Queensland University of Technology

**Name – Liam Allan**

**Student Number** – 10220607

**Name – Patrick Gallagher**

**Student Number** – 10265147

**Name – Ben Bloss**

**Student Number** – 9815848

**Due date** – 02/06/19

Cab302 Assessment Item 2

Documentation

# Statement of Completion

The application has been completed to a reasonable degree, with the implementation of a proper amount of additional functionality. The necessary VEC file importing and exporting tools have been implemented, with all the shapes necessary to create the files requested. The ability to create shapes with a fill colour has been implemented for all of the shapes created in the program. Rectangles, Ellipses, Lines, Plots and Polygons are all supported, and are all accessible through the Vector Design tool. An Undo option for the drawing tool has also been implemented to erase the latest drawing operation. Additionally, the additional functionality of a Zoom tool and multi-image support have been implemented.

# Statement of Contribution

* Patrick Gallagher – GUI, zoom, multi-image support, Report
* Liam Allan – Testing, Canvas, Report
* Ben Bloss – Drawing commands, VecCommand structure, exporting and importing, Report

# Statement of Agile Software Development Processes Implementation

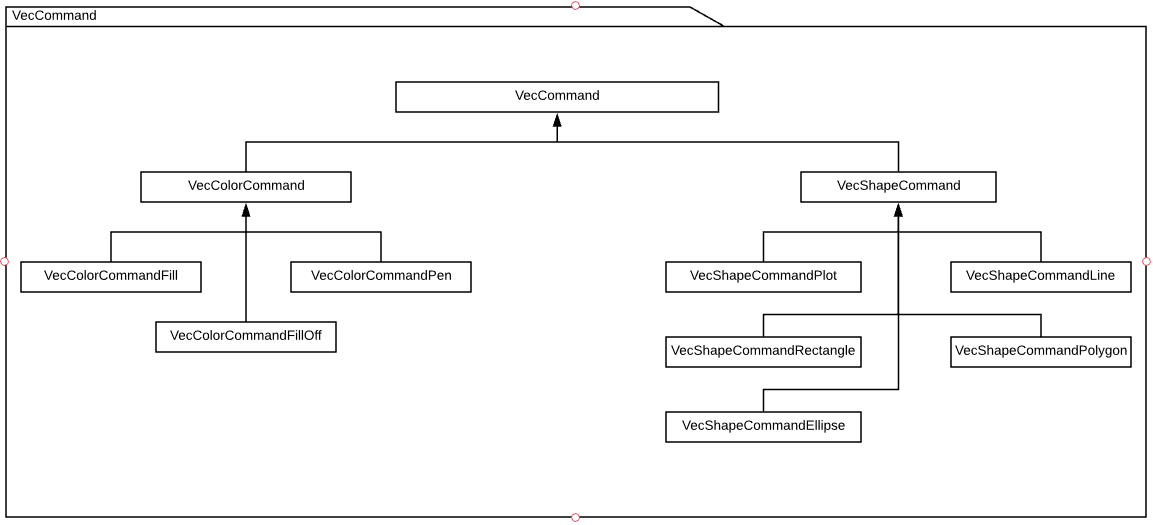
* Pre-Project
* Feasibility Study
* Functional Model Iteration
* Design and Build Iteration
* Implementation
* Post-Project

# Documentation of Software Architecture

Within the project there are multiple different classes. One such class is the GuiComponent class, which is an extension of JFrame and the main class of the project. From here all other classes are generated and applied. All GUI components such as buttons, tabbed panes, and scrollable panes are also generated and applied to the frame. From this class, the ColourChooser class. This class is also an extension of JFrame and is generated when the colour or fillcolour buttons are pressed. It’s main purpose is an implementation of JColourChooser to select and change colours on the GUI frame.Also implemented in the GuiComponent class is the VecFile class. This class is and extension of JPanel and is responsible for calling all draw commands and adding and running those commands in the command stack. These commands are as described below.

For this assignment it was necessary to create several classes to represent commands used by a vector graphics printer. These commands were all required to behaviour in context specific ways while exposing a unified interface. For this purpose, we create the VecCommand class to capture at an abstract level the behaviour of all VecCommands, specifically all commands where required to be executable, each providing a method for changing the application state, and to be printable, returning a string representation of the command for writing to a “.vec” file to be read and executed by a printer. A VecCommandEnum was created to easily differentiate the types of commands and a VecCommandFactory was created to ensure the correct object was create based on a supplied type.

A Subclass of VecCommand is VecColourCommand and VecShapeCommand. These two classes are for colour commands and shape commands respectively. VecColourCommand contains the subclasses VecColourCommandPen, Fill, and FillOf while VecShapeCommand contains the subclasses VecShapeCommandRectangle, Plot, Line, Ellipse, and Polygon.



Example of the VecCommand structure

# Documentation of Advanced Object-Oriented Programming Principles

Abstraction

To ensure execution of drawing commands is smooth, every command must be of the same type. To ensure this, an abstract VecCommand class was made. By using this class, every command can be assured to be of type VecCommand. The VecColourCommand and VecShapeCommand are also defined as abstract to separate shape commands from colour commands.

Encapsulation

Encapsulation is implemented throughout the VecFile class as private variables. This allows multiple VecFiles to exist at the same time. The VecCommands attached to each VecFile are also encapsulated to allow multiple of the same command to occur without issue and be specific to its related VecFile.

Inheritance

Every shape contains and array of points. As such by inheriting the points and printToFile from VecShapeCommand, every shape will be easily executed and printed to a file. The same occurs within VecColourCommand with colour and printToFile being inherited by VecColourCommand Pen and Fill to ensure the commands are written to file with the correct format.

Polymorphism

To enable drawing polymorphism is applied to the paintComponent command of JPanel, making what appears on the frame editable. Polymorphism also occurs for each of the events as the observers are overloaded to allow specific commands to execute when an event occurs.

# Documentation of Software Use

Guide to how to use software, with screenshots.