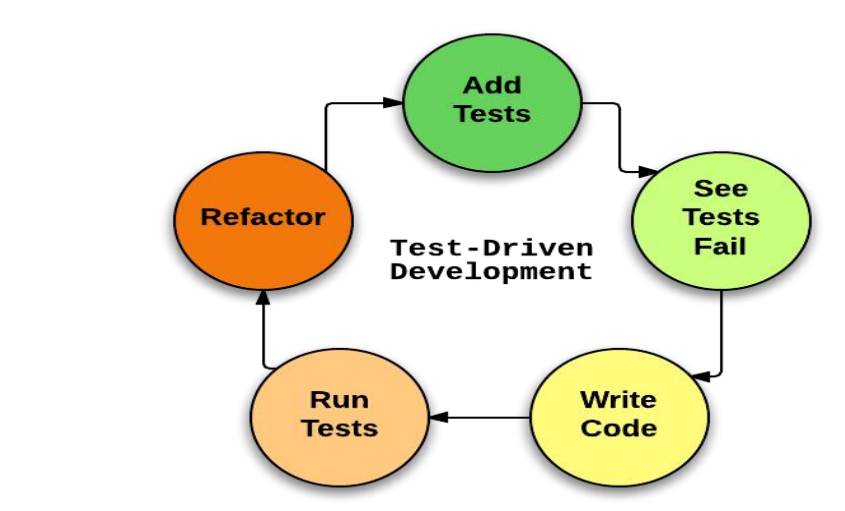
**Create an infographic illustrating the Test-Driven Development (TDD) process. Highlight steps like writing tests before code, benefits such as bug reduction, and how it fosters software reliability.**

**Test-Driven Development (TDD) Process Infographic**




**Write Test**:

**Step**: Write a failing test before writing any production code.
**Action**: Developer creates a test case for the desired functionality.

**Run Test**:

**Step**: Run the test to ensure it fails.
**Action**: Execute the test, expecting it to fail due to absence of the feature.

**Write Code**:

**Step**: Write the minimum code to pass the test.
**Action**: Developer writes the simplest code possible to make the test pass.

**Run Test Again**:

**Step**: Re-run the test suite.
**Action**: Ensure the newly written code passes the test.

**Refactor Code**:

**Step**: Refactor the code while ensuring the test still passes.
**Action**: Improve the code without altering its functionality, maintaining the passing test.

**Repeat**:

**Step**: Iterate the process for each new feature or enhancement.
**Action**: Continue adding tests, writing code, and refactoring iteratively.

## Benefits of Test-Driven Development (TDD):

**Bug Reduction**:Catching bugs earlier in the development process reduces the likelihood of them manifesting in production.

**Improved Code Quality**:By focusing on writing tests first, developers arecompelled to write cleaner, modular, and well-structured code.

**Enhanced Software Reliability**:Continuous testing ensures that the software behaves as expected, leading to higher reliability and fewer unexpected behaviors.

**Faster Development**:Despite the initial investment in writing tests, TDD often results in faster development cycles due to reduced debugging time and better code design.

**Early Detection of Design Flaws**:Writing tests upfront often exposes design flaws or architectural issues early in the development process, allowing for timely adjustments.

## Disadvantages of Test-Driven Development (TDD):

**Initial Learning Curve**: Adopting TDD requires developers to learn new techniques and practices, which can initially slow down the development process.

**Time-Consuming**: Writing tests before writing code can be time-consuming, especially for complex projects. This upfront investment in testing may seem to slow down development initially.

**Overhead Maintenance**: Maintaining a suite of tests requires additional effort. As the codebase evolves, tests need to be updated accordingly, adding overhead to the development process.

# TASK

| Test-Driven Development (TDD) | Behavior-Driven Development (BDD) |
|---|---|
| Implementation details of code | System behavior from user perspective |
| Write tests before writing code, focus on unit tests | Write human-readable behavior scenarios before code |
| Uses programming languages' built-in testing frameworks | Utilizes domain-specific language (DSL) for tests |
| Tests focus on code functionality and logic | Tests focus on system behavior and user interactions |
| Typically involves developers writing tests independently | Encourages collaboration between developers and stakeholders |
| Often automated using programming language test frameworks | Automation facilitated by BDD frameworks like Cucumber |

# TASK - 2

WRITE A FEATURE SCENARIO FOR LOGIN & TRANSFER FUNDS---------->

Given happy have his banking app installed and hir login credentials ready,
When he opens the app and enters hir username and password,
Then he should be logged in successfully.

Given happy is logged into his banking app,
When he selects the option to transfer funds,
Then he should be presented with choices for different transfer methods.

Given happy chooses to transfer funds to a friend,
When he enters hir friend's details and the amount to be transferred,
Then he should be prompted to confirm the transaction.

Given happy confirms the transaction,
When he provides any required additional authentication, like fingerprint or face ID,
Then the transaction should be processed securely.

Given the transaction is successfully authenticated,
When happy checks his transaction history or account balance,
Then he should see the transferred amount deducted from hir account and the transaction listed in hir history.

**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):

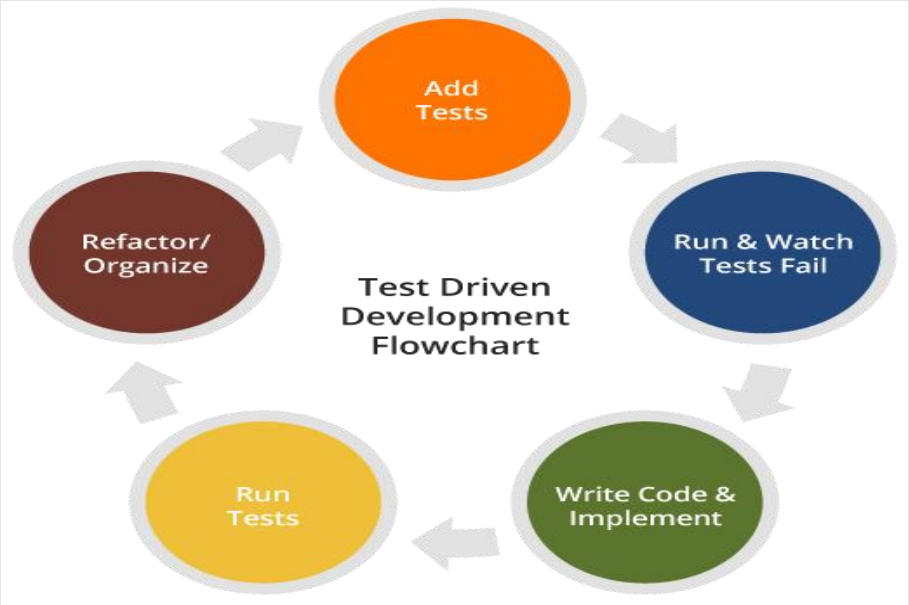
**Approach**: Test-first approach where tests are written before code.

**Benefits**:Ensures code is thoroughly tested from the beginning.

Improves code quality and design by promoting modular, testable code.

Provides a safety net for refactoring and code changes.

**Suitability**: Ideal for projects where code quality and reliability are paramount, such as safety-critical systems or libraries.




### Behavior-Driven Development (BDD):

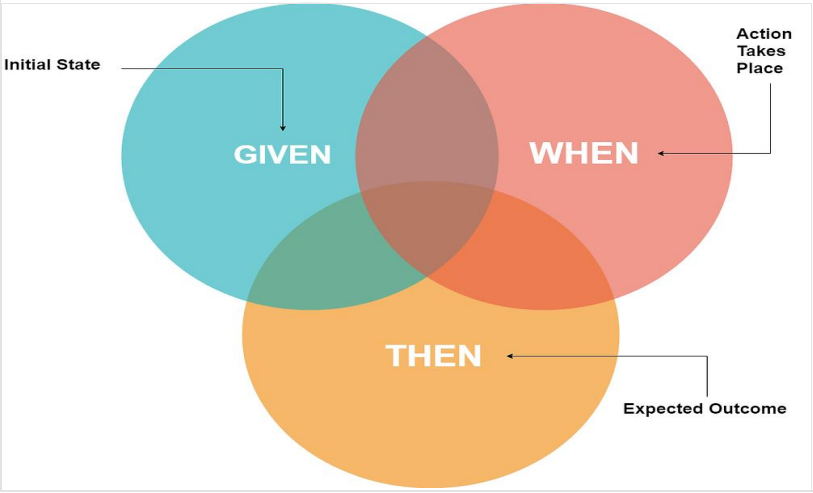
**Approach**: Focuses on defining behavior through scenarios written in a natural language format.

**Benefits**:Encourages collaboration between developers, testers, and business stakeholders.

Provides a common language for discussing system behavior.

Promotes a user-centric approach to development.

**Suitability**: Well-suited for projects with complex business requirements or where close collaboration with stakeholders is essential, such as enterprise applications or consumer-facing products.




## Feature-Driven Development (FDD):

**Approach**: Emphasizes building features incrementally based on domain modeling and feature lists.

**Benefits**:Provides a structured approach to development through feature breakdowns and iterative development cycles.

Focuses on delivering tangible, working features to users quickly.

Scales well for large teams and complex projects.

**Suitability**: Effective for large-scale projects with well-defined requirements and a need for rapid delivery, such as enterprise software or large-scale systems integration projects.