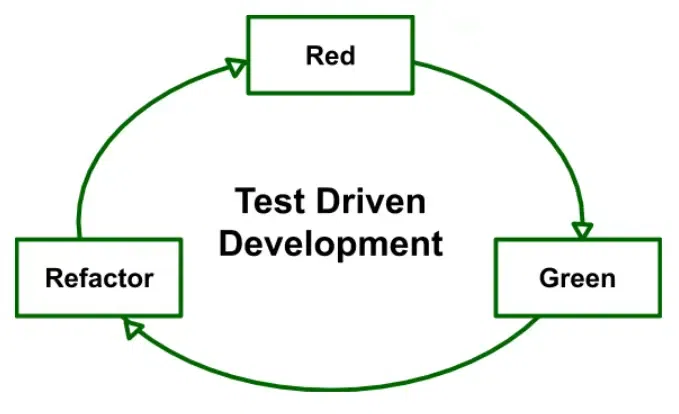
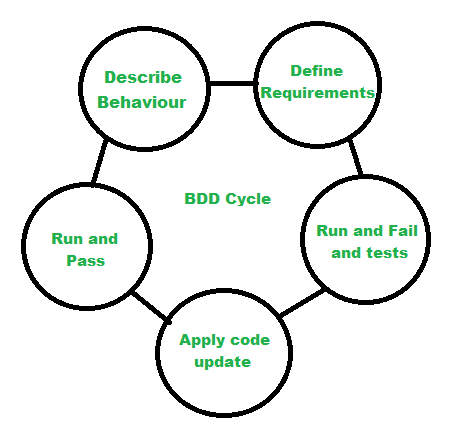
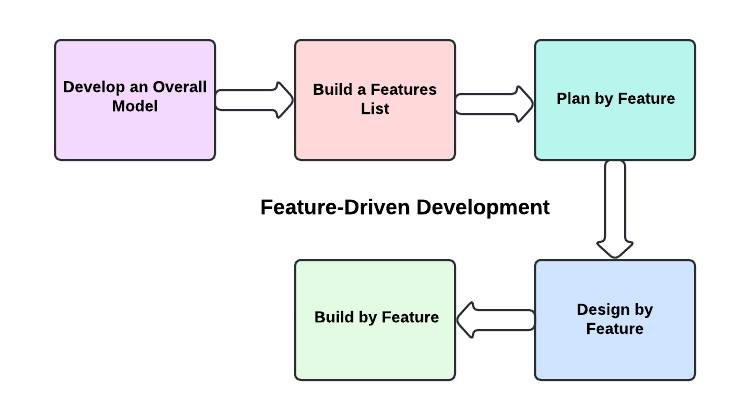
**Assignment No 3 [2]**

**Question:** 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.

**Solution:**

****

****



|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **TDD** | **BDD** | **FDD** |
| **Approach** | 1. **Write a Test:** Define test cases before writing code. 2. **Run the Test:** Ensure the test fails initially. 3. **Write Code:** Develop code to pass the test. 4. **Run All Tests:** Verify all tests pass. 5. **Refactor:** Improve code quality.   Repeat the cycle for each functionality. | 1. **Describe behavior –**This includes the flow and features of the product means the main vision. 2. **Define requirements –**Modeled requirements with business rules for a shared understanding. 3. **Run and fail the tests –**Develop and run the test cases. 4. **Apply code update –**Refactor it according to the requirement. 5. **Run and pass the tests –**Run the updated code and pass the test cases. | 1. **Develop Overall Model:** Create a high-level model of the system. 2. **Build Feature List:** Identify and prioritize features. 3. **Plan by Feature:** Develop a plan for feature implementation. 4. **Design by Feature:** Design each feature in detail. 5. **Build by Feature:** Implement and test each feature. |
| **Benefits** | 1. Early bug detection. 2. Ensures code reliability. 3. Improved design and documentation. 4. Confidence in refactoring. | 1. Enhances collaboration between stakeholders. 2. Clear communication of requirements. 3. Focus on user experience. 4. Detailed documentation of features. | 1. Scalable and adaptable for large teams. 2. Focus on delivering tangible features. 3. Encourages iterative development. 4. Clear tracking of progress. |
| **Suitability** | 1. Projects requiring high reliability. 2. Continuous integration environments. 3. Developers focusing on code correctness. | 1. Agile and Scrum environments. 2. Projects requiring frequent stakeholder interaction. 3. Teams emphasizing user-centric design. | 1. Large-scale projects. 2. Teams with a clear feature-based approach. 3. Environments needing detailed planning and tracking. |