LAB TWO:BY **JOSEPH NKWATE NDOMU**

**1) Propose the most appropriate generic software process model that might be used as a basis for the development of following systems. Give reasons for your answers.**

**a) A complex real-time system whose requirements can be relatively easily identified and are stable.**

**b) A web-site for a local library. Requirements are vague and are likely to change in the future.**

**c) An order processing system with a web-site for a local business. Requirements are vague but stable (i.e., unlikely to change in the near future).**

**2) Describe the software process model that you have proposed in question 5(a) highlighting its strengths and weaknesses.**

***ANSWERS***

1. **WATERFALL MODEL.**

• This type of system is safety-critical. A well-thought-out analysis is needed before implementation. Therefore, a plan-driven approach to development is needed to ensure that the requirements are analyzed carefully.

• Therefore, the most appropriate generic process model to use is the waterfall model.

1. **INCREMENTAL MODEL**

An incremental development approach has the following advantages: The process is more responsive to changing user requirements than a waterfall approach — later sub- systems can be re-specified. Also, a modular approach can mean maintenance changes are simpler and less expensive.

1. **V Model**

This model is preferable since All the requirements are gathered at the start and cannot be changed. You have a corresponding testing activity for each stage. For every phase in the development cycle, there is an **associated testing phase.**

1. **WATERFALL MODEL.**

Many consider the waterfall method to be the most traditional software development method. The waterfall method is a rigid linear model that consists of sequential phases (requirements, design, implementation, verification, maintenance) focusing on distinct goals.

* The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
* But the major setback of this model is No working software is produced until late during the life cycle. High amounts of risk and uncertainty. Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects.