Prevent, Mitigate, and Recover (PMR) Insight
Collective Knowledge System (PICK)
Software Configuration Management Plan
Version <1.6>
02/06/2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

Initial Release:	1.0
Current Release:	1.6
Indicator of Last Page in Document:	&
Date of Last Review:	02/06/2020
Date of Next Review:	02/08/2020
Target Date for Next Update:	02/10/2020

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:
- Dr. Gates
- Dr. Salamah
- Dr. Roach
- Elsa Tai Ramirez
- Peter Hanson

Customer:
- Dr. Oscar Perez
- Vincent Fonseca
- Herandy Denisse Vazquez
- Baltazar Santaella
- Florencia Larsen
- Erick De Nava

Software Team Members:
- Charlie Juarez
- Miriam Juarez
- Angelica Marquez
- Andrew Munoz
- Aaron Rodriguez

Change Summary

The following table details changes made between versions of this document

Version	Date	Modifier	Description
1.0	02/06/2020	Aaron Rodriguez	Section 1, Section 3.1
1.1	02/06/2020	Charlie Juarez	Section 2 Intro, Section 2.2
1.2	02/06/2020	Andy Munoz	Section 3 Intro, Section 3.2
1.3	02/06/2020	Aaron Rodriguez	Formatting, Check
1.4	02/06/2020	Miriam Juarez	Section 3.3, Section 4
1.5	02/06/2020	Angelica Marquez	Section 2.1, Section 4
1.6	02/06/2020	Aaron Rodriguez	Formatting, Final Check

TABLE OF CONTENTS

1.　Introduction

This document is intended to convey a coherent and systematic plan for how changes will be made to PICK, which is the project that will enable our clients to perform log and vector analysis with efficiency. The SCM (Software Configuration Management) plan is detailed in the following sections: Software Configuration (SC) Identification, SC Control, and SC Auditing. SC Identification section includes the plan to identify configuration items (CI) and how they will be organized. The SC Control section describes how proposed changes will turn into realized changes, and how this process will be documented. The SC Auditing section will explain the process for deciding how closely the configuration matches the intended system. The audiences for this document are the guidance team, customers, and software team.

1.1.　References

[1] S. Roach et al, Software Requirements Specification, Lethality, Survivability, and HSI Directorate (LSH), 2019.

2. Software Configuration Identification

This section will cover the configurations of which the software; Prevent, Mitigate, and Recover (PMR) Insight Collective Knowledge System (PICK) will contain. Thus, the section will cover the documentation of the items that have been identified for the configuration of this software. As well as contain in detail the version and configurations that the system will undertake in order to run. Once the configurations have been documented they will also entail the versions of releases/builds for the software that follow a standardized hierarchical structure. All of which are subject to change per version/releases/builds by the mandate of the guidance team members and/or customers listed on the Software Requirements Specification version 7 (SRS v7) document.

2.1. Software Configuration Item Identification

The software needed for the PMR Insight Collective Knowledge consists of CMU Sphinx, which converts speech from audio and video files to text. For the optical character recognition (OCR) we need to install Tesseract, which will read the characters from an image and convert it into text. Splunk parses the data from all non-image, non-audio, and non-video log files, so that these log files can be validated and ingested into the system as log entries. Mango DB functions the storing of cleansed log files, log entries, nodes, graphs, vectors, and all logging of system actions. Finally, Graphviz is the graphing tool and assists the system in creating graphs for the analyst. The following table contains the links to each source code and documentation for the software needed.

Software	Source Code	Documentation	User manual
CMU Sphinx	https://cmusphinx.github.io/wiki/download/	https://cmusphinx.github.io/wiki/	
Tesseract	https://github.com/tesseract-ocr/tesseract/releases	https://tesseract-ocr.github.io/	https://tesseract-ocr.github.io/tessdoc/
Splunk	https://www.splunk.com/en_us/download.html		
Mango DB	https://www.mongodb.com/download-center/enterprise	https://docs.mongodb.com/manual/core/document/	https://docs.mongodb.com/manual/
Graphviz	https://gitlab.com/graphviz/graphviz/	https://graphviz.gitlab.io/documentation/	

2.2. Software Configuration Item Organization

2.2.1. Version Builds/Releases

The versions of builds/releases of the PICK software will follow a standard with the only exception of the first demo to the clients. Each of the versions will be traceable through the pushes made to the master branch of the GitHub repository named 'CS4311-spring-2020/pick-tool-team10-team-work' will be tracked by the guidance team members listed on the SRS document.

2.2.2. Version Control Access

Each of the releases known as versions of the PICK software will contain a branch labeled 'VersionX' in the GitHub repository afore mentioned. The branch will only contain any functionality/service leading up to the release of the version number. Each of the versions will also contain a 'README.md' containing the features of the software version as well as a standalone executable of the software version along with the code associated with the executable. All of which pertain only to the Linux platform.

2.2.3. Version 1.0

The first version of the software will release as version 1 (v1.0) which will contain the graphical user interface and the base functionality of the software which meets the specifications of the SRS v7 document.

2.2.4. Version 2.0

The second version of the software will release as version 2 (v2.0) which will contain improved functionality of the software and base implementation of an appropriate database/s which meet the specifications of the SRS document of which the version will be relative to the appropriate date of release.

2.2.5. Version 3.0

The third version of the software will release as version 3 (v3.0) which will contain improved implementation of the database/s which meet the specifications of the SRS document of which the version will be relative to the appropriate date of release.

2.2.6. Updates

Each version will undoubtably have updates relative to the functionality of each version of the software. Thus, when a major component of the software has been integrated to the software there will be a GitHub push to the master branch (mentioned earlier) with the comment containing the version number with the following naming convention 'PICK Version#.#', e.g. 'PICK Version1.1', 'PICK Version2.5', 'PICK Version3.2'. Note that when an update is done to the software it will not contain a standalone executable like the version releases/builds.

2.2.7. Software Backups

The system will not handle backups of the information as the customers have previously mentioned that the information the software will handle is sensitive. Therefore, backing up the data could compromise their work. Furthermore, there is no mention of backups on the SRS v7 document, which is subject to change.

2.2.8. Project Database

Since there is nothing on the SRS v7 document relating to databases we are not using servers or connecting to the internet. The host device will contain a local database from which the software will be pulling data. In the case of having multiple people working on the same project on the software a peer to peer communication will be established between the host device hosting the database and the other team members.

3. Software Configuration Control

The purpose of this section is to identify what mechanisms will be used to control access to items in the configuration. This includes establishing a detailed mechanism for preparing, evaluating, and approving or disapproving change proposals to configuration items in the system's life cycle. This will facilitate team members working on the system simultaneously by preventing unauthorized updates and collisions between the team members.

3.1. Documentation

TeamWork will use GitHub and Google Drive as storage for the documentation on proposed changes to the PICK system. However, the formal documentation will only exist on GitHub. To be prepared for sudden changes that need to be made on the system, any member that recognizes a change that needs to be made will initially document this on Google Drive. Once a noted change is documented on Google Drive, the Lead Programmer will decide if this change should become a formally documented change on GitHub. From this stage, the Lead Programmer will decide who will complete the change and if the form of implementation is correct. From these steps, the submission of a change request is encompassed in the process of uploading an initial document of change on Google Drive. The acceptance or denial of a change request is shown through the Lead Programmer's decision on whether or not the requested change needs to formally documented and implemented.

3.1.1. Distributed Version Control

GitHub will enable changes to be documented through having a master branch and 6 non-master branches, one for each TeamWork member. The formal documentation for a requested change will be pushed into the branch of whoever requested the change. The Lead Programmer will assign the implementation of the requested change to a member who complete the change and push it into his/her respective branch. Only the Lead Programmer is allowed to merge changes into the master branch.

3.2. Configuration Control Board

The Configuration Control Board for this project is team 10 which is composed of the following members; Aaron Rodriguez, Charlie Juarez, Andrew Munoz, Miriam Juarez, and Angelica Marquez. The configuration control board is in charge of formally evaluating and approving or disapproving proposed changes to the software system. Every team 10 member is allowed to fully modify each of their components in the system and must propose their changes to be approved or disapproved. Any team 10 member can also make changes to any other person's components as long as they inform the component's original implementor and have them involved in the evaluation process for approving the change. The lead programmer (Charlie Juarez) will be in charge of distributing the required changes to be made in the software system. For proposed changes to be approved, the implementer of the changes and one other team member should review the changes and approve it with the lead programmer. To approve changes, the modified factors should be evaluated to what is specified in the Software Requirements Specification (SRS). The V&V (Angelica Marquez) will be in charge of formally reporting encountered errors in the code to the team and keep record of them. The changed artifacts are to be placed in the current working version directory on GitHub with appropriate copies of previous artifacts saved.

3.3. Procedures

Backups will be done periodically in every meeting done. Such backups will be done Tuesdays and Thursdays before 5 pm. The backups will be identified by date and, if it is necessary by time, letting the team to review changes, or go back to previous stages.

Each member will ensure that his/her changes or assigned activities are being updated in the proper way and manner. If any member of the team wants to do modifications to the actual baseline, he/she is required to have the approbation of every team member. Each modification must have the proper documentation and description, so all the team members can follow others work in clear manner.

To keep all elements, versions, and changes in one same place and available for every team member, team 10 will be using GitHub to host the system elements. That way each member will have the opportunity to delete,

modify, and create new aspect of PICK system, according to the assigned and approved to and for the team members.

4. Software Configuration Auditing

TeamWork will track the changes using GitHub and Google Drive for documentation and version control. For documentation, the table of changes on each document will be changed every time there is a change in a document. Before a deliverable, Team-Work members will ensure that the latest version reflects all the changes made by the members. Also, the software will be presented with test cases to guarantee it works and that it meets the objectives described on the SRS.

&