



Yazılım Testi

Bir yazılım ürünün testi ; ürünü son kullanıcıya teslim edilmeden önce yazılımın tüm yönleriyle kontrol edilmesidir.



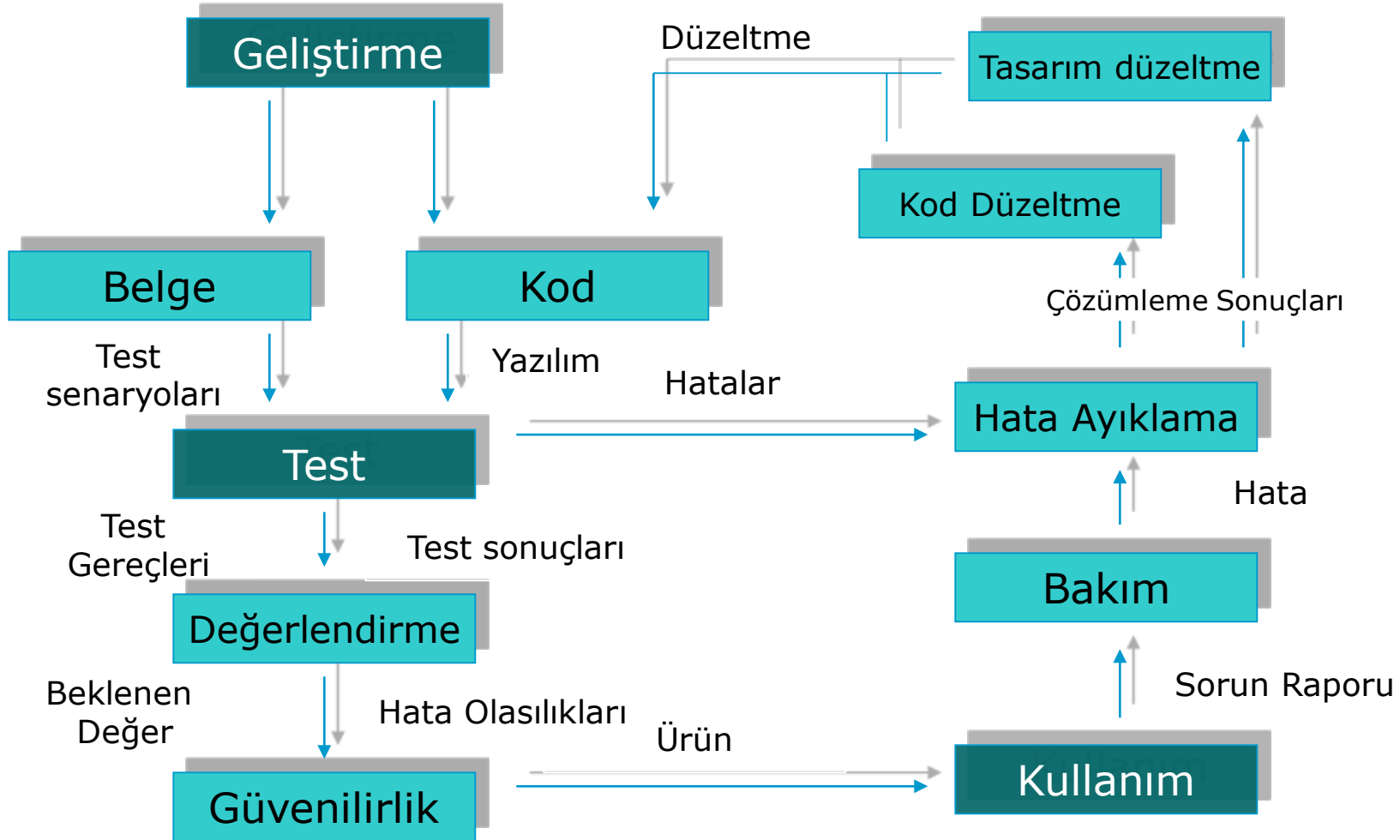
Sorunsuz ürün sunmaktır.

- **Deneme testleri**
- **Kabul testleri**

Test Yapılışı :

- **Yazılım isterleri belirtimi**
(Nelerin test edilmesinin istendiğini belirtir.)
- **Tasarım Belirtimi**
(Nelerin test edilmesinin gerektiğini belirtir.)
- **Kaynak Kod**
(En son üretim ve hata bulma amacıyla kullanılır.)
- **Test Ortamı**
(Hedef sistem donanımının bir benzeri ve kendisidir.)
- **Test Planlaması**
(Hangi testi ne zaman ve nasıl yapılacağını belirtir.)
- **Test Tanımlaması**
(Test senaryolarını ayrıntılı olarak anlatır.)
- **Test Yardımcı Gereçleri**
(Ölçüm yada veri giriş/çıkışı için kullanılır.)
- **Benzetim Araçları**
(Gerçek ortamdaki davranışları denetim altında oluşturabilmek üzere kullanılır.)

Test Aşamaları



Testin iki ana amacı vardır ;

Deneme (Prosedür testi)

Kabul testleri (Kullanıcı arayüz başarımı,güven)

Saydam Kutu Testi :

Sistemin işlevleriyle ilgili tüm yapısının ortaya serilmesiyle ilgili testtir.

İki Aşamada değerlendirme yapılır ;

- Tasarım tabanlı test (isterler baz alınır...)
- Kod Tabanlı test (Mantık,kodlama, akış yolu,Yazım ...)

Kara Kutu Testi :

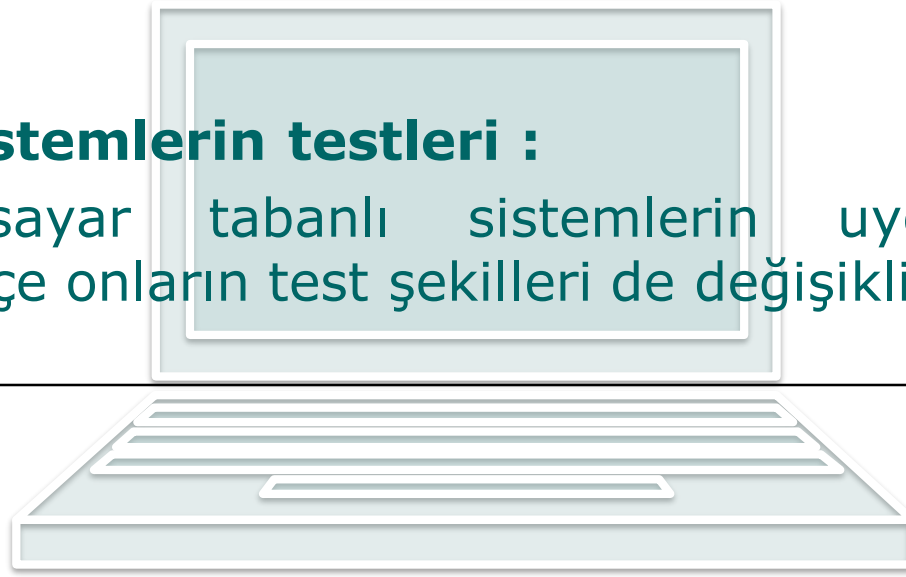
Yazılımın arayüzü düzeyinde yapılan testtir.

Belirlenen girdilerin doğru çıktıyı üretmesi beklenir.

Belirlenen sistem gereksinimleri esas alınarak kullanıcıya yönelik test durumları oluşturulur ve sonuçlar gözlemlenir.

Özel Sistemlerin testleri :

Bilgisayar tabanlı sistemlerin uygulama alanları değiştikçe onların test şekilleri de değişiklik gösterir.



- Gömülü sistemler
- Gerçek zamanlı sistemler
- Büyük veri tabanı sistemleri
- Güvenlik sistemleri
- Geniş paket yazılımları

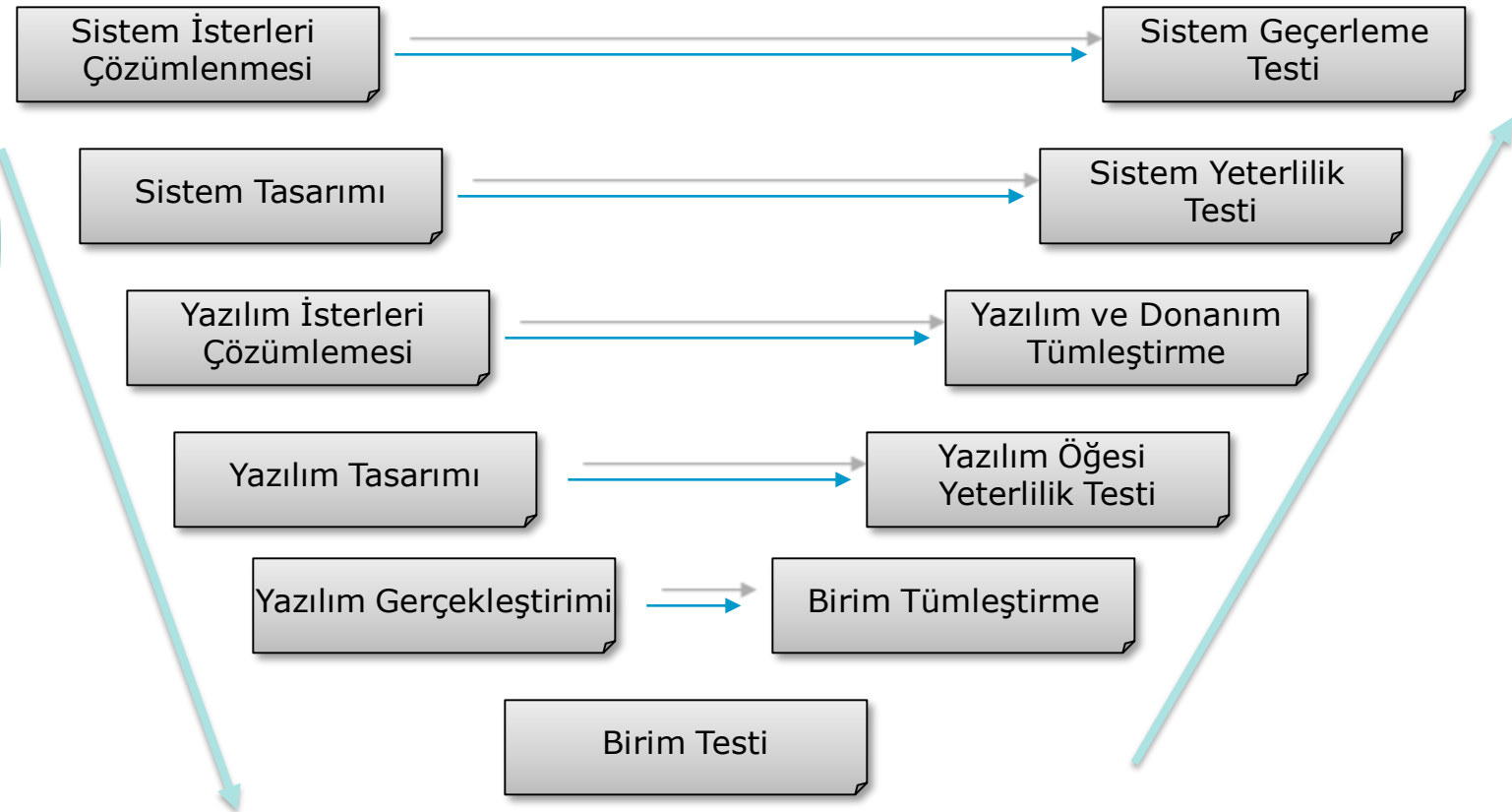
İsterleri karşılayacak şekilde test edildikten sonra kullanıma sunulmalıdır.

Otomatik Test Araçları :

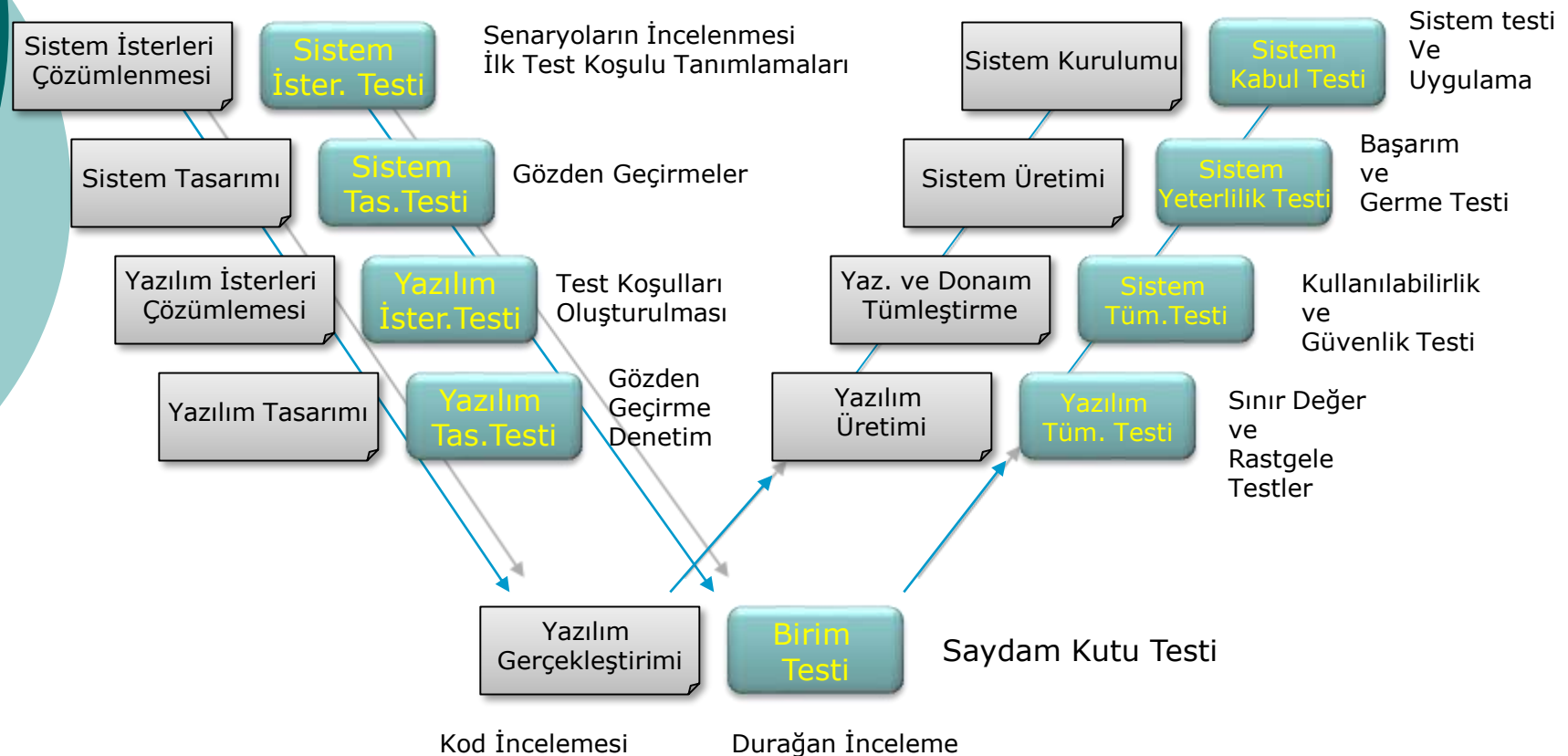
Yazılım testleri çok uzun ve maliyetli olduğu için işin bir kısmını kolay ve otomatik olarak gerçekleştiren araçlar kullanmak mümkündür.

- **Akıllı derleyiciler** (hassas tip kontrolü ...) ,
- **Durağan Çözümleyiciler** (Kaynak kod kontrolü ...),
- **Benzetim Ortamları** (Sanal ortam üzerinde test ...),
- **Girdi Dosyaları** (uygulama için gerekli verinin üretilmesi...),
- **Test Yazılımları** (benzer yazılımlar ile senaryolar üretmek...) ,
- **Simgesel Testler** (belirli algoritmaların sayısal testleri yapılır...),
- **Çevre Benzeticileri** (Çevre şartlarını sağlayan ortamda ...).

Sistem Geliştirmede "V" Modeli

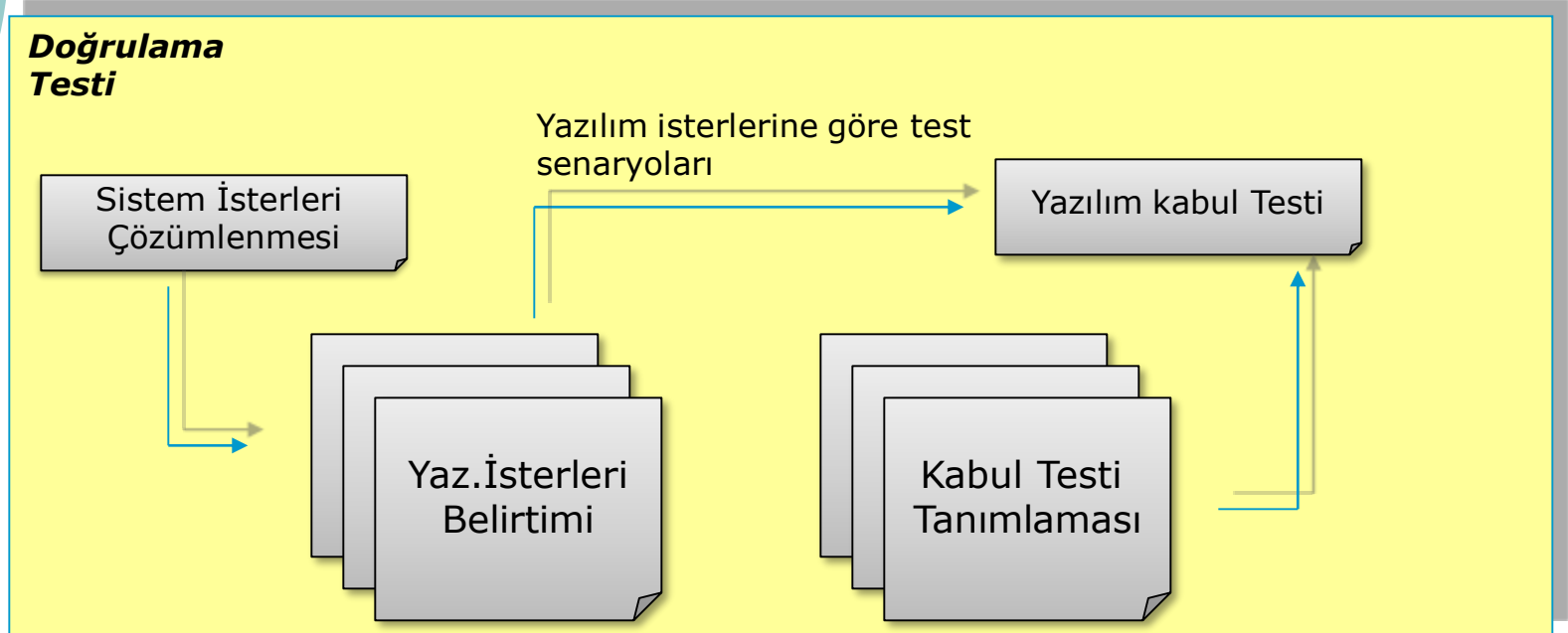


Sistem Geliştirme "W" Modeli

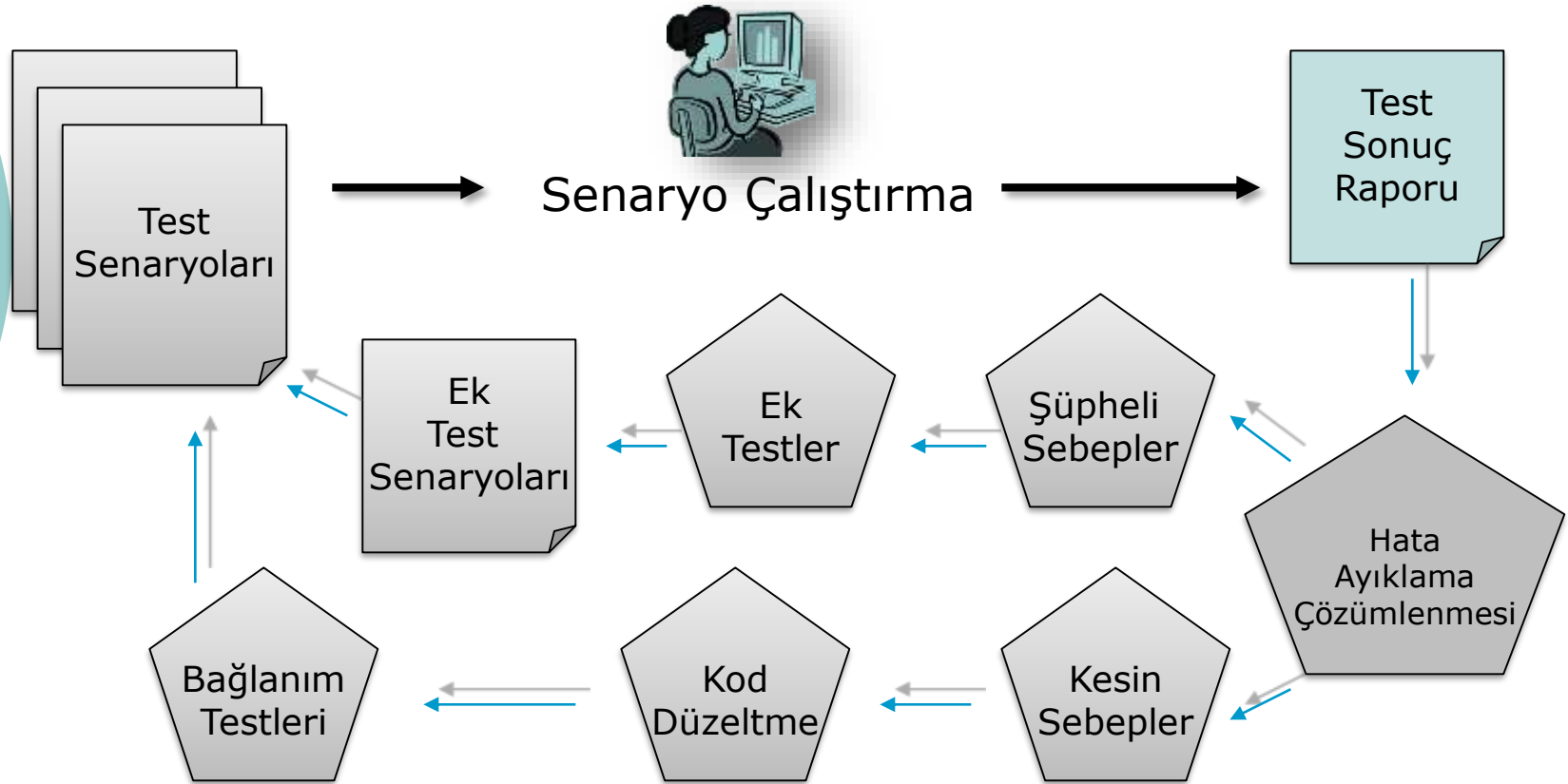


1. Birim Testi (En küçük birimlerin testi ...)
2. Tümlleştirme Testi (Yaz.ve Don.kay.birlikte çalışma testi ...),
3. Yeterlilik Testi (Tümlleştirme testin sonra yapılan testlerdir ...),
4. Sistem Testi (Sistemin tümü üzerinde yapılan testlerdir...)
5. Kabul Testleri (Müş.İsteklerinin kabulüne yönelik testlerdir...),

Test Yönetimi (Büyük Projeler için geçerli ...).



Hata Ayıklama : En büyük sorunlardan biri hata durumunda hangi bileşenin kusurlu olduğunun belirlenmesidir.



Entegrasyon testi;

Yazılım yeterlilik testlerinin gerçekleştirilmesi sonrasında sistem tarafında da çalışmalar Sistem Entegrasyon Testi Hazırlıklarını Yapma aşaması ile devam eder.

Entegrasyon test adımları:

- Test tanımları belirlenir
- Test araçları oluşturulur.
- Test araçları için donanım geliştirme sürecine uygun olarak planlama yapılır.
- Test dokümanları tanımlı sürece uygun olarak gözden geçirilir.
- Aşamalı olarak temin edilen birimler bütünleşik sistem elde edilene kadar ihtiyaca göre entegre ve test edilir.

- Test hazırlıklarının tamamlanması sonrasında Sistem/Alt sistem Testini Yap ve Entegrasyon Test Sonuçlarını Gözden Geçir aşaması gerçekleştirilir.
- Entegrasyon test kontrol adımları:
- Entegre edilen her test edilecek yapı üzerinde hazırlanan dokümanlara uygun doğrulama testleri uygulanır ve test sonuçları kaydedilir. Bu kayıtların analizi ve yorumlanması ile test raporu hazırlanır.
- Hazırlanan rapor tanımlı sürece uygun olarak gözden geçirilir.
- Donanım ve Yazılıma dayalı her birimin entegrasyonu tamamlanır ve entegrasyon doğrulanması amacıyla entegrasyon testleri gerçekleştirilir.
- Entegrasyon testleri kapsamında öncelikle donanım ve yazılım arayüzlerinin doğruluğu kontrol edilir. Arayüzler doğrulandıktan fonksiyonel ve performans testleri yapılır.
- Test sonuçları Sistem / Altsistem Test Raporu ile raporlanır. Test sonuçlarına göre birimlere ilişkin dokümantasyonda gerekli düzeltmeler yapılır.
-

Yeterlilik testi;

Sistem entegrasyon ve testleri tamamlandığında sistem işletme testlerinin yapılması ile sistemin doğrulanması sağlanır.

Yeterlilik testi için adımlar:

Sistem İşletme Test Planı içinde yer alan test gereksinimlerinin sağlanmasında kullanılacak olan test tanımları yazılır.

Sistem, gerçek kullanım ortamına uygun olarak test edilir ve raporlanır.

Raporlar gözden geçirilir ve doğrulanmış sistem elde edilir.

Sistem yeterlilik testlerine yönelik çalışmalar Sistem Gereksinim Özellikleri dokümanı hazırlandıktan ve gözden geçirme süreci tamamlandıktan sonra başlar.

Sistem İşletme Test Tanımı dokümanına her bir gereksinim için doğrulama metoduna uygun olarak test tanımları yazılır ve testler uygulanır.

Kabul testi;

- Tasarım ve doğrulama süreci tamamlanan sistemin müşteri isteklerini karşılayıp karşılamadığı kabul testleri ile ispat edilir. Müşteri tarafından belirlenmiş bütün gereklerin doğrulanması için muayene kabul dokümanı oluşturulur.
- Sistem bu dokümana göre önce şirket içerisinde doğrulandıktan sonra müşteri davet edilir ve testler müşteri ile birlikte tekrarlanır.
- Sistem kabul testlerinin, sistem özelliklerine bağlı olarak şirket içerisinde ve/veya arazide yapılması gerekebilir.

İyi bir test sürecini belirleyici bazı kriterler :

- İlk olarak temel fonksiyonların testini yapmak
- Yapılan son güncellemelerin testini yapmak
- Beceriye güvenilirlikten önce test etmek
- En fazla sorun olabilecek modüller den başlayarak testi planlamak
- Genel riskleri, özel durumlara ait risklerden önce test ederek
- Her aşamada test sonuçlarını ekip çalışanları ile paylaşmak
- Ürünü belirlenmiş bir plan dahilinde (temel işlerler, sınırlar, olması gereken, ...) teste devam etmek
- Genel durumları, özel durumlardan önce test ederek
- Ürünü talep eden işletmenin istek ve taleplerini önce test etmek
- Test sonuç değerlendirmesi yaparak, üretim süreci için önleyici ve iyileştirici çalışmalar planlamak
- Tüm süreç için kalite çerçevesinde tavsiyeler oluşturmak.

Yazılım Test Teknikleri :

Fonksiyonel Test

Fonksiyonel test, öncelikle sistemin yapması beklenen işlevlerine odaklanır. Tanımlı spesifikasyonların sağlanıp sağlanmadığının kontrol edilmesinin yanı sıra, tanımlı olmayan, düşünülmemiş durumları da incelenir. Özellikle, beklenmeyen koşullar altında sistemin davranışı saptanarak, olumsuz durumların varlığı aranır.

Gerçek kullanıcılar için data kaybı, runtime hataların alınması gibi durumlar güvenilirliği etkileyeceğinden fonksiyonel testlerin işletilmesi önem taşır. Test ekibi tarafından test altındaki uygulamanın operasyonları çok yönlü olacak şekilde analiz edilerek gerekli test teknikleri geliştirilir. Uygulamanın test kapsamının belirlenmesi, sistem noktalarının ortaya çıkarılması ile tüm fonksiyonel spesifikasyonların teste dahil edilmesi sağlanır.

Kullanıcı Arayüzü Testi

- Uygulamanın fonksiyonel işleyişini sağlayabilmek için kullanıcı arayüzünün bu doğrultuda tasarlanması ve kodlanması gereklidir. Uygulamadaki, ekranlar, menüler, butonlar, resimler, yazılar vs. kullanıcı arayüzü öğelerinin atlanmadan test edilmesi, pozitif ve negatif test adımlarında davranışlarının değerlendirilmesi gerekmektedir.
- Manuel ya da otomatik olarak test edilmesi ile gereksinimlerin karşılandığı görülüp, beklenmeyen sonuçların da ortaya çıkarılması sağlanır.

Test Otomasyonu

Test otomasyonu çeşitli tekniklerin dahil edildiği, başarı için ürdürülebilirliğin ve güvenilirliğin şart olduğu bir alandır. Test otomasyonunun başarılı olabilmesi için öncelikle uygun best practice'lerin seçilmesine önem verilmelidir.

Test otomasyonunun sağladığı yararları şöyle sıralayabiliriz;

- 1- Zamandan tasarruf edebilmek
- 2- Manuel testlerde harcanan insan gücünü azaltıp, proje maliyetlerini düşürebilmek
- 3- Test edilebilirliği artırmak
- 4- Tekrar test edilebilirliği sağlamak

Uygun test aracının seçilmesi önem taşıdığı için otomasyon projelerinde müşteri yazılımlarına göre ihtiyacın belirlenmesi ve implementasyon gereksinimlerinin ortaya net bir şekilde konması gerekir.

Bu sayede süreçlere bağlı, doğru test planlaması yapılarak testler hızlandırılabilir.

Manuel Test

- Manuel testler, test ekibinin test senaryolarını belirleyerek, ihtimam ile test adımlarını işletmeleri ile yürütülür. Sistematiik yaklaşım ile yürütölen testler, her adımda gerekli detaylandırma yapılarak, otomasyon ile test edilmesi uygun olmayan testlerde dahil edilerek yapılır.
- Bir test mühendisi, her zaman çantasında taşıdığı, merak, sabır, çözüm takip edebilme erdemleri ile daha fazla hata yakalayabilmeyi amaçlamalıdır.

Performans Testi & Yük Testi

- Çeşitli iş yükleri altında sisteminizin performansınızı ölçmeniz gerekecektir. Beklenen işlevlerin yerine getirilip getirilemediği, hangi noktalarda darboğazlar oluştuğu, başarısız senaryo adımları değerlendirilerek performans analizi yapılmalıdır.
- Test ekibi mevcut sistemin performans senaryolarını oluşturarak, client/server tarafında belirlenen metriklerin değerlerini toplayıp, gerekli analiz sonuçlarını ilgili birimlere sunarak, tahminlere dayalı geliştirmemelerini sağlamalıdır.
- Test mühendisleri, özel lisanslı/açık kaynak performans test araçlarını kullanarak, tanımlanan sayıda sanal kullanıcılar ile gerçek ortam koşullarını simüle edebilmektedirler.
- İhtiyaçları karşılayabilecek en uygun test aracının belirlenmesi ile test senaryoları çıkarılabilmektedir. Yük tanımlarının işletilmesi ile de yük altındaki sistemin detaylı performans istatistikleri belirlenmektedir.

- **Lokalizasyon Testi**
- Test altındaki uygulamanın farklı yönlerden belirgin bir kültür veya lokal odaklı testlerinin yapılabilmesi gerekmektedir. Lokalizasyon testinde; desteklenmesi beklenen diller, kullanım durumları/alışkanlıkları, bölgesel saat/tarih ayarları gibi öğelerin test kapsamına alınıp, dikkatli hazırlanacak test planları ile teste tabi tutulması sağlanır.
- Çeşitli pazarları hedefleyen müşteriler, bu testler sayesinde iş gereksinimlerinin karşılanıp karşılanamadığını görebilmiş olur.

Kara-kutu Testi

- Bu tür testlerde yazılımın programatik yapısı, tasarımı veya kodlama tekniği hakkında herhangi bir bilgi olması gerekli değildir. Yazılımın gereksinme duyulan şeylere yanıt verip veremediği ve işlevselliği sınanmaktadır.

Verim (Performans) Testi

- Yukarıda da belirtildiği gibi, bu test 'zorlanım' ve 'yük' testi ile eş anlamlı olarak ta kullanılabilir. Ancak, yapılması gereken performans testinin ne olduğunun gereksinimler veya kalite güvencesi veya test planlarında açıklanmış olmasıdır.

Kullanışlılık Testi

- Tahmin edilebileceği üzere, kişisel yargılara göre değişen bir test olup hedeflenen son kullanıcı veya müşteri kitlesine bağlı olarak değişir. Kullanıcı yorumları, kullanıcı oturumlarından video kayıtları veya diğer teknikler kullanılabilir. Programcılar ve test uzmanları genellikle bu tür testler için uygun değildir, yani bu testlerin doğrudan son kullanıcılar üzerinde yapılması gerekir.
- Bu test genellikle geliştirme sürecinin erken aşamalarında yapılır, böylelikle uygulamanın kullanıcı arayüzlerinde önemli değişiklikler yapılması mümkün olur. Test yürütücüsünün hedeflenen kullanıcılar üzerinde test yapabilme yeteneği önemlidir.

Güvenlik Testi

Yazılımın, gerek iç ve gerekse dış kaynaklı yetkisiz erişimlere, kötü amaçlı kullanımlara karşı korunması ya da güvenliğini test etmek için yapılır. Çok karmaşık ve özel test tekniklerinin kullanıldığı bir test türüdür.

Uyumluluk Testi

Yazılımın özel bir donanım, yazılım, işletim sistemi, ağ veya ağ protokolü vb. ortamda beklenen şekilde çalışıp çalışmadığını sınamak için yapılan testlerdir.

Doyum Testi

Yazılımın, son kullanıcı veya müşteri tarafından beğenilip beğenilmediğini, ya da ihtiyaçlarını karşılayıp karşılamadığını belirlemek için yapılır.

Kurma/Kaldırma Testi

Bu test, yazılımın kurulması ve kaldırılması ile ilgili tüm seçenekler ve özelliklerin düzgün şekilde çalışıp çalışmadığını sınamak için yapılır. Kurulumda, tüm gerekli dizinler ve bunlarda yer alacak dosyaların (.dll, .cfg, .txt vb) oluşturulması gereklidir. Ayrıca yazılımın sistemden kaldırılması sırasında herhangi bir artık kalmamalıdır.

Ağ Testi

Çok kullanıcıli uygulamaların ağ ortamında gerçekten ağ üzerinde çalışabilme yeteneklerini ortaya koymak için yapılan bir testtir. İstenirse, farklı ağ işletim ortamları ve iletişim kuralları altında test yapılması tercih edilmelidir.

Beyaz-kutu Testi

Bu tür testler, uygulama kodunun iç mantığı üzerindeki bilgiye bağlıdır. Yazılım kodundaki deyimler, akış denetimleri, koşullar vb elemanlar sınanır.

Birim Testi

Mikro ölçekte yapılan bu testte, özel fonksiyonlar veya kod modülleri test edilir. Bu test, test uzmanlarınca değil programcılar tarafından yapılır ve program kodu ayrıntılarına ve içsel tasarım biçiminin bilinmesi gereklidir. Uygulama kodu çok iyi tasarlanmış bir mimaride değilse oldukça zor bir testtir.

Artımsal Tümleştirme Testi

Uygulamanın yeni işlevsel elemanları eklendikçe sürekli test edilmesidir. Bu testte uygulamamanın tüm parçaları tamamlanmadan önce yeni eklenen parçanın işlevselliğin öncekilerden yeteri ölçüde bağımsız şekilde çalışıp çalışmadığı sınanmaktadır. Test uzmanları ve/veya programcılar tarafından yapılan bir testtir.

Tümleşim Testi

Bir uygulamanın farklı bileşenlerinin beraberce uyum içinde çalışıp çalışmadığını sınamak için yapılan bir testtir. Bileşenler, modüller, bağımsız uygulamalar, istemci/sunucu uygulamaları biçiminde olabilirler. Bu tür testlere, özellikle istemci/sunucu uygulamaları ve dağıtık sistemlerin testinde başvurulmaktadır.

İşlevsellik Testi

Bir uygulamanın işlevsellik gereksinimleri üzerine odaklandırılan kara-kutu testidir. Bu tür testler, test uzmanları tarafından yapılır, ancak bu uygulama yayınlanmadan önce kodların programcılar tarafından incelenmeyeceği anlamına gelmez. Testin herhangi bir aşamasında program kodlarının da incelenmesi gerekir.

Regresyon Testi

Uygulama ve uygulama ortamlarında gerekli değişiklikler ve sabitlemeler yapıldıktan sonra yeniden yapılan testlere çekilme (regresyon) testi denilir. Böylece, önceki testlerde belirlenen sorunların giderildiğinden ve yeni hatalar oluşmadığından emin olunur. Uygulamanın kaç kez yeniden test edilmesi gerektiğini belirlemek güçtür ve bu nedenle, özellikle uygulama geliştirme döneminin sonlarına doğru yapılır.

Kabul Testi

Son kullanıcı veya müşteri siparişi (veya isteklerine) dayanan son test işlemidir. Ayrıca, son kullanıcıların belli bir süre kullanımlarından elde edilen sonuçlar üzerinde de yapılabilmektedir.

Yük Testi

Uygulamanın çok ağır yükler (veya işlem yoğunluğu) altında test edilmesidir. Örneğin, bir Web sitesi için sistem tepkisinin hangi noktada azaldığı veya yanıt veremez olduğunu belirlemek için yapılan testler gibi.

Alfa Testi

Bitirilme aşamasına yaklaşmış olan bir uygulama için yapılan testtir. Bu test sonucunda ürün üzerinde küçük değişiklikler yapılabilir. Programcılar veya test uzmanlarınca değil, son kullanıcılar tarafından yapılır.

Beta Testi

Uygulamanın tamamlanması ve zorunlu testleri yapıldıktan sonra, son sürümü çıkarmadan önce hatalar ve/veya sorunları saptamak üzere yapılan testlerdir. Programcılar veya test uzmanlarınca değil son kullanıcılar tarafından yapılır.

Belgelendirme : Plan dahilinde yapılan belgelendirmelerdir. Sistemin bir bütün olarak nasıl test edileceği, doğrulanmasının ne şekilde yapılacağı, bunun için gerekli olan yardımcı araçlar bir planda belirtilir.

Riskler : İyi hazırlanmadan başlanan bir testin önce kendisi büyük bir risk oluşturur.

Temel riskler ;

1. **Tümleştirmede karmaşa**
2. **Plan dışı yürütme**
3. **Yordamların yetersizliği**
4. **Sıralama**
5. **Paralel test işlemleri**
6. **Yüksek maliyetli testler**
7. **Bazı testlerin yapılmaması**

