SOFTWARE DEVELOPMENT MANAGEMENT PLAN

for

Data Storage System for Monitoring System

Release 1.0

Version 1.0 approved

Prepared by Baxter Halder, Claire Hall,
Kayla Jamar, Jennifer Olszyna

Contents

List of Figures **4**

List of Tables **5**

1 Introduction **7**
1.1 Purpose 7
1.2 Terminology 8
1.3 Acronyms 9
1.4 References & Standards 9

2 Project Overview **10**
2.1 Software Overview 10
2.2 Schedule 11
2.3 Budget 12
2.4 Project Deliverables 12

3 Management Approach **13**
3.1 Organization and Responsibilities 13
3.1.1 Software Team Lead 13
3.1.2 Testing Lead 14
3.1.3 Quality Lead 14
3.1.4 Software Engineer 14
3.1.5 Test Engineer 15
3.1.6 Software Quality Assurance Engineer 15
3.2 Software Risk Management 16
3.2.1 Process Risk Management 16
3.2.2 Programmatic Risk Management 16
3.2.3 Technical Risk Management 17

3.3 Customer Communications 17
3.4 Team Training 18

4 Technical Approach **19**
4.1 Development Process 19
4.2 Development Tools 19
4.3 COTS & FOSS Tools 20
4.4 Software Reuse 20
4.5 Testing Process 20

5 Configuration Management **22**
5.1 CM Responsibilities 22
5.2 CM Resources 22
5.3 Change Control Procedures 22

6 Verification & Validation **23**
6.1 V&V Procedures 23
6.2 Independent V&V 23

List of Figures

3.1 Organization Chart Listing Project Roles 13

List of Tables

1.1 Definitions of Terms 8
1.2 Acronyms and Abbreviations 9
1.3 References 9

2.2 Project Deliverables 12

3.1 Software Process Risk Management 16
3.2 Software Programmatic Risk Management 16
3.3 Software Technical Risk Management 17
3.4 Training 18

Revision History

Date	Description	Revised by
02/16/22	Initial draft	Data Store
02/21/22	Revision of initial draft	Data Store
02/21/22	Revision of draft	Data Store
03/2/22	Final edits on draft	Data Store

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the currently named "CS455 Monitoring System" software. This document will illustrate the purpose and complete declaration for the development of the data storage sub-system as well as explain the sub-system's: constraints, interface, and interactions with other external applications. This document is primarily intended to be proposed to a customer for their approval and act as a reference for developing the first version of the system for the development team.

1.2 Terminology

Term	Definition
Admin/Administrator	System administrators are those with special privileges to managing and accessing the system.
Users	IT professionals who looks at the dashboard.
Agents	Software that continuously and autonomously collects specified information from client machines connected to the network for the monitoring system.
Engine/Monitoring Engine	Software on the network that will receives the data collected by the agents and continuously store it in the data storage.
Data Store/Data Storage	Database where the data collected by the agents is stored for the monitoring system to access.
Dashboard	a user interface that displays the collected data in the data storage to the admin.

Table 1.1: Definitions of Terms

1.3 Acronyms

Acronym or Abbreviation	Definition
KPI	Key Performance Indicators are quantifiable measure of performance over time for a specific objective that provides the admin insights into the network.
QA	Quality assurance is a system for evaluating performance.
GUI	Graphical User Interface
SRS	Software Requirements Specifications
UNA	University of North Alabama
CS455	The computer science class at the University of North Alabama in which this project is being made

Table 1.2: Acronyms and Abbreviations

1.4 References & Standards

Standard	Reference
TCP/IP	RFC 1180
SQL and PL/SQL Coding Standards	PL/SQL best practice Standards tips
Standard SQL Naming Conventions	Oracle naming standards tips
SEI CERT for C++	SEI CERT Secure Coding Practices for C++

Table 1.3: References

2 Project Overview

The CS455 Monitor System is a network monitor system that will monitor UNA's CS network in order to help administrators identify and address potential network problems as they occur. The system will automatically collect and store data, as well as provide warnings and error messages when certain events trigger the system. However, the system will rely on administrators to see the messages and act on them; the system is **not** automated to handle the events. The monitor systems contains four sub-systems working in tandem; agents, monitoring engine, data store, and a dashboard. This team's focus is the data storage unit of the system which will store the data collected by the agents.

2.1 Software Overview

The user will never directly interact with the database. The database acts solely as a place to store collected metrics and logs. The database will be scalable to accommodate up to hundreds of active servers and a dozen active dashboards. The database system will be hosted on UNA's CS server and host data for up to a year. The data storage will take data from the monitoring system and sort it into one of two two fundamental types of data, metrics or logs. Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios. Logs contain different kinds of data organized into records with different sets of properties for each type.

2.2 Schedule

Date	Item Due	Description
02/11/22	Client Consultation	Initial meeting with client to ask questions regarding client's initial request
02/16/22	Initial draft	Initial rough draft of requirements document and software development management plan
02/23/22	Revision	Revised draft of requirements document and software development management plan
3/02/22	Final draft	Final draft of requirements document and software development management plan
03/09/22	Initial design review	Review initial design for software
03/16/22	Implementation begins	Final design of software is completed and implementation begins
03/23/22	Initial testing	Reviewing test plan and prototype
04/06/22	Testing Phase 1	Reviewing unit test results and prototype
04/13/22	Testing Phase 2	Reviewing unit test results and prototype after input from prior week
04/20/22	Testing Phase 3	Reviewing test results and prototype, then begin integration testing
04/27/22	Integration and Acceptance Testing	Reviewing integration testing results and begin acceptance testing
05/03/22	Software delivery	Deliver finished software and project deliverables
05/03/22	Software Presentations	Team presentations (3:30 - 5:30)

2.3 Budget

N/A

2.4 Project Deliverables

The Data store team is responsible for producing a working SQL Database that is compliant with both the specifications set in this document and the SRS document. The Data store team will all deliver all of the software and documentation associated with the data storage unit no later then May 3, 2022. A single copy of each of the deliverables listed below will be produced by the team.

Name of Deliverable
PDf of SRS document
PDf of SEI CERT Coding Standards documentation
Database system prototype to be used as data storage
Database system final version to be used as data storage
PDF of SDMP document
Software Design Document including a traceability matrix
Test plan document including a traceability matrix
Test results including a traceability matrix
User's manual for operating the software
Software source code files
Copy of software version control repository
Software executable
Team meeting minutes
Team communications summary document
Copy of CM system data
PDF of presentation slides

Table 2.2: Project Deliverables

3 Management Approach

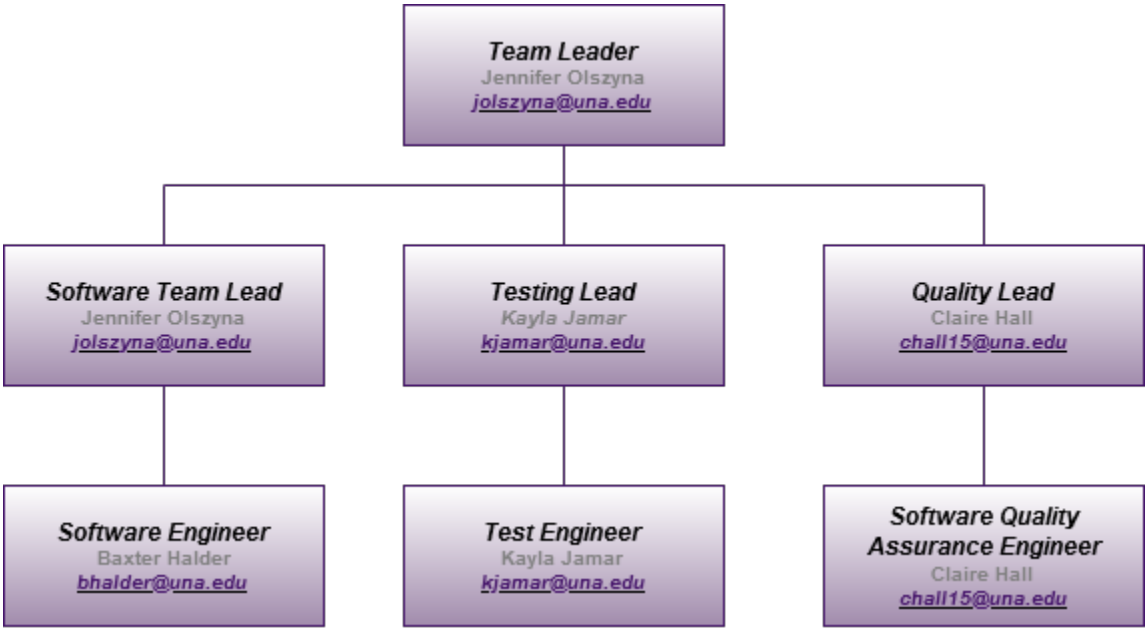

Figure 3.1: Organization Chart Listing Project Roles

3.1 Organization and Responsibilities

3.1.1 Software Team Lead

A Software Engineering Team Leader is the member of the team who is responsible for their team's execution, the quality they produce, the speed

at which they produce. The Software Engineering Team Leader is also the member of the team who represents the team at the weekly stand-up progress report meetings.

3.1.2 Testing Lead

The test leader plays an important role at an outset of the project.Test leaders collaborate with other stakeholders, devise the test objectives, organizational test policies, test strategies and test plans. They decide when test automation is appropriate and they put effort and plan to select the tools and ensure training the testing team. The team leader is also responsible of making sure the test environment is set up and verified before test execution.

3.1.3 Quality Lead

A Quality Lead is responsible for the management and organization of quality testing. They ensure that a project will have a successful promotion by providing quality software and testing the product to ensure that it follows according to the guidelines. They are also involved in finding a solution to any conflicts between teams, and they give motivational advice for the team.

3.1.4 Software Engineer

Software engineers design and create computer systems and applications to solve real-world problems. Software Engineer know how to use the right programming languages, platforms, and architectures to develop everything from computer games to network control systems. In addition to building their own systems, software engineers also test, improve, and maintain software built by other engineers.

3.1.5 Test Engineer

Software test engineer has the role of coordinating the process for analyzing software programs. This process will involve creating and implementing testing methods, recording the test results, and providing recommendations to improve software programs based on the results. Test engineers should be tasked with interfacing with end users to ascertain areas of improvement, such as cost reduction solutions and automation solutions.

3.1.6 Software Quality Assurance Engineer

A software quality assurance engineer monitors every phase of the development process to ensure that the design and software adhere to company standards. A software quality assurance engineer perform the following tasks with some regularity:document test cases and risk analysis, record test progress and results, lead team on overall testing processes(both manual and automated), identify any potential problems that users might encounter.

3.2 Software Risk Management

The risks below are broken up into the three subcategories of process risks, programmatic risks, and technical risks.

3.2.1 Process Risk Management

Process risks are problems arising from improper process implementation.

Risk	**Description**	**Mitigation Strategy**
Semantic Risks	Data is mis-stored in wrong field	data migration testing
Extended Downtime Risk	Data transfer from engine take longer then allotted time	Alert engine, and attempt to retry transfer

Table 3.1: Software Process Risk Management

3.2.2 Programmatic Risk Management

Programmatic risks include things like schedule, resources, and mismanagement.

Risk	**Description**	**Mitigation Strategy**
Inadequate Knowledge	Member(s) are unaware or familiar with SQL	Make sure all members are trained in SQL and

Table 3.2: Software Programmatic Risk Management

3.2.3 Technical Risk Management

Technical risks are threats to the quality, security, and performance of the software.

Risk	**Description**	**Mitigation Strategy**
Data Corruption	Unintended changes or errors in data that occur during transfers or storage of the data.	Utilize the log files of SQL Server to restore the database.
Data Loss	Incomplete data transfer or un-explainable loss of data	Data migration, reconciliation testing in instance of system errors or detected loss
Data Leakage	Data loss of sensitive information	implement a Password Security Policy
Inadequate Space	Data amount exceeds allotted database storage	Give alert to Dashboard when at 95 percent storage capacity
Interference risk	Multiple users attempting to access data base simultaneously	data store will limit concurrent user assesses

Table 3.3: Software Technical Risk Management

3.3 Customer Communications

A weekly stand up meeting with all members of project where software team leads of each division will summarize weekly activities and plans for the upcoming week. Meeting notes will be recorded by the team and stored in the project repository.

3.4 Team Training

There is no explicitly defined team training plan for this project. The main training sources for this project are the UNA program courses, which include training in the following areas: Methods of software development, software architecture, analysis of software artifacts, management of software development, models in software development, and optionally Database systems. As database systems was not a required course, those on the team not familiar with the domain will need to familiarize himself or herself with the basics of Database Management Systems

Training Type	**Description of Training**	**Member**	**Date**
Basic SQL Training	Basics of SQL scripts	Entire Team	
Basic Database Training	Theories and applications of database system course at UNA	Entire Team	12/07/21
Secure Coding Training	Basics of secure coding standards set by SEI CERT	Entire Team	2/16/22

Table 3.4: Training

4 Technical Approach

4.1 Development Process

The team plans to use the standard data language Structured Query Language, specifically MySQL. MySQL is an open-source relational database management system. The relational database will organizes data passed to it from the engine into one or more data tables in which data types are related to one another to structure the data. The standard SQl names and coding conventions will be used. Code will be broken into components, and each member will take responsibility for a component. Peer review will be periodically done by other members of the team. Peer reviews purpose is to remove defects as early as possible in the development process. The process includes removing defects in the original documents such as the requirement and design documents, test plans and procedures. The peer should be someone that is a technical expertise and equal standing, while the review should involve the careful judgement. The process begins with a determined scope of review.Next, a determine rigor level leading a plan to review in detail. finally, we ensure observations are resolved. The team makes an effort of considering all requirements within reviews.Reviews are sent to the technical expertise for missed expectations once discussed by entire team. Those mistakes will be corrected every week leading to useful product.

4.2 Development Tools

The standard Windows and Linux OS terminal, build in compiler and online IDE environment's compiler will be used as development tools.

4.3 COTS & FOSS Tools

As the data storage unit will be created using the MySQL relational database management software, the main environment will be the terminal. The team will utilize the MariaDB monitor that is already available on the CS network. MariaDB is a open-source GNU General Public License community-developed, commercially supported fork of the MySQL relational database management system,

4.4 Software Reuse

N/A

4.5 Testing Process

The team plans to use standard SQL testing to check that the database works as it should. blank will be done to ensure compliance with the SEI CERT style guide. Communication and accessibility tests will be conducted first. The team will test to see if the data store can communicate with the engine and the dashboard over the network. They will also test to see if the user can interact with text forms of the data from the dashboard. Once communication and accessibility tests have been passed, testing will move forward to the next step. Information retrieval and view-ability tests will be conducted next. The team will then test to see if system administrators can retrieve information regarding CPU usage, memory usage, disk usage, and network usage from the data store. Once information retrieval and view-ability tests have been passed, scale-ability and longevity tests will be conducted. The team will test to see if the system will be able to scale up to at least fifty servers. They will also test to see id data will be able to be stored for at least a year. Once scale-ability and longevity tests have been passed, data transfer tests and sanity checks will occur. The team will test to ensure that only valid data, not invalid data, makes it through the management engine to the data store. Once data transfer tests and sanity checks have been passed, dependability and security checks will be

conducted. The team will test the fault tolerance of the system. They will also test to see if information can be be saved and sent again if the server goes down. For more detailed information, please refer to the Data Store Test Plan.

5 Configuration Management

5.1 CM Responsibilities

The configuration management will fall to the software assurance engineer. The responsibilities of a configure manager consists of plan and execute CM throughout the project lifestyle, including development, migration, deployment and security. They also identify, organize, and control software and hardware configuration changes.

5.2 CM Resources

Tools for CI would include, Overleaf LaTex for the documentation and the messaging platform Discord for communication outside meetings.

5.3 Change Control Procedures

Notes are taken at weekly meetings detailing the minutes and topics discussed.

6 Verification & Validation

6.1 V&V Procedures

The verification process will be a series of tests specified in the project SRS that show the product works. Then the validation process will be a series of tests specified in the SRS that show the product build is what was intended for the greater project.

6.2 Independent V&V

N/A