

GAZİ ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BMT-310 YAZILIM PROJE YÖNETİMİ
LABARATUVAR FÖYÜ

HAZIRLAYANLAR

Doç. Dr. Aydın ÇETİN

Arş. Gör. Betül BAYRAK

Arş. Gör. Yasemin YAMAK

ŞUBAT 2019

ÖĞRENME FAALİYETİ

AMAC

1. Integration (entegrasyon) Test nedir?
2. Neden Integration teste ihtiyaç duyarız?
3. Integration Test çeşitleri

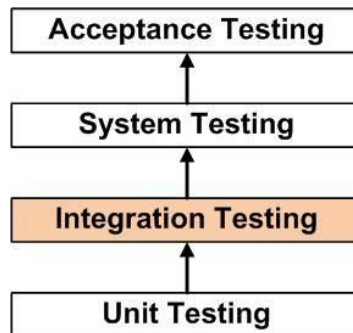
TEORİK BİLGİ

Integration Test Nedir?

Bir yazılım projesi farklı programcılar tarafından kodlanan birden fazla modülden oluşmaktadır. Entegrasyon testi, yazılım modüllerinin mantıksal olarak entegre edildiği ve grup olarak test edildiği bir test türü olarak tanımlanır.

Neden Integration Teste ihtiyaç duyarız?

Entegrasyon testinin amacı mantıksal olarak birbirine entegre edilmiş yazılım modülleri arasındaki etkileşimdeki hataları ortaya çıkarmaktır.



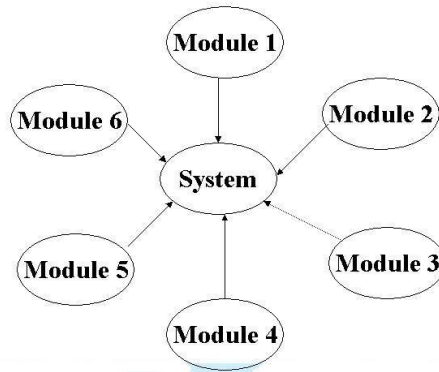
Her yazılım modülüne Unit Test yapılmasına rağmen çeşitli nedenlerden dolayı tekrar hata ile karşılaşılabilir.

- Her bir modül tek tek Unit test ile test edildiğinde bir sorun oluşmamasına rağmen yazılım projesi bu modüllerin mantıksal olarak birbiriyle entegre şekilde çalışması sonucunda başarılı bir şekilde tamamlanmaktadır. Bu yazılım modüllerinin birlikte çalıştığını doğrulamak için Integration Test gereklidir.
- Modül geliştirme aşamasında müşteri tarafından gereksinimlerde büyük değişiklikler meydana gelebilir. Yeni gereksinimler Unit Test ile test edilemeyebilir.
- Verinin tipi veya yapısı bir modülden başka bir modüle transfer edildiğinde değişebilir. Bazı değerler eklenip çıkarılabilir, bu da sonraki modüllerde hataya neden olabilir.

Integration Test Çeşitleri

1. Big Bang Approach

Bu yaklaşımda modüller aynı anda bağlanarak test edilir. Hataya neden olan modülün tespiti zor olduğundan küçük sistemler için uygundur.



2. Incremental Approach (Artırmalı Yaklaşım)

Artırmalı yaklaşımda, mantıksal olarak ilişkili iki veya daha fazla modül birleştirilerek test edilir. Ardından diğer ilgili modül eklenerek doğru çalışıp çalışmadığı test edilir. Tüm modüller başarıyla birleştirilip test işlemleri bitene kadar süreç devam eder.

Artırmalı Yaklaşım, Stubs and Drivers adlı sahte/kukla programlar kullanılarak gerçekleştirilir. Bu programlar sadece çağrı modülü ile veri iletişimini simüle eder.

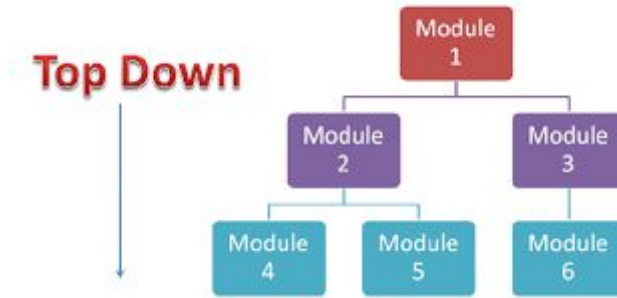
Stubs : Üst modelden gelen girişleri/istekleri ve sonuçları/yanıtları döndüren bir kod parçası olarak tanımlanır.

Drivers : Çağırın fonksiyonun bulunmadığı bir durumda en alttaki modülün fonksiyonlarını çağırmak için kullanılan sahta bir programdır.

Artımlı yaklaşım farklı yöntemlerle gerçekleştirilir.

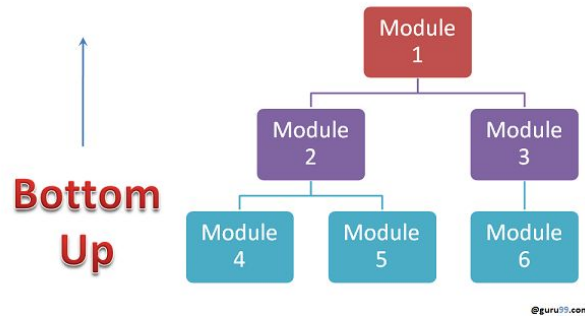
a) Top Down Approach(Yukarıdan Aşağıya Yaklaşım)

Bu yaklaşımda test en üstteki modülden başlayarak en alttaki modüllere doğru ilerler. Birçok stubs kullanımı gerekirken; hatalı modül tespiti kolaydır ve erken prototip elde etme imkanı sağlar



b) Bottom Up Approach(Aşağıdan Yukarıya Yaklaşım)

Bu yaklaşımda test en alttaki modüllerden başlayarak en üstteki modele doğru ilerler. Hatalı modülün tespiti kolaydır ancak uygulama akışını kontrol eden kritik modüller en son test edilir.



c) Hybrid/Sandwich Approach(Hibrit Yaklaşımı)

Bottom Up ve Top Down Yaklaşımların kombinasyonudur. Bu yaklaşımda üst modüller alt modüllerle test edilirken aynı zamanda alt modüller üst modüller ile entegre edilerek test edilir.

