Project Plan

For KDD- Service based Numerical Entity Searcher

(KSNES)

Version 2.0

Submitted in partial fulfillment of the Masters of Software Engineering degree.

Naga Sowjanya Karumuri
CIS 895 – MSE Project
Department of Computing and Information Sciences
Kansas State University

Committee Members

Dr. William. H. Hsu

Dr. Torben Amtoft

Dr. Mitchell Neilsen

Change Log

Version #	Changed By	Release Date	Change Description
Version 1.0	Sowjanya	02/08/09	Initial Release
Version 1.1	Sowjanya	02/16/09	Added new cost estimation model
Version 2.0	Sowjanya	03/21/09	Make changes to Section 2.1 and the project schedule and added Section 4.

List of Figures

List of Tables

Table of Contents

1. Test Breakdown

1.1. Project Phases

The Unified Process divides the project into three phases: Inception, Elaboration and Production.

1.1.1. Inception Phase

The inception phase is the first and smallest phase in the project. In this phase the project scope, use cases diagrams, functional requirements, project overview, project design, risks in the project and preliminary project schedule are analyzed before beginning the project development and documented in three different documents – Vision Document, Project Plan and Software Quality Assurance Plan .

Vision Document outlines the scope, purpose, goals and functional requirements of the project. In the Project Plan document, effort estimation and project schedule details are logged. The list of the required documentation, standards and conventions, steps to maintain quality of the project are recorded in the Software Quality Assurance Plan.

This phase is completed when the developer gets all the documentation to be reviewed and approved by the supervisory committee. To complete the phase, an initial prototype developed by the developer is given as the first presentation.

1.1.2. Elaboration Phase

In the Elaboration Phase the architectural design plan of the system is developed. The main objective of this phase is to capture the healthy majority of the system requirements using OCL diagrams and develop a test plan to specify the testing techniques as well as the method of documenting, tracking and debugging. Along with this, the documents developed in the inception phase are updated as per the supervisory committee suggestions from the first presentation. Two formal technical inspectors, who are the fellow MSE students, will perform technical inspections of the architectural design and provide feedback from their findings. Including the suggestions given by the supervisory committee in the previous presentation, a second prototype will be developed in this stage by modifying the

first prototype that will support the key system functionalities and exhibit the right behavior in terms of performance, scalability and cost.

The second version of the documentation is submitted to the supervisory committee for changes and approval. The second presentation is given at the end of this phase with the demonstration of a stable and executable architecture baseline.

1.1.3. Production Phase

The Production phase is the largest phase in the project where in the complete system is build and coded to develop an executable release of the software. The developer also documents the code and deliverables like User Manual, Project Evaluation, Test Logs etc.,

The developer submits the executable version of the project and the supporting documents for the supervisory committee for reviewing and approval. The final presentation will be given at the end of this phase with a demonstration of the complete project.

1.2. Project Schedule

The Gantt chart below presents the schedule for the KDD-Numerical Sub-Chunker project. A separate document is submitted for better view of the chart.

ID	Task Name	Duration	Start	Finish
1	**Phase One - Inception Phase**	**26 days**	**Wed 1/28/09**	**Thu 2/26/09**
2	**Documentation**	**24 days**	**Thu 1/29/09**	**Wed 2/25/09**
3	Vision Documentation 1.0	19 days	Thu 1/29/09	Thu 2/19/09
4	Software Quality Assurance Plan 1.0	3 days	Thu 2/12/09	Mon 2/16/09
5	Project Plan 1.0	4 days	Sat 2/7/09	Wed 2/11/09
6	Presentation 1 Slides	9 days	Mon 2/16/09	Wed 2/25/09
7	**Coding**	**17 days**	**Tue 2/3/09**	**Mon 2/23/09**
8	Implementing Numerical Sub-Chunker	10 days	Fri 2/13/09	Wed 2/25/09
9	POS Tagger installation	2 days	Sat 2/7/09	Mon 2/9/09
10	Preparing logic for Phrase Extractor	14 days	Mon 2/9/09	Wed 2/25/09
11	Project Prototype 1.0	1 day	Mon 2/23/09	Mon 2/23/09
12	**Presentation 1**	**1 day**	**Thu 2/26/09**	**Thu 2/26/09**
13	**Phase Two - Elaboration Phase**	**25 days**	**Fri 2/27/09**	**Thu 3/26/09**
14	Action Items	1 day	Thu 2/26/09	Thu 2/26/09
15	**Documentation**	**21 days**	**Fri 2/27/09**	**Sun 3/22/09**
16	Software Quality Assurance Plan 2.0	3 days	Fri 2/27/09	Sun 3/1/09
17	Vision Document 2.0	3 days	Mon 3/2/09	Wed 3/4/09
18	Project Plan 2.0	1 day	Mon 3/9/09	Mon 3/9/09
19	Formal Requirements Specification	3 days	Tue 3/10/09	Thu 3/12/09
20	Architecture Design 1.0	4 days	Tue 3/10/09	Fri 3/13/09
21	Test Plan 1.0	3 days	Mon 3/16/09	Wed 3/18/09
22	Inspection Checklist 1.0	3 days	Thu 3/19/09	Sat 3/21/09
23	Formal Technical Evaluation	5 days	Wed 3/18/09	Sun 3/22/09
24	Presentation 2 slides	1 day	Mon 3/23/09	Mon 3/23/09
25	**Coding**	**21 days**	**Fri 2/27/09**	**Sun 3/22/09**
26	Connecting GUI to tagger	6 days	Fri 2/27/09	Tue 3/3/09
27	POS Tagger service setup	4 days	Sun 3/1/09	Wed 3/4/09
28	Numerical Phrase Extractor Implementation	14 days	Thu 3/5/09	Sat 3/21/09
29	Project Prototype 2.0	2 days	Sat 3/21/09	Sun 3/22/09
30	Presentation 2	1 day	Mon 3/23/09	Mon 3/23/09
31	**Phase Three - Production Phase**	**26 days**	**Tue 3/24/09**	**Fri 4/24/09**
32	Action Items	1 day	Tue 3/24/09	Tue 3/24/09
33	**Documentation**	**24 days**	**Tue 3/24/09**	**Wed 4/22/09**
34	Software Quality Assurance Plan 3.0	3 days	Wed 3/25/09	Fri 3/27/09
35	Vision Document 3.0	2 days	Fri 3/27/09	Mon 3/30/09
36	Project Plan 3.0	3 days	Tue 3/31/09	Thu 4/2/09
37	Component Design	2 days	Fri 4/3/09	Mon 4/6/09
38	Assessement Evaluation	4 days	Tue 4/7/09	Fri 4/10/09
39	User's Manual	1 day	Sun 4/12/09	Sun 4/12/09
40	Formal Technical Inspection Letters	3 days	Mon 4/13/09	Wed 4/15/09
41	References	1 day	Thu 4/16/09	Thu 4/16/09
42	Project Evaluation	11 days	Tue 4/7/09	Sat 4/18/09
43	Presentation 3 Slides	4 days	Mon 4/20/09	Thu 4/23/09
44	**Coding**	**20 days**	**Wed 3/25/09**	**Sat 4/18/09**
45	Patching all the modules	17 days	Wed 3/25/09	Wed 4/15/09
46	Completing GUI	6 days	Mon 3/30/09	Mon 4/6/09
47	Setting up the services	4 days	Tue 4/7/09	Sat 4/11/09
48	Sub-Chunked Output	4 days	Sun 4/12/09	Wed 4/15/09
49	Deliverables Executable	4 days	Sat 4/18/09	Wed 4/22/09
50	**Testing**	**45 days**	**Fri 2/27/09**	**Tue 4/21/09**
51	Functionality Testing	28 days	Wed 3/11/09	Tue 4/14/09
52	Testing Evaluation	8 days	Wed 4/15/09	Thu 4/23/09
53	**Presentation 3**	**1 day**	**Fri 4/24/09**	**Fri 4/24/09**

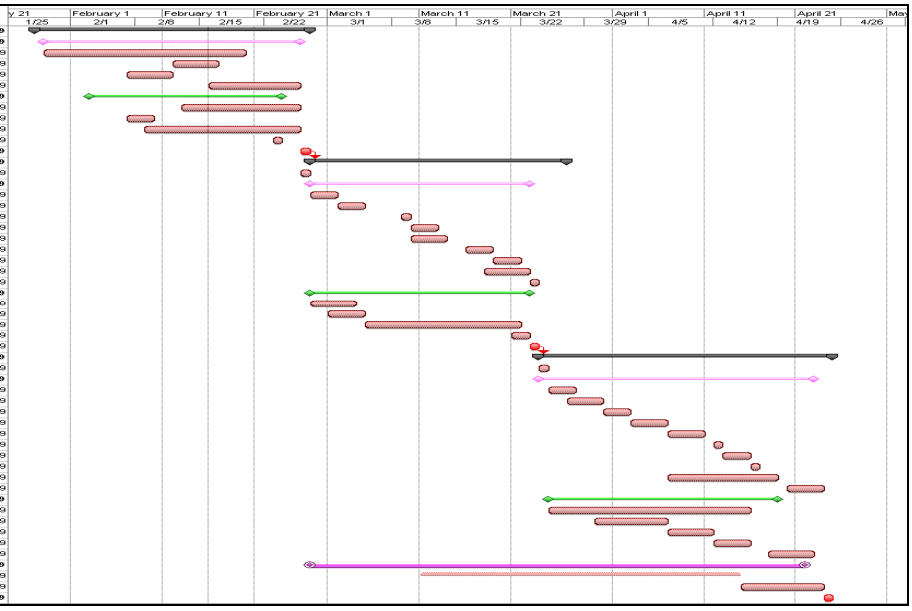

Figure 1: Project Schedule

2. Cost Estimate

To estimate the effort, cost and schedule for a project many different measures can be taken but the most efficient method is the COCOMO II model as it considers the factors like complexity, reliability, database and memory usage, experience of the developer into consideration into account during calculations. As these are the major factors that would affect the development of a software project this model is considered to be the right choice to estimate the development cost.

2.1.COCOMO II

Original COCOMO model has been very successful, but it does not apply to newer software development practices as well as it does to traditional practices. COCOMO II is the latest major extension to the original COCOMO model published in 1981. COCOMO II is being developed by Dr. Barry Boehm and his students at USC and it targets the software projects of the 1990s and 2000s, and will continue to evolve over the next few years. It consists of three sub-models each one offering increased fidelity the further along one is in the project planning and design process. Listed in increasing fidelity, these sub-models can be explained as follows:

- The Application Composition Model – Suitable for projects built with modern GUI-builder tools based on new Object Points.
- The Early Design Model – This model can be used to get a rough estimate of a project's cost and duration before determining its entire architecture. It uses a small set of new cost drivers and new estimating equations based on unadjusted function points or KSLOC.
- The Post-Architecture Model – This is the most detailed COCOMO II model. It is used before we develop the architecture.

The present KSNES project can be considered as The Post-Architecture Model as we can see its architecture from Vision document. The following formulae are used for the cost estimation:

$$\text{Effort} = 2.45 * \text{EAF} * (\text{KSLOC})^{1.09}$$

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

Where in : Effort = the number of person months (PM)
Time = Duration time in months for project
KSLOC = Estimated number of source lines of code for the project (expressed in thousands)

EAF = Effort Adjustment Factor

Effort Adjustment Factors are 15 in number and their values differ within a given range. Each adjustment factor may fall in one of the categories – very low, low, nominal, high, very high, extra high and based on these categories a value is given to EAF. EAF is the product of the values of the 15 factors shown in the below table:

Identifier	**EAF**	**Possible Range Of Values**
RELY	Required Software Reliability	0.75 – 1.40
DATA	Size of Application Database	0.94 – 1.16
CPLX	Complexity of the Product	0.70 – 1.65
TIME	Run-time Performance Requirements	1.00 – 1.66
STOR	Memory Constraints	1.00 – 1.56
VIRT	Virtual Machine Volatility	0.87 – 1.30
TURN	Required Turnabout Time	0.87 – 1.15
ACAP	Analyst Capability	1.46 – 0.71
AEXP	Applications Experience	1.29 – 0.82
PCAP	Software Engineer Capability	1.42 – 0.70
VEXP	Virtual Machine Experience	1.21 – 0.90
LEXP	Programming Language Experience	1.14 – 0.95
TOOL	Use of Software Tools	1.24 – 0.82
MODP	Use of Modern Software Practices	1.24-0.83
SCED	Required Development Schedule	1.23 – 1.10

Table 1: COCOMO Effort Adjustment Factors

Based on the KDD-Numerical Sub-Chunker project these above defined factors can be assigned a value in the range specified above. In the table given below each identifier is classified and given a value. The reason for choosing a specific value for each factor is also mentioned in the table.

Identifier	Classification	Value	Reasoning
RELY	Low	0.88	Project is not safety critical, and does not have to be completely reliable
DATA	Low	0.94	Data used by the project is only the user's input which is relatively small in size
CPLX	High	1.02	Processing natural text and extracting the required features from them are a complicated
TIME	Nominal	1.00	Response time is important but not a overly critical issue
STOR	Nominal	1.00	Tagging and Numerical Phrase Extraction will not require high memory usage
VIRT	Low	0.87	Low complexity of the hardware and software
TURN	Low	0.87	The turnaround time on results is low as the project is developed by a single developer
ACAP	High	0.86	Developer has 3+ years experience in Software Engineering
AEXP	Nominal	1.00	Developer has 2+ years experience in applications development
PCAP	Nominal	1.00	Developer has 6 month of experience in the application area
VEXP	Low	1.10	Developer has 6 months of experience and is learning Java Virtual Machine
LEXP	Low	1.15	Developer has 6 months of experience and is learning Java
TOOL	Nominal	1.00	Moderate experience with the tools being used
MODP	Very High	0.83	Developer has 4+ years experience in employing modern Software Engineering Practices
SCED	Nominal	1.00	Project is tightly scheduled but is a bit flexible

Table 2: Project Effort Adjustment Factor Values

EAF is the product of the values of the adjustment factors. From the above table we have the calculated value of EAF as 0.58. To make the system work, coding is required for making the GUI, patching the modules in the system, making the client-server system work. Based on this the estimated size of the project will be around 1 KLOC.

Using these figures, Effort and Time can be calculated as:

$$\text{Effort} = 2.45 * 0.58 * 1^{1.09} = 1.42$$

$$\text{Time} = 2.5 * 1.42^{0.38} = 2.85$$

From the above calculations we can see that the COCOMO model estimates the effort to be 1.42 person months to be required to complete the project. From the Time value calculation the project should be taking around 2.85 months to complete it and is almost near to the project schedule shown in the Gantt chart.

There is a limitation with the COCOMO II model since there could be a slight variation in the above calculated values based in the misjudged EAF values, increased project complexity and scope of the project.

3. Architecture Elaboration Plan

Before presenting the second presentation, elaboration phase comes to end by completing all the required documents and artifacts. The details about the documents to be submitted are as follows:

3.1. Vision Document Revision

A revised version of the Vision Document is required created to include the changes suggested by the committee after the first presentation. This version also includes the complete list of requirements which are ranked based on the priority. This is submitted to the major professor for approval.

3.2. Project Plan Revision

Project Plan document is revised as per the feedback from the committee members in the first presentation. Based on the changes in the schedule and the cost estimation, the Gantt chart and the COCOMO estimates are updated. This is submitted to the major professor for approval.

3.3. Architectural Design

The Architectural Design document explains the components and the scenarios of the present system using UML diagrams. This is submitted to the major professor for approval.

3.4. Prototype Development

Based on the suggestions from the supervisory committee during the first project presentation and by adding new functionalities a prototype is developed. This prototype would be an extended version of the prototype developed in the inception phase. This is submitted to the major professor for approval.

3.5. Test Plan

To ensure that all the requirements specified in the Vision document are satisfied, a test plan is developed that outlines all the testing activities required. This document also contains the instructions on how to evaluate the product and will be submitted to the major professor for approval.

3.6. Formal Technical Inspections

Two yet to be determined fellow MSE students will act as the formal inspectors of the project. These inspectors will use a formal inspection checklist that will be produced during the Elaboration phase based on their findings.

3.7. Formal Requirements Specification

Using the USE (UML – based Specification Environment) tool, at least one of the modules of the project will be specified using OCL. This is submitted to the major professor for approval.

4. Software Production Plan

4.1. Test Plan Revision

As per the suggestions form the committee members after presentation 2, the changes will made to the test plan and will be submitted to the major professor.

4.2. Architectural Design Revision

The revision of the Architectural Design will be done after the second presentation as per the suggestions form the committee members. The major professor will approve the changes to the document.

4.3. Component Design

UML will be used to convey detailed information about the software components in the component design document. It will include all the attributes and methods for the classes in the project and this document is submitted to the majo professor for approval.

4.4. Final Software Executable

The protytpe developed during the Architecture Elaboration Phase will be developed and improved with additional features during the Production Phase. The additional features would be the required functionalitiy and the suggestions form the committee members.

4.5. Formal Technical Inspections

Two MSE students will provide feedback to the project after inspecting the formal specifications. Both the inspector will report their formal inspection checklist based on their findings.

4.6. User’s Manual

To help the user to use the system , a user manual will be provided by the software developer. This manual will guide to use the different sections of the system and lists the various troubleshooting problems and solutions.

4.7. Test Assessment

At the end of the software development, the developer will run the test cases logged in the Test Plan document and will record the results. The Test Assessment document contains the results of running these tests.

4.8. Technical Instructins for Reuse and Extension

The developer also provides a written copy of the guidelines that help other developers to use the current system for other MSE projects. The document also contains the features that could be added to the project to adapt it for different uses.

4.9. Project Assessment

Project assessment is the document that the developer writes at the completion of the software development and the testing about his/her opinion on the project. The document will describe the issues that went well, the scope of improvement and the things that did not work,. This will contain the final metrics for the project.