Test-Driven Development

CSM2020 (18/19)

Department of
Computer Science

Alexandros Giagkos (alg25, LL-C42)

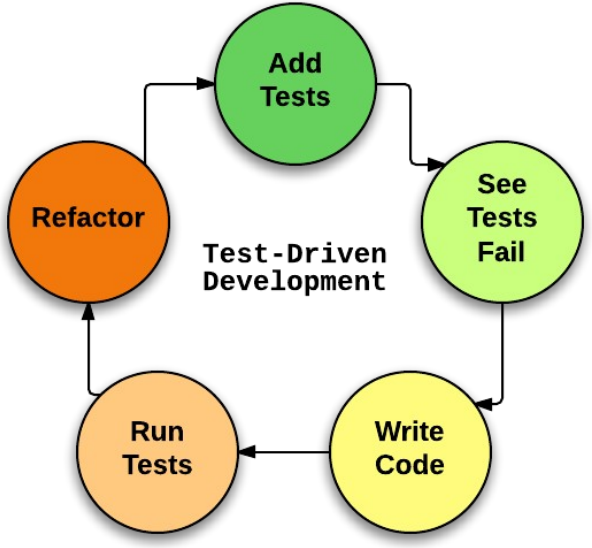
Aims and Objectives

- Brief introduction to TDD
- Types of tests
- Best practices

What is TDD?

- Involves writing tests before the actual implementation
- Relates to eXtreme Programming cycle Red-Green-Refactor

1. Describe the code you wish you had
2. Write a set of tests for each feature
3. Tests are usually independent



Why TDD?

- Higher code quality
- Gives confidence, in particular when refactoring
- No need to constantly test features manually
- Encourages feature-by-feature design and implementation

“Tests can show the presence of bugs, never their absence” - E. Dijkstra

What is BDD?

- Behaviour Driven Development
- Focusing on behaviour not structure
- Stakeholders look for operations that are carried out properly
- Tests written in a form of scenarios:

1. Given some initial context
2. When I [the user] do something
3. And some condition holds
4. Then something should happen

Types of tests

- Unit tests
- Integration tests
- Functional tests
- Behavioural tests

Unit Tests

- Each test covers its own small area of an application
- Do not rely on each other and run in a random order
- Small and pretty simple
- Focus on edge cases

Integration Tests

- Ensures that parts of the application work properly when integrated
- Do unit tests run after you integrated individual parts?

Functional Tests

- Part of the quality assurance workflow
- Given some input, check that the output is correct
- Ensures that specifications are met

Behavioural Tests

- Designed after considering the input from stakeholders
- Behaviours (in terms of user stories) tested
- Output should be of “business value”
- Black box testing

Best practices in TDD/BDD

- Avoid unnecessary functional complexity
  - write only enough of a unit test to fail – write only enough of a production code to pass the failed test
- Focus on what is to be achieved
  - naming conventions, meet the requirement at the end of development cycle
- Test repeatedly
- Code sanity; version control, comments

Summary

- Introduced TDD
- Discussed types of tests
- Reminded best practices