BATT5

Software CRC Document

2.0

09/24/2019

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 9/24/19 |
| Current Release: | 2.0 |
| Indicator of Last Page in Document: | & |
| Date of Last Review: | 9/24/19 |
| Date of Next Review: |  |
| Target Date for Next Update: |  |

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Gates

Dr. Salamah

Dr. Roach

Elsa Tai Ramirez

Jose Cabrera Maynez

Customers: Dr. Jaime C. Acosta

Dr. Oscar A. Perez

Vincent Fonseca

Herandy Denisse Vazquez

Baltazar Santaella

Florencia Larsen

Juan Ulloa

Jesus Martinez

Software Team Members:

Adal Rivas

Alain Sanchez

Juan Gaucin

Andrea Labrado

Mark Nunez

Ana Hernandez

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 |  |  | Creation of Document |
| 1.1 | 09/21/2019 | Alain, Adal, Juan | Document draft |
| 1.2 | 09/23/2019 | Andrea | Started Section 1, 2.9, and 2.10 |
| 2.0 | 09/24/2019 | Team 5 members | Finished draft |

Table of Contents

[Document Control ii](#_Toc20225266)

[Approval ii](#_Toc20225267)

[Document Change Control ii](#_Toc20225268)

[Distribution List ii](#_Toc20225269)

[Change Summary ii](#_Toc20225270)

[1. Introduction 1](#_Toc20225271)

[1.1. Purpose and Intended Audience 1](#_Toc20225272)

[1.2. Scope of Product 1](#_Toc20225273)

[1.3. References 1](#_Toc20225274)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc20225275)

[1.4.1. Definitions 1](#_Toc20225276)

[1.4.2. Acronyms 2](#_Toc20225277)

[1.4.3. Abbreviations 2](#_Toc20225278)

[1.5. Overview 2](#_Toc20225279)

[2. Detailed Description of Component <name> 3](#_Toc20225280)

[2.1. Class <UI\_BATT5> 3](#_Toc20225281)

[2.1.1. Superclass 3](#_Toc20225282)

[2.1.2. Subclasses 3](#_Toc20225283)

[2.1.3. Responsibilities 3](#_Toc20225284)

[2.2. Class <UI\_Controller> 3](#_Toc20225285)

[2.2.1. Superclass 3](#_Toc20225286)

[2.2.2. Subclasses 3](#_Toc20225287)

[2.2.3. Responsibilities 3](#_Toc20225288)

[2.3. Class <Errors> 4](#_Toc20225289)

[2.3.1. Superclass 4](#_Toc20225290)

[2.3.2. Subclasses 4](#_Toc20225291)

[2.3.3. Responsibilities 4](#_Toc20225292)

[2.4. Class <Radare2> 4](#_Toc20225293)

[2.4.1. Superclass 4](#_Toc20225294)

[2.4.2. Subclasses 4](#_Toc20225295)

[2.4.3. Responsibilities 4](#_Toc20225296)

[2.5. Class <Plugin\_Manager> 4](#_Toc20225297)

[2.5.1. Superclass 4](#_Toc20225298)

[2.5.2. Subclasses 5](#_Toc20225299)

[2.5.3. Responsibilities 5](#_Toc20225300)

[2.6. Class <Data\_Accessor> 5](#_Toc20225301)

[2.6.1. Superclass 5](#_Toc20225302)

[2.6.2. Subclasses 5](#_Toc20225303)

[2.6.3. Responsibilities 5](#_Toc20225304)

[2.7. Class <Output\_Manager> 5](#_Toc20225305)

[2.7.1. Superclass 5](#_Toc20225306)

[2.7.2. Subclasses 5](#_Toc20225307)

[2.7.3. Responsibilities 6](#_Toc20225308)

[2.8. Class <Project\_Manager> 6](#_Toc20225309)

[2.8.1. Superclass 6](#_Toc20225310)

[2.8.2. Subclasses 6](#_Toc20225311)

[2.8.3. Responsibilities 6](#_Toc20225312)

[2.9. Class <POI> 6](#_Toc20225313)

[2.9.1. Superclass 6](#_Toc20225314)

[2.9.2. Subclasses 6](#_Toc20225315)

[2.9.3. Responsibilities 6](#_Toc20225316)

[2.10. Class <Database> 7](#_Toc20225317)

[2.10.1. Superclass 7](#_Toc20225318)

[2.10.2. Subclasses 7](#_Toc20225319)

[2.10.3. Responsibilities 7](#_Toc20225320)

# Introduction

## Purpose and Intended Audience

The purpose of creating this document is to outline all candidate classes and responsibilities for the system. It is intended for the guidance team, clients, and team 5 members to be able to identify and understand the potential classes and responsibilities of the system specified in the provided Software Requirements Specification (SRS) document.

## Scope of Product

The goal of this product is to allow analysts from the Development and Engineering Command reduce the amount of time required to perform binary analysis and perform a more targeted type of binary analysis via a graphical interface.

## References

[1] Elsa Tai Ramirez, Software Requirements Specification, 2019.

## Definitions, Acronyms, and Abbreviations

### Definitions

|  |  |
| --- | --- |
| **Overlay** | A temporary window that displays additional information above the main window. |
| **Project** | A folder containing a binary file that will be analyzed. All projects will be visible in the system’s project directory. |
| **Breakpoint** | During Dynamic Analysis, this is an intentional stopping or pausing place for debugging purposes. |
| **Output Field** | What the system produces depending on the plugin being used during analysis. |
| **Protocol Structures** | A Protocol structure defines how a network packet is encapsulated. The network packet consists of a header and data sections. |
| **Point of Interest** | A Point of Interest (POI) is a specific point location that someone may find useful or interesting depending on what plugin the user is using. When using the network plugin, any string, function call, structure, or instruction used in a file are considered points of interest. |
| **Static Analysis** | Static analysis is when the system converts the executable file into Human-Readable Form |
| **Dynamic Analysis** | When the points of interest identified during static analysis are executed for debugging purposes. |
| **Debug** | When Dynamic Analysis is performed. |
| **Binary File** | Any executable file that the system will be able to analyze. |
| **Plugin** | The plugins used by the system will dictate the points of interest identified during static analysis. Additionally, each plugin will come with a set of custom controls. |

### Acronyms

|  |  |
| --- | --- |
| **BATT5** | Binary Analysis Tool by Team 5 |
| **POI(s)** | Point(s) of Interest |
| **RDECOM** | U.S. Army Research Development and Engineering Command |
| **UML** | Unified Modeling Language |
| **UCD** | Use Case Diagram |
| **SRS** | Software Requirement Specification |
| **DLL** | A DLL file, short for Dynamic Link Library |
| **BEAT** | Behavior Extraction and Analysis Tool |

### Abbreviations

|  |  |
| --- | --- |
| **e.g.** | For example |
| **Assy** | Assembly |
|  |  |
|  |  |
|  |  |

## Overview

The section that follows will cover all the classes. The classes listed in the next section are all the candidate classes that were identified in the CRC process. Each individual class will have a detailed description. If appropriate each class will list their respective supper class and subclasses. Each class will have a list of their responsibilities identified by a unique tag for reference. For each class if appreciate collaborations will be listed as a reference to a tag that corresponds to a responsibility in the collaborating class.

# Detailed Description of Components

This section contains a detailed description of our candidate classes, their responsibilities, and collaborations.

## Class <UI\_BATT5>

The UI\_BATT5 class setups and creates the user interface to be displayed.

### Superclass

UI\_BATT5 will not have a superclass.

### Subclasses

Errors class (error popup windows), and Output\_Manager (additional popup windows).

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **UI\_1** | -Setup user interface  **Collaborations:** ctrl\_4 |
| **UI\_2** | -Send user input to the backend  **Collaborations:** ctrl\_1 |
| **UI\_3** | -Update user interface accordingly  **Collaborations:** ctrl\_3 |

## Class <UI\_Controller>

The UI\_Controller class will be the middleman between the frontend (User Interface) and the backend (i.e. Button clicked) of the BEAT project. It will receive the input from the UI\_BATT5 class, process it in the Radare2, Project\_Manager, Plugin\_Manager, Output\_Manager, or POI class.

### Superclass

UI\_Controller will not have a superclass

### Subclasses

UI\_Controller will not have any subclasses

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **ctrl\_1** | -Receive user input  **Collaborations:** UI\_2 |
| **ctrl\_2** | -Send user input for processing  **Collaborations:** R2\_1, R2\_2, OM\_2 |
| **ctrl\_3** | -Populate user interface with processed  **Collaborations:** UI\_3 |
| **ctrl\_4** | -Initialize the User Interface  **Collaborations:** UI\_1 |
| **ctrl\_5** | -Receive processed information  **Collaborations:** R2\_1, R2\_2, OM\_2 |
| **ctrl\_6** | -Display popup windows.  **Collaborations:** err\_1, OM\_2 |

## Class <Errors>

The Errors class will host all of the error popup messages. It will create them and send them to the UI\_Controller class for display.

### Superclass

Errors class will not have a superclass.

### Subclasses

Errors will not have any subclass.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **err\_1** | -Create the error windows with the appropriate error message.  **Collaborations:**  ctrl\_3, UI\_3 |
|  |  |

## Class <Radare2>

The Radare2 class provides the functionality of Radare2 to the system. This will be used to perform both static and dynamic analysis. It also provides handles command line arguments.

### Superclass

Radare2 will not have a superclass.

### Subclasses

Radare2 has no subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **R2\_1** | -Perform static analysis on a given binary file.  **Collaborations:** ctrl\_3 |
| **R2\_2** | -Perform binary analysis on a given a binary file.  **Collaborations:** ctrl\_3 |
| **R2\_3** | -Handle radare2 commands from the command line.  **Collaborations:** ctrl\_2 |

## Class <Plugin\_Manager>

This Plugin\_Manager class is in charge of adding, removing, and modifying the plugins defined within the BATT5 system.

### Superclass

Plugin\_Manager will not have a superclass.

### Subclasses

Plugin\_Manager does not have any subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **PM\_1** | -Read XML Files containing Plugin definition and store it into a python object  **Collaborations:** DA\_1 |
| **PM\_2** | -Remove python object and XML files pertaining to Plugin when prompted by the user.  **Collaborations:** DA\_1 |
| **PM\_3** | -Allow user to modify XML file and python object to modify plugin when prompted by the user  **Collaborations:** DA\_1 |
| **PM\_4** | -Allow user to create XML plugin configurations from GUI  **Collaborations:** UI\_2 |

## Class <Data\_Accessor>

The Data\_Accessor class is tasked with accessing all data stored in any database used by the BATT5 system.

### Superclass

Data\_Accessor will not have a superlass.

### Subclasses

Data\_Accessor has no subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **DA\_1** | -Retrieve information from the database  **Collaborations:** ctrl\_5, R2\_1, R2\_2 |
|  |  |

## Class <Output\_Manager>

This class will be responsible of interfacing with jinja to generate the output script.

### Superclass

Output\_Manager will not have a superclass

### Subclasses

Output\_Manager has no subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **OM\_1** | -Generates the output script utilizing the tools from jinja.  **Collaborations:** R2\_2 |
| **OM\_2** | -Creates the window for generating output script and choosing destination.  **Collaborations:** ctrl\_6 |

## Class <Project\_Manager>

This Project\_Manager class is in charge of adding, removing, and modifying the projects defined within the BATT5 system.

### Superclass

Project\_Manager will not have a superclass.

### Subclasses

Project\_Manager has no subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **PRM\_1** | -Read XML Files containing Project definition and store it into a python object  **Collaborations:** DA\_1 |
| **PRM\_2** | -Remove python object and XML files pertaining to Project when prompted by the user.  **Collaborations:** DA\_1 |
| **PRM\_3** | -Allow user to modify Project file and python object to modify plugin when prompted by the user  **Collaborations:** DA\_1 |
| **PRM\_4** | -Allow user to create XML project configurations from GUI  **Collaborations:** UI\_2 |

## Class <POI>

The POI class provides important identifiers or data about a plugin. For example, the Network plugin will contain variables, strings, DLL, function, packet protocol, and structs.

### Superclass

POI will not have a superclass.

### Subclasses

POI has no subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **poi\_1** | -Identify important data points given a selected plugin.  **Collaborations:** Plugin |
| **poi\_2** |  |

## Class <Database>

The Database class will be the component in charge of storing and organizing data from a binary analysis, saving the project state, and the xml files defining the projects, plugins, and poi definitions.

### Superclass

Database will not have a superclass.

### Subclasses

Database has no subclasses.

### Responsibilities

|  |  |
| --- | --- |
| **Responsibilities** | **Responsibility Description** |
| **db\_1** | -Store data from analysis  **Collaborations:**  R2\_1, R2\_2 |
| **db\_2** | -Organize data and changes in data.  **Collaborations:** R2\_1, R2\_2 |

&