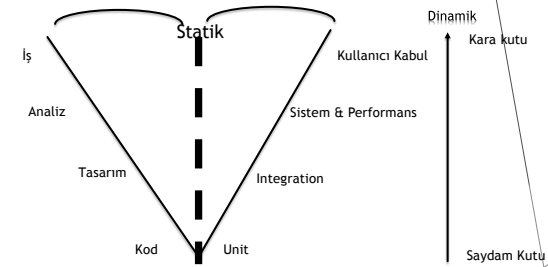


Test Seviyeleri

Birim Testi, Entegrasyon Testi, Sistem Testi, Kabul Testi

1

V -Model



2

Birim Testi - Genel Bilgi

- Birim testi, yazılımın en küçük parçalarını, genellikle fonksiyon veya metot seviyesinde, bağımsız olarak test eder. Birim testleri, geliştiriciler tarafından yazılır ve kodun doğru çalışıp çalışmadığını kontrol eder.

3

Birim Testi - Amaç ve Faydalar

- Amaç:
 - - Kodun doğru çalıştığını doğrulamak
 - - Hataları erken tespit etmek
- Faydalar:
 - - Daha az hata ve düşük bakım maliyeti
 - - Kodun daha kolay anlaşılması ve yönetilmesi
 - - Refactoring sırasında güven sağlar

4

Birim Testi - Örnek

```

▶ public int Topla(int a, int b) {
▶     return a + b;
▶ }

▶ // Birim Testi
▶ public void ToplaTest() {
▶     assertEquals(5, Topla(2, 3));
▶     assertEquals(0, Topla(0, 0));
▶     assertEquals(-1, Topla(-1, 0));
▶ }

```

5

Entegrasyon Testi - Genel Bilgi

- ▶ Entegrasyon testi, birimlerin bir araya getirilmesiyle oluşan bileşenlerin birlikte çalışabilirliğini test eder. Bu testler, birimlerin doğru şekilde entegre olup olmadığını kontrol eder.

6

Entegrasyon Testi - Amaç ve Faydalar

- ▶ Amaç:
 - Birimlerin doğru şekilde entegre olup olmadığını kontrol etmek
 - Veri akışının doğruluğunu sağlamak
- ▶ Faydalar:
 - Entegrasyon sırasında ortaya çıkan hataları tespit etmek
 - Modüllerin birlikte uyumlu çalışmasını sağlamak

7

Entegrasyon Testi - Stratejiler

- ▶ 1. Büyük Patlama (Big Bang) Yöntemi:
 - Tüm birimler aynı anda entegre edilir ve test edilir.
- ▶ 2. Aşamalı (Incremental) Yöntem:
 - Aşağıdan Yukarıya (Bottom-Up): Alt birimlerden başlayarak yukarı doğru entegre edilir.
 - Yukarıdan Aşağıya (Top-Down): Üst birimlerden başlayarak aşağı doğru entegre edilir.
 - Karma (Sandwich): Hem yukarıdan aşağıya hem de aşağıdan yukarıya yöntemlerinin kombinasyonu.

8

Sistem Testi - Genel Bilgi

- Sistem testi, tüm sistemin, sistem gereksinimlerine göre test edilmesidir. Bu testler, sistemin beklenen işlevselliği ve performansı sağladığını doğrular.

9

Amaç:

- Sistemin beklenen işlevselliği ve performansı sağladığını doğrulamak
- Tüm bileşenlerin birlikte uyumlu çalışmasını sağlamak

Faydalar:

- Kullanıcı gereksinimlerinin karşılandığını doğrulamak
- Sistemin genel kalitesini artırmak

10

1. Fonksiyonel Testler:

- Yazılımın belirlenen fonksiyonlarını test eder.

2. Fonksiyonel Olmayan Testler:

- Performans, güvenlik, kullanılabilirlik gibi fonksiyon dışı özellikleri test eder.

11

Kabul testi, yazılımın kullanıcı gereksinimlerini karşıladığını doğrulamak için yapılan testlerdir. Bu testler, yazılımın son kullanıcıya teslim edilmeden önceki son kontrol aşamasıdır.

12

Amaç:

- Yazılımın kullanıcı gereksinimlerini karşıladığını doğrulamak
- Kullanıcının yazılımı kabul edip etmeyeceğini belirlemek

Faydalar:

- Kullanıcı memnuniyetini sağlamak
- Yazılımın piyasaya sürülmeye hazır olduğunu doğrulamak

13

1. Kullanıcı Kabul Testi (UAT):

- Son kullanıcılar tarafından yapılan testlerdir.

2. Operasyonel Kabul Testi (OAT):

- Sistem yöneticileri tarafından yapılan testlerdir.
- Yazılımın operasyonel çevrede çalışabilirliğini doğrular.

14