## 1.6 METHODOLOGY

The requirements were obtained after a careful study of the existing system. This was done through interviews and careful observation. High priority will therefore be given to user requirements and will be incremented throughout the development of the system.

[3]The waterfall model is an example of a plan-driven methodology—in principle, you must plan and schedule all of the process activities before starting work on them. The approach we are using for the system development is the ***incremental waterfall process* model**4. This approach was chosen over the evolutionary waterfall process model, where one phase can be iterated several times as much as requirements keep changing. Whereas in the single step or incremental waterfall process model, every phase is completed before the next phase can begin. Software development shall thus follow an incremental development process in which continually expanding software versions are based on learning from earlier development.

In principle, the result of each phase is one or more documents that are approved (‘signed off’). The following phase should not start until the previous phase has finished. In practice, these stages overlap and feed information to each other. During design, problems with requirements are identified. During coding, design problems are found and so on. The software process is not a simple linear model but involves feedback from one phase to another. Documents produced in each phase may then have to be modified to reflect the changes made. Because of the costs of producing and approving documents, iterations can be costly and involve significant rework. Therefore, after a small number of iterations, it is normal to freeze parts of the development, such as the specification, and to continue with the latter development stages. Problems are left for later resolution, ignored, or programmed around. This premature freezing of requirements may mean that the system won’t do what the user wants. It may also lead to badly structured systems as design problems are circumvented by implementation tricks. During the final life cycle phase (operation and maintenance) the software is put into use. Errors and omissions in the original software requirements are discovered. Program and design errors emerge and the need for new functionality is identified. The system must therefore evolve to remain useful. Making these changes (software maintenance) may involve repeating previous process stages.

The principal stages of the waterfall model directly reflect the fundamental development activities:

1. **Requirements and Analysis Definition:** the system’s services, constraints, and goals are established by consultation with system users. Through direct interviews and observation, the requirements of the system are spelt out and will be used in further development. They are then defined in detail and serve as a system specification.

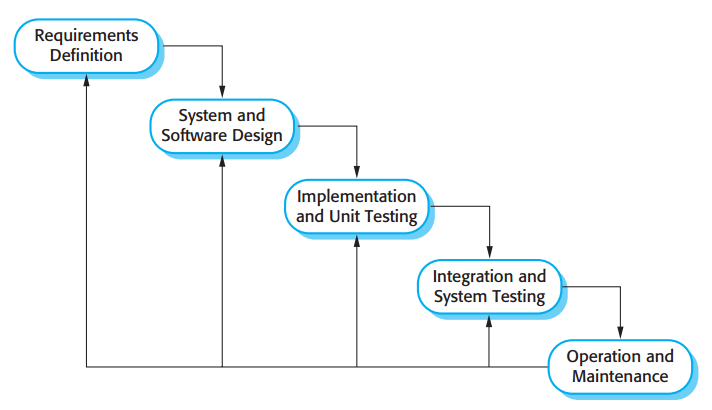
Analysis helps to establish the overall system’s architecture. It involves establishing a relationship between the requirements and system design. The analysis phase thus checks the feasibility of the systems requirements.

2. **System and Software Design:** the systems design process allocates the requirements to either hardware or software systems by establishing an overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships whilst hardware design specifies the type of hardware that can support the system.

3. **Implementation and Unit Testing:** during this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

4. **Integration and System Testing:** the individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

5. **Operation and Maintenance:** normally (although not necessarily), this is the longest life cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered.



***Figure 1.6.1: Incremental Development Process diagram- waterfall model (Software life – cycle)***

This model had its origin from the manufacturing and construction industries. The first known implementation describing use similar phases in software engineering was held by Herbert D. Benington on 29th June 1956. [3] This model places a lot of emphasis on documentation to aid in future developments. The model begins with specifying the systems requirements and analyzing these requirements to produce a good design. Based on the design, the system is then implemented/coded to ensure that the integration of the system proceeds smoothly. This process is incremented from the top in that one phase leads to another. The incremented system is finally tested and put into operation. In evolutionary waterfall model, the whole process is iterative and therefore one process can be performed more than once.

## 1.6 PROJECT ACTIVITY PLAN

The entire project is expected to cover a period of twenty weeks (five months) spanning from January 2016 to May 2016 by which time we hope to have accomplished all the above stated objectives of the project.

The project has in effect been divided into a number of phases or stages. These phases are stated and briefly explained bellow;

* **Requirements Analysis Phase:** this phase comprises all the studies that will be done in order to give the team a good understanding of the existing system. Activities such as interviewing of fuel station workers, observation of operations, internet research and the likes.
* **Design Phase:** after all the analyses have been done, the platform is set for the new system to be designed. This phase takes care of that aspect of the project thus the designing of the new project is done (usually on paper) and even tested to see how well it will work with the system under study.

**Implementation Phase:** this phase actually contains the bigger bulk of the work to be done. Here the designed system is created using appropriate software applications and support systems. The implementation stage is actually divided into other smaller activities which include; **coding / building** and **testing**. Where the coding/building involves actually building the device and/or typing source codes to display the design in a computer program form and the testing involves making use of the program to see if it behaves as expected and if not, changes are effected to make sure it does.