

Yazılım Test Mühendisliğine İlk Adım: STLC ile Tanışma ve Örneklerle Anlama

Bugün HUAWEI TURKIYE stajımın ilk odak günüydü. Amacım, yazılım test mühendisliği dünyasını keşfetmek, bu rolün ne yaptığını, neden önemli olduğunu ve yazılım geliştirme sürecindeki yerini net bir şekilde anlamaktı.

Bu dosya, hem kendim için bir öğrenme günlüğü hem de sektöre yeni adım atanlara rehber olacak bir kaynak olarak hazırlandı.

1. Yazılım Testi Nedir?

Yazılım testi, geliştirilen bir yazılımın **müşteri gereksinimlerini karşılayıp karşılamadığını** ve **doğru şekilde çalışıp çalışmadığını** anlamak için yapılan kontrol sürecidir.

İki önemli kavram:

- **Verification (Doğrulama):** Ürünün belirlenen gereksinimlere uygun geliştirilip geliştirilmediğini kontrol eder.
"Doğru şekilde geliştirdik mi?" sorusuna cevap verir.
- **Validation (Geçerleme):** Ürünün kullanıcı ihtiyaçlarını gerçekten karşılayıp karşılamadığını test eder.
"Doğru ürünü geliştirdik mi?" sorusuna cevap verir.

Örnek:

Bir mobil bankacılık uygulamasında para transfer ekranı geliştirdiğimizi düşünelim.

- **Verification:** Ekrandaki tüm butonların çalıştığını, formun doğru şekilde veri gönderdiğini kontrol etmek.
- **Validation:** Kullanıcı gerçekten rahat bir şekilde para transferi yapabiliyor mu? İşlem hızlı mı? Kullanıcı beklentisine uygun mu?

Bu süreci yöneten ve yürüten kişiye **yazılım test mühendisi** denir.

2. Yazılım Test Mühendisliği Nedir?

Yazılım test mühendisliği, yazılım ürünlerinin **kalitesini garanti altına almak** için test planlarının hazırlanması, testlerin uygulanması, hataların raporlanması ve geliştirme ekibiyle koordineli çalışılması sürecidir.

Bir test mühendisi sadece hataları bulmaz; aynı zamanda:

- Potansiyel riskleri öngörür
 - Kalite standartlarını uygular
 - Yazılımın bakımını kolaylaştıracak süreçler oluşturur
-

3. Yazılım Geliştirme Sürecindeki Önemi

Yazılım geliştirme **SDLC (Software Development Life Cycle)** ile yönetilir.

Bu döngüde test mühendisliği başlı başına bir fazdır.

Testin önemi:

- **Erken Hata Tespiti:** Üretim ortamına çıkmadan önce hataları yakalamak, maliyeti büyük ölçüde azaltır.
- **Karmaşıklığın Önlenmesi:** Yeni özellik eklenirken mevcut sistemin bozulmasını engeller.
- **Kullanıcı Memnuniyeti:** Sorunsuz çalışan bir ürün, kullanıcıların güvenini artırır.

Örnek:

Bir e-ticaret sitesinde ödeme sistemi bozulursa, bu hem gelir kaybına hem de marka imajına zarar verir.

Test süreci, böyle kritik hataların önceden tespit edilmesini sağlar.

4. Test Mühendisinin Rolü

Test mühendisi, projenin **her aşamasında** aktif rol alır.

Bu sürece **STLC (Software Testing Life Cycle)** denir.

Başlıca testçi türleri:

- **Manual Tester:** Testleri el ile gerçekleştirir, özellikle kullanıcı deneyimi ve görsel detaylar üzerinde yoğunlaşır.
- **Automation Tester:** Testleri kod ve araçlar kullanarak otomatikleştirir, zaman tasarrufu sağlar.

5. Sorumluluklar

Bir test mühendisinin temel görevleri:

- Gereksinimleri anlamak
- Test senaryoları ve test case yazmak
- Test yürütmek
- Hataları raporlamak
- Otomasyon kullanmak

Gerçek Hayat Örneği:

Bir mobil uygulamada giriş formu testi:

Test Case: "Doğru kullanıcı adı ve şifre ile giriş yapılabilmesi."

Beklenen Sonuç: Kullanıcı ana sayfaya yönlendirilmeli.

Test Adımları:

1. Uygulamayı aç
2. Geçerli kullanıcı adı ve şifre gir
3. "Giriş" butonuna bas
4. Başarılı giriş yapılmalı

6. Testin Önemi – Daha Derin Bakış

Test süreci sayesinde:

- Kalite güvencesi sağlanır
- Zaman ve maliyet tasarrufu yapılır
- Kullanıcı memnuniyeti artar
- Hatalar erken bulunur
- Yazılım ölçeklenebilir hale gelir

Örnek:

NASA'nın **Mars Climate Orbiter** projesi, metrik ve imperial ölçü birimleri karışıklığı yüzünden 125 milyon dolara mal oldu.

Bu, test sürecinde tespit edilebilecek bir hataydı.

7. STLC (Software Testing Life Cycle) Adımları

1. Requirement Analysis

Gereksinimleri toplama ve analiz etme.

Örnek: Bir e-ticaret sitesinde "kargo takip" özelliği istendiğinde, hangi verilerin nereden alınacağını anlamak.

2. Test Planning

Hangi test türlerinin kullanılacağı, gerekli kaynaklar ve zaman planı belirlenir.

3. Test Case Development

Test adımlarının net şekilde yazılması.

Örnek: "Sepete ürün ekle" test case'i → Ürün sayfasından sepete ekle butonuna basıldığında sepet sayısının artması beklenir.

4. Test Environment Setup

Testin yapılacağı ortamın hazırlanması.

Örnek: API testleri için Postman kurulumu, mobil test için emülatör ayarları.

5. Test Execution

Planlanan testlerin çalıştırılması ve sonuçların kaydedilmesi.

6. Test Closure

Tüm testlerin raporlanması, öğrenilen derslerin paylaşılması ve sürecin kapatılması.

8. Kapanış – Bugünkü Kazanımlarım

Bugün şunu öğrendim: Test mühendisliği sadece "**hata bulma**" işi değil, yazılımın geleceğini güvence altına alan stratejik bir süreç.

Bu bakış açısı, hem mobil uygulama geliştirme hem de yapay zeka projelerimde bana farklı bir vizyon kazandıracak. Kod yazarken potansiyel hataları daha erken fark edebilecek ve daha kaliteli ürünler ortaya koyabileceğim.
