# School of Creative and Digital Industries

# Software Engineering CW1 Research Report: Academic Year 2021 to 2022

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Title:** | Software Engineering | **Module Code:** | CO557 |
| **Assignment No/Title:** | Research Report (CW1) | **Assessment Weighting:** | 50% |
| **Submission Date:** | Friday 1st April 2022 Semester 2 by 23:59 | **Feedback Date:** | + 2 Weeks |
| **M****odule Tutor:** | Mike Everett | **Degree:** | BSc. (Hons) Cyber Security |
| **Student Name:** | **Jamie Chopra** | **Student ID:** | **22015094** |

## Title

**ELEMENT (3) - Create a Prototype based upon the proposed System User Interface.**

The following areas should be included: -

a) Overview of Rapid Software Development with clear explanation and illustration of the following stages:

* Establishing Objectives **(5)**
* Define Prototype Functionality **(10)**
* Development Process **(10)**
* Evaluation – usefulness and problems **(10)**

1. Requirement Document addressing the areas of:

* Subsection of the User Requirement Definition highlighting the agreed System User Interface **(10**)

1. Critical Evaluation with respect to the appropriateness of the prototyping approach**.(5)**

### Table of Contents:

[Title 2](#_Toc23108376)

[Report 4](#_Toc23108377)

## Report

When developing a Prototype utilising Rapid Software Development, it is first important to explore the method used for the development: Agile. Agile is a Rapid Software Development method in which has 3 key elements: The customer is directly involved in the testing and development. Incremental development in which includes incremental delivery where each requirement is implemented separately and tested individually, and the final element is to maintain the simplicity of code to avoid complexity when implementing changes to the application. It is important when using Rapid Software Development to keep detailed specification to a minimum and functionality of the software takes priority over the quality of the software.

The first step in Rapid Software Development is to establish objectives. In order to Establish the Objectives a process of elicitation and validation is used to evaluate and restructure each of the clients’ requirements into a programmable list of features which is then presented to the client to confirm the features will satisfy the clients requirements. A key process that can be utilised when Establishing Objectives in Rapid Software Development is to restructure the requirements into story cards (a practice of Extreme Programming).

Extreme Programming (XP) is an accelerated version of Agile in which increments are implemented more often in the form of small releases. Extreme Programming requires a more structured plan for the developers which is fulfilled in the form of Story Cards. Story Cards records each requirement and organise them based upon the time to develop and their priority.

Incremental development is another process in which each requirement is broken down and organised for implementation based on their priority/functionality to the customers’ requirements, incremental development first focuses on creating the best understood requirements.

Prototyping is creating a system to test design options for fulfilling the users’ requirements, prototypes are throwaways meaning the system is discarded once the best route for system specification has been identified. Prototyping focuses on the least understood requirements and the tests to be carried out on the application are created prior to the prototype, this is known as test-first development and is implemented so that the prototype can be developed around passing the tests (thereby suiting the requirements of the client).

Each test within the test-first development process of prototyping is coded as a program in which can be automatically executed when needed to test an application. Once the application has been developed enough for the test to succeed, the developers will move onto the next test in the chain and start to develop code to successfully pass the next test, this format allows a coder to retain simplicity of code and minimise its length during the development of an application and provides a format whereby each piece of code is comprehensible throughout a development team allowing anyone within the team to later edit/refactor it.

Overall, the advantages of Rapid Software Development are the ability to have an interactive development process between developers and client to ensure all requirements are met/satisfied and each application is developed with speed and simplicity.

Rapid Software Development also lays the foundation for the further development of the software as the development process is based upon the anticipation of constant changes and therefore will reduce costs in upgrading the software in the future of the software life cycle.

Another important benefit is the improved design quality that is discovered through prototyping, the constant testing and redesigning of the prototype allows the developers to identify the most efficient design for the software to minimalize maintenance costs and reduce future development effort.

However, Rapid Software Development in turn presents the disadvantages whereby a system when created is not best suited for functionality as it may only meet the minimum requirements for each function of the application and therefore may not be scalable/applicable for real-time use.

The developers are also not concerned with non-functions meaning they are not interested in developing any extra requirements outside the scope of the clients’ stated requirements. This therefore limits the extent to which an application developed under Rapid Software Development is user-friendly or operable by a larger audience.

The first stage in the development process of the Fire and Security Alarm Monitoring System (FASAM) was to establish the objectives, as presented below each of the client’s requirements 1-5 have been reconstructed into a list of features in which will be programmed to create a fire alarm application. These features are as followed:

1. If multiple sensors identify a fire automatically “Confirm” a fire and alert emergency services and perform security features
2. Create an option to “Confirm” a fire connected to the activation of a sprinkler system to shut down electrical equipment at all exits of the premises
3. Automatic activation of lights in direction indicators and cue the recorded sound Alarm alerting staff to evacuate once a fire has been confirmed.
4. Lock fire doors (chosen by the client) once a fire has been confirmed
5. Implement the calling of emergency services into the "Confirm" button for a fire

These features will be demonstrated through a throwaway UI prototype for testing purposes of the client and will present a visual table of sample rooms (chosen by the client), an option to “Confirm” a fire alert for each room, and a status table for each Alarm system including Door Closed/Open, Sprinkler On/Off, Direction Indicators On/Off and Alarm On/Off. It will also print a message within the UI when the Emergency Services have been alerted of the fire.

In order to structure the development process of this application, story cards as shown below have been created, along with the creation of their test cards (for test-first development) with the results of the application developed:

**Task** **Test**

|  |
| --- |
| Task 1: If multiple sensors identify a fire automatically “Confirm” a fire |
| 1. If the control area is unmanned and an alarm is activated this alarm should not be ignored if it is potentially serious. Emergency services should be called automatically. |

|  |
| --- |
| Test 1: Multiple sensors identifying a fire |
| Expected Result:  Fire is automatically confirmed |
| Actual Result:  Fire identified by sensors in both the Board Room and the Chairman Room causing the MULTIPLE SENSORS CONFIRM to activate  Graphical user interface, text  Description automatically generated |

|  |
| --- |
| Task 2: Create an option to manually “Confirm” a fire connected to the activation of a sprinkler system and shutdown the electrical system |
| 1. Some but not all parts of the building may be equipped with sprinkler systems or systems to shut down electrical equipment. These should be activated if a fire alarm is confirmed. They should not be activated if there are people in the same room. |

|  |
| --- |
| Test 2: “Confirm” option to manually activate the sprinkler system and shutdown the electrical system |
| Expected Result:  Sprinkler System – ON  Electrical Equipment – SHUT DOWN |
| Actual Result:  Input ‘F’ to confirm fire  Sprinkler System – ON  Electrical Equipment – SHUT DOWN  Text  Description automatically generated |

|  |
| --- |
| Test 3: “Confirm” option automatically turns on Directional Indicators and sounds Alarm |
| Expected Result:  Alarm – ON  Directional Indicators - ON |
| Actual Result:  Input ‘F’ to confirm fire  Alarm – ON  Text  Description automatically generatedDirectional Indicators - ON |

|  |
| --- |
| Task 3: Automatically turn on Directional Indicators and cue the recorded sound Alarm once fire has been “Confirmed” |
| 1. The building may be equipped with direction indicators, which illuminate the route to the nearest exit. These should be activated when a fire alarm is confirmed. At the same time, an audible signal should sound alerting occupiers to leave the building. |

|  |
| --- |
| Test 5: Emergency Services contacted when fire has been “Confirmed” |
| Expected Result:  Message appears stating Emergency Services have been alerted after:  Fire Warning - CONFIRMED |
| Actual Result:  “EMERGENCY SERVICES CALLED”  Text  Description automatically generated |

|  |
| --- |
| Task 4: Automatically Lock fire doors (chosen by the client) once a fire has been “Confirmed” |
| 1. A security alarm may cause some internal doors to be locked automatically. It should be possible to isolate complete zones by automatic door locking. |

|  |
| --- |
| Test 4: Fire doors locked when fire has been “Confirmed” |
| Expected Result:  FIRE WARNING - CONFIRMED  Fire Doors - CLOSED |
| Actual Result:  Fire Doors - CLOSED  Text  Description automatically generated |

|  |
| --- |
| Task 5: Only contact emergency services once a fire has been “Confirmed” |
| 1. False alarms are common, and it might be normal practice to have an alarm confirmed before alerting emergency services. There are different ways of confirming an alarm. In the case of a fire alarm, multiple sensors detecting a problem may confirm it. |

Graphical user interface, text

Description automatically generatedThe application can be improved upon by the implementation of a more user-friendly UI which will be necessary for the focus group (workplace employees) in which the input is not command-line but rather a button-based interaction to ensure any user understands the software and can perform the operations included within the software.

Another issue with the application that can be improved upon is the notification that a fire has been identified, as currently it is presented by a change in numbers (from 0 to 1), however a more visually representable notification such as a pop-up would make the application easier to use, however this is out of the scope for a Rapid Software Development task as they focus mainly on the operability of an application as opposed to its user-friendliness.

Overall, this report has explored the fundamentals of Rapid Software Development, defining each stage. It started by establishing objectives and the process of elicitation and validation between a client and development team to create an application that suits the clients’ requirements, whilst upholding the simplicity and operational processes of Rapid Software Development.

It then defined the role of prototyping and the creation of throw-away applications in order to identify the best route for development of the application, prototyping was also discussed through test-first development process in which a prototype is created based on a series of tests that fulfil the clients’ requirements.

The processes for development were explored through interactive development and the constant exchange between a client and development team during Rapid Software Development, this is accomplished through both the utilisation of Story Cards and breaking down each requirement into smaller tasks which can each be individually tested with the client to ensure the application is designed to the client’s requirement specification.

The advantages and disadvantages of Rapid Software Development were also discussed to identify the key components of the development which includes applications in which are created directly to meet the requirements of the client, the advantages of this allow for speed of production of the software and simplicity of code, however it is juxtaposed by the limit to which the application can be used in real-time.

Each stage of the Rapid Software Development process was demonstrated in the creation of the FASAM application, starting with the elicitation and validation process of identifying each feature that must be created to fulfil the clients’ requirements.

These features were then implemented into the development process via the creation of Story Cards and Test Cards to facilitate a test-first development of a prototype fire security system. The test cards also included screenshot-based evidence of the functionality of the application and the implementation of the requirements of the client being fulfilled.

Finally, improvements for the application were explored which mainly included features that would not satisfy a Rapid Software Development process. Therefore, the application and Rapid Software Development has demonstrated its ability to satisfy the scope of the fire alarm applications requirements.