**Software development paradigms**

Software development paradigms refer to the methods and steps, which are taken while designing the software. The paradigms is usually chosen based on the nature of the project or application.

### ****There are different types of Software Development paradigms or models.They Include.****

* Linear sequential/process models“Waterfall model”.
* Evolutionary process models.
* Incremental process model.

Other specialized process models are Big Bang Model,Agile Model, and V-Model, which take on many of the characteristics of one or more of the traditional models presented in the above.

**The Evolutionary Process Models (Iterative)**

They are Iterative meaning, they are characterized by repetition of phases, you go back and for between phase till the whole process is completed. They are grouped into two;

* prototyping model
* spiral model

**prototyping model**

In prototyping model the customer is involved through out the development process. With this model, the developer listen to the customer first, and develop a prototype (mock up) prior to the actual software.

Prototype (mockup)

Is a version of software developed briefly which displays the functionality of the product under development, but no the actual product. Prototype looks like the real thing but not the real product.

The prototype will be sent back to the customer to test and the customer gives feedback about the prototype. Another mock up is developed based on the customer feedback.

It continues until the customer is satisfied and then the actual development of the product begins.

**Reasons why I prefer Prototype model over the other models as a developer and a user;**

* In the development process of this model users are actively involved.
* The development process is the best platform to understand the system by the user.
* Gives quick user feedback for better solutions.
* It identifies the missing functionality easily. It also identifies the confusing or difficult functions.
* Prototype model need not know the detailed input, output, processes, adaptability of operating system and full machine interaction.
* Less risk on final product rejection, since errors are detected much earlier and corrected.