

# STLC

**software testing life cycle**  
(yazılım test yaşam döngüsü)

**Ayşe Sezen**

**Full-Stack Automation Engineer**

# Software Testing Life Cycle (STLC)

## Yazılım Test Yaşam Döngüsü

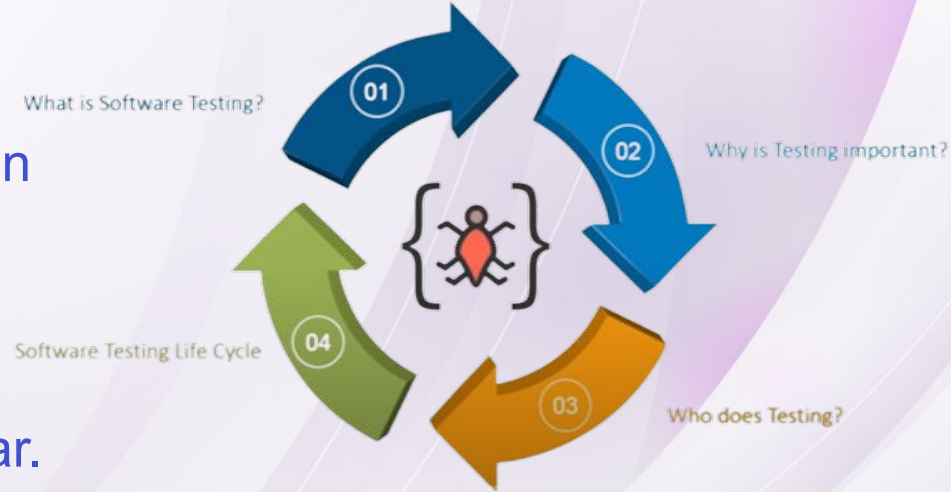
**(STLC)**, yazılımı test etmek ve kalite standartlarının karşılandığından emin olmak için kullanılan bir süreçtir. Testler sistematik olarak birkaç aşamada gerçekleştirilir.

Ürün geliştirme sırasında, (ürün piyasaya sürülmeye uygun görülene kadar) STLC'nin aşamaları birden çok kez gerçekleştirilebilir.



# Software Testing Niçin Önemlidir?

- Yazılımın güvenilirliğini kontrol etmek için yazılım testi gereklidir.
- Yazılım testi, sistemin arızaya neden olabilecek herhangi bir hata içermemesini sağlar.
- Yazılım testi, ürünün müşterinin gereksinimine uygun olmasını sağlar.
- Sonunda, yazılım, hepsi farklı bakış açılara ve yaklaşımlara sahip bir Developer ekibi tarafından geliştirilir. En zeki insan bile hata yapma eğilimindedir. Sıfır hata ile yazılım oluşturmak mümkün değildir dolayısıyla Test Yaşam Döngüsü yazılım geliştirme döngüsüne muhakkak dahil edilir.



# Software Testing Niçin Önemlidir?

- Yazılım testi olmadan yazılım istenen kaliteye ulaşamaz.
- Tester'lar application'ın kodlarına dahil olmasalar bile, kodun kalitesini artırmak için Developer'larla yakın çalışmalıdırlar.
- En iyi sonuçlar için yazılım testi ve kodlamanın iç içe gitmesi önemlidir.
- Test, yazılımın kalitesini değerlendirmeye yardımcı olur.
- Tüm test faaliyetleri planlama gerektirir.
- Test, yazılımın tam olarak gereksinimlerde belirtildiği gibi davranmasını ve uygulanmasını sağlar.
- İhtiyaca(requirement) uygun olmayan her şey bir hatadır(defect,BUG).
- İyi test edilmiş yazılım kaliteli ve daha iyi geri bildirim ve yorumlar anlamına gelir (iyi feedbackler).

# Software Quality Assurance (QA) Nedir, Wise Quarter

## Niçin Önemlidir?

Yazılım Kalite Güvencesi, tüm yazılım geliştirme sürecini içerir - sürecin izlenmesi ve iyileştirilmesi, üzerinde anlaşılan süreçlerin, standartların ve prosedürlerin takip edilmesini, sorunların bulunmasını ve ele alınmasını sağlar.

Kalite kontrol (Quality Assurance – QA), bir sürecin kalite etkinliğini azaltacak durumlara karşı (hatalara karşı) önlem alarak kaliteye hakim olma anlamına gelir.



# Software Testing Nedir, Niçin Önemlidir?



Yazılım testi, geliştirilen bilgisayar yazılımının doğruluğunu, bütünlüğünü ve kalitesini belirlemek için kullanılan bir süreçtir.

Ürünün son kullanıcılara sunulmadan önce düzeltilebilmesi için yazılımda hata bulmak amacıyla yapılan bir dizi etkinliği içerir.

Basit bir ifadeyle, yazılım testi, **gerçek sonuçların beklenen sonuçlarla** eşleşip eşleşmediğini kontrol etmek ve yazılım sisteminin hatasız olmasını sağlamak için yapılan bir etkinliktir.



# Software Testing Life Cycle (STLC)

## Aşamaları

Gereksinim  
Analizi

Requirement Analysis

STEP  
01



Phases of  
Software  
Life cycle

STEP  
02

Test Planning

Testin Planlanması

Test Senaryosu  
Geliştirme

Test case development

STEP  
03



Test Ortamının  
Kurulması

STEP  
04

Test Environment setup

Testin  
Gerçekleştirilmesi

Test Execution

STEP  
05



Testin Döngüsü  
Kapanışı

STEP  
06

Testing closure

# 1. Requirements Analysis (Gereksinimlerin Analizi)



Bu aşamada yapılması gereken en önemli iş **User Story**'de açıklanan **yazılım gereksinimlerinin** gözden geçirilmesidir.

- Susuz bir kişi olarak susuzluğumu gidermek için su istiyorum.
- Modaya ilgi duyan bir insan olarak içeceğimin güzel görünmesi için şemsiye konulmasını istiyorum.
- Sıcaktan bunaldığım için suyumla birlikte beni ferahlatacak 1 dilim limon istiyorum.
- Çok susadığım için büyük bir bardak su istiyorum.
- Hava sıcak olduğu için suyumun soğuk olmasını istiyorum.
- Cool görünmek için suyu içeceğim bir pipet istiyorum.

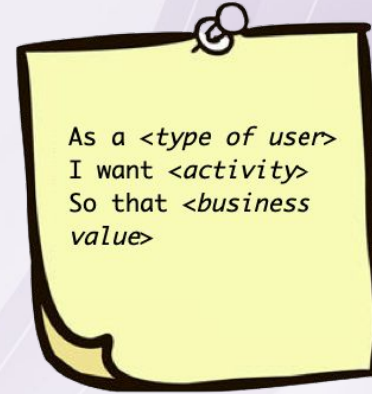


# User Story (Kullanıcı Hikayesi)

User story; son kullanıcı tarafından kullanılacak ve ona fayda sağlayacak bir özelliğin(feature), yine bu kullanıcı tarafından en basit ve anlaşılır şekilde açıklanmasıdır.

Bu açıklamayı gerçekleştirirken 3 soruya cevap vermesi gerekiyor.

1. Talep eden kullanıcı kim?
2. Talep edilen ihtiyaç nedir?
3. Neden bu talebe ihtiyaç var?



<Kullanıcı> olarak, <aktivite> gerçekleştirmek istiyorum.  
Böylece <fayda> sağlayabilirim.

# User Story (Kullanıcı Hikayesi)

## User story:

- İstenilen ürün veya özelliğin kısa ve anlaşılır tanımıdır.
- Ürün veya özelliği isteyen kişinin perspektifinden yazılır.
- Ürünün eski bir kullanıcısı veya muhtemel kullanıcısı olarak yazılır.
- Üç bölümden oluşur.



Bir **(kullanıcı)** olarak, **(kullanıcıların memnun olduğu bir telefon araması)** yapabilmek için **(puanlara göre arama yapmak)** istiyorum.

# User Story (Kullanıcı Hikayesi)

## User story:

Bir kullanıcıya değer sağlayacak işlevselliğin kısa açıklaması

## İçerik:

- Başlık
- Açıklama
- Acceptance Criteria  
(Kabul etme kriteri/kabul şartları)
- Definition of Done (Bitti tanımı)



**Fonksiyon:** Login Fonksiyonu

**On Kosul:**

- Kullanıcının login olabilmesi için,daha önceden basarili bir uye kaydi olmalidir.

**Gereksinimler:**

- Kullanıcının eopsta adresi girmesi için bir text box olmalidir.
- Kullanıcının sifresini girmesi için bir text box olmalidir.
- Kullanıcının bu bilgileri sunucuya gondermesi için bir buton olmalidir.

**Test senaryolari:**

Test Senaryolari	Test Durumlari
E-posta alanina fazla karakter girilmesi	Negatif
Sifre alanina fazla karakter girilmesi	Negatif
E-posta alanina e_posta formatinda bir deger girilmemesi	Negatif
E-posta ve sifre alanlarına dogru bir deger girilmesi	Pozitif
E-postasi dogru sifresi yanlis bir deger girilmesi	Negatif
Alanlara hic bir deger girilmeden formun gonderilmesi	Negatif
Gonderme butonuna iki kere tiklanmasi	Negatif
Daha once uye olmayan bir e-posta ile login denemesi	Negatif

# User Story (Kullanıcı Hikayesi)

User storyler 3 bölümden oluşur. Bunlar;

1. Kullanıcının hikaye tanımı (Value statement)
2. Kabul kriterleri (Acceptance criterias)
3. Bitti tanımı (Definition of done)

# Kullanıcı Hikayesi Örneği

USER STORY #1	
Hikâye Tanımı (Value Statement)	Ürün ekibi yetkilisi olarak, paket yenilemelerini arttırabilmek için; üyelerimize paketlerinin bitmesine 1 ay kala “yenileme yapmak ister misiniz” mail gönderimi yapılmasını, mail şablonu içerisinde X, Y, Z paket detayları ve ilgili müşteri temsilcilerinin ad, soyad, iletişim bilgilerinin yer almasını istiyorum.
Kabul kriterleri (Acceptance criterias)	<ul style="list-style-type: none"><li>• Paket bitimine 1 ay kalan kullanıcılara mail gönderimlerinin başarıyla yapılması</li><li>• Üye – müşteri temsilcisi eşleşmesinin doğru yapılmış olması</li><li>• Gönderimlerin sadece mail gönderimine izin verilen kullanıcılara yapılması</li><li>• Mail opt-out kurgularının sağlıklı bir şekilde çalışması</li></ul>
Biti Tanımı (Definition of Done)	<ul style="list-style-type: none"><li>• Kod kontrol edildi (Code review)</li><li>• Birim testleri yapıldı (Unit tests)</li><li>• Kabul kriterleri karşılandı (Acceptance criterias)</li><li>• Product Owner(PO) story kabulü (PO accepts user story)</li></ul>

# Characteristics of User Story



**Independent** : Bağımsız bir şekilde geliştirilebilmeli ve teslim edilebilmeli.

**Negotiable** : Ekip ve iş birimleri üzerinde tartışabilmeli.

**Valueable** : Kullanıcıya değer üretebilmeli.

**Estimable** : Efor tahmini yapılabilirmeli.

**Small** : Tek sprint içerisinde tamamlanabilecek parçalara ayrılmış olmalı.

**Testable** : Test edilebilir nitelikte olmalıdır.



# Characteristics of Acceptance Criteria Wise Quarter



## Specific – Spesifik

Bir görevin ve/ya hedefin, herkesin anlayabilmesi ve bunun herkes tarafından aynı şekilde anlaşılması için yeterince spesifik olması gerekir. Olabildiğince özel, kesin ve **başka hedeflerle çakışmayacak** şekilde tanımlanmalıdır.

## Measurable – Ölçümlenebilir

Hedefin ölçümlenebilir olması, **somut ve sayısal** metriklerle kanıtlamak anlamındadır. Yani bir hedef “Sayfanın performansını arttırmak” gibi göreceli değil; “Sayfanın performansını %10 arttırmak” gibi daha somut olacak şekilde tanımlamak daha doğru olacaktır.

## Achievable – Ulaşılabilir

Hedef ulaşılması zor ya da **imkansız olmamalıdır**. “Hedefi yüksek tutmak lazım” gibi bir yaklaşım doğru değildir.

# Characteristics of Acceptance Criteria Wise Quarter



## Relevant – Uygun

Belirlediğiniz hedefin size gerçekten değer verip vermediğine emin olunmalı. **Vizyon ve misyona** uygun hedefler belirleyerek yola devam edilmelidir.

## Time Bound (Time Boxed) – Zaman Sınırlı

Hedefleri, gerçekleşmesini **uygun** gördüğümüz **zamanı belirleyerek** tanımlamak gerekir. Çünkü zaman kısıtlamaları bir aciliyet duygusu yaratır. Bu da hedefe odaklanmayı sağlar.

## ! HATIRLATMA:

Hikayeleri son kullanıcıların yazması başta zor olacaktır. Hatta en çok karşılaşılan durum; kullanıcıdan gelen hikayeyi geliştirici ekibin değişmez görmesidir.

“Gelen hikaye kapsamında bunlar yoktu ...” gibi söylemlerle karşılaşılabılır.

Burada bir kullanıcı hikayesinin “Negotiable”(tartışılabilir) olduğu unutulmamalı, geliştirici ekip ve son kullanıcılar işbirliği ile o hikayeyi detaylandırmalı, varsa değişiklikler değerlendirilmelidir.

## 2.Test Planning (Test Planlaması)

Neyin test edilmesi gerektiğine dair genel bir fikir topladıktan sonra, testler için 'plan yapılır'.

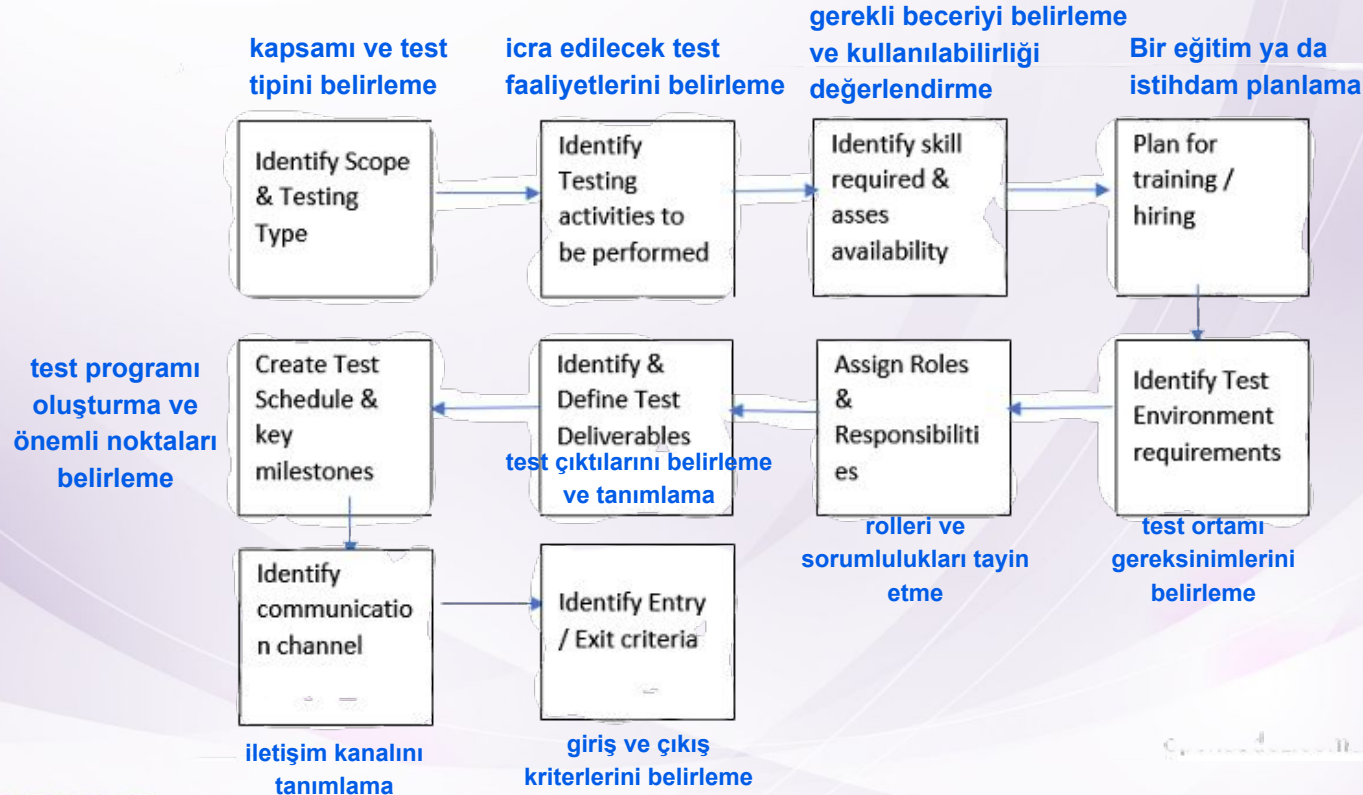
Test Planı belgesi, Ürün Açıklaması, Yazılım Gereksinimi Spesifikasyonu (Software Requirement Specification SRS) veya Kullanım Senaryosu Belgelerinden (Use Case Documents) türetilmiştir.

Amaçlanan test faaliyetlerinin kapsamını, yaklaşımını, kaynaklarını ve programını açıklayan bir belgedir.

**Test Planı belgesi** genellikle **Test Lead** veya **Test Manager** tarafından hazırlanır ve belgenin odak noktası neyin test edileceğini, nasıl test edileceğini, ne zaman test edileceğini ve hangi testi kimin yapacağını açıklamaktır.

# Test Planning Phases

## (Test Planlaması Aşamaları)



## 3. Test Case Development (Test Senaryosu Oluşturma)

Test case'ler gereksinimlere göre hazırlanan girdiler(input), olaylar ya da aksiyonlar ve bunlar sonucu oluşması beklenen sonuçların belirtildiği dokümanlardır. Test case'ler yazılımın temellerini oluşturan gereksinimler ve dizayndaki problemlerin, eksikliklerin de ortaya çıkarılmasını sağlar.

What is a test case in software testing? (Yazılım testinde test senaryosu nedir?)

- En basit biçimde, bir test case, bir test yazılımının gereksinimleri karşılayıp karşılamadığını ve işlevlerini doğru bir şekilde yerine getirip getirmediğini belirlediği bir dizi koşul veya değişkendir.
- Test case, bir tester'ın gerçekleştirdiği tek bir yürütülebilir testtir. Tek tek aşamalar takip edilerek (step by step) yapılır.
- Bir test case'i, bir şeyin davranması gerektiği gibi davrandığını doğrulamak için bir dizi talimat olarak düşünebilirsiniz.



# Test Case Development (Örnek Test Senaryosu)

Test Scenario ID	Login-1	Test Case ID	Login-1A				
Test Case Description	Login – Positive test case	Test Priority	High				
Pre-Requisite	A valid user account	Post-Requisite	NA				
Test Execution Steps:							
S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://www.facebook.com/	Facebook home	Facebook home	IE-11	Pass	[Priya 10/17/2017 11:44 AM]: Launch successful
2	Enter correct Email & Password and hit login button	Email id : test@xyz.com Password: *****	Login success	Login success	IE-11	Pass	[Priya 10/17/2017 11:45 AM]: Login successful

# Test Case Development (Örnek Test Senaryosu)

	A	B	C	D	E	F
1	TEST NO	TEST NAME	TEST STEP	RESULT	STATUS	BUG
2	1	TCKN Label Kontrolü	TCKN alanına maksimum 11 nümerik ve zorunlu karakter girilmelidir.	Eksik veya hatalı karakter girince Tooltip'de uyarı vermeli, Hata vermediye Grid'e kayıt etmelidir.	DONE	
3	2	Çalışan(EMPLOYEES) Bilgileri Kontrolü	GSM(Telefon) Combosuna 0 hariç, 10 nümerik ve zorunlu karakter girilmelidir.	Eksik veya hatalı karakter girince Tooltip'de uyarı vermeli, Hata vermediye Grid'e kayıt etmelidir.	DONE	
4	3	Çalışan(EMPLOYEES) Bilgileri Kontrolü	Maaş(SALARY) bilgisinin servisten gelmesi ve Maaş Label'ın da görüntülenmesi.	Maaş bilgisi Disable olarak gözükmeli ve Boş olmamalıdır.	UNDONE	2
5						
6						
<div><div>FRONT-END</div><div>DATABASE</div><div>SERVICE</div><div>+</div><div></div></div>						

	A	B	C	D	E
1	DATABASE ÖRNEK SENARYO				
2	TEST NO	TEST NAME	TEST STEP	RESULT	STATUS
3	1	EMPLOYEES Tablosu Kontrolü	Maaş Bilgisi EMPLOYEES.SALARY kolonunda tutulup JOBS.MIN_SALARY AND JOBS.MMAX_SALARY arasında değer almalıdır.	HR.EMPLOYEES.SALARY Nümerik olup Data Type NUMBER (8,2) ve NULL olmamalıdır.	DONE
4					
5					
6	SERVICE ÖRNEK SENARYO				
7	TEST NO	TEST NAME	TEST STEP	RESULT	STATUS
8	1	GetEMPLOYEES Servis Kontrolü	Çalışanların Adres Bilgisi GetEMPLOYEES servisten beslenmektedir.	<STREETADDRESS></STREETADDRESS> Output Tagı içinde adres bilgileri servisten alınıp Adress Label'da Disable göstermelidir.	DONE

# Test Case'leri Kim Yazar?

- Genellikle KG (QA) ekibinden tecrübeli biri test case'leri yazar. Test case hazırlamak için her ekip kendi standart şablonunu kullanır.
- Genelde Manual Tester'lar yapar ve aynı dökümanı kullanarak Automation Tester'lar scriptlerini (kodlarını) yazarlar.
- Developerlar normalde bir test data oluştururlar veya hazır olan bir datayı BA'den veya PO'dan alırlar sonra onu günceller ve geliştirirler.
- Biz de kendimiz Developerlar'dan alıp aynı test datayı geliştirir ve test ederiz. Veya Test Lead Kabul Kriterlerine göre kendisi bir test data hazırlayabilir.

## SDET (Software Development Engineer In Test ):

Bir Framework'u sıfırdan oluşturabilecek kişidir.

## Automation Tester :

Hazır yazılmış test case'leri kullanarak test yapar, oluşturulmuş framework'leri kullanır.

# Test Case Development (Örnek Test Senaryosu)

A	B	C
User Story ID	Description	Acceptance Criteria
US 0002	Facebook login (Giris sayfasi) gecersiz kimlik bilgileriyle erisilmemelidir	Gecersiz kullanıcı adı ile erişim sağlanamaz
		Geçersiz şifre ile erişim sağlanamaz
		Geçersiz kullanıcı adı ve şifre ile erişim sağlanamaz

# Test Case Development (Örnek Test Senaryosu)

A	B	C	D	E	F	G	H	I
User Story ID	Test Case ID	Test Objective	Pre-Condition	Steps	Test Data	Expected Result	Actual Result	Status
US 0002	TC_001	Gecersiz kullanıcı adı ile erişim sağlanamaz	Login erişilebilir olmalıdır //www.facebook.com/ da	1_ https://www.facebook.com/ gidi	URL 3: https://www.facebook.com kullanici adi ="pes_etmek_yok" sifre ="yapabilirim"	kullanici erisim elde edememelidir	kullanici erisim elde edemedi	Pass
				2_kullanici adi textbox a tiklayiniz				
				3_yanlis bir kullanıcı adini giriniz				
				4_sifre textbox ina tiklayiniz				
				5_dogru bir kullanıcı sifresi giriniz				
				6_login butonuna tiklayiniz				
US 0002	TC_002	Geçersiz şifre ile erişim sağlanamaz	Login erişilebilir olmalıdır //www.facebook.com/ da	1_ https://www.facebook.com/ gidi	URL 3: https://www.facebook.com kullanici adi = techproedusa@gmail.com sifresi = "yanlis_sifre"	kullanici sifre hatasi elde etmelidir ve giris izni verilmemelidir	Email adresi yanlis veerisim elde edilemedi	Fail
				2_kullanici adi textbox a tiklayiniz				
				3_dogru kullanıcı adini giriniz				
				4_sifre textbox ina tiklayiniz				
				5_yanlis bir kullanıcı sifresi giriniz				
				6_login butonuna tiklayiniz				
US 0003	TC_003	Geçersiz kullanıcı adı ve şifre ile erişim sağlanamaz	Login erişilebilir olmalıdır //www.facebook.com/ da	1_ https://www.facebook.com/ gidi	URL 3: https://www.facebook.com username ="yanlis_username" password ="yanlis_sifre"	kullanici erisim elde edememelidir	kullanici erisim elde edemedi	Pass
				2_kullanici adi textbox a tiklayiniz				
				3_yanlis kullanıcı adini giriniz				
				4_sifre textbox ina tiklayiniz				
				5_yanlis sifre giriniz				
				6_login butonuna tiklayiniz				



## 4. Test Environment Setup (Test Ortamı Kurulumu)

Testinizi nerede ve hangi ortamda gerçekleştireceğinizi bilmelisiniz ve test verilerinize de sahip olmalısınız!

- Dev (Development)
- Test (Test ortamı)
- Stage (sahne)
- Production (üretim)



## 5. Test Execution (Testin İcrası)

Her şey hazır olduğunda, planlanan tüm functionality(leri) test edebilir ve yürütebilirsiniz.

Manuel Testerlar testlerini manuel olarak yapar.

Automation (Cross-functional) Tester scriptlerini yazarak test ederler.

# ÖZET

**Test case** => Adım adım neyi test edeceğinizi açıklayan senaryolarınızdır.

**Test plan** => Test kapsamı, yaklaşım kaynakları ve amaçlanan test faaliyetlerinin zamanlamasını tanımlayan giriş ve çıkış kriterleri.

**Test execution** => Scriptler hazır olduğunda çalıştırır veya manuel olarak testinizi gerçekleştirirsiniz.

**Test data** => Functionality (leri) test etmek için kullanmanız gereken veriler.

**Acceptance criteria** => Functionality'yi test etmenin ayrıntılı tanımı.

**Expected result** => Yapılan testin beklenen sonucu

**Actual result** => Testi yaptıktan sonra ortaya çıkan netice.

**Environments (Test yapılan ortamlar (URL))** => Dev, Test, Stage, Prod

# KISA TEKRAR

## ➤ **Software Testing Nedir ve Niçin Önemlidir?**

Yazılımın güvenilirliğini kontrol etmek, ürünün müşterinin gereksinimine uygun olmasını sağlamak, istenilen kaliteye ulaşmak için testler yapılmalıdır.

## ➤ **Software Testing Life Cycle**

Gereksinim Analizi, Test Planı, Test Senaryosu Geliştirme, Test ortamı Kurulumu, Testin gerçekleştirilmesi, Test döngüsü kapanışı aşamalarını içerir.

## ➤ **User Story**

- İstenilen ürün veya özelliğin kısa ve anlaşılır tanımıdır. Bir kullanıcıya değer sağlayacak işlevselliktir. Ürün veya özelliği isteyen kişinin perspektifinden yazılır. En kısa haliyle aşağıdaki kalıpla user story oluşturulur;
- Bir (kullanıcı) olarak, (kullanıcıların memnun olduğu bir telefon araması) yapabilmek için (puanlara göre arama yapmak) istiyorum.
- User Story'nin hikayesi kullanıcının yazımı ile başlar, Scrum Team'in Product Backlog'a eklemesi ve Sprinte dahil edip üzerinde çalışması ile gerçek hayata geçmiş olur.

# KISA TEKRAR

## ➤ Test Case

- En basit biçimde, bir Test Case , bir test yazılımının gereksinimleri karşılayıp karşılamadığını ve işlevlerini doğru şekilde yerine getirip getirmediğini belirlediği bir dizi koşul veya değişkendir.
- Genellikle QA ekibinden tecrübeli biri Test Case'leri yazar.
- Test Case hazırlamak için her ekip kendi standart şablonunu kullanır.