



Sessional Assignment

Information System & Data Processing

Submitted By: Muhammad Zain Ul Abideen
ID# 13740
BSSE

Submitted To: Mr. Muhammad Abrar Khan (Lecturer)

Sessional Assignment

- 1) Define organization; also explain the structure of an organization by giving an example of a well known organization. (Note: every student should take the example of different organization from another).

Ans: An **Organization** is referred to a social unit of people that is structured and managed to meet a need or to pursue collective goals. All organizations have a management structure that determines relationships between the different activities and the members, and subdivides and assigns roles, responsibilities, and authority to carry out different tasks. Organizations are open systems--they affect and are affected by their environment.

Structure of Organization:

The organizational structure of a business defines its entire culture. It affects how its employees communicate and operate to achieve the goals of the company. Organizational structure also influences productivity, employee relations, and marketing strategies. It is important for a company to define the most strategic organizational structure that will allow it to effectively and efficiently accomplish its goals. The organizational structure of a small company may differ from that of a large company. Let's take a look at some of the most common organizational structures.

Organisational Structure of the World Trade Organization (WTO)

The **Ministerial Conference** is the highest organ of the WTO and is to meet at least once every two years. It is normally composed of all the Ministers of Trade of the Members of the WTO. The Ministerial Conference has supreme authority over all matters.

The **General Council** is composed of representatives of all the members – normally country delegates based in Geneva. The General Council is in session between the meetings of the Ministerial Council. In essence this is the real engine of the WTO and has all the powers of the Ministerial Council when that body is not in operation. The General Council also acts as the Dispute Settlement Body and the Trade Policy Body.

The **Council for Trade in Goods**, the **Council for Trade in Services** and the **Council for Trade-Related Aspects of Intellectual Property Rights (TRIPS)** have been established with specific spheres of responsibility. In fact, there are separate agreements within the framework of WTO that define and confine their operation.

The **Committee on Trade and Development**, the **Committee on Balance-of-Payments Restrictions** and the **Committee on Budget, Finance and Administration** have self-evident functions. Likewise the **Director-General** and the **Secretariat** operate on a purely

administrative basis. However, it should be stated that the Director-General and the staff of the Secretariat shall be exclusively international in character and they shall not seek or accept instructions from any government or any other authority external to the WTO.

2) Explain System Development Life Cycle; also explain different types system development life cycle.

Ans: System Development Life Cycle (SDLC):

In systems engineering, information systems and software engineering, the systems development life cycle (SDLC), also referred to as the application development life-cycle, is a process for planning, creating, testing, and deploying an information system. The systems development life cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both. There are usually six stages in this cycle: requirement analysis, design, development and testing, implementation, documentation, and evaluation.

A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information systems. Like anything that is manufactured on an assembly line, an SDLC aims to produce high-quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time frames and cost estimates.

Types of System Development Life Cycle:

1. Waterfall Model

Waterfall is the oldest and most straightforward of the structured SDLC methodologies — finish one phase, then move on to the next. No going back. Each stage relies on information from the previous stage and has its own project plan. Waterfall is easy to understand and simple to manage.

But early delays can throw off the entire project timeline. And since there is little room for revisions once a stage is completed, problems can't be fixed until you get to the maintenance stage. This model doesn't work well if flexibility is needed or if the project is long term and ongoing.

2. V-Shaped Model

Also known as the Verification and Validation model, the V-shaped model grew out of Waterfall and is characterized by a corresponding testing phase for each development stage. Like Waterfall, each stage begins only after the previous one has ended.

This model is useful when there are no unknown requirements, as it's still difficult to go back and make changes.

3. Iterative Model

The Iterative model is repetition incarnate. Instead of starting with fully known requirements, you implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready.

One advantage over other SDLC methodologies: This model gives you a working version early in the process and makes it less expensive to implement changes. One disadvantage: Resources can quickly be eaten up by repeating the process again and again.

4. Spiral Model

One of the most flexible SDLC methodologies, the Spiral model takes a cue from the Iterative model and its repetition; the project passes through four phases over and over in a "spiral" until completed, allowing for multiple rounds of refinement.

This model allows for the building of a highly customized product, and user feedback can be incorporated from early on in the project. But the risk you run is creating a never-ending spiral for a project that goes on and on.

5. Big Bang Model

A bit of an anomaly among SDLC methodologies, the Big Bang model follows no specific process, and very little time is spent on planning. The majority of resources are thrown toward development, and even the client may not have a solid grasp of the requirements. This is one of the SDLC methodologies typically used for small projects with only one or two software engineers.

Big Bang is not recommended for large or complex projects, as it's a high-risk model; if the requirements are misunderstood in the beginning, you could get to the end and realize the project may have to be started all over again.

6. Agile Model

By breaking the product into cycles, the Agile model quickly delivers a working product and is considered a very realistic development approach. The model produces ongoing releases,

each with small, incremental changes from the previous release. At each iteration, the product is tested.

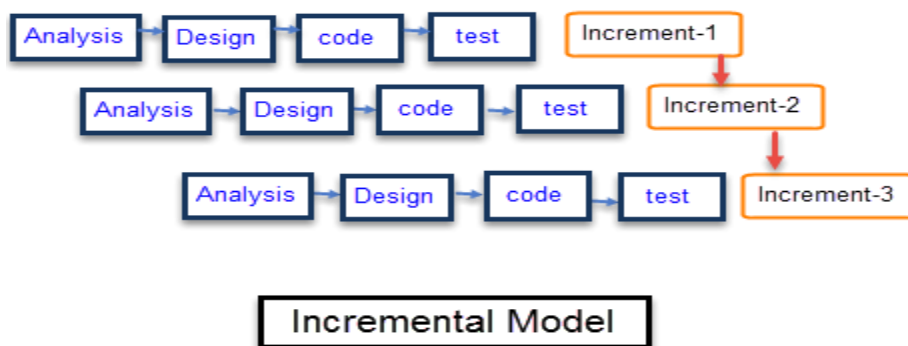
This model emphasizes interaction, as the customers, developers and testers work together throughout the project. But since this model depends heavily on customer interaction, the project can head the wrong way if the customer is not clear on the direction he or she wants to go.

3) Explain Incremental model and Spiral; also explain main difference between spiral and incremental model.

Ans: **Incremental model:**

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented. The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

Like:



Characteristics of an Incremental module includes

- System development is broken down into many mini development projects
- Partial systems are successively built to produce a final total system
- Highest priority requirement is tackled first
- Once the requirement is developed, requirement for that increment are frozen

Phases of Incremental:

Incremental Phases	Activities performed in incremental phases
Requirement Analysis	Requirement and specification of the software are collected
Design	Some high-end function are designed during this stage
Code	Coding of software is done during this stage
Test	Once the system is deployed, it goes through the testing phase

Advantages and Disadvantages of Incremental Model

Advantages

The software will be generated quickly during the software life cycle.
It is flexible and less expensive to change requirements and scope.
Throughout the development stages changes can be done.
This model is less costly compared to others.
A customer can respond to each building.
Errors are easy to be identified.

Disadvantages

It requires a good planning designing.
Problems might cause due to system architecture as such not all requirements collected up front for the entire software lifecycle.
Each iteration phase is rigid and does not overlap each other.
Rectifying a problem in one unit requires correction in all the units and consumes a lot of time.

When to use Incremental models?

- Requirements of the system are clearly understood
- When demand for an early release of a product arises
- When software engineering team are not very well skilled or trained
- When high-risk features and goals are involved
- Such methodology is more in use for web application and product based companies

Spiral Model

The spiral model was defined by Barry Boehm in his 1988 article. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration

matters. Spiral model is an evolutionary software process model which is a combination of an iterative nature of prototyping and systematic aspects of traditional waterfall model.



Phases of Spiral Model

Planning

- The objectives, alternatives and constraints of the project are determined and are documented.

Risk Analysis

- All possible alternatives, which can help in developing a cost effective project are analyzed
- This phase identify and resolve all the possible risks in the project development

Engineering

- The actual development of the project is carried out
- The output of this phase is passed through all the phases iteratively in order to obtain improvements in the same.

Customer Evaluation

- Developed product is passed on to the customer in order to receive customer's comments and suggestions.
- This phase is very much similar to TESTING phase.

Advantages of Spiral Model

- Changing requirements can be accommodated.
- Allows for extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Early and frequent feedback from users

Disadvantages of Spiral Model

- End of project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex
- Spiral may go indefinitely.
- Risk assessment expertise is required.

When to use Spiral Model

- When costs and risk evaluation is important.
- For medium to high-risk projects.
- Users are unsure of their needs.
- Requirements are complex.
- Significant changes are expected.

Main difference between spiral and incremental model.

A “**spiral model**” is a type of iterative model where the individual iterations in the spiral model take the form of mini-waterfalls. There are many other kinds of iterative models that are much simpler than the spiral model. For example, Scrum would be considered to be both an incremental and an iterative approach.

The **Incremental** approach is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements