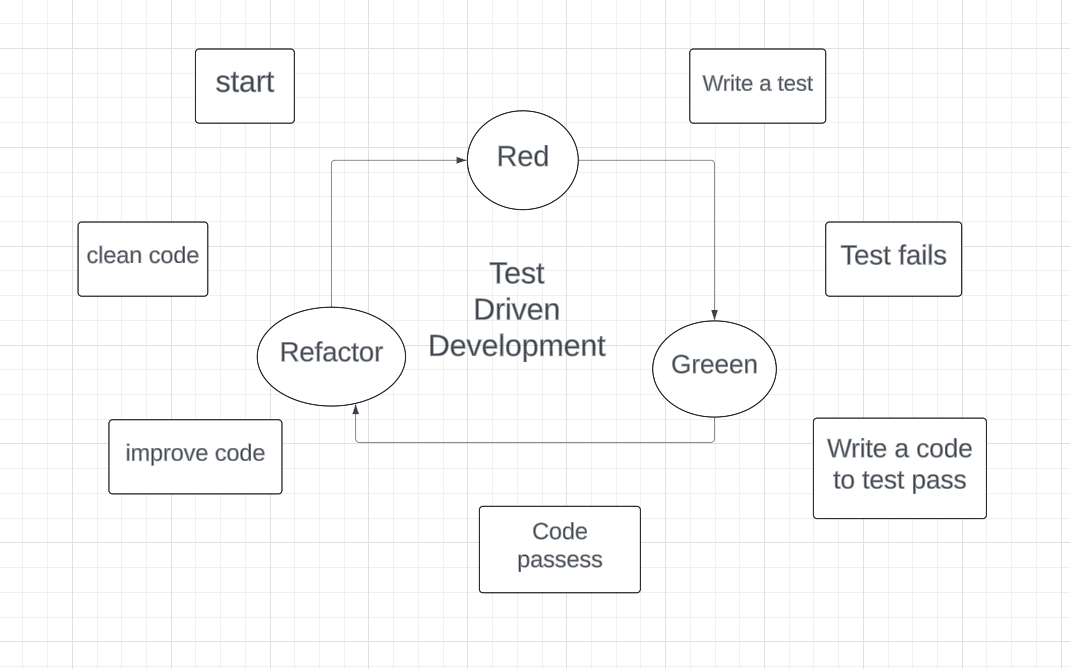
**Test Driven Development**

****

Test Driven Development (TDD) is a software development practice that focuses on creating unit test cases before developing the actual code. It is an iterative approach combining programming, unit test creation, and refactoring.

* The TDD approach originates from the Agile manifesto principles and Extreme programming.
* As the name suggests, the test process drives software development.
* Moreover, it’s a structuring practice that enables developers and testers to obtain optimized code that proves resilient in the long term.
* In TDD, developers create small test cases for every feature based on their initial understanding. The primary intention of this technique is to modify or write new code only if the tests fail. This prevents duplication of test scripts.

**Calculator Function**: When building a calculator function, a TDD approach would involve writing a test case for the “add” function and then writing the code for the process to pass that test. Once the “add” function is working correctly, additional test cases would be written for other functions such as “subtract”, “multiply” and “divide”.

Advantages

1. Fosters the creation of optimized code.
2. It helps developers better analyze and understand client requirements and request clarity when not adequately defined.
3. Adding and testing new functionalities become much easier in the latter stages of development.
4. Test coverage under TDD is much higher compared to conventional development models. The TDD focuses on creating tests for each functionality right from the beginning.
5. It enhances the productivity of the developer and leads to the development of a codebase that is flexible and easy to maintain.

**Business Driven Development**

Business-driven development (BDD) is a software development methodology that focuses on business needs over technology to create solutions that meet business requirements. BDD is an agile approach that emphasizes collaboration between developers, testers, and business stakeholders. It's designed for enterprise software, which is different from consumer-facing apps.

BDD is based on three principles:

* Focus on desired behavior or outcomes
* Collaborate between developers, testers, and business stakeholders
* Use a common language for communication and understanding

BDD builds on agile practices like sprint planning, user stories, and acceptance criteria, and makes them more effective. BDD tests are more user-focused and based on the system's behavior, and are written in a shared language to improve communication.

BDD is different from test-driven development (TDD) in what and how it tests. BDD focuses on the end user's perspective, while TDD focuses on smaller sections of functionality.

**Feature Driven Development**

Feature-Driven Development (FDD) is an Agile framework for software development that focuses on features that meet client needs. FDD is an iterative and incremental process that organizes work around features, rather than milestones or other progress indicators. Teams release features every 2–10 days, which allows them to analyze user feedback and improve the product experience.

FDD is a good choice for teams looking for a structured Agile methodology. It has several benefits, including: Focused scope, Early and frequent delivery, Improved communication, and Increased flexibility.