Unit Testing

Testing, Unit Testing

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

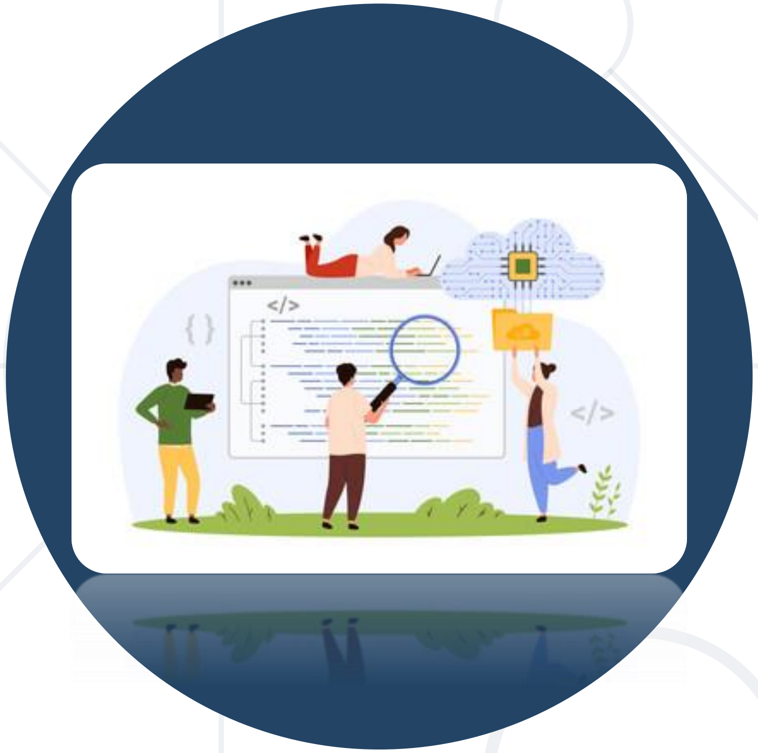
https://softuni.bg

Have a Question?

sli.do

#java-web

Table of Contents



Testing

Testing

- **Testing** is an important part of the application lifecycle
  - In our ever-changing environment, testing is a necessity
  - New features need to be verified, before delivered to the clients

Testing

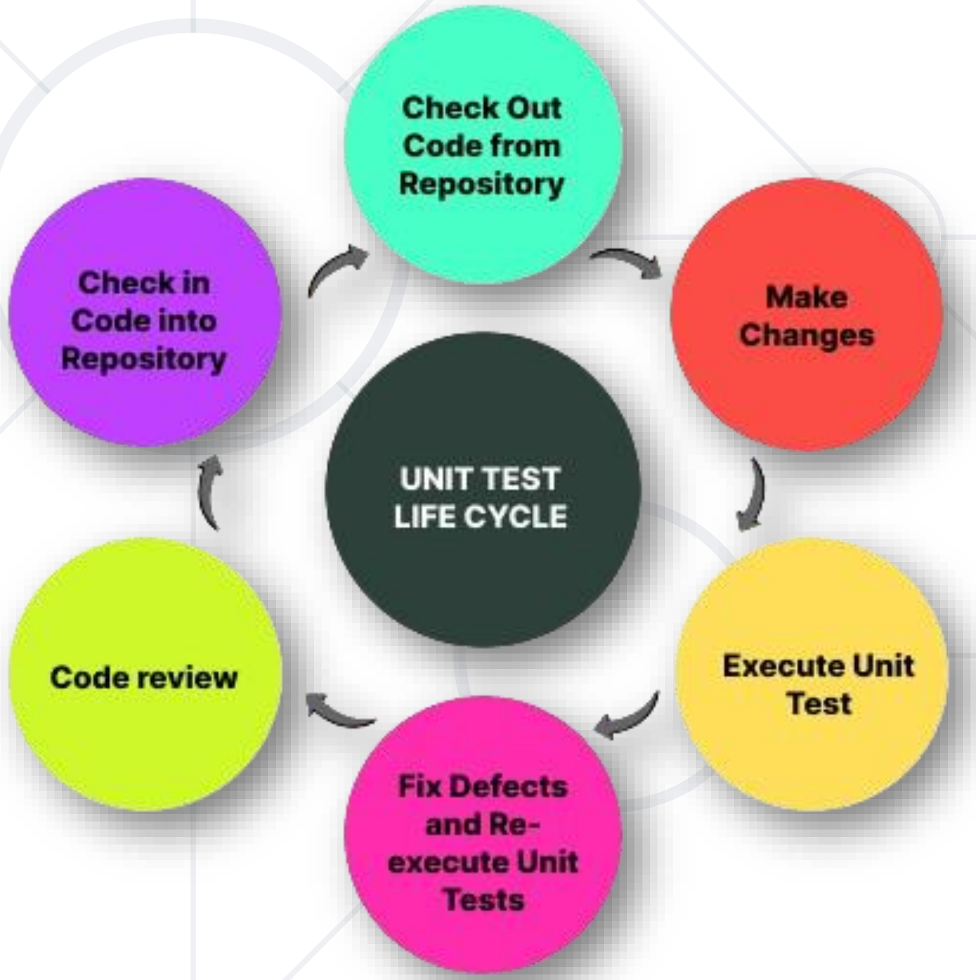
- Testing is a wide area of application development
  - There are several **levels** of testing
  - It does not affect only programmers
  - It has many **concepts** of development
  - There are **different types** of testing

Unit Testing

Unit Testing

- **Unit Testing**
  - A level of software testing where **individual components are tested**
  - The purpose is to validate that **each unit performs as designed**
  - The **lowest level of software testing**
  - Often isolated in order to ensure individual testing

Unit Testing Life Cycle


UNIT TEST LIFE CYCLE
Check Out Code from Repository
Make Changes
Execute Unit Test
Fix Defects and Re-execute Unit Tests
Code review
Check in Code into Repository

Mocking

- Software practice, primarily used in **Unit Testing**
  - An object under test may have **dependencies** on other objects
  - To **isolate** the behavior, the other objects are replaced
    - The replacements are **mocked objects**
    - The mocked objects **simulate** the behavior of the **real objects**

Benefits

- Unit testing **increases confidence** in **changing**/**maintaining code**
- Development is faster:
  - Verifying the correctness of new functionality is not manual
  - Localizing bugs, introduced in development is much faster
- The code is modular and reusable (necessary for Unit testing)

Unit Testing

- **Unit Testing** for web apps is similar to the unit tests we've done
  - Writing test methods to test classes and methods (functionalities)
    - Testing individual code components (**units**)
    - Independently from the **infrastructure**
  - You still use the same testing frameworks as in casual unit testing

Unit Testing

- When using a web frameworks such as **Spring MVC**
  - Built-in logic does not need to be **tested**
    - It is already tested during the development of the **framework** itself
  - You still need to test your **custom** functionality

Unit Testing

- Testing a simple service with mocking in an **Spring MVC** app

```
@Entity
@Table(name = "users")
public class User {
    private String id;
    private String username;
    private String password;

    ...
}
```

```
@Repository
public interface UserRepository
extends JpaRepository<User, String> {
    User findByUsername(String username);
}
```

```
public interface UserService {
    User getUserByUsername(String username);
}
```

```
@Service
public class UserServiceImpl implements UserService {
    ...
    public User getUserByUsername(String username) {
        return this.userRepository.findByUsername(username);
    }
}
```

Unit Testing

- Testing a simple service with **mocking** in an **Spring MVC** app

```
public class UserServiceTests {
    private User testUser;
    private UserRepository mockedUserRepository;

    @Before
    public void init() {
        this.testUser = new User() {{
            setId("SOME_UUID");
            setUsername("Pesho");
            setPassword("123");
        }};

        this.mockedUserRepository = Mockito.mock(UserRepository.class);
    }}
```

Unit Testing (Arrange)

- Testing a simple service with **mocking** in an **Spring MVC** app

```
public class UserServiceTests {
    @Test
    public void
userService_GetUserWithCorrectUsername_ShouldReturnCorrect() {
        // Arrange
        Mockito.when(this.mockedUserRepository
                .findByUsername("Pesho"))
                .thenReturn(this.testUser);

        UserService userService = new
                    UserServiceImpl(this.mockedUserRepository);
        User expected = this.testUser;
    }}
```

Unit Testing (Act)

- Testing a simple service with **mocking** in an **Spring MVC** app

```
public class UserServiceTests {
    @Test
    public void
         userService_GetUserWithCorrectUsername_ShouldReturnCorrect() {
        ...

        // Act
        User actual = userService.getUserByUsername("Pesho");

        ...
    }
}
```

Unit Testing (Assert)

- Testing a simple service with **mocking** in an **Spring MVC** app

```
public class UserServiceTests {
    @Test
    public void
        userService_GetUserWithCorrectUsername_ShouldReturnCorrect() {
        ...
        // Assert
        Assertions.assertEquals("Broken...", expected.getId(),
                                               actual.getId());
        Assertions.assertEquals("Broken...", expected.getUsername(),
                                               actual.getUsername());
        Assertions.assertEquals("Broken...", expected.getPassword(),
                                               actual.getPassword());
    }}
```

Testing

- There are also different concepts and practices of test development
  - **Code-first** approach (The usual Development)
  - **Test-first** approach (Test-Driven Development)

Testing

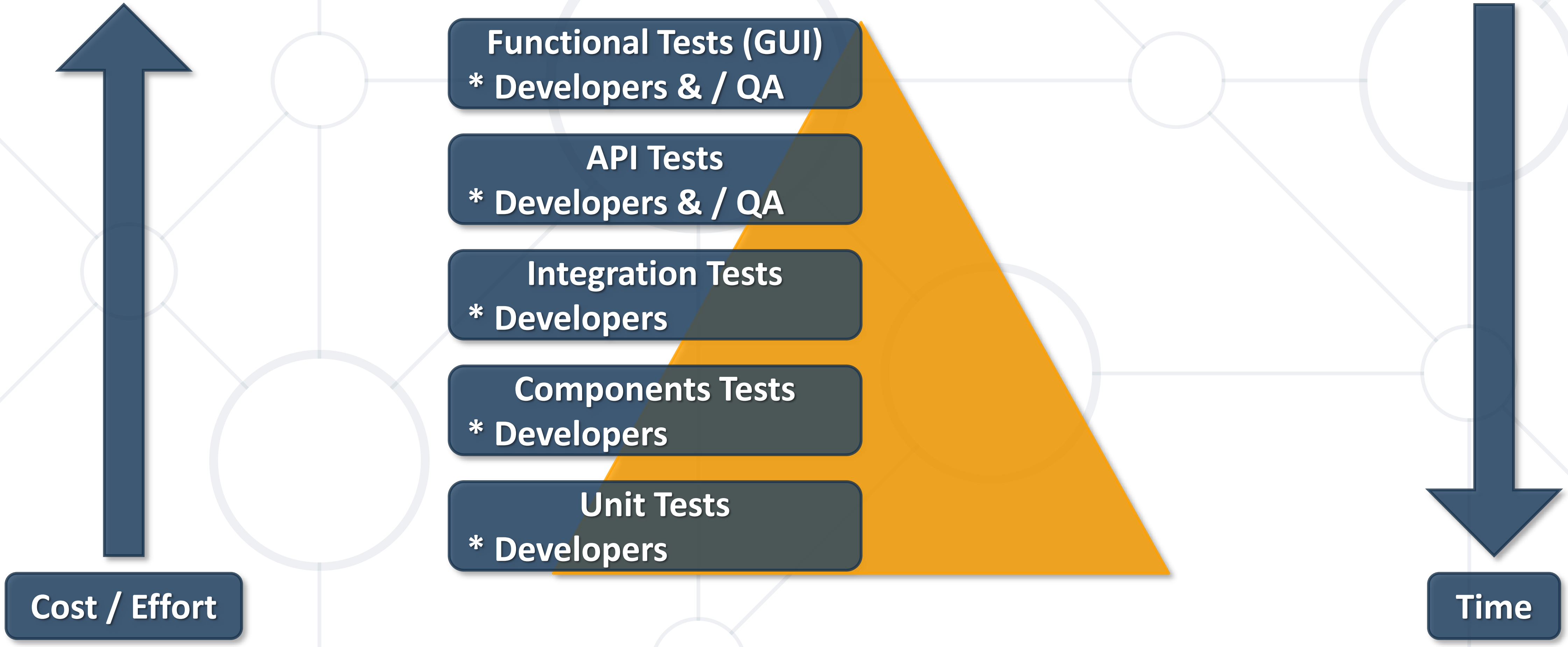
- Each has its own **advantages** and **disadvantages**
  - The **Code-first** approach ensures **flexibility** & **fast** development
  - The **Code-first** approach requires **additional refactoring**
  - The **Test-first** approach ensures **quality** and **edge case coverage**
  - The **Test-first** approach is **complicated** and is an "**initial delay**"

Testing

- Unit testing **ensures** the correctness of a particular unit
  - Not testing **all components** may lead to **false** results
    - A **single unit** may function correctly, independent of the infrastructure
  - Combining components and testing them **collectively** is necessary
  - **Every** level of testing is **essential** to an application's lifecycle

Different Testing levels

- Different Testing levels **require different** time and resources



Summary

- **Testing** is an important part of the application lifecycle
  - New features need to be verified, before delivered to the clients
- **Unit Testing**
  - A level of software testing where individual components are tested
  - The purpose is to validate that each unit performs as designed

Software University

Questions?

SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

СофтУни БУДИТЕЛ

Finance Academy

SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR

Serving the high achievers

SOFTWARE GROUP

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License