Sebastian perry

S5132483

2805ICT PACMAN requirements analysis

Contents

[Activity 1 2](#_Toc46442511)

[Activity 2 2](#_Toc46442512)

[Activity 3 2](#_Toc46442513)

[Activity 4 3](#_Toc46442514)

[Activity 5 3](#_Toc46442515)

[Appendix 1: References 4](#_Toc46442516)

# Functional Requirements

Below is a list generated of the functional requirements for the project.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Swift:**

Swift offers contain a large portions of C’s features, and more; which has empowered developers to make large complex software systems more efficiently and with less time consuming errors when compared to c. in addition to all the basic data types offered by C, Swift also offers the data type known as tuple which it useful for quickly passing around a number of variables. Swift is known as a type safe language which allows the user to specify what types parts of you code can work with; if swift detects that the code is using an unspecified type such as an optional int instead of a non-optional int, swift will stop the code and return an error. Features such as type safe allow developers to efficiently develop software by finding and squashing bugs early.

# Activity 2

What follows is a list of just a few benefits to using a version control system such as git:

1. It allows easy backups to be made of a project.
2. It can allow multiple programmers to work on the same project at any one time.
3. It allows programmers to branch of the main build and make potentially breaking changes while preserving the original working copy.
4. It allows a team to easily see what changes were made in a given commit, enabling them to more easily debug and changes that may have broken the project.

# Activity 3

Ariane 5

In 1996 on the 4th of June, 37 seconds after launch of the first the first Ariane 5 rocket, a software fault caused the rocket to flip 90 degrees in the wrong direction and get torn apart and consumed in a gigantic fireball of liquid hydrogen due to the aerodynamic forces involved. This software fault cost approximately $370m. It was later discovered that this fault was caused by the parsing of a floating-point number from a 64-bit variable to a 16-bit variable. This number known as the horizontal bias told the computer system if it was pointing up or down, at the start of the launch when the velocities were relatively low the number worked as intended, it was only later when the velocities got higher that the problem revealed itself.

The development team failed in several areas:

* They didn’t case check all aspects of the software to ensure it was performing predicted behaviour at all times.
* This software was actually reused from the Ariane 4, the team failed to inspect the software to insure it would work properly with the new hardware.
* They failed to readjust the software based on its revised requirements, as the Ariane 5 was set to take off at a steeper trajectory then the Ariane 4

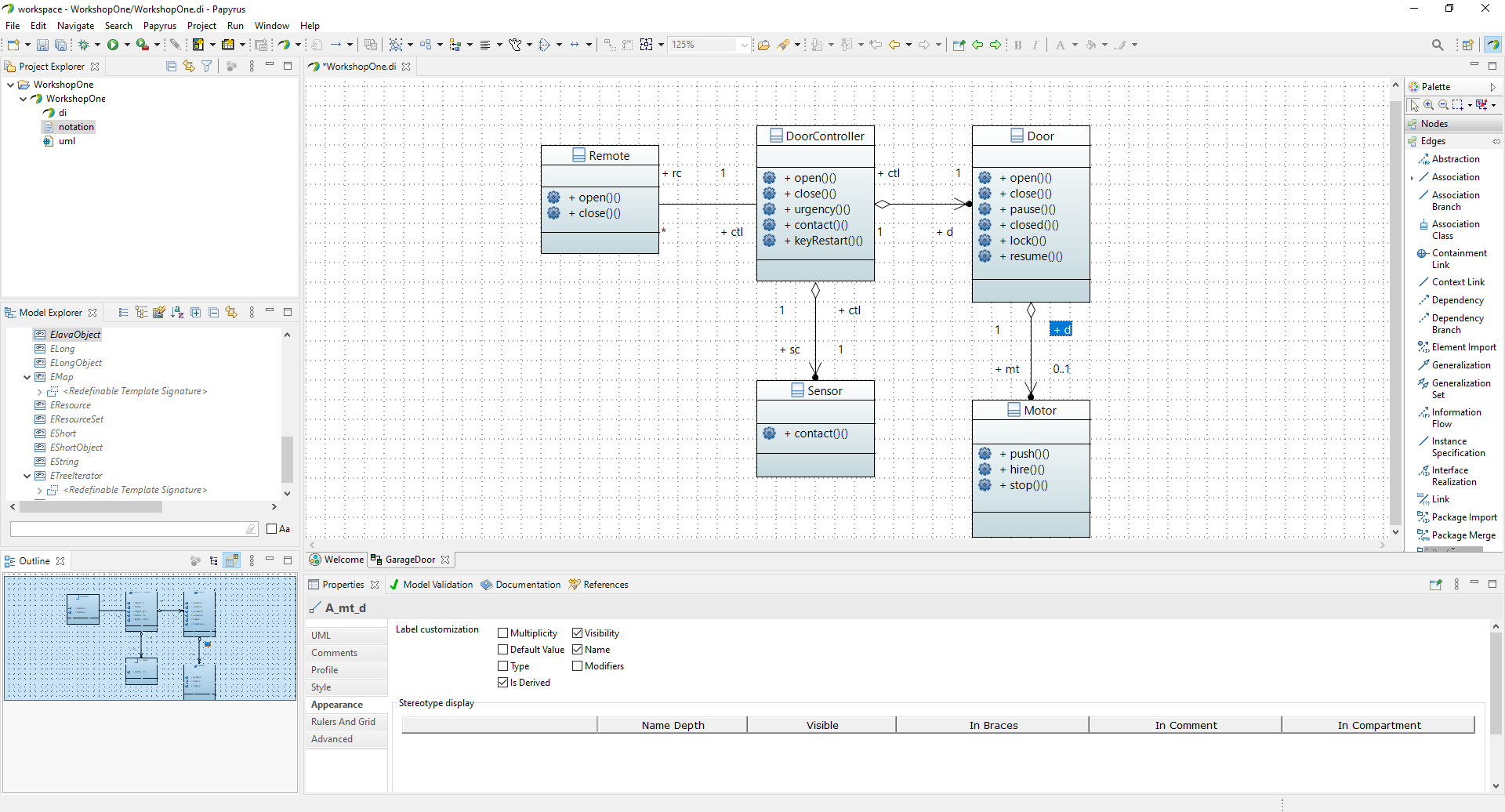
# Activity 4

Software building is such a complex activity as it requires a vast array of different skills in many different areas displayed by all team members to achieve the goal. Without proper planning a team could find that the direction to success gets harder and harder as the project draws on. The plan must cover: The projects requirements, how the project should be carried out, when should certain tasks be completed and who should complete them.

Furthermore, the plan should be revisited and revised regularly throughout production, and especially if new project requirements come to light. However, even if a team meets all these requirements, the project can still fail for a multitude of different reasons such as: poor team cohesion, lack of communication with the client, of communication with the upper management, or inaccurate estimates on the amount of time or money needed to complete the project.

# Activity 5

1. Papyrus was downloaded and made operational on the computer.
2. Basic Papyrus guides were observed.
3. The diagram shown in figure 1 was recreated and screenshot proof of the fact will be displayed below:



# Appendix 1: References

Lynch, J., 2017. The Worst Computer Bugs In History: The Ariane 5 Disaster | Bugsnag Blog.

[online] Bugsnag.com

<https://www.bugsnag.com/blog/bug-day-ariane-5-disaster>

[Accessed 23 July 2020]