

1. Aşağıdaki kavramları örneklerle açıklayın.

- Unit test
- Integration test

Unit Test: Unit Test, bir yazılımın en küçük test edilebilir parçalarının, ayrı ayrı ve bağımsız olarak doğru çalışıp çalışmadığının incelendiği bir yazılım geliştirme sürecidir. Unit Test, yazılım testinin ilk seviyesidir ve entegrasyon testinden önce gerçekleştirilir. Geliştiriciler, Unit Test'leri kendileri yazıp çalıştırırlar. Aşağıda, toplama ve çarpma methodlarının Unit testleri örnek olarak verilmiştir. [1]

```
using System;

using NUnit.Framework;

using TDD_with_VS2012;

namespace NUnitTests{

    [TestFixture]

    public class UnitTests{

        [Test]

        public void Calculator_AdditionTest(){

            var calculator = new Calculator();

            Assert.AreEqual(calculator.Addition(2, 3), 5);

        }

        [Test]

        public void Calculator_MultiplicationTest(){

            var calculator = new Calculator();

            Assert.AreEqual(calculator.Multiplication(2, 3), 6);

        }

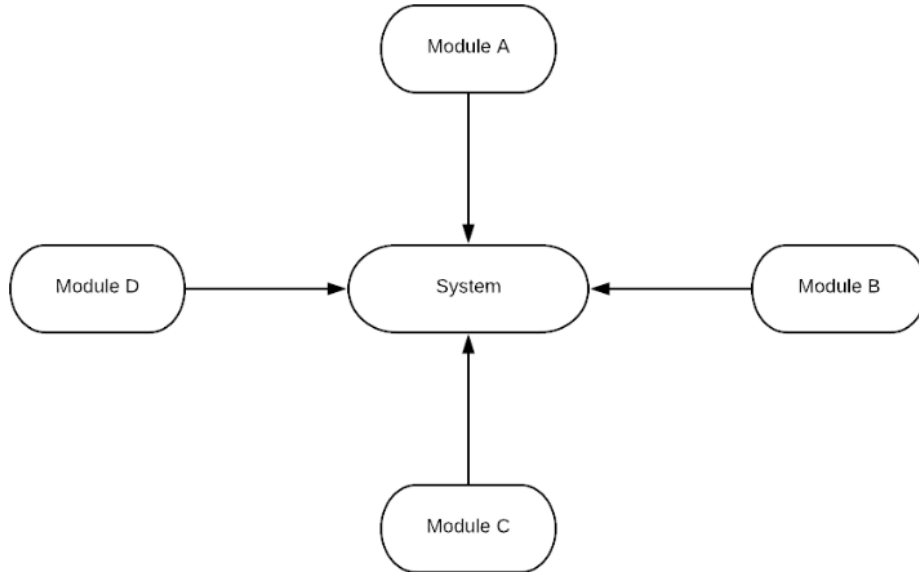
    }

}
```

Integration test: Entegrasyon testi, bir yazılımın farklı bileşenlerinin bir araya getirilerek çalışma uyumluluğunun ve etkileşimlerinin test edildiği bir yazılım testi türüdür. Yazılım geliştirme sürecinde, farklı modüller veya bileşenler genellikle ayrı ayrı geliştirilir ve daha sonra bir araya getirilerek bir bütün oluştururlar. Entegrasyon testleri, bu bileşenlerin birbirleriyle uyumlu bir şekilde çalıştığını ve veri akışının doğru olduğunu doğrulamayı amaçlar.

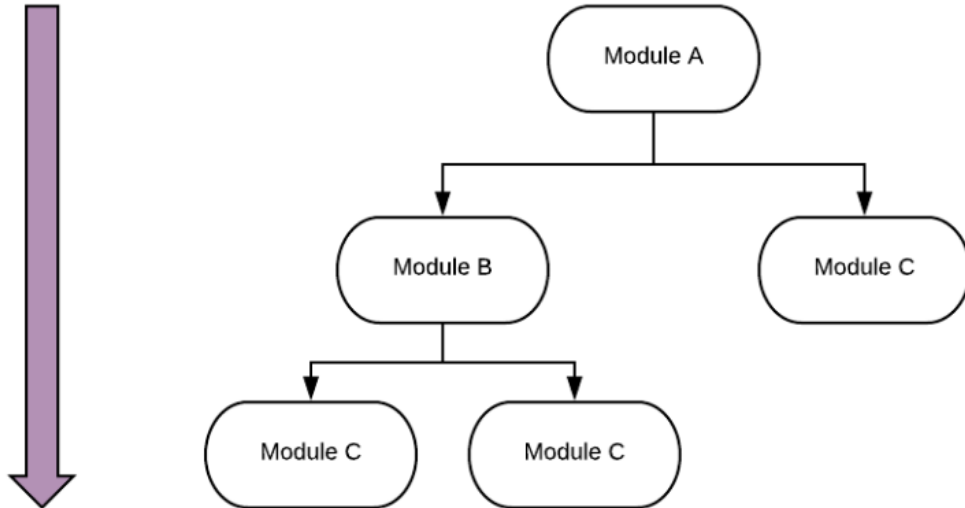
Entegrasyon testleri dört'e ayrılır:

1) Big Bang Integration Test:



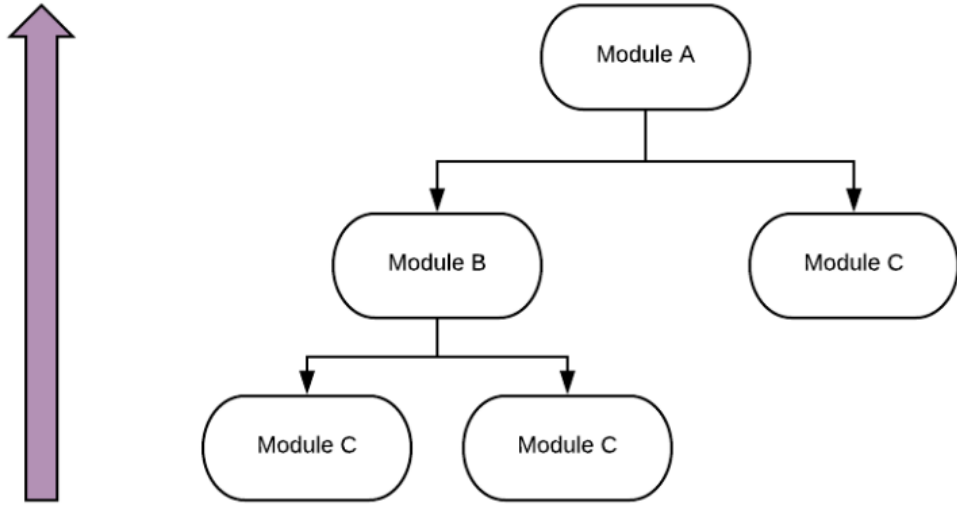
En yaygın kullanılan entegrasyon test tipidir. Bu test, tüm geliştirilmiş modüllerin bir araya getirilerek test edilmesini içerir. Modüllerin birlikte çalıştıklarında beklenen işlevselliği sağlayıp sağlamadığını kontrol eder. Ancak, bu yaklaşımın bir dezavantajı, her birim veya metot düzeyindeki doğruluğunun bazen gözden kaçırılabilmesidir.

2) Top-Down Integration Test:



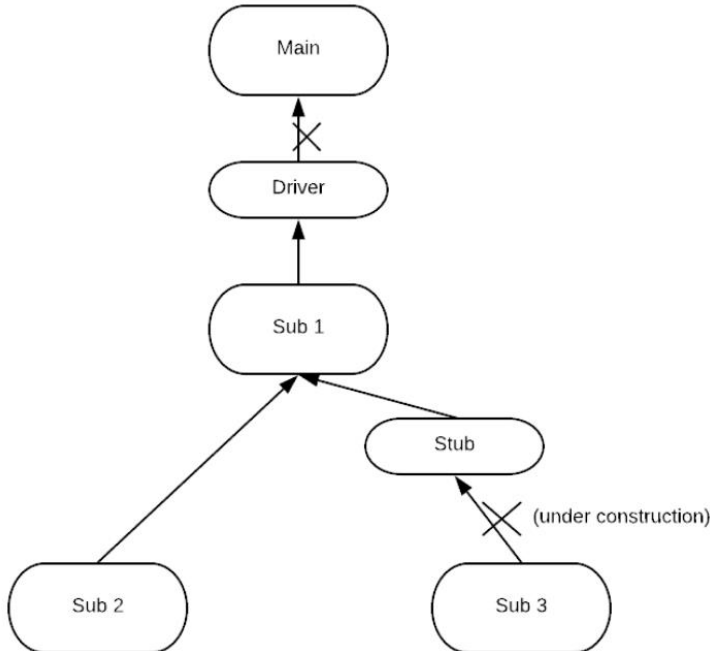
Bu entegrasyon testinin amacı, modüller arasında geçiş yapılırken hatalı olan modülü kolayca tespit etmektir. Test süreci, yukarıdan aşağıya doğru ilerler ve her bir modülün başarılı bir şekilde test edilerek ilerlemesi gerekmektedir. Her bir modül testleri stub olarak adlandırılmaktadır. Modül ağacının son dallarında, her bir stub kendi içinde test edilir ve test süreci tamamlanır.

3) Bottom-Up Integration Test:



Bu test yaklaşımı, Unit Testlerle birlikte ilerler. Alt seviyedeki tüm stublar, Unit Testlerden geçirilir ve bu testlerin başarıyla tamamlanmasıyla birlikte yukarı doğru hareket edilir. Yukarıya doğru ilerlerken, Top-Down yaklaşımında olduğu gibi, her bir testin başarılı olması için Unit Testler kullanılır. Tüm stublar için Unit Testler oluşturulduktan sonra, bir üst seviyede hepsi bir araya getirilerek test süreci tamamlanır. Bu test yönteminin amacı, stublardan başlayarak olası hataların en kısa sürede tespit edilmesini sağlamaktır.

4) 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.

Entegrasyon testine örnek: [2]

```
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);
}
```

KAYNAKÇA

[1] <https://medium.com/@cengizhandumlu.35/unit-testing-nedir-ve-nas%C4%B1l-yaz%C4%B1%C4%B1r-446073767e60>

[2] <https://www.gokhan-gokalp.com/entegrasyon-integration-testi-nedir-ve-tipleri-nelerdir/>