**ASSIGMNENT karthikavula76@gmail.com**

**Topic: Modern Development Methodologies**

1.Create an info graphic 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.

Ans:

Test-Driven Development (TDD) Process



* **Write Test Cases**: Developers write test cases based on requirements before writing any code.
* **Run Tests**: Run the tests, ensuring they all fail initially because there's no corresponding code.
* **Write Code**: Write the minimum amount of code necessary to pass the tests.
* **Run Tests Again**: Rerun the tests. They should all pass now because of the code written in the previous step.
* **Refactor Code**: Refactor the code for better design, performance, or readability, while ensuring all tests still pass.

Benefits of Test Driven Development (TDD):

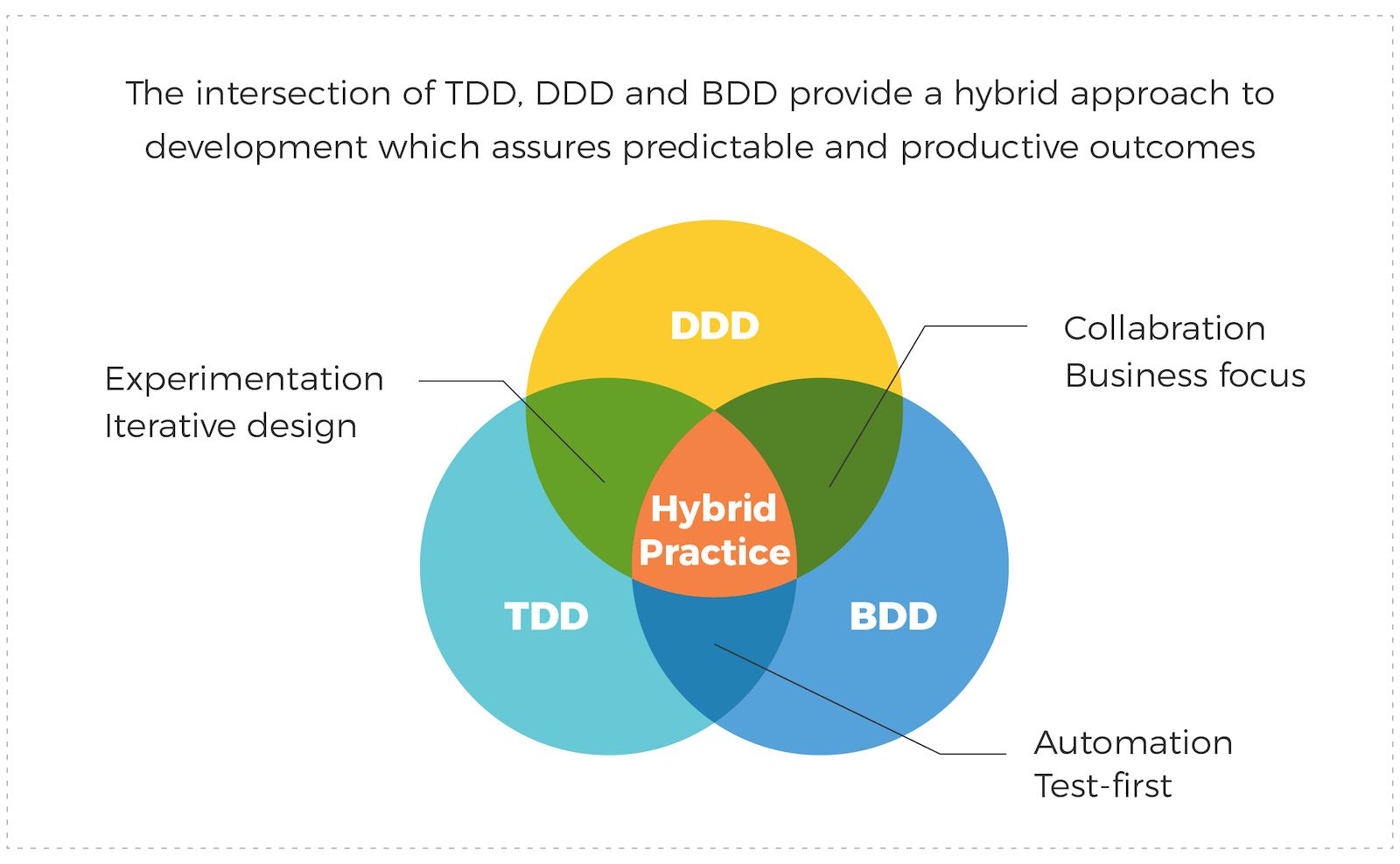
Bug Reduction: By catching issues early, TDD reduces the number of bugs in the final product.

Fosters Reliability: Since tests are written before code, TDD helps ensure that the code meets the specified requirements, leading to more reliable software.

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

Ans

Let’s delve into a comparison of **Test-Driven Development (TDD), Behavior-Driven Development (BDD),** and **Feature-Driven Development (FDD)**methodologies**.**



1. **Test-Driven Development (TDD)**:

* **Approach**: In TDD, developers write automated tests **before** writing the actual code.
* **Focus**: It emphasizes code correctness and ensures that the code meets the expected behavior defined by the tests.

**Process**:

* **Create Tests**: Developers define the expected behavior of a small unit of code through test cases.
* **Execute Specific Test**: Run the test (which initially fails) to confirm its reliability.
* **Write Code**: Develop the code to make the test pass incrementally.

**Benefits**:

* Continuous feedback for faster bug identification.
* Improved code quality.

**Suitability**:

* Well-suited for developer-centric teams.
* Ideal for small, co-located teams.

1. **Behavior-Driven Development (BDD)**:

* **Approach**: BDD focuses on system behavior and collaboration between stakeholders.

**Process**:

* **Express Behavior**: Stakeholders (including testers and users) express system behavior using Gherkin syntax.
* **Code Implementation**: Developers write code based on these Gherkin expressions.

**Benefits**:

* User-centric approach.
* Promotes collaboration.

**Suitability**:

* Suitable for cross-functional teams.
* Ideal for scenarios where stakeholders’ input matters.

1. **Feature-Driven Development (FDD)**:

* **Approach**: FDD focuses on planning, tracking progress, and delivering features.

**Process**:

* **Identify Features**: Break down the system into features.
* **Design by Feature**: Develop a design for each feature.
* **Build by Feature**: Implement features incrementally.

**Benefits**:

* Clear feature-based progress tracking.
* Efficient for large projects.

**Suitability**:

* Well-suited for large-scale projects.
* Ideal when features need to be delivered iteratively.