Project Plan

For a GMoDS-based Runtime Agent Role Interpreter

Version 1.0

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

Kyle Hill
CIS 895 – MSE Project
Kansas State University
Table of Contents

1 Introduction............................................................................................................................................. 3
2 Work Breakdown Structure ................................................................................................................. 3
  2.1 Inception Phase .............................................................................................................................. 3
  2.2 Elaboration Phase ........................................................................................................................... 4
  2.3 Production Phase ............................................................................................................................ 4
3 Cost Estimate ........................................................................................................................................ 4
  3.1 COCOMO ......................................................................................................................................... 4
4 Architecture Elaboration Plan .............................................................................................................. 6
  4.1 Revise Vision Document ................................................................................................................... 6
  4.2 Revise Project Plan ............................................................................................................................ 6
  4.3 Create Formal Specification ............................................................................................................. 6
  4.4 Create Architectural Design ............................................................................................................ 7
  4.5 Create Test Plan ............................................................................................................................... 7
  4.6 Conduct Technical Inspection .......................................................................................................... 7
  4.7 Create Executable Architecture Prototype ...................................................................................... 7
1 Introduction

This document serves as the initial project plan for the GMoDS-based Runtime Agent Role Interpreter project. It outlines the basic work breakdown structure, provides a cost estimate using the COCOMO estimation model, and describes the activities to be performed during the elaboration phase of this project.

2 Work Breakdown Structure

This project will follow a basic iterative process consisting of three phases: Inception, Elaboration, and Production. Within each phase, requirements gathering, design, and construction activities will occur. The first phase will be completed during the fall of 2010, while the other two phases will be completed during the spring of 2011. Figure 1 below shows a basic work breakdown structure in the form of a Gantt chart.

![Basic Project Gantt Chart](image)

2.1 Inception Phase

During the inception phase of this project, a vision document, a project plan, a Software Quality Assurance (SQA) plan, and an initial prototype will be created. The vision document will provide a project overview and detail the critical software requirements. The project plan will detail the work to be accomplished during each phase of the project. The SQA plan will outline documentation that will be generated as part of the software project, as well as provide an overview of the standards and conventions to be followed during the development process. Finally, an initial software prototype will be created to demonstrate the feasibility of the project. These work products will be presented to the supervisory committee once the initial prototype has been completed.
The inception phase will end when these deliverables have been approved by the supervisory committee.

2.2 Elaboration Phase

During the elaboration phase of this project, the original vision document and project plan documents will be revised according to input received during construction of the initial prototype and from committee members during the project presentation. The complete design of the project architecture will be provided in the form of UML diagrams. In addition, at least one component of the project will be specified using a formal specification methodology. During this phase, a software test plan will also be created to document the project’s testing process. Additionally, two technical inspectors will review the software architecture and provide feedback. Finally, an executable architecture prototype will be constructed and presented to the supervisory committee.

The elaboration phase will end when these documents and the executable architecture prototype have been approved by the supervisory committee.

2.3 Production Phase

During the production phase of this project, low-level component design will be completed, and the final software deliverables will be constructed. In addition, each component will be unit tested and the final project will undergo integration testing according to the test plan provided during the elaboration phase. Once testing is completed, a user manual will be written and the final project evaluation will take place.

The production phase will end when the final project deliverables has been presented and approved by the supervisory committee.

3 Cost Estimate

3.1 COCOMO

The COCOMO software cost estimation model, created by Barry Boehm in 1981, is used to estimate the cost of this project in terms of both effort and time. The GMoDS-based Runtime Agent Role Interpreter will be a stand-alone software component of average complexity. The software will be implemented by a single developer with good experience and with somewhat flexible requirements. Based on the software classifications provided by the COCOMO model, this project is considered organic. According to this classification, the effort and time estimations are provided by the following equations:

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

\[
\text{Time (in months)} = 2.5 \times (\text{Effort})^{0.38}
\]
The Size parameter is measured as the estimated number of significant lines of code, in thousands, (KSLOC). Based on experience with similar projects and a review of other existing MSE projects, a safe upper bound for the Size parameter would be 5.0.

The EAF parameter is the product of 15 different adjustment factors listed in Figure 2 below. These factors are classified as one of Very Low, Low, Nominal, High, Very High, and Extra High. A particular adjustment factor’s classification determines where within the given range that factor’s value lays.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Effort Adjustment Factor</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELY</td>
<td>Required Reliability</td>
<td>0.75 - 1.40</td>
</tr>
<tr>
<td>DATA</td>
<td>Database Size</td>
<td>0.94 - 1.16</td>
</tr>
<tr>
<td>CPLX</td>
<td>Product Complexity</td>
<td>0.70 - 1.65</td>
</tr>
<tr>
<td>TIME</td>
<td>Execution Time Constraint</td>
<td>1.00 - 1.66</td>
</tr>
<tr>
<td>STOR</td>
<td>Main Storage Constraint</td>
<td>1.00 - 1.56</td>
</tr>
<tr>
<td>VIRT</td>
<td>Virtual Machine Volatility</td>
<td>0.87 - 1.30</td>
</tr>
<tr>
<td>TURN</td>
<td>Computer Turnaround Time</td>
<td>0.87 - 1.15</td>
</tr>
<tr>
<td>ACAP</td>
<td>Analyst Capability</td>
<td>0.71 - 1.46</td>
</tr>
<tr>
<td>AEXP</td>
<td>Applications Experience</td>
<td>0.82 - 1.29</td>
</tr>
<tr>
<td>PCAP</td>
<td>Programmer Capability</td>
<td>0.70 - 1.42</td>
</tr>
<tr>
<td>VEXP</td>
<td>Virtual Machine Experience</td>
<td>0.90 - 1.21</td>
</tr>
<tr>
<td>LEXP</td>
<td>Language Experience</td>
<td>0.95 - 1.14</td>
</tr>
<tr>
<td>MODP</td>
<td>Use of Modern Practices</td>
<td>0.82 - 1.24</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of Software Tools</td>
<td>0.83 - 1.24</td>
</tr>
<tr>
<td>SCED</td>
<td>Required Development Schedule</td>
<td>1.10 - 1.23</td>
</tr>
</tbody>
</table>

Figure 2 – COCOMO Effort Adjustment Factors

Based upon the COCOMO documentation and experience with similar projects, the following adjustment factor classifications were chosen.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Classification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELY</td>
<td>Low</td>
<td>0.88</td>
</tr>
<tr>
<td>DATA</td>
<td>Nominal</td>
<td>1.00</td>
</tr>
<tr>
<td>CPLX</td>
<td>High</td>
<td>1.15</td>
</tr>
<tr>
<td>TIME</td>
<td>Nominal</td>
<td>1.00</td>
</tr>
<tr>
<td>STOR</td>
<td>Nominal</td>
<td>1.00</td>
</tr>
<tr>
<td>VIRT</td>
<td>Low</td>
<td>0.87</td>
</tr>
<tr>
<td>TURN</td>
<td>Low</td>
<td>0.87</td>
</tr>
<tr>
<td>ACAP</td>
<td>Low</td>
<td>1.19</td>
</tr>
<tr>
<td>AEXP</td>
<td>Nominal</td>
<td>1.00</td>
</tr>
<tr>
<td>PCAP</td>
<td>High</td>
<td>0.86</td>
</tr>
<tr>
<td>VEXP</td>
<td>High</td>
<td>0.90</td>
</tr>
<tr>
<td>LEXP</td>
<td>High</td>
<td>0.95</td>
</tr>
<tr>
<td>MODP</td>
<td>High</td>
<td>0.82</td>
</tr>
</tbody>
</table>
In general, this project has low hardware constraints and reliability requirements. This is a somewhat complex product, as it will be used as an agent architecture within a multiagent system. However, to offset this, a small and experienced development team consisting of myself will handle development of the system. I am confident in my abilities to program within the chosen environment.

Based on these selected adjustment factors, the EAF value is 0.55. Thus, the Effort value evaluates to the following:

\[
\text{Effort} = 3.2 \times 0.55 \times (5)^{1.05} = 9.54 \text{ staff-months}
\]

Time, therefore evaluates to the following:

\[
\text{Time} = 2.5 \times (9.54)^{0.38} = 5.89 \text{ months}
\]

Based on my projected timeline for this project, 5.89 months seems like reasonable value, if a little on the high side.

4 Architecture Elaboration Plan

The following tasks will be completed during the elaboration phase of this project.

4.1 Revise Vision Document

The original vision document will be revised to provide a more complete representation of the system use cases and requirements. These changes will be based on knowledge gained during the initial prototype construction, as well as from feedback received from committee members during the first project presentation. The revised vision document will be submitted to the major professor for approval.

4.2 Revise Project Plan

The original project plan will be updated to reflect any schedule changes that have taken place during the course of the inception phase. Additional feedback provided by committee members will be incorporated as well. The revised project plan will be submitted to the major professor for approval.

4.3 Create Formal Specification

The model used for representing the role interpreter’s goal-capability map will be formally specified. This formal specification will be submitted to the major professor for approval.
4.4 Create Architectural Design
A complete architectural design will be created using UML diagrams to the component interface level. This design will be reviewed by the project technical inspectors and be submitted to the major professor for approval.

4.5 Create Test Plan
A test plan will be created that contains tests to show that the final software deliverables meet the requirements specified in the project vision document. This plan will include unit and integration testing criteria for all critical use cases. This plan will also include a set of test data to simulate real usage scenarios. This test plan will be submitted to the major professor for approval.

4.6 Conduct Technical Inspection
The project’s architectural design will be inspected by Shylaja Chippa and Mike Fraka. An inspection checklist will be developed and provided to the technical inspectors. Any feedback provided by the inspectors will be reviewed and incorporated into the project’s work products. The inspection checklist and reviewer feedback will be submitted to the major professor for approval.

4.7 Create Executable Architecture Prototype
An executable architecture prototype will be constructed to demonstrate that the provided architecture is suitable to satisfy all critical use cases outlined in the project vision document. This architectural prototype will be demonstrated to the supervisory committee by comment and approval.