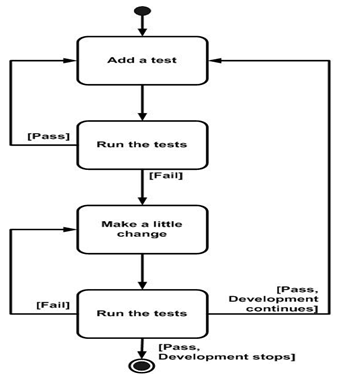
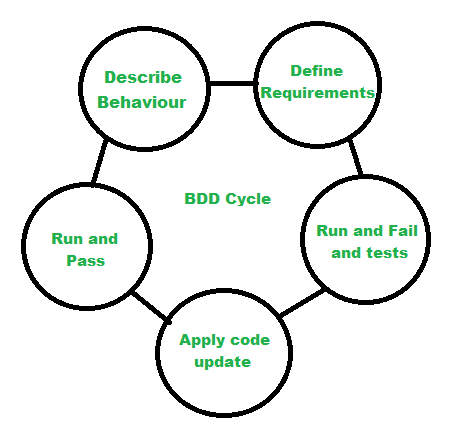
**TDD (Test-Driven Development):**

* **Approach**: Developers write tests before they write the functional code. The code is then written to pass these tests.
* **Benefits**:
  + Ensures code is thoroughly tested.
  + Forces developers to think about requirements upfront.
  + Reduces bugs and defects early in the development cycle.
* **Suitability**: Best suited for Agile environments and projects where requirements are likely to change.
* **Flow chart**:



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

* **Approach**: Focuses on the behavior of the system from the end user's perspective. Tests are written in a natural language format (like Gherkin) to describe system behaviors.
* **Benefits**:
  + Improves collaboration between developers, QA, and business stakeholders.
  + Enhances clarity and understanding of requirements.
  + Encourages a customer-centric approach to development.
* **Suitability**: Ideal for projects with complex business logic or where there's a need for clear communication between technical and non-technical team members.
* **FlowChart**:



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

* **Approach**: Breaks down development into small, manageable features. Each feature is developed individually and then integrated into the system.
* **Benefits**:
  + Focuses on delivering tangible, working features incrementally.
  + Provides a structured approach to development with clear roles and responsibilities.
  + Emphasizes on collaboration and communication within the team.
* **Suitability**: Well-suited for large-scale projects with multiple teams, where features can be developed and integrated independently.
* **FlowChart:**

