

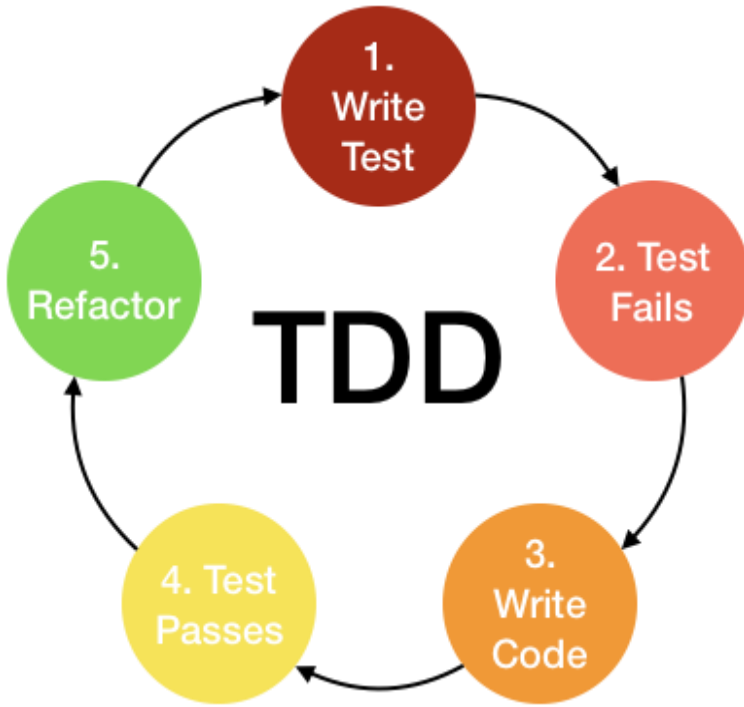
## TDD(Test Driven Development)

### TDD(Test Driven Development) Nedir?

TDD'yi Türkçe'ye Test Güdümlü Geliştirme olarak çevirebilir. Kent Beck tarafından Extreme Programming'in bir parçası olarak bulunmuş olan bir programlama tekniğidir. Tarih açısından bakacak olursak 1994 yılında Beck, Smalltalk için SUnit test framework'ünü yazmıştır. 1998 yılında XP'de "testleri çoğunlukla ilk sırada yazmalıyız" diye ifade ettiği Test First kavramını ortaya artmıştır. 2002 Kasım'ında piyasaya çıkan Test Driven Development: By Example isimli kitabında Test-First kavramını Test Driven Development olarak daha da detaylı bir teknik olarak ele alınmıştır.

TDD çok basit bir kaç adımdan oluşmaktadır.

1. Bir test yazılır.
2. Test başarısız olur.
3. Test başarılı hale getirilir.
4. Mevcut bütün testlerin başarılı olması sağlanır.
5. Kod refactor edilir. Yani kodda iyileştirme ve(ya) temizleme yapılır.



Önceden kodu yazıp sonradan bunların testini yazmanın TDD olmadığını özellikle belirtmek gerekmektedir. TDD mutlaka test ilk sırada yazılmalıdır.

### TDD vs Geleneksel Test

Test driven development yaklaşımı, kaynak kodunuzun doğrulayıcı düzeyde test edilmesini sağlar. Geleneksel testlerde ise başarılı bir test, bir veya daha fazla kusur bulur.

Test driven development, sisteminizin kendisi için tanımlanan gereksinimleri karşılamasını sağlayarak sisteminiz hakkında güveninizi geliştirmenize yardımcı olur.

Geleneksel testler, test senaryosu tasarımına daha fazla odaklanılır.

TDD’de geleneksel testlerden farklı olarak her kod satırı test edilme sürecinden geçer.

Hem geleneksel test hem de TDD, sistemin test edilmesinin önemine yol açar.

#### TDD Avantajları

TDD, programcının “belki ilerde kullanılır, bu metodu eklemekte fayda var” düşünce tarzını engeller. Bu sayede TDD, proje maliyetini düşürür.

TDD ile test kapsama alanı oldukça geniştir.

Testler koda olan güveni artırır.

Testler sistemin nasıl çalıştığını gösteren dokümantasyon olarak düşünülebilir.

Design first mentalitesi kazandırır ve over-engineeringten kaçınmamızı sağlar.

#### TDD ile Geliştirme Yaparken Dikkat Edilmesi Gerekenler

Test kodlarımız normal kodlar gibi ele alınmalı gelişime açık kodlar tasarlanmalıdır.

Testler yazılırken negatif caseler’e de önem verilmeli.

Testler bir işin nasıl yapıldığından çok ne yapılmaya çalışıldığıyla ilgilenmelidir.

Testlerin çalışma sırası birbirini etkilememelidir.

#### TDD ile Geliştirme Yaparken Dikkat Edilmesi Gerekenler

Test kodlarımız normal kodlar gibi ele alınmalı gelişime açık kodlar tasarlanmalıdır.

Testler yazılırken negatif caseler’e de önem verilmeli.

Testler bir işin nasıl yapıldığından çok ne yapılmaya çalışıldığıyla ilgilenmelidir.

Testlerin çalışma sırası birbirini etkilememelidir.

Görseller anlık görüntü sürecinizi geliştirebilir ancak TDD’nin yerini alamaz. Bunun yerine, görsel regresyon ve anlık görüntü araçları görsel stili test etmenize yardımcı olabilir. Görsel tasarımlar sıklıkla değişme eğilimindedir ve genellikle nesnel olmaktan çok öznel. Bir şeyin nasıl görünmesi gerektiğine ve bu görünümün biçimlendirme ve stillerle nasıl elde edilebileceğine dair bilimsel bir tahmin bulmak zordur.

Yararlanılan Kaynaklar;

<https://talentgrid.io/tr/test-driven-development-nedir/>

<https://medium.com/@nerobianchi/tdd-nedir-965c1a26e68f>