

# Software Quality Assurance Plan

For an Intrusion Detection System using a Neural Network

Version 1.0

*Submitted in partial fulfillment of the requirements of the degree of MSE*

Blake Knedler

CIS 895 – MSE Project

Kansas State University

## Table of Contents

1	Purpose.....	3
2	References.....	3
3	Management.....	3
3.1	Management Organization .....	3
3.1.1	Supervisory Committee .....	3
3.1.2	Major Professor.....	3
3.1.3	Software Engineer.....	3
3.1.4	Technical Inspectors .....	3
3.2	Tasks.....	3
3.3	Roles and Responsibility .....	3
3.3.1	Supervisory Committee .....	3
3.3.2	Major Professor.....	4
3.3.3	Software Engineer.....	4
3.3.4	Technical Inspectors .....	4
4	Documentation.....	4
4.1	Purpose .....	4
4.2	Minimum Documentation Requirement .....	4
4.2.1	Phase 1 – Objective Phase .....	4
4.2.2	Phase 2 – Architecture Phase.....	4
4.2.3	Phase 3 – Implementation Phase.....	4
5	Standards, Practices, Conventions, and Metrics .....	5
6	Software Reviews .....	5
7	Test.....	5
8	Problem Reporting and Corrective Action .....	5
9	Tools, Techniques, and Methodologies .....	5
10	Media Control .....	6
11	Record Collection, Maintenance, and Retention .....	6
12	Risk Management .....	6

# 1 Purpose

This Software Quality Assurance Plan document is intended for the project “Intrusion Detection System using a Neural Network” as a Master of Software Engineering final project for Blake Knedler.

# 2 References

1. Knedler, Blake. “Project Plan 1.0: Intrusion Detection System using a Neural Network.”
2. IEEE Std. 730-1998, Standard Software Quality Assurance Plans, IEEE, 1998.

# 3 Management

## 3.1 Management Organization

The following are the management organization for this project.

### 3.1.1 Supervisory Committee

- Dr. Mitchell Neilsen
- Dr. Torben Amtoft
- Dr. Daniel Andresen

### 3.1.2 Major Professor

- Dr. Mitchell Neilsen

### 3.1.3 Software Engineer

- Blake Knedler

### 3.1.4 Technical Inspectors

- Tracy Marshall
- Keith Moyer

## 3.2 Tasks

All project tasks are defined and scheduled in the Project Plan 1.0.

## 3.3 Roles and Responsibility

### 3.3.1 Supervisory Committee

The supervisory committee is responsible for watching the three presentations at the conclusion of each phase and providing any guidance and feedback as a result of the presentations.

### **3.3.2 Major Professor**

The major professor is responsible for providing guidance and feedback throughout the project development cycle as deemed necessary or upon request by the main software engineer.

### **3.3.3 Software Engineer**

The software engineer is responsible for the completion of each of the documentation artifacts and the executable deliverables. Additionally, it will be the software engineer to present and the conclusion of each phase.

### **3.3.4 Technical Inspectors**

The technical inspectors are responsible for performing a formal review of the systems architectural design as provided by the software engineer. The inspectors will review the design and provide feedback to the software engineer in charge of the project.

## **4 Documentation**

All documentation will be given to the major professor of the project.

### **4.1 Purpose**

The project documentation serves as the main artifacts of information about the project.

### **4.2 Minimum Documentation Requirement**

#### **4.2.1 Phase 1 – Objective Phase**

- Time Log
- Vision Document 1.0
- Project Plan
- SQA Plan

#### **4.2.2 Phase 2 – Architecture Phase**

- Time Log
- Vision Document 2.0
- Project Plan 2.0
- Architecture Design 1.0
- Formal Requirements Specification
- Technical Inspection Checklist
- Test Plan

#### **4.2.3 Phase 3 – Implementation Phase**

- Time Log
- Component Design 1.0

- Technical Inspection Letters
- User Manual
- Project Evaluation
- Source Code

## **5 Standards, Practices, Conventions, and Metrics**

The Intrusion Detection System using a Neural Network will follow standards outlined in the IEEE Software Quality Assurance standards.

## **6 Software Reviews**

At the conclusion of each of the three phases, a presentation will be given demonstrating the progress of that phase and any executable prototypes developed during that phase. Additionally, formal technical reviews will be conducted by the technical inspectors.

## **7 Test**

A detailed test plan will be given during the second phase of the project. This test plan will include a detailed plan to properly test and assess each of the critical requirements in unit and integration testing forms.

## **8 Problem Reporting and Corrective Action**

As issues arise in the course of this project, the responsibility of reporting these issues and requesting guidance will fall on the software engineer in charge of the project. As the project continues, any problems discovered by the supervisory committee or the technical inspectors will be reported and possible corrective action will take place.

## **9 Tools, Techniques, and Methodologies**

The project will be developed in Python version 3.5 using Visual Studio Community IDE. All project documentation that is provided will be using the Microsoft Office 365 Suite of tools. All UML diagrams will be created using Visual Paradigm.

## **10 Media Control**

All project documentation and source code will be stored on a revision NFS local to the main software engineer developing this system. Additionally, these documents will be stored locally on the software engineer's computer and will be emailed to the main professor of the project.

## **11 Record Collection, Maintenance, and Retention**

All project documentation and source code will be stored on a revision NFS local to the main software engineer developing this system and locally on their computer.

## **12 Risk Management**

All risk reporting and management will be the responsibility of the main software engineer of the project. Any risks noted will be reported to the main professor of the project. The major professor will then be responsible for guidance through any risk to help continue the project's success.