

1] Software Process

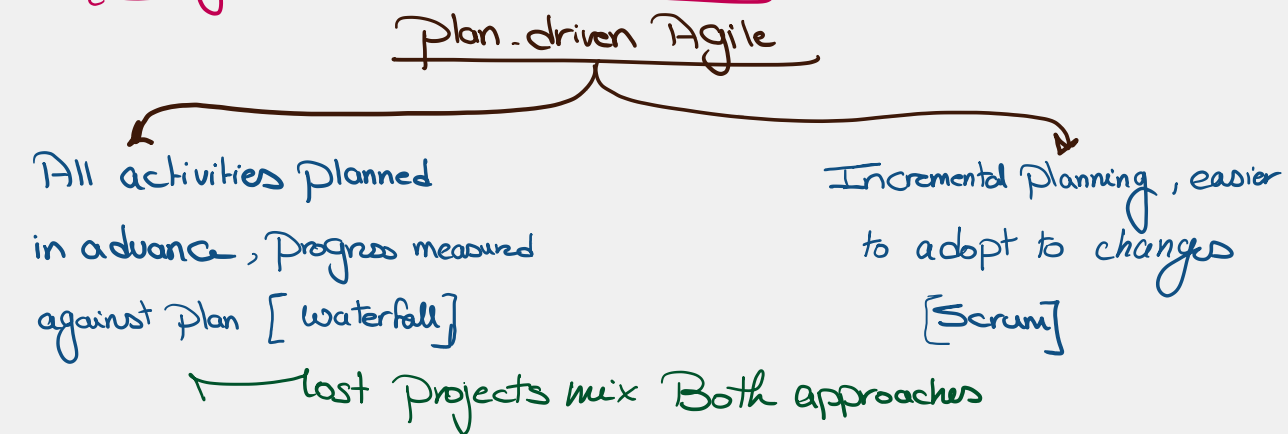
↳ The structured set of activities needed to develop a Software System.

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 Models: Abstract representation of these activities

2] Types of Software processes



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 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 → Adopt requirements → System design → Integration with reuse

- Common in business days 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 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 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.