**7/2/24**

TDD - Test Driven Development:

* Consistently performing tests whilst designing the code

Agile

* Eliminate the need for elaborate requirements and intricate design. From this provides collaboration and teamwork
* Scrum: able to delegate work into goals to the team to be completed in time iterations called sprints.

**9/2/24**

Requirements:

* What the client requires for the project

Specifications:

* Categorised into functional and non-functional specifications
* More specific and detailed information.

Functional Specifications:

* Taking in data (input) and transforming it into information (output)
* "The system shall calculate and store the average of all raw assessment marks for every student studying each course." Here, the raw assessment marks, tasks, and associated courses act as input data, undergoing processing to yield a set of averages for each course.
* Functional specs describe the processes the software must perform

Non-functional Specifications:

* Does not involve transforming inputs into outputs

**12/2/24**

Level 0 diagrams

* Context diagrams illustrate how the system interacts with its environment, specifying the data and information flowing through each interface and its direction. The data entering the system must be sufficient to generate all the information leaving the system.

Data Flow Diagrams:

* how data moves through a system, without indicating the timing of events. DFDs focus on data changes between processes, not step-by-step process logic
* External entities (users, other organisations, and systems) are shown as labelled squares. Circles represent processes and have labels indicating their actions. Labelled data flow arrows represent data movement between processes. Finally, an open rectangle is used for data stores, where data is stored before and after processing

**Fun With Numbers Software Code**

**14/2/24**

Language - Python

**Home Screen Information:**

Welcome to Fun With Numbers

Choose from the menu below:

(A) Check number features

(B) Plot numbers

(C) Check overall stats

(X) Save and exit

Choice:

User Input: (a, b, c or x) opens new module

**Option A - Check Number Features**

* Option A, allows you to check the features of most numbers as long as they aren’t too big. Large numbers will take a while or not work at all. The max number of digits that the software can process is 8 digits.
* The features of 12 are...
* Positive
* Even
* Factors are 1 2 3 4 6 12
* Is not a prime number

**Option B - Plot Numbers**

* Allows you to plot coordinates on a number plane to create lines, curves etc. Any numbers that exceed the numbers on the plane, it keeps prompting for numbers within the numbers shown.

**Option C - Check Overall Stats**

* By inputting ‘C’ in the console, it prints a new array of text expressing the users’ input statistics in the software. At the bottom of the printed text, there is an option to return to the landing page by pressing enter. You must have run Option A at least once prior to opening Option C, otherwise the software will crash.

Here are your statistics of overall use:

Numbers entered: 3

Total of numbers: 8489400

Average of numbers: 2829800.0

Smallest number entered: 0

Largest number entered: 8489324

Coordinates plotted: 9

Press enter to return to the menu…

Option X - Save and Exit

* Closes screen

Notes -

* On the plotting b option, even if you press a number instead of y or n (yes or no), it still goes to the next function.
* When a number above the max (x=38,y=12) is inputted into the system, rather than displaying an error message, it continues to ask for the x and y axis.
* Should be able to delete coordinates
* Proper error messages instead of just the software crashing without sending a message to the user.
* Plotters places the ‘x’ between rows
* ‘0’ should be valid in the table