Project Plan Document

Planr, an Agile Project Planning Application

Version 2.0

Submitted in partial fulfillment of the requirements of the degree of Master of Software Engineering

Prepared by

Michael Blakeman

CIS 895 – MSE Project

Kansas State University

Table of Contents

[1. Introduction 3](#_Toc83632542)

[1.1 References 3](#_Toc83632543)

[1.2 Terms 3](#_Toc83632544)

[2. Work Breakdown Structure 3](#_Toc83632545)

[2.1 Inception Phase 3](#_Toc83632546)

[2.2 Elaboration Phase 4](#_Toc83632547)

[2.3 Production Phase 4](#_Toc83632548)

[3. Cost Estimate 4](#_Toc83632549)

[3.1 COCOMO 2.0 4](#_Toc83632550)

[4. Architecture Elaboration Plan 6](#_Toc83632551)

[4.1 Revise the Vision Document 6](#_Toc83632552)

[4.2 Revise the Project Plan 7](#_Toc83632553)

[4.3 Develop a Formal Specification 7](#_Toc83632554)

[4.4 Prepare the Architectural Design Document 7](#_Toc83632555)

[4.5 Prepare the Test Plan 7](#_Toc83632556)

[4.6 Conduct a Technical Inspection 7](#_Toc83632557)

[4.7 Implement an Executable Architecture Prototype 7](#_Toc83632558)

# Introduction

This is the initial project plan for the Planr application for the Master of Software Engineering final project.

## 1.1 References

Additional references will go here.

## 1.2 Terms

Any needed terms will go here.

# Work Breakdown Structure

See Figure 1. for preliminary schedule for the project in the form of a Gantt chart.

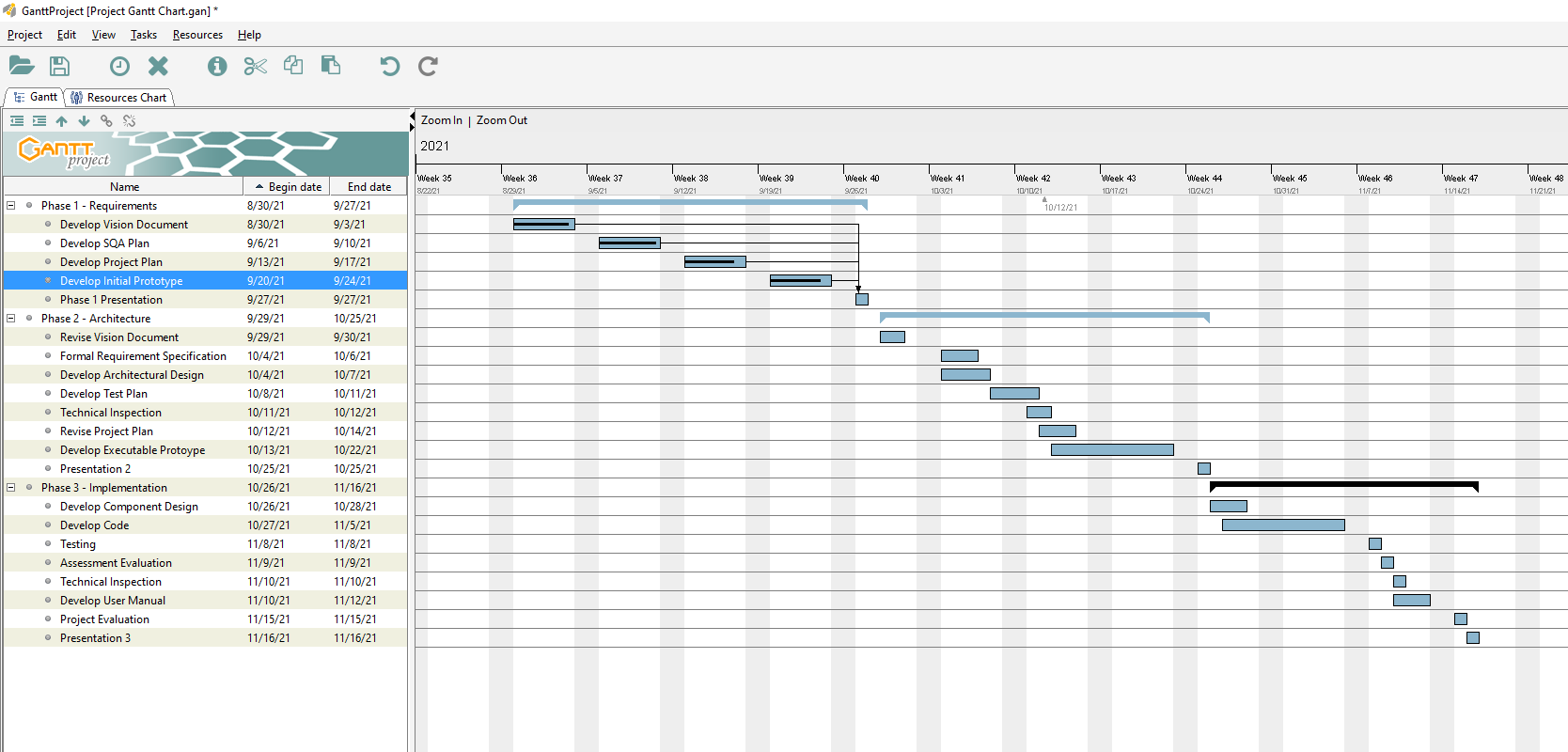


Figure 1. Project Gantt Chart

## 2.1 Inception Phase

The inception phase includes the tasks to prepare a vision document, project plan document, software quality assurance plan document, developing an initial prototype, and presenting the inception phase output to the project supervisory committee.

The prototype will demonstrate a user interface that takes already input data from a project, with features, and resources (engineers) and will output a roadmap in basic form. The initial prototype will have minimal functionality but will demonstrate the schedule output with UI / UX updates to come in later phases.

The inception phase will conclude upon approval of the supervisory committee.

## 2.2 Elaboration Phase

The elaboration phase includes tasks to revise the vision and project plan documents, develop a formal specification of one aspect of the software, prepare the architectural design document, prepare a test plan, implement an executable architecture prototype, conduct a technical inspection of one elaboration phase artifact, and present elaboration phase products to the supervisory committee.

The executable architecture prototype will demonstrate the architecture of the software on the critical requirements.

The elaboration phase will conclude upon approval of the supervisory committee.

## 2.3 Production Phase

The production phase encompasses the tasks to prepare the component design document, develop remaining code and tests, conduct testing, evaluate the project, and present production phase outputs to the supervisory committee.

The production phase presentation will include the production phase outputs and a final demonstration of the software.

The production phase will end upon approval of the supervisory committee.

# Cost Estimate

Using the COCOMO Model, Barry Boehm’s Constructive Cost Model based on numbers of lines of code (LOC), I can use what is known about the current project phase and software left to develop to estimate a cost for the Planr project and application. Boehm characterized three different project types: Organic, Semi-Detached, and Embedded. An organic project is one that requires a small team, has a well understood problem that has been solved in the past and the team has minimal experience regarding the problem. A semi-detached project is more difficult to develop for and require more experience and better guidance than an organic project. An embedded software project requires the highest level of complexity and experience, thus also requiring a larger team that is sufficiently experienced in the problem at hand.

## 3.1 COCOMO 2.0

The Planr project is lower to average complexity, and I have experience in the development language, project planning, and have a general idea of the planning algorithm needed. I would categorize Planr as being an *organic* software project. This allows for the COCOMO Organic Project cost estimation formula of:

*E* = a(KLOC)*b*

*time* = *c*(*Effort*)*d*

Using the basic COCOMO model for organic projects we can calculate effort as:

Effort(*E)* = 2.4(3.5)1.05 = 8.94 person months

Scheduled Time (*time*) = 2.5(8.94)0.38 = 5.75 scheduled months

Using Effort Adjustment Factors, I can calculate cost drivers. The Delphi technique uses 15 adjustment factors and ranges of values as seen in Figure 2.



Figure 2. COCOMO81 cost drivers and effort multipliers



Figure 3. Planr effort adjustment factors

The numbers in Figure 3. allow for me to calculate an adjusted effort factor value of 0.94. This applied to the effort calculation earlier gives me:

Eadj = 2.4(3.5)1.05 \* 0.94 = 8.41 person months

*time*adj = 2.5(8.41)0.38 = 5.62 scheduled months.

# Architecture Elaboration Plan

## 4.1 Revise the Vision Document

The student must implement suggested changes recommended by the supervisory committee into the vision document at which time the updated vision document will be submitted to the major professor for approval.

## 4.2 Revise the Project Plan

The student must implement an updated project plan that provides a detailed implementation phase plan and revised cost estimate at which time the updated project plan document will be submitted to the major professor for approval.

## 4.3 Develop a Formal Specification

The student must formally specify the UI (user interface) of the Planr app, including user interactions in UML at which time the formal specification will be submitted to the supervisory committee for approval.

## 4.4 Prepare the Architectural Design Document

The student must develop an architectural design document to the level of abstraction of component interfaces using appropriate diagrams at which time the architectural design document will undergo technical inspection and be submitted to the supervisory committee for approval.

## 4.5 Prepare the Test Plan

The student must prepare a test plan for the software to be executed in the production phase. The test plan must include unit, integration, and component and system-level functional tests at which time the test plan will be submitted to the supervisory committee for approval.

## 4.6 Conduct a Technical Inspection

The student must develop a technical inspection checklist for the architectural design document and coordinate the inspection with the inspectors. Austin Gray and Eric Haslag are the designated technical inspectors for the project. The inspection checklists and letters will be submitted to the supervisory committee for approval.

## 4.7 Implement an Executable Architecture Prototype

The executable prototype will demonstrate the architecture for the critical requirements. The demonstration and presentation to the supervisory committee will expose the top technical risks in the project.