Yazılım neden başarısız olur?
Öncesinde tanımlanan özellik, işlev ve gereksinimlerin daha azını karşılarsa,
başlangıçta planlanan bütçesini aşarsa,
tahminlenen bitirilme süresinden daha uzun sürerse,
ya da tamamlanmadan iptal edilirse

İnsanlar hata yaparlar, bu hatalar kodda, yazılımda, sistemde ya da dokümanda defect oluşturur. Defect olan kod çalıştırıldığında sistem beklenen fonksiyonları gerçekleştiremez ve başarısız olur. Ayrıca zaman kaygısı ve gerekli kontrollerin yapılmaması da nedenler arasında sayılabilir.

Testin rolü nedir?
İleriye dönük kodun geliştirilme masraflarını azaltmak,
Ürün çalıştırılmadan önce kalitesini ölçmek,
Yapılan hataların tekrarlanmasını önlemek,
Zaman ve maliyet tasarrufu yapmak,
Saygınlık kazanmak, şeklinde yazılım testi amaçları sıralanır.

Kalite maliyetinin bileşenleri nelerdir?
Kalitesizlik maliyeti: iç hata maliyeti ve dış hata maliyetinin
Kaliteyi sağlama maliyeti: değerlendirme maliyeti ve önleme maliyetinin

FMEA (Failure Mode Effect Analysis) adımları nelerdir?
Risklerin kategorize edilmesi
> Fonksiyonelite, Yük & Kapasite, Güvenilirlik, Sürdürebilirlik, Veri kalitesi, Performans, Dökümantasyon, Arayüz,Entegrasyon

Risklerin ölçeklendirilmesi
> Sistem açısından; Veri kaybı, işlevsellik kaybı, kısmi işlevsellik kaybı, kozmetik riskler
> Müşteri açısından; Acil, zorunlu, önemli, düzeltilmesi iyi olacak, isteğe bağlı
> Gerçekleşme olasılığı açısından; muhtemel, mümkün, ihtimal dahilinde olmayan

Risklerin RPN (risk priority number) belirlenmesi
> Ölçeklerin hepsi toplanır, değer ne kadar azsa önemi o denli artar.

Temel test seviyeleri nelerdir?
Unit (Birim) Test: Ufak modüllerin kendi içindeki testi
Modül Seviyesi: Ayrık parçaların birbirleriyle etkileşim testi
Sistem Seviyesi: Parçalar birleştirildikten sonra bütün test
Kabul Seviyesi: Üretilen yazılım müşteri tarafından testi

Test uzmanında olması gereken özellikler?
Süreci iyi bilmeli,
Metadolojisi olmalı,
Planlı olmalı,
Net olmalı,
Araçları olmalı,
Test ortamı olmalı,
Şüpheci olmalı.

Test uzmanının yapması/yapmaması gereken nelerdir?
Hatayı bul, erken bul, çözüldüğünden emin ol.

Hatayı çöz, iyi niyetli test et, gereksinimler olmadan test et, çok belirgin hataları raporlama, hataları yüzünden başkalarıyla alay et.

Test yaşam döngüsü nedir?

Analiz > Teknik Analiz > Development > Test > UAT > Production

Analiz > Teknik Analiz : Müşteriyi ve ürünü tanımak, ürün mevcutsa durum saptaması, risk analizi, test zamanlarının belirlenmesi, test planı oluşturmak, use case ve senaryolar hazırlamak.

Development : Senaryorları risk durumlarına göre sıralamak, senaryoları test case'ler ile detaylandırmak, birim testler yapmak, analizdeki değişiklikleri takip etmek.

Test : Smoke testler yapmak, test senaryolarını çalıştırmak, hataları raporlamak, çözülen hataları tekrar test etmek, riskli olan senaryoları tekrar test etmek.

UAT : Uygulamadaki tüm özellikleri müşteriye test ettirmek, çıkan hataları raporlamak, çözülen hataları test edip müşteriye onaylatmak, test raporu yayınlamak.

Test case oluştururken olması gereken adımlar nelerdir?

Set of inputs, execution conditions, expected outputs.

Test hazırlığı nasıl yapılır?

Analiz > Teknik Analiz : Müşteriyi ve ürünü tanımak, ürün mevcutsa durum saptaması, risk analizi, test zamanlarının belirlenmesi, test planı oluşturmak, use case ve senaryolar hazırlamak.

Sık kullanılan test yöntemleri nelerdir?

Fonksiyonel Olan: Unit Test, Integration test, User AcceptanceTesting, Web Services Testing, Migration Testing, Maintenance Testing

Fonksiyonel Olmayan: Performance Testing, Load Test, Stress Test, Compatibility Test, Security Test, Usability test, Localization Test

Temel test prensipleri nelerdir?

Test hataların varlığını gösterir.
Exahustive test mümkün değildir.
Test yaşam döngüsünde olabildiğince erken başlamalıdır.
Hatalar yazılımın belirli alanlarında yoğunlaşır. Defect kümelenmesi
Test içerik bağımlıdır.
Yeni hata bulunmaması hata yokluğu yanılgısıdır.
Test yaklaşımı projeye göre değişiklik gösterir.
Pesticide paradoksu

Web yazılımlarını test ederken, diğerlerine göre farklı olan unsurlar neler olur?

Hız, güvenlik, kolay üyelik, klavye kullanımı, hata mesajları, sadece gerekli bilgi.
Web uygulama testi, içerik testi, veritabanı testi, arayüz testi, kullanilabilirlik testi, uyumluluk testi, performans testi.

Server yazılımlarını test ederken, diğerlerine göre farklı olan unsurlar neler olur?

Server güvenliği ve sorguların hızlı çalıştırılabilmesi en önemli unsurdur. Hiyerarşik yapı ve yapılan işlemlerin iyi kaydedilmesi önemlidir. Veri kaybı riski çok mühimdir.

Bir yazılım için iç riskler ve dış riskler nelerdir?

Zaman riskleri: Projenin, yanlış görev ve malzeme paylaşımından dolayı beklenen süre içerisinde gerçekleşmesine veya tamamlanmasına engel olan veya sebep olan risk çeşididir.

Bütçe riskleri: Gerçekçi olmayan bütçe tahminleri sonucu finansal sorunlara yol açan risklerdir. Bu risklerin gerçekleşmesi durumunda Tablolar değişmekte, maliyetler artmaktadır.

Yönetim riskleri: Yönetim riskleri; amaçların net olmayışı, planlama eksikliği, yönetim tecrübesi ve eğitim eksikliği, iletişim sorunları, örgütsel sorunlar, otorite eksikliği ve kontrol problemlerini kapsamaktadır.

Teknik riskler: Genelde fonksiyonların yanlış olmasından kaynaklanır. Müşteri taleplerinin sürekli değişmesi, gelişmiş tekniklerin kullanılmaması ve geliştirilecek olan projenin zor faaliyetler içermesi gibi sebeplerden kaynaklanmaktadır.

Program riskleri: Proje kapsamının dışına çıkan, kontrol dışı durumlardan veya önceliklerin sürekli değişmesinden doğan risklerdir

Sözleşme ve yasal riskler: Sözleşme ve yasal riskler; değişen ihtiyaçları, pazar odaklı programları, sağlık ve güvenlik sorunları, hükümet düzenlemeleri ve ürün garantisi konularını içerir.

Personel riskleri: Personel riskleri; personel duraklamaları, deneyim ve eğitim sorunları, etik ve ahlak konularını, personel çatışmalarını ve verimlilik sorunlarını içermekedir.

Diğer riskler: Diğer kaynaklı riskler, mevcut olmayan veya geç teslim edilen ekipman ve sarf malzemeleri, yetersiz aracı, yetersiz tesisleri, dağıtılan bölgeleri, bilgisayar kaynaklarının olmayışı ve yavaş tepki sürelerini kapsamaktadır.

Bir yazılımın güvenli olması ile güvenilir olması arasındaki farklar nelerdir?

Yazılım güvenilirliği, sistem veya bileşenlerinin, belirli bir ortamda, belirli bir zaman dilimi içinde kendilerinden beklenilen işlevleri yerine getirebilme olasılığı olarak tanımlanmaktadır. Temel kavramları; hata, aksaklık ve arızanın tanımıdır.
Yazılım güvenliği, bilginin veya hizmetlerin istenmeyen veya yetkilendirilmemiş erişimden, değişimden veya yıkımdan korunmasıdır.

Gorilla ve exhausted testler arasındaki fark nedir?

Gorilla test: Belirli bir modül / bileşen veya performansın, çeşitli meşru ve yanlış bilgilerle büyük ölçüde tekrar tekrar test edilmesini içeren test stratejisi.
Exhausted test: Test için tüm olası veri kombinasyonlarının kullanıldığı bir test yaklaşımıdır. Keşif testi, testin başlangıcında yazılım / verilerin durumunda bulunan örtülü veri kombinasyonlarını içerir.

Tehlike ve risk arasındaki fark nedir?

Tehlike, zarara neden olabilecek bir şeydir.
Risk, herhangi bir tehlikenin aslında birisinin zarar görmesine neden olacak, yüksek veya düşük bir şanstır.