

1 Software Process

(*) The Structured Set of activities needed to develop a Software System.

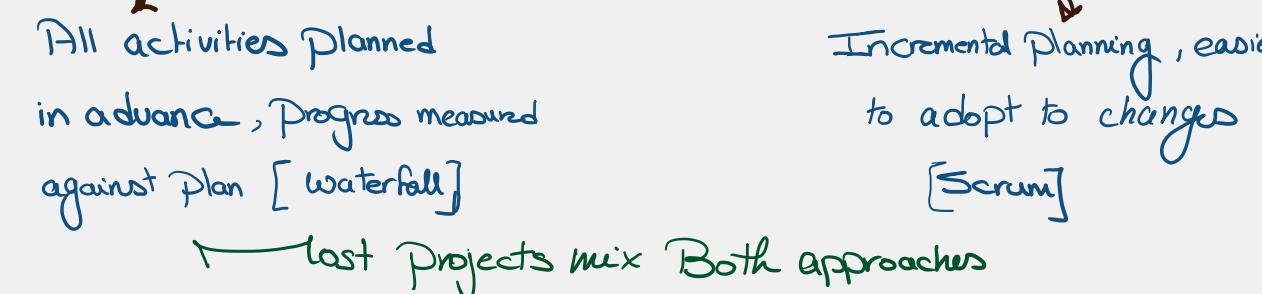
Main Activities:

- ① Specifications: Define what System Should do
- ② Design of implementation: Build the System
- ③ Validation: check if it meets Customer needs
- ④ Evaluation: update the System as requirements change

Process Model: Abstract representation of these activities

2 Types of Software processes

Plan-driven Agile



3 Process Models

① Waterfall Model

Requirements → Design → Implementation → Testing → Maintenance

* Good for Stable requirements

* Problem: very rigid, hard to handle change.

② Incremental Developments

Build Software in increments → Small version.

* Benefit: Easier to adapt changes, faster customer feedback, earlier delivery

* Problem: System Structure may degrade over time, process less visible to managers.

③ Reuse-oriented Software engineering

Build System from existing Components
Component analysis → Adopt requirements → System design → Integration with reuse

- Common in business today.

4 Core process Activities

① Specifications

Requirements engineering [feasibility study, elicitation, Specifications, validation]

② Design and implementation

- Architectural design → Overall Structure
- Interface design → Define Connections between Components
- Component design → Internal workings of models
- Database design → Structure and Storage

③ Validation [Testing]

- ① Component testing → Individual Modules
- ② System Testing → Whole System
- ③ Acceptance testing → With customer data.

④ Evolution: Software must adapt to new business needs, technology or environment

5 Coping with change

change avoidance

* Anticipate changes

change tolerance

* Design process so changes are easy.

6 Prototyping

- * Benefits: Better usability, better requirement understanding, reduced development effort
* Usually thrown away → Not used in final System.

7 Incremental Delivery

* Deliver System in Small usable parts [increments]

* Advantages: early customer value, reduced project risk, acts as prototype

* Problems: Hard to define shared facilities upfront, Conflicts with Contracts that require full System Specs.