

DT/NT : DT/NT - CUMA
LESSON : STLÇ
SUBJECT: TEST ÇEŞİTLERİ

BATCH:189
29.09.2023



TECHPRO
EDUCATION



techproeducation.com



+1 (585) 304 29 59





Önceki Derste Neler Öğrendik?

1. STLC Nedir?
2. STLC Neden Önemlidir?
3. STLC Aşamaları:
 - a. Gereksinim Analizi
 - b. Test Planlama
 - c. Test Senaryosu Geliştirme
 - d. Ortam Kurulumu
 - e. Test Yürütme
 - f. Test Döngüsü Kapatma





YAZILIM TESTİ ve TEST ÇEŞİTLERİ

STLC



Yazılım Testine Giriş

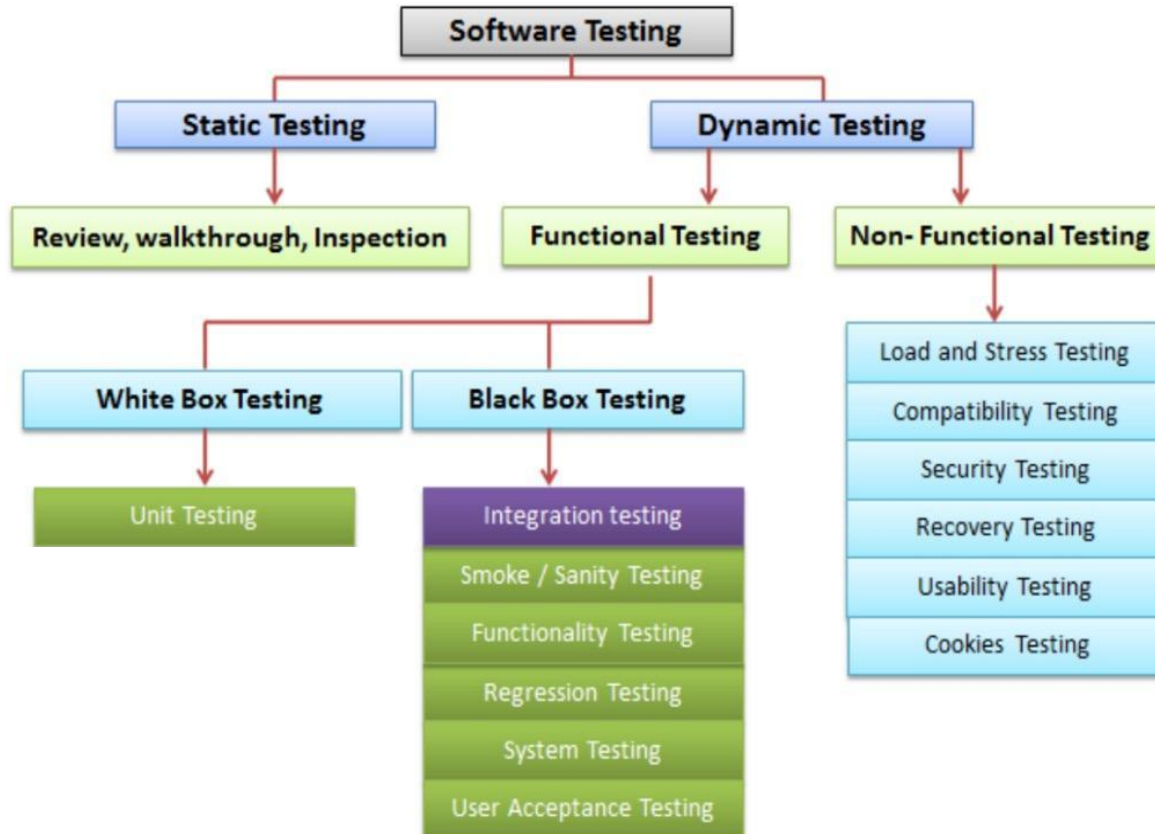
Yazılım testi;

- Kalite ve güvenilirlik sağlar.
- Hataların tespit edilip düzeltilmesini sağlar.
- Erken aşamalarda sorunları tespit eder.
- Maliyet ve zaman tasarrufu sağlar.
- Gereksinimleri karşılar.
- Müşteri memnuniyetini artırır.





Test Çeşitleri





Statik Test

Yazılım Testi ⇒ **Statik Test**

- Kodun veya diğer dokümanların **manuel olarak** gözden geçirilmesidir.
- Dinamik testlere **geçilmeden önce** yapılmalıdır.
- Erken aşamalarda hataları tespit etmeye yardımcı olur.

Statik Test Uygulaması:

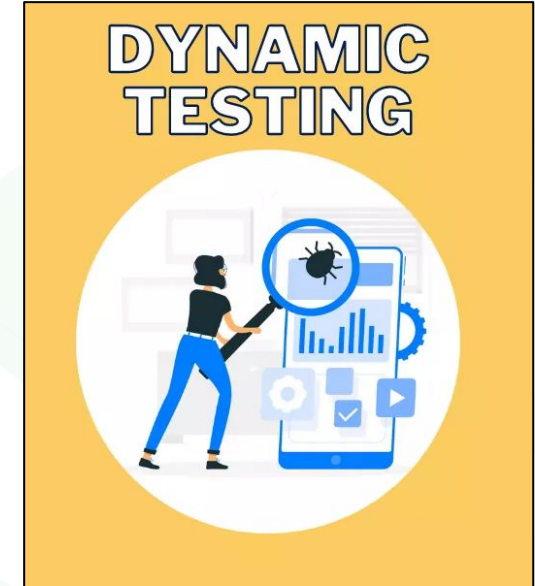




Dinamik Test

Yazılım Testi ⇒ **Dinamik Test**

- Yazılımın, **çalışma zamanında** test edilmesidir.
- **Gerçek dünya senaryolarıyla** yazılımın performansı değerlendirilir.
- İki türü vardır:
 - Fonksiyonel testler
 - Fonksiyonel olmayan testler





Fonksiyonel Testler

Yazılım Testi ⇒ Dinamik Test ⇒ **Fonksiyonel Testler**

- **Fonksiyonların, gereksinimlerin ve iş süreçlerinin** yerine getirilip getirilmediğini değerlendiren testlerdir.
- İki temel türü vardır:
 - Beyaz kutu testi
 - Siyah kutu testi

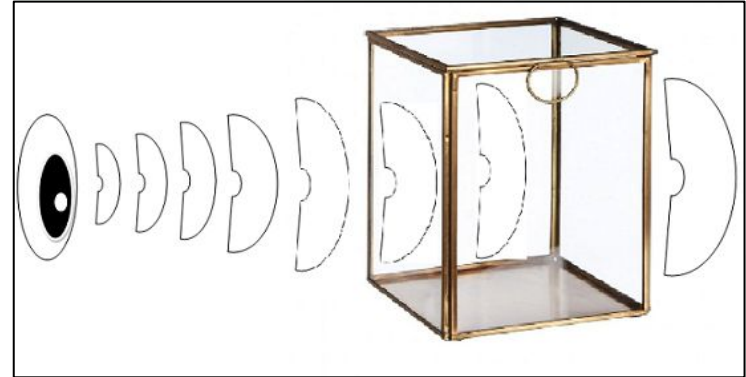
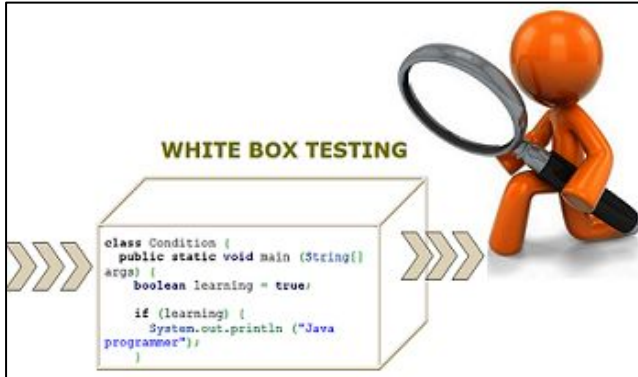




Beyaz Kutu Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ **Beyaz Kutu Testi**

- Yazılımın **iç yapısını** inceleyen **fonksiyonel** bir test türüdür.
- **Kodları test etmeyi** ve yazılımın doğru çalışmasını amaçlar.
- **Geliştiriciler tarafından yapılır** ve hataların erken aşamalarda yakalanabilmesine olanak sağlar.
- **Birim testi** en yaygın beyaz kutu testi türüdür.





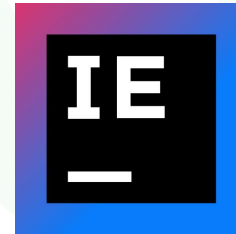
Birim Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Beyaz Kutu Testi ⇒ **Birim Testi**

- Yazılım modüllerinin **bağımsız olarak** test edilmesidir.
- Entegrasyon testlerinin yürütülmesinden **önce** gerçekleştirilir.
- Developer ekibi tarafından gerçekleştirilir.
- **JUnit**, NUnit gibi araçlar kullanılarak testler yapılabilir.

Birim Testi Uygulaması:

https://github.com/gurkaybirinci/B171_STLC.git

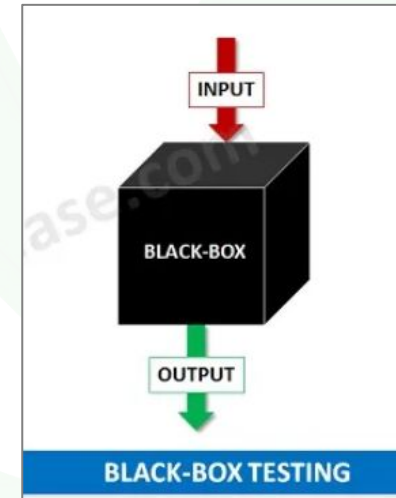




Kara Kutu Testi

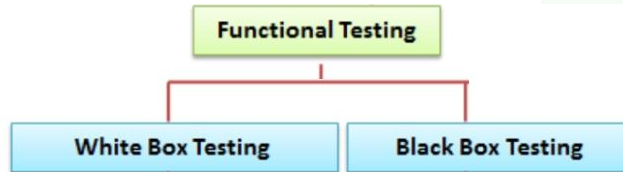
Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ **Kara Kutu Testi**

- Yazılımın **koduna bakmaz**, sadece girdileri ve çıktıları test eder.
- Testler gereksinimlere göre yapılır.
- **Tester'lar** tarafından gerçekleştirilir.
- Yazılımın farklı yönlerini test etmek için kullanılan **farklı türleri vardır**:
 - Entegrasyon testi
 - Duman testi
 - Sağlamlık testi
 - İşlevsellik testi
 - Regresyon testi
 - Sistem testi
 - Kullanıcı kabul testi

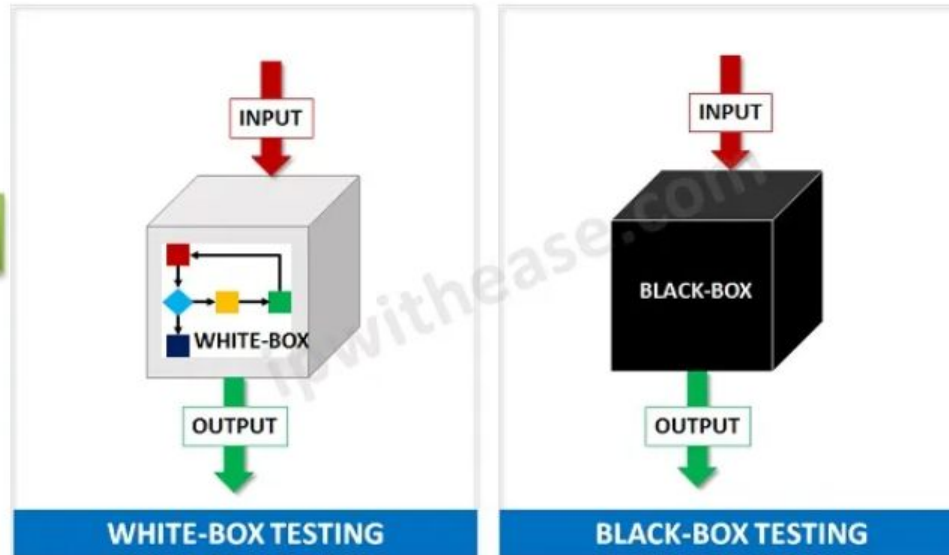




Beyaz Kutu ve Kara Kutu Testi



WHIE-BOX TESTING vs BLACK-BOX TESTING



Unit Testing

Integration testing

Smoke / Sanity Testing

Functionality Testing

Regression Testing

System Testing







User Acceptance Testing



Entegrasyon Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Entegrasyon Testi**

- Yazılımdaki **birimlerin etkileşimleri** test edilir.
- **Birim testi** yapıldıktan sonra yapılır.
- **Developer'lar ve tester'lar** tarafından gerçekleştirilir.

	Unit Testing	Integration Testing
		
		

ProgrammerHumor.io





Entegrasyon Testi

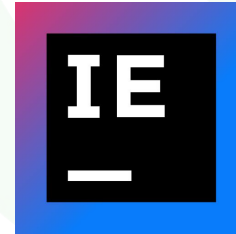
Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Entegrasyon Testi**

Entegrasyon Hatalarına Örnekler:



Entegrasyon Testi Uygulaması:

https://github.com/gurkaybirinci/B171_STLC.git





Smoke Test (Duman Testi)

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Duman Testi**

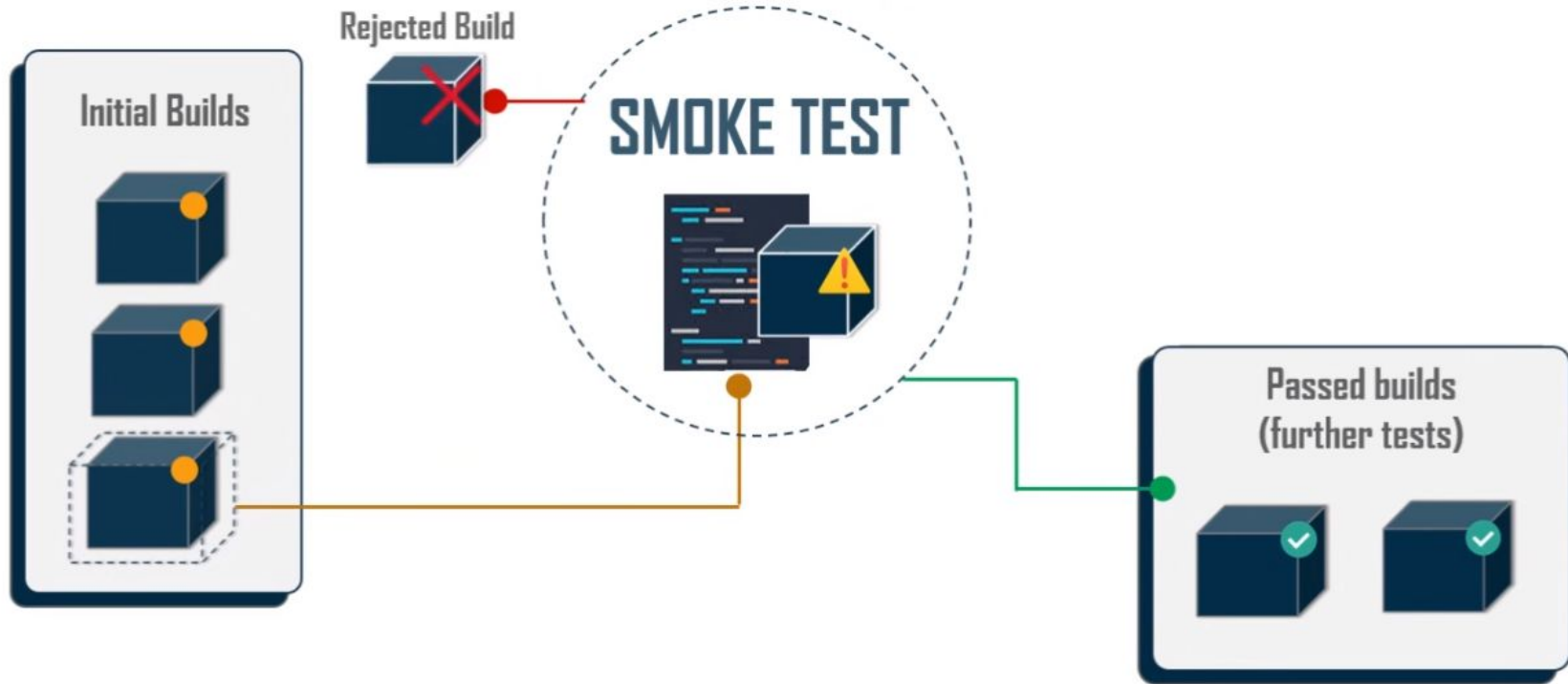
- Yazılımın **temel işlevlerinin** test edildiği hızlı ve basit bir test türüdür.
- Yazılımın **iskeleti tamamlandıktan sonra** gerçekleştirilir.
- **Kritik fonksiyonların** düzgün çalıştığından emin olmayı amaçlar.
- Yazılımın kararlı ve **daha fazla teste hazır olduğunu** garanti eder.
- Otomatize edilmelidir (**Jenkins**).





Smoke Test (Duman Testi)

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Duman Testi**





Sanity Test (Sağlamlık Testi)

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Sağlamlık Testi**

- Yazılımın test için **hazır olup olmadığını** belirlemek için yapılır.
- Yazılımda **değişiklikler** yapıldıktan sonra gerçekleştirilir.
- Yapılan değişikliklerin **yeni sorunlara yol açıp açmadığını** test eder.





Regresyon Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Regresyon Testi**

- Yazılımda yapılan versiyon değişikliklerinin mevcut işlevselliği etkileyip etkilemediğini kontrol etmek için yapılan test türüdür.
- Kapsamlıdır ve zaman alıcıdır.
- Yeni senaryolar tasarlanmaz, mevcut olanlar kullanılır.
- Eski ve yeni test sonuçları karşılaştırılarak yapılır.





Sistem Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Sistem Testi**

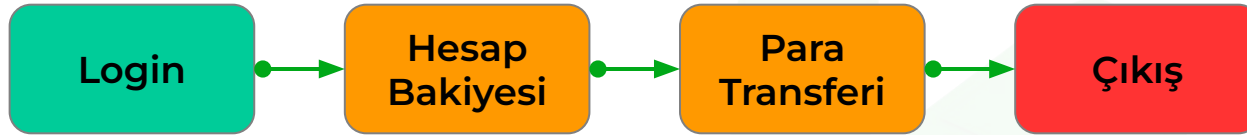
- Yazılım sisteminin genel işlevselliğini sağlamak için gerçekleştirilir.
- Sistemin tamamının test edilmesini içerir.
- UI ve Backend testleri yapılır.
- Uçtan uca (E2E) senaryolarla testler koşulur.
- Fonksiyonel olmayan testler de bu sürece dahil edilebilir.



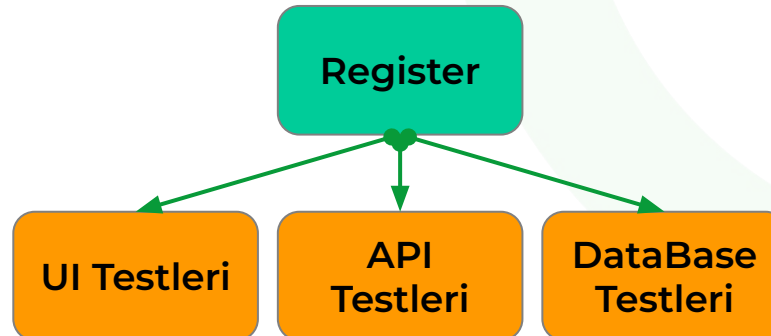
Sistem Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Sistem Testi**

- İki şekilde yapılabilir.
 - Farklı fonksiyonların birleştiği uçtan uca senaryolar ile..



- Aynı modülün Backend ve UI tarafında ele alındığı senaryolar ile..

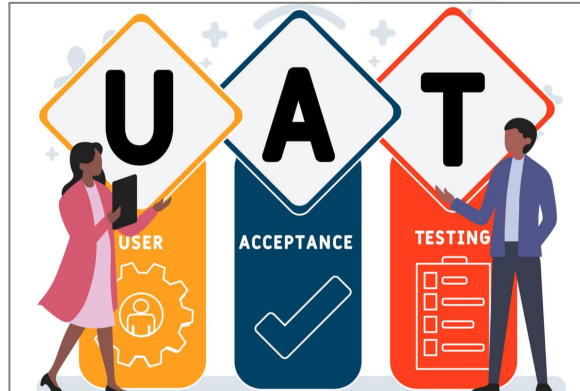




Kullanıcı Kabul Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Testler ⇒ Kara Kutu Testi ⇒ **Kullanıcı Kabul Testi**

- Müşterinin ürünü yazılımsal olarak ilk gördüğü aşamadır.
- Son kullanıcıları temsil eden bir grup tarafından gerçekleştirilir.
- Amaç hata bulmak değil, gereksinimlerin karşılanmasıdır.
- Yazılım yayınlanmadan gerçekleştirilen son testtir.
- Beta testi olarak ta adlandırılır.





Fonksiyonel Olmayan Testler

Yazılım Testi ⇒ Dinamik Test ⇒ **Fonksiyonel Olmayan Testler**

- Performans, kullanılabilirlik ve güvenlik gibi faktörleri değerlendiren testlerdir.
- Bazı türleri:
 - Yük ve stres testi
 - Uyumluluk testi
 - Güvenlik testi
 - Kullanılabilirlik testi





Yük ve Stres Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Olmayan Testler ⇒ **Yük ve Stres Testi**

- Yoğun yük ve stres koşullarında performansı değerlendiren tekniktir.
- Yüksek trafik ve yoğun kullanım senaryolarını taklit eder.
- Yanıt süreleri, çökmeler gibi performans sorunlarını belirler.
- JMeter





Uyumluluk Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Olmayan Testler ⇒ **Uyumluluk Testi**

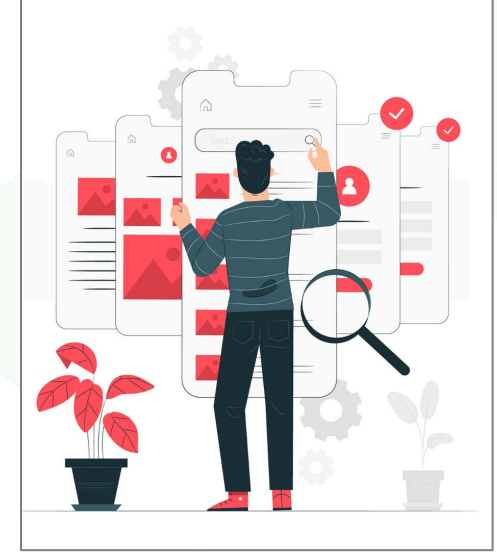
- Farklı donanım, yazılım ve ağ yapılandırmalarıyla uyumluluğu kontrol eden tekniktir.
- Farklı işletim sistemleri, tarayıcılar, veritabanları ve donanım yapılandırmalarıyla test edilir.
- Uyumluluk sorunlarını tespit etmeyi amaçlar.
- Yazılımın farklı ortamlarda doğru şekilde çalışabilmesini sağlar.



Kullanılabilirlik Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Olmayan Testler ⇒ **Kullanılabilirlik Testi**

- Yazılımın kullanıcılar için kullanılabilir ve anlaşılır olmasını sağlamaya odaklanan test.
- Gerçek kullanıcılarla yazılımın deneyimlenmesini içerir.





Güvenlik Testi

Yazılım Testi ⇒ Dinamik Test ⇒ Fonksiyonel Olmayan Testler ⇒ **Güvenlik Testi**

- Uygulamanın güvenliğini değerlendiren bir test türüdür.
- Veri güvenliğini, yetkilendirme mekanizmasını ve saldırılara karşı dayanıklılığı kontrol eder.
- Güvenlik açıklarını tespit etmek ve düzeltmek için yapılır.

