

# Advanced Processes:

Looking at the Merit criteria for this standard:

- effectively using project management and version control tools and techniques to manage the development of a digital technologies outcome
- trialling multiple components and/or techniques and selecting those which are most suitable
- using information appropriately from testing and trialling to improve the functionality of the digital technologies outcome
- addressing relevant implications.

The project management technique we will use will be the Agile technique. Our version control is Git hub (watch videos and make the repository in the video)

Let's consider a brief:

Amy has an large bowl of fruit at home.

It currently contains 5 Apples , 7 Pears , 2 Mangoes, 9 KiwiFruit and 3 Peaches.

Amy would like a program designed so that she monitor the consumption of her fruit.

We are assuming that we will make this in Python and that it will be a console program.

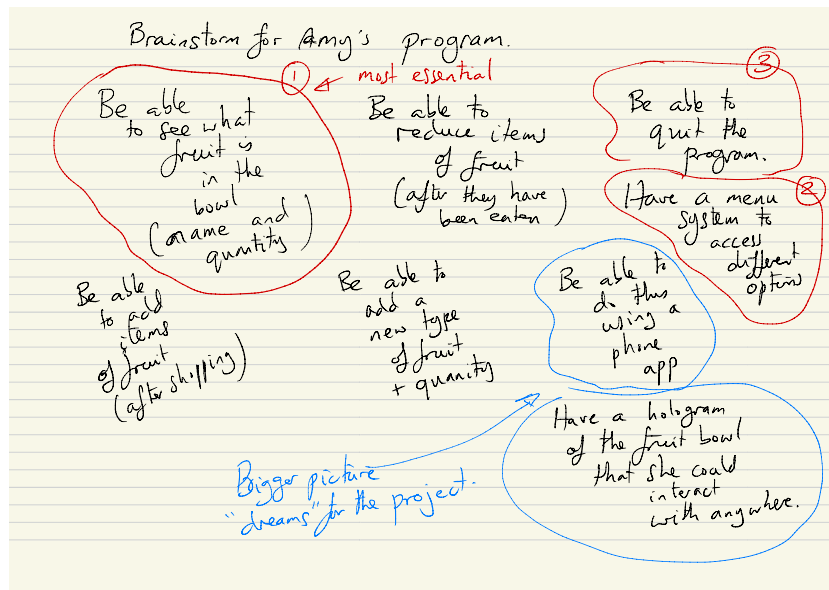
However, we can think a little more widely before we start.

Using a brainstorming process we can think about all the things we would like the program to do.

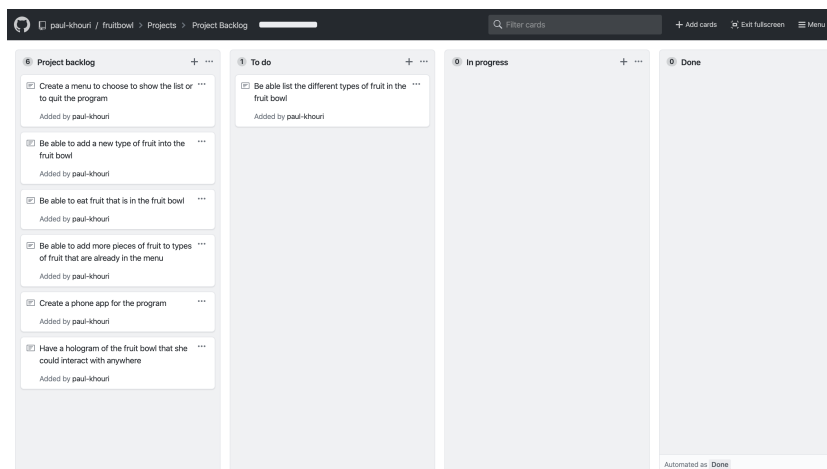
We then break these ideas up into pieces and place them in a **Project Backlog**

# 1 Project Backlog

- The project backlog is a list of all the things the program might do.
- It can (theoretically) be infinite in size
- The project backlog can be updated regularly during a project
- The project backlog should also contain the **simplest possible** ideas for the project
- The project backlog should be ordered by **priorities**, most essential at the top, working down



The project backlog gets set up as a trello/kanban whatever type board.



## 2 The Sprint

- A sprint is a planned passage of work that is completed in a short timeframe and leads to a tangible outcome.
- It “adds value” to the project.
- Ideally we should take the highest priority item in the project backlog (as our sprint) and plan to complete it in a short period of time (2 or 3 days).
- Ideally the project backlog is being updated to break things down into achievable pieces.

### 2.1 Sprint 1

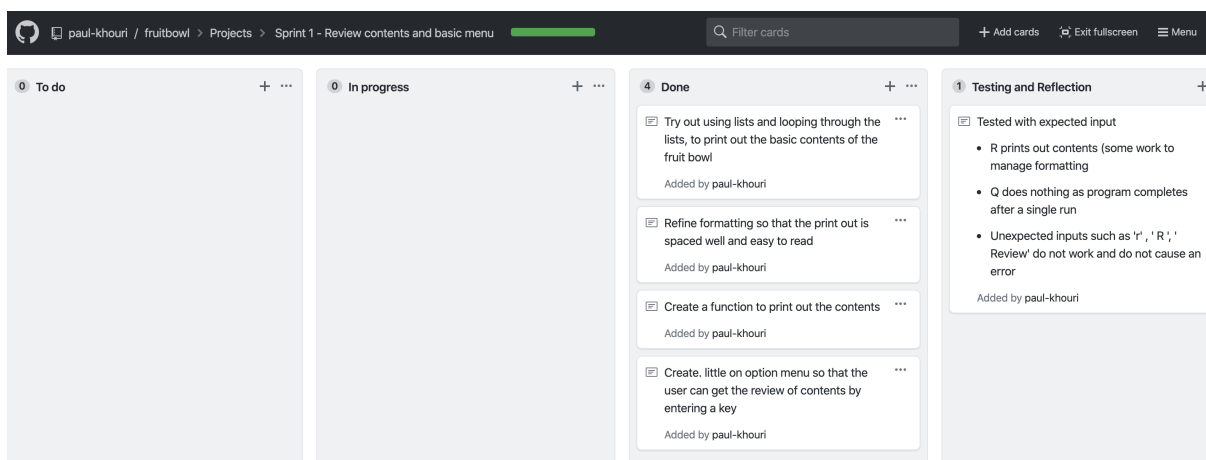
Taking the aim “to list the contents of the fruit bowl” , we plan the sprint , then do it.

Every sprint should follow the requirments below

Not every sprint needs to be documented in meticulous detail, but **key** sprints should be.

You want to

- Aim
- Trello or Kanban Board
- Natural language sketch planning
- Standup comments
- Testing
- Sprint Review



```

fruit_list = [ ["Apples", 5], ["Pears", 7], ["Mangoes", 2],
               ["Kiwi Fruit", 9], ["Peaches", 3] ] # 2 dimensional list.

def print_contents(l): # takes 2
    for x in l: # d list and print out (loop)
        out_string = "{} {}".format(l[0], l[1])
        print(out_string)

# menu. (first idea)

print("R : Review contents")
print("Q : Quit")
user_option = input("Please choose your option")
if user_option == "R":
    print_contents(fruit_list)

```

```

1 /usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/
  FruitBowlGitHub/fruit_sprint_1.py
2 R : Review Contents
3 Q : Quit
4 Please choose your options: -> R
5 Apples          -          5
6 Pears           -          7
7 Mangoes         -          2
8 Kiwi Fruit      -          9
9 Peaches         -          3
10
11 Process finished with exit code 0

```

Listing 1: Successful Test

## 2.2 After the sprint

- The project backlog should be reviewed and updated.
- Files are uploaded to github and commits periodically documented

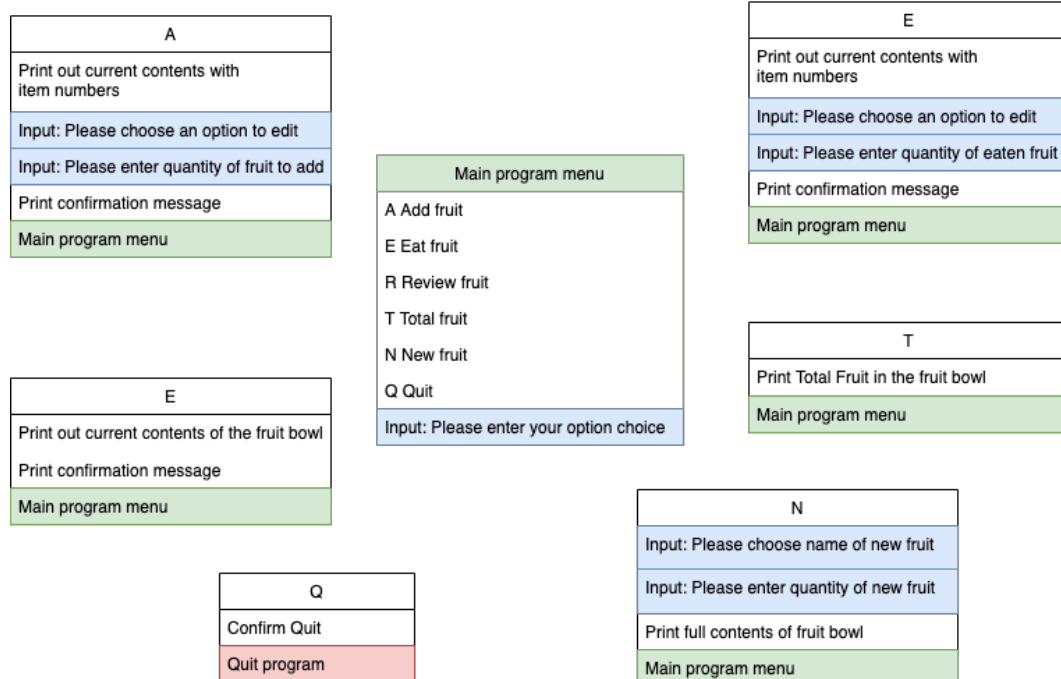
## 2.3 Running the project overall

We follow a cycle of sprint after sprint, building up the sophistication of the program.

In the case of our fruit bowl the sprints would be somthings like:

- Allow the user to add an amount of a fruit (i.e choose a fruit and add a certain amount)
- Allow the user to remove an amount of a fruit (i.e choose a fruit and remove a certain amount)
- Allow the user to add a new type of fruit
- Allow the user to remove a type of fruit.
- At some stage improve the menu structures.
- At some stage review the structure of the program and re organise if necessary
- Manage user interaction so that inputs are validated and assist the user

**Program Plan - The program runs using a base menu and calls specific functions for different actions**



Each sprint could, in theory, been seen as the process of creating one of these functions.

Validation functions would also have to be built.

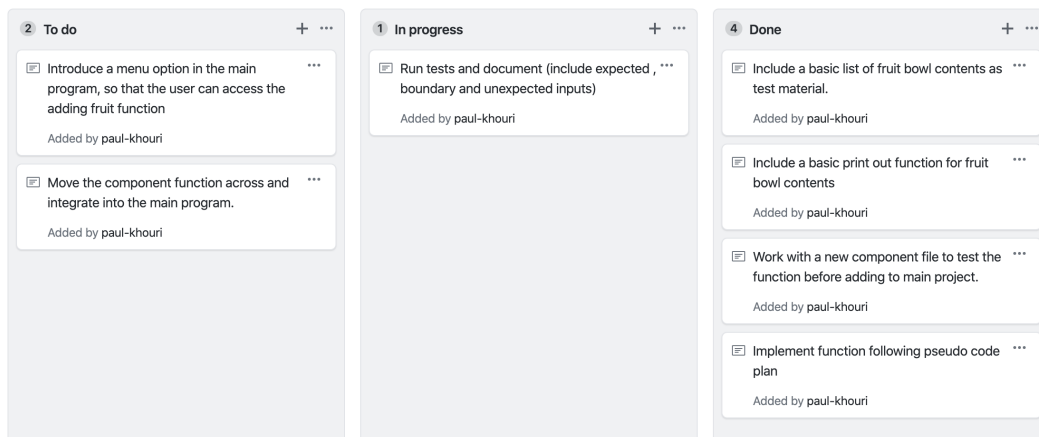
## 2.4 A sprint for adding fruit to the fruit bowl

Aim: For the user to be able to add fruit to the fruit bowl. The component should print out the current contents of the bowl, allow the user to choose which fruit they want to add to, add some fruit and then get confirmation.

Once the function is working properly, move it into the main version and connect it with the main menu through a function call.

```
1 create list fruit_bowl= [ ["apples", 2], ["oranges", 3],... ]
2
3 define function add_fruit(L) takes m x [string,integer] list
4 print out list with item numbers (beautifully formatted :) )
5 input: item to edit?
6 input: how many apples would you like to add
7 add apples to the value
8 return None
9
10 Print out confirmation list
```

Listing 2: Code outline for add fruit function



### 2.4.1 Testing

Basic principles of testing:

- Testing exists to find errors in a program (not to avoid finding errors).
- Tests should consciously seek to find errors by exploring different input patterns.
- We have a general principle of testing for **expected**, **boundary** and **unexpected** inputs
- Complete the tests you have planned before trying to fix errors.
- Not all tests need to be documented in detail, but you must demonstrate solid evidence of good testing processes in your overall planning work.

```
adding_fruit_component_test x
/usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py
Item # Fruit Quantity
0 Apples 5
1 Pears 7
2 Mangoes 2
3 Kiwi Fruit 9
4 Peaches 3
Please choose an item number to add fruit to? 3
How many Kiwi Fruit would you like to add? .5
Traceback (most recent call last):
  File "/Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py", line 30, in <module>
    add_fruit(amy_fruit_bowl)
  File "/Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py", line 24, in add_fruit
    user_number = int(input(message))
ValueError: invalid literal for int() with base 10: '.5'

Process finished with exit code 1
```

Test looking at entering .5 rather than 5 for adding to the Kiwi Fruit. The integer cast leads to a program crash. This needs fixing.

```
adding_fruit_component_test x
/usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py
Item # Fruit Quantity
0 Apples 5
1 Pears 7
2 Mangoes 2
3 Kiwi Fruit 9
4 Peaches 3
Please choose an item number to add fruit to? 5
Traceback (most recent call last):
  File "/Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py", line 30, in <module>
    add_fruit(amy_fruit_bowl)
  File "/Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py", line 23, in add_fruit
    message = "How many {} would you like to add? ".format(L[user_choice][0])
IndexError: list index out of range

Process finished with exit code 1
```

Looking at entering an item value that is not is on the menu list. This is calling an index that doesn't exit on the fruit bowl list. This leads to a "index out of range" crash.

```
adding_fruit_component_test x
/usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/pizzas/adding_fruit_component_test.py
Item # Fruit Quantity
0 Apples 5
1 Pears 7
2 Mangoes 2
3 Kiwi Fruit 9
4 Peaches 3
Please choose an item number to add fruit to? 1
How many Pears would you like to add? 4
You now have 11 Pears in the fruit bowl

Process finished with exit code 0
```

Expected input test, works correctly.

### **2.4.2 Sprint Review**

The problems with program crashes with unexpected input and starting to annoying.

Solution: move validation (which helps the user by preventing errors, or giving feedback on how to detect errors) up the project backlog and make this the next sprint.

## **3 Component Testing**