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.**

Write a Test

↓

Run the Test

and See it Fail

↓

Write the Code

to Make the Test

Pass

↓

Run the Test

and See it Pass

↓

Refactor the Code

to Make it Better

↓

Repeat the Cycle

for the Next Test

**Test-Driven Development (TDD) Process :**

**Step 1**: Write a Test

Write a test that covers a specific piece of functionality in your code This test should be independent of the implementation details.

**Step 2**: Run the Test and See it Fail

Run the test and see it fail because the code hasn't been written yet.

**Step 3**: Write the Code

Write the minimal amount of code necessary to pass the test Focus on making the test pass, not on writing perfect code.

**Step 4**: Run the Test and See it Pass

Run the test and see it pass because the code has been written.

**Step 5**: Refactor the Code

Refactor the code to make it maintainable, efficient, and easy to understand Remove any duplication, simplify the code, and make it more modular.

**Advantages of TDD**

**Bug Reduction**

Writing tests before code helps catch bugs early in the development process.

Reduces the number of bugs in the codebase.

**Software Reliability**

TDD ensures that the code is reliable and works as expected.

Gives developers confidence in the code they write.

**Improved Code Quality**

TDD promotes writing clean, modular, and maintainable code.

Encourages developers to think about the design and architecture of the code.

**Faster Development**

TDD helps developers write code faster because they know exactly what they need to implement.

Reduces the time spent on debugging and fixing bugs.

**Better Design**

TDD promotes good design principles, such as loose coupling and high cohesion.

Encourages developers to think about the interfaces and APIs of the code.

**Reduced Fear of Change**

TDD gives developers the confidence to make changes to the code without fear of breaking it.

Encourages developers to refactor and improve the codebase over time.

By following the TDD process, developers can write high-quality code that is reliable, maintainable, and efficient. TDD helps reduce bugs, improves code quality, and promotes good design principles, making it an essential practice for any software development team.

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.**

**Methodology Comparison**

**TDD (Test-Driven Development)**

Write automated tests write code to pass tests refactor code for quality.

**Benefits**

Reduces defects and bugs improves code quality and maintainability fast feedback and early detection of issues.

**Suitability**

Small to medium-sized projects well-defined requirements agile development methodologies.

**TDD Cycle**

+---------------+

| Write Test |

+---------------+

|

|

v

+---------------+

| Write Code |

+---------------+

|

|

v

+---------------+

| Refactor |

+---------------+

**BDD (Behavior-Driven Development)**

Describe system behavior write automated tests develop code to implement behavior.

**Benefits**

Improves communication and collaboration reduces misunderstandings and errors delivers user-centric software.

**Suitability**

Complex projects with multiple stakeholders projects with rapidly changing requirements agile development methodologies.

**BDD Flow**

+---------------+

| Describe |

| Behavior |

+---------------+

|

|

v

+---------------+

| Write Test |

+---------------+

|

|

v

+---------------+

| Develop Code|

+---------------+

**FDD (Feature-Driven Development)**

Break down project into features develop each feature in iterations involve clients in development process.

**Benefits**

Delivers value to clients quickly reduces scope creep and project delays improves collaboration and communication.

**Suitability**

Large and complex projects projects with rapidly changing requirements client-involved development methodologies.

**FDD Process**

+---------------+

| Break Down |

| Project into |

| Features |

+---------------+

|

|

v

+---------------+

| Develop |

| Feature in |

| Iterations |

+---------------+

|

|

v

+---------------+

| Involve Client|

+---------------+