



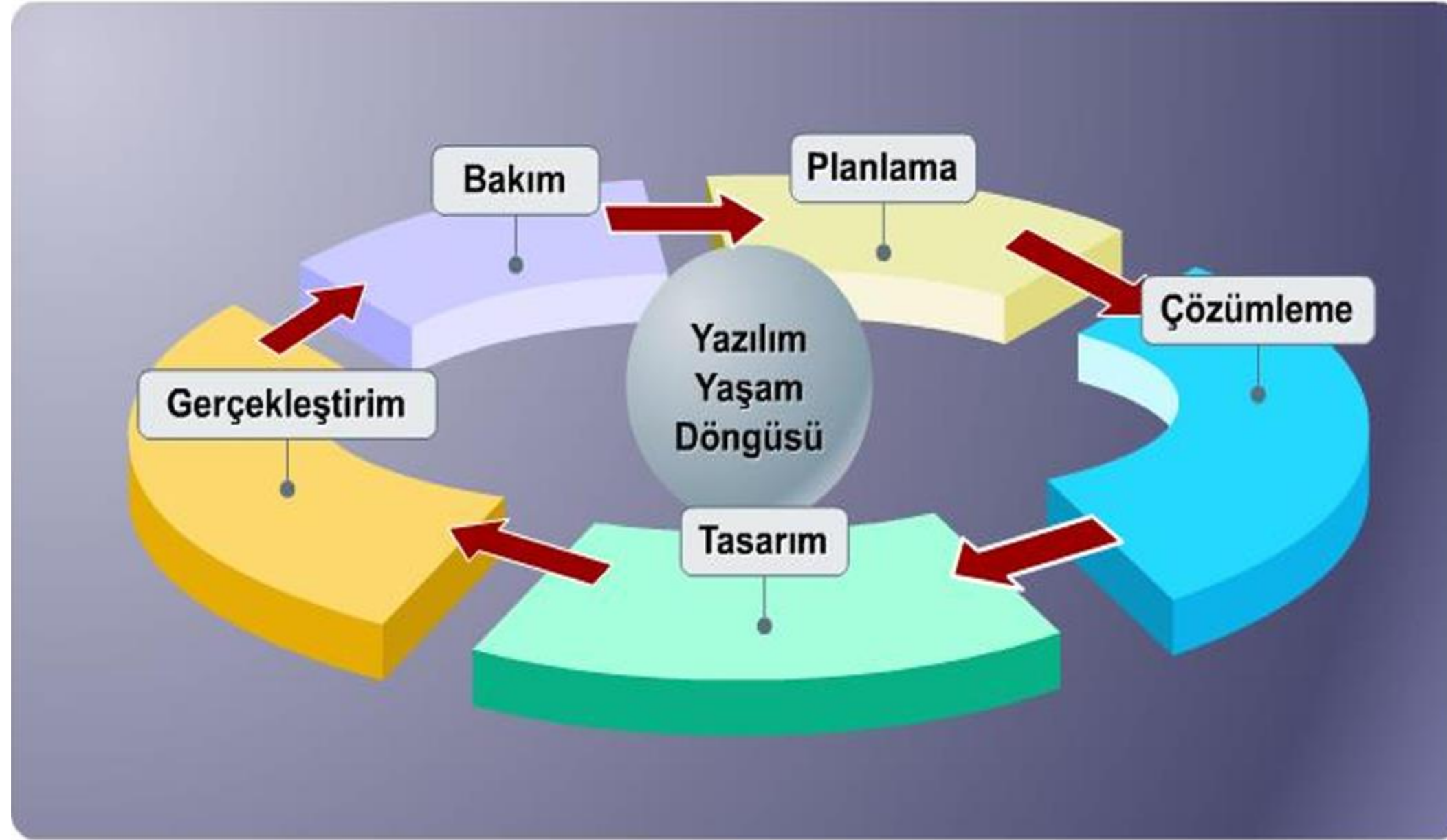
# YAZILIM GEÇERLEME VE SINAMA

Yazılım Geçerleme ve  
Sinamaya Giriş

# Yazılım Yaşam Döngüsü

- Yazılım yaşam döngüsü, herhangi bir yazılımın, üretim aşaması ve kullanım aşaması birlikte olmak üzere geçirdiği tüm aşamalar biçiminde tanımlanır.
- Yazılım işlevleri ile ilgili gereksinimler sürekli olarak değiştiği ve genişlediği için, söz konusu aşamalar bir döngü biçiminde ele alınır.
- Döngü içerisinde herhangi bir aşama da geriye dönmek ve tekrar ilerlemek söz konusudur.
- Yazılım yaşam döngüsünün tek yönlü ve doğrusal olduğu düşünülmemelidir.

# Yazılım Yaşam Döngüsü



# Doğrulama ve Geçerleme Yaşam Döngüsü

- Gerçekleştirim aşamasına kadar olan süreçlerde doğrulama ve geçerleme işlemlerinin planlaması yapılır.
- Planlama genellikle;  
alt-sistem,  
bütünleştirme,  
sistem ve  
kabul sinamalarının  
tasarımlarını içerir.
- Gerçekleştirim aşamasının sonunda ise söz konusu plan uygulanır.

Geliştirilecek bilgi sistemi yazılımın doğrulanması ve geçerlemesi işlemi üretim süreci boyunca süren etkinliklerden oluşur. Bu etkinlikler;

- Her bir etkinlik sonunda alınan çıktıların tamam, doğru, açık ve tutarlı olduğunun doğrulanması
- Her etkinlikte ürünün teknik yeterliliğinin değerlendirilmesi ve uygun çözüm elde edilene kadar aktivitelerin tekrarlanması
- Geliştirilen belirtilmelerin önceki belirtilmelerle karşılaştırılması
- Yazılım ürünlerinin tüm uygulanabilir gereklerinin sağlandığının gerçekleşmesi için sınamaların hazırlanıp yürütülmesi şeklinde özetlenebilir.

# Tanım ve Kavramlar

## Doğrulama (Verification):

**Tanım:** Yazılımın belirtilen gereksinimlere uygun olarak doğru bir şekilde geliştirilip geliştirilmediğini kontrol etme sürecidir.

**Amaç:** Yazılımın teknik ve fonksiyonel gereksinimlere uygun olarak geliştirilip geliştirilmediğini doğrulamaktır.

**Örnek:** Bir fonksiyonun, belirtilen algoritma gereği doğru sonuçları üretip üretmediğini kontrol etmek.

# Tanım ve Kavramlar

## Geçerleme (Validation):

**Tanım:** Yazılımın belirli bir kullanım için uygun olup olmadığını belirleme sürecidir.

**Amaç:** Yazılımın son kullanıcının gerçek dünya ihtiyaçlarına uygun olup olmadığını doğrulamaktır.

**Örnek:** Bir e-ticaret sitesinin, kullanıcının alışveriş deneyimini kolaylaştırıp kolaylaştırmadığını kontrol etmek.

# Tanım ve Kavramlar

## Sinama (Testing):

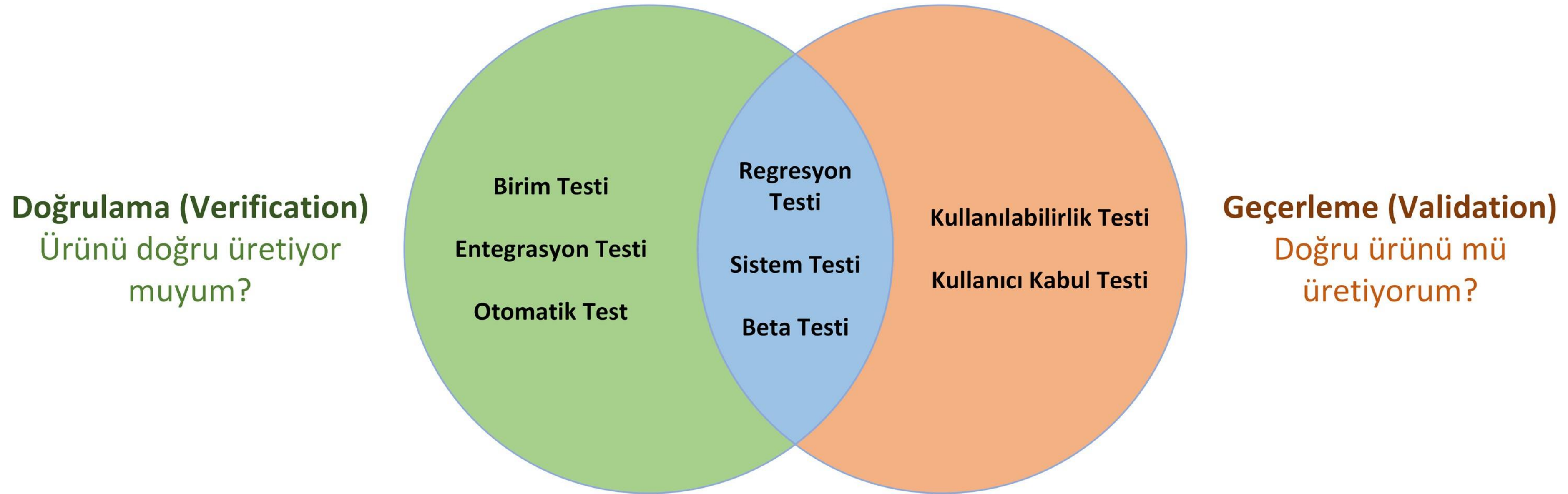
**Tanım:** Yazılımın belirlenen gereksinimleri ve işlevleri doğru bir şekilde karşılayıp karşılamadığını kontrol etme sürecidir.

**Amaç:** Yazılımın hatalarını veya kusurlarını bulmak ve yazılımın beklenen davranışı gösterip göstermediğini doğrulamaktır.

**Örnek:** Bir e-ticaret sitesinin ödeme işlemlerini güvenli ve hatasız bir şekilde gerçekleştirip gerçekleştirmediğini kontrol etmek.



# Tanım ve Kavramlar



# Doğrulama ve Geçerleme Arasındaki Fark

## DOĞRULAMA

**Soru:** "Doğru şeyi mi yapıyoruz?"

**Odak:** Teknik ve fonksiyonel gereksinimlere odaklanır. Yazılımın belirtilen standartlara ve gereksinimlere uygun olarak geliştirilip geliştirilmediğini kontrol eder.

**Araçlar:** Statik analiz, kod incelemeleri, birim testleri

## GEÇERLEME

**Soru:** "Doğru şeyi doğru mu yapıyoruz?"

**Odak:** Kullanıcının gerçek dünya ihtiyaçlarına odaklanır. Yazılımın belirli bir kullanım için uygun olup olmadığını belirler.

**Araçlar:** Kabul testleri, kullanılabilirlik testleri, performans testleri.

### Örnek:

Doğrulama: Bir hesap makinesi uygulamasının toplama işlevinin matematiksel olarak doğru sonuçları verip vermediğini kontrol etmek.

Geçerleme: Bu hesap makinesi uygulamasının kullanıcı için kolay kullanılıp kullanılamayacağını, kullanıcının ihtiyaçlarını karşılayıp karşılamadığını kontrol etmek.

# Doğrulama, Sınama ve Geçerleme

## Doğrulama

- Gereksinimlere uygunluk kontrolü yapar.
- Statik analiz yöntemleri kullanılır (kod incelemesi, tasarım belgelerinin gözden geçirilmesi vb.).
- "Yazılımın doğru şekilde geliştirilip geliştirilmediği" üzerine odaklanır.

## Sınama

- Yazılımın belirlenen işlevleri doğru bir şekilde karşılayıp karşılamadığını kontrol eder.
- Dinamik analiz yöntemleri kullanılır (yazılımın çalıştırılması).
- Hataların veya kusurların tespiti üzerine odaklanır.

## Geçerleme

- Yazılımın son kullanıcının gerçek dünya ihtiyaçlarına uygun olup olmadığını kontrol eder.
- Genellikle gerçek dünya senaryolarında test edilir.
- "Yazılımın doğru şeyi yapıp yapmadığı" üzerine odaklanır.

### Ortak Yönler:

- Her üç süreç de yazılımın kalitesini ve doğruluğunu sağlamak için gerçekleştirilir.
- Yazılımın belirlenen gereksinimleri ve beklentileri karşılayıp karşılamadığını kontrol etmeye yöneliktir.
- Hedef, müşteri memnuniyetini ve yazılımın başarılı bir şekilde çalışmasını sağlamaktır.

# Tanım ve Kavramlar

- Doğrulama ve geçerleme süreçleri, geliştirilen yazılımın spesifikasyonunu karşılayıp karşılamadığını ve yazılımı satın alan kişinin beklediği fonksiyonelliğe sahip olup olmadığını kontrol etmekle ilgilidir.
- Bu kontrol süreçleri gereksinimler tanımlandığı anda başlar ve yazılım geliştirme sürecinin tüm aşamaları boyunca devam eder.

# Doğrulama ve Geçerleme

- Doğrulama ve geçerleme süreçlerinin amacı yazılım sisteminin **kullanıma uygun** olduğu konusunda güven oluşturmaktır.
- Bunun anlamı, sistemin planlanan kullanım için yeteri kadar iyi olmak zorunda olmasıdır.
- Gereken güven seviyesi sistemin amacına, sistem kullanıcılarının beklentilerine ve sistemin pazarlanacağı çevreye bağlıdır.



# Doğrulama ve Geçerleme

- **Yazılımın amacı**

Yazılım ne kadar kritik ise güvenilir olması o kadar önemlidir.

- **Kullanıcı beklentileri**

Kullanıcılar bazen güvenilir olmayan yazılımlardan edindikleri kötü tecrübeler nedeniyle yazılım kalitesi konusunda daha düşük beklentilere sahip olabilirler. Piyasaya yeni sürülen bir sistem hatalarını düzeltme masraflarından daha çok fayda sağlıyorsa kullanıcı bu hataları görmezden gelebilir. Ancak sistem ne kadar tanınmış, yerleşmiş ise kullanıcı o kadar güvenilir olmasını bekler.

# Doğrulama ve Geçerleme

- **Pazarlama çevresi**

Bir yazılım şirketi bir sistemi pazarlamak istediğinde rakiplerinin benzer ürünlerini, müşterilerin sistem için ne kadar ödemeyi kabul edebileceğini ve sistemi teslim etme zamanlamasını göz önüne almak zorundadır. Rekabetin yüksek olduğu bir çevrede yazılım şirketi bu ürünü pazara süren ilk şirket olmak istediğinde yazılımı tamamıyla test etmeden ve hatalarını ayıklamadan piyasaya sürmeye karar verebilir. Yazılım ürünü ve uygulaması çok ucuzsa kullanıcılar düşük seviyede bir güvenilirliğe katlanabilirler.

# Neden Önemlidir?

## Maliyet Faktörü:

- Yazılım hatalarının erken aşamalarda tespit edilmesi, düzeltilmesi için daha az maliyet gerektirir. Geliştirme sürecinin ilerleyen aşamalarında bu hataların tespiti ve düzeltilmesi maliyeti artırır.
- Özellikle büyük ölçekli projelerde, hataların geç fark edilmesi projenin bütçesini ve zamanlamasını olumsuz etkileyebilir.



# Neden Önemlidir?

## Kullanıcı Memnuniyeti:

- Doğru ve geçerli bir yazılım, kullanıcıların ihtiyaçlarını doğru bir şekilde karşılar. Bu, kullanıcı memnuniyetini ve marka sadakatini artırır.
- Kullanıcı ihtiyaçlarına uygun olmayan yazılımlar, kullanıcı kaybına ve olumsuz geri dönüşlere neden olabilir.

# Neden Önemlidir?

## Güvenlik:

- Hatalı veya eksik doğrulama ve geçerleme, yazılımda güvenlik açıklarının oluşmasına neden olabilir. Bu açıklar, saldırganların yazılıma erişim sağlamasına veya veri ihlallerine yol açabilir.
- Güvenlik ihlalleri, kurumların itibarını, mali durumunu ve kullanıcı güvenliğini olumsuz etkileyebilir.

# Doğrulama ve Geçerleme Süreçleri

## 1. Gereksinim Doğrulama:

- Gereksinimlerin eksiksiz, tutarlı ve gerçekleştirilebilir olup olmadığını kontrol eder.
- Örnek: Bir e-ticaret sitesinde kullanıcının birden fazla adres ekleyebilmesi gereksinimi, doğru ve eksiksiz bir şekilde tanımlanmış mı?

## 2. Tasarım Doğrulama:

- Tasarımın gereksinimlere uygun olup olmadığını kontrol eder.
- Örnek: E-ticaret sitesinin tasarımı, kullanıcının birden fazla adres eklemesini destekliyor mu?

# Doğrulama ve Geçerleme Süreçleri

## 3. Kod Doğrulama:

- Kodun tasarıma uygun olup olmadığını kontrol eder.
- Örnek: Adres ekleme özelliği için yazılan kod, tasarlanan arayüz ve işlevselliği doğru bir şekilde yansıtıyor mu?

## 4. Ürün Geçerleme:

- Ürünün son kullanıcının ihtiyaçlarını karşılayıp karşılamadığını kontrol eder.
- Örnek: E-ticaret sitesi kullanıcıları, birden fazla adres ekleyebiliyor mu ve bu işlevsellik kullanıcı ihtiyaçlarına uygun mu?

# Doğrulama ve Geçerleme Yöntemleri

## Statik Yöntemler:

Statik yöntemler, yazılımın çalıştırılmadan, yani kodun çalışma zamanı dışında incelendiği yöntemlerdir. Bu yöntemler, yazılımın kalitesini ve doğruluğunu değerlendirmek için kullanılır.

**1. Kod İncelemeleri**

**2. Statik Analiz**

# Doğrulama ve Geçerleme Yöntemleri

## 1. Kod İncelemeleri:

- **Tanım:** Kod incelemeleri, yazılım geliştirme uzmanları tarafından yazılan kodun detaylı bir şekilde gözden geçirilmesi sürecidir.
- **Amaç:** Bu süreçte, kodun kalitesi, uyumluluğu, performansı ve güvenliği değerlendirilir. Ayrıca, kodun belirlenen standartlara ve en iyi uygulamalara uygun olup olmadığı kontrol edilir.

# Doğrulama ve Geçerleme Yöntemleri

- **Yöntem:** İnceleme süreci, genellikle bir grup uzmanın bir araya gelerek kodu satır satır incelediği bir toplantı şeklinde gerçekleştirilir. Bu toplantıda, kodun potansiyel sorunları, hataları ve iyileştirme alanları belirlenir.

# Doğrulama ve Geçerleme Yöntemleri

## 2. Statik Analiz:

- **Tanım:** Statik analiz, otomatik araçlar kullanılarak kodun yapısal hatalarının, güvenlik açıklarının ve performans sorunlarının tespit edilmesi sürecidir.
- **Amaç:** Bu süreç, kodun daha derinlemesine analiz edilerek potansiyel hataların ve risklerin erken bir aşamada tespit edilmesini amaçlar.



# Doğrulama ve Geçerleme Yöntemleri

- **Yöntem:** Otomatik analiz araçları, kodun belirli kurallara ve standartlara uygun olup olmadığını kontrol eder. Bu araçlar, kodda bulunan hataları, güvenlik açıklarını ve diğer potansiyel sorunları raporlar.

# Doğrulama ve Geçerleme Yöntemleri

## **Dinamik Yöntemler:**

Dinamik yöntemler, yazılımın çalıştırılarak, yani kodun çalışma zamanında incelendiği yöntemlerdir. Bu yöntemler, yazılımın gerçek dünyada nasıl performans gösterdiğini değerlendirmek için kullanılır.

**1. Testler**

**2. Birim Testi**

**3. Entegrasyon Testi**

**4. Sistem Testi**

# Doğrulama ve Geçerleme Yöntemleri

## 1. Testler

**Tanım:** Yazılımın belirli bir işlevselliği gerçekleştirip gerçekleştirmediğini kontrol etmek için yapılan denemeler.

**Amaç:** Yazılımın beklenen davranışı gösterip göstermediğini doğrulamak.

# Doğrulama ve Geçerleme Yöntemleri

## Özellikler:

- Testler, yazılımın farklı bileşenlerini, işlevlerini ve özelliklerini değerlendirir.
- Testler, yazılımın belirli koşullar altında nasıl tepki verdiğini belirlemek için kullanılır.

# Doğrulama ve Geçerleme Yöntemleri

## 2. Birim Testi

**Tanım:** Tek bir bileşenin veya fonksiyonun izole bir şekilde test edilmesi.

**Amaç:** Bireysel bileşenlerin veya fonksiyonların doğru çalışıp çalışmadığını kontrol etmek.

# Doğrulama ve Geçerleme Yöntemleri

## Yöntem:

- Test Senaryosu Oluşturma: Test edilecek fonksiyon veya bileşen için olası girdi ve beklenen çıktıları belirleyerek senaryolar oluşturulur.
- Testin Uygulanması: Oluşturulan senaryolar çalıştırılır ve sonuçlar beklenen değerlerle karşılaştırılır.

# Doğrulama ve Geçerleme Yöntemleri

## Faydaları:

- Hataların erken tespiti sayesinde yazılımın genel kalitesi artar.
- Kodun yeniden yapılandırılması (refactoring) sırasında güvence sağlar.

# Doğrulama ve Geçerleme Yöntemleri

## 3. Entegrasyon Testi

**Tanım:** Farklı bileşenlerin veya modüllerin birlikte çalışıp çalışmadığının test edilmesi.

**Amaç:** Bileşenler arası etkileşimlerin ve bağlantıların doğru çalışıp çalışmadığını doğrulamak.



# Doğrulama ve Geçerleme Yöntemleri

## Yöntem:

- Bileşen Seçimi: Test edilecek bileşenler ve modüller belirlenir.
- Senaryo Oluşturma: Bileşenler arası etkileşimleri test edecek senaryolar oluşturulur.
- Testin Uygulanması: Oluşturulan senaryolar çalıştırılır ve sonuçlar değerlendirilir.

# Doğrulama ve Geçerleme Yöntemleri

## Faydaları:

- Bileşenler arası potansiyel uyumsuzluklar erken bir aşamada tespit edilir.
- Veri akışının doğruluğu ve etkileşimlerin istenildiği gibi çalışıp çalışmadığı kontrol edilir.

# Doğrulama ve Geçerleme Yöntemleri

## 4. Sistem Testi

**Tanım:** Yazılımın tamamının, gerçek dünya senaryoları altında test edilmesi.

**Amaç:** Yazılımın tüm bileşenlerinin birlikte doğru çalışıp çalışmadığını doğrulamak.

# Doğrulama ve Geçerleme Yöntemleri

## Yöntem:

- Senaryo Oluşturma: Yazılımın tüm işlevselliğini kapsayan test senaryoları oluşturulur.
- Testin Uygulanması: Oluşturulan senaryolar çalıştırılır ve sonuçlar değerlendirilir.

# Doğrulama ve Geçerleme Yöntemleri

## Faydaları:

- Yazılımın genel performansı ve istikrarı değerlendirilir.
- Kullanıcı gereksinimlerinin ve beklentilerinin karşılanıp karşılanmadığı kontrol edilir.

# Test Türleri

Yazılım test türleri, yazılımın farklı yönlerini ve özelliklerini değerlendirmek için kullanılır.

- Fonksiyonel test
- Performans testi
- Güvenlik testi
- Kullanılabilirlik testi
- Uyumluluk testi

## 1. Fonksiyonel Test:

- **Tanım:** Yazılımın belirli işlevleri ve özellikleri yerine getirip getirmediğini kontrol etmek amacıyla yapılan test türüdür.
- **Amaç:** Yazılımın belirtilen gereksinimleri ve işlevleri doğru bir şekilde karşılayıp karşılamadığını doğrulamaktır.

# Test Türleri

- **Özellikler:**

- Kullanıcı senaryoları, iş akışları ve kullanım durumları temel alınarak gerçekleştirilir.
- Genellikle yazılımın belirli bir özelliği veya işlevi üzerinde odaklanır.
- Hataların ve eksikliklerin tespiti için kullanılır.



## 2. Performans Testi:

- **Tanım:** Yazılımın belirli yük ve stres koşulları altında nasıl performans gösterdiğini değerlendirmek amacıyla yapılan test türüdür.
- **Amaç:** Yazılımın hızını, tepki süresini, kararlılığını ve ölçeklenebilirliğini ölçmektir.

# Test Türleri

- **Özellikler:**

- **Yük Testi:** Yazılımın belirli bir yük altında nasıl performans gösterdiğini ölçer.
- **Stres Testi:** Yazılımın maksimum yük altında nasıl performans gösterdiğini ve ne zaman çöktüğünü belirler.
- **Dayanıklılık Testi:** Yazılımın uzun süreli yük altında nasıl performans gösterdiğini değerlendirir.

## 3. Güvenlik Testi:

- **Tanım:** Yazılımın güvenlik açıklarını, zayıflıklarını ve risklerini tespit etmek amacıyla yapılan test türüdür.
- **Amaç:** Yazılımın yetkisiz erişimlere, saldırılara ve tehditlere karşı korunmasını sağlamaktır.

# Test Türleri

- **Özellikler:**

- **Sızma Testi:** Yazılıma dışarıdan erişim sağlamaya çalışarak güvenlik açıklarını tespit eder.
- **Yetki Testi:** Kullanıcıların yetkilerini aşarak işlem yapmaya çalışmalarını engellemeyi amaçlar.
- **Zararlı Yazılım Taraması:** Yazılımda zararlı yazılımların olup olmadığını kontrol eder.

## 4. Kullanılabilirlik Testi:

- **Tanım:** Kullanıcıların bir yazılımı veya uygulamayı ne kadar kolay ve etkili bir şekilde kullanabildiğini değerlendirmek amacıyla yapılan test türüdür.
- **Amaç:** Kullanıcı deneyimini optimize etmek ve kullanıcının ihtiyaçlarına en uygun hale getirmektir.

# Test Türleri

- **Özellikler:**

- Gerçek kullanıcılarla gerçekleştirilir.
- Kullanıcıların karşılaştığı sorunları ve zorlukları tespit etmek için kullanılır.

## 5. Uyumluluk Testi:

- **Tanım:** Yazılımın farklı cihazlarda, işletim sistemlerinde ve tarayıcılarda nasıl performans gösterdiğini değerlendirmek amacıyla yapılan test türüdür.
- **Amaç:** Yazılımın farklı platformlarda ve ortamlarda sorunsuz çalışıp çalışmadığını doğrulamaktır.

# Test Türleri

- **Özellikler:**

- Farklı cihazlarda, ekran çözünürlüklerinde ve tarayıcılarda test edilir.
- Yazılımın geniş bir kullanıcı kitlesine uygun olup olmadığını belirlemek için kullanılır.



## Sonuç

- Doğrulama ve geçerleme, yazılım geliştirme sürecinin ayrılmaz bir parçasıdır.
- Bu süreçler, yazılımın kalitesini, güvenliğini ve kullanıcı memnuniyetini artırmada kritik bir rol oynar.

## ÖZET

- Doğrulama, yazılımın belirtilen gereksinimlere uygun olarak geliştirilip geliştirilmediğini kontrol eder.
- Geçerleme, yazılımın belirli bir kullanım için uygun olup olmadığını belirler.
- Doğrulama, teknik ve fonksiyonel gereksinimlere odaklanırken, geçerleme kullanıcının ihtiyaçlarına odaklanır.

## ÖZET

- Yazılım hatalarının erken tespiti maliyetleri düşürür.
- Doğru ve geçerli yazılım kullanıcı memnuniyetini artırır.
- Eksik doğrulama ve geçerleme güvenlik risklerini artırabilir.
- Gereksinim, tasarım ve kod doğrulamaları yazılımın farklı aşamalarında gerçekleştirilir.

## ÖZET

- Ürün geçerleme, yazılımın son kullanıcının ihtiyaçlarını karşılayıp karşılamadığını kontrol eder.
- Statik yöntemler, kodun çalıştırılmadan incelendiği yöntemlerdir.
- Dinamik yöntemler, kodun çalıştırılarak incelendiği yöntemlerdir.

## ÖZET

- Fonksiyonel testler, yazılımın belirli işlevleri yerine getirip getirmediğini kontrol eder.
- Performans testleri, yazılımın yük altında nasıl performans gösterdiğini değerlendirir.
- Güvenlik testleri, yazılımdaki potansiyel güvenlik açıklarını tespit eder.

## ÖZET

- Kullanılabilirlik testleri, kullanıcı deneyimini değerlendirir.
- Uyumluluk testleri, yazılımın farklı cihazlarda ve platformlarda nasıl çalıştığını kontrol eder.

# ÖRNEK SORULAR

- Doğrulama ve Geçerleme arasındaki temel fark nedir?
- Doğrulama, Sınama ve Geçerleme arasındaki ortak yönler nelerdir?
- Hangi test türü, yazılımın belirli işlevleri ve özellikleri yerine getirip getirmediğini kontrol etmek amacıyla kullanılır?
- Yazılımın farklı cihazlarda, işletim sistemlerinde ve tarayıcılarda nasıl performans gösterdiğini değerlendirmek için hangi test türü kullanılır?
- Hangi süreç, yazılımın belirtilen gereksinimlere uygun olarak doğru bir şekilde geliştirilip geliştirilmediğini kontrol eder?
- Hangi test türü, yazılımın belirli yük ve stres koşulları altında nasıl performans gösterdiğini değerlendirmek amacıyla kullanılır?

# ÖRNEK SORULAR

- Statik yöntemler nedir ve bu yöntemler neleri içerir?
- Hangi süreç, yazılımın son kullanıcının ihtiyaçlarını karşılayıp karşılamadığını kontrol eder?
- Yazılımın belirli bir bileşeninin veya fonksiyonunun izole bir şekilde test edilmesi hangi test türü ile gerçekleştirilir?
- Hangi test türü, yazılımdaki potansiyel güvenlik açıklarını tespit etmek amacıyla kullanılır?
- Hangi süreç, yazılımın belirtilen standartlara ve en iyi uygulamalara uygun olup olmadığını kontrol eder?