<<Git Guys>>

<Finding and Reporting Information Console>

Software Configuration Management Plan

Version <1.0>

<September 11, 2020>

Document Control

Approval

The Guidance Team and the customers will approve this document.

Document Change Control

Initial Release	0.1
Current Release	1.1
Indicator of Last Page in Document	</3
Date of Last Review	8/31/2020
Date of Next Review	9/04/2020
Target Date for Next Update	8/30/2020

Distribution List

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

Guidance Team Members: Dr. Salamah

Customer: Cyber Experimentation & Analysis Division (CEAD)

Software Team Members: Alex Vasquez, Luis Soto, Andrew Clanan, Isaias Leos, Jacob Padilla

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
0.1	8/28/2020	Alex Vasquez	Setting up parts of the Document Control TEAM ASSIGNMENTS

			Alex: Introduction and Audit Isaias: Identification Luis: Identification Andrew: Control Jacob: Control
0.2	8/28/2020	Isaias Leos Luis Soto	Worked on Section 2.2 and 2.1
0.3	8/30/2020	Andrew Clanan Jacob Padilla	Worked on Section 3.1, 3.2, and 3.3
0.4	8/31/2020	Alex Vasquez	Finished section 4, pending team approval
0.4.1	8/31/2020	Isaias Leos	Looking over section 2.2 and 2.1 and added comments to other sections
0.5	9/2/2020	Git Guys	Reviewed documents, fixed issues, cleared ideas, got on the same page.
0.6	9/3/2020	Isaias Leos	Improved upon section 2.1 and 2.2 | Filled missing items from those sections and added introduction for both sections.
0.6.1	9/3/2020	Luis Soto	Added description for section 2
0.7	9/4/2020	Alex Vasquez	Document Hygiene
1.0	9/7/2020	Jacob Padilla	Fixed/ commented on parts of sections 1 introduction, and 3.1
1.1	9/10/2020	Luis Soto	Fixed part of the comments in part 2.2

1.2	9/11/2020	Andrew & Jacob	Address section 3 comments
1.3	10/12/2020	Isaias Leos	Updated the directory structure to match what we currently have.

Table of Contents

1. Introduction

The Finding and Reporting Information Console is a tool that the Cyber Experimentation and Analysis Division will use to report findings and vulnerabilities with mitigations reported to their clients. The system will be accessed through a web browser and will follow all guidelines and requirements that are stated in the official Software Requirements Specification.

The purpose of the Software Configuration Management Plan (SCM) is to control, organize, and main traceability of the changes that are made to any document or source code for the Finding and Reporting Information Console (F.R.I.C.). The SCM is broken down into three parts which are Identification, Control and Auditing. The Identification section of the document will provide baselines for the various entities inside of F.R.I.C. along with any updates. The Control section of this documents the process of making a change to F.R.I.C. along with procedures that are to be followed consistently throughout the development of the system. Lastly the Auditing section, provides a detailed way to determine if the current configuration of F.R.I.C. matches F.R.I.C. in regard to the established baselines.

This document is only intended to be viewed by the Cyber Experimentation and Analysis Division team, the development team, and the guidance team as there is information that is confidential to all parties not mentioned above. Contents shall not be disclosed, discussed, or shared with any individuals or parties that are not mentioned above.

1.1. Reference

1. Hans Van Vliet, Software Engineering, Principles and Practice, 3rd edition, John Wiley & Sons, 2008. Chapter 4.

2. H. E. Bersoff, Elements of Software Configuration Management. IEEE Transactions on Software Engineering, 10(1):79-87, Jan 1984.

2. Software Configuration Identification

The following section includes all the items that require control at the time that a baseline is established.

2.1. Software Configuration Item Identification

The SCI (Software Configuration Items) listed below will be experiencing some kind of change through the “lifecycle” of the project [2].

Items	Elements	Justification
Source Code	Project Source Code - This is all the source code for the project.	Source code will be changing through the semester as the different parts of the project changes, from fixing bugs to modifying different parts of the project according to other software configuration item identification.
Design Documents	**Class Diagram** - Provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes **Use Case Diagram** - Graphic representation of the interactions among the elements of the system **Data Flow Diagram** - Representing a flow of data through a system **State Transition Diagram** - Describe the behavior of systems	Design documents will be changing depending on the clients needs on adding new features to the project.
Test Suites	**Unit Test** - Testing the basic parts of code like classes, interfaces, and functions.	Test suite will serve the purpose of testing the system to behave as intended through a variety

	System Testing - Running and testing the project in a different environment. **Blackbox Testing** - Focuses on the output of the program to see if its what its intended.	of test cases. As the F.R.I.C system continues to be developed this test suite will improve or change to fit the needs of the system.
Requirement Documents	**SRS** - Description of the software system in development. The document lays out functional and nonfunctional requirements. Besides the aforementioned, this document describes the user interactions that the software must provide to the user with the use of different diagrams.	Requirement documents will be changing depending on the specifications given by the clients throughout the semester. Any specification can be added, edited, or erased, depending on the client's need.
SCM	SCM Document	This document will serve the purpose of maintaining the project standards.This document will try to preserve the standards as the system will expect change through its lifecycle this document will continue to be improved upon.

2.2. Software Configuration Item Organization

The Software Configuration Item (SCI) section will provide tools to show and organize the system in different areas. Some of the following areas will be from the naming structure of the program-code files, naming schemes for different versions, detailed directory structure and various ways that we will be backing up the project.

Naming Structure:

- Camel case will be used for the naming structure for this project.

Naming Scheme for Versions:

- Major.Minor.Patch
    - Major - Label for a functionality or technical change to the application. A functional change would be any new feature added, and technical change is any change to any implementation already existing in the code.
    - Minor - Label for an aesthetic change to the application. Aesthetic change would be any change in fonts, font sizes, spacing between elements inside the application, color, among other related items related to the appearance of the application.
    - Patch - Label for a bug fix.

Directory Structure:

- The “src” folder in the root directory will be used to store the code of the project.
    - The “assets” folder in the “src” directory is used to content images, video, and other things that aren’t code, configuration files, and will all be static.
    - The “reports” folder inside the “src” directory will be used to contain all the reports that are generated by the project or read by the project to meet requirements.
    - The “logs” folder inside the “src” directory will be used to contain log information. The project will automatically produce a file that contains a record of events done within the project.
    - The project will also have folders based on the work each file will have the username of the individual engineers won’t be mentioned and just have folders with the name of the files.

Storage Options:

All source code/documentation and non-code files will be hosted on github. Each member will be in charge of pulling and pushing to the project database. The project will have 6 branches, one for each member and one master branch that’ll hold the stable version of the project. If a member worked on the project they will be required to push code into their testing branch at least once a day. The naming scheme for the program-code files will be done in camelCase and will be named based off what code will be doing. Each person will have a copy of the project while also having a custom way of backing up the project from either personal onedrive or google drive folder. Github will also be included into the backup of the project files as a secondary backup. The team will have to keep their custom backups up-to-date at all times. All releases of the project base will be hosted on github.Once approved by the team, the current documentation(s) will be uploaded to both the google (GitGuys classroom) drive, and github.

3. Software Configuration Control

Software Configuration Control (SCC) defines the procedures involved with modifying the dependencies of F.R.I.C. The following sections describe how changes are to be proposed, the Configuration Control Board (CCB) involved in committing these changes, and the procedures used to implement these changes.

3.1. Documentation

Requests to make a change to the software system of F.R.I.C. will be recorded and submitted to the Configuration Control Board for approval. The request must contain all of the following information to be reviewed.

- Name of the person requesting the change
- Proposed start date of applying change(s)
- Proposed delivery dates
- Description of the change(s)
- Priority level based on an (low, medium, high) evaluation
    - Low - The change is not part of the main functionality of F.R.I.C.
    - Medium - The change is a fix that can be considered after deployment.
    - High - The change is part of the main functionality of F.R.I.C.
- Business justification (the importance of the request)
- Impact assessment -
    - What are the changes to be made ?
    - How might it affect other areas of the system ?
- Difficulty to make change
    - How many hours are required to implement the change?
    - Are any additional resources needed?
    - Does the change affect the main functionality?
- Does this change affect any other part of the system? If so, which part?

Once the Configuration Control Board has approved the changes they will sign the document and assign the actual start date and delivery date. If the request for change is disapproved by the CCB. A justification as to why the change isn't deemed necessary will be given by the CCB.

3.2. Configuration Control Board

Any adjustment(s) (change(s) requested will be examined by the software team as a whole, however the final vote for approving/disapproving will come from the CCB (Configuration Control Board).The CCB will consist of the V&V, system analyst, and the current deliverable lead. These team members are selected due to their role and understanding of the client's

requirements. The decision to approve or deny a request for change will be analyzed on; the amount of engineering hours required for change(s), and the priority level of the change(s). If the CCB approves the request for change the procedures for making the change(s) will be assigned amongst team members. A justification as to why the change isn't deemed necessary will be given by the CCB if the request isn't approved. The V&V will conduct a series of tests to determine if there are any errors. If any errors are found they will be reported to the development team to determine a course of action.

3.3. Procedures

3.3.1 Introduction

This section outlines the procedures used in the software configuration management of F.R.I.C.. Such procedures include: steps to create a baseline in the software lifecycle, guidelines for managing configuration items, the tools used for documentation, how version numbers are to be used, steps to check in/out items, and the control board approval process [1].

3.3.2 Approval Documentation

In order to secure approval for modifications, adjustments, and add-ons to the software at any point during the life-cycle, the request form must be filled out appropriately.The request form information required upon submission is listed below.

- Name of the person requesting the change
- Proposed start date of applying change(s)
- Proposed delivery dates
- Description of the change(s)
- Priority level based on an (low, medium, high) evaluation
    - Low - The change is not part of the main functionality of F.R.I.C.
    - Medium - The change is a fix that can be considered after deployment.
    - High - The change is part of the main functionality of F.R.I.C.
- Business justification (the importance of the request)
- Impact assessment -
    - What is the change(s) to be made ?
    - Does this change(s) affect any other part of the system? If so, which part?
- Difficulty to make change
    - How many hours are required to implement the change?
    - Are any additional resources needed?
    - Does the change affect the main functionality?

Documentation will be carried out through Git, Google Docs, and the IDE used for implementation. In order to change the documentation processes', approval of the configuration control board is required.

3.3.3 Configuration Control Board

The Configuration Control Board will consist of: V&V, lead analyst, and the lead for the current deliverable. The approval process will be as such, any request that requires action by the configuration control board will come to a vote of the three members; the vote will determine the verdict of the item, and how the item will be distributed. If not approved the CCB will give an explanation as to why the request wasn't approved.If the team member still wants to make the request. The team member can make modifications to the request form (section 3.3.2) in order to resubmit.

3.3.4 Modifications

To change any item of F.R.I.C. approval by the configuration control board is required. Only after approval of change is the modification allowed. To modify an item of the system, you will need to go to the corresponding documentation tool. For example to modify a document, use Google Docs, to modify code use git. Pull the current version of the item and make the approved modifications. After the modification, make a request to apply to the master file/branch. The configuration manager, who will be identified as the V&V, will view the request and approve or deny it. If the modification is approved the version number will change by incrementing the first digit if it is a major change, the second digit if it is a minor change, and the third digit if it is a bug fix.

If the change is rejected by the CCB the team member may re-evaluate the change and resubmit the request. If the team member does not wish to re-evaluate the change then the change will not be implemented.

3.3.5 Baselines

A baseline will be determined by the configuration control board. The configuration control board will discuss and review the current version of F.R.I.C. and determine if it is a baseline. Once a baseline has been established the major version number (X.0.0) will be increased.

3.3.6 Check In/Out

In the event that a client or developer needs an item of F.R.I.C., they will need approval of the configuration control board. If the configuration control board approves the person checking in/out an item will need to go to the respective tool (see modifications) and make a request for the document,tool,etc… The configuration manager will approve or deny the request.

4. Software Configuration Auditing

Software Configuration Auditing is in place to ensure that the current configuration of the software system mirrors the software system pictured in the baselines and the requirements documentation. The auditing will only be done once a certain implementation, update, or change is completely done. The development team will be responsible for ensuring the following:

a. All procedures are strictly followed for creating a baseline, any update, or any change
b. Implementation of a given baseline meets a requirement of a client
c. Proper documentation of a change
d. Any change is approved by the control board

The CCB (Configuration Control Board) will add any new baseline or update to the Audit Form which will be filled with binary questions that ensure the verification and validation of everything stated above. The Audit form is listed below.

Lastly, the development team will follow a Requirements Matrix in which every implementation meets specified requirements in the SRS. Upon auditing this form, the CCB will vote on whether or not the implementation met the a specified requirement on the SRS. If the implementation did indeed meet the requirement the CCB will label the section requirement with a “YES”. If the implementation did not meet a specified requirement then the CCB will label it with a “NO” and bring it to the attention of the development team. The process will then start from the beginning as far as auditing this form. Requirements that do not currently have implementations are to be left empty.

4.1 Finding and Reporting Information Console Audit Form

This form is to be used solely by the Configuration Control Board. In order for the audit or update to be complete the following questions must all be answered with “YES”. If any of the following questions are labeled with “NO” it is the responsibility of the CCB to notify the appropriate entities. This is the sole Audit document so any new change or update will simply be added to the bottom of the previous change or update with the exact same format.

Change, Update or Baseline Description:

_____

_____

Audit Questions:

1. Has the request been formally made?

   YES:___ NO:___

2. Has the request been formally reviewed by the CCB?

   YES:___ NO:___

3. Describe the reasoning behind the request for change/update:

   _____

4. Has the specified change been made?

   YES:___ NO:___

5. Has a formal technical review been conducted to assess technical correctness?

   YES:___ NO:___

6. Have the proper procedures been followed and standards been applied?

   YES:___ NO:___

7. Have the SCM procedures for noting the change, recording it, and reporting it been followed?

   YES:___ NO:___

8. Was the change, update, or baseline approved by the CCB?

YES:___ NO:___

Signatures

Were ALL procedures stated in the SCM followed for the proposed change/update to the F.R.I.C system?

CCB MEMBER 1: YES:___ NO:___

CCB Member 1 Signature:

_____

Were ALL procedures stated in the SCM followed for the proposed change/update to the F.R.I.C system?

CCB MEMBER 2: YES:___ NO:___

CCB Member 2 Signature:

_____

Were ALL procedures stated in the SCM followed for the proposed change/update to the F.R.I.C system?

CCB MEMBER 3: YES:___ NO:___

CCB Member 3 Signature:

_____

</3