Agile development encompasses various approaches aimed at delivering projects iteratively and incrementally while emphasizing adapbility, customer collaboration and responsiveness to change. Below a comparatively analysis of major Agile methodologies:

1. Scrum:

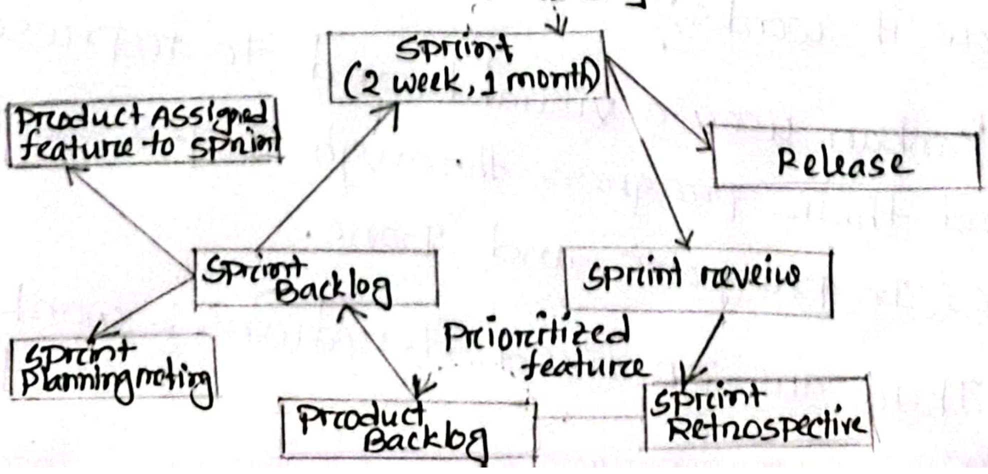
How it works?

⇨ Scrum divides work into fixed-length iterations called sprints, typically lasting 2-4 weeks.

⇨ Team operate in specific roles (Scrum Master, Product Owner and Developers) and ceremonies such as daily standups, sprint planning, reviews, and retrospectives.

⇨ Deliverables are prioritized in a product Backlog, with a focus on incremental value.

↳ Lifecycle of scrum:



Applicability

→ Best for projects requiring deliverables and stakeholder feedback.

→ suitable for complex and dynamic projects where requirements envolve over time.

Effectiveness in terms of costs

→ Cost Efficiency: Highly effective in managing resources due to predictable sprint timelines and early identification of potential issues

→ Example: A software Company building a dynamic e-commarce platform can benefit from scrum as it follows iterative feature releases and Constant stakeholder input.

2. Kanban

How it works?

→ Kanban uses a visual board to represent tasks and their progress through stages such as To Do, In progress, and Done.

→ There are no fixed iterations; work progresses continuously based on task prioritization.

→ Focus on managing work-in-progress (WIP) to ensure Smoth flow of and prevent bottlenecks.

Applicability

→ Ideal for maintenance projects, operational Support, or term· team with up· unpredictable workloads.

→ works well for projects needing continuous delivery rather than fixed iterations.

Effectiveness in terms of costs

→ Cost Efficiency: Reduces wastes and ensures team productivity by minimizing context Switching.

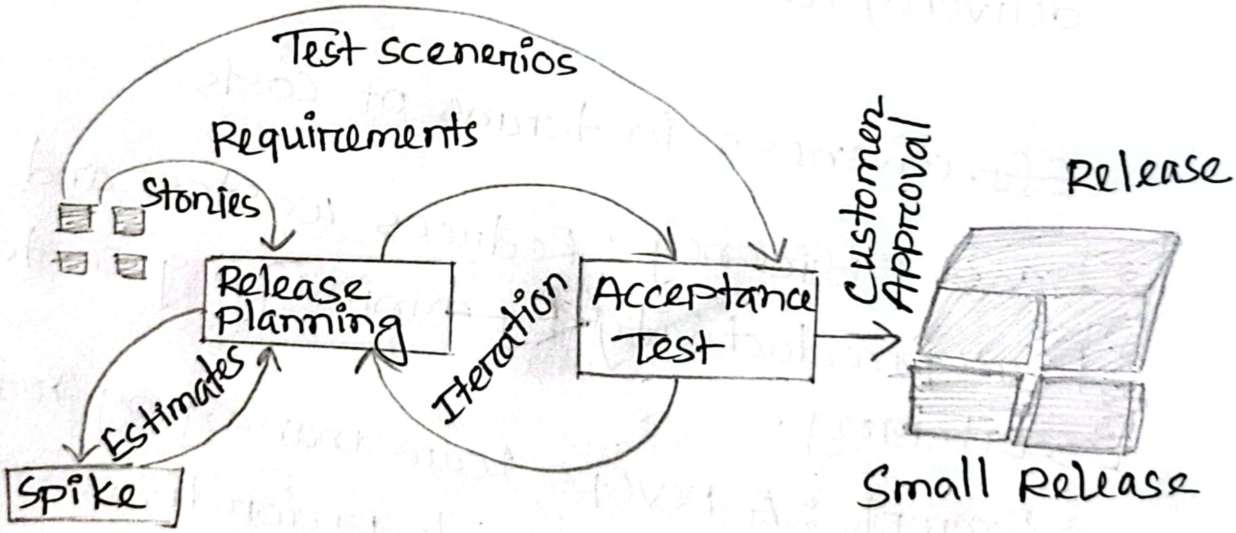
→ Example: A DevOps team managing infractures issues can be kanban to prioritize and resolve task based on urgency.

3. Extreme programming (xp)

How it works?

→ xp. emphasizes technical excellance through practices like pair programming, test-driven developement (TDD) and continuous integration (CI).

→ Iterations are short (1-2 weeks) and customer feedback is critical at every stage.



Applicability

→ Best for projects with rapidly changing requirements and a high need for quality assurance.

→ suitable for small to medium-sized terms focusing on software developement.

Effectiveness in terms of costs

→ Cost Efficiency : Ensure high quality code, reducing long-term maintainance cost but may require more upfront investment in skilled developers and tools.

→ Example : A Startup developing a real-time messaging app can leverage xp to maintain code quality under tight deadlines.

4. Lean Software Development :

How it works?

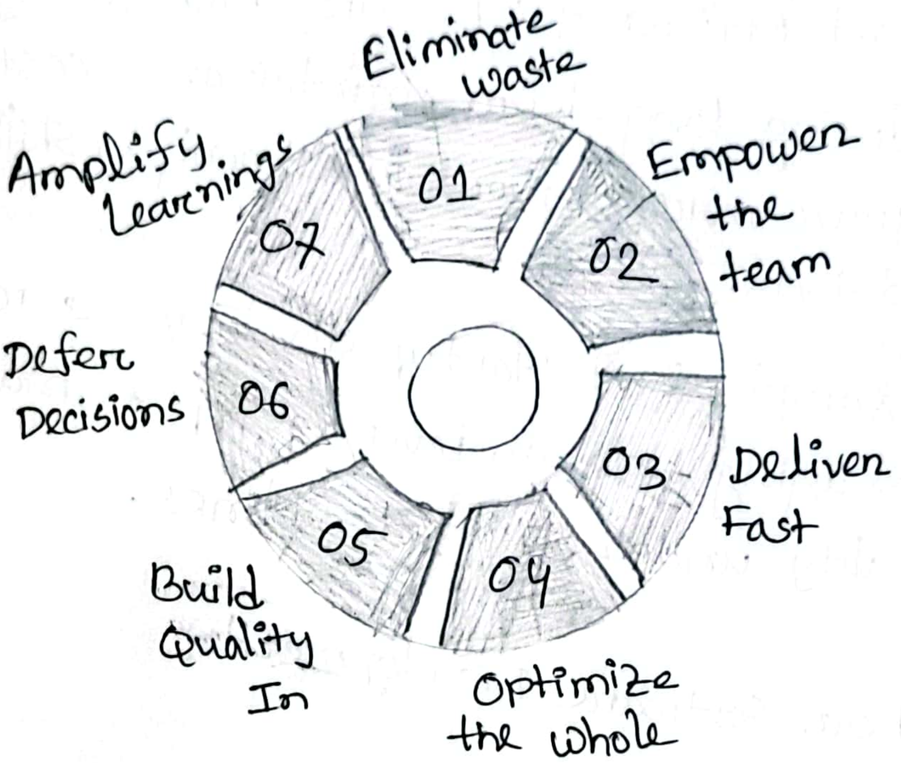
→ Lean focuses on eliminating waste, delivering fast and fast and ensuring optimal value for customers.

→ Practices include limiting WIP, delivering just-in-time features and empowering teams to make decisions.

Applicability

→ suited for projects where efficiency and reduces resources optimization are critical.

→ Applicable in both software and non-software projects.



Agile Lean software developement

Effectiveness in terms of cost

→ Cost Efficiency : Reduces resource waste, making it deal with for budget - constrained Projects.

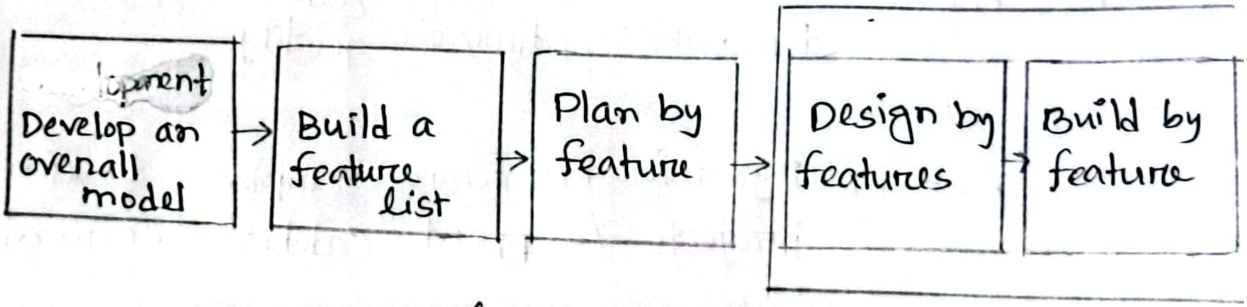
→ Example : A manufacturing firm adopting lean principles can streamline their software for inventory management.

5. Feature Driven Developement (FDD)

How it works ?

→ FDD structures around features (small, client-valued deliverables) and follows a five-step process :

1. Develop an overall model
2. Build a feature list
3. Plan by feature
4. Design by feature
5. Build by feature.



5 Steps of FDD

Applicability

→ Suitable for large-scale, complex projects with clearly defined requirements.

→ works well when there's need for consistance documents.

Effectiveness in terms of costs

→cost Efficiency: Effectiveness in large terms with teams with high coordination needs; initial setup may be resource-intensive.

→Example: A banking Applicable where precise feature delivery is paramount can use FDD for structured progress.

Comparative Analysis Diagram:

Aspects	Scrum	Kanban	XP	Lean	FDD
Iteration	fixed (2-4 weeks)	Continuous	Short (1-2 weeks)	Continous	Feature-based
Focus	Incremental deliverables	Workflow optimization	Code quality	Efficiency	Feature development
Best for	Dynamic Projects	Maintenance /support	Complex Codebases	Efficiency critical	Large well-defined
Cost Efficiency	High	Moderate	High (long-term)	High	Moderate