Software Engineering

Assignment

Question:- Different Agile Approaches and their comparive. Analysis.

Ans:-

Agile:- In Software Engineering, Agile refers to a development methodology based on iterative and incremental process that enphasize flexibility, collaboration and customer centric approaches.

1) Scrum:-

How it works:-

- Iterative and incremental development with short sprints (usually 2-4 weeks)
- Roles:- Product owner, Scrum master, Development Team.
- key events:- Sprint planning, Daily Stand-ups

Sprint · Reviews, Retrospectives ·

- Artifacts:- Product Backlog, Sprint · Backlog, Int: Increment.

Applicability:-

- Suitable for project with - defined roles and deliverables.
- Commonly used in Software development and product - focused industries.
- works best for Small to medium sized teams

Effectiveness in Terms of Costs:-

- Cost-effect due to focused Sprints and reduced wastage,
- Continuous delivery reduced risk of melon financial Setbacks.

Example :-

- A team developing an e-commerce platform uses Scrum to delivery feature

like a shopping cart, product search, and payment integration incrementally.

2) Kanban:-

How it works:-

- Focuses on visualizing workflows and limiting work in progress (WIP)
- uses a kanban board with columns like To Do, In progress and Done,
- Continous Delivery ; no fixed timboxs.

Applications:-

- Best for operations and maintainance project or ongoing support.
- Effective where work priorities frequently change.

Effectiveness in Terms of cost

- Minimal overhead cost.
- improve workflow efficiency, reducing waste.

Example:-

- A Team managing IT support tickets visualizes incoming tasks on a kanban board to prioritize and resolve issue.

3) Extreem programming (XP):-

How it works:-

- Emphasizes technical practices like Test Driven Development (TDD), Continuous Integration (CI) and pair programming.
- Short Iteration with frequent releases.
- Customer involvement is integral.

Application:-

- Ideal for projects requiring high-quality code and rapid changes.
- Common in startups or environments with rapidly evolving requirements.

Effectiveness in terms of costs

- Initial costs may be higher due to pair programming and Testing.
- Long-Term Savings due to reduced defects and maintenance.

Example:-

- A financial Software project where quality and accuracy are critical employs XP to ensure rebust and reliable code.

4) Learn Software Development:-

How it works:-

- Focuses on eliminating waste and delivering ~~vales~~ valve.
- principles: Build quality in, amplify learning, defer commitment, deliver quickly, respect people, and optimize the whole.

- Encourages first feedback and decision-making.

Applicability:-

- Effective in startups or innovation driven environments,
- works well for cross functional teams,

Effectiveness in Term of Costs :-

- Reduces costs by eliminating non-value-adding activities,
- Helps control budgets through efficient resource utilization.

Example:-

- A startup building a minimum product (MVP) uses Lean principles to prioritize essential features.

5) Feature Driven Development(FDD):-

How it works:-

- Focuses on designing and building features iteratively,
- Steps: Develop overall model, build feature list, plan by feature, design by feature, and build by feature.
- Works on delivering tangible, client-valued results.

Applicability:-

- Suitable for large projects with complex systems,
- work well in structured environments with clean requirements,

Effectiveness in Terms of costs:-

- costs can be higher initially due to detailed planning,
- long term savings due to structured ~~feature~~

Bindu
IT-21003

Feature delivery.

Example:-

- A telecom project uses FDD to deliver individual network features like call routing and data Streaming.