*6.7.1 Work Breakdown Structure*

Automatic Calve Feeder/Dispenser

Conclusion + Reflection

System Validation

System Verification

System Implementation

System Design

Requirement Control Documentation.

Acknowledgements

Introduction

Final Report

User Testing

System Testing

Unit Testing

Integration Testing

Testing

Implementation

Requirement Gathering

Requirement Justification + Analysis

Product Design

Design

Risk Assessment

Link Simulation to Raspberry Pi.

Create Simulation on LabVIEW.

Link Graphical User Interface to Database.

Create Database for Graphical User Interface.

Configure Raspberry Pi with Touch Screen Module.

Create Graphical User Interface.

Initial Plan

Literature Review + Analysis

Project Objectives

Project Context + Aims

Initial Report

**6.1 Justification of Software Lifecycle Methodology:**

In the software industry today, many companies use a different approach to each project when it comes to which lifecycle they choose to use. There are several different methodologies that can be used. Some methods many be more applicable to a project than others. Below I have summarised the benefits and cons of each method to help me decide which method I think will be appropriate to use and implement in a project.

|  |  |  |
| --- | --- | --- |
| LifeCycle Model | Advantages | Disadvantages |
| Agile | Changes to Requirements can be adapted at any stage of the development process. | Can be difficult with large projects to access effort required. |
|  | Feedback is display throughout whole process meaning a better product is delivered, meeting all customers’ needs. | Project may be taken of course if customer doesn’t know a final clear outcome. |
|  | Testing conducted throughout development so problems can be identified early. |  |
|  | More customers’ interaction during development. |  |
|  |  |  |
| Waterfall | Simple and Easy to understand and implement. | Can be very hard to go back and implement a change if not identified at the concept stage. |
|  | Management of project is easier as each phase has a specific deliverable. | Can be hard to work with complex projects or object-oriented projects. |
|  | Phases do not overlap. | Poor model to use for long or on-going projects. |
|  | Ideal for smaller projects. | Not suitable for projects in which requirements have a high risk of changing. |
|  |  |  |
| Incremental | Customer can respond to each build. | Needs good planning and design. |
|  | Lowers initial delivery cost. |  |
|  | Easier to debug and test during a smaller iteration. |  |
|  |  |  |
|  |  |  |

After looking at the table above and considering all the points, for this project I think I will use an agile methodology. I have chosen this methodology as this model will allow for changes in requirements at any stage of the development. For me, this is the main reason why I choose Agile. It means if I have missed anything or the stakeholders would like anything added it can be implemented with limited impact on the whole project. Another key feature this model offers is more interaction with the customer through feedback and development. This will hopefully lead to a better designed product which will meet all of their desired requirements and exceed expectations. One final point which influenced my decision to an agile method was the fact that testing occurs throughout the development process meaning problems may be identified earlier. Thus ensuring when delivering the final product it will not be delayed.

|  |  |
| --- | --- |
|  | Initial Report |
|  | Design |
|  | Implementation |
|  | Testing |
|  | Final Report |

*6.7.2 Gantt Chart:*

*6.7.3 Resources Identification:*Below I will highlight what resources will be needed for this project:   
 1. Laptop / PC *(Hardware)*  
 2. PyCharm IDE (Integrated Development Environment) *(Software)*  
 3. Raspberry Pi 3 *(Hardware)*  
 4. Touch-Screen Module for Raspberry Pi *(Hardware)*  
 5. LabVIEW *(Software)*  
 6. Github Repositories *(Software)*

1. Laptop / PC: These both will be used for the day to day development of project.

2. PyCharm: This IDE will be used for the development of the Graphical User Interface and Python code needed for the project. It will help ensure the code is clean and easily readable. PyCharm can connect to Github allowing for Version Control of code.

3. Raspberry Pi 3: This will power the system and the GUI will also be powered of this.

4. Touch – Screen Module for Raspberry Pi 3: This is used to allow the User to interact with the system and select available functions.

5. LabVIEW: This is used to create the simulation part of my project. It allows me to use logic to create simulation.

6. Github Repositories’: This is free online software in which I will save my project to. This will allow the use of version control to be used in the project.