Communication Model for Cooperative Robotics Simulator

Project Plan

Version 1.1

---

1 Introduction

This document provides an overview of project plan for Communication Model for Cooperative Robotics Simulator project.

1.1 Purpose

The purpose of this document is to provide a project plan, cost estimation and architecture elaboration plan of "Communication Model for Cooperative Robotics Simulator" project.

1.2 Scope

This document covers project plan for "Communication Model for Cooperative Robotics Simulator" including project schedule, cost estimation and architecture elaboration plan. Project plan will detail the phases, iterations, and milestones that will comprise the project. Cost estimation will provide detailed estimate on the size, cost and effort required for the project. The Architecture Elaboration plan will define the activities and actions that must be accomplished prior to the Architecture Presentation.

1.3 References

- Royce, Walker. *Software Project Management A unified framework*, 1st ed. Addison-Wesley, 1998.
- Pressman, Roger. *Software Engineering a practitioner's approach*, 5th ed. McGraw-Hill international edition, 2001.

2. Project schedule

Phase I Inception Phase	**February 2 – February 27**
Study on material of presentation 1	February 2 – February 3
Project Plan	February 4 – February 6
Project overview	February 4 - February 5
Software Quality Assurance	February 4 – February 5
Architecture elaboration plan	February 5 –February 6
Software Requirements Specifications	February 9 – February 11
Cost Estimation	February 10 – February 11
Software demo	February 10 – February 13
First Presentation slides	February 16 – February 17
First presentation	February 27
Phase II Elaboration Phase	**February 26 – April 26**
Review formal requirement specification	February 26 – February 27
Refine vision document	February 27 – March 1
Architecture design	March 4 – March 10
Formal requirement specification	March 16 – March 23
Executable architecture prototype	March15 –March 22
Inspector checklist	March 23 – March 24
Implementation plan	March 29 – March 31
Updated project plan	April 1 – April 2
Updated cost estimation	April 5 – April 6
Test plan	April 6 – April 12
Second Presentation slides	April 21 – April 22
Second presentation	April 26
Phase III Construction Phase	**June 23 – August 19**
Modify Phase II document	June 23 – June 29
Create component diagram	June 29 – July 7
Coding	July 8 – July 13
Testing	July 14 – July 28
User manual	July 29 – July 30
Project evaluation	August 2 – August 6
Final presentation slides	August 9 – August 10
Prepare final documentation	August 11 – August 12
Final presentation	August 19

3. Cost Estimation

3.1 Function Point Analysis

First, Function Point Analysis is used to estimate number lines of code. The project's features are classified into five categories, inputs, outputs, inquiries, files and interfaces. The whole process of FPA consists of three major parts, which are calculating Unadjusted Function Points (UFP), calculate Adjusted Function Point (AFP) and calculate SLOC

3.1.1 Calculate Unadjusted Function Points

Measurement Parameters		Simple	Average	Complex	Total
Input	Sign on request Request to send messages. Request to start up/ shutdown links Set up parameters	3 3 3 3			12
Outputs	Success or failure to sign on Success or failure to set parameters. Distribute messages to all recipients	4 4	 5		13
Inquiries					0
Files					0
External Interfaces					0
Total	**Unadjusted Function Points**				**25**

3.1.2 Calculate Adjusted Function Points

To compute Adjusted Function Points, the following equation is used.

$$FP = \text{Unadjusted Function Points} * (0.65 + 0.01 * \Sigma F_i)$$

(where Fi are complexity adjustment factors)

The system is rated on a set of complexity adjustment factors on a scale from 0 to 5 where 0 is no influence and 5 is essential.

Complexity Adjustment Factors	Value
Are data communications required	5
Is the code designed to be reusable	3
Total Complexity Adjustment Factors	**8**

Therefore, total Adjusted Function Points is 25 * 0.73 = 18.25

3.1.3 Calculate number of lines of code (SLOC)

Language factor of Java programming is 50. Therefore, SLOC = 21.17 * 50 = 912.5 or 0.9125 KSLOC

3.2 COCOMO I

COCOMO or COnstructive COst Model developed by Barry Boehm will be used to estimate total cost the project in terms of time and effort, since COCOMO II is used widely in large development team. The cost estimation is based on that this project is uncomplicated; therefore the organic mode cost estimation relationship will be used.

$$\text{Effort} = 3.2 * \text{EAF} * (\text{Size})^{1.05}$$

where:

Effort = number of person-months
EAF = an effort adjustment factor that characterizes the domain, personnel, environment, and tools used to produce the artifacts of the process. (Since EAF is difficult to determine, EAF is not considered at this time.)
Size = size of the end product (in human-generated source code), measured by the number of delivered source instructions.(in KSLOC)

$$\text{Effort} = 3.2 * (0.9125)^{1.05}$$

Effort = 2.90 person-months

$$\text{Time} = 2.5 * (\text{Effort})^{0.38}$$

where:

Time = total number of months

$$\text{Time} = 2.5 * 2.90^{0.38}$$

Time = 3.74 months

In conclusion, this project requires one person to complete in 3.74 months. As Boehm mentioned, there are 152 working hours in a month. However number of working hours per person in a month is varied, time to complete this project will be different.

4. Architecture Elaboration Plan

4.1 Purpose

The Architecture Elaboration Plan will define the activities and actions that must be accomplished before the Architecture Presentation.

4.2 Updated Vision Document

The Vision documents, project overview and Software Requirements Specifications, will be updated with any modification based on the project committees' suggestion. The requirements will be ranked according to importance and a set of critical requirements will be identified.

4.3 Updated Project Plan

The project plan will be adjusted with any modification, corrected phase and deliverables along with the estimated completion date.

4.3.1 Cost Estimation

The cost estimation will be updated on the size, cost and effort required for the project implementation

4.3.2 Implementation Plan

The Implementation Plan will be developed to define the activities that must be accomplished during implementation.

4.4 Architecture Design

The complete architecture design will be documented using UML diagram such as class diagram, object diagram and sequence diagram.

4.4 Formal Requirement Specification

The class diagram from architecture design will be formally specified using UML/OCL methodology. The USE tool, a UML-based Specification Environment, will be used to implement UML/OCL.

4.5 Test Plan

The test plan will be developed to address the required tests to show that the product satisfies the requirements. Evaluation criteria for all critical requirements will be included in the test plan.

4.6 Formal Technical Inspection

The class diagram will be subjected to a formal technical inspection by two MSE students (inspectors), Esteban Guillen and Kevin Sung. Each inspector will provide a report on the result of his/her inspection that will be documented.

4.7 Executable Architecture Prototype

An executable prototype will be built in one or more iterations. The prototype will identify all the critical requirements, which is addressed in the vision document.