

EX-SELL

Project Plan

Version 1.2 approved

Prepared by Zenith

Yiu Hong Sum U1621435G  
Lim Kian Hock Bryan U1620949L  
Cheng Yang Zhen U1521618K  
Lim Yan Jun U1622311B  
Soong Jie Ming U1521535B  
Chen Zhenni U1622603D  
Chang Keat Lueng Aaron U1622360F

13th April 2018

**Document Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Description of Change** | **Approved by** | **Date** |
| **0.10** | **Introduction, Project Organization** | **Hong Sum** | **06/03/2018** |
| **0.20** | **Resource Allocation** | **Bryan** | **07/03/2018** |
| **0.30** | **Process Definition** | **Aaron** | **09/03/2018** |
| **0.40** | **Schedule** | **Bryan** | **09/03/2018** |
| **0.50** | **Project Estimate** | **Aaron** | **10/03/2018** |
| **0.60** | **Product Checklist** | **Yang Zhen** | **12/03/2018** |
| **0.70** | **Best Practice Checklist** | **Hong Sum** | **14/03/2018** |
| **0.80** | **Risk Analysis** | **Zhen Ni** | **14/03/2018** |
| **0.90** | **Quality Assurance** | **Jie Ming & Yang Zhen** | **15/03/2018** |
| **1.00** | **Monitoring & Control** | **Yan jun** | **15/03/2018** |
| **1.10** | **Final Compilation** | **Hong Sum** | **16/03/2018** |
| **1.20** | **Amendments to calculations** | **Bryan** | **03/04/2018** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Content**

[1. Introduction 5](#_Toc509014425)

[1.1. Project Overview 5](#_Toc509014426)

[1.2. Project Description and Scope 5](#_Toc509014427)

[2. Project Organization 5](#_Toc509014428)

[2.1. Team Structure 5](#_Toc509014429)

[2.2. Roles and Responsibilities 6](#_Toc509014430)

[2.3. Team Communication 7](#_Toc509014431)

[**3** **Resource Allocation** 7](#_Toc509014432)

[3.1. Hardware / Software Resource 7](#_Toc509014434)

[4. Process Definition 7](#_Toc509014435)

[4.1. Lifecycle Model 7](#_Toc509014436)

[4.2. Capability Maturity Model 8](#_Toc509014437)

[4.2.1. Key Process Areas 8](#_Toc509014438)

[4.2.2. Description of covered key process areas (Level 2 & 3) 9](#_Toc509014439)

[4.2.3. Detailed Process Definition 11](#_Toc509014440)

[5. Schedule 18](#_Toc509014441)

[5.1. Activity Dependencies and Schedule 18](#_Toc509014442)

[5.2. Work Breakdown Structure 19](#_Toc509014443)

[5.3. Work Packages 19](#_Toc509014444)

[5.4. Activity Dependencies 20](#_Toc509014445)

[5.5. Work Package Details 20](#_Toc509014446)

[6. Project Estimates 25](#_Toc509014447)

[6.1. Code Size Estimation using Function Points 25](#_Toc509014448)

[6.1.1. Unadjusted Function Points 25](#_Toc509014449)

[6.1.2. Adjusted Function Points 28](#_Toc509014450)

[6.1.3. Lines of Code 29](#_Toc509014451)

[6.2. Efforts and Duration Estimation 30](#_Toc509014452)

[6.2.1. Constructive Cost Model 30](#_Toc509014453)

[6.2.2. Top-Down Estimation 31](#_Toc509014454)

[6.2.3. Bottom-Up Estimation 32](#_Toc509014455)

[6.2.4. Distribution of Effort 32](#_Toc509014456)

[6.3. Cost Estimation 33](#_Toc509014457)

[6.3.1. Hardware: 33](#_Toc509014458)

[6.3.2. Software: 33](#_Toc509014459)

[6.3.3. Other Resources: 34](#_Toc509014460)

[7. Product Checklist 34](#_Toc509014461)

[8. Best Practice Checklist 35](#_Toc509014462)

[9. Risk Analysis 36](#_Toc509014463)

[10. Quality Assurance 38](#_Toc509014464)

[11. Monitoring & Control 39](#_Toc509014465)

# Introduction

## Project Overview

Ex-sell is an e-commerce web application providing a platform for buying and selling of past and current university resources. The service provided is exclusively for existing NTU students and staff via the web. Users who are interested in a listing will be able to directly message the seller for any enquiries.

## Project Description and Scope

As the target audience are NTU students, transaction can be done within the compound of NTU, providing greater ease of convenience for buyers and sellers to complete the transaction. Available on both mobile and the web, Ex-sell takes advantage of the widespread availability of smartphones to build an effective communication channel between NTU students. The app aims to be the platform-to-go for reliable buying and selling of school necessity. Users will be able to buy and view the list of items being sold by others; for each items, buyers can save it to their favourite list, chat with the seller and buy the item.

Due to the nature of the system, time and budget constraints, Ex-sell comes with several limitations:

* The system’s main source of items is from user listings. As such, the products might be limited.
* Given the limited period and human resources, the team decided to focus on developing the web app for browsers, the most popular platform first. This shortens the development cycle and allows the team to gather early user feedback.
* Development for the Android and iOS platform will start when the Web Application meets the expectations.

# Project Organization

## Team Structure

The following is the list of executive roles, as required by CMM level 3.

* Project Manager: Yiu Hong Sum
* Release Engineer: Chang Keat Lueng Aaron
* Lead Developer: Bryan Lim Kian Hock
* Front-end Developer: Cheng Yang Zhen
* Back-end Developer: Lim Yan Jun
* Quality Assurance Manager: Soong Jie Ming
* Quality Assurance Engineer: Chen Zhenni

## Roles and Responsibilities

**Project Manager: Yiu Hong Sum**

* Oversees project progress
* Approves and executes project plan
* Assigns tasks and reports status of project to team members
* Manages and motivates team members
* Represents the team/firm during external meetings with clients/external parties.

**Release Manager: Chang Keat Lueng Aaron**

* Provides progress update to the CTO
* Implements and manages release processes
* Manages risks and resolves issues affecting release schedule
* Communicates all key project plans. commitments, and changes

**Lead Developer: Bryan Lim Kian Hock**

* Designs logical system based on requirements
* Translates logical design into detailed design
* Creates detailed design document

**Front-end Developer: Cheng Yang Zhen**

* Implements product based on detailed design document
* Integration of coded modules into functioning system
* Manages the user-facing functionality of the product

**Back-end Developer: Lim Yan Jun**

* Implements product based on detailed design document
* Integration of coded modules into functioning system
* Implements the server-side web application logic

**Quality Assurance Engineer: Chen Zhenni**

* Ensures acceptable software quality
* Designs testing strategies
* Creates and manages test plan
* Verify software requirements
* Executes test procedures

## Team Communication

Ex-Sell communication channels include the following:

* Weekly meetings are held on Thursdays.
* Group announcements and updates are sent through WhatsApp.
* Project documents are regularly updated on GitHub and the Wiki page.
* Source code and issues are tracked on GitHub.
* Video conferences are held as necessary.
* Split up into subgroups as necessary, to work more cooperatively on specific problems.

1. Resource Allocation

## Hardware / Software Resource

* The basic software required for this project includes Microsoft Office Word 2016 for documentations, Microsoft Project 2016 for project scheduling.
* There will be one server and one web-based client for project deployment.
* The technologies that we will be using for the implementation of Ex-Sell are the IIS server, .NET and MySQL, server running on HP machine.
* Front-end web-based application will be done on one team member’s computer, which consist of I3 CPU with internet connections. The software packages include (1) Open source Bootstrap framework for website HTML, CSS and JS interface design.
* Client website testing will be done on one of the team member’s computer, which consist of I3 CPU with internet connections. The basic software packages includes (1) Google Chrome, Adobe flash player plugin.

# Process Definition

## Lifecycle Model

As Ex-sell is a ready to market E-commerce system, it is of the utmost importance that we must properly agree on the requirements and functions which the system possess and have. This would allow the completed system to fulfil the business requirements which the system should achieve and prevent unnecessary time wastage should there be any changes or addition to the requirements of the system.

The Software Development Model which we will be using would be the Software Prototyping Model, which essentially require constant feedback and showcase of prototypes to the customers to ensure that the system will have all the required functions before further development is done to fully complete the functions.

Requirement Elicitation will be done during the Initial Requirements stage, where we will elicit the requirements necessary for the system, and formally specify them in the System Requirement Specification Document. Next, continuous prototype design and evaluation will be done between the development team and client company, to ensure that both sides agree with the system and the functions that are required, before moving into development phase. Afterwhich, the development will start proper. Tests and maintenance will be done to ensure that the system will function properly while preventing any failures detected.

## Capability Maturity Model

As Ex-Sell may be a marketable application, our team intends to achieve a maturity level of 3 for our software development process, according to the Capability Maturity Model. As the process is defined by a collective group, this would ensure that we would have a stable build with consistent implementation. As level 4 and 5 is known to be very complex, the limitation and strict time frame of the project would mean that it would not be feasible, also taking into account that our team do not have enough experience to operate on those levels.

### Key Process Areas

The Key Process Areas (KPA) for this maturity level include:

**(Level 2)**

* Requirements management
* Software project planning
* Software project tracking & oversight
* Software quality assurance
* Software configuration management

**(Level 3)**

* Organization process focus
* Organization process definition
* Training program
* Integrated software management
* Software product engineering
* Intergroup coordination
* Peer reviews

However, given the nature of our project and team, the following KPA(s) are not applicable:

* Intergroup coordination: the project will only be developed and implemented by one team, there will not be a need for coordination amongst groups.
* Software subcontract management: there will not be subcontracting of modules or parts involved as the team is responsible for the entire project.
* Training Program: the project requires on-the-job learning and application. Given the limited period, there is insufficient time and resources for a training program.

### Description of covered key process areas (Level 2 & 3)

**Level 2**

* **Requirements Management**

The team will refer to the lab manual as well as assistance from lab supervisors to ensure the correct understanding of the requirements for the project.

* **Software Project Planning**

Software project planning will be done by the entire team, agreed upon during team meetings to ensure a consistent understanding for the project implementation.

* **Software Project Tracking & Oversight**

Software project tracking and oversight will be performed by the Project Manager, who will ensure that all deliverables and code are completed on time for the project. This is in accordance to the lab sessions and the necessary deliverables stated in the lab manual.

* **Software Quality Assurance**

The Quality Assurance Engineer will ensure software quality assurance. The QA Engineer will define and conduct the necessary tests to ensure the quality of our product.

* **Software Configuration Management**

Software configuration management will be performed by the Release Engineer, who will create baselines for the product. Changes to documentation will be tracked via the Revision History.

**Level 3**

* **Organization Process Focus**

The team will continually streamline our processes and work methods along the duration of the project as deemed necessary. This will ensure optimal work capability.

* **Organization Process Definition**

The team has decided on a set of rules and guidelines, such as attending weekly meetings and completing deliverables on time to ensure the smooth and constant flow of the project.

* **Integrated Software Management**

The team will meet regularly (weekly) to ensure that both documentation and code are in line with each other, and that everything is coherent and up to date.

* **Software Product Engineering**

This process defines the technical activities of the project, such as requirements analysis, design, code, and test. These activities will be as per required from the deliverables stated in the lab manual to produce a functioning product along with appropriate documentation at the end of the term.

* **Peer Reviews**

During each team meeting, the team will conduct peer reviews on each of the members’ deliverables. The peer review process follows two methods:

* Walkthrough review, where the member in charge (author) of the deliverable will go through the content of the deliverable with the rest of the team
* Inspection review, where members will be tasked to go through the content of the deliverable beforehand, and come to the meeting prepared with any necessary changes and fixes

### Detailed Process Definition

|  |  |
| --- | --- |
| **Process 01** | **Requirement Management** |
| Entry Criteria | Specify formal requirements  Define requirements |
| Process Description | This process defines procedures to establish the requirements document. It ensures the quality and completion of the requirements documentation |
| Exit Criteria | Well-built requirement specification |
| Begin              End | -Obtain requirements from customer through formal or informal way  -Analyse requirements  -Define requirement feasibility and testability  -Prevent ambiguities in specification  -Interpret and clarify requirements  -Specify acceptance criteria  -Allocate requirements for processes  -Review the requirement |

|  |  |
| --- | --- |
| **Process 02** | **Software Project Planning** |
| Entry Criteria | Assign project manager, working standards Project description |
| Process Description | The Software Project Planning Process defines the working procedures and the project goals and commitments. It considers the quality management, and analyses risk and proposes steps to mitigate those risks. |
| Exit Criteria | Approved Software Project Plan |
| Begin                  End | -Plan the project and design the process structure  -Establish project schedule according to lab schedule  -Establish project commitments  -Work breakdown structure  -Assign each team member tasks  -Estimate project timeline  -Plan risk avoidance and mitigation  -Plan project quality management  -Plan for change control  -Plan testing |

|  |  |
| --- | --- |
| **Process 03** | **Software Project Tracking and Oversight** |
| Entry Criteria | Reach a project milestone, Plan for process review, set the data and identify review process |
| Process Description | This process defines the procedure of improving project process |
| Exit Criteria | Revised process definition  Statistical analysis data for the revision of process |
| Begin            End | -Review processes at milestones  -Document review data  -Revise project process  -Gather process data  -Compare actual and predicted errors/schedules and resources  -Document the result  -Conduct process improvement |

|  |  |
| --- | --- |
| **Process 04** | **Software Quality Assurance** |
| Entry Criteria | Establish the quality goals  Define quality measurements  Define quality statistics metrics |
| Process Description | Define the management principle to ensure that quality must be measured and managed, to ensure that quality assurance is applied through the whole lifecycle of software development |
| Exit Criteria | Desired quality standard is met and have a well documented quality statistic for future reference |
| Begin          End | -Plan quality assurance  -Establish quality goals  -Define quality quantitative metrics  -Track project quality activities  -Document all qualities metrics  -Compare result with the goals and do an analysis |

|  |  |
| --- | --- |
| **Process 05** | **Software Configuration management** |
| Entry Criteria | Establish configuration standard  Identify configuration tools |
| Process Description | This process focuses on standardizing the management of every set of configuration items needed for each release. A collated document will be produced to indicate the revision number of a module that is needed for a certain release build. This will encourage programmers to supply comments regarding revisions and log the date, time and the programmer |
| Exit Criteria | Configuration status report |
| Begin            End | -Establish configuration management standard  -Identify configuration items  -Maintain configuration item descriptions; Control change requests; Version release control;  -Maintain configuration items history  -Report configuration status  -Do a review  -Document the review result and make appropriate changes |

|  |  |
| --- | --- |
| **Process 06** | **Organization Process Focus** |
| Entry Criteria | Assess current process, identify improvement opportunity  Define the scope of improvement |
| Process Description | The Process improvement defines the necessary procedures to ensure the improvement on process. It identifies the improvement scope, and defines the measure of the impact. |
| Exit Criteria | Improved process  Report the improvement of the new process |
| Begin                End | -Plan the improvement of the process  -Access the current process, identify the opportunity  -Define the scope of improvement  -Prioritize the improvement  -Define the measures of impact  -Change process accordingly  -Conduct pilot trial of new process  -Assess new process  -Document the improved process report and implement the new process if there is improvement |

|  |  |
| --- | --- |
| **Process 07** | **Organization Process Definition** |
| Entry Criteria | None |
| Process Description | This process sets up the basic structure of the team for the project. Roles are decided, as well as general organizational standards and practices according to the lab manual and template documentations. |
| Exit Criteria | Establishment of teams and project managers  Standardization of report and documentation of processes  Define roles  Established standard processes of the team |
| Begin      End | -Define teams, including selection of project manager  -Define project management responsibilities of each member.  -Establish the standards of reporting - how to define processes, success, failure, control points, milestones, quality control, metrics |

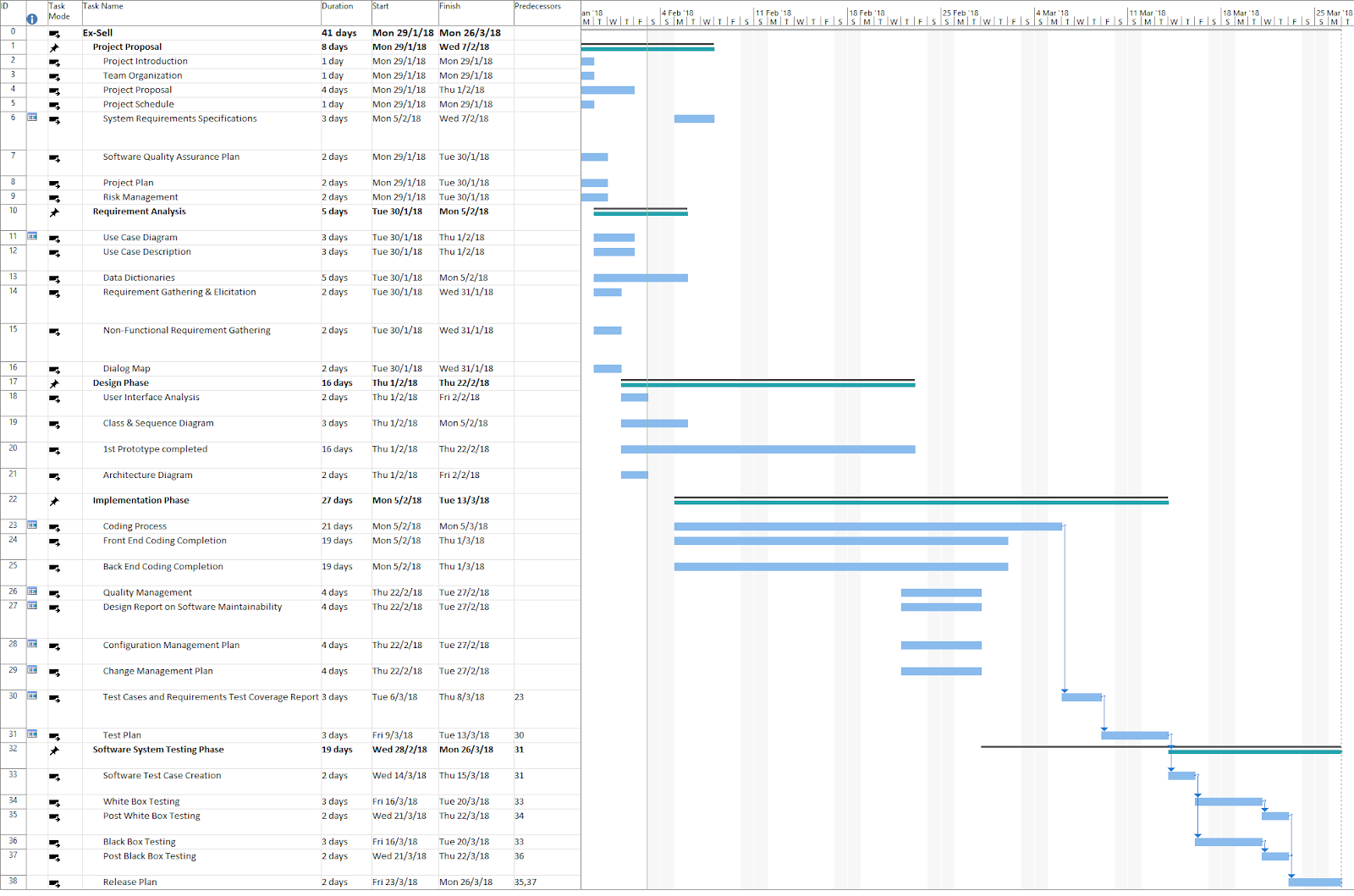
|  |  |
| --- | --- |
| **Process 08** | **Integrated Software Management** |
| Entry Criteria | Master list of project documents  Identify documentation tools  Establish documentation format standard |
| Process Description | This Process standardizes all management in writing so that all management can be measured and statistical data can be collected. |
| Exit Criteria | Well documented management  Verify the documentation tool, adopt more effective one if available |
| Begin            End | -Make a new list from the master list of project documents  -Determine documentation requirements  -Establish documentation standards  -Develop document  -Control document issue and consistently maintain it  -Verify the actual management  -Revise accordingly |

|  |  |
| --- | --- |
| **Process 09** | **Organization Process Definition** |
| Entry Criteria | None |
| Process Description | The software engineering tasks are defined, integrated, and consistently performed to produce the software.  Software work products are kept consistent with each other. |
| Exit Criteria | The software engineering tasks are defined, integrated, and consistently performed to produce the software.  Software work products are kept consistent with each other. |
| Begin      End | -Integrate software engineering methods and tools into the project's defined software process  -Develop and document software requirements by analyzing the allocated requirements  -Implement the software requirements and software design in code  -Test the software according to the project's defined software process  -Do a system and acceptance test  -Develop the documentation that will be used to operate and maintain the software  -Maintain consistency across software work products like SRS, software design, code, and test plans |

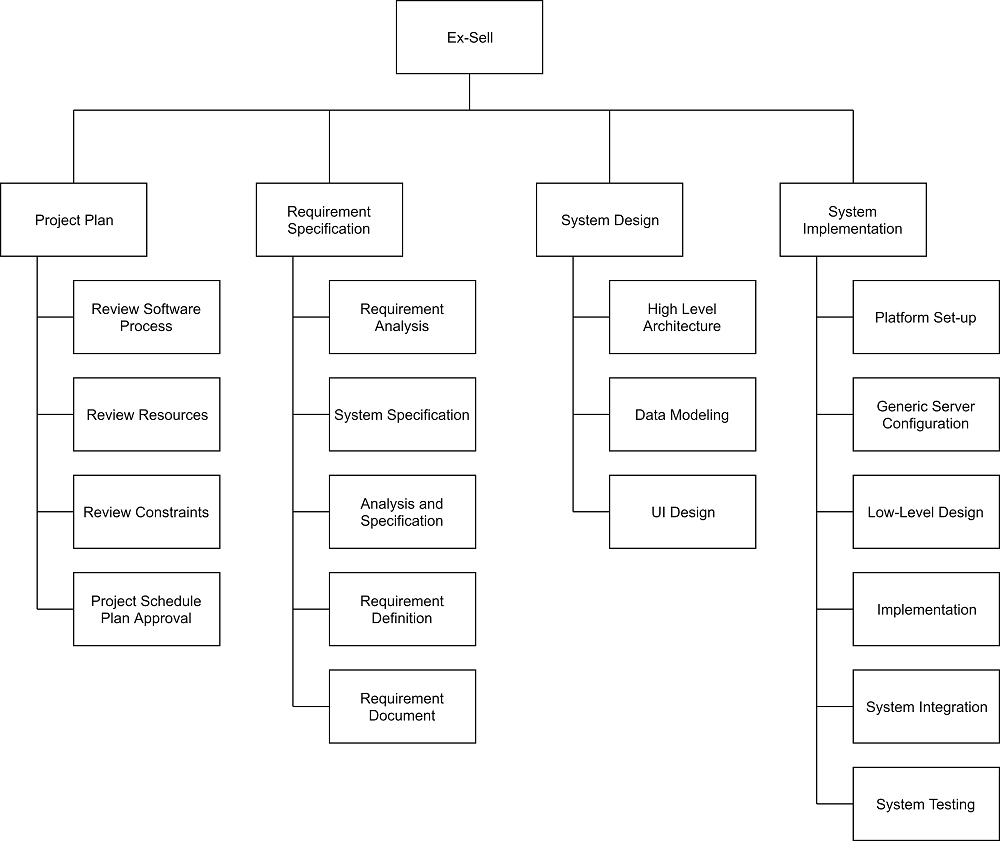
|  |  |
| --- | --- |
| **Process 10** | **Peer Review** |
| Entry Criteria | Plan the review and distribute the review materials. |
| Process Description | This process focuses on standardizing the review procedure and ensuring the quality of reviews. |
| Exit Criteria | Document the review results and take actions for review results |
| Begin            End | -Plan peer review, identify the review standard, decide on the method to be used for the review  -Establish completion criteria and review criteria  -Set the date and time to do the review  -Distribute the review materials  -Set the place to do the review  -Conduct the review  -Document the review result  -Take action accordingly |

# Schedule

## Activity Dependencies and Schedule



## Work Breakdown Structure



## Work Packages

The entire project work is broken down by the important phases of the software development life cycle.  They include the following:

* 1. Project Plan
  2. Requirement Specification
  3. User Interface
  4. Technical Architecture
  5. Data Modelling
  6. Coding & Unit Testing
  7. Integration & Quality Assurance

## Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

|  |  |  |  |
| --- | --- | --- | --- |
| Work Package # | Work Package Description | Duration | Dependencies |
| 1 | Project Plan | 5 days | -- |
| 2 | Requirement Specification | 7 days | -- |
| 3 | User Interface | 4 days | 2 |
| 4 | Technical Architecture | 21 days | 1, 2, 3 |
| 5 | Data Modelling | 4 days | 4 |
| 6 | Coding & Unit Testing | 19 days | 5 |
| 7 | Integration & Quality Assurance | 5 days | 6 |

## Work Package Details

Work packages are listed below.

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 1— Project Plan (1 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 5 Days |
| **Start Date** | 29 January 2018 |
| **Purpose** | To determine an overview of the project. |
| **Inputs** | None. |
| **Activities** | This work package includes providing an overview of the project, its objectives, and a set of proposed project deliverables throughout the development of the software cycle. |
| **Outputs** | A written document of the Project Plan. |

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 2 — Requirement Specification (2 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 7 Days |
| **Start Date** | 5 February 2018 |
| **Purpose** | To establish a common understanding between the customer and the software project team of the customers’ requirements to be addressed by the project. |
| **Inputs** | Customer’s Requirements |
| **Activities** | This work package includes identifying the customer, interview customer, write and inspect customer requirement specification in accordance to the requirements. |
| **Outputs** | A written document of the Requirement Specification. |

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 3 — User Interface (3 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 4 Days |
| **Start Date** | 01 February 2018 |
| **Purpose** | To create an effective communication medium between the system and the customer so that the system is friendly to the customer. |
| **Inputs** | User Information, Requirement Specification |
| **Activities** | This work package includes identifying interface actions, create screen layouts that forms the basis for an user interface prototype and display the result of request. |
| **Outputs** | A written document of the User Interface Design Report. |

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 4 — Technical Architecture (4 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 21 Days |
| **Start Date** | 05 Feburary 2018 |
| Purpose | To do the high level architecture design. |
| **Inputs** | Project Plan, Requirement Specification, User Interface |
| **Activities** | This work package includes identifying architecture of software system, components and relationships between them, deciding on the software and  hardware infrastructures, deciding the language used to implement the software, and addressing the design topics including maintainability, portability, and reusability. |
| **Outputs** | A written document of the High-Level Design and Architectural Specification. |

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 5 — Data Modelling (5 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 4 Days |
| **Start Date** | 01 March 2018 |
| **Purpose** | To build the project’s database |
| **Inputs** | Project Plan, Requirement Specification, User Interface, Technical Architecture |
| **Activities** | This work package includes analysing the data flow relationships, entity relationships. |
| **Outputs** | A written document of the Data Modelling. |

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 6 — Coding & Unit testing (6 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 19 Days |
| **Start Date** | 28 February 2018 |
| **Purpose** | To implement the system as per the requirements specification and other  associated documents. This work package includes such additional activities as  preliminary unit testing. |
| **Inputs** | Project Plan, Requirement Specification, User Interface, Technical Architecture, Data Modelling |
| **Activities** | This work package includes implementing the modules according to the design specifications noted in the Specification document, preparing unit test plans, and examining the different paths through the modules. |
| **Outputs** | A written document of the Unit Test Report, source code and header files. |

|  |  |
| --- | --- |
| **Project** | Ex-Sell |
| **Work Package** | 7 — Integration & Quality Assurance (7 of 7) |
| **Assigned To** | Yiu Hong Sum, Lim Kian Hock Bryan, Cheng Yang Zhen, Lim Yan Jun, Soong Jie Ming, Chen Zhenni, Chang Keat Lueng Aaron |
| **Effort** | 5 Days |
| **Start Date** | 06 March 2018 |
| **Purpose** | To perform black box testing, white box testing to check for logical errors, identify and fix syntactical errors produced during the implementation of the system and evaluating overall project performance to provide confidence that the project is satisfy with the relevant quality standards. |
| **Inputs** | Project Plan, Requirement Specification, User Interface, Technical Architecture, Coding & Unit testing, Data Modelling, Coding & Unit testing |
| **Activities** | This work package includes examining issues such as system performance and integrity and use metrics to develop strategies for improving the software process, and thus, improving the quality of the end product. |
| **Outputs** | A written document of the Test Report and Quality Assurance Report. |

# Project Estimates

## Code Size Estimation using Function Points

The unadjusted function point is calculated based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

### Unadjusted Function Points

Ex-Sell supports the following proposed functions:

**Seller:**

* Login
* Register as a new user
* List a product for sale
* List product details
* Communicate with Live Support Operator
* Receive message notification from buyer

**Buyer:**

* Login
* Register as a new user
* View list of products
* View product details
* Favourite/Bookmark product
* Communicate with Live Support Operator

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity.

|  |  |  |
| --- | --- | --- |
| Element | Complexity | Detail |
| Inputs | Low | Registration |
| Low | Login |
| Low | List Product for Sale |
| Medium | Message Live Support Operator |
| Medium | View Product for Sale |
| Medium | Accept/Reject Buyer/Seller |
| Outputs | High | Display Product Details |
| Low | Display User information |
| Low | Display Messages |
| Inquiries | High | Select Product |
| Medium | Selecting products according to various tags or item types |
| Low | Selecting product information |
| Medium | Selected seller/buyer details |
| Logical Files | High | Product Information |
| Medium | Users Account |
| Interfaces | High | MessageAPI |

Calculation of Unadjusted Function Points:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Characteristic** | **Low** |  | **Medium** |  | **High** |  |
| Inputs | 3 | × 3 | 3 | × 4 | 0 | × 6 |
| Outputs | 2 | × 4 | 0 | × 5 | 1 | × 7 |
| Inquiries | 1 | × 3 | 2 | × 4 | 1 | × 6 |
| Logical Files | 0 | × 7 | 1 | × 10 | 1 | × 15 |
| Interfaces | 0 | × 5 | 0 | × 7 | 1 | × 10 |
| Unadjusted FP | 20 |  | 30 |  | 38 |  |
| Total=L+M+H | 88 | | | | | |

### Adjusted Function Points

|  |
| --- |
| Scoring (0 – 5) |
| 0 = No influence |
| 1 = Insignificant influence |
| 2 = Moderate influence |
| 3 = Average influence |
| 4 = Significant influence |
| 5 = Strong influence |

|  |  |  |
| --- | --- | --- |
| **Influence Factors** | **Score** | **Detail** |
| Data Communications | 4 | Data and control information used in the Ex-Sell are sent or received over communication facilities. |
| Distributed Functions | 4 | Distributed processing and data transfer are online and in   both directions. |
| Performance | 3 | Each response of the user inputs for Ex-Sell is significantly reliable. |
| Heavily used | 0 | Ex-Sell on a moderately used web application. |
| Transaction rate | 3 | Daily transaction period is anticipated. |
| On-line data entry | 4 | More than 15% of transactions are interactive. |
| End-user efficiency | 3 | Ex-Sell should be reasonably user friendly. |
| On-line data update | 3 | Online update of internal logical files is included. |
| Complex processing | 0 | Ex-Sell will involve no complex processing or algorithms. |
| Reusability | 4 | Ex-Sell is specially packaged and documented to ease re-use. |
| Installation Ease | 4 | No special significant installation issues since Ex-Sell is a web-based application. |
| Operational Ease | 1 | Effective backup and recovery procedures were provided. |
| Multiple sites | 0 | Ex-Sell will be located in only one facility. |
| Facilitate change | 3 | Flexible query is provided that can handle complex requests such as changes in products. |
| Total score | 36 | |
| **Influence Multiplier**  = Total score × 0.01 + 0.65 = 36 × 0.01 + 0.65 = 1.01 | | |
| **Adjusted FP**  = Unadjusted FP × Influence Multiplier = 88 × 1.01 = 88.88 | | |

### Lines of Code

According to Capers Jones statistics, each Function Point requires 57 lines of code if the application is implemented using ASP.NET.

Therefore, the **Lines of Code** = 88.88 FP × 57 LOC/FP = **5066 LOC**

## Efforts and Duration Estimation

### Constructive Cost Model

The Constructive Cost Model (COCOMO) 81: Intermediate Model will be used to estimate project effort and duration. There are three modes which are organic, semidetached and embedded. Ex-Sell is a web application, which will have average complexity and fair flexibility. Therefore, Ex-Sell is an organic mode project. COCOMO cost estimation for organic mode projects follow this formula:

Effort = 3.2\*(KDSI) ^1.05\*EAF

Time = 2.5\*(Effort) ^0.38

where:

Effort = number of staff months

KDSI = number of lines of code for completed product. It is measured in thousands of lines of code

EAF = Effort Adjustment Factor

Time = total number of months

The Effort Adjustment Factor is the product of the 5 adjustment parameters. Each adjustment parameter is defined as very low, low, nominal, high, or very high.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **RATINGS** | | | | |
| **COST DRIVERS** | **Very Low** | **Low** | **Nominal** | **High** | **Very High** |
| Software Reliability | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 |
| Product Complexity | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 |
| Analyst Capability | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 |
| Applications Experience | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 |
| Programmer Capability | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 |

Calculation of Effort Adjustment Factor:

|  |  |
| --- | --- |
| **COST DRIVERS** |  |
| Software Reliability | 0.75 |
| Product Complexity | 1.30 |
| Analyst Capability | 1.19 |
| Applications Experience | 1.13 |
| Programmer Capability | 0.86 |
| **EAF Value** | 1.13 |

**Effort = 3.2\*(5.066) ^1.05\*1.13 = 19.9 person months**

**Time = 2.5\*(19.9) ^0.38 = 7.80 months**

### Top-Down Estimation

In this method, we use function points as the basis to calculate Effort, Duration, Team size, Compression rate and finally the schedule.

* Working days include 5 days in a week.
* Effort = Size / Production Rate = (5066 LOC) / (39 LOC/PD) = 130 PD
* Duration = 3 × (Effort) 1/3 = 3 × (130) 1/3 = 15.1 PD
* Initial schedule = 130 PD / 5 days a week = 26 week
* Team size = 130 PD / 15.1 PD = 8.6 person
* But since we have 7 persons, the Compression rate = 8.6 person / 7 person = 1.228
* Desired schedule = Initial schedule × Compression rate = 26 week × 1.228 = 32 weeks
* Total person-hours = 130 PD × 8 hours = 1040

### Bottom-Up Estimation

In this method, we base our estimation on the following factor:

* Balance of time between work and leisure

The calculation is as following:

* There are 7 group members; each spends 10 hours/week; then total hours per week is 70.
* The remaining time of project schedule is 2 weeks, so the total person-hours is 140.

### Distribution of Effort

The schedule is 3 months. The team size is 7. Working days include 5 days in a week.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1990’s Industry Data** | **Work Package** | **Distribution** | **Top-Down**  **Estimates** | **Bottom-Up Estimates** |
| Preliminary Design 18 % | Project Plan | 9% | 11.88 | 12.6 |
| Requirement Specification | 9% | 11.88 | 12.6 |
| Detailed Design 25 % | User Interface | 7% | 9.24 | 9.8 |
| Technical Architecture | 11% | 14.52 | 15.4 |
| Data Modelling | 7% | 9.24 | 9.8 |
| Code & Unit Testing 26 % | Code & Unit testing | 26% | 34.32 | 36.4 |
| Integration & Test 31 % | Integration & Quality Assurance | 31% | 40.92 | 43.4 |
|  | **Extrapolated total effort** |  | 132 | 140 |
|  | 2% for project management |  | 2.64 | 2.8 |
|  | 3% for contingency |  | 3.96 | 4.2 |
|  | **Total effort** |  | 138.6 | 147 |

These duration estimates are based on the assumption that each team member works an equal amount on any given work package.

## Cost Estimation

### Hardware:

**Developer workstations:**

|  |  |
| --- | --- |
| Digital Ocean Web Server | Total  $0.00 |
| 7 - HP Pte Ltd Workstation |
| Kohler Power Supply |
| Western Digital Network Attached Storage |

### Software:

**Software License Provided by Third Party:**

|  |  |
| --- | --- |
| Microsoft Office 2016 | $0.00 |
| Microsoft Project 2016 | $0.00 |
| MySQL Server | $0.00 |
| Microsoft Visual Studio 2017 Enterprise | $0.00 |

### Other Resources:

|  |  |
| --- | --- |
| **Staff**: |  |
| Project Manager | $5000 |
| System Developers | $9000 |
| Quality & Release Managers | $6000 |
|  | Total  $20,000 |
| **Stationary**: |  |
| Paper, photocopying and other miscellaneous cost | $90 |

**Total: $20,090**

The customer will supply the required hardware and software necessary. Zenith is not responsible for supplying said systems.

# Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

|  |  |
| --- | --- |
| **Project Deliverable** | **Estimated Deadline** |
| Project Proposal | 8 February 2018 |
| System Requirement Specifications | 26 February 2018 |
| Software Quality Assurance | 26 February 2018 |
| Project Plan | 21 March 2018 |
| Risk Management Plan | 21 March 2018 |
| Design Report on Software Maintainability | 4 April 2018 |
| Configuration Management Plan | 4 April  2018 |
| Change Management Plan | 4 April  2018 |
| Release Plan | 4 April  2018 |
| Test Plan | 18 April 2018 |
| Test Cases and Requirements Test Coverage Report | 18 April  2018 |
| CMMI Level 2 Definition | 18 April  2018 |

# Best Practice Checklist

|  |  |
| --- | --- |
| Project Deliverable | ✓ |
| All documentation must be in a standardized format. |  |
| Check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification. |  |
| Design the functionality that meets the customer requirements. |  |
| Utilizing proper UML diagrams to document use cases and analysis models. |  |
| The Project Manager must have good communication with his employees; require developers to make code available for review; review design for appropriateness. |  |
| All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes. |  |
| Be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance. |  |
| Conduct code reviews to find software defects. Plan and manage code reviews between team members. |  |
| Document the software defects, so development team can view and edit the document based on whether the issue has been solved. |  |
| Software testing will use both black box and white box testing. It will involve unit, functional, integration and acceptance testing. |  |

# Risk Analysis

The following risks have been identified for the Ex-sell project:

**Underestimation of funding due to changes in price**

* Impact severity: High
* Probability: 25%
* Impacts: Depending on the stage at which changes occur, we could seek other cheaper alternatives or request for additional funding.
* Risk Reduction: Always request quotations from multiple companies so that we could always have more than one options when the previously selected option is no longer available.

**Staff Unavailability**

* Impact Severity: High
* Probability: 20%
* Impacts: Delay in work completion and slowing down the progress of the project.
* Risk Reduction: Assign two or more people in a team that focus on a work assignment (e.g. Development Team for coding and Documentation team for work documentation purposes).

**Extra Time required to complete the work**

* Impact Severity: Moderate
* Probability: 10%
* Impacts: Reputation will be damaged, and the client’s business may be affected due to the delays.
* Risk Reduction: Plan and provide extra buffer time and shorten the time required to complete the task before the originally expected time.

**System Performance Issue**

* Impact Severity: High
* Probability: 5%
* Impacts: Clients’ business activities would be affected due to the performance issues encountered.
* Risk Reduction: Create a progressive benchmark to ensure the performance is consistent with the client’s expectations.

**Error during the presentation**

* Impact Severity: Medium
* Probability: 30%
* Impacts: Client’s time would be wasted due to the failure of execution by the software and client’s faith in our ability would decrease.
* Risk Reduction: Multiple tests should be conducted before the deadline and the development log has to be updated consistently.

**Requirement Changes by the Clients**

* Impact Severity: Medium
* Probability: 50%
* Impacts: Some of the resources used on previously wanted features would be wasted and extra cost may be incurred due to new changes.
* Risk Reduction:  If the change is minor, simple changes could be made accordingly. However, if the change is substantial, the development team will estimate and provide the cost incurred to the client for reconsideration if the change is worth the cost.

**Uncertainty on how to use the interface/software**

* Impact Severity: Extreme
* Probability: 5%
* Impacts: If the User Interface is not user-friendly, the client wouldn’t be able to use it properly and optimally.
* Risk Reduction: Create a guidebook or provide training for the user to familiarize themselves with the system.

**Server crash**

* Impact Severity: Extreme
* Probability: 5%
* Impacts: The website will not be able to run, customers will not be able to access the website, business will lose money and reputation will go down.
* Risk Reduction: Have a 24/7 technician assigned to fix the server within an hour.

**Database corrupted/deleted due to malware**

* Impact Severity: Extreme
* Probability: 5%
* Impacts: Frustrations with the user will increase and company reputation will go down.
* Risk Reduction: Create a backup of the database once every 12 hours to ensure an updated database will be available at any time.

# Quality Assurance

To assure high quality standards for the Ex-sell web application, two important aspects of the application must be tested. These two aspects are:

* **System Core Functionalities**
* **Usability**

System Core Functionalities refers to the basic functions that our web application is expected to perform. These were previously detailed in the Software Requirements Specifications and initial Project Proposal. Thus, it is paramount to thoroughly test and check for bugs and error in the application.

The usability of the web application is also an important aspect as it is expected that there will be a lot of human traffic perusing the web application. As such, it is therefore important to ensure that the web application is intuitive, easy to use and allows users to achieve their objectives fast. The level of usability will come from the feedback provided throughout the testing phase.

In order to test the web application, two different forms of testing phase will be conducted, namely:

* **White Box Test**
* **Black Box Test**

White box testing will be conducted by the Quality Assurance personnel in our team. This is because white box testing requires knowledge on the inner workings of the web application. This will help to ensure that the system performs as expected, and any errors or exceptions are handled with grace.

Black Box Testing will be conducted by both the Quality Assurance Team and by a few users from the client’s company with no knowledge of the inner workings of the web application. Black box testing will cover any loopholes that were missed during the white box testing phase due to the differences in perspective of the developer and the user.

With both White Box and Black Box testing, full coverage on the quality of the web application can be assured, allowing the team to deliver a product of satisfactory level to our clients

# Monitoring & Control

Risk Monitoring and Control is the process of identifying, analysing, and planning for newly identified risks, monitoring previously identified risks, and revaluating existing risks to verify the planned risks response strategies for their effectiveness.

To ensure successful delivery of the software which adheres to the project plan, monitoring and control is necessary during the software development phase. Some of the required control and monitor actions are listed below.

There will always be some risks that we overlooked while identifying, analysing and planning to handle the risks that we have anticipated as well as relapse of the previously identified risks.

These are some of the things or actions that we will do in order to **monitor and control** existing and potential risks that could appear during the development process:

* Have weekly meeting to update one another on the status of the software development processes
* Review and sort the priority of risks based on its level of complexity and probability of occurrence
* Perform a regular test run (White Box & Black Box test) whenever a new feature or change is made.
* Take note of the bugs encountered during the test run & inform the development team about it.
* Revise and rerun the software, to ensure that the matters were solved
* Ensure that no other issues arise during the process of fixing the latest bugs
* Ensure that all previously recorded risk are properly proper documented.
* Provide detailed description of action taken towards previously documented bugs.
* Create a default first-response actions list to be taken in the event that new risks appear, or old risks resurface