

## Comparative infographic of TDD, BDD, and FDD methodologies:

### Infographic

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

**Approach:** Write tests before writing code

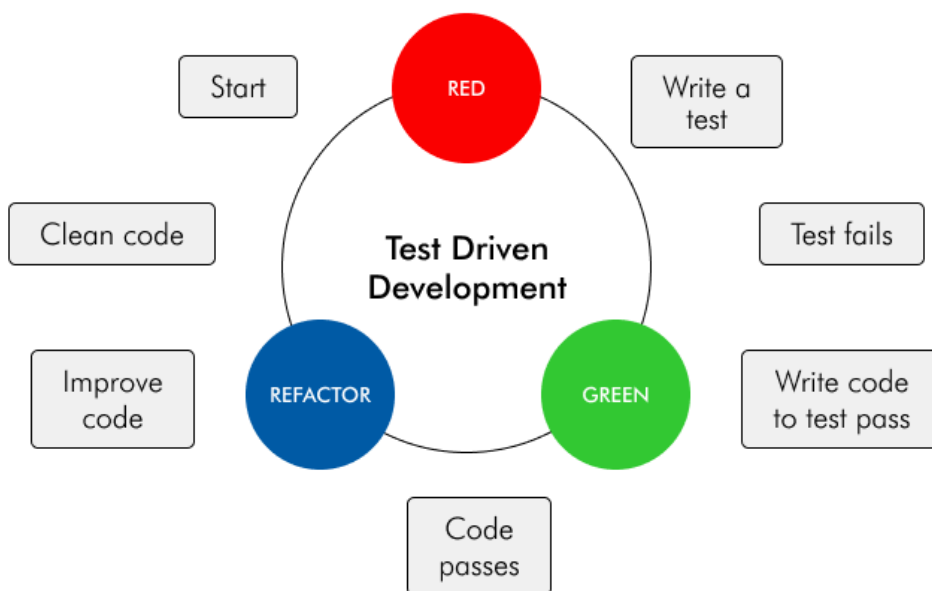
**Benefits:**

- Ensures code quality and reliability
- Catches bugs early in development
- Encourages simple and focused code

**Suitable for:**

1. Small to medium-sized projects
2. Teams with experienced developers
3. Projects with well-defined requirements

#### Flow Chart / Diagram:



## 2. **BDD (Behavior-Driven Development)**

**Approach:** Write behavior-based tests before writing code.

### **Benefits:**

- Improves communication between developers and non-technical stakeholders
- Ensures code meets business requirements
- Encourages collaborative development

### **Suitable for:**

1. Medium to large-sized projects
2. Teams with diverse skill levels
3. Projects with complex business requirements

### **Flowchart / Diagram:**



### 3. FDD (Feature-Driven Development)

**Approach:** Deliver functional features to end-users

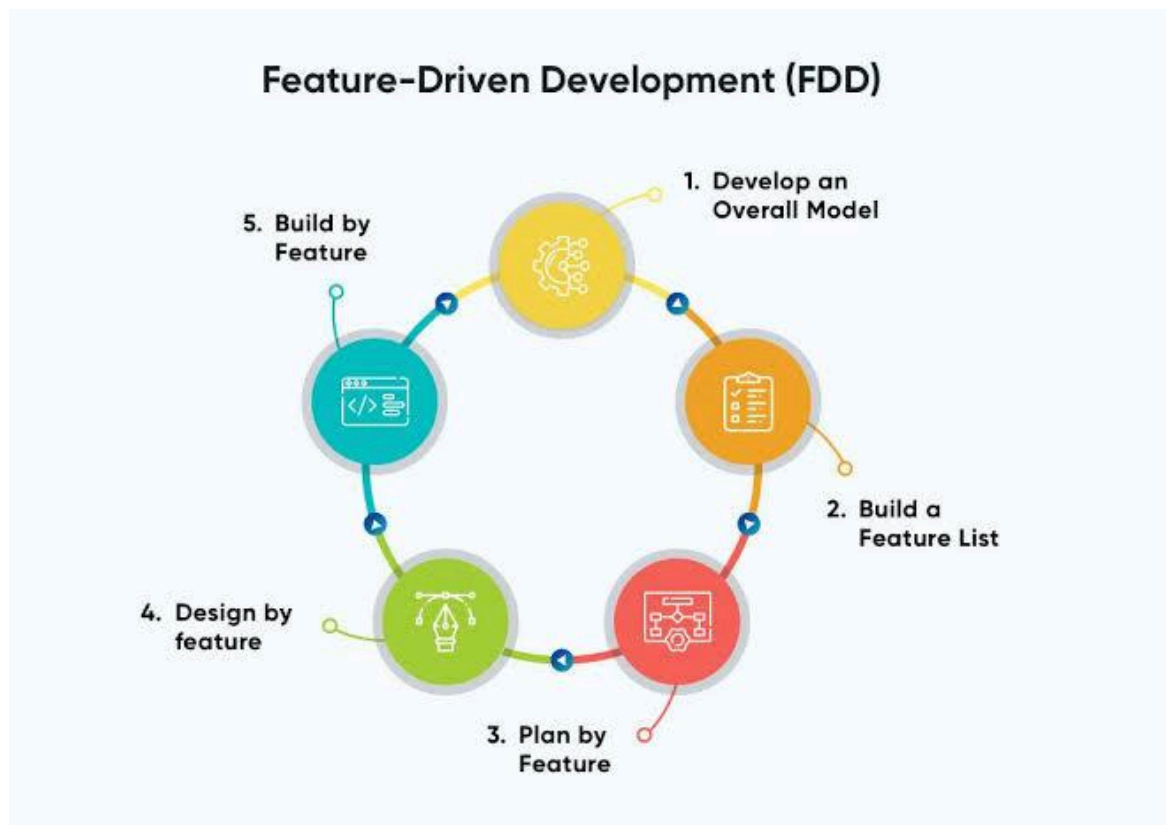
**Benefits:**

- Focuses on delivering working software
- Encourages collaboration and prioritization
- Improves project visibility and tracking

**Suitable for:**

1. Large and complex projects
2. Teams with varying skill levels
3. Projects with rapidly changing requirements

**Flowchart / Diagram:**



## **Comparison**

- TDD focuses on code quality, BDD on behavior, and FDD on features
- TDD and BDD emphasize testing, while FDD prioritizes delivery
- BDD and FDD involve non-technical stakeholders, while TDD is developer-centric