ASSIGNMENT- 03

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.

TEST DRIVEN DEVELOPMENT PROCESS:

- **Write Test Cases** :
  - Define specific test cases for each functionality.
  - Tests should cover all possible scenarios.
- Write Code to Pass Test :
  - Develop the minimum code required to pass the tests.
  - Focus on functionality, not optimization
- Run Tests :
  - Execute all test cases against the code.
  - Ensure that all tests pass successfully.
- Refactor Code :
  - Optimize and improve the code without changing its functionality.
  - Enhance readability, maintainability, and performance.
- Repeat The Cycle :
  - Write new test cases or modify existing ones for added functionalities.
  - Follow the same process iteratively for continuous improvement.

Benefits of TDD :

- Bug Reduction :
  - Early detection of bugs through automated testing.
  - Addresses issues before they propagate into the codebase.
- Improved Reliability :
  - Rigorous testing ensures reliable and stable software.
  - Enhances confidence in the codebase's correctness.

This infographic visualizes the iterative nature of Test-Driven Development, emphasizing its focus on writing tests before code, the benefits of bug reduction, and its contribution to software reliability.

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.

TEST DRIVEN DEVELOPMENT :

Approach :

- Write Tests befor writing code
- Red-Green-Refactor cycle
- Focus on small, incremental development.

Benefits :

- Early bug detection.
- Improved code quality.
- Ensures test coverage.
- Supports Agile development.

Suitability :

- Well-suited for Agile and iterative development.
- Effective for projects with changing requirements.
- Ideal for maintaining clean and maintainable codebases.

BEHAVIOR-DRIVEN DEVELOPMENT:

Approach :

- Define behavior using human-readable scenarios (Given-When-Then).
- Collaborative approach involving developers, testers, and business stakeholders.
- Tests serve as specifications and documentation.

Benefits :

- Improved communication between stakeholders.
- Encourages collaboration.
- Focuses on user needs.
- Facilitates automated testing.

Suitability :

- Ideal for projects with complex business logic.
- Useful for teams with diverse skill sets.
- Effective for ensuring alignment between development and business goals.

FEATURE-DRIVEN DEVELOPMENT :

Approach :

- Develop features iteratively.

- Emphasizes domain object modeling
- Focuses on delivering tangible results.

Benefits :

- Clear project structure and organization.
- Efficient utilization of resources.
- Emphasizes client value.
- Supports scalability.

Suitability :

- Suitable for large-scale projects.
- Ideal for teams with strong technical expertise.
- Effective for projects with well-defined requirements.