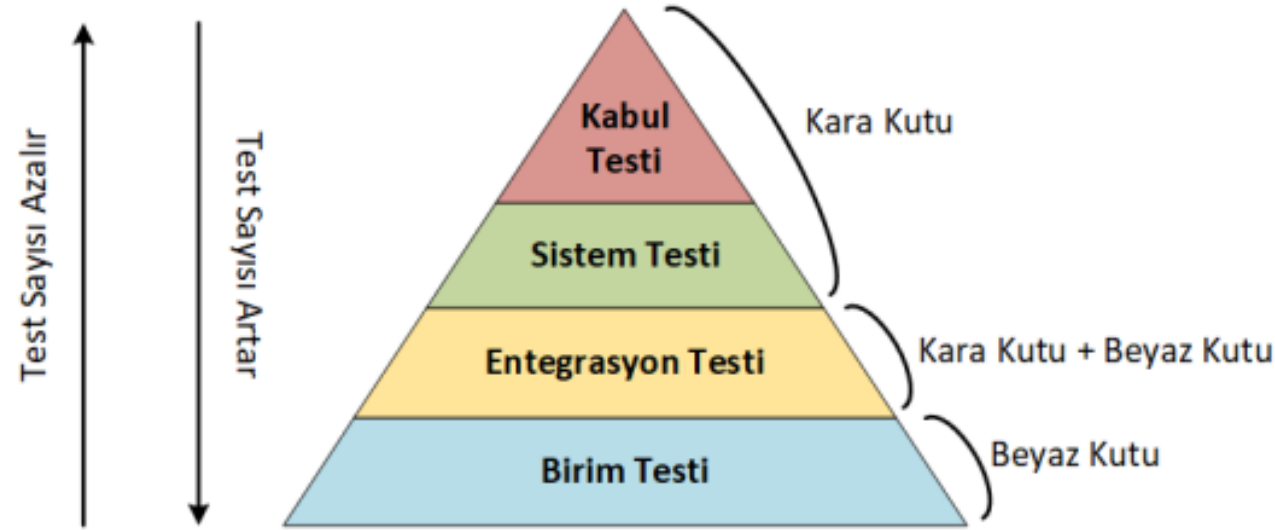


# Yazılım Test Mühendisliği

Dr. Öğr. Üyesi Hasan YETİŞ

# YAZILIM TEST SEVİYELERİ

- Test seviyelerinin amacı, yazılım testinin sistematik bir hale getirilmesi ve belirli bir seviyedeki olası tüm senaryoların kolayca belirlenip kontrol edilmesini sağlamaktır.



Şekil 5.1. Yazılım test seviyeleri

# Birim Testi

- Modül Testi, Bileşen Testi ve Unit Test isimleriyle de bilinir.
- Sistemin derlenebilir, çalıştırılabilir en küçük parçasının test edilmesidir.
- Geliştiriciler tarafından gerçekleştirilir.
- Birim testinin sağlıklı sonuçlanması, sistemin ileri aşamalarında gerçekleştirilecek olan diğer test seviyeleri ve türleri için zaman ve maliyet kazancı sağlar

# Entegrasyon (Birleştirme) Testi

- Birleştirme (entegrasyon) testi, birbirinden bağımsız test edilen ve başarısı doğrulanan iki modülün, birbirine bağlanması ile **birlikte çalışabilirlik düzeyinin test edilmesidir**.
- Entegrasyon testi ile farklı modüllerin geçişleri sırasında arayüzlerin davranışları ve yine farklı modüller arasındaki veri akışı kontrol edilir.
- Entegrasyon testi dört farklı yaklaşım ile gerçekleştirilir.

# Entegrasyon (Birleştirme) Testi

## **Büyük Patlama Yaklaşımı**

- Big Bang Testi olarak da bilinir.
- Bu yöntemde, birleştirilecek modüllerin testleri bir arada yapılır.
- Bu yaklaşımın kullanımı için küçük projelerin uygun olduğu savunulur.
  - Çünkü entegrasyon sırasında bulunan bir hatanın, potansiyel olarak entegre edilen modüllerden herhangi birine ait olduğu bilinse de hatanın yerelleştirilmesi oldukça zordur.
  - Bu da hata lokalizasyonu konusunda güçlük yaşanacağını gösterir.

# Entegrasyon (Birleştirme) Testi

## **Yukarıdan Aşağıya Yaklaşım**

- Bu test tekniği gerçekleştirilirken, yazılım sisteminin kontrol akışı için belirtilen adımlar, yukarıdan aşağıya entegrasyon yapılmasıyla sağlanır.
- En üst düzey birimlerin önce test edildiği ve alt düzey birimlerin bundan sonra adım adım test edildiği bir yaklaşımdır.

# Entegrasyon (Birleştirme) Testi

## **Aşağıdan Yukarıya Yaklaşım**

- Bu test tekniği gerçekleştirilirken, yazılım sisteminin kontrol akışı için belirtilen adımlar, aşağıdan yukarıya entegrasyon yapılmasıyla sağlanır.
- Her bir modül test edildikten ve başarısı kesinleştikten sonra bir üst seviyede bulunan modül ile birleştirilerek tekrar test işlemi gerçekleştirilir.

# Entegrasyon (Birleştirme) Testi

## **Hibrit Entegrasyon Testi**

- Bu test yaklaşımı, yukarıdan aşağıya ve aşağıdan yukarıya test tekniklerinin birlikte kullanılmasıyla gerçekleştirilir.
- Bu yaklaşım, birkaç alt projesi olan çok büyük projeler için kullanışlıdır.
- Hibrit test yöntemi, test esnasında hem alt modüllerin hem de ana modüllerin eş zamanlı yürütülmesinden dolayı yüksek maliyet gerektirir.



# Sistem Testi

- Bu test tekniğinde sistem bir bütün olarak ele alınır ve test edilir.
- Tamamı entegre edilen bir yazılım ürününün testidir.
- Sistemin işleyişinin doğruluğu, entegrasyon testinde gözden kaçabilecek hataların keşfi için yüksek kalitede bir ürün sunmak adına gerçekleştirilir
- Sistem testi, test uzmanları tarafından, eksiksiz ve tam entegre bir yazılım ürününün doğrulanması için gerçekleştirilen seviyedir.
- Ürünün iş gereksinimlerini karşıladığından emin olmalarını ve işletim ortamında sorunsuz bir biçimde çalıştığını belirtmeleri için önemli bir adımdır

# Kullanıcı Kabul Testi

- Bu test seviyesi, son kullanıcılar ve test ekibi tarafından gerçekleştirilir
- Bu test seviyesinde amaç, sistemin kullanıcı kriterlerine uygunluğunu ve kullanıcı tarafından kolay kullanılabilir olduğunu test etmektir.
- Yazılım, son süreçte yer alan kullanıcı kabul testi ile Alfa, Gama ve Beta testlerine tabi tutulmaktadır.

# Kullanıcı Kabul Testi

## **Alfa Testi**

- Sınırlı sayıdaki gerçek kullanıcıların dahil edildiği, geliştiriciler ve test ekibinin gözetim ve kontrolü altında gerçekleştirilen kullanıcı kabul testi aşamasıdır.
- Alfa testinde sistem, müşteri ya da son kullanıcı tarafından geliştirme ortamında test edilir.
- Yazılım sistemleri, Alfa testinde başarılı çıktıktan sonra Beta testi gerçekleştirilir.
- Bu aşamada, sistemde kullanıcı isteklerine uymayan, kolay kullanıma engel olan ve sistemde var olan sorunlar düzenlenir.

# Kullanıcı Kabul Testi

## **Beta Testi**

- Alfa testinin başarıyla sonuçlanmasından sonraki aşamadır.
- Sınırlı sayıda gerçek kullanıcı tarafından kullanıcı ortamında yürütülen bir test çalışmasıdır.
- Bu aşamada temel amaçlarından biri, farklı yazılım ve donanım konfigürasyonları, ağ bağlantı türleri ile yazılım uyumluluğunun doğrulanmasıdır.
- Beta testinin bir diğer amacı, yazılımın kullanılabilirliği ve işlevselliği hakkında kullanıcılardan geri bildirimler almaktır.

# Kullanıcı Kabul Testi

## **Gama Testi**

- Yazılımın tamamlanmış sürümünden önce gerçekleştirilen son aşamada yapılan testtir.
- Gama testinin odak noktası sistemin güvenliği ve işlevselliğidir.
- Yalnızca sınırlı sayıda kullanıcı tarafından gerçekleştirilir.
- Sistemin yalnızca belirli noktaları test edilir.
- Gama testinde tespit edilen hatalar, yüksek öncelik ve önem derecesine sahip olmadığı sürece sistemde bir değişiklik gerçekleştirilmez
  - Fark edilen eksiklik veya hataların düzeltilmesi, sisteme sonraki sürümler için güncelleme olarak yansır.

# YAZILIM TEST TÜRLERİ

- Web, masaüstü ve mobil tabanlı yazılım uygulamalarının birçok farklı test türleriyle ölçülmesi mümkündür
- Ancak her yazılım için bütün test türlerinin uygulanması mümkün değildir.
- Her yazılım önemlidir, ancak birbirlerine kıyasla önem derecesi aynı değildir.

# YAZILIM TEST TÜRLERİ

- Fonksiyonel ya da İşlevsel Testte dikkat edilen husus, ürünün kullanıcı isteklerini karşılaması sonucu kaliteli bir ürün elde etmektir.
- İşlevsel olmayan yazılım testlerinde, bir yazılım sisteminin performans, kullanılabilirlik, güvenlik gibi işlevsel olmayan özellikler test edilir. İşlevsel olmayan testler yapılarak, ürünün kullanılabilirliği, verimliliği, sürdürülebilirliği, taşınabilirliği arttırılır.
- Uluslararası Yazılım Test Yeterlilikler Kurulu'nun (ISQTB) tarafından yapılan araştırmaya göre en çok kullanılan testlerde %92 oranla ilk sırada işlevsel test gelmektedir. En yakın %64 oran ile performans testi ve sonrasında %62 oran ile kullanılabilirlik testi takip etmektedir

# YAZILIM TEST TÜRLERİ

Arayüz Testi	İki veya daha fazla sistemleri arasındaki iletişimin sağlıklı bir şekilde gerçekleştirilmesini, yazılım modülü arasındaki bağlantıyı doğrulamak için gerçekleştirilir. En önemlilerden biri web ve uygulama sunucusunun arayüzü, diğeri ise uygulama ve veri tabanı sunucusunun arayüzüdür.	Veri akışı, Modüller arası iletişim, Modüller arası bilgi alışverişi.
Çapraz Tarayıcı Testi	Genellikle web siteleri gibi dinamik olan ve internet bağlantısı gerektiren sistemler için farklı tarayıcılardaki çalışma durumları test edilir.	Temel işlevler, Grafiksel kullanıcı arayüzü, İşlevin yanıt verme süresi ve çalışma hızı.



# YAZILIM TEST TÜRLERİ

Dayanıklılık Testi	Performans sorunlarının tespit edilmesi amacıyla uzun süre büyük değerlerde veri girişi kullanılarak olası hataların ortaya çıkması amacıyla uygulanır.	Geçersiz veri girişi, Yoğun şekilde veri girişi.
Depolama Testi	Sistemin veri tabanı test edilir. Verilerin uygun dizinlerde saklanması, yetersiz disk alanlarının hangi şartlarda ortaya çıktığı incelenir.	Veri formatları, Veritabanı tabloları.
Erişilebilirlik Testi	Uygulamanın renk körlüğü, işitme engeli gibi özelliklere sahip bireyler tarafından erişilebilirliğini ve kullanım kolaylığını test etmek için şartlara uygun bireyler tarafından gerçekleştirilmesi gereken bir yazılım testidir.	Farklı özelliklere sahip bireyler için arayüzlerin kullanım kolaylığı ve arayüzler arası geçişler.

# YAZILIM TEST TÜRLEİİ

Eşzamanlılık Testi	Sisteme birden fazla kullanıcının giriş yapması durumunda sistemin davranışı incelenir ve var olan hatalar tespit edilir.	Kullanıcılara verilen yanıt süresi, Çıktıların doğruluğu, Görev paylaşımı.
Güvenlik Testi	Geliştirilen sistemin, herhangi bir açıklık veya tehlikeden ötürü erişim sorunu yaşanmasını veya büyük oranda veri kaybına neden olmasını engellemek amacıyla gerçekleştirilir.	Ağ güvenliği, Erişim izinleri, Verilerin izinsiz kişilerce yayılması ve aktarılması.
Hacim Testi	Sistemin büyük miktarda veri girişine karşı gösterebileceği davranış ve yanıt verme süresinin ölçümünde kullanılır.	Yoğun veri girişi sırasında form işlemleri, Veri girişine karşılık yanıt verme süresi.

# YAZILIM TEST TÜRÜLERİ

Hata Enjeksiyon Testi	Olası hataların ortaya çıkabilmesi, hatanın boyutlarının ve risklerinin fark edilmesi, sistemin hata durumunda davranışının görülebilmesi için kod bloğuna hatalı kodların yerleştirildiği bir test türüdür.	Modül içerisinde yapay hata oluşturmaya bağlı sistem davranışı.
Performans Testi	Donanım ürünleri yardımıyla sistemin performansını etkileyecek olumsuzlukları en aza indirmeyi amaçlar. Hata tespiti için değil, sistemin verimliliği test edilir.	Çalışma hızı, Kapladığı alan, Eşzamanlı kullanıcı sayısı, Batarya kullanımı, Ortalama yanıt verme süresi, CPU kullanımı.

# YAZILIM TEST TÜRLERİ

Senaryo Testi	Farklı senaryolara bağlı kalarak büyük ve karmaşık yapıdaki yazılımların tüm modüllerinin test edilmesidir.	Senaryo baz alınarak modüller arasındaki geçiş, Veri alışverişi, Arayüzler, Kullanıcı girdileri.
Yol Testi	Çoklu giriş ve çıkışa sahip karmaşık sistemlerde bulunan döngülerin veya koşullu ifadelerin tüm yollarının test edilmesi için kaynak kodun incelendiği bir test çeşididir.	Tekrarlı, koşullu ve döngüsel ifadelerdeki tüm olası durumlar.

# YAZILIM TEST TÜRLERİ

Kullanılabilirlik Testi	Kullanıcı-sistem etkileşimi sırasında kullanıcı davranışlarını ve duygusal tepkilerini gözlemleyerek yazılımın kolay kullanılmasını ve müşteri memnuniyetinin arttırılmasını sağlar.	Arayüzlerin kullanım kolaylığı Arayüzler arası geçişler.
Kova Testi	Genellikle web tabanlı sistemlerde gerçekleştirilir. En az iki farklı arayüz varyasyonları arasından kullanıcı kriterlerine en uygun olanı seçmek için gerçekleştirilir.	Yazı tipleri, Bilgi grafik görüntüleri, Renk düzenleri, Buton konumları ve tasarımları, Form ekranları.

# YAZILIM TEST TÜRLERİ

Kurulum Testi	Yazılım sisteminin tüm özellikleri ve donanımsal araçlarıyla birlikte doğru bir şekilde kurulduğunu veya tüm bileşenleri ile kaldırıldığını kontrol etmek için gerçekleştirilir.	Ağ sistemleri, Donanım aygıtları, İşletim sistemleri.
Uyumluluk Testi	Bu testin gerçekleştirilmesindeki esas amaç, sistemin her türlü platform, konfigürasyon, donanım ve yazılıma gelebilecek eklentilerle uyum içinde başarılı bir işleyişe sahip olduğundan emin olmaktır.	Donanımsal aygıtlar, Ağ sistemleri, İşletim sistemleri, Yazılım sürümleri, Farklı tarayıcılar.



# YAZILIM TEST TÜRLERİ

Kullanıcı Arabirim Testi	Sistemin grafik arayüzlerinin kullanıcı ve test ekibi tarafından test edilmesidir. Amaç, uygulamanın hiyerarşik ve grafiksel ön yüzünün test edilmesi, farklı cihazlarda grafiksel görünümün sabitliğinin test edilmesidir.	Menüler, Butonlar, Görsel destek elemanları, Simgeler, Sayfalar arasındaki akış.
Karşılaştırma Testi	Mevcut yazılım ve gelecekteki yazılım arasındaki iyileştirilmelerin belirlenmesi amacıyla piyasada bulunan aynı amaçla kullanılan farklı yazılımları belirli kriterler açısından kıyaslar.	Dokümanlar, Veritabanı içerikleri, Metrik değerleri.

# YAZILIM TEST TÜRLERİ

Konfigürasyon Testi	Sistemin çalışabileceği en uygun yazılım sistemleri ve donanım araçlarının belirlendiği test türüdür. Bu testin yapılması sırasında bir çok yazılım ve donanım kombinasyonu gerçekleştirilir.	Sistemin çalışması için en uygun işletim sistemi, Web tarayıcısı, En uygun işlemci hızı gibi teknik bilgiler.
Uygunluk Testi	Uygunluk testi, sistemin büyük ve bağımsız kuruluşlar tarafından tanımlanan <i>IEEE</i> , <i>W3C</i> ya da <i>ETSI</i> gibi büyük ve bağımsız kuruluşlarca belirlenen standartlara uygunluğunun uzman bir test ekibi tarafından kontrol edilmesidir.	Sisteme ait doküman, şartnameler ve gereksinim belgelerinin uygunluğu, Sistemin belirlenen standartlara uygunluğu.



# YAZILIM TEST TÜRÜLERİ

Maymun Testi	<i>Rastgele test</i> olarak da bilinen bu test türünde, yanlış veya hatalı bilgilere karşı sistemin performansı ve davranışı test edilir. Herhangi bir test senaryosu olmadan gerçekleşen testte farklı eylemlere karşı sistemin çökme durumu gözlemlenir.	Form girişleri, Beklenen ve girilen veri formatları, Kullanıcı bilgileri.
Paralel Sınama Testi	Farklı iki yazılım/sürüm veya test senaryosu üzerinde testlerin paralel olarak gerçekleştirilmesini amaçlar. Genellikle sürümler arasındaki farkların ve güncel sürümün performans kriterlerine uygunluğunun test edilmesinde kullanılır.	Sistemin farklı tarayıcılardaki davranışları, Farklı sürümlerdeki veri formatları arasındaki tutarlılık.

# YAZILIM TEST ARAÇLARI

- **TestNG, JUnit** – Birim test
- **Cypress** – Javascript
- **Appium** – UI Test (iOS, Android, Tizen), browser (Chrome, Firefox, Safari), desktop (macOS, Windows), TV (Roku, tvOS, Android TV, Samsung)
- **Jira** – Test Süreç Yönetimi
- **Jenkins** – Sürekli Entegrasyon
- **Selenium** – Web test
- **Jmeter** – Yüksek trafik performans