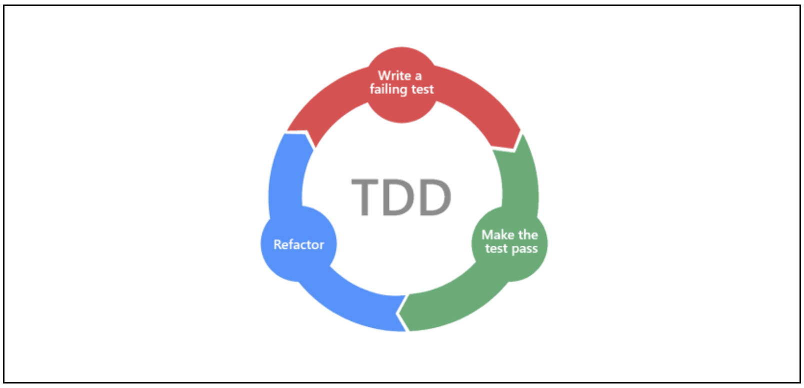
Development Models

Test-Driven Development

"*Write tests first*" is the key mantra of Test-Driven Development. The key concept is that a developer shouldn't write any code until they have written tests for that code first. Test-Driven Development has two goals:

1. To ensure that tests are actually written, and written well. Too often developers leave the design of tests until after the development process, and then don't bother because the code seems to work.
2. To help the developers envisage exactly what the code will do, and what processes and modules it will interact with, thus testing becomes part of the design process.



Test-Driven Development Cycle

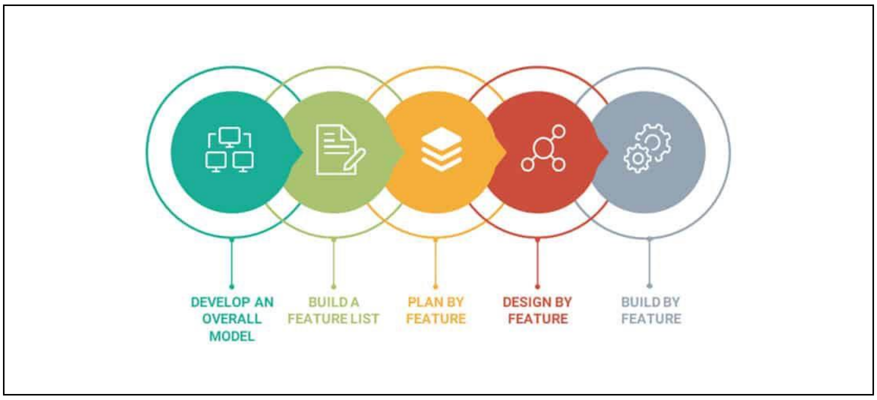
A typical test-driven development cycle is as follows:

1. *Add a test*: Each new feature begins with writing a test.
2. *Run all tests and see if the new test fails*: This validates that the test harness is working correctly.
3. *Write the code*: The next step is to write some code that causes the test to pass.
4. *Run tests*: If all test cases now pass, the programmer can be confident that the new code meets the test requirements.
5. *Refactor code*: The growing code base must be cleaned up regularly during test-driven development.
6. *Repeat*: Starting with another new test, the cycle is then repeated to push forward the functionality.

Test-driven development offers more than just validation of correctness, it can also drive the design of a program. By focusing on the test cases first, we have to imagine how the functionality is used by end-users (in the first case, the test cases). So, we are concerned with the interface before the implementation.

Development Models

Feature Driven Development
Feature Driven Development (FDD) is a framework that organizes the software development process around developing one feature at a time until a complete system is finished. It was originally developed for a large team, and as such is designed to compensate for the range of skills that could be found in a large team.



Feature-Driven Development Cycle
A typical feature-driven development cycle is as follows:

1. *Build a domain object model*, in an intense, highly iterative, collaborative and generally enjoyable activity involving "*domain and development members under the guidance of an experienced object modeller in the role of Chief Architect*". While not mandatory, the object model is typically built using Peter Coad's modelling in colour technique.
2. *Build feature list*, features are small, client-valued requirements. Feature typically take 1-3 days to implement, occasionally 5 days but never 10 or more days to implement.
3. *Plan by Feature*, the last initial phase involves constructing an initial schedule and assigning initial responsibilities. The development team sequence the feature sets based on activities that represent best relative business value.
4. *Design by Feature*, Each feature is tackled by a feature team (3-5 designers and developers usually working together for 1-3 days).
5. *Build by Feature*, This involves the team members coding up the features, testing them at both unit level and feature level, and holding a code inspection before promoting the completed features into the project's regular build process.

FDD mandates code inspections is that research has shown time and again that when done well, inspections find more defects and different kinds of defects than testing. Not only that but by examining the code of the more experienced, knowledgeable developers on the team and having them explain the idioms they use, less experienced developers learn better coding techniques.