1. **Team Name: Outsourced™**
2. **Team Leader for this deliverable: Dakota Methvin**
3. **Team Members: Tod Jones, Vince Seely**
4. **Meetings:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Time-date** | **Attendees** | **Agenda** | **Action Items (who will do what)** |
| 12:30 – 9/25 | All | Project Plan creation | Dakota – MPP creation  All – Task/subtask discovery, time estimation, sequencing and dependency analysis |
| 12:30 – 9/27 | All | Structured Analysis creation | Dakota – Group Report creation  All – Understand artifact creation, interpretation of generated graphs/reports |

1. **Weekly Time Logs:**

|  |  |  |
| --- | --- | --- |
| **Person** | **Total Time in minutes** | **Tasks** |
| Tod Jones | 117 | Understand artifacts, structured analysis |
| Vince Seely | 180 | Understand artifacts, structured analysis |
| Dakota Methvin | 149 | Understand artifacts, structured analysis, group report |
| **Total Time:** | 446 |  |

1. **Issues:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue Number** | **Discovery Date** | **Resolution Date ( Est. – Act. )** | **Responsible Person** | **Description ( Prob / Resolution )** |
| 001 | 9/11 | EST: 9/13  ACT: In progress | Vince | Dependency issue in existing codebase (.NET core 2.1 vs lab-available 2.0) |

1. **Files and their locations:**

|  |  |  |
| --- | --- | --- |
| **Filename** | **Location** | **Contents** |
| the-godfather repository | https://github.com/VinceSeely/the-godfather | Forked project from original source |

1. **Plans for Coming Week:**

* **Begin maintenance**
* **Test design**

1. **Comments:**

**Engineer 1:** *Tod Jones*

This week I dug through the Understand analysis of the-godfather to get a better understanding of the program structure and what can be improved. I also looked through some of the code to confirm what the Understand analysis was saying. I then discussed with Vince maintenance strategies and how to refactor/improving code and how to best divide the tasks. Potential fixes we discussed involved improving naming conventions, removing Dsharpplus dependencies and reducing cyclomatic complexity. I now feel we have enough to a plan to start the process of maintaining the code.

**Engineer 2:** *Vince Seely*

This week I looked into some places that maintenance and refactoring could be done in order to increase test coverage. I did this looking by looking at the code and by looking at the understand tool output. I personally got more by digging through the code than I did from using the Understand tool. This is because of what I was looking for: I was looking to make the system more testable and was looking at how to reduce the dependency on the external tools to run in order to decrease issues that would arise in actually setting up tests around some of these areas. I also helped with some of the time estimates for doing the refactoring, through looking at how extensive the change to removing the heavy dependency on DSharpPlus I have realized that most of the project has a lot of work to remove this and instead put in an interface that can be mocked out to fake the responses, so the estimate here is to do as much work as I think is feasible given how much time we have for this project. I also gave Tod some pointers on where I could help him understand the process of wrapping an external dependency and what some of the other changes that would be useful are.

**Engineer 3:** Dakota Methvin

This week I spent most of my efforts attempting to produce usable data from the SciTools Understand project. While unsuccessful, we were mainly looking to confirm our suspicions – that this project is not well structured. In the meetings, I was responsible for creating the deliverables and spearheading their submission. Since I have taken a semester-long class on Microsoft Project, this seemed like the most effective way to spend my time. To avoid a scenario like last week, I was more careful to get other team member’s contributions into a single document for merging.