Team 12 - Feathersoft

PICK PMR Insights Collective Tool

Software Configuration Management Plan

Version <0.2>

<2/23/2020>

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 0.1 |
| Current Release: | 0.2 |
| Indicator of Last Page in Document: | & |
| Date of Last Review: | 2/23/2020 |
| Date of Next Review: | 3/1/2020 |
| Target Date for Next Update: | 3/5/2020 |

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:

Jake Lasley

Steven Roach

Customer:

White Sands Missile Range

Software Team Members:

Jay James

Adrian Sosa

Cristian Molina-Simental

Sergio Nogami

Jesus Gomez

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 0.1 | 02/05/2020 | All Software Team Members | Created first draft of the document for review purposes |
| 0.2 | 2/24/2020 | All Software Team Members | Created descriptions of all sections |

Table of Contents

[Document Control ii](#_Toc61860422)

[Approval ii](#_Toc61860423)

[Document Change Control ii](#_Toc61860424)

[Distribution List ii](#_Toc61860425)

[Change Summary ii](#_Toc61860426)

[1. Introduction 1](#_Toc61860427)

[1.1. References](#_Toc61860428) 1

[2. Software Configuration Identification](#_Toc61860429) 2

[2.1. Software Configuration Item Identification](#_Toc61860430) 2

[2.2. Software Configuration Item Organization](#_Toc61860431) 2

[3. Software Configuration Control](#_Toc61860432) 4

[3.1. Documentation](#_Toc61860433) 4

[3.2. Procedures](#_Toc61860435) 4

[4. Software Configuration Auditing 6](#_Toc61860436)

# Introduction

The Lethality Survivability & Human Systems Integration Directorate (LSH) works with the Department of Defense (DOD) to perform operational testing on technology systems to provide secure, resilient capabilities in the expected operational environment. The LSH or White team will analyze the flow of events that occurred during adversarial attacks between the Red and Blue teams to validate what happened during the attack. The Red team performs the attacks and the Blue team tries to mitigate them. The scope of PICK PMR Insights Collective tool will cover the need of the LSH to draw relationships between the events that occurred between the Red and Blue teams. The PMR tool will not draw the correlation of events itself but rather ease the process of drawing the correlations for the analyst.

The PMR will satisfy the clients’ need to validate, search, sort, filter, and ingest log files as well as create a graph to draw associations between the flow of events during an adversarial attack (AA). Before a file is ingested by the PMR it will be validated or checked for any structural errors including missing/invalid timestamps or empty rows, and/or columns. If a file contains picture data, it will be parsed using an Optical Character Recognizer (OCR) tool. If the file contains audio or video, the audio from will be transcribed into text and split into one-minute intervals before ingestion. After a file is ingested into the PMR, it becomes a log entry which will be displayed in a tabular view that an analyst can edit. A log entry will represent one event that occurred during the AA. The PMR will allow the analyst to flag a log entry as significant to the event they want to display on a graph by assigning it to a vector. Once a log entry is flagged as significant it becomes a node that will be used to display an event on the graph and any changes made to the log will be committed to a database. The vector will then be displayed on the graph which the analyst can edit and make desired associations. The analyst can then export a view of the graph that other analysts can see.

The purpose of the Software Configuration Management (SCM) plan is to be able to track and control changes in the software while maintaining the integrity and traceability of the configuration throughout the system’s lifecycle. The SCM document explains the guidelines and policies to be followed by the developers of the PICK PMR Insight Collective Tool. The SCM document is separated into three main components: software configuration identification, software configuration control and software configuration auditing.

## References

[1] O. Perez et al, Requirements Definition Document, Lethality, Survivability and HSI Directorate, 2019.

[2] “Components and Containers in AWT”. Internet: [https://www.cs.utexas.edu/~mitra/csSpring2009/cs313/lectures/GUIComponents.html](about:blank), 2009 [Jan. 28, 2019]

# Software Configuration Identification

This section is catered towards identifying software items that will be part of the configuration, while explaining the relevance of each software item. This section also discusses where the software items will be organized by team 12 while emphasizing the labeling schemes and baselines for updates.

## Software Configuration Item Identification

A **configuration item** (CI) is any service component, infrastructure element, or other **item** that needs to be managed in order to ensure the successful delivery of services. Below is a list of software items that are part of the configuration item list. Configuration items that are likely to change are the following:

|  |  |
| --- | --- |
| Software Item | Description |
| Source Code | The source code used to provide the correct functionality for the system. This will include all necessary algorithms as well as interface implementations for the system. |
| Requirements Document | The requirements document, the SRS, specifies what is being requested by the clients. All items detailed in the SRS must be present in the final system. |
| Prototype | A demonstration of the functionality of the software to present to the clients in order to obtain feedback and ensure the clients expectations are being met. The prototype will be constantly updated to show the progress of the systems development. |
| User Guide | A document detailing how to use the system for users that are unfamiliar with the system. The user guide shall be easy to understand and shall explain the systems functionality. |
| Test Sets | Test sets used to test the correctness and functionality of the system. |
| Maltego TRX (version 1.3.6) | Maltego is a third party software that is detailed as a requirement in the SRS. It will be responsible for the graphical representation of events in the PICK system. |
| Splunk SDK (version 1.6.12) | Splunk is a third party software that is detailed as a requirement in the SRS. It will be responsible for the indexing and searching of the directories, log files, and log entries in the system. |

## Software Configuration Item Organization

Our project database will be stored in a GitHub repository using Git VCS. Backups of the most recent version of the system will be stored on each team members local machine through pulling the most recent changes. In the case that the git repository is lost, the team will meet to decide which team member has the most recent changes. The chosen member will push the changes to all other team members.

**Labeling Scheme**

The labeling scheme for baselines will be the baseline number followed by the number of updates to the baseline.

Files will be named with a version number appended to the end in the form “\_v00” increasing with each update to a file. For example:

ProgramFile\_v01.py

**Directory Structure**

The directory structure will be separated into the following folders:

1. **Assignments**

This folder will hold assignment submissions for the team.

1. **Change Requests**

This folder will hold all change requests for the system, detailed in section 3.1.5.

1. **Configurations**
   1. **Icons** – Includes icons used in the GUI
   2. **Root** - Sample files meant to simulate a root directory with red, blue and white team subfolders.
   3. **Rwo** - Links GUI components into a window
2. **Venv** – Virtual environment folder that includes packages used in the system

├───Assignments

├───Change Requests

├───Configurations

│ ├───icons

│ ├───root

│ │ ├───red

│ │ └───white

│ └─── rwo

└───venv

├───Include

└───Lib

└───site-packages

# Software Configuration Control

## Documentation

Any changes made to the system happen via a change request form. The individual requesting the change will submit a form highlighting their reasoning, expected timeframe, description, priority level, and effort needed to make the change. The change request will be formally reviewed by the members of the software team (see “Software Team Members” under “Distribution List” at beginning of document) who will assess the impact of the change on the system. The purpose of the change request form process is to prevent unauthorized updates and collisions between team members working on the system.

Change requests will be in the form of a word document stored in the “Change Requests” directory on GitHub with the following template:

* 1. Title of change being made in the system.
  2. Name of person making change.
  3. Reason for making change.
  4. Priority level of the change (Low, Medium, High).
  5. The date the form was submitted.
  6. The date the member wants the change done.
  7. The new version of the item being changed.
  8. Status of the change (Open if the change hasn’t been received or closed if the change has been approved/disapproved.)

## Procedures

For any reference in which the team member pulls, pushes or modifies the main project repository on GitHub go to section 1.3.

This section will describe the process for controlling changes to the software system. Any change will be reported by a change request as stated in section 1.4. There will be a folder in GitHub named “Change Requests” were all change request will be posted. This will serve as a repository for any changes made to the software system. Under this folder there will be a tree like structure sub folder system that will be separated by Configuration Items stated in section 1.2 where change requests can be placed in their corresponding folders.

Changes to the software system will be located in local storage. Only the Change Requests will be uploaded to the main project branch on GitHub before “Change Meetings”, a weekly mandatory meeting. The meeting can be attended physically or virtually. The meetings will take place at The University of Texas at El Paso every Friday at 10:00 A.M. The date and time for the Change Meeting can be rescheduled if all the team members agree. If for any reason a team member is not able to attend the meeting, the voting process will continue.

Change Meetings will involve group members taking a vote on what changes will be approved under the Change Request folder on GitHub. The approval process will consist in the team members reviewing each change request and taking a vote for approval and denial. Majority of the vote will determine the course of action for that specific change. If the change is approved, then the change will be merged into the main project branch on GitHub. If the change is denied, then the change will be not be merged to the main project branch on GitHub. The team consists of 5 members. The odd vote count should ensure lack of tie outcomes. If for any reason the vote is tied due to team members missing the meeting, the lead of that section of the software system will have the final vote.

This will also be the process for requesting and reporting errors in the software system when test are conducted by any V & V test suit.

The following table describes the lead for the sections of the software system:

|  |  |  |
| --- | --- | --- |
| **Software Item** | **Member with the authorization create change request** | **Lead Member** |
| Requirements Document | Systems Analyst  V & V  Systems Architect Designer  Lead Programmer | Guidance Team  LSH |
| Design documents | Systems Analyst  V & V  Systems Architect Designer  Lead Programmer | SCM Plan (Analyst)  CRC (Architect)  SDD (Designer)  Test Plan (V&V)  Prototype (Programmer)  CLient Demonstration (Designer)  Team Demonstration (V&V) |
| Prototype | Systems Analyst  V & V  Systems Architect Designer  Lead Programmer | Programmer |
| Source Code | Systems Analyst  V & V  Systems Architect Designer  Lead Programmer | Programmer |
| User Guides | Systems Analyst  V & V  Systems Architect Designer  Lead Programmer | V & V |
| Test Sets | Systems Analyst  V & V  Systems Architect Designer  Lead Programmer | V & V |

# Software Configuration Auditing

The software team will take feedback from clients and guidance team to include in the next version of the configuration and implement it.

The software team will review any issues/bugs caused by the latest implementation or updates to any configuration items and agree on whether the proposed changes mirror the intended software system.

The software team will update the current version number of the configuration.

The software team will present changes to the stakeholders and collect feedback.

&