**CHAPTER 4**

**SOFTWARE PROCESS MODEL**

**SOFTWARE PROCESS**

Software process models have been used for many years in an effort to bring order and structure to software development. Each of these models suggests a different process flow, but all perform the same set of activities. A process model for software engineering is chosen based on the nature of the project, the methods and tools to be used and required.

A software process model has predefined steps, accomplishing those steps ultimately results in a software. So, the software process model is followed step by step to develop software.

There are many software process models. These are as follows,

* Linear Sequential Model.
* The Prototyping Model.
* The RAD Model.
* Incremental Model.
* Spiral Model.

Linear sequential model is sometimes called the Classic life cycle or waterfall model, the linear sequential model suggests a systematic, sequential approach to software development that begins at the system level and progress through analysis, designing, coding, testing and implementation. The linear sequential model has the following activities:

Communication

Modeling

Construction

Deployment

* Communication.
* Modeling.
* Construction.
* Deployment.

**4.1) COMMUNICATION**

Communication works to understand the business problems and the information characteristics. That the software must accommodate. Communication should be simple not to be difficult that the user and the software developer did not understand each other. And the problems remain undefined. There should be no communication gap that no ambiguity remains.

**4.1.1) MODELING**

The requirement gathering process is focused specially on software. To understand the nature of the programs to be built, the software engineer must understand the information for the software as well as required functions, behavior, performance, interface.

Requirements for both system and the software are documented and reviewed with the customer.

The process of analyzing the organization and its environment, developing a system design that accurately represents the organization’s functioning in the real world, and implementing that system using an appropriate methodology.

**4.1.2) CONSTRUCTION**

The code generation step performs the task of developing a project. If design is performed in a detailed manner, code generation can be accomplished mechanistically in a specified computer language and tool.

At the accomplishing the coding phase is start. The testing process focuses on the logics of the software, ensuring that all statements have been tested, and all the external functions, that is conducting tests to uncover errors and ensure that defined input will produce actual output that is according to the required results.

**4.1.3) DEPLOYMENT**

Once the code is developed and accomplished the software is ready for deployment after the whole phase of construction including testing. Deployment includes delivery, support and feedback from the user.

**4.2) DATA FLOW DIAGRAM**

Data flow diagrams (DFD’s) show the flow of the data through the organization. They provide a useful model for communicating with the users about the proposed system. Data flow diagrams may be constructed at several levels, to show different amounts of detail about the system. There are few rules to be followed in constructing the data flow diagram, showing the various aspects of the system and various levels of details.

**4.3) TYPES OF DATA FLOW DIAGRAM**

There are three types of data flow diagrams:

* Zero (0) level data flow diagram.
* Context level data flow diagram.
* Detailed level data flow diagram.

**4.3.1) Zero (0) LEVEL DATA FLOW DIAGRAM**

Zero level data flow diagram showing the system itself. It pictures the major processes along with the external entities, data stores and data flow.

**4.3.2) CONTEXT LEVEL DATA FLOW DIAGRAM**

The system as a whole is represented by a bubble and the external entities are shown as squares from which input flows and to which output is directed.

**4.3.3) DETAILED LEVEL DATA FLOW DIAGRAM**

Detailed level data flow diagrams are explosions of higher-level representations and are detailed pictures of major processes.

**4.4) SYMBOLS FOR DATA FLOW DIAGRAM**

**SYMBOLNAMEMEANING**

Entity A person or organization.

Data flow Shows flow of data.

Data base Where data is stored.