Project Management Plan

for

Pivi

Version 3.0

Prepared by

Giridhar Peddabuttaiahgari

Sri Kiran Panchavati Ganesh

Snehal Shendaware

Nithya Kogaleru

Pujitha Kara

Arizona State University

11/23/2016

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  | 3.0 |

1. INTRODUCTION

This is the software system proposal document for the project Pivi: An eclipse Plug-In for Visual Parallel Programming sponsored by Dr. Javier Gonzalez. This project is being undertaken by the Group 14. The team is comprised of graduate students majoring in Software Engineering at Arizona State University. The team members are enrolled in a two-semester senior project course Software Factory. Successful delivery of the desired software product will fulfill the senior project requirement for the student team members.

PROJECT SPONSOR

Name Javier Gonzalez

Company ASU

Division/Unit CIDSE

Email [javiergs@asu.edu](mailto:javiergs@asu.edu)

DEVELOPMENT TEAM

|  |  |
| --- | --- |
| **Name** | **Email ID** |
| Giridhar Peddabuttaiahgari | [gpeddabu@asu.edu](mailto:gpeddabu@asu.edu) |
| Nithya Kogaleru | [nkogaler@asu.edu](mailto:nkogaler@asu.edu) |
| Pujitha Kara | [pkara1@asu.edu](mailto:pkara1@asu.edu) |
| Snehal Shendaware | [sshendwa@asu.edu](mailto:sshendwa@asu.edu) |
| Sri Kiran Panchavati Ganesh | [spanchav@asu.edu](mailto:spanchav@asu.edu) |

* 1. Purpose

The purpose of [this document](https://en.wikipedia.org/wiki/Project_planning) is to identify the scope of the project, [estimate](https://en.wikipedia.org/wiki/Estimation_in_software_engineering) the [work](https://en.wikipedia.org/wiki/Work_(project_management)) involved, and create a [project schedule](https://en.wikipedia.org/wiki/Gantt_chart).  It provides a baseline of what has to be achieved and gives an overview of the project and deliverables.

* 1. Scope

This document provides project overview, project organization, project management and control, technical process, the activities and schedule. The document does not cover technical details, requirement details. This is an initial proposal of the project details.

1. PROJECT OVERVIEW

This Eclipse plug-in focused on facilitating teaching techniques and implications of parallel, concurrent and multicore computing. It builds inside Eclipse an atmosphere of visual programming based on icons. The tool allows students to create a visual model of a problem (by the interconnection of representative icons of atomic structures) building a graph that represent a complete programmable system. Then, automatically generates code of the problem in Java or C/C++, inlaying the necessary elements for concurrent or parallel systems, to compile, execute and run.

* 1. Project Summary

The overall project is majorly divided into three phases. The first phase includes understanding and gathering the requirements. This phase also includes understanding the purpose and scope of the project. The second phases involve prototyping a design and later implementing the design. The final phase includes testing and a hand over of the project to the sponsor.

* 1. Project Deliverables

The project deliverables can be classified into two phases. The first set of deliverables include the following

* Project Charter
* Project Management Plan
* Software Requirement & Specification (SRS)
* Software Design & Specification (SDS)
* Demo of the project idea.

The second set of deliverables include the following

* Version 1 Rollout
* Version 2 Rollout
* User’s manual
* Hand Over
  1. The Management Plan and the Planning Process

An estimation of deadlines for completing each of the above mentioned phases are as follows,

|  |  |  |
| --- | --- | --- |
| **Phase** | **Estimated Start Date** | **Estimated End Date** |
| Project Charter | 10/26/2016 | 11/16/2016 |
| Project Management Plan | 10/26/2016 | 11/16/2016 |
| Requirement Elicitation | 11/02/2016 | 11/16/2016 |
| Demo | 11/14/2016 | 11/30/2016 |
| Design Document | 11/09/2016 | 11/20/2016 |
| Version 1 Rollout | 01/09/2017 | 03/31/2017 |
| Version 2 Rollout | 04/01/2017 | 04/30/2017 |
| User Manual | 04/15/2017 | 04/30/2017 |
| Hand Over |  | 04/30/2017 |

The above mentioned estimated dates can be modified depending on the progress made and sponsor’s requirements.

1. PROJECT ORGANIZATION

This part describes the phases of work that will be done during the course of the project. It explains in detail about the process model, organization structure of the team and responsibilities fulfilled in each phase.

* 1. Process Model

For this project we have decided the process model followed would be an agile model. We will be using Taiga as the online scrum board where we can view and act on the product backlog decided by the product owner. Each sprint will last for 15 days and contents of one particular sprint will be added as per the requirement and delivered at the end of the sprint which will move further for quality assurance.

* 1. Organizational Structure and Interfaces

The team consists of 6 members who will interchange roles like developer, tester, integrator every sprint. Although the role of manager is assigned to a single person in the team. The project manager will be point of contact to the sponsor. The developer is responsible for application development and unit testing tester will be taking care of quality assurance.

* 1. Project Responsibilities

During the first phase of the project we would have constant interactions with the product owner and discuss the scope and exact requirements of the project. Then we start drafting a Software requirement specification document and review it with the sponsor to make changes and get a sign off for the final requirements. We will start research on the technologies and practices closely related to the development of the project. This involves brainstorm session, discussions, critical analysis of the technologies that are appropriate for the application development. We will then create a prototype of the original model as a proof of concept to showcase our product to the owner and ask for feedback.

Once this is approved we move forward to implement the actual product on eclipse as a plugin with version 1 being the UI for the application with basic visualizations with in the requirements. Once this is completed we proceed to the second version where development of visualizations for multithreading operations take place and a code for the same is generated behind the scenes. This tool will be developed and tested in perspective of the end user who will be a student using this tool to learn parallel processing and multithreading using visualizations in Java.

On successful completion of this phase it will be put on Eclipse Marketplace for open source usage and if time permits we will move ahead to third version which optimizes second version and enhances it such that it can be used at an enterprise level to help develop other software.

1. PROJECT MANAGEMENT AND CONTROL

We will use Agile methodology for managing and developing this project as it can accommodate changes better. The team members will form a group in which each member will be responsible for one or more of the activities needed. We will use Git and Taiga for monitoring and measuring the progress of the project. We are planning to have two week sprints. First two sprints will be exploratory sprints where we spend time on activities like gathering requirements, understanding requirements, creating project plan, creating software requirement specification, Learning the technologies and validating the requirements. The project is validated by constant monitoring by the sponsor. Completeness of the requirements is measured by showing a proof of concept to the sponsor and accommodating his feedback.

* 1. PROJECT MANAGEMENT OBJECTIVES AND PRIORITIES

The requirements as specified by the sponsor are of highest priority in terms of project completion. The major objectives include regularly meeting the sponsor, collecting the requirements incrementally, implementing them and making changes depending on the feedback provided. The project status is provided on a weekly basis by personally meeting the sponsor. The progress of the project can be tracked anytime by monitoring the Github and Taiga tools. Other documents that are submitted as a part of the course work include minutes of meeting, Internal notes with an agenda for the week and the work contributed individually by each team member. The meetings with the professor/ TA are scheduled every week and the major agenda is to update the project status and to collect weekly goals. The meetings with the sponsor are scheduled as and when required and the frequency is weekly. The taiga tool can be used as a reference for project log which clearly mention the tasks assigned to each individual with a timeline and the status of the tasks.

* 1. ASSUMPTIONS, DEPENDENCIES, and CONSTRAINTS

Due to the quasi-academic nature of the project, there is a time constraint for the project to be completed by 6 months. The system being developed should be compatible with java as multithreading visualization is only supported by java. This project targets to develop a plug-in for eclipse, using which a user can program a multi-threaded application in java. This is based on the assumption that the existing system is compatible with eclipse IDE.

The progress made determines the exact scope of the project. The sponsor proposed three phases. The first phase includes developing the plug in. The second phase include the UI, demonstrate working of threads and support for parallel programming. The third phase includes having advanced features and developing the plug in for use at enterprise level and make the plug in available in eclipse marketplace. The first phase should be delivered and depending on the progress made, a commitment is made on delivering the second and third phases.

* 1. SCHEDULE CONTROL

The team uses Taiga Scrum Board, Slack and GitHub to keep a track of the progress against the baseline schedule. All of these three platforms have been integrated with each other by the team. Each of the different tasks for each of the phase is divided as a user story on Taiga and is assigned to each of the team member.

The version 2 of the project is mainly targeted to students for learning purposes. It is used to facilitate learning of parallel programming. Version 3 of the project will be aimed at enterprise level to be included in the Eclipse Marketplace. This version will have more optimizations and will have enhanced security features to be used at the enterprise level.

1. TECHINCAL PROCESS:

The team will present various technical documents such as the Software Design document, Software Requirements document, Software Management Plan, Project Charter and User Guide that aids in representing the technical details that will need to be recorded and published during project development.

The team will use Taiga Scrum Board, Slack and GitHub to keep a track of the progress against the baseline schedule. All of these three platforms have been integrated with each other by the team. Each of the different tasks for each of the phase is divided as a user story on Taiga and is assigned to each of the team member.

* 1. Methods, Tools and Techniques

The team will present the UML model and the set of diagrams to represent various aspects of the software system to be developed. The requirement specifications are represented with a set of Use Cases and their associated specifications.

In addition, the informational requirements of the system are represented with an information model depicted as an Entity Relationship Diagram (ERD).

The plug in will be deployed in eclipse. Back end code will be generated in java. It can be run on any machine where we can install Eclipse IDE and JDK. All the code will be integrated using GitHub. The progress of the project will be monitored by Taiga Scrum Board.

* 1. Software Documentation

The team will present various technical documents such as the Software Design document, Software Requirements document, Software Management Plan, Project Charter and User Guide that aids in representing the technical details that will need to be recorded and published during project development.

User guide document and a few examples are additional documents required in order for the developed system to be operationally useful. As this project is mainly targeted for students to learn concurrent programming, the team will have additional task of including further more documentation including installation document, user guides, and basic examples of how the system works. These documents help the students to use the system more efficiently.

* 1. Documents

The various technical documents presented by the team are described as follows:

* Project Charter: This document describes the main intent of creating a mutual understanding between the team and the sponsor of what is expected over the course of the project.
* Project Management Plan: The purpose of this document is to identify the scope of the project, estimate the work involved, and create a project schedule. It provides a baseline of what has to be achieved and gives an overview of the project and deliverables.
* Software Requirement Document: This document identifies the product whose software requirements are specified in this document, including the revision or release number.
* Software Design Document: The Software Design document describes all of the essential components of the system to satisfy the requirements given by the client. This report is used especially during implementation of the system.
* User Guide: This document gives details about how to use the product developed along with its specifications and the pre requisites required.
* Minutes of Meeting: This document provides the time the team spent with the sponsor during the course of the project.

1. ACTIVITIES AND SCHEDULE

The detail description of the activities and tentative schedules are described below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Start Date** | **End Date** | **Duration** | **% Complete** | **Status** |
| Template Tutorial & Tips |  |  |  |  | Resources |
|  |  |  |  |  |  |
| **Project Planning** | **10-19-16** | **11-30-16** | **31d** | **80%** | **In Progress** |
| Meeting the sponsor | 10-19-16 | 11-30-16 | 31d | 60% | In Progress |
| Create Project Management Plan | 10-26-16 | 11-16-16 | 16d | 100% | Completed |
| Create Project Charter | 10-26-16 | 11-16-16 | 16d | 100% | Completed |
| **Requirement Elicitation** | **10-26-16** | **11-16-16** | **16d** | **100%** | **Completed** |
| Requirement Gathering | 10-26-16 | 11-09-16 | 11d | 100% | **Completed** |
| SRS Documentation | 11-02-16 | 11-16-16 | 11d | 100% | **Completed** |
| **Project Design** | **11-02-16** | **11-28-16** | **19d** | **50%** | **In Progress** |
| Research of technologies | 11-02-16 | 11-04-16 | 3d | 100% | **Completed** |
| Finalize Technologies | 11-02-16 | 11-11-16 | 8d | 100% | **Completed** |
| Planning of system design | 11-09-16 | 11-17-16 | 7d | 60% | **In Progress** |
| Create SDS Document | 11-09-16 | 11-21-16 | 9d | 30% | **In Progress** |
| Create Prototype | 11-14-16 | 11-28-16 | 11d | 10% | **In Progress** |
| **Software Development V-1.0** | **01-09-17** | **03-31-17** | **60d** |  | **Not Started** |
| Implement 4.1 (SRS) | 01-09-17 | 01-30-17 | 16d |  | **Not Started** |
| Implement 4.2 (SRS) | 01-31-17 | 02-08-17 | 7d |  | **Not Started** |
| Implement 4.5 (SRS) | 02-09-17 | 02-17-17 | 7d |  | **Not Started** |
| Implement 4.3 (SRS) | 02-17-17 | 03-03-17 | 11d |  | **Not Started** |
| Implement 4.4 (SRS) | 03-03-17 | 03-17-17 | 11d |  | **Not Started** |
| Testing and bug fixes | 03-17-17 | 03-31-17 | 11d |  | **Not Started** |
| **Software Development V-2.0** | 03-31-17 | 05-01-17 | 22d |  | **Not Started** |
| Implement Concurrent Programming | 03-31-17 | 04-20-17 | 15d |  | **Not Started** |
| Testing and bug fixes | 04-21-17 | 05-01-17 | 7d |  | **Not Started** |
| **Documentation** | 04-12-17 | 04-28-17 | 13d |  | **Not Started** |
| User Manual | 04-12-17 | 04-28-17 | 13d |  | **Not Started** |
| Examples | 04-12-17 | 04-28-17 | 13d |  | **Not Started** |