Project Plan

Accident Analysis Software

Nicolas Donaldson: S5256284

Table of Contents

[1.0 Introduction 3](#_Toc46748287)

[1.1 Problem Background 3](#_Toc46748288)

[1.2 Scope 3](#_Toc46748289)

[1.3 Document contents 3](#_Toc46748290)

[2.0 Work Breakdown Structure 4](#_Toc46748291)

[3.0 Activity Definition & Estimation 5](#_Toc46748292)

[4.0 Gantt Chart 6](#_Toc46748293)

# Introduction

## Background

The ability to correctly analyse data becomes increasingly difficult when no software is involved, and the data belongs to a large dataset. This effects the ability to make accurate decisions and has great potential to jeopardise the functionality of a business. This project aims to rectify the problem associated with large data analysis, by developing and implementing a software that can process and visualise large amounts of data. The client for this project is the Department of Transport for Victoria, and the software in question will analyse data from road accidents. The intended outcomes of the project involve meeting the required needs of the client and encapsulating those needs within a software that is fast, consistent, and reliable. This will be completed through intensive planning, time management, and thorough documentation.

## Scope

The deliverable of this project is a data analysis software; however, it is important to consider the tasks and processes required to successfully complete the project in the required time. In the context of time management, a work breakdown structure and Gantt chart are required. These essentially ensure the distribution of tasks, ensure that all tasks are considered, and that all tasks can be completed within the given timeframe. A system design document details the problems, vision, capabilities, requirements, and all the design components associated with the software. Python Programming files containing the code for the software must also be completed. This is an essential component of the project that dictates how the software works. It is critical to the project that this is working effectively. These files intend to implement the requirements of the client, which range from data retrieval to analysis and visualisation. In the final stages of the project, an executive summary, instruction manual/user guide, and software testing report are to be completed. The executive summary provides a review of the project, whilst the instruction manual/user guide provides instructions on the correct use of the software. The software report shall contain the results of the thorough testing that will be conducted, to ensure the software is working correctly. Furthermore, this project planning document includes background information surrounding the project, the work breakdown structure & Gantt chart, and the activity definition & estimation. This, along with the system design, work breakdown structure, and Gantt chart, will undergo systematic reviews and updates where necessary, to provide the project with as much advantage as humanly possible.

## Document contents

This document contains background information surrounding the project, the work breakdown structure, the activity definition & estimation, and a Gantt chart.

*Include some background information about the problem, the scope and what this document will contain.*

# Work Breakdown Structure

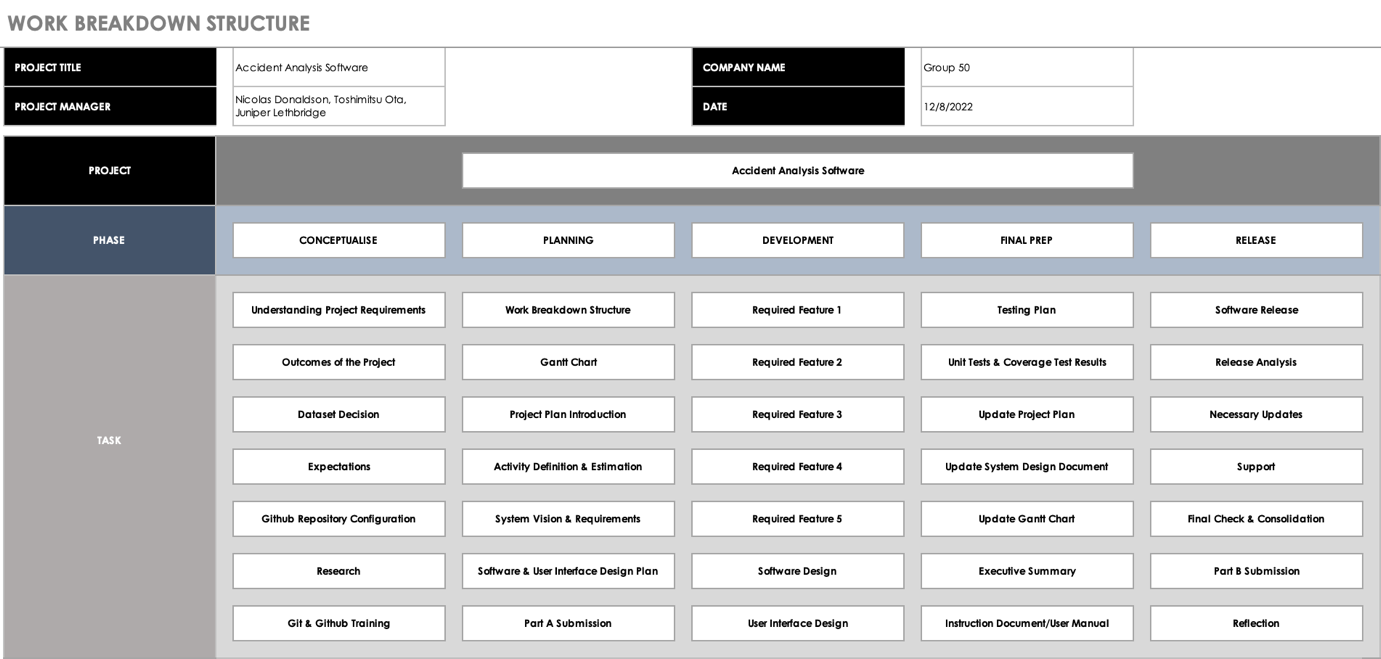


Image: Work Breakdown Structure for Project

*This section should include the work breakdown structure for the whole project. The elements from the WBS should be used to generate your activity definition and those activities should then be scheduled in the Gantt Chart. Remember to consider ALL project activities – anything you do or will need to do should be included in the WBS*

*WBS’s are usually presented as some kind of hierarchical diagram/chart etc. The details what is involved each work unit should be provided in section 3:* ***Activity Definition***

*You do NOT need to do a WBS Dictionary for this project – the activity definition (whilst slightly different) will suffice. The WBS is focussed on SCOPE. The Activity definition is focussed on TIME.*

# Activity Definition & Estimation

1. **Conceptualise**
   1. **Understanding Project Requirements:**

Understanding what is required of the project in terms of deliverables, outcomes, and expectations. What is the task and what exactly is to be submitted?

**Duration**: 48 Hours

* 1. **Outcomes of the Project:**

What would we like to have achieved by the end of the project? What skills are expected to be enhanced and what additional skills would be beneficial to implement into the project?

**Duration**: 48 Hours

* 1. **Dataset Decision:**

What Dataset will the project proceed with? The best dataset is the one that most aligns with our interests and everyone’s strengths as a programmer. An analysis into each optional dataset and its requirements must be conducted.

**Duration**: 48 Hours

* 1. **Expectations:**

What are the expectations of each project team member? What is the expected method of communication, submission date, etc.

**Duration**: 24 Hours

* 1. **Github Repository Configuration:**

Ensure that a project team member clones the official repository and adds the remaining project team members to the repository. Solve any issues associated with team members experiencing difficulties cloning the repository into their own local directory.

**Duration**: 48 Hours

* 1. **Research:**

Conduct individual research on methods used for effective software, techniques, tips, etc. Are there examples of software that resemble the software required for this project?

**Duration**: 1 Week

* 1. **Git & Github Training:**

For project team members with no Git or Github experience, use additional resources to discover the benefits associated with version control and git. Websites such as Codeacedemy provide demonstrations on how Github works and Git commands.

**Duration**: 1 Week

**Total Duration**: 1 Week

1. **Planning**
   1. **Work Breakdown Structure:**

Allocating the scope of the project in sequential order, using one of the many variations of a Work Breakdown Structure. Categorise each task into phases so that the scope of the project is more defined.

**Duration**: 48 Hours

* 1. **Gantt chart:**

Convert the Work Breakdown Structure and the Activity Definition & Estimation into a Gantt chart. This will visualise how long a task shall take, and what tasks need to be completed by a certain time.

**Duration**: 48 Hours

* 1. **Project Plan Introduction:**

Complete the Project Plan Introduction. This includes the problem background, which discusses additional background information surrounding the causes of the project and associated info. The scope, which outlines all the deliverables of the project, and finally, the contents of the project document.

**Duration**: 48 Hours

* 1. **Activity Definition & Estimation:**

Provide a definition for each activity outline in the Work Breakdown Structure. Additionally, provide an estimated duration for the task.

**Duration**: 72 Hours

* 1. **System Vision & Requirements:**

Provide background information surrounding the client and the need for the project deliverables. Describe an overview of the system, and potential benefits that the system may offer. Detail how the user is to interact with the software along with the functionality the software will provide. Provide use cases to describe how the software may be used

**Duration**: 5 Days

* 1. **Software and User Interface Design Plan:**

Provide a diagram to illustrate how the software might work. Provide a preliminary list of all functions in the software. Provide a list of all data structures in the software (e.g., linked lists, trees, arrays etc) or eternal data sources. Provide Pseudocode for all non-standard / non-trivial algorithms that operate on data structures. Finally, describe the tools used for the design stage and complete the Structural and Visual Design Plan. **Duration**: 1 Week

* 1. **Part A Submission:**

Ensure all the requirements for Part A have been fulfilled, and after a final project team discussion, have a project team member submit all required documents before the due date.

**Duration**: 48 hours

1. **Development** 
   1. **Required Feature 1:**

Write Python code that displays information for all accidents that happened, within a user-selected period.

**Duration**: 2 Weeks

* 1. **Required Feature 2:**

Write Python code that produces a chart showing the number of each accident in each hour of the day (on average), for a user-selected period.

**Duration**: 2 Weeks

* 1. **Required Feature 3:**

Write Python code that retrieves all accidents caused by an accident type that contains a keyword (user entered), e.g., collision, pedestrian, for a user-selected period.

**Duration**: 2 Weeks

* 1. **Required Feature 4:**

Write Python code that allows the user to analyse the impact of alcohol in accidents – ie: trends over time, accident types involving alcohol, etc.

**Duration**: 2 Weeks

* 1. **Required Feature 5:**

Write Python code for one other analysis or insight tool.

**Duration**: 2 Weeks

* 1. **Software Design:**

Using the Software Design document and the python files, implement and design the Software

**Duration**: 2 Weeks

* 1. **User Interface Design:**

Using the User Interface Design Plan, Implement and Design the User Interface

**Duration**: 2 Weeks

1. **Final Prep** 
   1. **Testing Plan:**

Create a brief testing plan describing how the Software Testing will be documented

**Duration**: 48 Hours

* 1. **Unit Tests & Coverage Test Results:**

Conduct Unit Tests & Coverage Tests outlined within the Testing plan. Document findings and make necessary adjusts to the software.

**Duration**: 72 Hours

* 1. **Update Project Plan:**

Update the Project Plan wherever necessary. Adjust the Work Breakdown Structure and the Activity Definition & Estimation.

**Duration**: 48 Hours

* 1. **Update System Design Document:**

Update the System Design Document so that it is up to date with the current design of the system. Ensure that the documentation surrounding the functionality of the software is up to date with the functionality of the current software.

**Duration**: 72 Hours

* 1. **Update Gantt Chart:**

Update the Gantt Chart to represent the current state of tasks and completion status. Recognise the similarities and differences between the old Gantt chart and the updated chart.

**Duration**: 24 Hours

* 1. **Executive Summary:**

Write a 100–150-word abstract describing the findings of the project.

**Duration**: 24 Hours

* 1. **Instruction Booklet/User Manual:**

Write an Instruction Booklet/User Manual describing in a clear and concise manner, how to correctly use and navigate the software.

**Duration**: 48 Hours

*From your WBS, define the activities required for your project. You will revise this document and add more detail for part B as you discover more about the project.*

*Each activity should be clearly identified by a number and should match up to your Gantt chart. You should provide some estimations for the time you think each activity will take. This should make it easy to prepare your Gantt chart.*

# Gantt Chart

*This section should contain your Gantt chart. The items in the Gantt chart should match the activity definition from section 3. You should also submit your Gantt chart file separately.*