**Project Plan**

For Multiagent Control of Traffic Signals

Version 1.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Bryan Nehl

CIS 895 – MSE Project

Kansas State University

Table of Contents

[1 Introduction 3](#_Toc308385544)

[1.1 References 3](#_Toc308385545)

[1.2 Terms 3](#_Toc308385546)

[2 Work Breakdown Structure 3](#_Toc308385547)

[2.1 Inception Phase 4](#_Toc308385548)

[2.2 Elaboration Phase 4](#_Toc308385549)

[2.3 Production Phase 4](#_Toc308385550)

[3 Cost Estimate 4](#_Toc308385551)

[3.1 COCOMO 2.0 4](#_Toc308385552)

[4 Architecture Elaboration Plan 4](#_Toc308385553)

[4.1 Revise Vision Document 4](#_Toc308385554)

[4.2 Revise Project Plan 4](#_Toc308385555)

[4.3 Create Formal Specification 5](#_Toc308385556)

[4.4 Create Architectural Design 5](#_Toc308385557)

[4.5 Create Test Plan 5](#_Toc308385558)

[4.6 Conduct Technical Inspection 5](#_Toc308385559)

[4.7 Create Executable Architecture Prototype 5](#_Toc308385560)

# Introduction

This is the initial project plan for the Multiagent Control of Traffic Signals (MACTS) Masters of Software Engineering project.

## References

1. W. Royce, Software Project Management: A Unified Framework, Addison-Wesley,

1998, p. 34, pp. 265-281.

1. B. Boehm et al., “Cost Models for Future Software Processes: COCOMO 2.0,” Annals of

Software Eng., Vol. 1, 1995, pp. 57-94.

1. K-State Master of Software Engineering web site, “MSE Portfolio Requirements,”
2. November 28, 2011; http://mse.cis.ksu.edu/portfolio.html.
3. Center for Systems and Software Engineering web site, “COCOMO II,” December 4, 2011: http://sunset.usc.edu/csse/research/COCOMOII/cocomo\_main.html.
4. The Code Project web site, “Software Project Cost Estimates Using COCOMO II Model,” December 4, 2011: http://www.codeproject.com/KB/architecture/cocomo2.aspx.
5. Naval Postgraduate School web site, “COCOMO II - Constructive Cost Model,” December 4, 2011: http://diana.nps.edu/~madachy/tools/COCOMOII.php.
6. Center for Software Engineering, USC, COCOMO II: Model Definition Manual Version 2.1, 2000.
7. The Code Project web site, “Calculating Function Points,” December 4, 2011: http://www.codeproject.com/KB/architecture/Calculate\_Function\_Point.aspx.
8. USC Center for Software Engineering website, “COCOMO II Affiliates,” December 4, 2011: http://csse.usc.edu/csse/affiliate/private/COCOMOII\_Driver+Calc\_Ss/SpreadSheet-COCOMOII.html.

## Terms

COCOMO is short for COnstructive COst MOdel.

SLOC is an acronym for Source Lines Of Code. These are lines of code that are neither comment or whitespace.

UFP is an acronym for Unadjusted Function Points.

**Data Functionality**

Internal Logical Files (ILF)

External Interface Files (EIF)

**Transaction Functionality**

External Inputs (EI)

External Outputs (EO)

External Queries (EQ)

# Cost Estimate

## COCOMO 2.0

### Computing Unadjusted Function Points

Unadjusted function points are used in the early design stage for project estimation. I followed the information I found about computing function points on the Code Project website[9]. I reviewed my use cases and arrived at the following table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case | ILF | EIF | EI | EO | EQ | Total |
| 1 | 2 |  |  | 1 |  |  |
| 2 |  |  |  | 2 | 2 |  |
| 3 | 1 |  |  | 1 |  |  |
| 4 | 1 |  |  | 1 |  |  |
| 5 |  | 1 | 1 | 1 | 1 |  |
| 6 | 1 |  |  |  |  |  |
| 7 | 1 | 1 |  | 1 | 1 |  |
| 8 | 1 |  |  | 1 |  |  |
| 9 |  |  | 1 | 1 |  |  |
| 10 | 1 |  |  | 1 |  |  |
| 11 | 1 | 1 |  | 1 | 1 |  |
| Sub Total | 9 | 3 | 2 | 11 | 5 |  |
| All weights Low | 7 | 5 | 3 | 4 | 3 |  |
| weighted | 63 | 15 | 6 | 44 | 15 | **128** |

### Estimate Variables

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Value (Text) | Factor | Description |
| LANG | 3rd Generation Language |  |  |
| PREC | prec, Low |  | Precedentedness |
| FLEX | flex, High |  | Development Flexibility |
| RELY | rely, Very\_High |  | Required Software Reliability |
| DATA | data, Low |  | Data Size |
| CPLX | cplx, Nominal |  |  |
| RUSE | ruse, Low |  |  |
| DOCU | docu, Nominal |  | Documentation to match life cycle needs |
| RESL | resl, High |  |  |
| TEAM | team, High |  |  |
| ACAP | acap, High |  | Analyst Capability |
| PCAP | pcap, High |  | Programmer Capability |
| PCON | pcon, Very\_High |  | Personnel Capability |
| APEX | apex, Nominal |  | Application Experience |
| PEXP | pexp, Low |  | (PLEX?) Platform Experience |
| LTEX | ltex, Low |  | Language and Tool Experience |
| PMAT | pmat, High |  |  |
| TIME | time, Very\_High |  | Execution Time |
| STOR | stor, Nominal |  | Main Storage Constraint |
| PVOL | pvol, Low |  | Platform Volatility |
| TOOL | tool, Nominal |  | Use of Software Tools |
| SITE | site, Extra\_High |  | Multisite Development |
| SCED | Nominal | 1.00 | Schedule |

### Early Design Calculations

lorem

|  |  |  |
| --- | --- | --- |
| RCPX | 1.33 | RELY + DATA + CPLX + DOCU |
| RUSE | 0.95 | RUSE |
| PDIF | 1.29 | TIME + STOR + PVOL |
| PREX | 1.12 | APEX + PLEX + LTEX |
| PERS | 0.63 | ACAP + PCAP + PCON |
| FCIL | 0.73 | TOOL + SITE |
| SCED | 1 | SCED |
| **EArch** | 0.83954965752 |  |
|  |  |  |
| **Size (KSLOC)** | 6.4 |  |
| Size (UFP) | 128 | Unadjusted Function Points |
| UFP->Lines of Code | 50 | Conversion factor from UFP to SLOC |
| PREC | 0.01 |  |
| FLEX | 0.03 | Flexibility |
| RESL | 0.03 | Risk Resolution |
| TEAM | 0.04 | Team cohesiveness and communication |
| PMAT | 0.03 | Process Maturity |
| **Process Exponent** | 1.15 |  |
|  |  |  |
| **Effort** | 17.39 | staff-months |
| **TDEV** | 2.82 | time to develop |
| **TDEV Early Design** | 3.47 |  |
| **TDEV 1997 Calibration** | 2.50 |  |

### Discussion

lorem

# Architecture Elaboration Plan

The following subsections detail the tasks that will be completed during the elaboration phase of the project.

## Revise Vision Document

## Revise Project Plan

## Create Formal Specification

## Create Architectural Design

## Create Test Plan

## Conduct Technical Inspection

## Create Executable Architecture Prototype

# Implementation Plan: Deliverables

Deliverables statement

## Action Items

adf

## Technical Inspection Letters

asdf

## Component Design Document

adsf

## User Manual

asfd

## Source Code

af

## Assessment Evaluation

asdf

## Project Evaluation

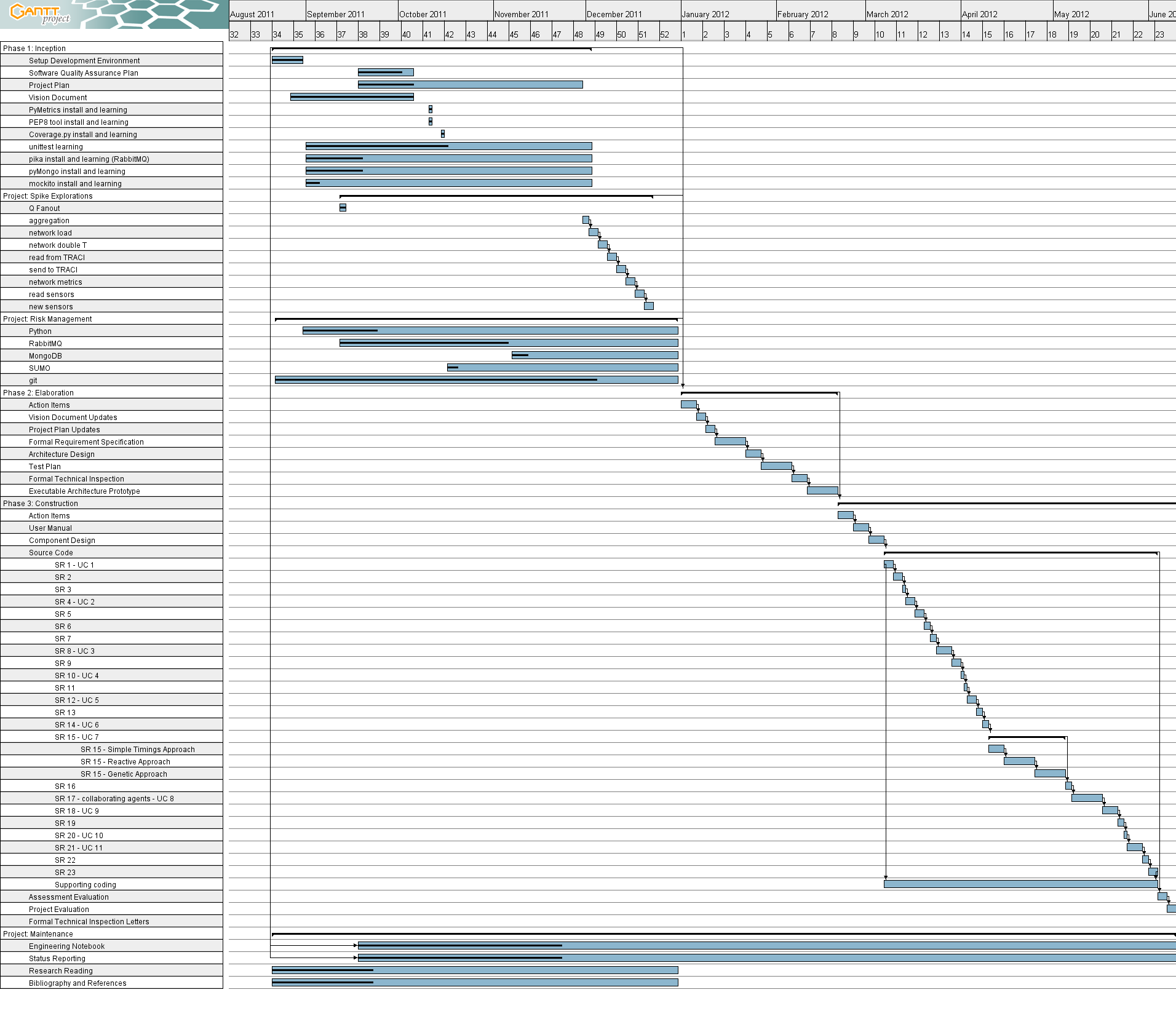
asdf

## References

adf

# Implementation Plan: Work Breakdown Structure

Words about the work breakdown structure here…



## Inception Phase

asdf

## Elaboration Phase

asdf

## Production Phase

asdf

# Maintenance?