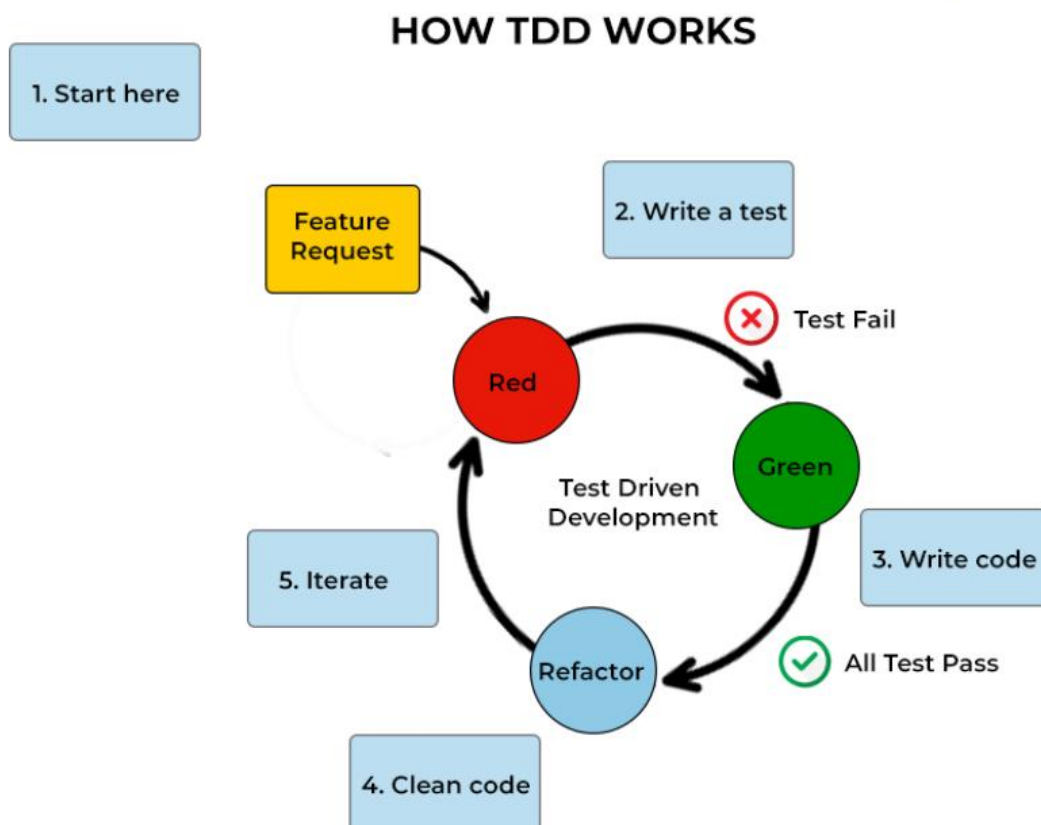


Test-Driven Development (TDD)

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.



Benefits of Test Driven Development (TDD)

- a. Fosters the creation of optimized code.
- b. It helps developers better analyze and understand client requirements and request clarity when not adequately defined.
- c. Adding and testing new functionalities become much easier in the latter stages of development.
- d. 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.
- e. It enhances the productivity of the developer and leads to the development of a codebase that is flexible and easy to maintain.

Conclusion

Test-Driven Development is a powerful methodology that emphasizes writing tests before code. It promotes bug reduction, improved design, faster development, and increased reliability, ultimately leading to higher-quality software products.

Visuals

You can add visuals like a flowchart showing the iterative nature of the process, icons representing each step, and graphs showing the correlation between TDD adoption and bug reduction or code quality improvement over time.