1. **Team Name:** 1337 haXX
2. **Team Leader for this deliverable:** John Polus
3. **Team Members:**John Polus, Alex Jacobs, Erik Rasmussen, Logan Brincks, Steven Karrmann
4. **Meetings:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Time-date** | **Attendees** | **Agenda** | **Action Items (who will do what)** |
| 3/21/17 | All | Initial analysis of code | All: Broad examination of code and how to run it. Compile general software metrics. |
| 3/23/17 | All | Complete detailed analysis of project and code | John : Work on P-spec  Alex : Work on P-spec  Logan: Work on data dictionary  Steven: Structure Diagram  Erik: Structure Diagram  All: Final report |
| 3/26/17 | All | Fix report mistakes. Finalize project phase. | All: Fix structure diagram and data dictionary.  All: Update group report. |

1. **Weekly Time Logs:**

|  |  |  |
| --- | --- | --- |
| **Person** | **Total Time in minutes** | **Tasks** |
| Alex Jacobs | 173 | P-spec, group report |
| Erik Rasmussen | 173 | Structure chart, group report |
| Logan Brincks | 134 | Data Dictionary, group report |
| John Polus | 136 | P-spec, group report |
| Steven Karrmann | 118 | Structure chart, group report |
| **Total Time:** | 734 |  |

1. **Issues:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Issue Number** | **Discovery Date** | **Resolution Date ( Est. – Act. )** | **Responsible Person** | **Description ( Prob / Resolution )** |
| 1 | 3/24/17 | 3/26/17 | Steven | Structure diagram used incorrect shapes. Proper shapes are now used. |
| 2 | 3/24/17 | 3/26/17 | Logan | Data dictionary included improper data and formatting. This is resolved. |
| 3 | 3/24/17 | 3/26/17 | Alex | P-spec preconditions were too general. These are now made more precise. |

1. **Files and their locations:**

|  |  |  |
| --- | --- | --- |
| **Filename** | **Location** | **Contents** |
| 1337\_hAXX\_d8\_structuredAnalysis.docx | \2\_ReengineeringProject\Structure Analysis\ | Structured analysis report/deliverable. |
| DataDictionary.doc | \2\_ReengineeringProject\Structure Analysis\ | Data dictionary in doc format. |
| DataDictionary.xlsx | \2\_ReengineeringProject\Structure Analysis\ | Data dictionary in excel format. |
| Group Report.docx | \2\_ReengineeringProject\Structure Analysis\ | Group report for project phase. |
| Structure Diagram.vsdx | \2\_ReengineeringProject\Structure Analysis\ | Structure Diagram of Nibbles. |

1. **Plans for Coming Week:**

The next phase of this project is the OO (Object-oriented design). To make sure this project can properly be reengineered into a modern programming language, it is important to ensure it follows modern programing paradigms.

As such, next week will focus on breaking down this program into an object-oriented structure and create a class diagram. For this to be done, system requirements, functional, and non-functional requirements will be determined and an appropriate class structure will be defined using these. However, it is important to note that creating a OO design of this project will take quite a bit more time than previous project phases. Therefore, next week will focus on a preliminary design rather than one in complete detail.

1. **Comments:**

**Engineer 1:** Alex Jacobs

This phase is probably one of the more interesting phases of software maintenance/reengineering. While it is not necessarily easy, it is very interesting to see other programmers’ code are parse through their implementation. This is especially true then you encounter unused code, strange function names, and complicated implementations. After theanalysis, I think one of the hardest parts about reengineering this code is going to be figuring out the meaning behind the numerous magic numbers used in this program. That is explains why having defined variables is so important.

**Engineer 2:** Erik Rasmussen

For this phase, I felt I got a much better hands on experience analyzing older code and getting a feel for the structure and practices used at the time. Simply by creating the structure chart, I have a good feel for what the program does and which parts interact with one another. In addition, when the naming is sub-par, I can still deduce what the end goal is by looking at its interactions. The groundwork has been lain for a successful maintenance project.

**Engineer 3:** Logan Brincks

My primary focus was creation of the data dictionary for this phase. I worked well with the other engineers and I finally feel like I understand what a P-Spec is and how to properly perform a structural analysis of code. I enjoyed this phase because it was based upon a language we haven’t used very much and the Understand tool not working allowed us to create our own charts and graphs. This gave us all a better understanding of the structure of the code.

**Engineer 4:** John Polus

For this deliverable, the main goal was to get a general overview of the game’s structure. So, I glanced through the entire program’s source code. After getting a basic picture of what procedures were important and what each one did, I saw that it would not be too difficult to convert it to OOP. Since this game is relatively simple, it was easy to understand how each part interacted with others. Still, it is good practice for converting from one programming strategy (procedural) to another (OOP). If the program were too big, it would be much harder to understand, more overwhelming and take a lot longer. So, this was good practice and preparation for converting bigger programs.

**Engineer 5:** Steven Karrmann

This deliverable gave us a good understanding of how Nibbles works. Taking the time to understand how the program’s piece’s fit together helps us with our preliminary design discussions. It was also very useful to make sure that we all had a handle on Basic syntax to better convert this program to an Object Oriented version. The data dictionary, in particular, will be very useful when designing a programming the updated version, because it will make referencing the code simple, and help with our understanding of what the developers wanted to do.