****

Motif Test

# Automated Test Tools for GUIs

**Project Initiation Document**

**Prepared by: *The Fire Ants***

**Version: 1.3**

**Date: *09/26/2019***

**Sponsors: *Joe Mcilvaine & Michelle Darby, ASRC***

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Updated By** | **Comments** |
| 1.0 | *09/16/2019* | *Robert Dumitrescu* | First Draft |
| 1.1 | *09/26/2019* | *Robert Dumitrescu* | Final Draft |
| 1.2 | *10/01/2019* | *Robert Dumitrescu* | Added more information to ROI |
| 1.3 | *10/07/2019* | *Robert Dumitrescu* | Updated Information in ROI and added a time saving table |

# Document Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature** | **Date** |
| Project Manager | Mohamed S. Mansaray |  |  |
| Product Owner | Robert Dumitrescu |  |  |
| Scrum Master | Julio Crespo |  |  |
| Developer | Donald Zellman |  |  |
| Developer | Jennifer Anonuevo |  |  |
| Developer | George Clelland |  |  |
| Developer | Shubham Patel |  |  |

# Defined Terms

|  |  |
| --- | --- |
| **Term** | **Description** |
| The Team | Refers to the Product Manager, Scrum Master, and Development Team (see Section 7, Organization and Governance) |
| The Sponsors | Refers to the individuals who serve as representatives from ASRC Federal |

## Table of Contents

[Automated Test Tools for GUIs 0](#_Toc21382171)

[Document History 1](#_Toc21382172)

[Document Approvals 2](#_Toc21382173)

[Defined Terms 3](#_Toc21382174)

[Table of Contents 4](#_Toc21382175)

[Swot Analysis 6](#_Toc21382176)

[**Strengths** 6](#_Toc21382177)

[**Weakness** 6](#_Toc21382178)

[**Opportunities** 6](#_Toc21382179)

[**Threats** 6](#_Toc21382180)

[PEST Analysis 6](#_Toc21382181)

[**Political** 6](#_Toc21382182)

[**Economical** 6](#_Toc21382183)

[**Social** 6](#_Toc21382184)

[**Technological** 6](#_Toc21382185)

[1 Purpose 7](#_Toc21382186)

[2 Scope 7](#_Toc21382187)

[**In scope** 8](#_Toc21382188)

[**Out of Scope** 8](#_Toc21382189)

[3 Project Background 9](#_Toc21382190)

[4 Feasibility Analysis 9](#_Toc21382191)

[5 Assumptions, Dependencies, and Constraints 10](#_Toc21382192)

[5.1 Assumptions 10](#_Toc21382193)

[5.2 Dependencies 10](#_Toc21382194)

[5.3 Constraints 10](#_Toc21382195)

[6 Initial Project Plan 11](#_Toc21382196)

[***Sprint 0: Learning the environment, Product Backlog, Project Initiation document*** 11](#_Toc21382197)

[**Sprint 1**: ***Recording user selection, Requirements document*** 12](#_Toc21382198)

[**Sprint 2**: ***create a testing environment using google test, Design Document*** 12](#_Toc21382199)

[**Sprint 3**: ***performing the actions recorded in the binary file, Validation PLan with Tracebility Matrix*** 13](#_Toc21382200)

[**Sprint 4**: ***Testing different prototypes, Test PLan and Test Scripts*** 13](#_Toc21382201)

[**Sprint 5**: ***Extra Functionality, User Manual with poster and abstract*** 13](#_Toc21382202)

[7 Organization and Governance 14](#_Toc21382203)

[**Organization Chart** 14](#_Toc21382204)

[**Team Organization** 15](#_Toc21382205)

[**Project Organization Worksheet** 16](#_Toc21382206)

[8 Communication Plan 17](#_Toc21382207)

[9 Quality Plan 17](#_Toc21382208)

[9.1 High-Level Requirements 17](#_Toc21382209)

[9.1.1 Functional Requirements 17](#_Toc21382210)

[9.1.2 Non-Functional Requirements 18](#_Toc21382211)

[9.2 Deliverable List 18](#_Toc21382212)

[10 Risk Assessment 19](#_Toc21382213)

[11 Estimated Cost and ROI 20](#_Toc21382214)

[11.1 Estimated Cost 20](#_Toc21382215)

[11.2 ROI 20](#_Toc21382216)

[11.3 Working time saved 21](#_Toc21382217)

# Swot Analysis

This analysis is used extensively by many business plan managers, with the help of this analysis we have managed to exhibit the importance of this project’s functionality within the ASRC environment and possibly in society’s future.

|  |  |
| --- | --- |
| **Strengths** | **Weakness** |
| * ***Save time and money*** * ***Decrease necessary human interaction*** * ***Increase productivity*** * ***Efficiency*** * ***Standardizes a testing routine*** * ***Minimizing the human error in testing*** | * ***Dependent on accuracy of google test results*** * ***Improper recording methods*** * ***The more tests are facilitated, the larger the system will become increasing the possibility of errors*** * ***Mistimed tests*** * ***Network and environment interruptions*** |
| **Opportunities** | **Threats** |
| * ***More relaxed testing environment*** * ***Can be used for future progression in many fields*** * ***Faster software version releases*** * ***Faster software version releases*** | * ***Any security threats from motif that have not been incorporated because of it being a legacy language*** * ***Environmental and external situations*** |

# PEST Analysis

This is yet another analysis that builds on the importance of this project’s functionality.

|  |  |
| --- | --- |
| **Political** | **Economical** |
| * ***More reliable software for offensive and defensive systems in the field.*** | * ***Saving time and money*** * ***Increase productivity in other positions*** |
| **Social** | **Technological** |
| * ***Creating jobs.*** | * ***Competing with companies that have similar software and functionality*** |

# 1 Purpose

This document describes the criteria and plan for the development and design of the ***Automated Test Tool for GUIs.*** The idea behind the automated test tool that is being developed, is to drastically increase the speed and efficiency of the testing abilities of the customer. The purpose of this document is to introduce and explain how this goal can be achieved. This document will outline and explain the projects schedule as it applies to both the scope and the background of the project. Along with an explanation of the project structure and organization, this document will give a cost and benefit analysis to highlight the core reasons for the need to create this test tool.

# 2 Scope

The primary goal of the ***Automated Test Tool for GUIs*** is to be able to ***ensure the integrity of all code written for any component within Aegis, more specifically this project is meant to assist in increasing the efficiency of testing for the ASRC display system department***. To achieve this goal, we must primarily address our key objectives.

1. ***Incorporating Motif and Linux Redhat***
2. ***Utilizing googletest and googleMock***
3. ***Development of a tool that is universal among AEGIS components***
4. ***Meet or exceed ARSC(customer) requirements***
5. ***Effective delivery of quality software***
6. ***Detail to technical excellence and solid design structure***

## **In scope**

|  |  |
| --- | --- |
| 1 | ***Design an interface in motif that is able to record a series of clicks and actions and save them in a binary or XML file.*** |
| 2 | ***Create a C++ tool that is capable of taking the file as an input and playing a recording of it. The tool will scan through the binary or XML file and perform the actions as if they are physically being done*** |
| 3 | ***Using google mock, mock the functionality of the interface we want to test*** |
| 4 | ***Using google test, test the application’s GUI without needing to directly access it*** |
| 5 | ***Make sure that testing produces the same result regardless of monitor resolution, size or any other attributes.*** |

## **Out of Scope**

|  |  |
| --- | --- |
| 1 | ***Make the recording control the mouse clicks and drags all of the GUIs components before saving them to a binary file*** |
| 2 | ***Add a loading bar that tells the user what percent of the tests are complete*** |
| 3 | ***Adding an additional textbox that tells the user in live view which of the functionalities have been tested and if they pass or fail, while testing is taking place.*** |
| 4 | ***Adding an extra functionality in which the tool will not only tell the user what is wrong, but it will also tell them how to fix the problem.*** |

# 3 Project Background

This project stems from the need to quickly test development GUIs. Previous testing procedures required manually initiating GUI mouse actions, a time-intensive process. Additionally, currently available testing suites are resolution dependent, while the customer routinely develops on and for various system configurations. This program aims to automate that testing process, while integrating with Google Test and utilizing Google Mock. Previous GUI actions will be stored in a file for input to Google Test, thus drastically reducing the overhead time required to test GUIs and C++ classes.

* ***The intent of this project is to create a fast and efficient way to facilitate test automation for web GUIs***
* ***Users may save time and money by using this tool.***

# 4 Feasibility Analysis

The feasibility of this project depends on the team’s ability to utilize free and open source third-party tools, such as the Linux OS (specifically Red-Hat) X windowing environment, and the C++ Motif GUI API. Our development team feels we have the necessary resources to build this project, though there are significant learning challenges in order to provide the desired functionality (see constraints in section 5).

# 5 Assumptions, Dependencies, and Constraints

## 5.1 Assumptions

The team will be developing in the Linux operating environment. This will provide the development team with the tools and resources to ensure that they are able to build the automated GUI testing software efficiently.  The team will be using the Motif C++ GUI API to build the user interface; this provides the most relevant functionality. Motif will be used to develop the primary event-recording GUI, as well as the test GUI, due to customer familiarity, as well as customer-defined requirements.

## 5.2 Dependencies

The team is dependent on the use of Linux, specifically the X windowing system, as this is the development environment of the customer. Additionally, due to the existing experience reservoir with the customer’s developers, the C++ Motif GUI API will be used.

## 5.3 Constraints

Constraints on this project are as follows:

09/05/2019 – 12/18/2019 time window. This should be enough time for the development team to complete this project. There is no budget to be concerned with, as this is an academic project. Development team environment familiarity; a major constraint is the learning curve for our development team in using the C++ Motif GUI API to display in the Linux X windowing system. Due to limited online resources supporting this legacy API, we will require significant time to learn its implementation before proceeding with further project milestones.

Additionally, our team is currently unfamiliar with the methods/processes to track mouse movements and actions between two separate GUIs. Lastly, we have no experience using Google Test with Google Mock. Given these limitations, we have significant research requirements before beginning production.

# 6 Initial Project Plan

## ***Sprint 0: Learning the environment, Product Backlog, Project Initiation document***

**Main Goal: *The main goal of this sprint is to meet with Mr. Joe Mcilvaine, present a vision on the functionality we will be attempting to create and deploy based on the granted time. Reach an agreement with our sponsor on our product backlog, and make sure we are on the same page with what our clients are looking for.***

**Goals:**

1. ***Organize the product backlog, write user stories and tasks, and prioritize tasks. This sprint will prepare the development team for the work they will be doing throughout every following sprint.***
2. ***The goal of this sprint is for us to set up a similar working environment with that of our customers, so that we can get familiar with Linux RedHat.***
3. ***Research Motif and use it in C++ to design a generic interface with a series of different buttons, such as push buttons, radio buttons, text boxes, and combo boxes.***
4. ***Running and testing our motif GUI interface within RedHat to ensure its proper functionality.***

## **Sprint 1**: ***Recording user selection, Requirements document***

1. ***Make a record GUI interface using motif, that will contain a record button, playback button, stop etc…***
2. ***Researching user selection.***
3. ***Research tracking the mouse on the screen.***
4. ***Add functionality to the buttons of the recording interface. Make the record button, record a series of clicks and actions and other necessary state information within the recording.***
5. ***Researching saving information within a binary or XML file.***
6. ***Make the recording GUI interface save this recording in a binary file or XML file.***
7. ***Test the functionality of the Record GUI within Redhat.***

## **Sprint 2**: ***create a testing environment using google test, Design Document***

1. ***Research google test and google mock***
2. ***In C++ make an application that receives the previously generated binary file as input.***
3. ***Make this application mock the GUI interface (using mocked objects(fake objects)).***
4. ***Design the mock motif objects in C++ in order to emulate actions that are recorded in the binary file.***
5. ***Make sure the functionality of the fake objects is incorporated correctly***
6. ***Test the functionality of this application within Redhat.***

## **Sprint 3**: ***performing the actions recorded in the binary file, Validation PLan with Tracebility Matrix***

1. ***Make the application perform the actions recorded in the binary file as if they are physically done, without accessing the original GUIs interface***
2. ***Check and see if the application mocked the interface correctly.***
3. ***Make the program output a script giving feedback to the user on whether the functionality of the components has passed or failed. Extra (tell the user what is wrong or what has failed in the GUI interface.***
4. ***Test this within Redhat***

## **Sprint 4**: ***Testing different prototypes, Test PLan and Test Scripts***

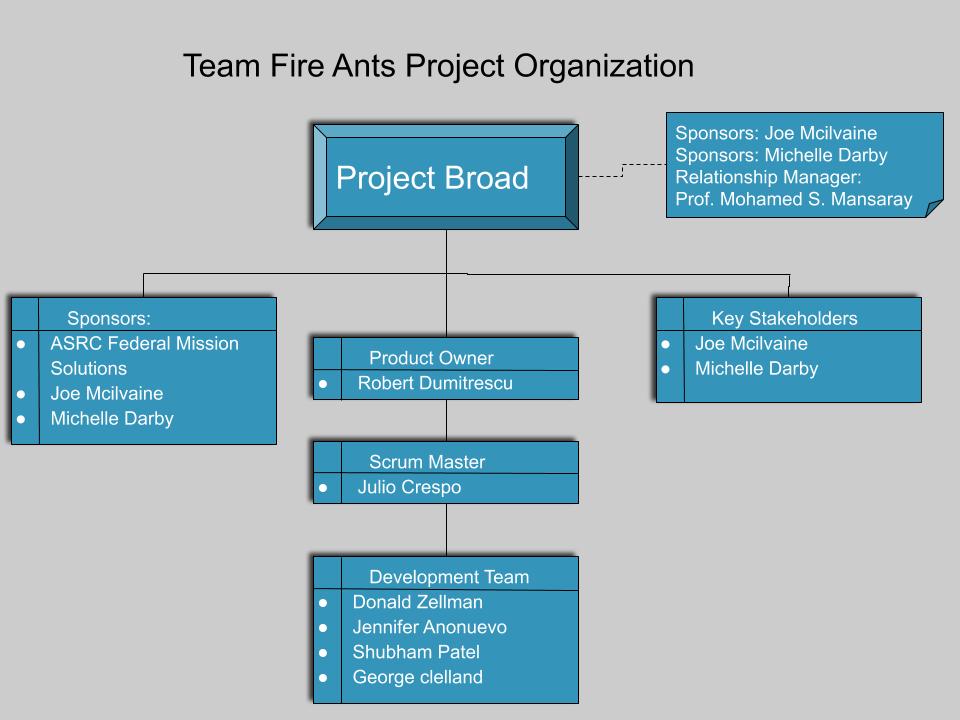
1. ***Create 2 or more GUI prototypes in motif, one of them should be a total success, the other should be a total failure.***
2. ***Test the prototypes using our C++ tool within the Redhat environment.***
3. ***Test the prototypes within Redhat on different resolution sizes.***

## **Sprint 5**: ***Extra Functionality, User Manual with poster and abstract***

1. ***In the final sprint, the development team will complete Automated Test Tool for GUIs, all required documentation, implement as many low priority features as time permits, and create a user’s manual.***
2. ***If time is within our benefit the team will try touching some of the out of scope objects listed earlier.***

# 7 Organization and Governance

## **Organization Chart**



## **Team Organization**

* The project’s contributors are ***Robert Dumitrescu, Julio Crespo, Donald Zellman, Jennifer Anonuevo, Shubham Patel and George clelland.***
* The project’s sponsors are ***Joe Mcilvaine***.
* The Team consists of three main groups: The Product Owner, the Scrum Master, and the Development Team
  + The Product Owner, ***Robert Dumitrescu***, is the primary communicator between the Team and the Sponsors. ***Robert Dumitrescu*** is also in charge of overseeing and delegating the project’s tasks to the Development Team.
  + The Scrum Master, ***Julio Crespo***, is tasked with assisting the Development Team. Julio will also ensure the Agile Scrum methodology is being followed.
  + The Development Team, consisting of ***Jennifer Anonuevo, Donald Zellman, George Clelland, Shubham Patel*** are the primary architects, designers, and coders for the project. It is their responsibility to follow the guidance set forth by the Sponsors and Product Owner and implement the project.

## **Project Organization Worksheet**

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| **Key Stakeholders** | * Owners of the project |
| **Project Manager** | * Overall management of the project time, scope and budget |
| **Project Board** | * Allocates expenditures. * Resolution of escalated issues. * Go/No go decision authority. * Ensure success. * Approve initial scope and changes to scope. |
| **Scrum of Scrums** | * Ensures that induvial team Scrum Masters are adhering to the Scrum Methodology. * Provides additional assurance of Scrum Team capabilities to key Stakeholders. |
| **Product Owner** | * Acts as the primary liaison between the project's key Stakeholders and the Development Team. * Represents the stakeholders and customers to the Development Team. * Manages the Product Backlog. * Optimizes the value of the Product. * Defines features and functionality. * **Has the final word** on the content and ordering of the Product Backlog. |
| **Scrum Master** | * Enacts Scrum values, practices, and rules throughout the organizaiton. * Ensures the Scrum Team is functional and productive - "healthy". * Provides guidance and support for the Scrum Team. * Removes impediments to the Development Team's success. * Adheres to Scrum values, practices and rules. * Understands and uses self-organization. * Increases the quality of Increments. * Manages the funcinality of the Trello Board. * Conducts risk management. * Facilitates the Sprint Retrospective to help the team reflect on their methods of development and helps them improve going forward. |
| **Development Team** | * Creates the product Increment by determining as a team what they can commit to for an iteration. * Operates in a series of Sprints. * Organizes itself and its work. * Collaborates with Product Owner to optimize value. * Individuals have not specialized designation. * Held accountable as a unit. * Authors of Epics and User Stories * Estimators of the User Stories * Define Acceptance Criteria |

# 8 Communication Plan

* The team meets for ***1 hour and 45 minutes*** daily scrums 3 times a week at the beginning of the Software Engineering class. They also meet later after that class for 1 hour for a secondary scrum.
* ***GoToMeeting, Discord, or phone*** is used for conference-call meetings 1 times a week for 1 hour on average.
* ***Sunday*** at ***12:00pm*** is blocked off by all team members for conference-call meetings each week.
* Sprint reviews will be held every two Monday at 9:30am for 30 minutes starting 9/30/19.

# 9 Quality Plan

## 9.1 High-Level Requirements

This is a list of the main high-level requirements which define the project.

### 9.1.1 Functional Requirements

* + ***Record a series of clicks and button pressing of buttons, radio buttons, sliders, text boxes, etc. into an XML or binary file.***
  + ***With the use of google mock, mock the functionality of the GUI interface for which tests are required to be run.***
  + ***Give feedback on which part of the GUI has the correct functionality.***
  + ***The software does not depend on the monitor’s resolution, monitor size, or other attributes.***

### 9.1.2 Non-Functional Requirements

* ***The software to be written in C++ using Motif.***
* ***Automated tests should function faster than a manual version***
* ***The fault tolerance is dependent on the system it is used on.***
* ***Response time is completely proportional to the amount of components the user is recording in one sitting.***
* ***Supportability is totally reliant on the customer’s software.***
* ***Test tools support different test subject matter with same efficiency.***
* ***Test system efficiency and design can handle a continuous run of all the test items.***

## 9.2 Deliverable List

* Project Initiation Document
* Requirements Document
* Design Document
* Validation Plan, with Traceability Matrix
* Test Plan, with Test Scripts
* User Manual, with Poster and Abstract

# 10 Risk Assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Affects** | **Probability** | **Effects** | **Mitigation** |
| ***Not having RedHat during testing Phase*** | ***project*** | ***Low*** | ***Tolerable*** | ***To plan ahead of time to have someone with the RedHat OS during testing phase*** |
| ***Unfamiliar with C++*** | ***Both*** | ***Low*** | ***Tolerable*** | ***Research and create sample programs with C++*** |
| ***Unfamiliar with Motif, GoogleMock, GoogleTest*** | ***Both*** | ***Moderate*** | ***Serious*** | ***Research is required at the start of the project to get familiar with them.*** |
| ***Damaging the system that is operating RedHat*** | ***project*** | ***Low*** | ***Tolerable*** | ***Have people aware of the system and to avoid damaging it.*** |
| ***A user falsely approving a buggy GUI*** | ***Product*** | ***High*** | ***Tolerable*** | ***The GUI will be reviewed before it is used for the project.*** |
| ***Not familiar with the RedHat OS*** | ***Project*** | ***Low*** | ***Tolerable*** | ***The team will practice using the operating system.*** |
| ***Unable to track the mouse movement.*** | ***Product*** | ***High*** | ***Serious*** | ***The team will research on how to track mouse movements while following the requirement.*** |

# 11 Estimated Cost and ROI

## 11.1 Estimated Cost

Assuming our sponsors can attend every sprint review for ***30 minutes***, they will spend a total of ***150 minutes*** on the project in person or via teleconference. There are no costs associated with the licensing of third-party software or equipment needed. The total cost of the project will be ***2 and a half*** hours of the sponsor’s time, with miscellaneous time added in the form of short burst communication. The software necessary to create the project is sourced at no cost to the development team and furthermore the customer.

## 11.2 ROI

As a return on the investment placed into this project, the ***Automated test tool for GUIs*** is planned to be designed intuitively enough where most common users can start using it quickly with limited prior knowledge. Initial sponsor investment does not include any direct financial obligations, due to this being an academic project development, other than 1-2 hours of requirement definition planning, and approximately 6 working-hours attending to product initiation planning and weekly scrum meetings.

Of the 100 employees that are part of ASRC’s display system department, roughly 70 percent of them will be actively engaging with this tool. Unit testing as of right now takes up to 25 percent of the labor. This equates to saving a week’s worth of work on a monthly basis, and beyond that it could save about a month of labor over the course of a year. Training time for users should be minimal. The methods that we look to encompass in our tool should make training easier with an estimated average time of 30 minutes per trainee. Since this is a tool that will be utilized well into the future, the return on the investment is very high. As the number of products and the resources allocated to testing them continues to increase, so will the resources saved by these enhanced testing methods.

* ***This program will save testers a lot of time and effort.***
* ***It would reduce the resources necessary to successfully complete testing.***
* ***Provide a detailed report automatically.***

## 11.3 Working time saved

|  |
| --- |
| **For one Employee** |
| **Type of Testing** | **Per Month** | **Per Year** | **Time Saved** |
| ***Manual*** | ***40 hours per employee*** | ***480 hours per employee*** | ***0 hours per year*** |
| ***Automated Testing***  ***(50% speed increase)*** | ***20 hours per employee*** | ***240 hours per employee*** | ***240hours per year*** |
| ***Automated Testing***  ***(75% speed increase)*** | ***10 hours per employee*** | ***120 hours per employee*** | ***360hours per year*** |
| **For 70 Employees** |
| **Type of Testing** | **Per Month** | **Per Year** | **Time Saved** |
| ***Manual*** | ***2800 hours per 70 employees*** | ***33600 hours per 70 employees*** | ***0 hours per year*** |
| ***Automated Testing***  ***(50% speed increase)*** | ***1400 hours per 70 employees*** | ***16800 hours per 70 employees*** | ***16800 hours per year*** |
| ***Automated Testing***  ***(75% speed increase)*** | ***700 hours per 70 employees*** | ***8400 hours per 70 employees*** | ***25200 hours per year*** |