Project Plan

For agentTool III (Dynamic)

Version 1.0

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

Binti Sepaha
CIS 895 – MSE Project
Kansas State University
# TABLE OF CONTENTS

1. Tasks Breakdown ........................................................................................................................ 3  
   1.1. Inception Phase ..................................................................................................................... 3  
   1.2. Elaboration Phase.................................................................................................................. 3  
   1.3. Production Phase ................................................................................................................... 3  
2. Cost Estimate ............................................................................................................................... 4  
   2.1. COCOMO ............................................................................................................................. 4  
3. Architecture Elaboration Plan ..................................................................................................... 6  
   3.1. Revision of Vision Document ............................................................................................... 6  
   3.2. Revision of Project Plan ........................................................................................................ 6  
   3.3. Architecture Design .............................................................................................................. 6  
   3.4. Development of Prototype .................................................................................................... 6  
   3.5. Test Plan ................................................................................................................................ 7  
   3.6. Formal Technical Inspection ................................................................................................. 7  
   3.7. Formal Requirements Specification ...................................................................................... 7
1. Tasks Breakdown

1.1. Inception Phase

The inception phase focuses on defining the requirements specification. For this purpose, a vision document will be created which gives an overview of the project and elaborates the requirements for the software. The major use cases are defined and elaborated in the requirements analysis.

A project plan will be developed in the inception phase that gives the schedule of all activities required to complete the project. A software quality plan is also laid out in this phase to ensure that the quality of the product is maintained throughout the development life cycle. A prototype is also developed with the motive to show the feasibility and the look and feel of the project.

The inception phase completes when the prototype along with the other documents are approved by the committee.

1.2. Elaboration Phase

In the elaboration phase, the architecture of the software will be finalized. The documents from phase I will be reviewed and completed as desired by the committee. A test plan would be developed which will mention the complete testing process including the reporting of bugs and solving them.

Formal requirements specification will be generated for a part of the project, if not for the entire system. The language used for this would be OCL. Another prototype would be developed to demonstrate more features in the tool. Two technical inspectors will review the design in this phase and report with their findings.

The elaboration phase will be marked as ‘complete’ when the committee reviews the documents and approves of the prototype, with or without changes.

1.3. Production Phase

The production phase deals with the complete implementation and testing of the software. In this phase the user will develop the code and make sure that it is fully documented. The code will be tested entirely to check if all the requirements are met. The testing results will be documented. A user manual will also be created that aids the
user to install and use the tool efficiently. A component design that describes the system at low level will also be developed.

The production phase is marked as ‘complete’ when the committee reviews the documents and approves of the entire source code. The final presentation should be given at the end of the phase.

The Gantt chart below gives a schedule for the completion of the above-defined tasks for each phase. (You can increase the screen view size in percentage to view this clearly). The shaded area is the non-working time period.

![Gantt Chart](image)

**Figure 1. Project Schedule**

2. **Cost Estimate**

2.1. **COCOMO**

This model is based on Barry Boehm's Constructive Cost Model (COCOMO). This is the top-level model, Basic COCOMO, which is applicable to the large majority of software projects. This is a simple on-line cost model for estimating the number of person-months required to develop software. The model also estimates the development schedule in months and produces an effort and schedule distribution by major phases.

Boehm says: "Basic COCOMO is good for rough order of magnitude estimates of software costs, but its accuracy is necessarily limited because of its lack of factors to account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and other project attributes known to have a significant influence on costs." Due to these facts, the estimate we find out may not be as accurate, but would give us a rough idea about the schedule of the project.
According to Boehm’s classification, this project falls under the ‘organic’ modes of development. The following equations are used to calculate effort and time:

\[
\text{Effort} = 3.2 \times \text{EAF (Size)}^{1.05}
\]

\[
\text{Time (in months)} = 2.5 \times \text{(Effort)}^{0.38}
\]

To calculate effort one needs to estimate the Size and EAF values. The Size is measured in KLOC. The EAF value stands for effort adjustment factor and is the product of 15 adjustment factors. Each adjustment factor is classified as very low, low, normal, high, or very high. The value of each adjustment factor lies within a range and the classification will determine where on the range the value will falls. The table below lists all the adjustment factors and their corresponding ranges.

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>EFFORT ADJUSTMENT FACTOR</th>
<th>RANGE</th>
<th>MY CLASSIFICATION AND VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELY</td>
<td>Required reliability</td>
<td>0.75 - 1.40</td>
<td>Normal</td>
</tr>
<tr>
<td>DATA</td>
<td>Database size</td>
<td>0.94 – 1.16</td>
<td>Very low</td>
</tr>
<tr>
<td>CPLX</td>
<td>Product complexity</td>
<td>0.70 – 1.65</td>
<td>High</td>
</tr>
<tr>
<td>TIME</td>
<td>Execution time constraint</td>
<td>1.00 – 1.66</td>
<td>Normal</td>
</tr>
<tr>
<td>STOR</td>
<td>Main storage constraint</td>
<td>1.00 – 1.56</td>
<td>Normal</td>
</tr>
<tr>
<td>VIRT</td>
<td>Virtual machine volatility</td>
<td>0.87 – 1.30</td>
<td>Low</td>
</tr>
<tr>
<td>TURN</td>
<td>Computer turnaround time</td>
<td>0.87 – 1.15</td>
<td>Low</td>
</tr>
<tr>
<td>ACAP</td>
<td>Analyst capability</td>
<td>1.46 – 0.71</td>
<td>High</td>
</tr>
<tr>
<td>AEXP</td>
<td>Applications experience</td>
<td>1.29 – 0.82</td>
<td>Normal</td>
</tr>
<tr>
<td>PCAP</td>
<td>Programmer capability</td>
<td>1.42 – 0.70</td>
<td>High</td>
</tr>
<tr>
<td>VEXP</td>
<td>Virtual machine experience</td>
<td>1.21 – 0.90</td>
<td>Normal</td>
</tr>
<tr>
<td>LEXP</td>
<td>Language experience</td>
<td>1.14 – 0.95</td>
<td>High</td>
</tr>
<tr>
<td>MODP</td>
<td>Use of modern practices</td>
<td>1.24 – 0.82</td>
<td>High</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of software tools</td>
<td>1.24 – 0.83</td>
<td>High</td>
</tr>
<tr>
<td>SCED</td>
<td>Required development schedule</td>
<td>1.23 – 1.10</td>
<td>High</td>
</tr>
</tbody>
</table>
The EAF value evaluated to 1.36. I estimated the size to be 3500 LOC based on the previous versions of the tool. The effort evaluates to:

\[
\text{Effort} = 3.2 \times 1.36 \times 2.5^{1.05} = 11.4 \text{ staff months}
\]

The time can now be calculated as:

\[
\text{Time} = 2.5 \times 12.9^{0.38} = 6.3 \text{ months}
\]

We will not consider the Time we have calculated here because this calculation takes teamwork into account where there are many developers. This project will only have one developer with a little interaction initially with another developer to come up with the object model. The Effort figure may be large for such a project since the COCOMO model is designed for large projects with the involvement of several large teams in an industrial setting, so it tends to over-estimate this value. I would estimate that 8-9 staff months for Effort would be a more realistic value for this project.

3. **Architecture Elaboration Plan**

   The following documents should be complete before the second presentation is made.

   **3.1. Revision of Vision Document**

   After the first presentation, changes as required by the committee should be made in the Vision Document. The major professor should approve these documents.

   **3.2. Revision of Project Plan**

   According to the changes suggested by the committee after the first presentation, the project plan should be modified to include those changes. These changes might be regarding the cost estimate or schedule. Also an extra section about the implementation plan should be added before the second presentation.

   **3.3. Architecture Design**

   Based on the use case diagram in the vision document and using other UML diagrams, an architecture design should be developed for the second phase. This design should be approved by the committee before proceeding to the actual complete implantation of the tool. It should also undergo formal technical inspection by two other MSE students.

   **3.4. Development of Prototype**

   This prototype will include the demonstration of the critical requirements identified in the Vision Document.
3.5. **Test Plan**
A complete test plan will be developed indicating the testing techniques used and the way bugs will be reported and solved. Unit testing, integration testing and system testing will be performed. This will be done to test whether all the requirements specified in the vision document are met or not.

3.6. **Formal Technical Inspection**
Deepti Gupta and Dominic Gelinas, two other MSE Students will review the architecture design produced by the developer and submit a report on their findings.

3.7. **Formal Requirements Specification**
The part of the project describing Sequence Diagrams will be formally specified using OCL. The tool used will be USE (UML-based Specification Environment). This will include operations like adding or deleting a class role, messages, alternative decisions etc. Messages would be checked for their source and destinations. Eg: A message to self should have the same source and destination.