## Server Side Technologies -Week 2

### Software Engineering The Software Process

#### Aim and Objectives

Aim : To understand the role of a software process

Objectives :To understand

* What a software process is
* System Development Lifecycle (AKA Waterfall)
* Incremental
* Iterative

### What is a Software Process?

* A set of activities whose goal is the development of software
* Generic activities in all software processes are:

**Specification** - what the system should do and its development constraints

**Development** - production of the software system

**Validation** - checking that the software is what the customer wants

**Evolution** - changing the software in response to changing demands

### Waterfall Development (SDLC)



### Waterfall Development - Phases

* Requirements analysis and definition
  + The system’s services, constraints and goals are established by consultation with system users.
* System and software design
  + Establishes an overall system architecture
  + Identifies and describes the the fundamental software system abstractions and their relationships.
* Implementation and unit testing
  + The software design is realised as a set of programs or program units.
* 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.
* Operation and maintenance
  + Normally this is the longest life-cycle phase.
  + Maintenance involves correcting errors, which were not discovered in earlier stages of the lifecycle.

### Waterfall Development

* The *Waterfall or linear approach* implies that you do each phase in sequence and that each phase cascades into the next stage.
* In the original waterfall model, a strict sequential approach is implied.
* This means that one phase has to be complete before the next phase begins.
* It also does not provide for feedback between phases or for updating or re-definition of earlier phases.
* Waterfall Development
* The drawback of the waterfall model is the difficulty of accommodating change after the process is underway
* Therefore, this model is only appropriate when the requirements are well-understood
* A.K.A. The “**S**oftware **D**evelopment **L**ife**c**ycle”.

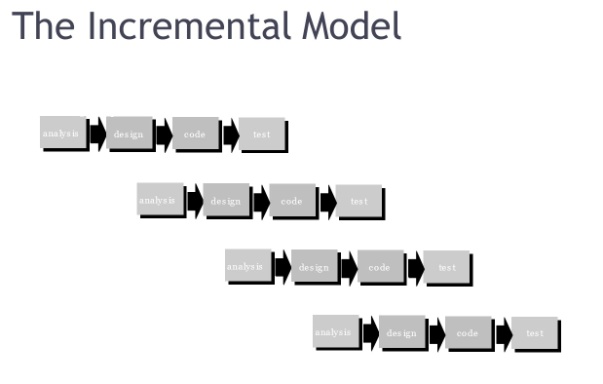
### Advantages and Disadvantages of the System Development Life Cycle

Advantages

* Used by nearly all organisations.
* For management it provides:
* A degree of predictability.
* Helps with resource allocation.
* A framework that fits nicely with project management techniques
* It is especially good for:
* Well-structured systems
* large systems
* Systems with a long life span.

Dsadvantages

* Model focuses on ‘what’ rather than ‘how’.
* It assumes the ‘what’ can be separated from the ‘how’. People think of solutions while solving problems
* Requirements are rarely complete.
* Some problems are not fully understood until they are solved
* Users don’t see anything until the end.
* Phases of the life cycle generate a great deal of documentation



### Types of Systems Development Lifecycles

The fountain or iterative approach

* This *iterative approach* implies that you do some analysis, then some design, and then some implementation.
* Based on what you learn, you cycle back through and do more analysis, etc. This supports human learning a lot better.

#### Prototyping

* Prototyping aims to enhance the accuracy of the designer's perception of the user's requirements.
* It is based on the idea of developing an initial implementation for user feedback, and then refining this prototype through many versions until a satisfactory system emerges.
* The specification, development and validation activities are carried out concurrently with rapid feedback across the activities.

#### Advantages of Prototyping

* Prototyping encourages more user involvement in system development.
* Iteration & change are a natural consequence of systems development, (that is, end-users tend to change their minds.)
* Prototypes are an active, not passive, model that end-users can experience.
* An approved prototype is a working equivalent to a paper design specification, with one exception - errors can be detected earlier.
* Prototyping can increase creativity because it allows for quicker user feedback, which can lead to better solutions.
* Ownership of the system is gradually transferred to the users.
* Generally lowers the cost of designing systems.

#### Disadvantages of Prototyping

* Prototyping is difficult to manage and control.
* Not suitable for large information systems. (???)
* Documentation tends to be poor.
* Often produces less efficient system than might be developed by the standard methodology approach.
* It is difficult to maintain user enthusiasm.
* Prototyping encourages a return to the “code, implement, and repair'' life cycle. (Unless used with discipline).
* You cannot completely substitute any prototype for a paper specification.
* There are many design issues not addressed by prototyping.
* Prototyping often leads to premature commitment to a design.

### Choosing the Best Development Process for a Project

* Rapid Application Development when the project is small and narrowly focused and it involves few people and a small number of business processes.
* It is quite effective if the project is "experimental" in nature and users simply want to provide a "pilot" system to try out.
* It should not be used if the risks of failure involve high stakes.
* You should use the traditional systems development life cycle approach when the project scope is wide and the system involves many people and processes.
* This is usually the situation when a project's outcome involves essential operational or managerial information processing that supports essential decision-making.
* Failure in those cases could result in financial losses or a disruption in critical business operations.
* Although the time and cost savings are an important advantage of the RAD approach,
* The traditional systems development life cycle (SDLC) has less risk.

### Still Other Process Models

* Component assembly model—the process to apply when reuse is a development objective
* Concurrent process model—recognizes that different part of the project will be at different places in the process
* Formal methods—the process to apply when a mathematical specification is to be developed
* Cleanroom software engineering—emphasizes error detection before testing

### Summary

* What a software process is
* development of software.
* System Development Lifecycle (AKA Waterfall)
* Requirements analysis and definition, System and software design, Implementation and unit testing, Integration and system testing, Operation and maintenance
* Incremental
* Iterative
* Prototyping, Rapid Application development