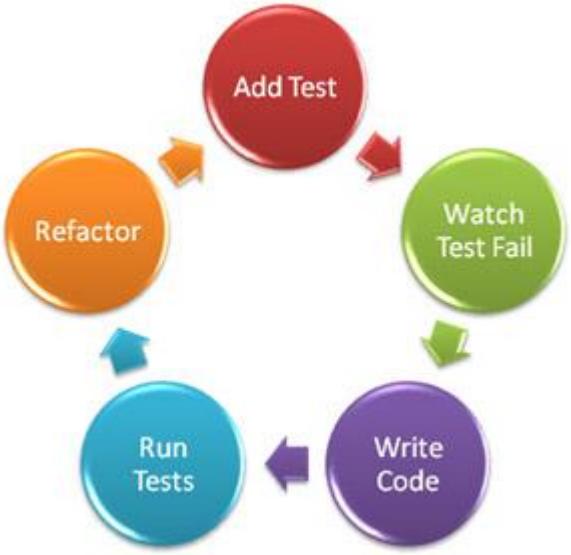
Assignment 2: Produce a comparative infographic of TDD, BDD, and FDD methodologies. Illustrate their unique approaches, benefits, and suitability for different software development contexts. Use visuals to enhance understanding.

Comparative Infographic: TDD vs BDD vs FDD

Title: Comparative Analysis of Software Development Methodologies



1. Introduction:

- Brief overview of TDD, BDD, and FDD methodologies in software development.
- Each methodology—TDD, BDD, and FDD—offers distinct approaches and benefits tailored to different software development contexts.
- Understanding their unique characteristics and suitability can help teams choose the most appropriate methodology based on project requirements, team dynamics, and desired outcomes.
- Whether focusing on early bug detection and code quality (TDD), fostering collaboration and user-centric development (BDD), or managing feature-driven development in large-scale projects (FDD), each methodology contributes to successful software delivery in its own way.

2. Test-Driven Development (TDD):

Approach:

- Write tests before writing code.
- Tests define the desired functionality.

Benefits:

- Early bug detection.
- Improved code quality.
- Software reliability through comprehensive testing.

Suitability:

- Agile environments.
- Projects with evolving requirements.

Visual: Icon of a test case before code.

3. Behaviour-Driven Development (BDD):

Approach:

- Focus on behaviour and user stories.
- Tests are written in a natural language format (Given-When-Then).

Benefits:

- Enhanced collaboration between developers, testers, and stakeholders.
- Improved communication of requirements.

Suitability:

- Collaborative environments.
- Projects requiring clear understanding of user behaviour and acceptance criteria.

Visual: Icon of a scenario in Given-When-Then format.

4. Feature-Driven Development (FDD):

Approach:

- Emphasizes iterative and incremental development of features.
- Features are defined, designed, and built in short iterations.

Benefits:

- Scalability for large projects.
- Clear focus on feature delivery.

Suitability:

- Large-scale projects with multiple teams.
- Projects requiring structured development phases.

Visual: Icon of feature modules being developed iteratively.

5. Comparative Analysis:

a. Methodology Focus:

- TDD focuses on unit testing and code quality.
- BDD emphasizes behaviour and collaboration.
- FDD prioritizes feature delivery and scalability.

b. Testing Approach:

- TDD: Tests at the unit level.
- BDD: Tests at the behaviour and acceptance level.
- FDD: Tests at the feature level.

c. Stakeholder Involvement:

- TDD and BDD involve stakeholders through tests and behaviour scenarios.
- FDD involves stakeholders through feature definition and delivery.

6. Conclusion:

- Summary of key differences and considerations when choosing between TDD, BDD, and FDD based on project requirements and team dynamics.

7. References:

- Include sources or further reading materials on TDD, BDD, and FDD methodologies.

Design Elements:

- Use icons or illustrations to represent each methodology's unique approach.
- Employ colour coding or sections to differentiate between TDD, BDD, and FDD.
- Include charts or graphs to visualize benefits or suitability for different contexts.

Visual Style:

- Choose a clean and organized layout to enhance readability.
- Use a consistent colour scheme that distinguishes between methodologies.
- Incorporate icons or symbols related to testing, behaviour, and feature development.

Tools:

- Design software like Adobe Illustrator, Canva, or PowerPoint can be used to create the infographic.
- Ensure clarity and simplicity in design elements to effectively convey information.