Requirements Definition Document

Prepared By: Nick Faccenda
Clifford Black
David Carlin
Bryan Nunez
Jacob Kershaw
Damen Tomassi

Date: 10/14/2016

Version: *1.0*

TABLE OF CONTENTS

This document presents an understanding of the customer's current environment, current system and process, business strategy, and the customer's desire to enhance, develop, or replace an existing system and/or process within their organization. This document is targeted to identify as much of the customer's existing and proposed business requirements as possible, and is not intended to be a design document.

The objectives of this Requirements Document are:

- Identify all customer-driven technical requirements.
- To detail, evaluate, understand, and validate customer requirements and expectations.
- Uniquely identify and describe each requirement.
- Describe the changes to the existing process and all new processes.

1 Executive Summary

1.1 Introduction

This document represents the Sprint 2 written deliverable requirement. This deliverable contains:

- *System functional requirements documentation*
- *Basic System Designs*

1.2 Rules Based Decision Engine Framework System Overview

1.2.1 System Objectives

The object of the Rules Based Decision Engine Framework is to provide ASRC Federal Mission Solutions Engineering a reliable prototype for monitoring sensors by utilizing a rule based system. This system should be quick to respond, expandable, and take into consideration a large amount concurrent users. This program must also provide a simple way of adding, removing, and modifying rules.

1.2.2 Key System Functions

Based on the discussions and research during the development cycle so far, the following key functions have been defined for the Rules Based Decision Engine Framework:

- *Multiple ways to obtain user input*
- *Simple GUI to allow users to utilize this tool*
- *Efficient application use*
- *Expandable*
- *Flexible*

1.3 Rules Based Decision Engine Framework Project Overview

The Rowan Computer Science Department and Mission Solutions Engineering approved the Rules Based Decision Engine Framework project in September 2016 for the Software Engineering I class. All development is divided into Six Sprints:

Sprint 0: Requirements Definition: The purpose of this first sprint is to identify the functional requirements, User requirements, and objectives of the Rules Based Decision Engine Framework project, overall data and information requirements and the tasks and resources required for Design and Implementation.

Sprint 1-4: Systems Design: These sprints are where the development of the software will be brought together in several parts,

which the order is to be selected based on both priority assigned from the Product Owner and from what the Development Team believes is important to the overall progress.

Sprint 5: Final Implementation: During this final sprint, the final components are developed and integration of all pieces developed is completed. At the end of this sprint we will be delivering the final prototype which is the product of all prior sprints.

2 Business Objectives

2.1 Introduction

This chapter documents the scope and objectives of the Rules Based Decision Engine Framework. This chapter summarizes the system requirements expressed to the team during multiple interviews with key users and corporate management. It is organized into three sections.

Overview

This section identifies the high-level objectives to be addressed by Rules Based Decision Engine Framework. These objectives are the results of conferences between members of Mission Solutions Engineering and the Product Owner.

Corporate Goals

This project will provide a functioning prototype for a rules engine that can be revised and implemented into larger, more secure projects. Will provide a suitable test simulator and documentation for easy manipulations for future changes to the program.

2.2 Statement of Business Objectives

Build a reliable prototype of a Rules Based Decision Framework for ASRC Federal Mission Solutions Engineering that will monitor sensor data and make user-defined decisions from this data. This system should be quick to respond, expandable, and take into consideration a large amount concurrent users. This program must also provide a simple way of adding, removing, and modifying rules, as well as supporting both forward and backward rules chaining. The final deliverable must be fully bug-free, meeting all functional and nonfunctional requirements, and must meet all of the company's software standards.

2.3 Current Process

Our process consists of 3 scrum meetings a week to review progress following our version of Agile Scrum. We manage our goals and progress by keeping a trello board updated and meet with our sponsors once every two weeks for sprint reviews. We maintain an open repository on GitHub. There are no requirements for being able to run the program at this time based on current program requirements.

In between these scrum meetings we are to develop code independently or in pairs depending on the difficulty of the task. Progress is shared and new tasks taken during our 'Daily Scrums', and the team meets as a larger group outside of

these scheduled meetings once a week to help brainstorm, breakthrough, and monitor our progress as a group to find points for improvement.

2.4 Rules Based Decision Engine Framework Business Functions

This section will provide a description of all the proposed business functions/requirements of the Rules Based Decision Engine Framework system. These process descriptions will be the basis for the design phase and serve as the guide for the technical development of this system. The processes covered in this section are:

- *Evaluate multiple Rules Frameworks and determine which one provides the most flexibility while providing strong performance.*
- *Create a whitepaper design document and prototype of the Rules Framework in action executing on multiple different decision points concurrently*
- *Document the performance and scaling attributes of the Rules engine against varying degrees of decision complexity and considering different data sources*

2.4.1 Functional Requirements

This section will provide a description of each of the functional requirements in the Rules Based Decision Engine Framework system. These functional requirement descriptions will be the basis for the design phase of these functions, and serve as the guide for both the technical development of this system and the development of the prototype. The processes covered in this section are:

- *Demonstrate branched decisions within a framework instance*
- *Demonstrate how two instances can be coupled/chained together*
- *Develop a simple UI to allow one to manage, configure, activate, and deactivate a Rules framework instance*
- *Monitor multiple sensors and alarms, interchange what is being monitored, and have multiple methods of rule manipulation*
- *File output management and storage*

2.5 Rules Based Decision Engine Framework Maintenance Functions

Since we are only building a prototype, there will be no maintenance functions.

3 Technical Requirements

3.1 Hardware and System Software Requirements

No Hardware or System Software Requirements.

3.2 Software System Interface Requirements

No System Interface Requirements.

3.3 Security Requirements

No Security requirements.

3.4 Commercial off-the-shelf Software

No Commercial off-the-shelf Software is being used for this project.

4 Operational and Performance Requirements

4.1 Network or Connectivity Requirements

Connected to a sensor to receive real-time updated information for optimal performance. This will allow the data reader to update the status of monitored sensors and alarms and act accordingly without any major delays.

5 User Testing Requirements

5.1 Categorize the Business Requirements into testing priorities and risk factors.

This will include various performance stress tests, response times, how rule manipulation comes into effect, and data/alert prioritization.

5.2 User Acceptance Test Environment

As a prototype, we will be creating a simulator that will provide a test environment for the software. This will be run on multiple scales and offer us a variety of scenarios and will assist in the development in each stage of the process. This will be in addition to any and all Acceptance Criteria testing required. This environment will be stress tested at the end stage and will monitor response time, processing time, and track how it prioritizes incoming data on a severity level.

6 Implementation Requirements

6.1 Introduction

As a prototype, the requirements will be an IDE with Drools installed as a plugin

6.2 Training Requirements

There will be minimal training needed to operate this program efficiently, just a breakdown of how the Excel Sheet is set up for a simple way to add or modify rules, how to perform basic operations on the GUI, and how to create a DRL file. The rest will be controlled by a user-friendly GUI.

6.3 System Roll Out Requirements

A fully functioning prototype will be completed by December 15th, 2016. This prototype will be well documented, efficient, and delivered with an explanation of how to use all functionality included.

6.4 User Documentation Requirements

This software will be fully documented to satisfy both Rowan University's and Mission Solution Engineering's standards. This will also include a well-written tutorial on setup and how to use the UI from a user perspective.