Question: Different Agile Approaches and their comparaive Analysis.

Answer: In software engineering, Agile refers to a development methodology based on iterative and incremental process that emphasize flexibility, collaboration and customer centric approaches.

① Scrum:

↳ How it works:

↳ Iterative and incremental development with short spirints (usually 2-4 weeks)

↳ Roles: Product owner, srum master, Development team.

↳ key events: sprint planning, Daily stand-ups, sprint Reviews, Retrospectives.

↳ 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 major financial setbacks.

Example: A team developing an e-commerce platform uses scrum to delivery features, 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.
  ↳ Continuous Delivery, no fixed timboxes.
↳ Applications:
  ↳ Best for operations and maintainance project or ongoing support.
  ↳ Effective where work priorities frequently change.

↳ Effectiveness in terms of costs:

↳ Minimal overhead cost.

↳ Improve workflow efficiency, reducing waste.

Example: A team managing IT support tickets visualize, 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 pain 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 requirments.

↳ Effectiveness in terms of costs:

↳ Initial costs may be higher due to pair programming and testing.

↳ Long-Term savings due to reduced defects and maintanance.

Example: A financial software project where quality and accuracy are critical employs XP to ensure robust and reliable code.