RIPHAH INTERNATIONAL UNIVERSITY

Activity#01

Course: **Software Construction and Development**

NAME: AHMED AZIZ

SAP ID: 55223

Software Development Process Model

Waterfall Model

Use Case:

- Small to medium projects.
- Requirements are clear, fixed, and not expected to change (e.g., Payroll System, School Management System).

Pros:

- Simple, easy to understand.
- Clear structure (step by step).
- Good for projects with stable requirements.

Limitations:

- No flexibility for changes.
- Testing happens very late defects found late.
- Not suitable for complex, evolving systems.

V-Model (Verification & Validation)

Use Case:

- Safety-critical and high-risk systems (e.g., Banking, Medical Devices, ATMs).

Pros:

- Testing planned from the start.
- High quality and reliability.
- Clear mapping between development & testing stages.

Limitations:

- Expensive (more documentation & testing).
- Rigid → not good for changing requirements.
-

Spiral Model

Use Case:

- Large, risky, and expensive projects.
- Systems where risk management is critical (e.g., Space Missions, Defense Systems).

Pros:

- Strong focus on risk analysis.
- Combines iterative development with risk handling.
- Customer feedback included at every cycle.

Limitations:

- Complex to manage.
- Costly not good for small projects.
- Requires skilled teams in risk analysis.

Iterative Model

Use Case:

- Projects where requirements are not fully known but will evolve (e.g., Word Processors, Game Development).

Pros:

- Early version available for testing.
- Errors can be found early.
- Improvements happen step by step.

Limitations:

- Customer feedback may still come late.
- More planning and resources needed.
- Can become costly with many iterations.

Incremental Model

Use Case:

- Medium to large projects where system can be divided into modules (e.g., E-commerce Website, Online Banking).

Pros:

- Early delivery of working software.
- Easier to test modules separately.
- Flexible changes can be applied to future increments.

Limitations:

- Needs proper planning for modules.
- Integration of increments can be complex.
- Not suitable if requirements keep changing a lot.

Agile Model

Use Case:

- Projects with changing requirements.
- Startups, mobile apps, web applications (e.g., Food Delivery App, Ride-Hailing App).

Pros:

- Very flexible, adapts to changes.
- Continuous customer feedback.
- Early and frequent delivery of usable product.
- Motivated team, strong collaboration.

Limitations:

- Needs highly skilled, self-organizing teams.
- Not suitable for very large, fixed-scope projects.
- Frequent meetings can feel heavy.

Scrum (Agile Framework)

Use Case:

- Team-based projects where close collaboration and feedback are possible.
- Web/mobile apps, SaaS products, startup projects.

Pros:

- Clear roles (Product Owner, Scrum Master, Team).
- Works in short sprints quick delivery.
- Regular feedback + improvement.

- High transparency.

Limitations:

- Not ideal for very large, distributed teams.
- Needs discipline and skilled team members.
- Scope creep risk due to changing backlog.

Summary

Model	Best Use Case	Pros	Limitations
Waterfall Model	Small, stable projects	Simple, structured	late testing
V-Model	High-risk, safety-critical	Quality, testing early	Costly, rigid
Spiral	Large, risky projects	Risk control, feedback	Complex, expensive
Iterative	Evolving requirements	Early versions, improvements	Costly, late feedback
Incremental	Modular systems	Early delivery, flexible	Integration complexity
Agile	Dynamic projects, startups	Flexible, customer-driven	Needs skilled teams
Scrum	Agile team projects	Clear roles, quick delivery	Not for very large teams