

**Software Project Management Plan  
(SPMP)  
Version 1.0**

**UML Sequence Diagram File Generator**

**April 05, 2019**

**Isis Curiel, Bruno Hnatusko III, Brayden Mccoy,**

**Dhyey Patel, Jesse Primiani, Jacob Taylor,**

**and Syed Arshiyah Ali Zaidi**

**Approvals:**

<b><u>Title</u></b>	<b><u>Signature</u></b>	<b><u>Date</u></b>
Project Manager	Isis Curiel	04/05/2019

## Revision History

Date	Revision	Description	Author
4-1-19	0.1	Created rough draft	Isis Curiel
4-4-19	0.4	Added the bulk of the content	Isis Curiel
4-5-19	0.5	Updated the document's style to match that of the other documents	Jesse Primiani
4-5-19	0.6	Proofreading and typo fixes	Bruno Hnatusko III
4-5-19	0.7	Added KLOC estimation calculations and made a few minor adjustments	Jesse Primiani
4-5-19	0.8	Review	Dhyey Patel
4-5-19	1.0	Updated to 1.0 and submitted	Isis Curiel

## **Table of Contents**

<b>1. Overview</b>	Pg. 4
1.1. Project summary	
1.2. Evolution of SPMP	
<b>2. References</b>	Pg. 4
<b>3. Definitions</b>	Pg. 4
<b>4. Project organization</b>	Pg. 4
4.1 External interfaces	
4.2 Internal structures	
4.3 Roles and responsibilities	
<b>5. Managerial process plans</b>	Pg. 5
5.1 Project start-up plan	
5.2 Work plan	
5.3 Control plan	
5.4 Risk management	
5.5 Project closeout plan	
<b>6. Technical process plans</b>	Pg. 7
6.1 Process model	
6.2 Methods, tools, and techniques	
6.3 Infrastructure plan	
6.4 Product acceptance plan	
<b>7. Supporting process plans</b>	Pg. 7
7.1 Configuration management	
7.2 Verification and validation plan	
7.3 Documentation plan	
7.4 Quality assurance plan	
7.5 Reviews and audit plans	
7.6 Problem resolution plan	
7.7 Subcontractor management plans	
7.8 Process improvement plan	
<b>8. Additional Plans</b>	Pg. 8
<b>9. Appedenix</b>	Pg. 9

# **1 Overview**

## **1.1 Project Summary**

This project is organized to generate a file from a Java SDM object. From this stage the user will be able to import the file into Lucidchart where they are free to view/edit the sequence diagram.

## **1.2 Evolution of the SPMP**

This document will be maintained on a weekly basis by the project manager. It is subject to configuration management by means of the SCMP. It is the project manager's responsibility to submit this document as a CI, and to keep it up to date. This SPMP mainly follows the format of IEEE 1058.1-1998.

# **2 References**

[IEEE) The applicable IEEE standards are published on "IEEE Standards Collection," 1997 edition.

# **3 Definitions**

CI = Configuration Item

CMM = Capability Maturity Model, the SEI's model for organizational improvement

IEEE = Institute of Electrical and Electronics Engineers

QA = Quality Assurance

SEI = Software Engineering Institute, Pittsburgh, PA

SCMP = Software Configuration Management Plan

SPMP = Software Project Management Plan (this document)

SRS = Software Requirements Specification

SDD = Software Design Document

STP = Software Test Plan

tbd = to be decided

# **4 Project Organization**

## **4.1 External Interfaces**

The project team will interface with the following individuals: Victor Guo from Beulah Works LLC, for the project's requirements, and Professor Ruijian Zhang from Purdue Northwest through the Software Engineering course.

## 4.2 Internal Structures

The project will be organized as a team of peers with designated roles. The roles are: project manager, integration engineer, software architects (2), requirement engineers (2) and a testing engineer. See Figure 3 in the Appendix for the names of who will fulfill these roles.

## 4.3 Roles and Responsibilities

The responsibilities of the participants in the project are shown in Figure 3. Being responsible for a document includes the following:

- Making sure that the document is created on time
- Having the team leader identify the writers of the document
- Keeping the document up-to-date throughout the project's life cycle

# 5 Managerial Process Plans

## 5.1 Project Startup Plan

### 5.1.1 Estimation Plan

550 lines of code that will be used in this project were written so far as of 04/05/2019, not including system or unit tests. Assuming that the remaining 10 methods for writing to the output adapter will be between 1 and 2 function points each on average, and 21 methods for reading from the input adapter will be between 0.5 and 1 function point each on average, there will be between 1.6 and 2.7 thousand lines of code written for this project, disregarding test code. Calculations are given below:

$$550 + ((1*10) + (0.5*21)) * 53 = 1.6 \text{ KLOC} \Leftarrow \text{Low}$$

$$550 + ((2*10) + (1*21)) * 53 = 2.7 \text{ KLOC} \Leftarrow \text{High}$$

### 5.1.2 Staffing Plan

The roles will be filled as stated in Figure 2: Roles and Responsibilities.

### 5.1.3 Resource Acquisition Plan

The implementation leader will ensure that all team members are supplied with the selected development environment within two weeks of project start.

### 5.1.4 Project Staff Training Plan

All team members are currently taking classes at Purdue Northwest, including but not limited to a Software Engineering course.

## 5.2 Work Plan

### 5.2.1 Work Activities

The work on this project will be divided into configuration management, quality assurance (including testing), requirements analysis, design, and implementation. The project roles and responsibilities are shown in Figure 2: Roles and Responsibilities.

### **5.2.2 Schedule Allocation**

The schedule is shown in Figure 1. Refer to the SQAP for the schedule of quality activities.

### **5.2.3 Resource Allocation**

The work breakdown structure is shown in Figure 2.

### **5.2.4 Budget Allocation**

This is intentionally left blank.

## **5.3 Control Plan**

The entire team will meet at the beginning of each phase (requirements, design, and implementation) within each iteration. There will be weekly project meetings on Fridays from 5:00 p.m. to 6:00 p.m. Team members are requested to keep Monday or Wednesday evenings from 5:00 p.m. to 6:00 p.m. open for an additional meeting, in case such a meeting becomes necessary. The team leader will inform the team 24 hours before if an additional meeting is to take place.

### **5.3.1 Requirements Control Plan**

The requirements engineer will report on the progress of the completion of the requirements at each meeting.

### **5.3.2 Schedule Control Plan**

The project leader will report to the team on the status of the schedule verbally each week.

### **5.3.3 Budget Control Plan**

This is intentionally left blank.

### **5.3.4 Quality Control Plan**

The testing engineer will provide verbal reports to team members at team meetings when needed.

### **5.3.5 Reporting Plan**

Verbal reports or SMS reports will be given to project manager. Issue affecting human safety will be reported to all project personnel and management, regardless of the plan in this document.

### **5.3.6 Metrics Collection Plan**

See the Software Quality Assurance Plan.

## **5.4 Risk Management**

### **Risk 1: Lack of understanding**

This risk refers to when any one or more team members have a misunderstanding about the project or are stuck at their task. The team member(s) will consult Jesse Primiani for matters regarding the project design, coding, as well as minor technical issues. Victor Guo of Beulah Works will be contacted for matters regarding the project's requirements.

### **Risk 2: Lack of effort**

This risk refers to when any one or more team members are not performing their assigned task. The other team members will split the remaining work of the team member(s) that are not performing the task between themselves. The professor will be made aware of lack of effort.

### **Risk 3: Technical issues with dependencies**

This risk refers to any major technical issues that may occur, such as if any dependant library does not support needed functionality, or if the required functionality is poorly documented. In the former case, Victor Guo of Beulah Works will be contacted, and the requirements may either be altered or another library implemented. In the latter case, a team meeting will be held to test and experiment with various means of using the library to implement the required functionality.

## **5.5 Project Closeout Plan**

The UML Sequence Diagram File Generator will be completed before the beginning of May 2019.

# **6 Technical Process Plans**

## **6.1 Process Model**

The whole project will be executed with using the unified development process with an iteration corresponding to each version. The first iteration will be a working prototype but will be fully documented. The second iteration will result in version 1 of this UML Sequence Diagram File Generator project.

## **6.2 Methods, Tools, and Techniques**

The UML Sequence Diagram File Generator project will be implemented in Java. Object Orientation methodologies are to be used throughout. Javadoc will be used for documentation as much as possible. Git and Eclipse will be used for code management.

## **6.3 Infrastructure Plan**

No external hardware is necessary, as every team member will use their personal laptops to complete their assigned tasks.

## **6.4 Product Acceptance Plan**

The final product will be showcased in a presentation for the client.

## **7 Supporting Process Plans**

### **7.1 Configuration management**

This plan can be found in the SCMP (Software Configuration Management Plan).

### **7.2 Verification and validation plan**

This plan can be found in the SVVP (Software Verification and Validation Plan).

### **7.3 Documentation plan**

This plan can be found in the SQAP (Software Quality Assurance Plan).

### **7.4 Quality assurance plan**

This plan can be found in the SQAP (Software Quality Assurance Plan).

### **7.5 Reviews and audit plans**

This plan can be found in the SQAP (Software Quality Assurance Plan).

### **7.6 Problem resolution plan**

This plan can be found earlier in this document under Risk Management, Section 5.4.

### **7.7 Subcontractor management plans**

This plan can be found in the SCMP (Software Configuration Management Plan).

### **7.8 Process improvement plan**

This plan can be found in the SQAP (Software Quality Assurance Plan).

## **8 Additional Plans**

None.



## 9 Appendix

**Figure 1: Schedule**

<i>Sunday</i>	<i>Monday</i>	<i>Tuesday</i>	<i>Wednesday</i>	<i>Thursday</i>	<i>Friday</i>	<i>Saturday</i>
1	2	3	4	5	6 • Weekly project meeting	7
8	9	10	11	12	13 • Weekly project meeting	14
15	16	17	18	19	20 • Weekly project meeting	21
22	23	24	25	26	27 • Weekly project meeting	28
29	30	31				

**Figure 2: Roles/Responsibilities**

Name	Isis Curiel	Jesse Primiani	Brayden McCoy	Dhyey Patel	Syed Zaidi	Jacob Taylor	Bruno Hnatusko
Role	Project manager	Software architect	Software architect	Integration Engineer	Requirements Engineer	Requirements Engineer	Testing Engineer
SDD		X	X	X			
SRS					X	X	
SPMP	X						
SCMP				X			
SQAP							X
STP							X
User Manual				X			
SVVP							X

**Figure 3: Milestones for Tasks**

<b><u>Task</u></b>	<b><u>Date Team Will Review</u></b>
SDD	March 1st 2019
SPMP	March 29th 2019
SCMP	February 4th 2019
SQAP	February 1st 2019
SRS	February 21st 2019
SVVP	February 1st 2019
STP	April 19th 2019
User Manual	April 19th 2019
Final Product	April 19th 2019

**Figure 4: Risk Analysis Table**

#	Risk Title	Likelihood 1-10 1= least likely	Impact 1-10 1= least likely	Retirement Cost 1-10 1=lower cost	Priority (lowest # handled first )	Retirement /Mitigation Plan	Responsible Engineer
1	Lack of understanding	6	8	1	2	Speak with J.P for clarification & Victor for requirements	Jesse Primiani (J.P)
2	Lack of effort	3	10	4	3	Report to I.C so wok can be divided and person reported to professor	Isis Curiel (I.C)
3	Technical issues with dependencies	7	9	3	1	Report to Victor Guo or have a team meeting	Isis Curiel