50002 - Software Engineering Design - Software Engineering Design

Oliver Killane

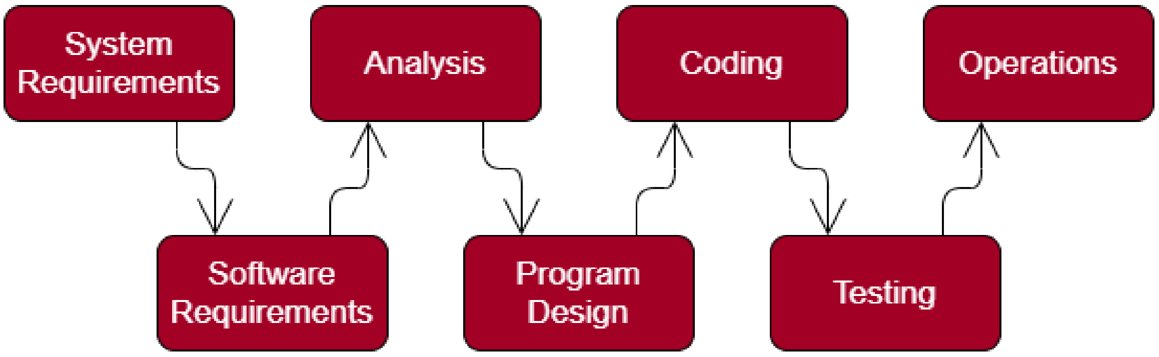
02/01/22

Test Driven Development

Design Principles

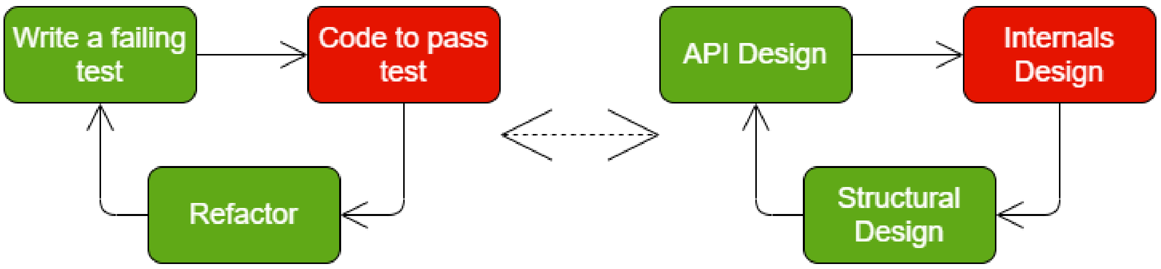
- **Simple Design**
  Smaller codebase is easier to reason about, add new features to.
- **Correct Behaviour**
  Thorough testing (e.g unit testing, automated test suites) used to ensure program behaves as required & is bug free.
- **Reduce Duplication**
  Duplication leads to a larger codebase, greater potential for bugs (changing one instance but nmot another), and makes code changes more arduous (to find and change duplications).
- **Maximum Clarity**
  Code must be easy to read & understand (e.g descriptive naming schemes, commenting, documentation)

Waterfall Development



- Often each stage is managed by a separate team.
- Difficult to correct issues caused by previous stages (e.g altering design when discovering an issue in coding, or fixing major code when finding bugs in testing).

Test Driven Development



Software developed in a cycle.

- Tests define the public api & vice-versa.
- Programming to satisfy tests requires altering program internals (implementation details).

- Refactoring alters the structure but not behaviour (structural design), keepuing the code in a green state.

```java
package testingPackage;

import org.junit.Test;

public class SomeObjectNameTests {

    // Code to run before each test:
    private final someTestObject = new someTestObject();

    ...

    @Test
    public void allowsBehaviourToRunCorrectly() {
        ...

        // Check result using
        assertThat(someResult, is(5));
    }

    ...
}
```