



BATCH : B103 QA  
LESSON : **Manual Testing**  
DATE : 31.12.2022  
SUBJECT : **Bug Report & Test Plan**

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ





TECHPROEDUCATION



# SOFTWARE TEST ENGINEERING COURSE

## **MANUAL TESTING**



# BUG LIFE CYCLE

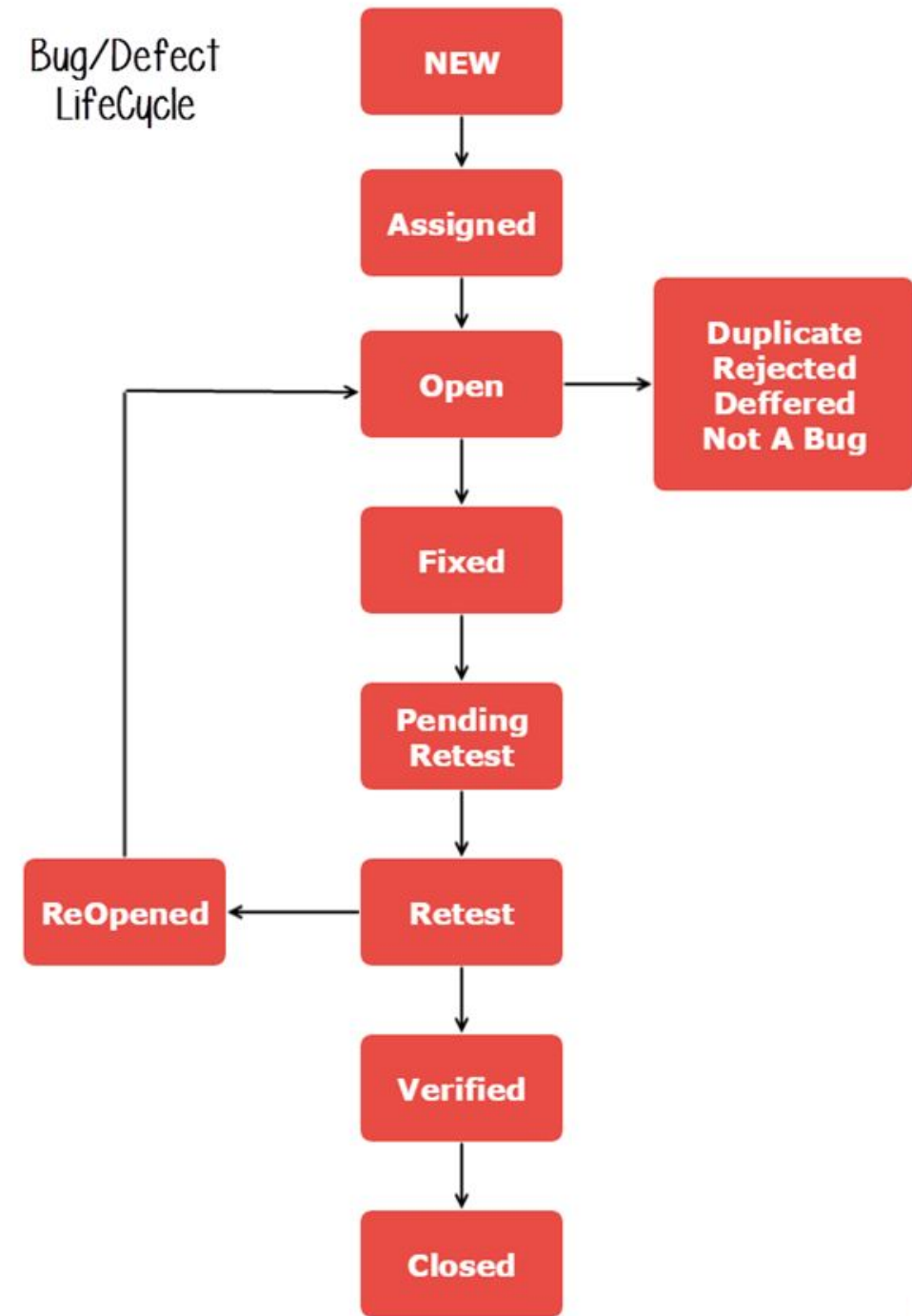
**New (Yeni)** : Bir hata ilk kez yayınlandığında, durum Yeni olur. (Jira, Alm, VersionOne gibi boardlarda..)

**Assigned (Atandı)** : Hata post edilince, test lead veya Proje lead, ilgili Developer'a "atanmış" olarak atayacaktır. (Meşru bir hata olup olmadığı ile alakalı Developerın kesinlikle bir girdisi olacaktır)

**Open (Acildi)** : Developer bu aşamada düzeltmek için hatayı analiz eder ve üzerinde çalışır.

**Fixed (Onarildi)** : Developer gerekli kod güncellemelerini yaptığında ve hatanın düzeltilip düzeltilmediğini doğruladığında, hata durumunu "Fixed" olarak değiştirebilir ve hata test ekibine aktarılır

Bug/Defect  
LifeCycle





# BUG LIFE CYCLE

## Bug Raporu (Bug Report)

Hata raporlama işlemi proje yönetim aracı üzerinde gerçekleştirilebilirken, aşağıdaki başlıklardan oluşan bir dokumanda oluşturulabilir. Bu doküman sayesinde geliştirici hatayı daha rahat anlayabilir.



**Başlık (Title) :** Hata başlığı diğer hatalarda ayırt etmek için yazılır. Hata başlığı kısa ve öz olmalıdır. Örnek vermek gerekirse *"Kullanıcı tanımlama ekranında kaydet butonuna basıldığında hata veriyor."*

**Atama (Assigned To) :** Bulunan hatanın hangi geliştiriciye veya takım liderine atanacağı bu alandan seçilmelidir.

**Öncelik (Priority) :** **Yazılım Test Planı'nda** hatalar önceliklerine göre derecelendirilmelidir. Bulunan hataların öncelik değerleri plana göre doğru şekilde girilmelidir.

**Önem Derecesi (Severity) :** Bulunan hatalar önem derecesine göre gruplandırılmalıdır. İlgili geliştirici hataları önem derecesine göre çözmelidir.

**Hata Kaynağı (Root Cause) :** Bulunan hatanın kaynağı belirtilmelidir. Bu bilgi test faaliyetleri sonucunda ölçüm ve analiz için kullanılabilir.





# BUG LIFE CYCLE

## Bug Raporu (Bug Report)

**Adımlar (Repro Steps) :** Bulunan hatanın ilgili geliştirici tarafından tekrar simüle edilebilmesi için hatanın adım adım yazılması gerekmektedir. Bunun için aşağıdaki örneği verebiliriz:

- Kullanıcı tanımlama sayfasına gidilir.
  - Ekranda yer alan tüm zorunlu alanlar doldurulur.
  - Kaydet komutu verilir.
- Beklenen Sonuç: Kullanıcı tanımlama işleminin başarıyla tamamlandığına dair bilgilendirme mesajı verilmesi gerekir.**
  - Gerçekleşen Sonuç: NullPointerException hata mesajı alındı.**

**Ekler (Attachments) :** Bulunan hatanın ekran görüntüsünü veya video kaydını alarak bu alana yüklenmesi test ekibinin hatayı geliştirme ekibine daha kolay aktarabilmesini sağlayacaktır.

	A	B	C
1	Category	Label	Value
2	Bug ID	ID number	#123
3		Name	CART - Unable to add new item to my cart
4		Reporter	Mike A
5		Submit Date	03/04/16
6	Bug overview	Summary	When my cart contains one item, I am unable to add a second item via the add to cart button on a product page
7		URL	www.example.com/product/abc
8		Screenshot	www.example.com/screenshot123
9	Environment	Platform	Macintosh
10		Operating System	OS X 10.12.0
11		Browser	Chrome 53
12	Bug details	Steps to reproduce	add one item to cart > go to product abc via the search bar > add new item to cart via "add to cart" button (see screenshot) > go to cart
13		Expected result	The cart should contain 2 items
14		Actual result	The cart contains only 1 item
15		Description	/
16	Bug tracking	Severity	Major
17		Assigned to	/
18		Priority	High
19	Notes	Notes	/





## 2- Test Planning (Test Planlamasi)



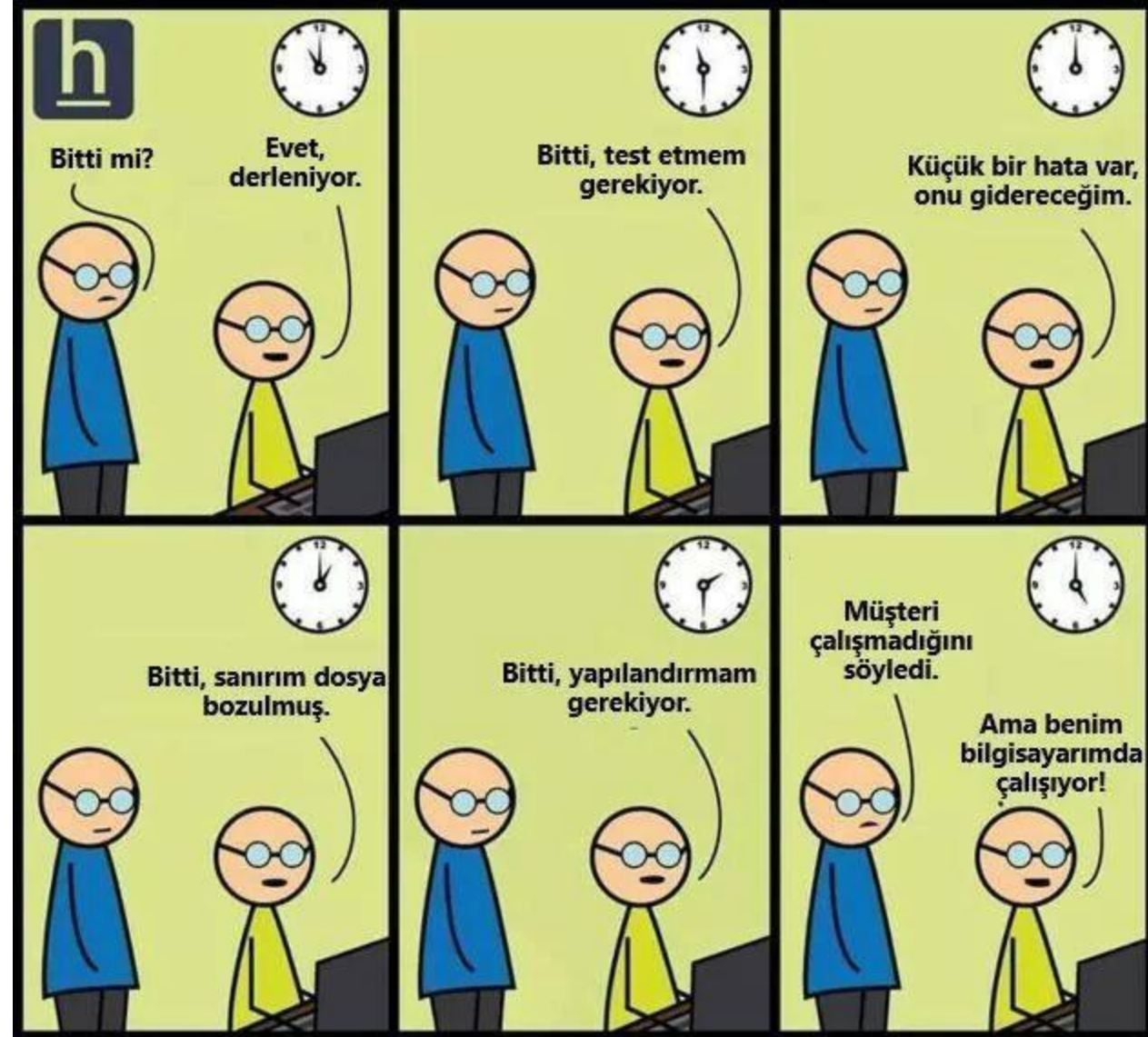
- Neyin test edilmesi gerektiğine dair genel bir fikir topladıktan sonra, testler için 'plan yapılır'.
- Test Planı belgesi, Ürün Açıklaması, Yazılım Gereksinimi Spesifikasyonu (Software Requirement Specification SRS) veya Kullanım Senaryosu Belgelerinden (Use Case Documents) türetilmiştir.
- Amaçlanan 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 Planı

Yazılım projelerinde test faaliyetlerine başlamadan önce kuşkusuz yapılması gereken ilk iş planlamadır. Yazılım Test Planı (YTP) gerçekleştirilecek test faaliyetlerinin genel bir planlamasını içerir. YTP dokümanının hazırlanmasında asıl sorumlu test ekibidir ancak test ekibi çeşitli konuları (test takvimi, giriş-çıkış kriterlerinin belirlenmesi vb.) proje yöneticisiyle konuşarak YTP dokümanına nihai şeklini verir.







# Test Planı Hazırlarken Nelere Dikkat Edilmelidir?

Test planı hazırladıktan sonra dikkat edilmesi gereken en önemli konu, test sırasında oluşabilecek sapmaların tespit edilip plana uyarlanmasıdır. Aksi hâlde oluşabilecek hataların ve çözümlerinin plan doğrultusunda yönetilebilir olması oldukça güçleşir.

Test planları öncelikle fikirlerin toparlanmasına ve güncellemelerin yönetilmesine yardımcı olur ve iletişim kurmaya yarar. Bu nedenle iyi bir test planı:

- Hangi maddelerin test kapsamında olup olmadığı,
- Test amaçlarının ne olduğu,
- Proje ve ürün risklerinin neler olduğu,
- Proje ve ürün için en önemli olguyu,
- Kısıtları,
- Nelerin test edilmeye müsait olduğu,
- Test için gerekli yazılım/donanımı,
- Zaman planı gibi bilgileri içermelidir.
- Kısa ve amaca odaklanmış olmalıdır.





# Test Planlama Adımları

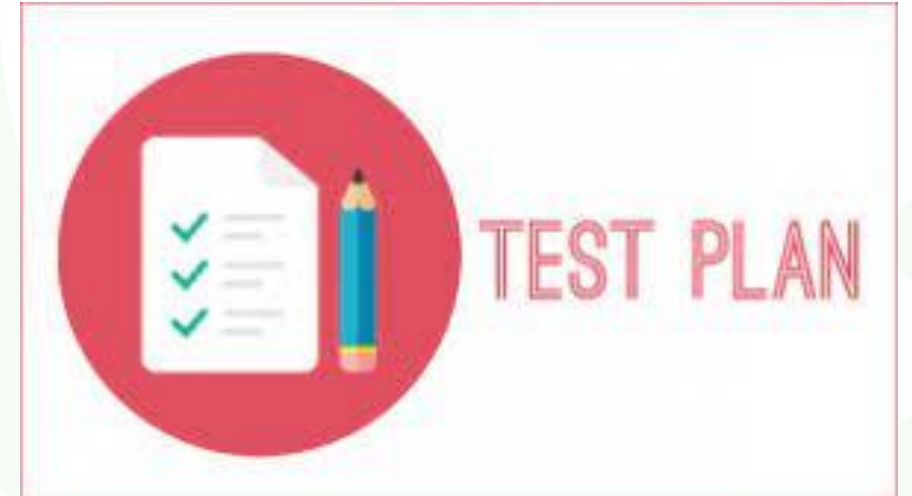
Hedef	•Kapsamın ve risklerin tanımlanması ve hedeflerin belirlenmesi
Tanım	•Test seviyelerinin, giriş çıkış kriterlerinin genel yaklaşımın tanımlanması
Koordine	•Test aktivitelerini, yazılımın yaşam döngüsü ile koordine ve entegre edilmesi
Karar	•Test kapsamının, test yönteminin ve sonuçların nasıl değerlendirilmesi konusunda karar verilmesi
Zaman	•Test Analizi ve Tasarım aktivitelerinin zaman planlamalarının yapılması
Zaman	•Testin uyarlanması, koşulması ve değerlendirilme zamanının planlanması
Kaynak	•Tanımlanan aktiviteler için kaynak ataması yapılması
Şablon	•Test dokümantasyonu için yapı ve şablonun tanımlanması
Metrik	•Testin hazırlığı, koşulması, hata yönetimi konularının kontrolü için yöntem belirleme
Prosedür	•Test prosedürlerinin ayrıntılarının seviyelerini belirleme



# Test Planlama Adımları

Bir Yazılım Test Planı dokümanı aşağıdaki ana başlıklardan oluşmalıdır.

- Test Stratejisi ve Test Edilecek Bileşenler
- Giriş ve Çıkış Kriterleri
- Hata Yönetimi
- Risk Yönetimi
- Görev ve Sorumluluklar
- Test Ortamı ve Test Araçları
- Test Takvimi







# Test Planlama Adımları

## Test Stratejisi ve Test Edilecek Bileşenler

Test edilecek modüller, kullanılacak test seviyeleri ve test teknikleri bu bölümde tanımlanır.

Örnek metin aşağıdaki gibidir.

X projesinde yazılım test faaliyetleri süresince birim, entegrasyon ve sistem testleri gerçekleştirilecektir.

Aşağıdaki yazılım modülleri bu plan kapsamında belirtilen testlere tabi tutulacaktır. (A Modülü, B Modülü, C Modülü)



# Test Planlama Adımları

## Giriş ve Çıkış Kriterleri

Plan kapsamında belirtilen testlerin hangi koşulda başlatılacağı ve hangi koşullar sağlandığında tamamlanacağı bu bölümde ayrıntılı bir şekilde tanımlanmalıdır. Giriş-Çıkış kriterleri proje yöneticisi ile beraber belirlenmektedir.

Başlama (Giriş)  
zamanını  
belirleme

Bitiş (Çıkış)  
zamanını  
belirleme

Başlama ve Bitiş  
kriterlerini  
belirleme



# Test Planlama Adımları

## Hata Yönetimi

Hata önemine göre derecelendirme kriterleri belirlenir. Bulunan hataların hangi proje ve iş yönetimi aracında (TFS, JIRA vb.) kayıt açılacağı belirtilir. Örnek metin aşağıdaki gibidir.

X projesi kapsamında bulunan hatalarda aşağıdaki derecelendirme kriterleri kullanılacaktır:

1. Derece Hatalar: Sistemin genel çalışmasını direkt etkileyen hatalardır.
2. Derece Hatalar: Sistemin genel çalışmasını direkt etkilemeyen fakat işlevsel olarak bazı kısımların çalışmasını engelleyen hatalardır.
3. Derece Hatalar: Sistemin çalışmasını engellemeyen, görsel yönden ortaya çıkan hatalardır.

X projesi kapsamında bulunan hatalar, JIRA altında kayıt altına alınacaktır. Bu hata kayıtları önem ve öncelik derecesine göre işaretlenecektir.





# Test Planlama Adımları

## Risk Yönetimi

Proje boyunca oluşacak her türlü risk Proje Risk Veri Tabanı ile kayıt altına alınmalıdır. YTP dokümanında ise test faaliyetlerinin başlamasını veya tamamlanmasını etkileyecek her türlü risk bu kısımda belirtilmelidir. Test faaliyetlerinin ilerlemesini engelleyebilecek riskler aşağıdaki gibi olabilir:

- Gereksinimlerin çıkarılması geciktiğinde test adımları ve test senaryolarının yazılması gecikebilir.
- Test ekibi farklı projelerde görev yapması durumunda test faaliyetleri aksayabilir.
- Proje kapsamında geliştirilecek modüllerin zamanında tamamlanmaması sonucunda test faaliyetleri başlamayacaktır.



# Test Planlama Adımları

## Görev ve Sorumluluklar

Proje kapsamında test seviyelerine göre görev dağılımı ve sorumluluklar bu kısımda tanımlanacaktır.

Test Seviyeleri Görev ve Sorumluluklar	
Birim Testleri	Geliştirme Ekibi
Entegrasyon Testleri	Test Ekibi
Sistem Testleri	Test Ekibi
Kabul Testleri	Müşteri ve Test Ekibi



# Test Planlama Adımları

## Test Ortamı

Test faaliyetlerinin yürütülmesi için kullanılacak bilgisayarların / sunucuların donanımsal ve yazılımsal özellikleri bu kısımda detaylı bir şekilde yazılacaktır. Test otomasyon araçları kullanılacaksa bunlar hakkında bilgi (versiyonu gibi) eklenmelidir. Testin yürütüleceği platformlarda (Windows, Linux, MacOS vb.) ayrıntılı şekilde belirtilmelidir. Eğer mümkünse test ortamını anlatan basit ve anlaşılır bir şema çizerek diğer paydaşlarında test ortamı hakkında bilgi edinebilmeleri sağlanır. Test Ortamı ve Test Araçları bölümü aşağıdaki şekilde yazılabilir.

X projesi kapsamında entegrasyon ve sistem test faaliyetleri 192.168.1.254 IP adresine sahip sunucu üzerinde yürütülecektir. Sunucuya ait özellikler aşağıdaki gibidir:

CPU: Intel Xeon 3.3 GHz

RAM: 8 GB DDR4 2133 MHz

HDD: 100 GB

OS: Windows Server 2012 R2

Yüklenecek Programlar: Java 8 Update 211 x86





# Test Planlama Adımları

## Test Araçları Versiyon Kullanım Amacı

**JUnit 4.0** Birim testleri için geliştirme ekibi tarafından JUnit kütüphanesi kullanılacaktır.

**Selenium WebDriver 3.141.0** Sistem testleri için test ekibi tarafından SeleniumWebDriver test otomasyon aracı kullanılacaktır.

**Apache JMeter 4.0** Fonksiyonel olmayan testler (Performans, Yük ve Stres) için Apache JMeter aracı kullanılacaktır.

**JIRA 2013** Test faaliyetleri sonucunda bulunan hatalar JIRA üzerinde ilgili geliştiriciye hata kaydı olarak açılacaktır.



# Test Planlama Adımları

## Test Takvimi

Test seviyelerine göre test faaliyetlerinin başlangıç, bitiş tarihleri ve tahmini süre bilgisi belirtilmelidir. Test takvimine Proje Yönetim Planında da yer verilebilir, bu durumda Proje Yönetim planını referans olarak gösterebilirsiniz. Referans olarak göstermek istemiyorsanız proje yöneticinizle belirlediğiniz test takvimini aşağıdaki tabloda olduğu gibi gösterebilirsiniz.

Test Seviyeleri	Gün	Başlangıç Tarihi	Bitiş Tarihi
Birim Testleri	90	01.01.2019	01.04.2019
Entegrasyon Testleri	90	01.04.2019	01.07.2019
Sistem Testleri	120	01.07.2019	01.11.2019



# Yedi Test Prensibi

1. Testin amacı, yazılımda hataların olduğunu göstermektir; yazılımda hata kalmadığını ispatlamak değildir.
2. Yazılımı %100 test etmek imkansızdır.( Exhaustive testing is impossible)
3. Erken test, zaman ve para tasarrufu sağlar.( Early testing saves time and money)
4. Hatalar yazılımın belli alanlarında yoğunlaşır(Defects cluster together)
5. Antibiyotik direnci(Beware of the pesticide paradox)
6. Yazılım testi, projenin bağlamına(çeşidi, türüne), koşullarına göre değişiklik gösterir(Testing is context dependent)
7. Yeni hata bulamıyoruz başarılı bir yazılım elde ettik yanılgısı (Absence-of-errors is a fallacy)