Web uygulaması güvenlik testleri

Xss
Sqli
Command injection
Csrf

Sistemi güvenlik açıklarına karşı koruma yöntemleri

Authorization(yetkilendirme):
İstemci yada sunucu tarafında sadece uygun yada yetkili kişilerin erişimine izin vermek mesela kullanıcı adı şifre gibi
Authentication(doğrulama):
Tüm sunucu ve istemci kimliklerini doğrulayan sistem. Her iki sistem de doğrulandığı zaman erişime izin verir
Firewall : internetten gelen bilgileri denetleyen ve ardından güvenlik duvarı ayarlarınıza göre erişimini engelleyen veya izin veren yazılım veya donanım
Encryption: gizlenmek istenen bir bilginin bir algoritma yardımıyla okunmasını ve değiştirilmesini engellemek için veri üzerinde yapılan işleme şifreleme denir.

Performans testleri

Yük testi: yükü çeşitli düzeylerde ve kombinasyonlarda yükleyerek uygulamanın davranışını incelemektedir. Kullanıcı için gerekli olan bağlantı hızını hesaplamada kullanılır
P = n(eş zamanlı kullanıcıların sayısı) * t(zaman birimi başına çevrimiçi işlem sayısı)* d (sunucu tarafından işlenen işlem başına veri yükü)
Stres testi: web uygulamasının ne kadar kapasitesinin olduğunu ve kırılma noktasını belirlemek için yapılır.

Mobil uygulama testleri

Kurulum testi
Performans testi
Güç tüketim testi
Fonksiyonel test
Kesme testi
Kullanılabilirlik testi

Mobil yazılım test ortamları

Emülatör : bir sistemin işleyişini taklit eder ve taklit ettiği sistemin sunduğu özellikleri aynen sağlar.

Simülatör : gerçek bir sistemi sadece modeller, yani sadece işleyişini örnekler, gerçeğe benzer bir ortam oluşturmaya çalışır. Örneklediği sistemin çalışmasının anlaşılmasına yardımcı olur. Simülatör, gerçek sistemin yerine geçebilecek bir sistem değildir.

Test süreci ve yönetimi

Test eylemleri yazılım yaşam döngüsünün şu aşamalarında gerçekleşir:

Gereksinim analizi
Tasarım
Kodlama
Test(yineleme testlerini gerçekleştirmek)
Bakım(yineleme testlerini gerçekleştirmek)

Yazılım test süreci

1. Test planlama

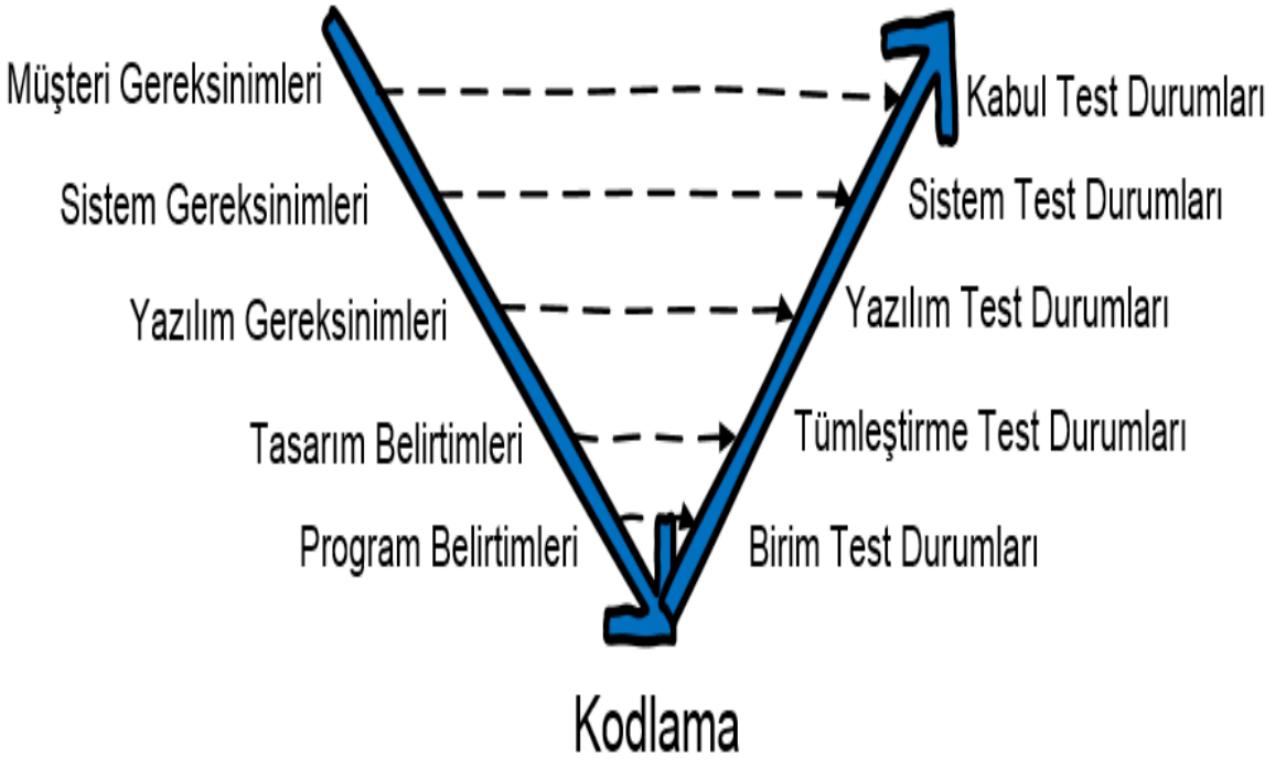
Birinci yaklaşım: bu yaklaşımda test planlama stratejisi, her bir seviye test için ayrı bir test planı geliştirilmesine dayanır. Bu yaklaşıma göre geliştirilebilecek test planları şunlardır:

* Test ana planı
* Kabul test planı
* Sistem test planı
* Tümleştirme test planı
* Birim test planı

İkinci yaklaşım: bu yaklaşıma göre icra edilebilecek tüm testler için tek bir test planı geliştirilir. Genel olarak kabul test planı veya sistem test planı diye adlandırılır.

2. Test tasarım

* Test ortamlarının hazırlanması
* Test yordamının hazırlanması
* Entegrasyon, sistem ve kabul testleri için test durumlarının yazılması



Hata önem dereceleri

Ölümcül: testlerin devam etmesini engelleyecek hatalar
Kritik : testler devam edebilir anca bu halde yazılım teslim edilemez
Büyük : testler devam edebilir ürün bu hata ile teslim edilebilir ancak yazılım kullanıldığında telafisi zor sonuçlar doğurabilir
Orta : testler devam edebilir ürün bu hata ile teslim edilebilir ancak yazılım kullanıldığında telafisi mümkün sonuçlar doğurabilir
Küçük : testler devam edebilir ürün bu hata ile teslim edilebilir yazılımın işleyişinde ortaya çıkacak olan hatalar önemli bir sonuç doğurmaz
Kozmetik : yazılım üzerinde ki font, renk, büyüklük gibi görsel hatalar. Testi durdurmaz ve teslimine engel olmaz.

---

* **Planlama**
* **Tasarım**
* **Gerçekleme**
* **Hata raporlama**
* **Test sonuç raporlama**
* **Değerlendirme**

Test metotları

Kara kutu testleri:

Uygulamanın iç yapısıyla ilgili hiçbir bilgiye sahip olmayan test tekniğidir.
Test uzmanı sistem mimarisiyle ilgilenmez ve kaynak kodlara erişmez.
Test uzmanı sistemin kullanıcı arayüzünde belirtilen girdileri sağlayarak çıktıların doğru olmasını bekler.
Bu test ile yakalanacak hatalar:
* Doğru olmayan yada kayıp fonksiyonlar
* Arayüz hataları
* Veri yapılarında ki hatalar yada harici veritabanı bağlantısı hataları
* Davranış yada performans hataları
* Başlatma ve sonlandırma hataları

Beyaz kutu testleri:

Beyaz kutu testi kodun yapısını ve iç mantık yapılarını detaylı olarak inceler.
Saydam kutu testi yada açık kutu testi olarak da bilinir.
Beyaz kutu testi için kodun iç çalışma yapısının bilinmesi gerekir.

Gri kutu testleri:

Kara kutu ve beyaz kutu testlerinin birleşimidir.
Veri tabanına ve dökümanlara erişimi vardır.
Böylece tasarıma ve verilere uygun test dökümanı üretebilir.
Uygulamanın iç işlemlerine kısmen izin verir.

Birim testi:

Birim testi, yazılım tasarımının en küçük biriminin doğrulanmasıdır.
Birim testi bir bileşenin sınırları içerisinde ki mantık ve veri yapıları gibi iç işlemler üzerinde çalışır.
Yazılan kodun her satırının başka bir kod tarafından test edilmesini sağlar.

Tümleştirme testi:

Birden fazla birimin bir araya getirilerek birbiri içerisinde uyum içinde çalışıp çalışmadığını test eder.
Amaç, birim testlerini başarıyla geçmiş modülleri alıp tasarımda belirtilen program yapısını ortaya çıkarmaktır.
Birimleri bir anda tümleştirmek yerine arttırımlı olarak tümleştirmek daha sağlıklıdır bunun için 3 yol vardır:
Yukarıdan aşağı tümleştirme :
Bu stratejide önce ana denetim birimin testi yapılır. Sonra ona en yakın düzeydeki birimlerden biri ile beraber test yapılır.

Aşağıdan yukarıya tümleştirme:
Alt düzey birimler birleştirilerek kümeler haline getirilir. Bu kümeler test edilir. Daha sonra bu kümelerin birleşmesinden ortaya çıkan daha üst düzeyde kümeler meydana gelir bu şekilde devama eder.

Geri çekilme (regression) testi:
Modül eklendiği veya değiştiği zaman yazılım değişir. Bu test uygulama ortamında yapılan tüm değişikliklerin yeni bir hata üretip üretmediğini kontrol amaçlı yapılan test türüdür.

Sistem testi:

* Performans testi: sistemdeki darboğazları ortaya çıkartır.
* Yük testi: sistemin sınırlarını zorlayarak en fazla ne kadar veri işleme kapasitesi olduğunu belirlemek, bu yükte davranışlarını kontrol etmek amacıyla yapılır.
* Germe testi: normal olmayan durumlarda hem yazılımın hem donanımın ne şekilde davranacağını görmek üzere yapılan stres testleridir
* Kurtarma testi : kurtarma testi önce yazılımı sonra da donanımı çeşitli olası şekillerde bilinçli bir şekilde çökerterek sistemin kendini tekrar toplamasının denenmesi, isterlerin doğrulanması amacıyla yapılır.
* Güvenlik testi : herhangi bir bilgi sızıntısı olup olmadığını kontrol eder. Yapılabilecekler başlıca şunlardır:

1. Zaafiyet taraması
2. Penetrasyon testi
3. Risk belirleme
4. Güvenlik denetimi
5. Şifre kırma

* Taşınabilirlik testi : var olan bir yazılım bileşeni veya uygulamayı yeni bir ortamda test etme işlemidir.
* Kullanılabilirlik testi : tasarımların veya arayüzlerin kullanıcı ile buluşmasından önce tasarımın kullanılabilirliğini ölçmek amacıyla yapılan testlere denir. Kullanılabilirlik 5 niteliksel özellik ile ölçülür:

1. Öğrenilebilirlik : kullanıcılar, tasarımı ilk kullandıklarında yerine getirmeleri gereken görevleri kolaylıkla yapabiliyorlar mı?
2. Verimlilik : kullanıcılar, tasarımın çalışma şeklini öğrendikten sonra gerçekleştirecekleri işlemleri ne kadar hızlı yapabiliyorlar?
3. Memnuniyet : tasarımı kullanmak kullanıcıları duygusal anlamda mutlu ediyor mu, kullanıcılar tasarımı kullanırken kendilerini rahat hissediyorlar mı?
4. Hatırlanabilirlik : kullanıcılar, bir süre tasarımı kullanmadıktan sonra tekrar kullanmaya başladıklarında tasarıma dair olan bilgilerin ne kadarını hatırlayabiliyorlar?
5. Hatalar : kullanıcılar, ne kadar hata yapıyor ve bu hataları ne sıklıkla tekrarlıyorlar, hataları ne kadar hızlı yok edebiliyorlar?

* Kabul testi : Çalıştırılmadan önce yazılımın son sınanmasıdır. Artık yapay veriler yerine gerçek veriler kullanılır. 2 türe ayrılır:

1. Alfa sınaması : geliştiricilerin kendi yerinde müşteri tarafından yapılır.
2. Beta sınaması : birçok kullanıcının kendi ortamında yapılır. Geliştirici genellikle bu testlere katılmaz, yalnızca belirli aralıklarla sonuçları ve yorumları alır.

Doğrulama ve Geçerleme

Doğrulama :
Sistemin hatasız ve iyi bir mühendislik ürünü olup olmadığını inceler. Doğrulama aşamasında bulunan hataların maliyeti daha azdır.

Geçerleme :
Sistemin kullanıcı gereksinimlerine uygunluğunu ölçer.

Doğrulama süreci

* Sözleşme doğrulaması
* Süreç doğrulaması
* İsterler doğrulaması
* Tasarım doğrulaması
* Kod doğrulaması
* Belgelendirme doğrulaması

Yazılım hataları

Error :
Kodlayıcı kaynaklı doğru olmayan bir sonuç elde edilmesi.

Failure :
Sistemin veya bir parçasının gerekli fonksiyonu yeterli performansta yerine getirememesi.

Fault :
Bir yazılım içerisinde doğru olmayan adım, işlem veya veri tanımı.

Hata ayıklama

Brute force:
Yürütme anındaki davranışlar izlenir, yazılım biriminin çeşitli noktalarına ekrana veya bir dosyaya o an akışın neresinde olduğunu, genel durumunu veya bir değişkenin değerini yazan deyimler eklenir. Bu bilgiler ışığında hataya neden olan yazılım kusuru aranır.

Backtracking :
Kodun okunarak geri izlenmesi esasına dayanır. Hatanın oluştuğu yerden geriye doğru gidilerek kod taranır.

Cause elimination :
Tümevarım veya tümdengelim yöntemlerine dayanarak elde edilen verilere göre hatanın nedeni araştırılır. Ortaya konan varsayımları doğrulayıcı yada çürütücü testler tasarlanır.

Testi kim yapar

* Yazılım test ekibi
* Yazılım geliştirici
* Proje lideri
* Son kullanıcı

Yazılım geliştirme metodolojilerinde test

Şelale modeli :
Yazılım süreci lineardir, bir sonraki safhaya geçebilmek için bir önceki safhada yer alan aktivitelerin tamamlanmış olması gerekir. Yani test aşamasına gelebilmek için diğer aşamaların tamamlanmış olması gerekir

V modeli :
Bu modelde geliştirme ve test paralel olarak yapılır. İsterlerin iyi tanımlandığı, belirsizliklerin az olduğu ve aşamalar halinde ilerlenmesi gereken projelerde v modeli iyi sonuç verir.

Spiral model :
Proje çevrimlere ayrılır ve her bir çevrimin riskleri ayrı ayrı ele alınır.

Agile :
Geleneksel yaklaşımın tersine, test profesyonelleri, yazılım yaşam döngüsünün en başından itibaren sürece dahil olurlar.

Kalite kontrol

Hata ayıklamaya yönelik, sistematik ve belirli bir anda yapılır.

Kalite güvence

Hata oluşmasını önlemeye yönelik, sistematik, zamana yayılmıştır.

Kullanıma göre ölçütler

* İşlevsellik
* Doğruluk
* Güvenilirlik
* Sağlamlık
* Verimli çalışma
* Kullanım kolaylığı
* Korunmalı olma
* Ekonomiklik
* İşletim sürekliliği

Taşınmaya yönelik ölçütler

* Tekrar kullanılabilirlik
* Uyumluluk
* Taşınabilirlik

Yenileştirmeye yönelik ölçütler

* Bakım kolaylığı
* Doğrulanabilirlik
* Genişleyebilirlik

Garvin’in kalite ölçütleri

* Performans kalitesi
* Özellik kalitesi
* Güvenilirlik
* Uygunluk
* Dayanıklılık
* Kullanılabilirlik
* Estetiklik
* Algılama

McCall’ın kalite ölçütleri

* Ürün taşıma
* Taşınabilirlik
* Yeniden kullanılabilirlik
* Birlikte çalışılabilirlik

* Ürün işletme
* Doğruluk
* Güvenilirlik
* Verimlilik
* Bütünlük
* Kullanılabilirlik

* Ürün inceleme
* Bakım kolaylığı
* Esneklik
* Test edilebilirlik

Iso 9126 kalite faktörleri

* Kullanılabilirlik
* Taşınabilirlik
* Verimlilik
* Güvenilirlik
* Fonksiyonellik
* Bakım kolaylığı