IT-21001

Assignment

☐ Different Agile Approaches and their comparative analysis.

→ Agile approaches provide frameworks for managing and delivering projects in an iterative and flexible manner. Here's a comparative analysis of the most popular Agile methodologies based on how they work, applicability, effectiveness in terms of costs, and other factors.

1. Scrum:

→ How It Works:

- A team-oriented, iterative framework divided into sprints (typically 2-4 weeks).
- Roles: Scrum Master, Product Owner, and Development Team.
- Uses artifacts like the Product Backl to manage and prioritize tasks.

→ Applicability:
- Ideal for software projects with evolving requirements.
- Works well in small to medium-sized teams where frequent feedback loops are required.

→ Effectiveness in Terms of Costs:
- Initial costs may include training Scrum roles.
- Reduces costs long-term by avoiding large rework cycles and managing scope effectively.

→ Key Strengths:
- Provides clear visibility into project progress.
- Encourages team collaboration and adaptability.

→ challenges:

- Requires cultural alignment and commitment from all team members.

2. XP (Extreme Programming)

→ How It Works:

- Emphasizes engineering practices like pair programming, test-driven development, and continuous integration.
- Focuses on delivering high-quality software in short iterations.

→ Applicability:

- Best for high-risk, complex, or rapidly changing environments.
- Suitable for projects requiring frequent and detailed customer feedback.

→ Effectiveness in Terms of Costs:

- High upfront costs due to rigorous

testing and continuous collaboration.
← Long-term savings by reducing defect rates and improving code quality.

→ Key strengths:
- High-quality output due to rigorous testing.
- Ideal for small, skilled teams.

→ Challenges:
- Demands skilled developers and a collaborative customer.

3. Kanban

→ How It Works:
- Visualizes work items on a board (e.g, Trello) with columns like To Do, In Progress, and Done.
- Limits work-in-progress (WIP) to prevent bottlenecks and improve flow efficieney.

→ Applicability:
- Effective in maintenance, support, or ongoing product improvement projects.
- Suitable for teams with varying task priorities.

→ Effectiveness in Terms of Costs:
- Low implementation costs (does not require dedicated roles).
- High cost efficiency by preventing task overflow and reducing delays.

→ Key Strengths:
- simple and easy to adopt.
- Provides clear insights into workflow efficiency.

→ challenges:
- Lacks time-boxing, making it harder to predict delivery times.

4. Lean Software Development:

→ How It Works:

- Derives principles from lean manufacturing to eliminate waste, improve flows, and maximize value delivery.
- Encourages continuous feedback, rapid iterations, and optimizing workflows.

→ Applicability:

- Ideal for projects with limited resources or where cost-efficiency is paramount.
- Works well in startups or organizations adopting a value-driven approach.

→ Effectiveness in Terms of Costs:

- Very cost-effective by focusing only on value-adding activities.
- Initial setup may be complex, especially

in large organizations.

→ Key Strengths:

- Encourages a customer-centric focus.
- Scales effectively with proper implementation.

→ Challenges:

- Difficult to implement in bureaucratic or hierarchical organizations.

5. Feature-Driven Development (FDD)

→ How It Works:

- Focuses on delivering features in short iterations.
- Uses predefined phases: Develop an overall Model, Build a Feature List, and Plan by Feature.

→ Applicability:
- Suitable for large-scale projects with clearly defined features.
- Works well in teams with diverse skill levels.

→ Effectiveness in Terms of Costs:
- Moderate cost due to structured processes and clear planning.
- Avoids waste by focusing on delivering only necessary features.

→ Key Strengths:
- combines structure with flexibility
- Provides predictable timelines.

→ Challenges:
- Less effective in environments with rapidly changing requirements.