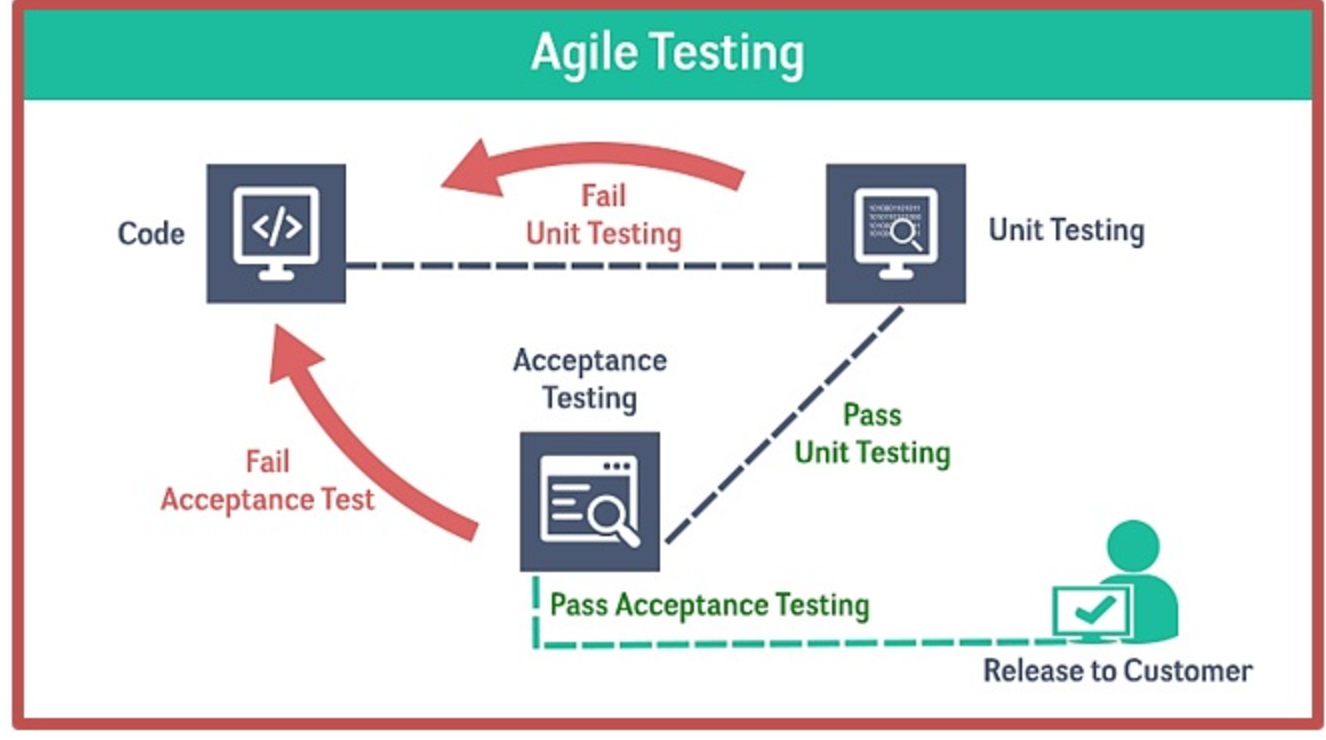
**Development Methods**

For this project I will be using **SCRUM Testing Methodology.**



I believe this is the best Testing Methodology to approach this project, as I was given requirements that are needed in the application, meaning I can focus on each functionalities instead of worrying about the whole project as a whole. I believe this is a good approach for a project with listed features, it also allows me to give time to relearn the Android Studio IDE as I go along.

Using this method will let me build the application slowly and allows me to integrate the application as a whole when all the functions pass the acceptance testing.

This will also save me a lot of time of finding more difficult/complex bugs later in the stage of development. By Identifying them early on the stage.

Using SCRUM also allows me to have goals to accomplish for each sprints.

Advantages of Scrum:

* Increase the quality of the deliverables
* Cope better with change (and expect the changes)
* Provide better estimates while spending less time creating them
* Be more in control of the project schedule and state

**Project Execution and Sprint Plan**

This project will be done in a span of 4 weeks divided into 3 sprints

**First Sprint - Week 1**

* Write up Project Plans
* Create Diagrams
* Write up Test Cases
* Design GUI

**Second Sprint - Week (2 and 3)**

* Create Database
* Code All Required Functionalities
* Rigorous Testing
* Refining Graphical User Interface ( Makes the Application Look Better )

**Third Sprint - Week 4**

* Add Extra Features if necessary
* Continue Testing
* Code Tidying
* Finish Documents

**Development Phases**

**Plan Phase:**

During this Phase I must plan what features that are needed to include during the development sprint, I must write a list of priority features that should be done as soon as possible, the requirements should be focused early on the sprint of the deliverable system as soon as possible to avoid time management issues.

**Design phase:**

After the plan phase, I will have to do plan out and do several diagrams like use cases and flow Charts, during this phase I also need to design the Graphical user interface for each functionalities that are needed to be done. For the GUI I will be drawing them all using the Android Studio Ide, this will also serve as the views/layout when I start coding the java classes, I also plan to use StarUML to create and designing the Database for this application, but some these plans might have to be changed during the sprint but maybe not much.

**Implementation Phase**

During this phase, this is where I'll start coding all the classes, function and writing up the database for this application, but I think when it comes to this phase since I am using the scrum testing methodology the testing and implementation will probably be done in this phase together, the aim is to complete all the function before during each sprints.

**Testing Phase**

During this phase I will be using this approach.

1. Test Plan - Write Test Plan Schedule
2. Test Case - Write Test Cases
3. Test Execution - Executes Test Cases
4. Test Results - Log Test Results

**Unit Testing -> Integration Testing -> System Testing**

**Unit Testing** - Test each units for each functionality

**Integration Testing** - put the units together as a whole making sure it works properly.

**System Testing** - Integrate all functionalities to a system and test it.

During the whole testing project I will be taking regression into account, I have to make sure everything works and nothing breaks when putting pieces together.

**Review Phase**

This is the last phase and it will consists mostly of checking/reviewing the features and documents for this project, maybe updating the documents if necessary.

**Risk Management**

Time Management - This is very important to take this into account that I stick to the the schedule of development as it is summer and I have things planned out to do.

Testing not done correctly - Poor unit or integration can lead further problems when putting all the codes together so it's best to take the time and do the test properly to avoid this.

Poor coding - Coding should be done neatly so that finding errors and bugs are faster and also easier for the professor to read the code.

Poor requirement Analysis - having poor requirement analysis can lead to missing functionalities and test scenarios so I must do the requirement analysis properly.

# **High Level System Architecture**

For this application the system architecture will divided into three parts.

**User Interface** - The system will be developed in java android application to accommodate the different levels of access, All the system interactions are perform through a GUI in the application itself, All informations get are from a local SQlite backend Database using the Data Object Access Pattern.

**Business Logic** - Different users can download the application for them to use, my plan is to change from a local database to an online database if I get access to a server.

**Data -** All Information that are sent/retrieve on this system are saved in the backend database. The application itself store the data to the phone's local storage with the help of the DAO pattern it allows a smooth information access.