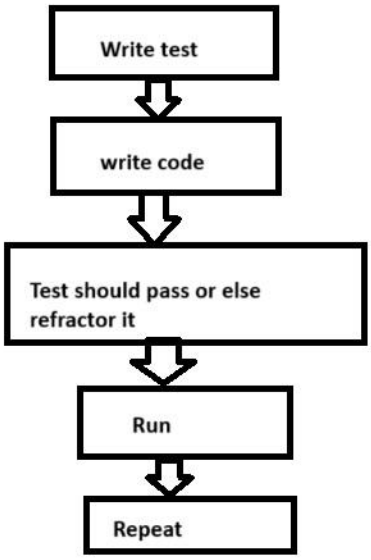
DAY-3 05-06-2024

Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability

Flowchart of Test-Driven Development:



TDD- Test Driven Development:

- First write a test
- Then write a code to pass the test.
- If not pass, refractor it
- Repeat the process, create the tests and pass it.
- Good quality code because test done until it passes
- Used for small projects
- Reduce bugs
- Writing tests first ensure better design.

Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

TDD (Test-Driven Development):

- **Approach:**
  i) Developers write tests first, then implement code to pass those tests.
  ii) Focuses on code correctness and incremental improvement.
  iii) Developer-centric.
- **Benefits:**
  i) Early Bug Detection: Identifies defects before code is fully developed.
  ii) Continuous Feedback: Provides insights for code improvement.
  iii) Disciplined Approach: Ensures reliable code.
- **Suitability:**
  i) Ideal for small, focused units of code (e.g., functions, methods).
  ii) Well-suited for stable requirements.

BDD (Behavior-Driven Development):

- **Approach:**
  i) Expresses system behavior using plain language specifications (Gherkin syntax).
  ii) Collaboration between developers, QA, and non-technical stakeholders.
  iii) User-centric**.**

- **Benefits:**
  i) Business Alignment: Focuses on meeting business needs and user requirements.
  ii) Collaboration: Involves stakeholders in defining behavior.
  iii) Behavior Clarity: Clear communication of system behavior.

- **Suitability:**
  i) Suited for user-facing features and end-to-end scenarios.
  ii) Effective for cross-functional teams.

FDD (Feature-Driven Development)

- **Approach:**
  i) Breaks develop into small, feature-focused iterations.
  ii) Emphasizes feature modeling and design patterns.
  iii) Incremental and iterative.

- **Benefits:**
  i) Feature-Centric: Aligns development with business features.
  ii) Predictable Progress: Clear milestones for feature completion.
  iii) Efficient Collaboration: Teams work on specific features.

- **Suitability:**
  i) Best for large-scale projects with multiple features.
  ii) Effective for complex systems.