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)
Relevant implications will be similar to last year

1 Agile

Much of this material has come from Atlassian https://www.atlassian.com/agile

Agile is fundamentally team based, but you can apply its thinking and methods to your individual projects.

Agile originally came from a "manifesto" of software developers in 2001.
At that stage software development involved:

- complete planning of the project before starting to make anything,
- long deadlines (six months or more),
- generally only seeing the clien at the beginning and at the end of a project,
- members of the team had highly specified rolls (" I only do databases").

The problem with this (Waterfall) approach is that building digital products is still a new area

- So that new needs or technical approaches can emerge at any time and people often learn as they go.
- By the time the client gets the product, it is out of date and is inflexible to new requirements or changes.

- The product is not being tested in the marketpalce until it is finished
- During the project, some team members will have lots to do whilst others are doing little because the needs are outside of their job description

Some of the fundemental ideas behind agile are

- A team is a **TEAM**. So members should be aware of what others are doing, may fast track actions so that another part of the team is not **blocked** and may take on different non-specialist rolls at different times.
- The emphasis is on working software (applications) that can be launched quickly and developed upon responsively to the market place and clients needs (have you ever had to download an update??)
- Being responsive to change (have a plan but change the plan if it is the right thing to do)

This is the general idea of Agile and its actual implementation is done using methodologies such as **Scrum** and **Kanban**.
We will use Scrum.

1.1 Scrum

All agile methods follow similar patterns.

- Break the project down into small parts (project backlog)
- Prioritize
- Plan and implement a short passage of work that will have a tangible outcome (sprint)
- Have regular, quick meetings or reviews "standups" to identify
  - What have I done (yesterday)?
  - What do I intend to do next (today)?
  - What is blocking me?
- Review completed passage of work (sprint review)
- Repeat the cycle (iterate)

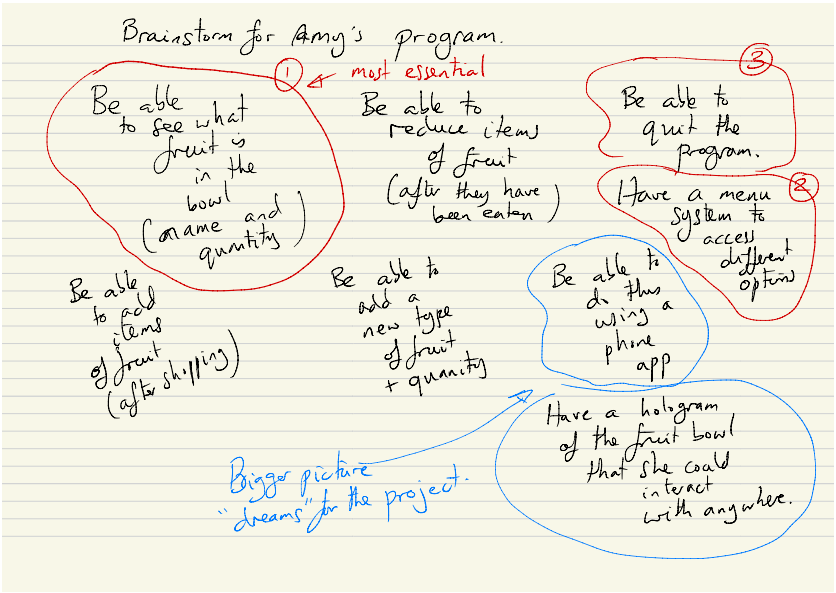
2 Let's consider a brief

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.

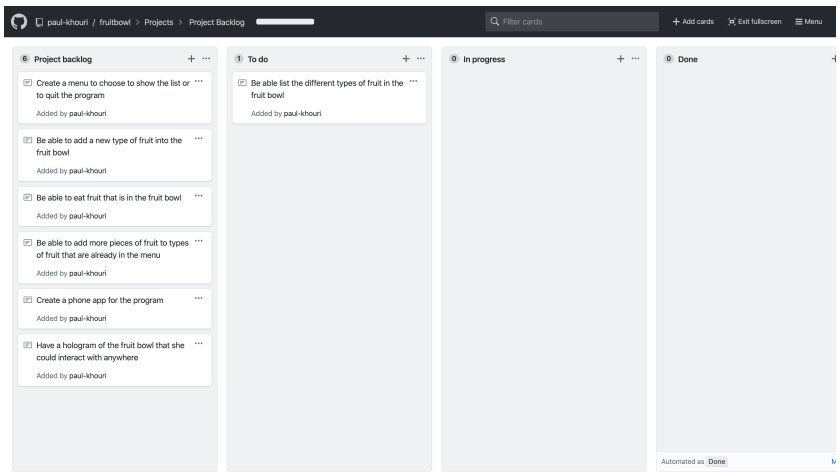
We are assuming that we will make this in Python and that it will be a console program.
However, we can think a little more widely before we start.
Using a brainstorming process we can think about all the things we would like the program to do.
We then break these ideas up into pieces and place them in a **Project Backlog**

2.1 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 whetever type board.



2.2 The Sprint

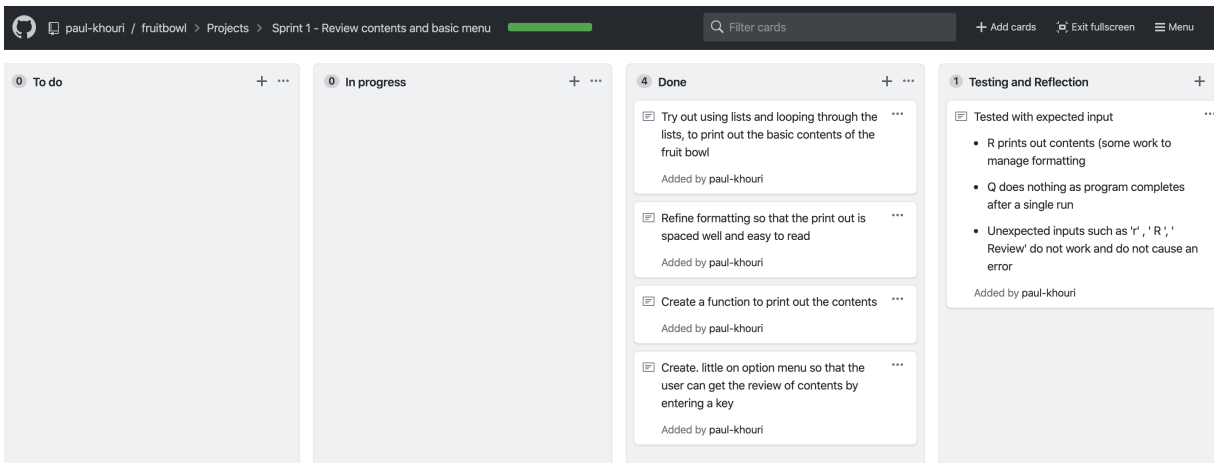
- A sprint is a planned passage of work that is completed in a short timeframe and leads to a tangible outcome.
- It "adds value" to the project.
- Ideally we should take the highest priority item in the project backlog (as our sprint) and plan to complete it in a short period of time (2 or 3 days).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

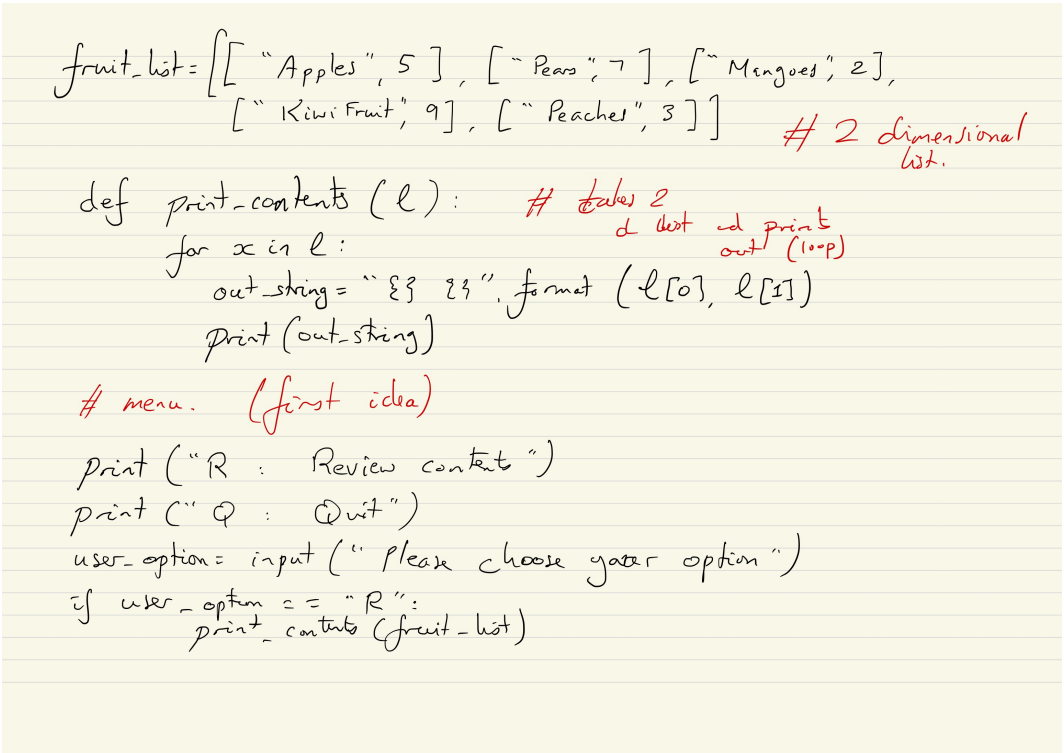
2.3 Sprint 1

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 include:

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






```
/usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/
    FruitBowlGitHub/fruit_sprint_1.py
R : Review Contents
Q : Quit
Please choose your options: ->   R
Apples          -          5
Pears           -          7
Mangoes         -          2
Kiwi Fruit      -          9
Peaches         -          3

Process finished with exit code 0
```

Listing 1: Successful Test

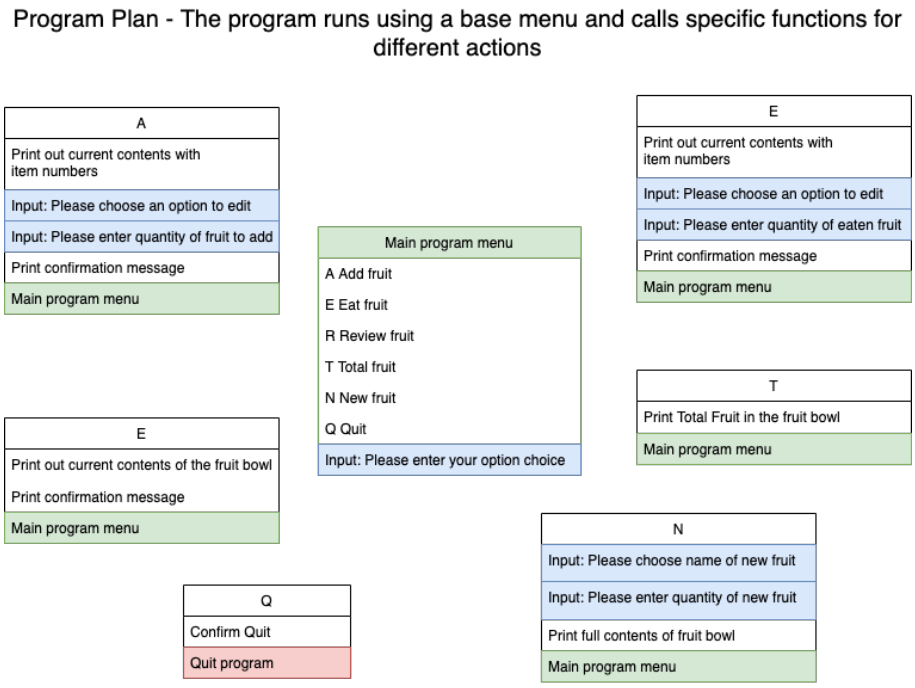
2.4 After the sprint (sprint review)

- The project backlog should be reviewed and updated.
- The top of the backlog is taken for the next sprint.
- Files are uploaded to github and commits periodically documented

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



Each sprint could, in theory, been seen as the process of creating one of these functions.
Validation functions would also have to be built.

3.1 A sprint for adding fruit to the fruit bowl

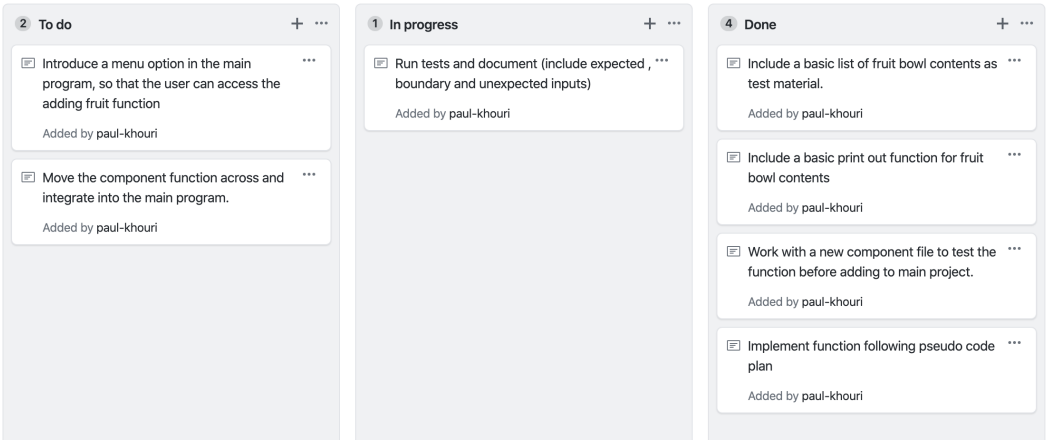
Aim: For the user to be able to add fruit to the fruit bowl. The component should print out the current contents of the bowl, allow the user to choose which fruit they want to add to, add some fruit and then get confirmation.
Once the function is working properly, move it into the main version and connect it with the main menu through a function call.

```
create list fruit_bowl= [ ["apples, 2], ["oranges , 3 ],... ]

define function add_fruit(L) takes m x [string,integer] list
print out list with item numbers (beautifully formatted :) )
input: item to edit?
input: how many apples would you like to add
add apples to the value
return None

Print out confirmation list
```

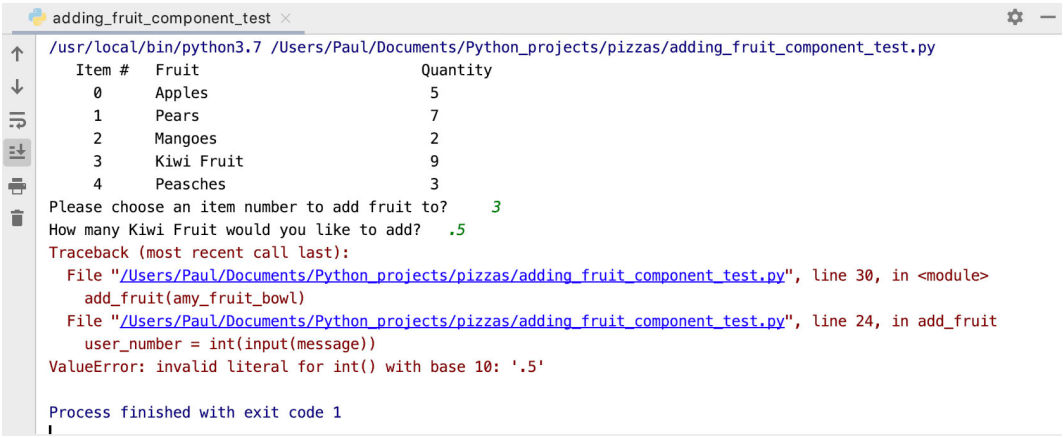
Listing 2: Code outline for add fruit function



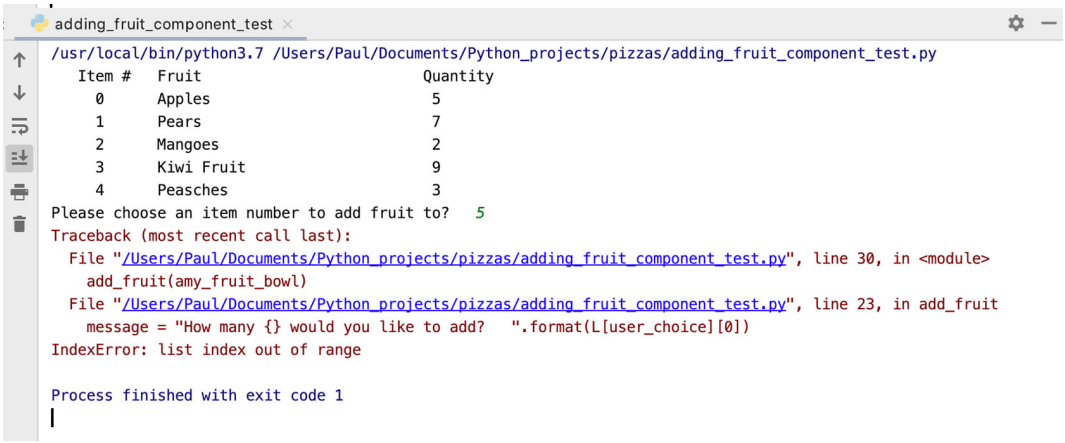
3.1.1 Testing

Basic principles of testing:

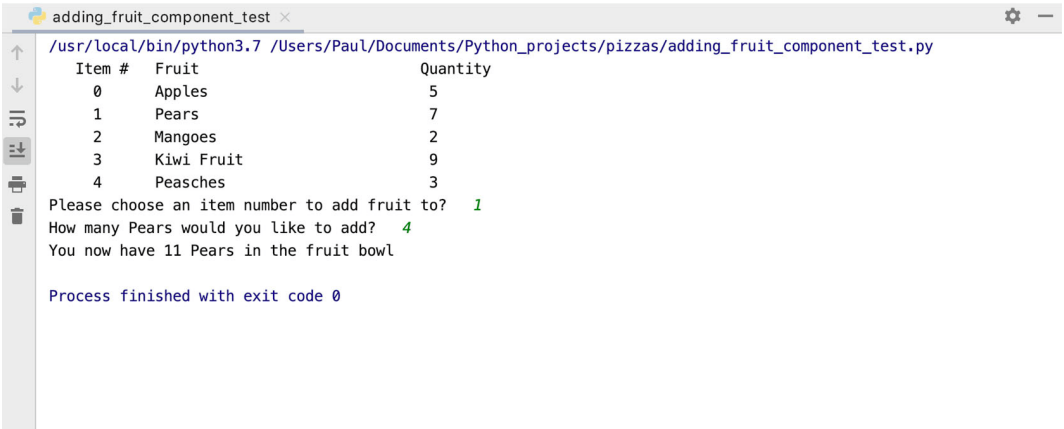
- Testing exists to find errors in a program (not to avoid finding errors).
- Tests should consciously seek to find errors by exploring different input patterns.
- We have a general principle of testing for **expected** , **boundary** and **unexpected** inputs
- Complete the tests you have planned before 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.



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.



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.



Expected input test, works correctly.

3.1.2 Sprint Review

The problems with program crashes with unexpected input and starting to annoying.
Solution: move validation (which helps the user by prventing errors, or giving feedback on how to detect errors) up the project backlog and make this the nest sprint.

4 Component Testing