4	✓	Define Roles and Responsibility	1 hr	Mon 2/6/12	Mon 2/6/12
5	✓	Define Deliverables	1 day	Mon 2/6/12	Mon 2/6/12
6	✓	Scope Complete	0 days	Mon 2/6/12	Mon 2/6/12
7		<b>Analysis/Software Requirements</b>	<b>11 days</b>	<b>Mon 2/13/12</b>	<b>Mon 2/27/12</b>
8		Conduct needs analysis	6 days	Mon 2/13/12	Mon 2/20/12
9		Draft preliminary software specifications	6 days	Mon 2/13/12	Thu 2/23/12
10	✓	Review software specifications/budget with team	1 hr	Mon 2/20/12	Mon 2/20/12
11		Incorporate feedback on software specifications	6 days	Mon 2/20/12	Mon 2/27/12
12		Develop delivery timeline	6 days	Mon 2/20/12	Mon 3/5/12
13	✓	Analysis complete	0 days	Mon 2/27/12	Mon 2/27/12
14		<b>Design</b>	<b>26 days</b>	<b>Mon 2/20/12</b>	<b>Mon 3/26/12</b>
15		Review preliminary software specifications	8 days	Mon 2/20/12	Mon 3/5/12
16	✓	Develop functional specifications	11 days	Mon 2/20/12	Mon 3/5/12
17		Develop prototype based on functional specifications	14 days	Mon 2/27/12	Mon 3/19/12
18		Review functional specifications	6 days	Mon 3/19/12	Mon 3/26/12
19		Incorporate feedback into functional specifications	6 days	Mon 3/19/12	Mon 4/2/12
20	✓	Design complete	0 days	Mon 3/26/12	Mon 3/26/12
21		<b>Development</b>	<b>26 days</b>	<b>Mon 2/27/12</b>	<b>Mon 4/2/12</b>
22		Review functional specifications	6 days	Mon 2/27/12	Fri 3/30/12
23		Identify modular/tiered design parameters	6 days	Mon 2/27/12	Fri 4/6/12
24	✓	Assign development staff	1 day	Mon 2/27/12	Mon 2/27/12
25	✓	Develop code	6 days	Mon 2/27/12	Mon 3/5/12
26		Developer testing (primary debugging)	21 days	Mon 3/5/12	Mon 4/2/12
27	✓	Development complete	0 days	Mon 4/2/12	Mon 4/2/12
28		<b>Testing</b>	<b>11 days</b>	<b>Mon 4/2/12</b>	<b>Mon 4/16/12</b>
29		Develop test plans	6 days	Mon 4/2/12	Mon 4/9/12
30		Review test plans	6 days	Mon 4/9/12	Mon 4/16/12
31	✓	Implement Test Plans	6 days	Mon 4/9/12	Mon 4/16/12
32		Unit testing complete	0 days	Mon 4/16/12	Mon 4/16/12
33		<b>Documentation</b>	<b>51 days</b>	<b>Mon 2/6/12</b>	<b>Mon 4/16/12</b>
34	✓	Project Proposal	9.2 wks	Mon 2/6/12	Mon 4/9/12
35		SPMP Document	12 days	Mon 2/6/12	Mon 4/9/12
36	✓	SQAP Document	46 days	Mon 2/6/12	Mon 4/9/12
37	✓	SCMP Document	46 days	Mon 2/6/12	Mon 4/9/12
38		Develop Project Schedule	18.5 days	Mon 2/6/12	Tue 4/17/12
39		Develop user manuals	3.9 wks	Mon 2/6/12	Mon 4/23/12
40	✓	Review all user documentation	11 days	Mon 4/9/12	Mon 4/30/12
41	✓	Incorporate user documentation feedback	16 days	Mon 4/9/12	Mon 4/30/12
42		Documentation complete	0 days	Mon 4/30/12	Mon 4/30/12
43		<b>Final Presentation</b>	<b>6 days</b>	<b>Mon 4/23/12</b>	<b>Mon 4/30/12</b>
44		Determine final presentation strategy	6 days	Mon 4/23/12	Mon 4/30/12
45		Prepare Powerpoint Slide	6 days	Mon 4/23/12	Mon 4/30/12
46		Deployment complete	0 days	Mon 4/30/12	Mon 4/30/12
47		<b>Post Presentation Review</b>	<b>1 day</b>	<b>Mon 4/30/12</b>	<b>Mon 4/30/12</b>
48		Document lessons learned	1 hr	Mon 4/30/12	Mon 4/30/12
49		Create software maintenance team	1 hr	Mon 4/30/12	Mon 4/30/12



ID		Task Name	Duration	Start	Finish
50		Software development template complete	0 days	Mon 4/30/12	Mon 4/30/12
51		<b>Deliverables</b>	<b>66 days</b>	<b>Mon 1/30/12</b>	<b>Mon 4/30/12</b>
52		Week 1 Team Report	1 day	Mon 1/30/12	Mon 1/30/12
53		Week 2 Team Report	1 day	Mon 2/6/12	Mon 2/6/12
54		Week 3 Team Report	1 day	Mon 2/13/12	Mon 2/13/12
55		Week 4 Team Report	1 day	Mon 2/20/12	Mon 2/20/12
56		Week 5 Team Report	1 day	Mon 2/27/12	Mon 2/27/12
57		Week 6 Team Report	1 day	Mon 3/5/12	Mon 3/5/12
58		Week 7 Team Report	1 day	Mon 3/19/12	Mon 3/19/12
59		Week 8 Team Report	1 day	Mon 3/26/12	Mon 3/26/12
60		Week 9 Team Report	1 day	Mon 4/2/12	Mon 4/2/12
61		Week 10 Team Report	1 day	Mon 4/9/12	Mon 4/9/12
62		Week 11 Team Report	1 day	Mon 4/16/12	Mon 4/16/12
63		Week 12 Team Report	1 day	Mon 4/23/12	Mon 4/23/12
64		Week 13 Team Report	1 day	Mon 4/30/12	Mon 4/30/12
65		PHASE I	1 day	Mon 2/13/12	Mon 2/13/12
66		PHASE II	1 day	Mon 2/27/12	Mon 2/27/12
67		PHASE III	1 day	Mon 3/26/12	Mon 3/26/12
68		FINAL PRESENTATION	1 day	Mon 4/30/12	Mon 4/30/12
69		Delivarables complete	0 days	Mon 4/30/12	Mon 4/30/12

## 6. References

1. Software Engineering: Modern Approaches / Eric J. Braude, Michael E. Bernstein,  
2<sup>nd</sup> Ed. Wiley, 2011.