Unit Test

Unit Test, bir yazılımın en küçük test edilebilir bölümlerinin, tek tek ve bağımsız olarak doğru çalışması için incelendiği bir yazılım geliştirme sürecidir. Unit Test yazılım testinin ilk seviyesidir ve entegrasyon testinden önce gelir. Unit Testleri geliştiriciler kendileri yazar ve yürütürler.

Unit Test'in yazılımın her biriminin tasarlandığı şekilde gerçekleştiğini doğrulamaktır. Unit Test yazmak kodda yeniden düzenleme(Refactor) işlemini yapmayı kolaylaştırır. Kodda değişiklik yaptığımızda, Unit Testi çalıştırıp oluşturduğumuz algoritmaya uygun bir şekilde çalışıp çalışmadığını kolaylıkla test edebiliriz. Unit Test' ler tüm hataları ortaya çıkarmaz, çünkü her parça izole şekilde test edilmekte ve entegrasyon yapıldığında her şeyin düzenli çalışacağı anlamına gelmez.

Unit Test Frameworkler: Robot Framework, JUnit, Spock, NUnit, TestNG, Jasmin, Mocha

```java
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

public class SampleUnitTestClass {

    Calculator calculatorTest = new Calculator();

    @Test
    public void test_add(){
        // given
       int firstNumber = 10;
       int secondNumber = 20;
       int expected = 30;

       // when
       int actual = calculatorTest.add(firstNumber, secondNumber);

       // then
       Assertions.assertEquals(expected, actual);
    }

    class Calculator{
       int add(int a, int b){
           return a + b;
       }
    }
}
```

Kodun kaynağı:
https://betulsahinn.medium.com/spring-boot-ile-unit-test-yazmak-f1e4fc1f3df

Integration Test

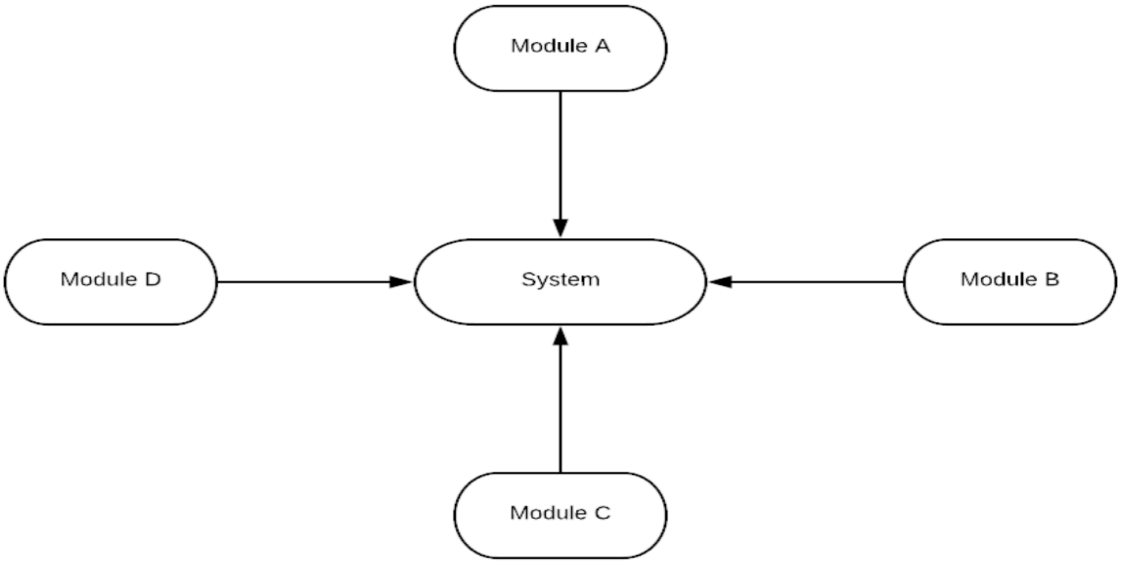
Integration testi, farklı yazılım bileşenlerinin bir araya getirilerek (entegre edilerek) çalışma şekillerinin ve etkileşimlerinin test edilmesini amaçlar.

Yazılım geliştirme sürecinde, genellikle bir proje farklı bileşenlere ayrılır ve her bir bileşen ayrı ayrı geliştirilir. Entegrasyon testi, bu bileşenlerin birleştirilip uyumlu bir şekilde çalışıp çalışmadığını kontrol etmek için yapılır. Bu test, farklı bileşenler arasındaki arayüzlerin doğru çalıştığından emin olmak için önemlidir.

Entegrasyon testi, bileşenlerin birleştirilmesi ve etkileşimlerinin test edilmesi sürecinde gerçekleştirilir. Bu test, her bir bileşenin tek başına doğru çalıştığı durumları kapsamaz; bunun yerine, bileşenlerin bir araya geldiğinde nasıl davrandığını kontrol eder. Böylece, yazılımın farklı parçalarının bir araya geldiğinde beklenen sonuçları üretip üretmediği test edilir.

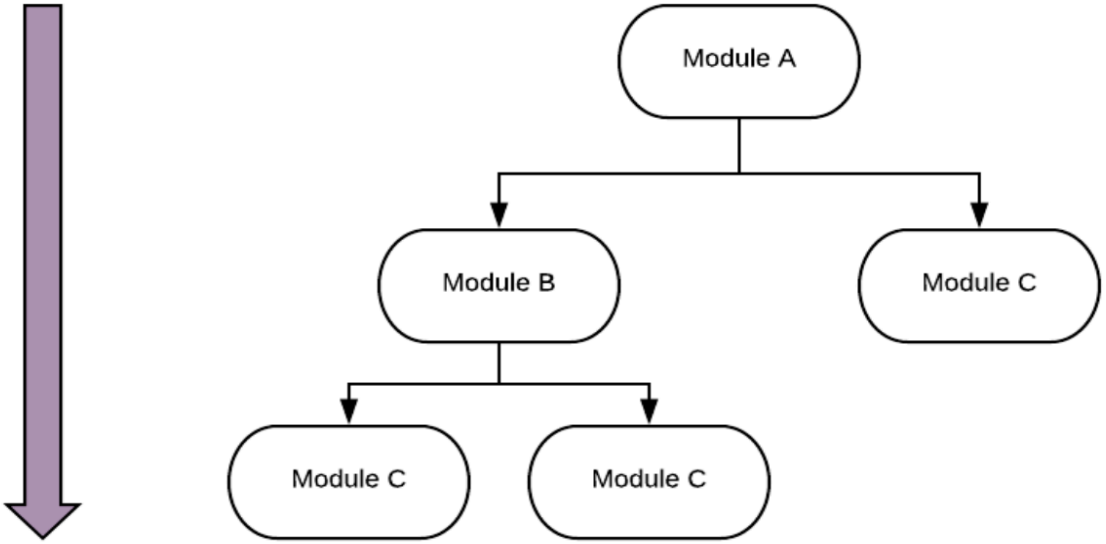
Entegrasyon testleri genellikle birim testler ve sistem testlerinin arasında yer alır ve yazılımın sağlam, uyumlu ve işlevsel bir şekilde çalıştığından emin olmak için önemli bir adımdır. Bu testler genellikle otomasyon araçları kullanılarak yapılır ve yazılımın karmaşıklığına bağlı olarak çeşitli seviyelerde gerçekleştirilebilir.

Big Bang Integration Test
En yaygın kullanılan entegrasyon test tipidir. Geliştirilmiş tüm modüller bir araya getirilerek yapılan testtir. Hızlı ve kolay bir şekilde birbirleri ile beraber çalıştıklarında anlam ifade eden modüllerin doğruluğunu sağlar fakat birim başı metot doğruluğunun gözden kaçınılması olasıdır.



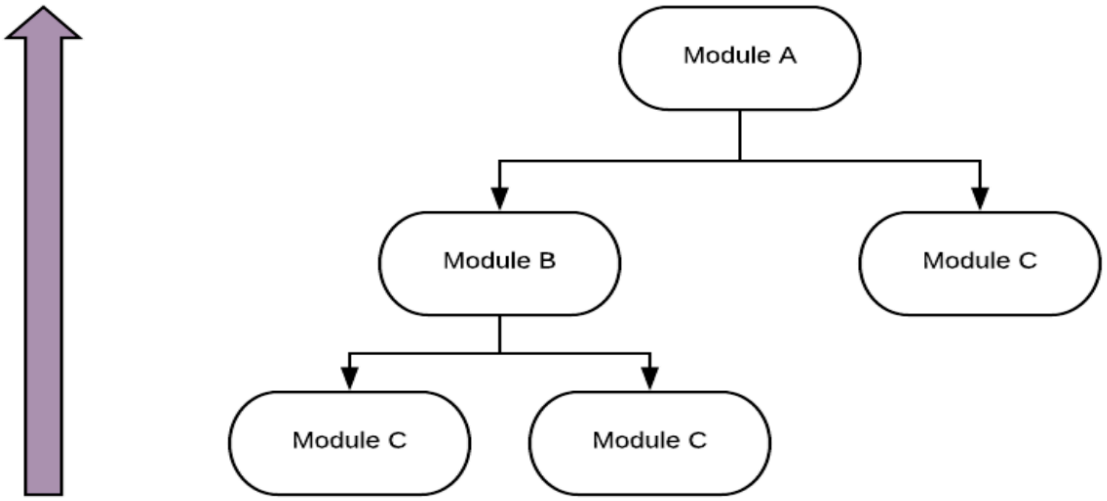
Top-Down Integration Test

Bu entegrasyon testindeki amaç ise modüller arası geçiş yapılırken hatalı olan modülün kolay bir şekilde bulunabilmesini sağlamaktır. Test işlemi yukarıdan aşağı doğru gerçekleşmektedir ve her birinin test işleminden başarılı bir şekilde geçerek ilerlemesi gerekmektedir. Her bir modül testleri stub olarak adlandırılmaktadır. Modül ağacının son bacaklarında ise her bir stub kendi içerisinde test edilerek test işlemi sonuçlandırılır.



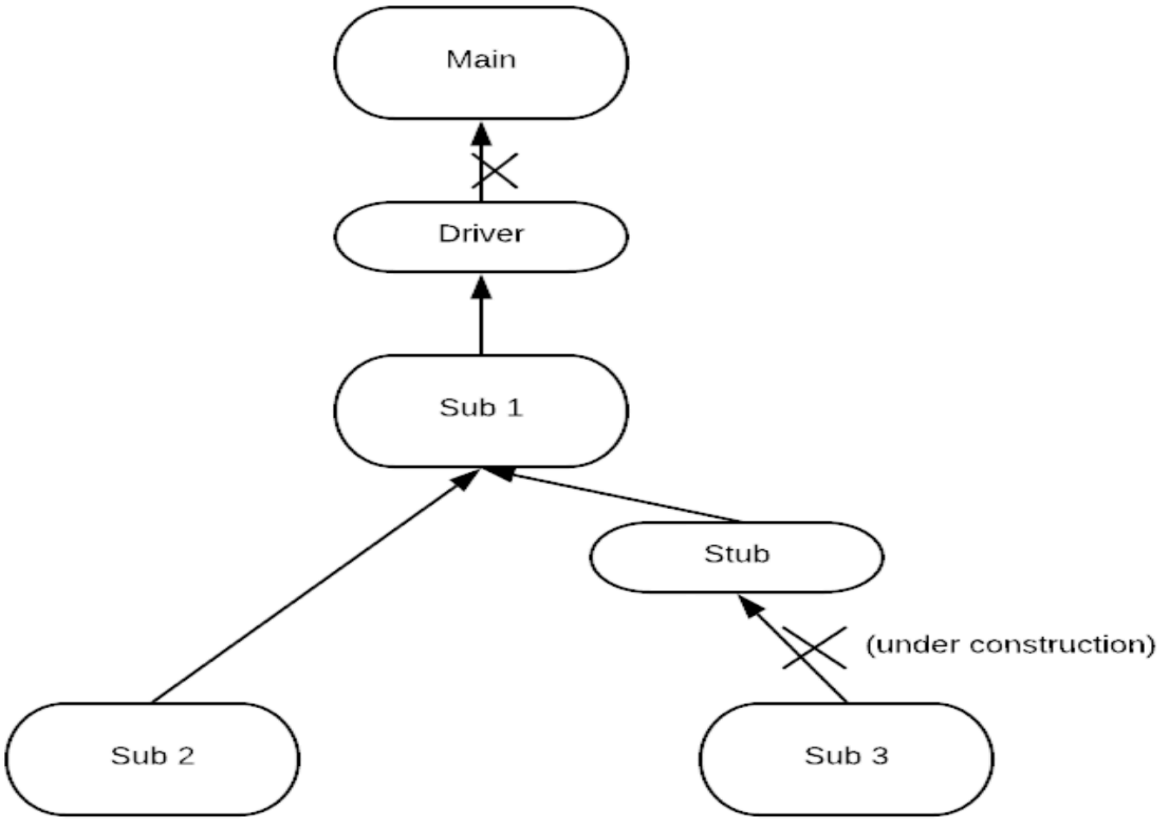
Bottom-Up Integration Test

Bu test yöntemi ise Unit Testler ile beraber ilerlemektedir. Alt tarafta bulunan tüm stublar, Unit Testlerden geçirilerek yukarıya doğru ilerlenir. Top-Down'da olduğu gibi yukarıya ilerlerken Unit Testler aracılığı ile her test başarılı olarak sonuçlanmalıdır. Tüm stublar için Unit Testler oluşturulduktan sonra bir üst seviyede hepsi bir ele alınarak test işlemi yapılır. Bu test tipindeki amaç ise stublardan başlayarak hataların en kısa sürede bulunabilmesidir.



Sandwich/Hybrid Integration Test

Modüllerin bir kısmı Top-Down, bir diğer kısmı ise Bottom-Up tiplerini kullanılarak gerçekleştirilen test tipidir. Bu karma tipteki amaç ise bazı modülleri gruplara ayırabilirken diğer modülleri ise ayrı bir şekilde test edebilmektir.




```
using (var context = new EFContext())
{
    // Arrange
    // Yeni bir entity üretelim
    var testValue = new Value() { Content= "Blabla" };

    // Act
    // DB'ye insert yapalım
    context.Values.Add(testValue);
    context.SaveChanges();
    // Insert yapmış olduğumuz entity'i geri okuyalım.
    var testValueFromDb = context.Values.Single(x => x.ID == testValue.ID);

    // Assert
    // Şimdi ise gelen entity'deki content'in eşitliğini kontrol edelim
    Assert.AreEqual("Blabla", testValueFromDb.Content);
}
```

Kaynak:
https://www.gokhan-gokalp.com/entegrasyon-integration-testi-nedir-ve-tipleri-nelerdir/