SPRING TESTING

- Software Engineering (CSI 5324)

Swapnil Saha

Graduate Student | Baylor University

Content

- What is Testing in Software Engineering?
- Why do we need Software Testing?
- Benefits of Software Testing
- Functional Testing
- Annotation
- WebTestClient

What is Testing in Software Engineering?

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. (Source)

Why do we need Software Testing?

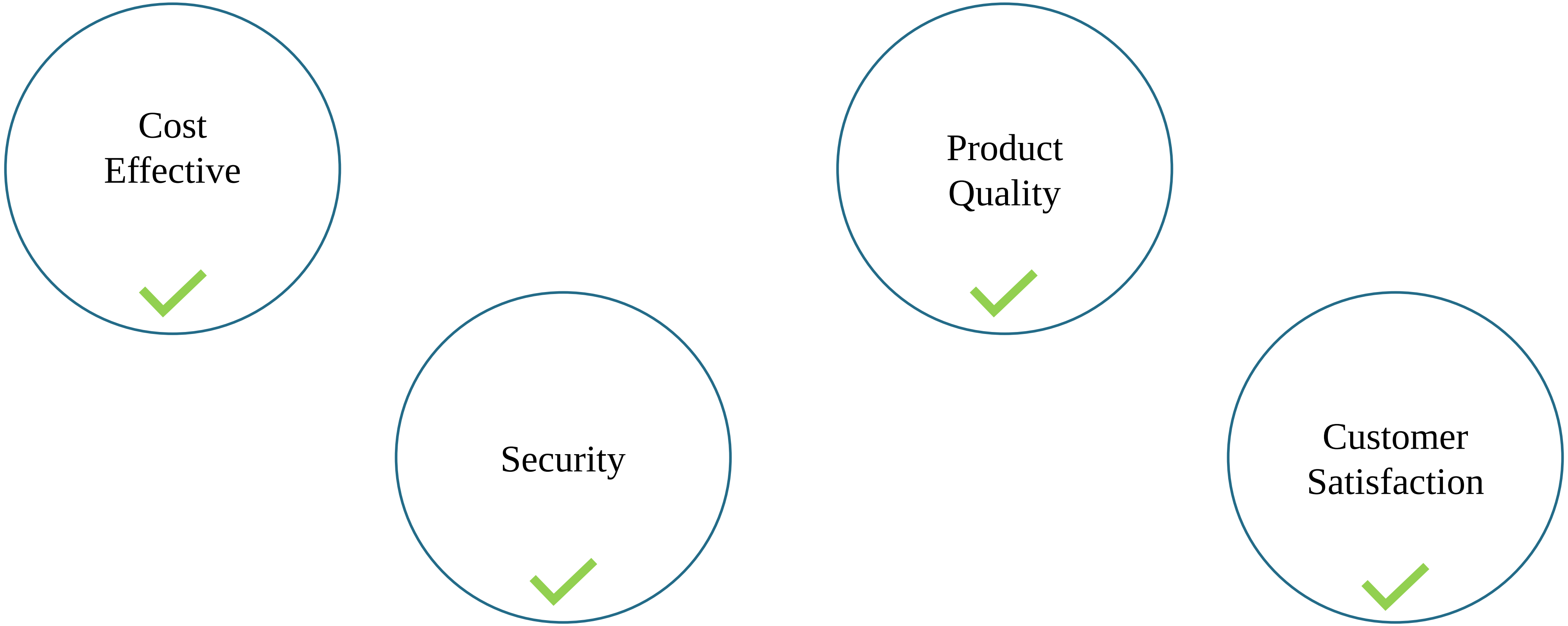
Software bugs can be both expensive and dangerous. It can also cause monetary and human loss.

Examples:

- In April 2015, Bloomberg terminal in London crashed due to software glitch affected more than 300,000 traders on financial markets. It forced the government to postpone a 3bn pound debt sale.
- Nissan cars recalled over 1 million cars from the market due to software failure in the airbag sensory detectors. There has been reported two accident due to this software failure.
- Starbucks was forced to close about 60 percent of stores in the U.S and Canada due to software failure in its POS (point-of-sale) system. At one point, the store served coffee for free as they were unable to process the transaction.

https://www.guru99.com/software-testing-introduction-importance.html

Benefits of Software Testing



Cost Effective ✓

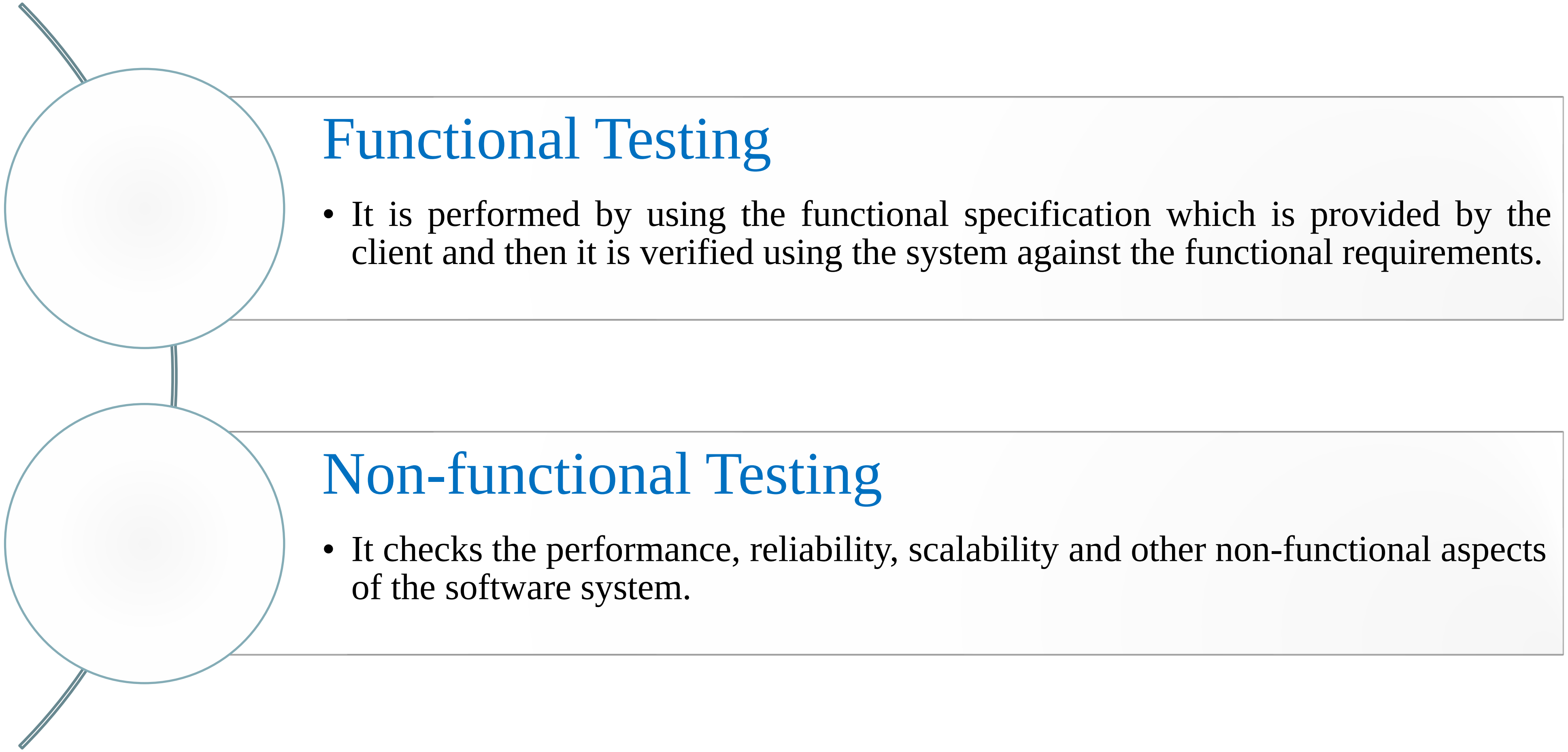
Security ✓

Product Quality ✓

Customer Satisfaction ✓

https://www.guru99.com/software-testing-introduction-importance.html

Types of Software Testing



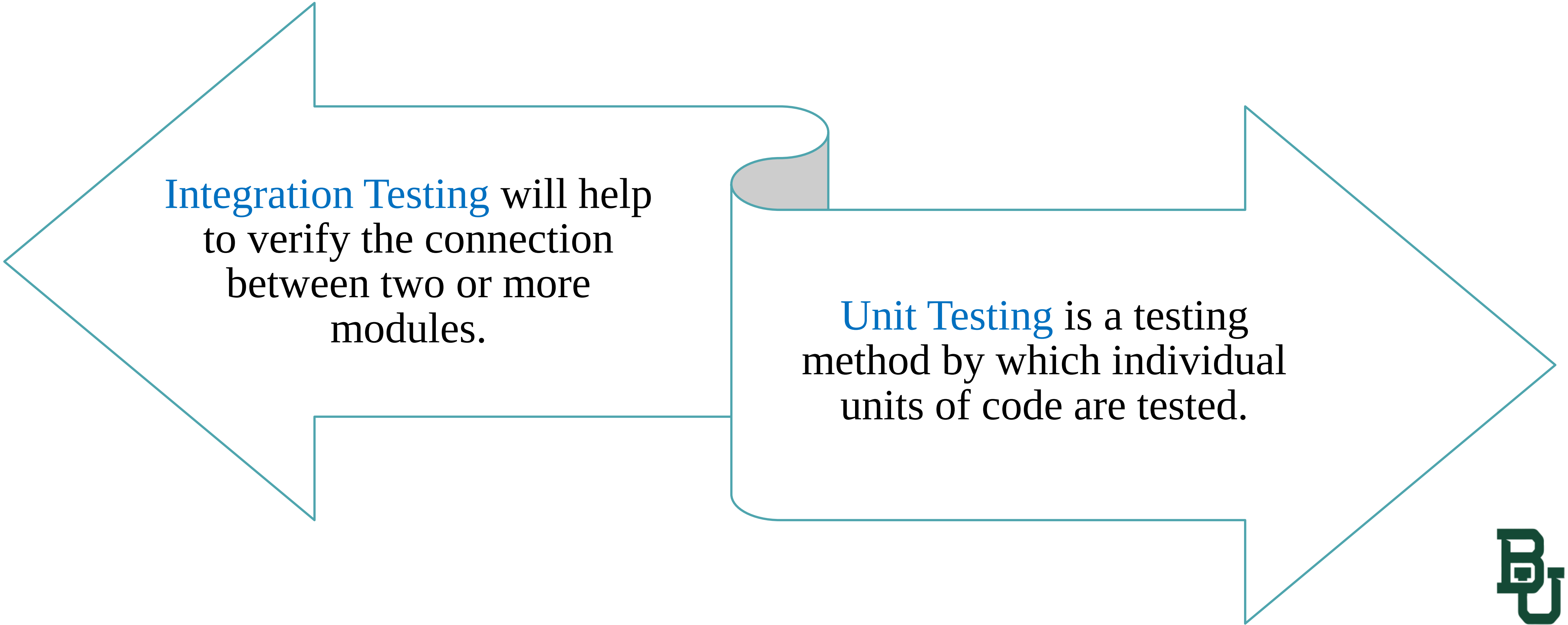
Functional Testing

- It is performed by using the functional specification which is provided by the client and then it is verified using the system against the functional requirements.

Non-functional Testing

- It checks the performance, reliability, scalability and other non-functional aspects of the software system.

Functional Testing



Integration Testing will help to verify the connection between two or more modules.

Unit Testing is a testing method by which individual units of code are tested.

Annotation

@Mock: This annotation is a shorthand for the Mockito.mock() method. It allows us to create a mock object of a class or an interface. We can then use the mock to stub return values for its methods and verify if they were called.

@MockBean: We can use the @MockBean to add mock objects to the Spring application context. The mock will replace any existing bean of the same type in the application context.

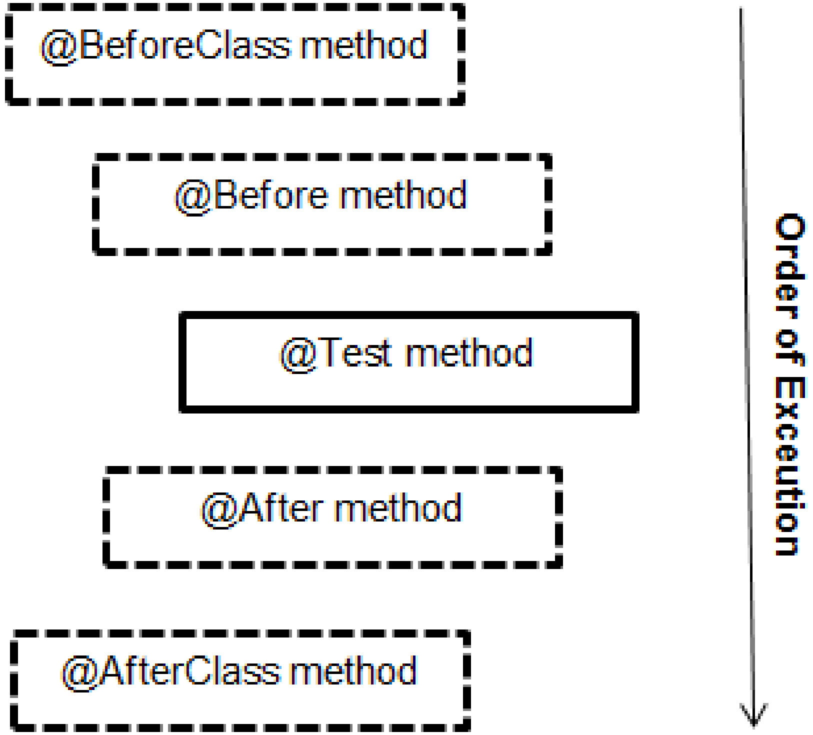
If no bean of the same type is defined, a new one will be added. This annotation is useful in integration tests where a particular bean, like an external service, needs to be mocked.

Annotation

The @BeforeClass and @AfterClass annotated method runs before the execution of test methods in a current class.

The @Before and @After annotation is used when different test cases share the same logic.

To mark a method as a test method, annotate it with the @Test annotation.



WebTestClient

- Non-blocking, reactive client for testing web servers.
- Uses the reactive WebClient internally to perform requests.
- Provides a fluent API to verify responses.
- WebTestClient can connect to any server over an HTTP connection.
- It can also bind directly to WebFlux applications using mock request and response objects, without the need for an HTTP server.

Q & A