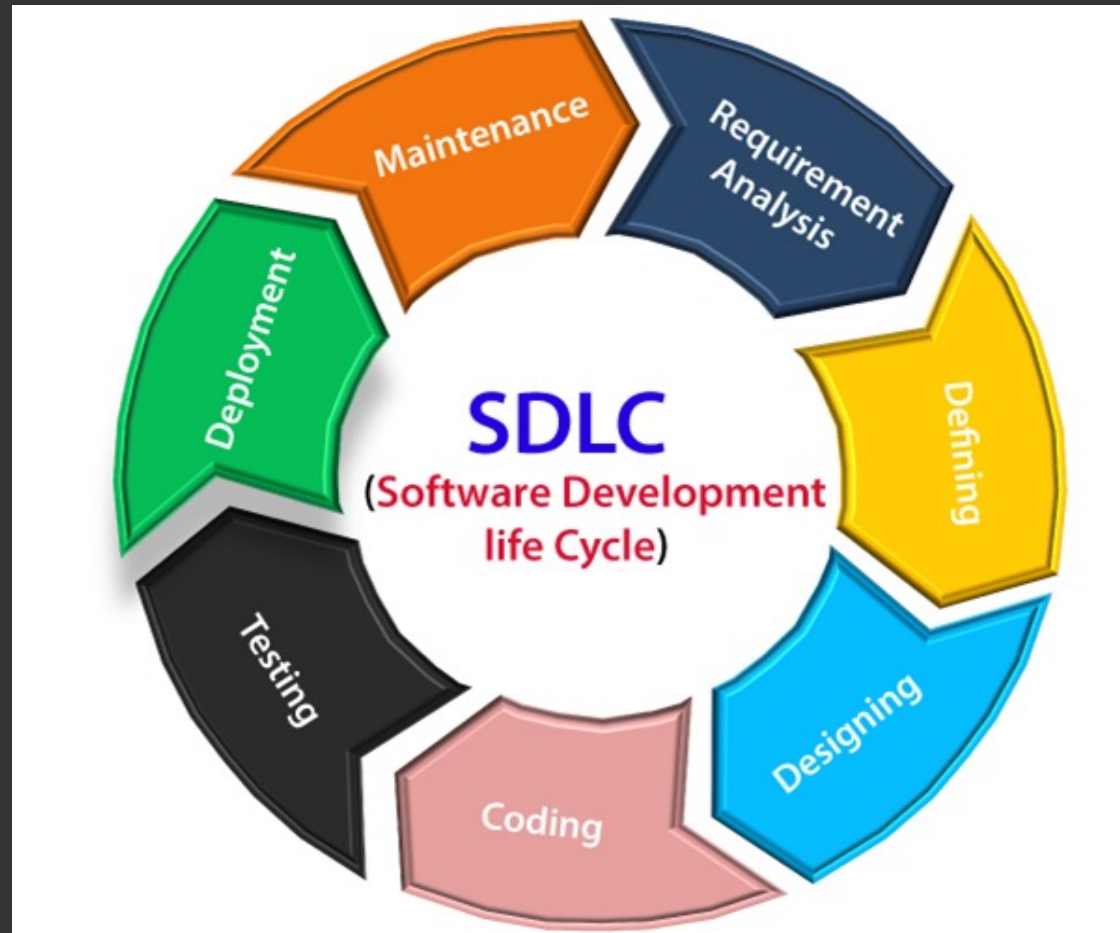




Spring Testing

Introduction to Testing in Spring Boot

SDLC



Importance of Testing

- Identifies Bugs Early and Saves Money
- Mitigates Deployment Risks and Ensures System Stability
- Increases Productivity and Speeds Up Development
- Promotes Confidence among Developers and builds Trust

Spring Boot Test Example

```
class EmployeeServiceTest {  
  
    @Mock  
    private EmployeeRepository employeeRepository;  
  
    @InjectMocks  
    private EmployeeServiceImpl employeeService;  
  
    @Test  
    void testGetEmployeeById() {  
        Employee employee = new Employee(1L, "John Doe", "john.doe@example.com");  
        when(employeeRepository.findById(1L)).thenReturn(Optional.of(employee));  
  
        EmployeeDto result = employeeService.getEmployeeById(1L);  
  
        assertThat(result.getName()).isEqualTo("John Doe");  
        assertThat(result.getEmail()).isEqualTo("john.doe@example.com");  
    }  
}
```

Test Writing Approaches

(Test-Driven Development) TDD :

1. Write a failing test,
2. write the minimum code to pass the test, and
3. refactor the code for better design.

(Behavior-Driven Development) BDD [*extension of TDD*]:

1. Define behavior in human-readable sentences.
2. Write scenarios to meet the behavior.
3. Implement code to pass scenarios.

There are also **Test-After Development** and **Simultaneous Development**

