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

- Plan-driven: All activities planned in advance, Progress measured against plan [waterfall]
- Agile: Incremental Planning, easier to adopt to changes [Scrum]

↳ Most Projects mix Both approaches

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.

* Benifit: Easier to adopt changes, faster Customer feedback, earlier delivery
* Problems: System Structure may degrade over time, process less visible to managers.

③ Reuse-oriented Software engineering → Build System from existing Components

Component analysis → Adapt requirements → System design with reuse → Integration

- Common in business days Today.

4 Core process Activities

① Specifications

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

② ① 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 is inevitable → Business requirements, Tech updates, new platforms.

- change avoidance ↳ Anticipate changes
- Change tolerance ↳ Design process so changes are easy.

6 Prototyping → early version of System to test Concepts

* Benefits: Better usability, better requirement understanding, reduced development effort
* ↳ usually thrown away → not used in final System.

7 Incremental ① Delivery → ① Deliver System in Small usable Parts [increments]

* Advantages: early Customer value, educed project risk, acts as prototype
* Problems: Hard to define Shared Facilities upfront, Conflicts with Contracts that require full System Specs.