SDLC

Software Development Life Cycle

Gun 6

17 Ekim 2020

TEST CESITLERI

Mehmet Bulutluoz

KISA TEKRAR

- Software Testing Nedir ve Nicin Onemlidir?
  Yazılımın güvenilirliğini kontrol etmek, ürünün müşterinin gereksinimine uygun olmasını sağlamak, istenilen kaliteye ulasmak icin testler yapilmalidir.
- Kalite Kontrol (Quality Assurance – QA),
  bir sürecin kalite etkinliğini azaltacak durumlara karşı önlem alarak kaliteye hakim olma anlamına gelir.
- Software Testing Life Cycle
  Gereksinim Analizi, Test Plani, Test Senaryosu Gelistirme, Test ortami Kurulumu, Testin gerceklestirilmesi, Test dongusu kapanisi asamalarini icerir
- User Story
  - Istenilen urun veya ozelligin kisa ve anlasilir tanimidir. Urun veya ozelligi isteyen kisinin perspektifinden yazilir. En kisa haliyle asagidaki kalipla user story olusturulur;
  - Bir (kullanici) olarak,(kullanicilarin memnun oldugu bir telefon aramasi) yapabilmek icin (puanlara gore arama yapmak) istiyorum
  - User Story'nin hikayesi kullanicinin yazimi ile baslar, Scrum Team'in Product Backlog'a eklemesi ve sprinte dahil edip uzerinde calismasi ile gercek hayata gecmis olur.

KISA TEKRAR

User Story

Fonksiyon: Login Fonksiyonu

Ön Koşul:

- Kullanıcının login olabilmesi için, daha önceden başarılı bir üye kaydı olmalıdır.

Gereksinimler:

- Kullanıcının eposta adresini girmesi için bir text box olmalıdır.
- Kullanıcının şifresini girmesi için bir text box olmalıdır.
- Kullanıcının bu bilgileri server a göndermesi için bir buton olmalıdır.

Test Senaryoları:

Test Senaryoları	Test Durumları
Eposta alanına fazla karakter girilmesi	Negatif
Şifre alanına fazla karater girilmesi	Negatif
Eposta alanına eposta formatında bir değer girilmemesi	Negatif
Eposta ve şifre alanlarına doğru bir değer girilmesi	Pozitif
Epostası doğru Şifresi yanlış bir değer girilmesi	Negatif
Alanlara hiç bir değer girilmeden formun gönderilmesi	Negatif
Gönderme butonuna iki kere tıklanması	Negatif
Daha önce üye olmayan bir eposta ile login denemesi	Negatif

Test Case

Test Case ID	Test Objective	Pre-Condition	Steps	Test Data	Expected Result	Actual Result	Status
TC_001	I should not login to Facebook	Login should not be available at www.facebook.com	1_Go to https://www.facebook.com/	username ="Never_Giveup" password ="ICanDoIt"	user should not be able to login		
			2_Click on username checkbox				
			3_enter username				
			4_Click on password checkbox				
			5_enter the password				
			6_click on login button				

Açıklama (Requirements)
Acceptance Criteria (Kabul etme kriteri)
Definition Of Done (Bitti tanimi)

(varsa) Onkosul

KISA TEKRAR

- Test Case

  - En basit biçimde, bir test case, bir test yazılımının gereksinimleri karşılayıp karşılamadığını ve işlevlerini doğru bir şekilde yerine getirip getirmediğini belirlediği bir dizi koşul veya değişkendir.

  - Genellikle KG (QA) ekibinden tecrubeli biri test case'leri yazar.

  - Test case hazırlamak için her ekip kendi standart şablonunu kullanır.

- SDET : Software Development Engineer In Test : Bir Framework'u sifirdan olusturabilecek kisidir.

  Automation Tester : Hazir yazilmis test case'leri kullanarak test yapar,olusturulmus framework'leri kullanir.

  Functional Tester & Cross-Functional tester : Manuel tester & Automation Tester

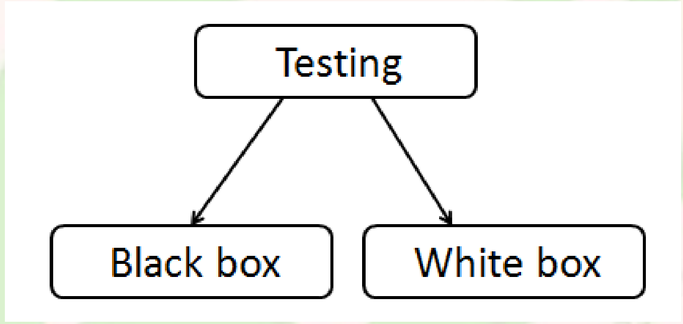
KISA TEKRAR

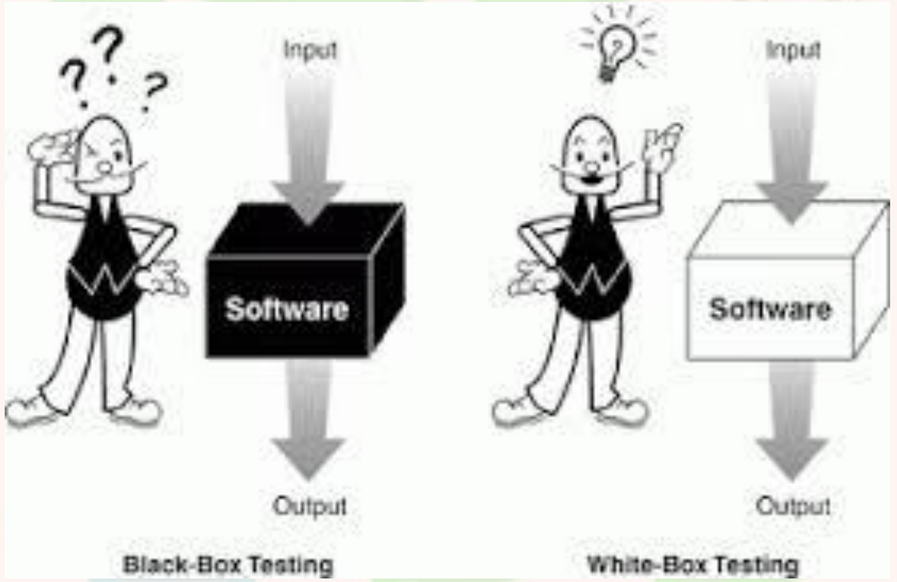
- Test case => adim adım neyi test edeceğinizi açıklayan senaryolarınızdır
- Test plan =>Test kapsamı, yaklaşım kaynakları ve amaçlanan test faaliyetlerinin zamanlamasını tanımlayan giriş ve çıkış kriterleri.
- Test execution => Scriptler hazır olduğunda, çalıştırır veya manuel olarak testinizi gercekleştirirsiniz.
- Test data => functionality (leri) test etmek için kullanmanız gereken veriler
- Acceptance criteria =>functionalityi test etmenin ayrıntılı tanımı
- Expected result => yapılan testin beklenen sonucu
- Actual result => testi yaptıktan sonra ortaya cıkan netice.
- Environments (test yapilan ortamlar (URL)=> Dev,Test,Stage,Prod

Test Cesitleri

Black Box Testing

Kriter	Kara Kutu Testleri
Tanım	Kara kutu testleri yazılımın iç yapısı, tasarım ve gerçekleştirme detayı bilinmeden testlerin tasarlandığı bir yazılım test tasarım tekniğidir.
Test Seviyesi	Kara kutu testleri, Kullanıcı Kabul Testleri (UAT), Sistem Testleri ve Yazılım Testleri seviyelerinde kullanılabilir.
Kim Yapar?	Genellikle yazılım test uzmanları tarafından gerçekleştirilir.
Kodlama Bilgisi	Gerekli değildir
Test Tanımı için gerekenler	Gereksinim belirtimleri





White Box Testing

Beyaz Kutu Testleri

Beyaz kutu testleri, yazılımın iç yapısı, tasarım ve gerçekleştirme detayı bilinerekten testlerin tasarlandığı bir yazılım test tekniğidir.

Beyaz kutu testleri ise daha alt seviyelerde olan birim ve entegrasyon test seviyelerinde kullanılmaktadır.

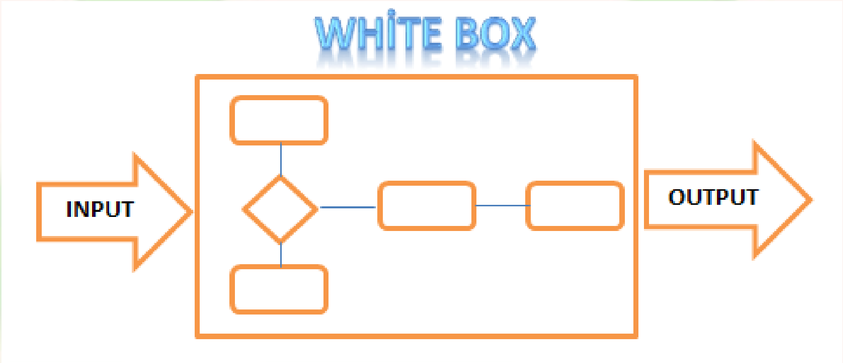
Genellikle, yazılım geliştiriciler tarafından gerçekleştirilir. Gerekli durumlarda testlerin tasarımında ve test yazılımlarının kullanımında test uzmanları yazılım geliştiricilere destek verebilir.

Mutlaka gerekir

Detaylı tasarım

Test Cesitleri

White Box Testing



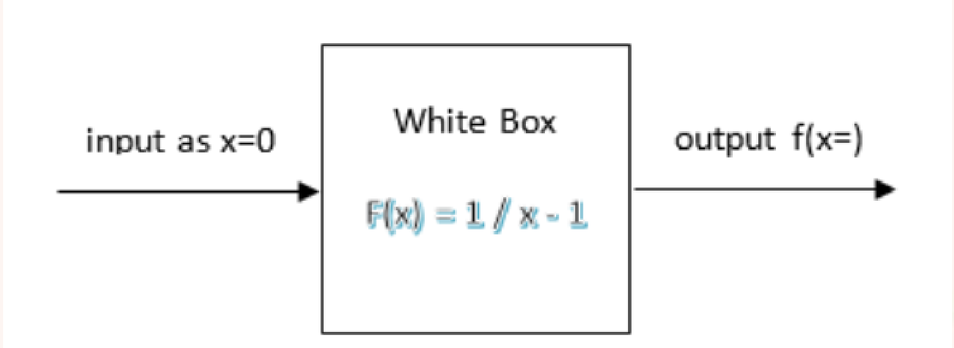
Beyaz kutu, cam kutu, saydam kutu gibi anlamlarla ifade edilen White Box Testi, sistemin iç yapısını bilen kişiler tarafından yapılan yazılım testine verilen tekniktir.

Beyaz kutu testi genellikle yazılım mühendisleri tarafından yapılır. Özellikle bir yazılım birimini hazırlayan geliştirici, birim testler(unit testing) hazırlayarak süreci başlatır.

Uygulamanın program yapısını inceleyen ve uygulamanın program mantığından test senaryoları türeten bir test yaklaşımıdır.

Test Cesitleri

White Box Testing



Yukarıdaki sistemde, f(x) = 1 / x - 1 modülü, beyaz kutu testi ile sınanmaktadır.

Burada testi yapan geliştirici bilmektedir ki yazılım modülü bir f(x) fonksiyonuna sahiptir ve içeriği bellidir. Testi yapan kişi buna bağlı olarak payda= 0 değerini, sistemin güvenilirliği ölçmek üzere özellikle sınamak ister.

Bu ise testi subjektif yapabilme dezavantajı yanında, kodun yeniden gozden gecirilmesi ve kodun güvenilirliği açısından önemli bir durumdur.

Geliştiriciler genelde Code Coverage(kod kapsama/kaplama) tekniklerini uygulayarak, yazdıkları kodların yapısallığını test etmek için beyaz kutu test tekniğini gerçekleştirirler.

Test Cesitleri

White Box Testing

Beyaz Kutu Testinin en büyük AVANTAJLARI
+ Kaynak kodun yan etkileri saptanır.
+ Kod optimizasyonu yapılır.
+ Developer' a kendini geliştirmesi için fırsat verir

Beyaz kutu testinin en büyük DEZAVANTAJLARI
- Testi yapan kişi şayet yazılım mühendisi ise, istemsiz de olsa testi subjektif, ön yargılı bir şekilde yapılabilir. Çünkü geliştirici hatanın olabileceği durumlara odaklandığı için diğer durumları unutabilir veya önemsemeyebilir. Bu durumda gözden kaçan hatalar testin güvenilirliği sarsmış olur.
- Testi yapan kişi şayet test mühendisi olacaksa, bu seferde kaynak kodun ona anlatılması ve bilgi transferi maliyetli olur.

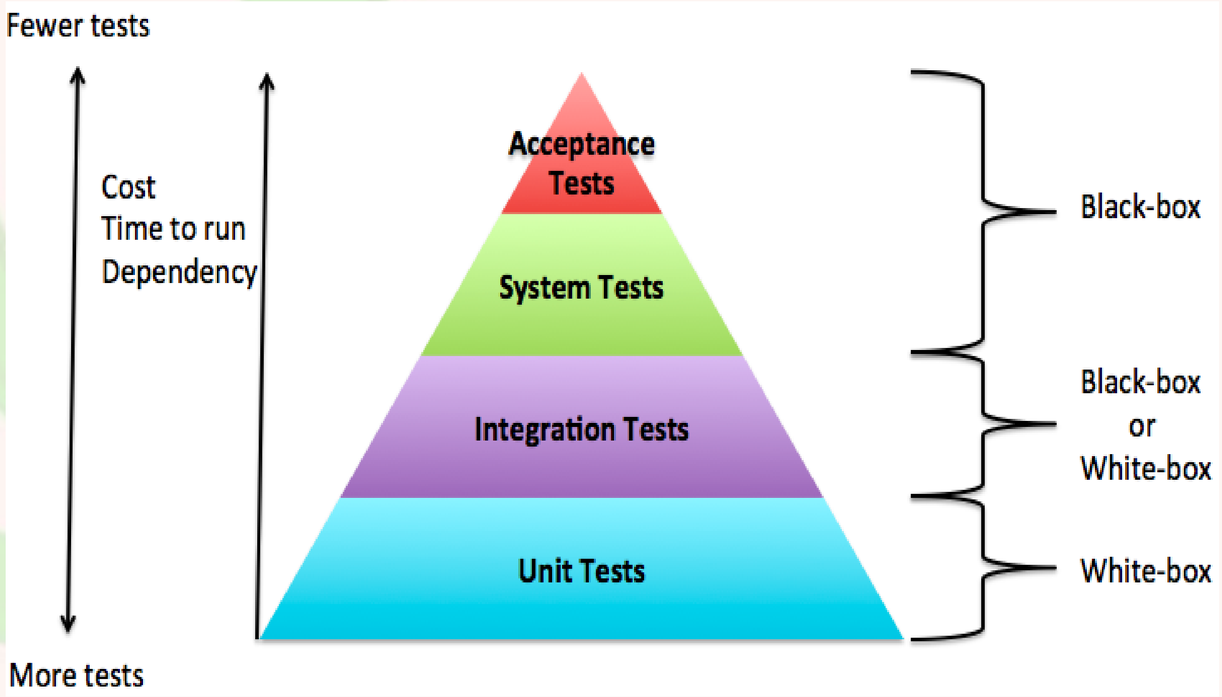
Test Cesitleri

1- Unit Testing (Dev) (White testing)

Birim testi, dinamik testin ilk düzeyidir ve önce Developerlarin ve ardından test mühendislerinin sorumluluğundadır.

Buradaki amacımız yazılımın her biriminin tasarlandığı şekilde gerçekleştiğini doğrulamaktır.

Birim testi, functionality hazir olunca bireysel olarak test edilir (Bir birinden ayri bir şekilde)



Unit Test' ler tüm hataları ortaya çıkarmaz, çünkü her parça izole şekilde test edilmekte ve entegrasyon yapıldığında herşeyin düzenli çalışacağı anlamına gelmez.

Test Cesitleri

2- Integration Testing (Black Box)

- Birim testi tamamlandıktan sonra entegrasyon testi başlar.
- Entegrasyon testinin amacı, uygulamanın farklı bileşenlerinin hala müşteri gereksinimlerine göre çalışmasını sağlamaktır.
- Gerçek sonuçlar ve beklenen sonuçlar aynı olduğunda veya farklılıklar istemci girdisine göre açıklanabilir / kabul edilebilir olduğunda, entegrasyon testinin tamamlandığı kabul edilir.
- Tümleyim Testi olarak da bilinen Entegrasyon testinde; Veri tabanı Data kontrolü, Backend, Mimari Altyapı, Arayüz kısımlarının hepsi test edilir.

Test Cesitleri

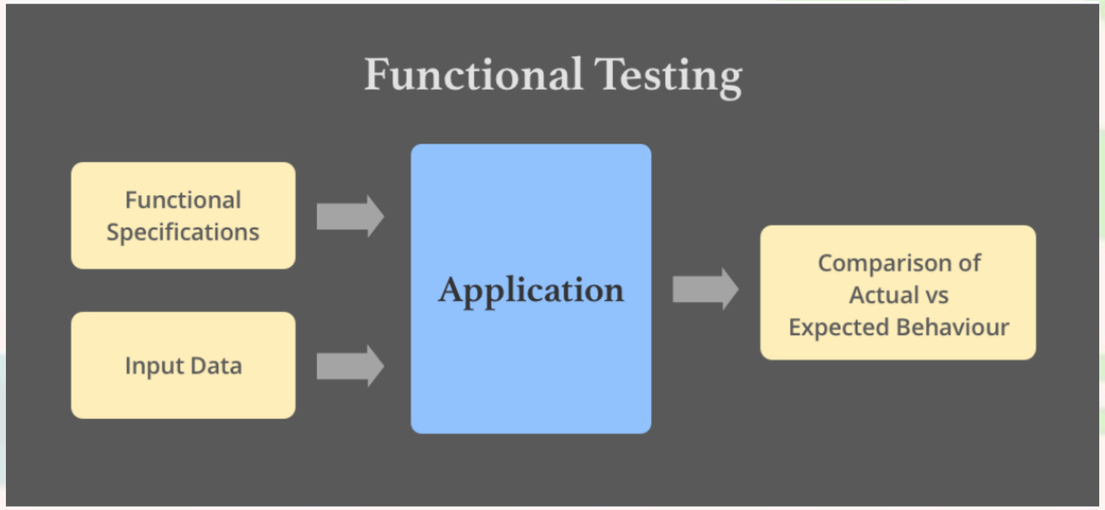
Integration Testing (Black Box)

- Bir proje Release öncesi mutlaka Entegrasyon testinden geçmelidir.
- Test verileri ve test senaryoları oluşturmaya başlamadan önce test stratejisi belirlenmelidir.
- Kritik modülleri tanımlanmalidir
- Test Envoirement testlere baslamadan once hazirlanmalidir
- Proje gelistirme ve Unit Test'ler yapilirken User Requirements degismis mi control edilmelidir.

Test Cesitleri

3- Functional Testing (Fonksiyon Testi) (Black Box)

- Fonksiyonel test, bir uygulamanın fonksiyonel gereksinimlerine yönelik bir testtir.
- Bu tür testler test uzmanları tarafından yapılmalıdır.
- Functional Testlerin yapilmasi, programcıların kodlarını Unit Test yapmamasi veya Integration Test yapilmasina ihtiyac olmadigi anlamina gelmez.
  Uc test birbirini tamamlayici testlerdir.



Test Cesitleri

4- Regression Testing (Tekrar Onay Testi)

- Regresyon testi, yapılan değişiklikler sonucunda sistemde değişiklik yapılan ve değişiklik yapılmayan tum alanlarda olusan yeni hataları bulmak için yapılan kapsamli testtir.
- Regresyon testi, mevcut functionality'lerin sorunsuz çalıştigindan emin olmak için daha once olusturulmus ve kullanilmis test case'lerin tumunun yeniden calistirilmasi demektir. (Bazen tum Test Case'ler yerine belirlenen bazi Test Case'ler de calistirilabilir)
- Bu test, yeni kod değişikliklerinin mevcut işlevler üzerinde yan etkileri olmadigini gormek için yapılır.
- Yazılıma geliştirme sürecinde eklenen her bir yeni işlevin var olan işlevleri bozmadığından emin olmak için, her bir döngüde Regresyon Testini uygulamak kaçınılmazdır.

Test Cesitleri

Regression Testing (Tekrar Onay Testi)

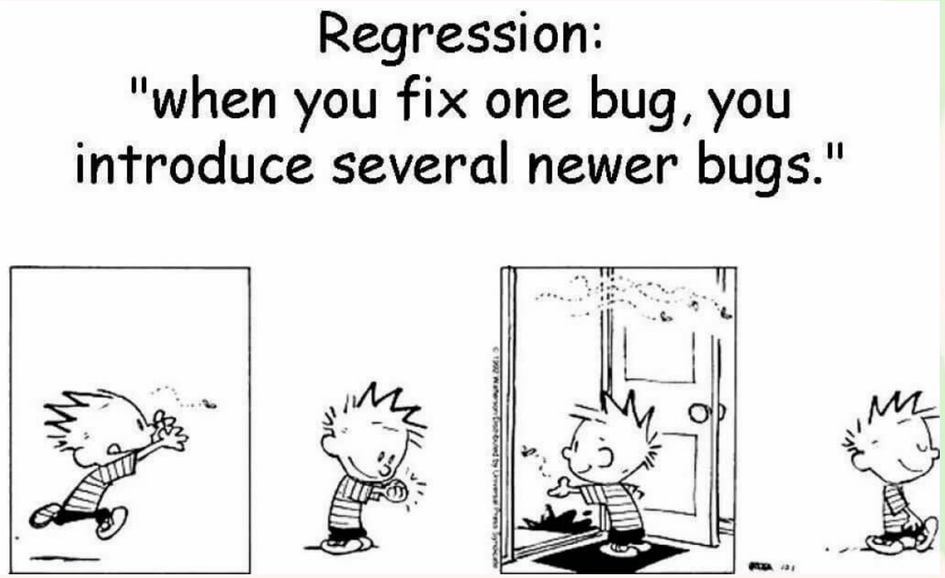
Tekrar testi, yazılım test edildikten sonra raporlanan hataların yazılım geliştiriciler tarafından çözüldüğü bilgisi alındıktan sonra hatanın düzelip düzelmediğini kontrol etmek için yapılan bir testtir.

Yeni kod değişikliklerinin mevcut işlevler üzerinde yan etkileri olmadigini gormek için yapılır.

,

Test Cesitleri

Regression Testing (Tekrar Onay Testi)



1) Kısmi Regresyon

Kısmi Regresyon, kodda değişiklikler yapıldığında ve bu birimin değişmemiş veya halihazırda mevcut olan kodla entegre edildiğinde bile kodun iyi çalıştığını doğrulamak için yapılır.

2) Tam Regresyon

Tam Regrasyon, kodda bir dizi modülde bir değişiklik yapıldığında ve ayrıca başka herhangi bir modüldeki bir değişikliğin etkisinin belirsiz olması durumunda yapılır. Değiştirilen kod nedeniyle herhangi bir değişikliği kontrol etmek için ürün bir bütün olarak test edilir.

Test Cesitleri

5- Sanity veya Smoke Testing (Saglik / Duman Testi)

- Bir programın en önemli fonksiyonlarının çalışıp çalışmadığını anlamak amacıyla detaylara girmeden yapılan testtir.
- Tipik olarak, yeni bir yazılım sürümünün büyük bir test çabası için kabul edecek kadar iyi performans gösterip göstermediğini belirlemek için bir ilk test çabası.
- Örneğin, yeni yazılım her 5 dakikada bir sistem çöküyorsa veya veritabanlarını bozuyorsa, yazılım mevcut durumunda daha fazla test yapılmasını gerektirecek kadar 'makul' durumda olmayabilir.

Test Cesitleri

6- System Testing (End To End Testing)

- Sistem Testi, yazılım gereksinim testlerinin tamamlanmasından sonra, sistem gereksinimlerine göre oluşturulan testleri kapsar.
- Yazılım tarafında yapılan Birim Testi(Unit Test) ve Entegrasyon Testi(Integration Test) adımlarından sonra yapılan sistem testleri, daha çok işlevi tamamlanan yazılımın, güvenlik, güvenilirlik, performans gibi faktörler altında yapılan test işlemlerini kapsar.
- Sistem testinin amacı, bir uygulamanın tasarlandığı gibi işlevleri gerçekleştirirken doğruluğunu ve eksiksizliğini doğrulamaktır.
- Gerçek sonuçlar ve beklenen sonuçlar sıraya girdiğinde veya farklılıklar, müşteri girdisine göre açıklanabilir veya kabul edilebilir olduğunda sistem testi tamamlanmış olarak kabul edilir.

Test Cesitleri

7- Ad-hoc Testing (Gecici Test)

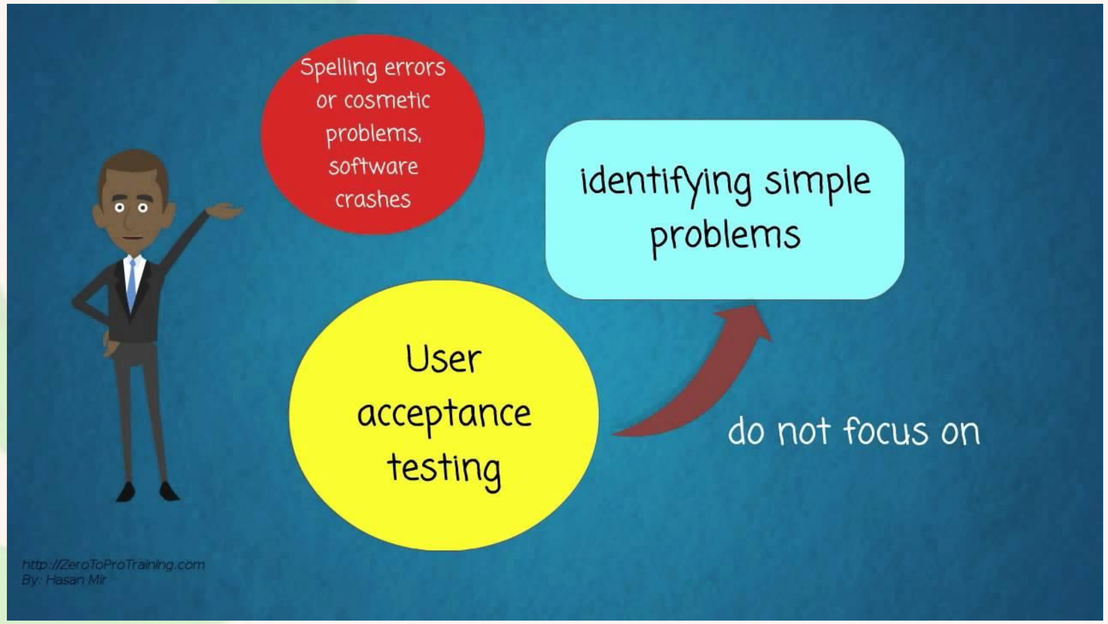
- Rastgele Test veya Maymun Testi olarak da bilinir, herhangi bir planlama ve belge olmadan bir yazılım testi yöntemidir.
- Testler herhangi bir resmi prosedür veya beklenen sonuç olmadan gayri resmi ve rastgele yapılır.

Scenario 1 Üretim sürümünden hemen önce bir sorun bulunursa (team icinde)
Scenario 2 Üretim ortamında bir sorun bulunursa (müşteriler tarafından)

Test Cesitleri

8- User Acceptance Testing UAT (Kullanıcı Kabul Testi)

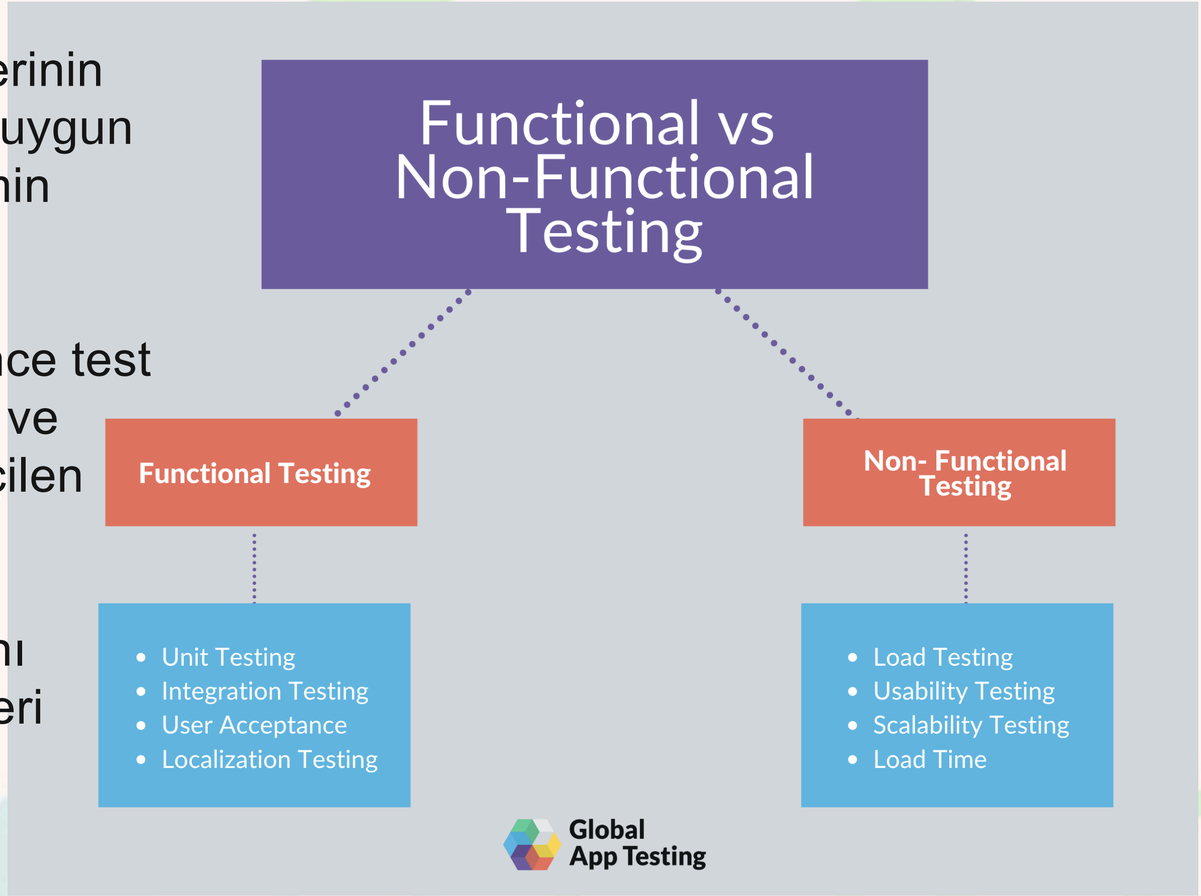
- Son kullanicının veya müşterinin spesifikasyonlarına veya son kullanıcılar/ müşteriler tarafından sınırlı bir süre boyunca kullanıma dayalı nihai test.



- Son Asama olarak her sey gözden gecirilir.
- Beta testi olarak da adlandirilir.
- Hiçbir sistem mükemmel değildir, bu yüzden sistem kurulmadan önce “kabul edilebilir” olması sağlanmalıdır. Kullanıcı Kabul Testi son kullanıcılara, ürünün nihai halini gerçekten kabul edip etmediklerini belirleme fırsatı verir.

Test Cesitleri

- Bir uygulamanın işlevlerinin gerekli spesifikasyona uygun olarak çalıştığından emin olmak için yapılır.
- Fonksiyonel testi gerçekleştirmek için önce test girdisini tanımlamamız ve beklenen sonuçları seçilen test girdi değerleri ile hesaplamamız gerekir.
- Sonra test senaryolarını yürütür ve gerçek verileri beklenen sonuçla karşılaştırırız.



- Uygulamanın beklenen iş yükü altında sorunsuz çalışmasını sağlamak için performans testi yapılır.
- Amaç, güvenilirlik, kaynak kullanımı vb. gibi performans sorunlarını bulmaktir.
- Performans testi yaparken aklımızda tutmamız gereken üç ana nokta hızlı yanıt, maksimum kullanıcı yükü ve çeşitli ortamlarda kararlılıktır.

Test Cesitleri

Non-Functional Testing, Performance Testing (Performans testi)

Performans testi, sistemin belirli durumlarda, belirlenen beklentileri verip vermediğini kontrol etmek amacıyla yapılan testlerdir.
Amacı sistemin belirli bir yük altındaki performansının ölçülmesi ve istenilen performansa ulaşmasını sağlamaktır.

1. Load Testing (Yük testi) (Time & Resource Usage) (Zaman ve Kaynak Kullanımı)

Uygulamaya giderek artan sayıda (Sistem çökene kadar)
sanal kullanıcıyla yüklenilerek sistemin sınırlarının çizilmesi sağlanılır.
Her ne kadar performans testinin bir parçası da olsa , kimi zaman sistemin sağlamlığını test etmek için kullanılır.

2. Stress Testing (Resource Usage “CPU”)

Maksimum sayıdaki kullanıcı ile periyodik bir şekilde sisteme yüklenilmesidir. Yoğunluk durumunda sistemin buna verdiği tepkiyi ölçmek ve yoğunluk sonrası sistemin toparlanma seviyesini belirlemeye yarar.

Test Cesitleri

Test-Tasarim Teknikleri

Equivalence Partitioning (Eşdeğer Aralık) Yöntemi

Aynı bölgedeki değerler girdi olarak kullanıldığında aynı sonucu verir ön koşulundan çalışmaktadır.Ornegin mail uzerinden calisan bir testte farkli mail cesitleri olabilir (Hotmail, gmail,yahoo gibi..) her gruptan en az bir mail ile test yaptiginizda digerlerinin de ayni sonucu verecegi degerlendirilir.

Boundary Value Analysis (Sınır değer testleri)

Değişim noktalarında yazılımın kararlılığını test etmek üzere yapılır. Bu yuzden eşdeğer aralıkların çıkartılması ve en büyük ve en küçük değişim noktalarının saptanması gerekir.
Yaslara gore insanlari grupladigimizi dusundugumuzde, 18 yasin ustu genc diye bir tanim varsa 17 ve 19 yaslari denenerek sistemin dogru calistigi test edilebilir.