
SDLC

Software Development Life Cycle

Gun 7

24 Ekim 2020

Bug Life Cycle

Genel Tekrar

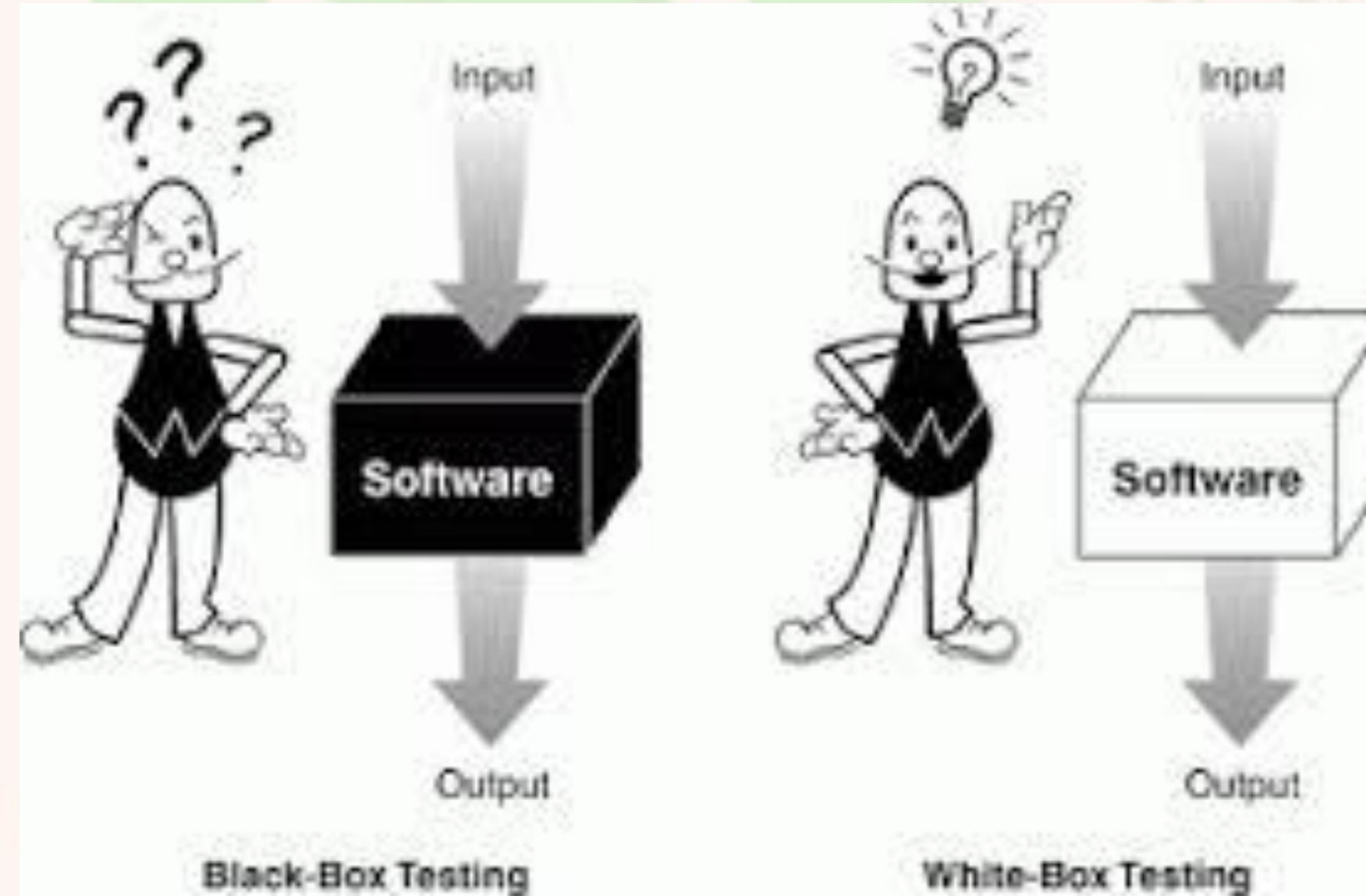
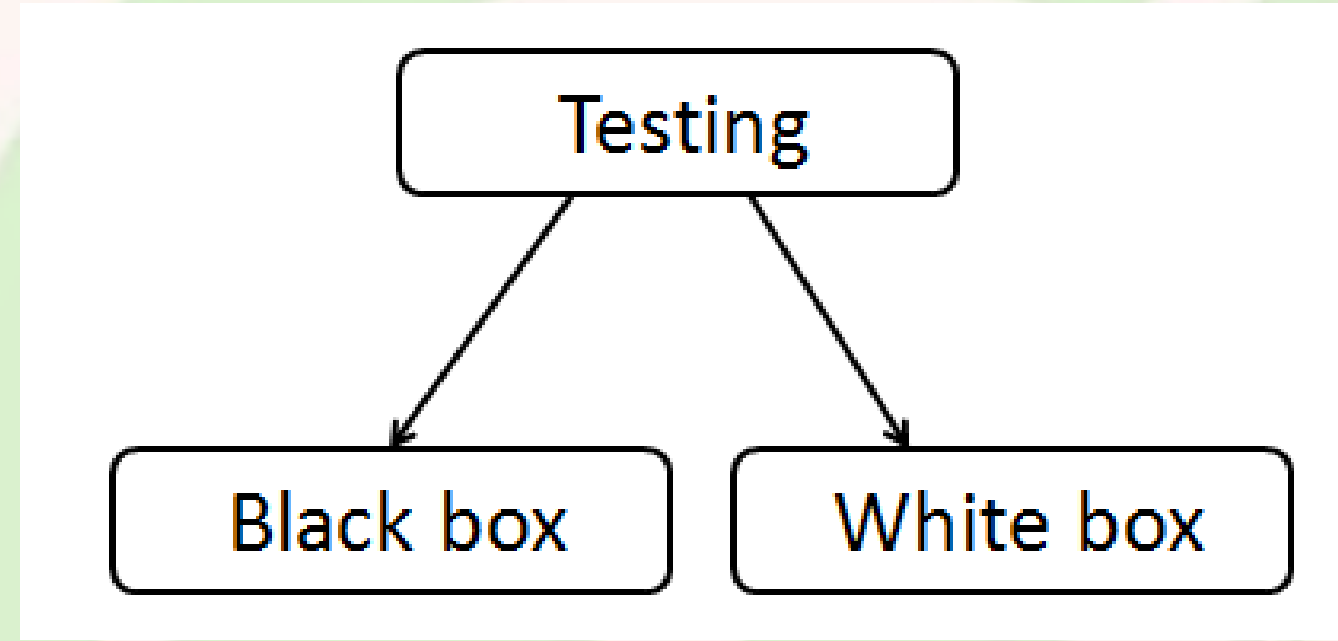
Mehmet Bulutluoz

TECHPROED

KISA TEKRAR

Black Box Testing

Kriter	Kara Kutu Testleri
Tanım	Kara kutu testleri yazılımın iç yapısı, tasarım ve gerçekleştirme detayı bilinmeden testlerin tasarlandığı bir yazılım test tasarım tekniğidir.
Test Seviyesi	Kara kutu testleri, <u>Kullanıcı Kabul Testleri (UAT)</u> , Sistem Testleri ve Yazılım Testleri seviyelerinde kullanılabilir.
Kim Yapar?	Genellikle yazılım test uzmanları tarafından gerçekleştirilir.
Kodlama Bilgisi	Gerekli değildir
Test Tanımı için gerekenler	Gereksinim belirtilimleri



White Box Testing

Beyaz Kutu Testleri

Beyaz kutu testleri, yazılımın iç yapısı, tasarım ve gerçekleştirme detayı bilinerekten testlerin tasarlandığı bir yazılım test tekniğidir.

Beyaz kutu testleri ise daha alt seviyelerde olan birim ve entegrasyon test seviyelerinde kullanılmaktadır.

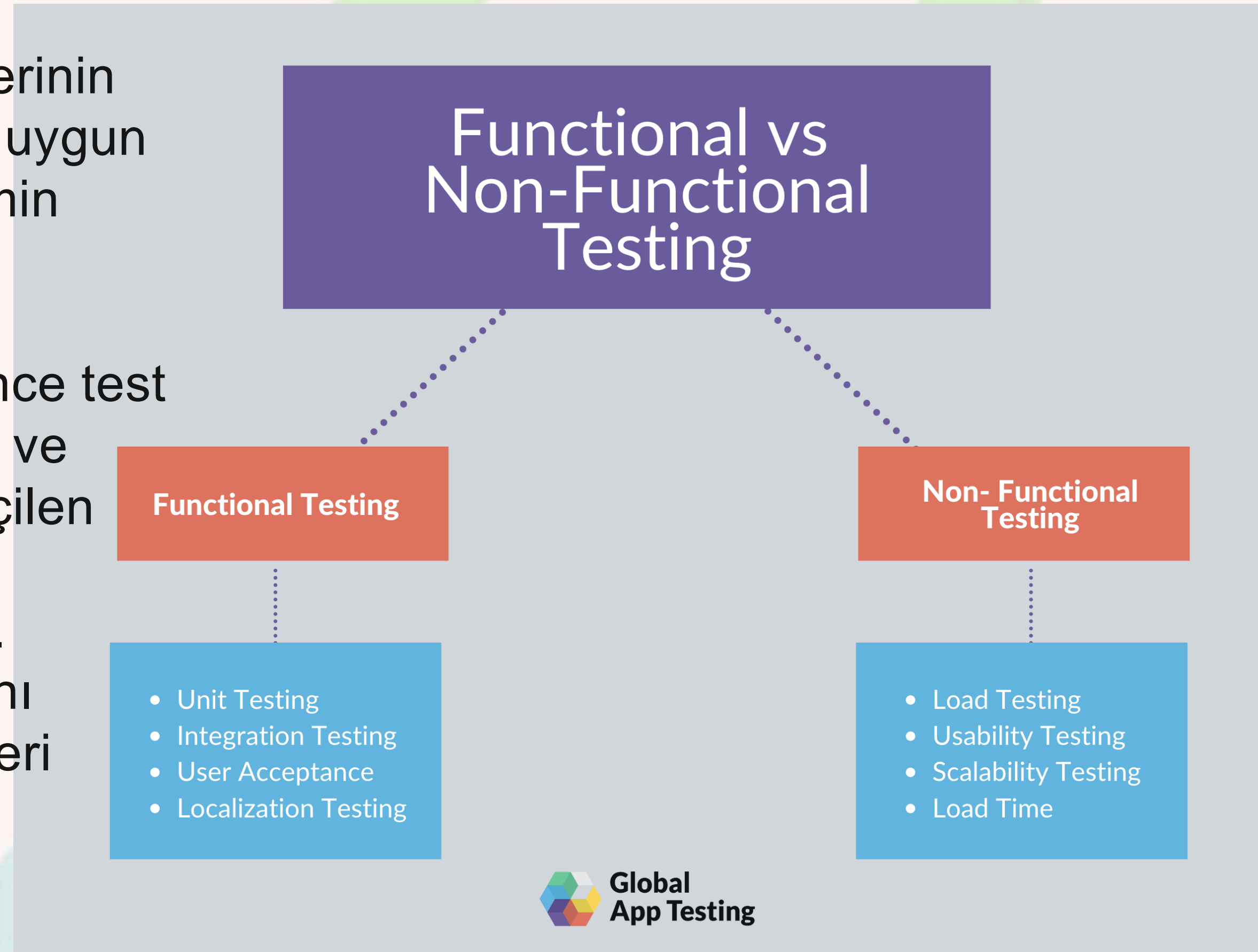
Genellikle, yazılım geliştiriciler tarafından gerçekleştirilir. Gerekli durumlarda testlerin tasarımında ve test yazılımlarının kullanımında test uzmanları yazılım geliştiricilere destek verebilir.

Mutlaka gerekir

Detaylı tasarım

KISA TEKRAR

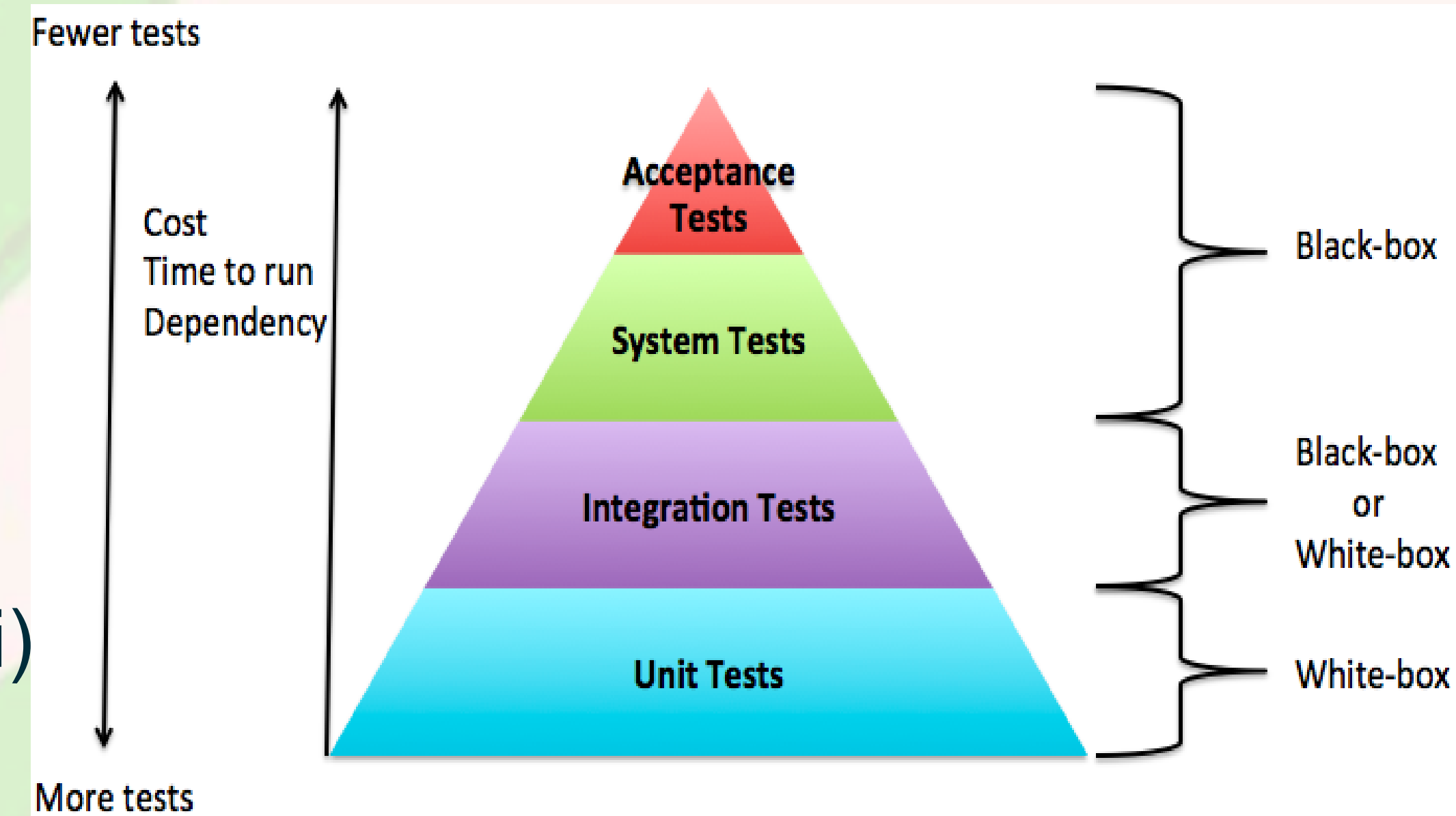
- Bir uygulamanın işlevlerinin gerekli spesifikasyona uygun olarak çalıştığından emin olmak için yapılır.
- Fonksiyonel testi gerçekleştirmek için önce test girdisini tanımlamamız ve beklenen sonuçları seçilen test girdi değerleri ile hesaplamamız gerekir.
- Sonra test senaryolarını yürütür ve gerçek verileri beklenen sonuçla karşılaştırırız.



- Uygulamanın beklenen iş yükü altında sorunsuz çalışmasını sağlamak için performans testi yapılır.
- Amaç, güvenilirlik, kaynak kullanımı vb. gibi performans sorunlarını bulmaktır.
- Performans testi yaparken aklımızda tutmamız gereken üç ana nokta hızlı yanıt, maksimum kullanıcı yükü ve çeşitli ortamlarda kararlılıktır.

KISA TEKRAR

- 1- Unit Testing (Dev) (White testing)
- 2- Integration Testing (Black Box)
- 3- Functional Testing (Fonksiyon Testi) (Black Box)
- 4- Regression Testing (Tekrar Onay Testi)
 - Kısmi Regresyon
 - Tam Regresyon
- 5- Sanity veya Smoke Testing (Sağlık / Duman Testi)
- 6- System Testing (End To End Testing)
- 7- Ad-hoc Testing (Gecici Test)t
- 8- User Acceptance Testing UAT (Kullanıcı Kabul Testi)



TECHPROED

Bug Life Cycle

Bug / defect nedir?

Application da Gereksinimlere uygun olmayan her hangi bir davranis

Unexpected behaviour of an application

Yazılım testinde **Kusur Yaşam Döngüsü** veya Hata Yaşam Döngüsü, kusur veya hatanın ömrü boyunca geçtiği belirli durumlar kümesidir.

Kusur yaşam döngüsünün amacı, tespit edilen kusurun mevcut durumunu raporlamak, ilgili kisilere atamak ve Bug Defect (Hata Duzeltme) sürecini sistematik ve verimli hale getirmektir.



Bug Life Cycle

Testerlar **B**ug bulduklarında ne yapmalıdır ?

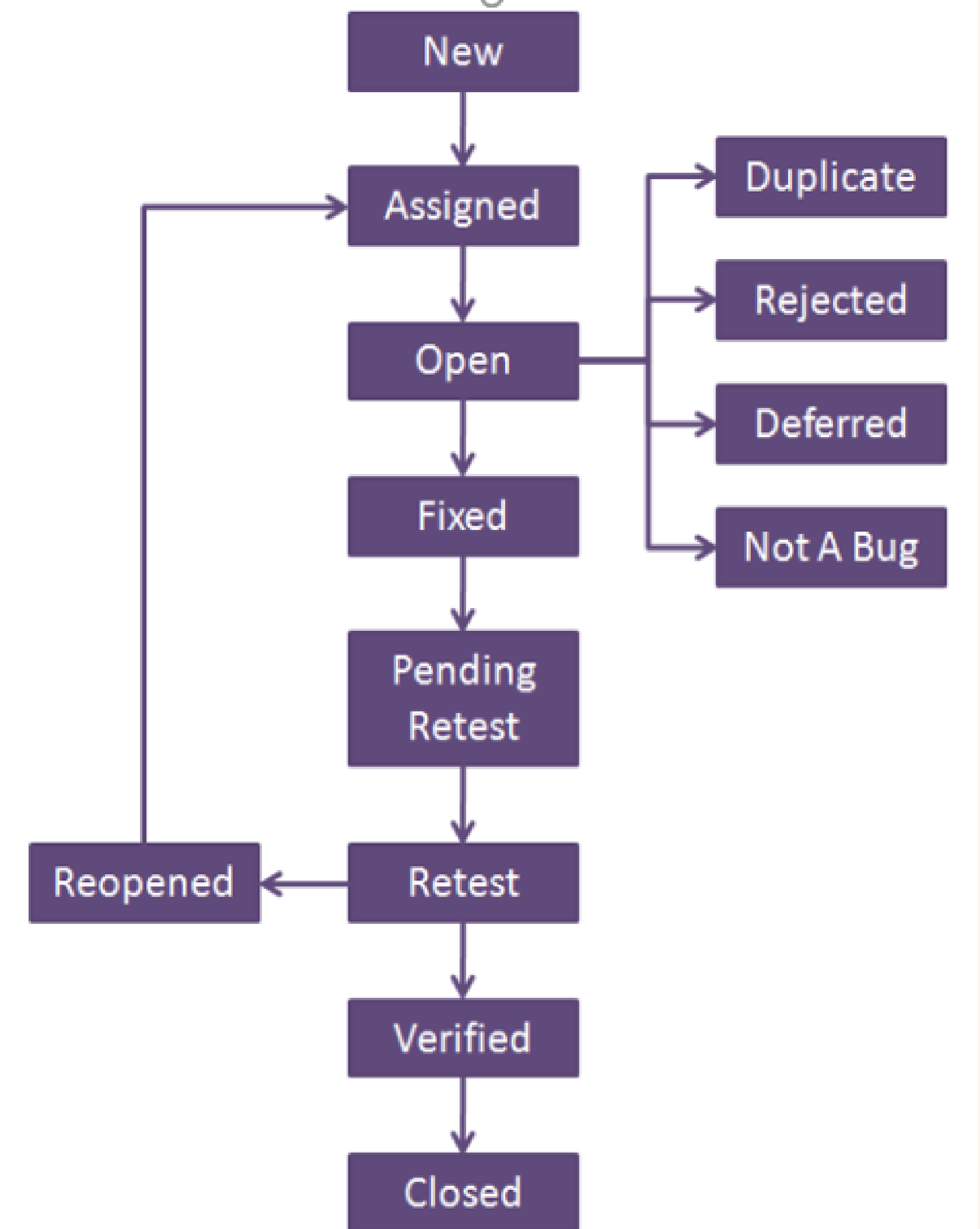
- Her şeyden önce, testerlar buldukları hatanın meşru bir hata (geçerli bir hata) olduğundan emin olmalıdır.
Bunun için testini tekrar etmelidir. (Reproduce)
- Bir defect buldugunuzda, sorunu ekip içindeki diğer testerlar ile tartismali (beraber gözden gecirmeli), böylece geçerli bir sorun olup olmadığı konusunda daha iyi bir fikre sahip olmalıdır.
- Test ekibi olarak mutabik kalırsa, raporlamadan önce developer ile gorusulebilir. Onun fikrini de alarak oluşturulacak Bug Ticket daha verimli olacaktır.



TECHPROED

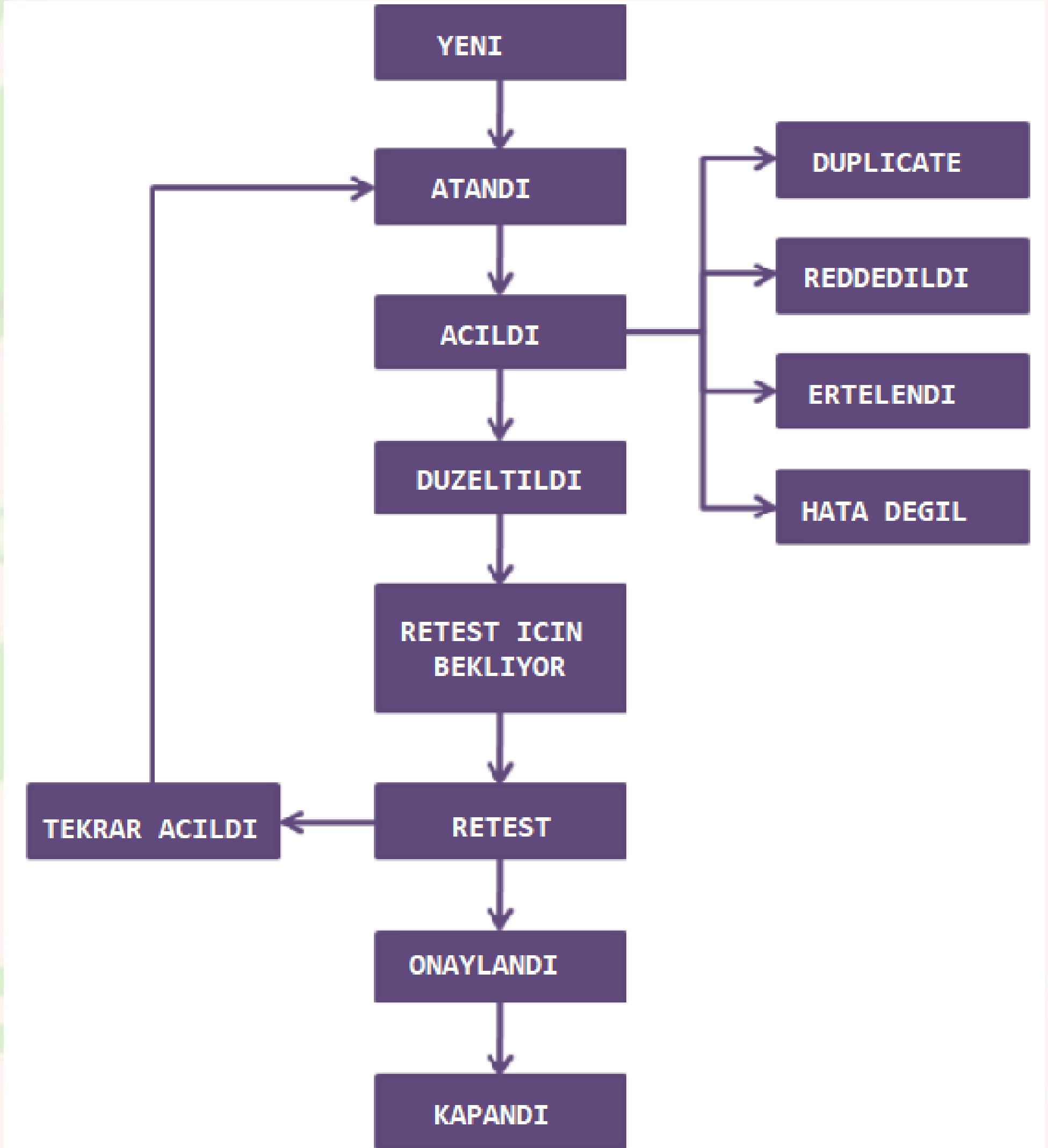
Bug Life Cycle

- **Status: New (Yeni)** Bir hata ilk kez yayınlandığında, durum Yeni olur. (Jira, Alm, VersionOne gibi boardlarda..)
- **Status: Assigned (Atandı)** Hata post edilince, test lead veya Proje lead, ilgili Developer'a "atanmış" olarak atayacaktır. (Meşru bir hata olup olmadığı ile alakalı Developerin kesinlikle bir girdisi olacaktır)
- **Status: Open (Acildi)** Developer bu aşamada düzeltmek için hatayı analiz eder ve üzerinde çalışır.
- **Status: Fixed (Onarıldı)** Developer gerekli kod güncellemelerini yaptığında ve hatanın düzeltilip düzeltilmediğini doğruladığında, hata durumunu "Fixed" olarak değiştirebilir ve hata test ekibine aktarılır.
- **Status: Pending retest (Tekrar Test İçin Bekliyor)** Arızayı giderdikten sonra Developer, yeniden test için bu özel functionalityyi deploy eder. Burada testler test aşamasında beklemede olur. Ve durum tekrar test etmeyi gerektirir.



Bug Life Cycle

- **Status: Retest (Tekrar Test)** Bu aşamada, tester işlevselliği farklı test verileriyle tekrar test eder ve buldukları defect ilgili her şeyi kapsadığından emin olur.
- **Status: Verified (Onaylandı)** Tester doğrulama yaptıktan ve hata yazılımda görünmediğinde, hatanın düzeltildiğini onaylar ve durumu “Doğrulandı” olarak değiştirir.
- **Status: Reopen(Tekrar Acildi)** Hata Developer tarafından düzeltildikten sonra bile hata hala mevcutsa, tester durumu "Yeniden Açıldı" olarak yapar. Bu yüzden hata aynı süreçten bir kez daha geçer.
- **Status: Closed =>** Hata giderildikten sonra tester tarafından test edilir. Tester yazılımda defect bulunmadığından emin olursa, hatanın durumunu “Kapalı” olarak değiştirir. Bu, hatanın düzeltildiği, test edildiği ve onaylandığı anlamına gelir.



Bug Life Cycle

Bug Ticket oluşturulduğunda Developer'ların cevapları

Rejected (Reddedildi): Kusur Developer tarafından meşru bir kusur olarak kabul edilmezse Developer tarafından 'Reddedildi' olarak işaretlenir.

Duplicate (Tekrar): Developer hatayı diğer defectlerle aynı bulursa, hata durumu Developer tarafından "cift" olarak değiştirilir.

Deferred (Ertelendi): Developer, hatanın çok önemli bir öncelik olmadığını ve sonraki sürümlerde giderilebileceğini düşünüyorsa, hatanın durumunu 'Ertelenmiş' olarak değiştirebilir.

Not a Bug (Bug Degil) Kusurun uygulamanın işlevselliği üzerinde bir etkisi yoksa, kusurun durumu 'Hata Değil' olarak değiştirilir.
