IMPaCT Intelligence

Workplace Violence (WPV) Prediction
Validation Plan and Traceability Matrix

Prepared by: Joseph DiPietro
Date: 04/14/2024
Sponsors: James Corbett (IMPaCT Intelligence), Arend van der Veen (WBAT Safety)

Table of Contents

1. Overview

1.1 Introduction

The validation plan for the Workplace Violence Prediction model will provide a rationale for testing on the individual requirements of the project to ensure they are being met. Each requirement of the project will be assessed with their risks and will be given reason for level of testing. Some requirements may be labeled as riskier than others, so they will be given reason to be focused on more, while less severe risks may not be as much of a focus.

The validation plan will provide good faith that the requirements of the project are being met and that any risks are being carefully checked to ensure a high-quality product.

The scope of validation will include both functional and non-functional requirements of the project.

1.2 Types of Testing

Three types of tests will be performed based on each risk, unit tests, integration tests, and system tests. Unit tests are more rigorous tests that focus on a single function within the project. These tests will be used for high-risk testing. Integration tests focus on the performance of multiple modules of the project running as a group. This will be used for moderate to low-risk testing. System testing focuses on end-user functions and will be used for moderate to low-risk testing.

2. Validation Plan

2.1 Roles and Responsibilities

Testing of the validation of the project will require different roles to test depending on the nature of the test. The development team are responsible for unit and Integration testing, since the development team has the ability to perform tests on specific units or modules within the project. System tests, or any end-user related tests will go to the customers/sponsors involved to ensure the part of the project they will be primarily interacting with works as expected. Any exceptions to these rules will be otherwise stated in the risk assessments.

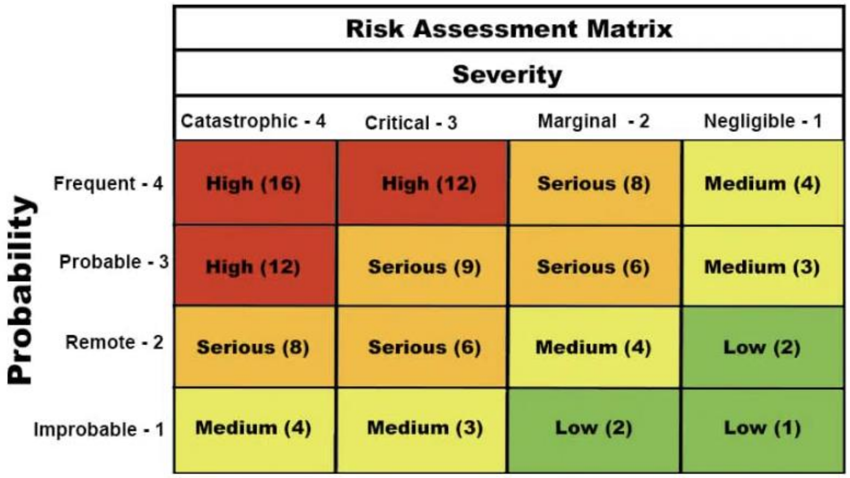
2.2 Validation Strategy

As mentioned in the Overview, the validation plan will focus on the requirements of the project and the risks associated with them. However, some requirements and therefore some risks will be too mild to assess at all. Validation will not include every single module of the project.

All Risks will be assessed with a risk matrix, a number of the “risk level” will be give not a risk based off of the following table. Risks will be evaluated based off of the severity of the nature of the risk and the likelihood of the risk itself. Next to each risk, there will be a number of the overall risk value along with probability and severity score (ex: Risk Level: 6 Prob: 3 Sev: 2)

3. Risk Assessment

All risk assessments are done using the following matrix:



		Risk Assessment Matrix			
		Severity			
		Catastrophic - 4	Critical - 3	Marginal - 2	Negligible - 1
Probability	Frequent - 4	High (16)	High (12)	Serious (8)	Medium (4)
	Probable - 3	High (12)	Serious (9)	Serious (6)	Medium (3)
	Remote - 2	Serious (8)	Serious (6)	Medium (4)	Low (2)
	Improbable - 1	Medium (4)	Medium (3)	Low (2)	Low (1)

3.1 Functional Requirements

Validation will include all functional requirements listed. unlisted functional requirements are considered to be too low-risk to require testing

2. Data Generation
   - Improper or extreme generated values – Risk Level: 1 (Prob: 1 Sev: 1)

     Due to the fact that any generated rows of data that get pushed into the database have a set algorithm to ensure that all rows are generated like expected, the likelihood of improper data being pushed into the database is extremely low. However, if in away way there is improper data being pushed into the database, the severity of the situation is also very low. Since the data we are generating is just for developer testing of the prediction model, there will be no effect on how the model accurately predicts instances of WPV violence. Quick unit testing of the data generation function and the push data into database function should be done to confirm functionality.

3. Testing and Automation
    - Imbalanced/inaccurate prediction output of the model – Risk Level: 8 (Prob: 2 Sev: 4)

      Predictions made by this model in a live hospital environment could potentially prevent workplace violence against hospital staff. This is why it is important to do extensive testing on the prediction model to ensure that it is predicting as expected. Non end-user functions typically are the developers' responsibility to test, but since every hospital environment is different, it should be customer/sponsor responsibility to test the model to see if it is working as expected with the hospital specific data it is working with. Unit testing on training and prediction is required.
4. Visualization
    - WPV prediction dial incorrectly displaying prediction visualization – Risk level: 3 (Prob: 1 Sev: 3)

      The dial shown on the webpage home screen is a visualization of the live predictions the model is making. If the dial somehow incorrectly shows the live predictions or the page itself goes down, staff could potentially miss a predicted threat.
    - Unable to access logging page or email management page – Risk level: 1 (Prob: 1 Sev: 1)

      If the pages of the website that allow users to log incidents or manage email notifications go down, staff will not be able to access those functions. However, these functions are not system-breaking if they do go down as they do not influence predictions or alerts in any way.

System testing on the entire webpage should be done to ensure the website is up, running, and functioning as expected.

5. Email Notification
    - Emails not being sent when WPV is being predicted – Risk level: 4 (Prob: 1 Sev: 4)

      If emails are not being sent out to staff when there is a high probability of WPV, there could potentially be a risk of someone not being able to prevent an incident when it occurs. Unit testing should be done on the notification system to ensure that the function works like normal.

3.2 Non-Functional Requirements

Validation will include all non-functional requirements listed. Unlisted non-functional requirements are considered to be too low-risk to require testing

3. Performance Requirements
    - Misleading training data – Risk level: 9 (Prob: 3 Sev: 3)

        If the training data given to the model when it is initialized is misleading or inaccurate to a live hospital environment, there is a high chance that the model will predict WPV inaccurately, and potentially give false negatives in an environment where false negatives could have large consequences. Extensive unit testing should be done on the model to test predictions after it is trained with the training set given.

    - Model is making predictions at an inefficient pace – Risk level: 4 (Prob: 1 Sev: 4)

        If the output of the model is somehow alerting staff at an inefficient pace, whether it be the webpage or the model itself underperforming, there is a risk of staff getting alerts too late when a WPV incident is happening. Integration testing should be done to ensure that all the parts of the project are running as efficiently as possible to get predictions rolling out fast.
4. Security
    - Hospital databases or project functions being accessed without proper authentication – Risk level: 3 (Prob: 1 Sev: 3)

        Authentication of all functions and access of all databases is crucial to ensure that any systems are not being accessed by unauthorized personnel. Developers should system test all functionality to ensure that everything is authentication required.

4. Traceability Matrix

User Requirement (Requirements Document Reference)	PBI Card Number(s)	Design Document Reference	Test Script Reference	Notes
11(a): The development team needs a mechanism to store (1) training data for the machine learning model, (2) data which is either polled directly from the hospital environment or simulated to be polled from such an environment, and (3) past model predictions along with the associated polled data. 11(b): The model connects to a MariaDB database by connecting to a remote server. 11(c): Team members are given their own credentials and granted access to query the database. Members can connect to the MySQL database by entering the host IP, port, and their personal credentials.	4, 4.2, 5	Figure 1.1, 1.4, 2.1.3, 2.2.3, 2.3.2		
2(b): Generated data will only approximate data which will be found in the workplace to the best of the team's knowledge. This data will not be pushed into the database during the testing phase–instead being fed directly into the model for easy testing. In later phases of the project, this data will be sanitized and stored.	2-2.2.1, 7, 7.1	Figure 1.2, 1.3, 1.4, 2.1.1, 2.1.2, 2.1.3, 2.2.2	5.2.1	
1(b): The greatest candidate for this project is the Random Forest model. This model consists of a group of decision trees that work together to process data and make predictions.	2.3, 2.4, 2.6	Figure 1.1 1.3, 1.4, 2.1.1, 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.3.1		

3(a): After the initial stages of model design have been completed, the developers have the capability to test the model's responses to various forms of data. Ideally, the model should be tested under a variety of input data from multiple departments.	6.5, 2.5.3.1, 2.5.3.2	Figure 1.3, 2.2.2		
3.c After the model has been tested thoroughly, the team needs a feature that can autofill generated data into the system to simulate polling data from a live hospital environment. The team also needs mechanisms to format and sanitize the data to ensure that it is in a format which the model can understand. This data will then be fed into the model and processed.	2.5.2.2, 2.5.2.3, 2.9	Figure 1.2, 1.4, 2.1.3, 2.2.2		
4(a) The team creates a user interface to display a visual representation of the model output. 4(b) Hospital workers need to be able to access all required functions from the UI. It will be split into three windows: the *department view*, the *email entry screen*, and the *logging screen*.	10.5, 12.1, 13.1, 9.2, 9.3, 11.1, 10.3, 10.1, 10.	Figure 2.1.3	5.2.3	
5(a) The system automatically sends email notifications to registered administrators and security personnel when high stress is detected in a given department. 5(b) As mentioned, hospital staff have the capability to opt in and out of receiving such emails.	9.1, 9.1.2, 9.	Figure 1.4, 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.2	5.2.4, 5.3	

12(a) HTTP routes provided by Django REST are created to link the data generation and model predictions to the backend. 12(b) JSON objects are the primary mechanism by which data is moved in and out of the database.	2.2.1, 4.1, 6.1, 8.1, 2.5.1A, 2.5.2, 6., 6.2	Figure 1.1, 1.3, 1.4, 2.3.2		
13(a) The GUI will be designed as simply as possible while still communicating all the data necessary for hospital staff to take action whenever risks of workplace violence are identified. It should be visually appealing and easy to navigate without prior experience.	10.5, 12.1, 13.1, 9.2, 9.3, 11.1, 10.3, 10.1, 10.	Figure 2.1.3	5.3	
3(b) Though some uncertainties are expected, hospital workers need a system that is reliable in its observations. Team Orangutan has decided on 90 percent accuracy as a baseline for model predictions. This requirement will partially be achieved by extensive research on the largest predictors for workplace violence, and will be further reinforced by rigorous testing of the model using SciKit's *accuracy_score* function.	6.5, 2.5.3.1, 2.5.3.2		5.2.2	
14 Hospital workers need prediction software that can manage a large stream of data and display readings quickly. The model must be able to continuously interpret data and update graphics within 10 seconds of that data being received in order to keep up with potentially volatile working conditions.	8.2, 14.1		5.2.5	

15.a. As of now, team Orangutan has decided on HTTP token authorization to protect the model from third-party requests. 15.b. Staff of the hospital will be able to access the primary interface by entering their company credentials.	6.4.2, 6.4.1	Figure 1.2, 1.3, 2.1.1, 2.3.1	5.2.6	

4.1 See Also…

For more information on the rationale behind our validation plan, see the Project Initiation Document and Requirements Document for descriptions of the requirements and associated risks. See also the Design Document for further validation on how we have implemented the requirements into the project. See the Test Script as well for testing on risk analysis.