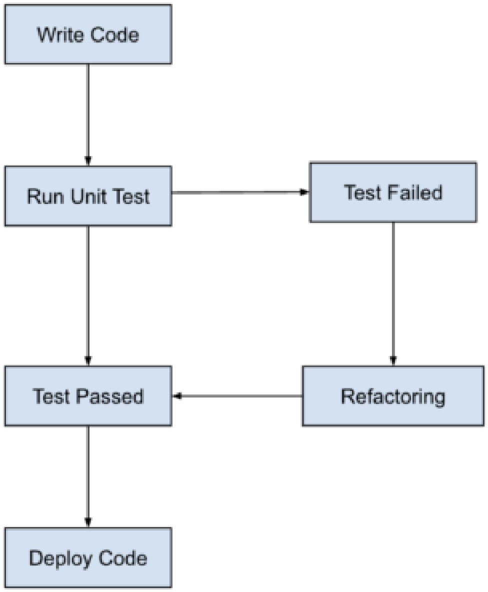
Assignment 1: Create an infographic illustrating the Test-Driven Development (TDD) process.

Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.

The Test-Driven Development (TDD) Process



Write Failing Tests (Test First):

Depict a programmer writing code for an automated test.

Show the test initially failing (marked with an "X").

Code to Make Tests Pass (Just Enough):

Show the programmer carefully coding to fulfill the failing test requirements.

Refactor (Clean Up): The programmer reviewing and improving the code for better readability and maintainability.

Benefits:

Reduced Bugs:

On the side of the infographic, showcase a magnifying glass finding and eliminating bugs (represented as insects).

Improved Code Quality:

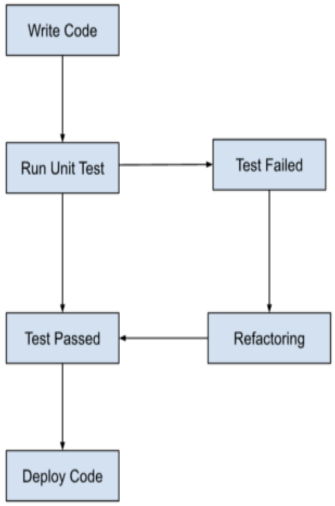
Depict a code snippet with clean structure and clear comments.

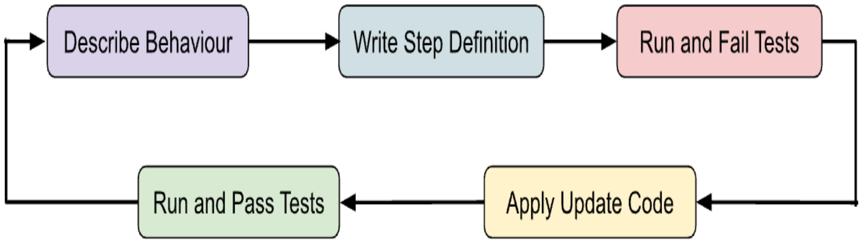
Reliable Software:

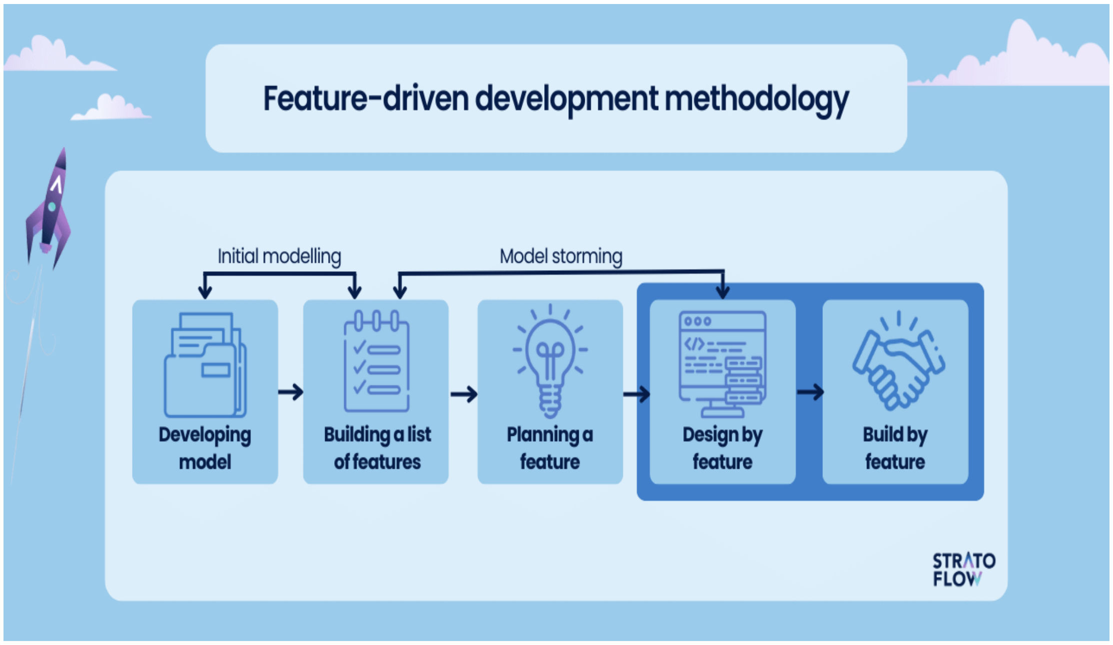
Show the completed castle/building structure standing strong (representing reliable software built using TDD).

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.







Test-Driven Development (TDD):

Approach: Write failing tests before writing code.

Benefits:

Bug Reduction

Reliability

Suitable Context: Agile environments with iterative development.

Behavior-Driven Development (BDD):

Approach: Define behavior using natural language specifications.

Benefits:

Improved Communication

Alignment with Business Goals

Suitable Context: Complex projects with diverse stakeholder requirements.

Feature-Driven Development (FDD):

Approach: Focus on delivering specific features in short cycles.

Benefits:

Scalability

Clear Progress Tracking

Suitable Context: Projects with clear, predefined requirements.

Introduction:

Brief explanation of each methodology: Test-Driven Development (TDD), Behavior-Driven Development (BDD), and Feature-Driven Development (FDD).

TDD (Test-Driven Development):

Approach:

Illustration of a developer writing a failing test before writing code.

Caption: "Write failing tests before writing code to define functionality."

Benefits:

Bug Reduction:

Illustration showing bugs caught early in the development process.

Caption: "Identify and fix bugs early, reducing overall development time."

Reliability:

Illustration of a sturdy bridge representing reliable software.

Caption: "Build reliable software with a comprehensive test suite."

Suitable Context:

Illustration of a small project with frequent code changes.

Caption: "Ideal for agile environments with iterative development."

BDD (Behavior-Driven Development):

Approach:

Illustration of collaboration between developers, testers, and stakeholders to define behavior using natural language.

Caption: "Define behavior using natural language specifications."

Benefits:

Improved Communication:

Illustration of a conversation bubble connecting stakeholders and developers.

Caption: "Enhanced communication between stakeholders and development teams."

Alignment with Business Goals:

Illustration of a target symbolizing alignment with business objectives.

Caption: "Ensure software development aligns with business objectives."

Suitable Context:

Illustration of a large project with multiple stakeholders.

Caption: "Well-suited for complex projects with diverse stakeholder requirements."

FDD (Feature-Driven Development):

Approach:

Illustration of breaking down development into manageable feature sets.

Caption: "Focus on delivering specific features in short cycles."

Benefits:

Scalability:

Illustration showing the growth of features over time.

Caption: "Easily scale development to accommodate project growth."

Clear Progress Tracking:

Illustration of a progress bar indicating completion of features.

Caption: "Track progress by focusing on feature completion."

Suitable Context:

Illustration of a project with a well-defined scope and timeline.

Caption: "Ideal for projects with clear, predefined requirements."

Conclusion:

Summarize the unique approaches, benefits, and suitability of each methodology.

Encourage consideration of project requirements and team dynamics when choosing a methodology.