

**Prevent, Mitigate, and Recover (PMR) Insight  
Collective Knowledge System (PICK)  
Software Configuration Management Plan**  
Version 2.0  
02/24/2020

## Document Control

### Approval

The Guidance Team and the customer shall approve this document.

### Document Change Control

Initial Release:	1.0
Current Release:	2.0
Indicator of Last Page in Document:	&
Date of Last Review:	02/24/2020
Date of Next Review:	02/29/2020
Target Date for Next Update:	02/30/2020

### 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  
Floencia 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, Check
1.7	02/16/2020	Charlie Juarez	Section 2 Intro, Section 2.2 (Update)

SCM	TeamWork	Date 02/24/2020	Page ii
-----	----------	--------------------	------------

## Software Configuration Management Plan

1.8	02/16/2020	Aaron Rodriguez	Section 1, Section 3.1 (Update)
1.9	02/16/2020	Andy Munoz	Section 3 Intro, Section 3.2 (Update)
1.10	02/23/2020	Miriam Juarez	Section 3.3, Section 4 (Update)
1.11	02/17/2020	Angelica Marquez	Did not complete – Section 2.1, Section 4
1.12	02/24/2020	Aaron Rodriguez	Section 3.3 (Update)
1.13	02/24/2020	Andy Munoz	Section 3.3 (Update)
2.0	02/24/2020	Aaron Rodriguez	Formatting, Final Check

## TABLE OF CONTENTS

<b>DOCUMENT CONTROL .....</b>	<b>II</b>
<b>APPROVAL .....</b>	<b>II</b>
<b>DOCUMENT CHANGE CONTROL .....</b>	<b>II</b>
<b>DISTRIBUTION LIST .....</b>	<b>II</b>
<b>CHANGE SUMMARY .....</b>	<b>II</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>1.1. REFERENCES .....</b>	<b>1</b>
<b>2. SOFTWARE CONFIGURATION IDENTIFICATION .....</b>	<b>2</b>
<b>2.1. SOFTWARE CONFIGURATION ITEM IDENTIFICATION .....</b>	<b>2</b>
<b>2.2. SOFTWARE CONFIGURATION ITEM ORGANIZATION .....</b>	<b>2</b>
2.2.1. <i>Version Builds/Releases .....</i>	<i>2</i>
2.2.2. <i>Version Control Access .....</i>	<i>2</i>
2.2.3. <i>Version 1.0.....</i>	<i>3</i>
2.2.4. <i>Version 2.0.....</i>	<i>3</i>
2.2.5. <i>Version 3.0.....</i>	<i>3</i>
2.2.6. <i>Version 4.0.....</i>	<i>3</i>
2.2.7. <i>Version 5.0.....</i>	<i>3</i>
2.2.8. <i>Version 6.0.....</i>	<i>3</i>
2.2.9. <i>Version 7.0.....</i>	<i>3</i>
2.2.10. <i>Updates.....</i>	<i>3</i>
2.2.11. <i>GitHub Organization: Hierarchy:.....</i>	<i>3</i>
<b>3. SOFTWARE CONFIGURATION CONTROL .....</b>	<b>5</b>
<b>3.1. DOCUMENTATION .....</b>	<b>5</b>
3.1.1. <i>Informal Proposed Change Form .....</i>	<i>5</i>
3.1.2. <i>Formal Proposed Change Form .....</i>	<i>5</i>
<b>3.2. CONFIGURATION CONTROL BOARD.....</b>	<b>5</b>
<b>3.3. PROCEDURES .....</b>	<b>6</b>
3.3.1. <i>Informal Proposed Change Steps .....</i>	<i>6</i>
3.3.2. <i>Formal Proposed Change Steps .....</i>	<i>6</i>
3.3.3. <i>Updating Master Branch .....</i>	<i>6</i>
<b>4. SOFTWARE CONFIGURATION AUDITING.....</b>	<b>7</b>

# 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.

SCM	TeamWork	Date 02/24/2020	Page 1
-----	----------	--------------------	-----------

## 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	<a href="https://cmusphinx.github.io/wiki/download/">https://cmusphinx.github.io/wiki/download/</a>	<a href="https://cmusphinx.github.io/wiki/">https://cmusphinx.github.io/wiki/</a>	
Tesseract	<a href="https://github.com/tesseract-ocr/tesseract/releases">https://github.com/tesseract-ocr/tesseract/releases</a>	<a href="https://tesseract-ocr.github.io/">https://tesseract-ocr.github.io/</a>	<a href="https://tesseract-ocr.github.io/tessdoc/">https://tesseract-ocr.github.io/tessdoc/</a>
Splunk	<a href="https://www.splunk.com/en_us/download.html">https://www.splunk.com/en_us/download.html</a>		
Mango DB	<a href="https://www.mongodb.com/download-center/enterprise">https://www.mongodb.com/download-center/enterprise</a>	<a href="https://docs.mongodb.com/manual/core/document/">https://docs.mongodb.com/manual/core/document/</a>	<a href="https://docs.mongodb.com/manual/">https://docs.mongodb.com/manual/</a>
Graphviz	<a href="https://gitlab.com/graphviz/graphviz/">https://gitlab.com/graphviz/graphviz/</a>	<a href="https://graphviz.gitlab.io/documentation/">https://graphviz.gitlab.io/documentation/</a>	

### 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', X being the number representing the appropriate release, 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. All of which pertain only to the Linux platform.

SCM	TeamWork	Date 02/24/2020	Page 2
-----	----------	--------------------	-----------

### 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. I.e. P1 listed on the 'Project Overview' document on piazza.

### 2.2.4. Version 2.0

The second version of the software will release as version 2 (v2.0) which will contain improved GUI functionality of the software which meet the specifications of the SRS document of which the version will be relative to the appropriate date of release. I.e. P2 listed on the 'Project Overview' document on piazza.

### 2.2.5. Version 3.0

The third version of the software will release as version 3 (v3.0) which will integrate Splunk which meets the specifications of the SRS document of which the version will be relative to the appropriate date of release. I.e. P3 listed on the 'Project Overview' document on piazza.

### 2.2.6. Version 4.0

The fourth version of the software will release as version 4 (v4.0) which will rework the UI and complete the log ingestion process which meets the specifications of the SRS document of which the version will be relative to the appropriate date of release. I.e. P4 listed on the 'Project Overview' document on piazza.

### 2.2.7. Version 5.0

The fifth version of the software will release as version 5 (v5.0) which will implement log entry management which meets the specifications of the SRS document of which the version will be relative to the appropriate date of release. I.e. P5 listed on the 'Project Overview' document on piazza.

### 2.2.8. Version 6.0

The third version of the software will release as version 6 (v6.0) which will respond to a client which meets the specifications of the SRS document of which the version will be relative to the appropriate date of release. This version will also have refactored code that enhances the maintainability and testability of the code. I.e. P6 listed on the 'Project Overview' document on piazza.

### 2.2.9. Version 7.0

The third version of the software will release as version 7 (v7.0) which will be the final project delivery to the client which meets the specifications of the SRS document of which the version will be relative to the appropriate date of release. I.e. P7 listed on the 'Project Overview' document on piazza.

### 2.2.10. Updates

Each version will undoubtedly have updates relative to the functionality of each version of the software. Thus, when a major component of the software has been updated/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'.

### 2.2.11. GitHub Organization: Hierarchy:

The following tree list contains the hierarchy of how the data for the PICK software will be organized on GitHub.

Repository:

- src (folder)
  - o build/settings (folder)
  - o main (folder)
    - icons (folder)

SCM	TeamWork	Date	Page
		02/24/2020	3

## Software Configuration Management Plan

- base (folder)
    - linux (folder)
  - python (folder)
    - \_\_pycache\_\_ (folder)
    - Windows (folder)
    - Dialogs (folder)
  - Project\_Info (folder)
    - Project\_Logs (folder)
- README.md (file)
- Documents
  - All documents

SCM	TeamWork	Date 02/24/2020	Page 4
-----	----------	--------------------	-----------



### 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

##### 3.1.1. Informal Proposed Change Form

- Author: The individual team member who made the change discovery.
- Discovery Date: The date at which the individual team member discovered the change.
- Change Category: The type of change this team member is proposing (bug, client request, user request).
- Change Description: The description of the change that needs to be made (2-5 sentence description).
- Main Code File: The main code file that needs to be altered for this change to be implemented.
- Importance: The level of importance the team member has attributed to this change (1-10 scale range, 1 being of the lowest importance and 10 being of the highest importance).
- Workload: The level of workload the team member has attributed to this change (1-10 scale range, 1 being the of the lowest workload and 10 being of the highest workload).

##### 3.1.2. Formal Proposed Change Form

- Discovery Author: The individual team member who made the change discovery.
- Discovery Date: The date at which the individual team member discovered the change.
- Fix Author(s): The team member(s) who will implement the change.
- Proposed Fix Date: The proposed date at which the change will be implemented.
- Approval Date: The date at which the change was approved by the Lead Programmer.
- Change Category: The type of change this team member is proposing (bug, client request, user request).
- Change Description: The description of the change that needs to be made (2-5 sentence description).
- Main Code File: The main code file that needs to be altered for this change to be implemented.
- Secondary Code File(s): Any secondary code file(s) that need to be altered for this change to be implemented.
- External System Effects: Any external systems that interact with PICK that can be affected by the implementation of this change.
- Importance: The level of importance the team member has attributed to this change (1-10 scale range, 1 being of the lowest importance and 10 being of the highest importance).
- Workload: The level of workload the team member has attributed to this change (1-10 scale range, 1 being the of the lowest workload and 10 being of the highest workload).

#### 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

SCM	TeamWork	Date	Page
		02/24/2020	5

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

#### 3.3.1. Informal Proposed Change Steps

1. Team member discovers possible change.
2. Team member creates informal proposed change form.
3. Team member uploads this informal proposed change form to TeamWork Google Drive storage.
4. Lead Programmer decides if this informal proposed change should turn into a formal proposed change.
5. Acceptance/Denial of informal proposed change.

#### 3.3.2. Formal Proposed Change Steps

1. Pre-Condition: Lead Programmer accepts the informal proposed change, so it turns into a formal proposed change.
2. Team member who discovered the change creates a formal proposed change form.
3. Team member uploads this form to their GitHub branch.
4. Lead programmer assigns team member(s) who will implement the change.
5. Implementation of change is made.

#### 3.3.3. Updating Master Branch

1. Team member implements a change/update to their GitHub branch.
2. Implementing team member provides a short commit message describing their change/update.
3. Implementing team member informs lead programmer their change/update is ready to be merged to GitHub master branch.
4. Lead programmer verifies the change/update can be merged to GitHub master branch, if it is not able to be merged, the lead programmer informs the implementing team member of the reason and the process is taken back to step 1.
5. The Lead programmer merges the change/update to the GitHub master branch.

Each member will ensure that his/her changes or assigned activities are being updated and saved in the proper way and manner. Every modification must have the right documentation and description, so all the team members can follow others work in a clear way.

SCM	TeamWork	Date 02/24/2020	Page 6
-----	----------	--------------------	-----------

## 4. Software Configuration Auditing

Team 10 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. The team will check for the requirements affected for the modifications, if any of those is not fully meet, the team will look for changes that fulfill such requirements.

&

SCM	TeamWork	Date 02/24/2020	Page 7
-----	----------	--------------------	-----------