

# **Yazılım Geliřtirmede Çevik Yöntemler**

Hafta 1

Çevik Yazılım Geliřtirme Yöntemlerine Giriř

Sibel Kuzgun Akın

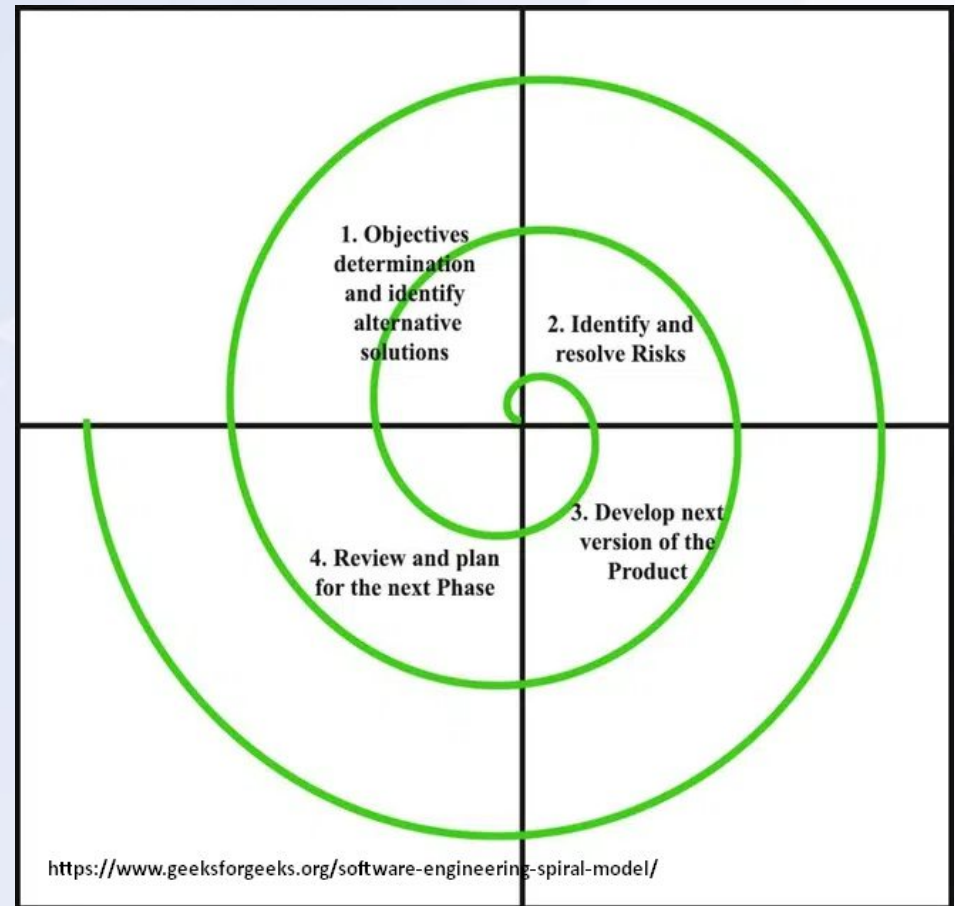
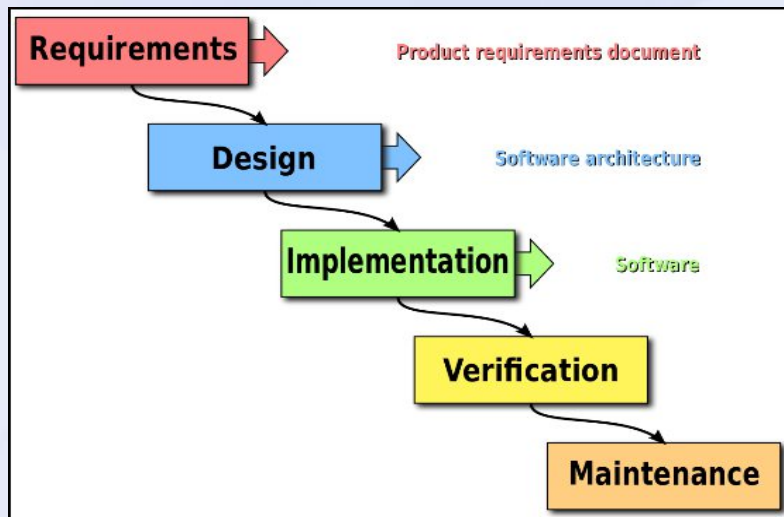
# Yazılım Geliştirme Aşamaları

- İhtiyaç analizi ve planlama
- Tasarım
  - Yazılım mimarisi
  - Veri tabanı
  - Kullanıcı arayüzü (UI)
  - Kullanıcı deneyimi (UX)
- Kodlama
- Test (sınama)
- Belgelendirme
- Devreye alma
  - Uyarlama (*implementation*)
- Bakım (*maintenance*)

# Yazılım Geliştirme Yaklaşımları

## Bazı klasik yaklaşımlar:

- Şelale (*Waterfall*)
- Prototip geliştirme
- Sarmal model



# Çevik Yaklaşımlar

- Uç programlama (*Extreme programming*)
- Scrum
- Kristal yöntemler
- Test güdümlü geliştirme (*Test driven development*)
- Özellik güdümlü geliştirme (*Feature driven development*)
- Kanban
- Yalın yazılım geliştirme
- Minimum uygulanabilir ürün
- Açık birleşik süreç (*OpenUP*)
- Dinamik sistem geliştirme yöntemi (*DSDM*)

# Çevik Yazılım Geliştirme



# Çevik Yazılım Geliştirme Bildirgesi

Hem kendimiz yaparak, hem de diğerlerinin yapmasına yardım ederek, yazılım geliştirmenin daha iyi yollarını ortaya koyuyoruz:

- Bireyler ve etkileşim, süreçler ve araçlardan...
- Çalışan yazılım, anlaşılır belgelendirmeden...
- Müşteri ile işbirliği, sözleşmeler üzerinde pazarlıktan...
- Değişiklikleri yönetmek, bir plana uymaktan...

Sağdaki şeyler de değerlidir ama soldaki şeyler daha değerlidir.

© 2001, the Agile Manifesto authors

This declaration may be freely copied in any form, but only in its entirety through this notice.



# Çevik Yazılım Geliştirme İlkeleri

1. Erkenden ve sürekli olarak yazılımın hazır kısımlarını devreye alarak müşteri memnuniyeti.
2. Değişen ihtiyaçları hoşgörüyle karşılamak, proje ilerlemiş olsa bile...
3. Sık sık çalışan yazılım kısımlarının devreye alınması (aylardansa haftalar tercih edilir)
4. İş insanları ile yazılım geliştirenler arasında yakın ve günlük işbirliği
5. Projeler güvenilebilecek azimli ve hevesli bireyler etrafında şekillenir.
6. Yüz yüze konuşmak iletişimin en iyi şeklidir. (Aynı yerde olmak)

# Çevik Yazılım Geliştirme İlkeleri

7. Projede ilerlemenin ilk ölçüsü çalışan yazılımdır.
8. Sabit bir tempoyla sürdürülebilir geliştirme
9. Teknik mükemmellik ve iyi tasarıma sürekli dikkat etmek
10. Basitlik —yapılmamış işleri maksimize etme sanatı— esastır.
11. En iyi mimari, ihtiyaç analizi ve tasarımlar kendi kendini organize eden takımlardan çıkar.
12. Geliştiriciler takımı düzenli olarak daha etkin nasıl çalışabileceğine kafa yorar ve kendini buna göre düzenler.

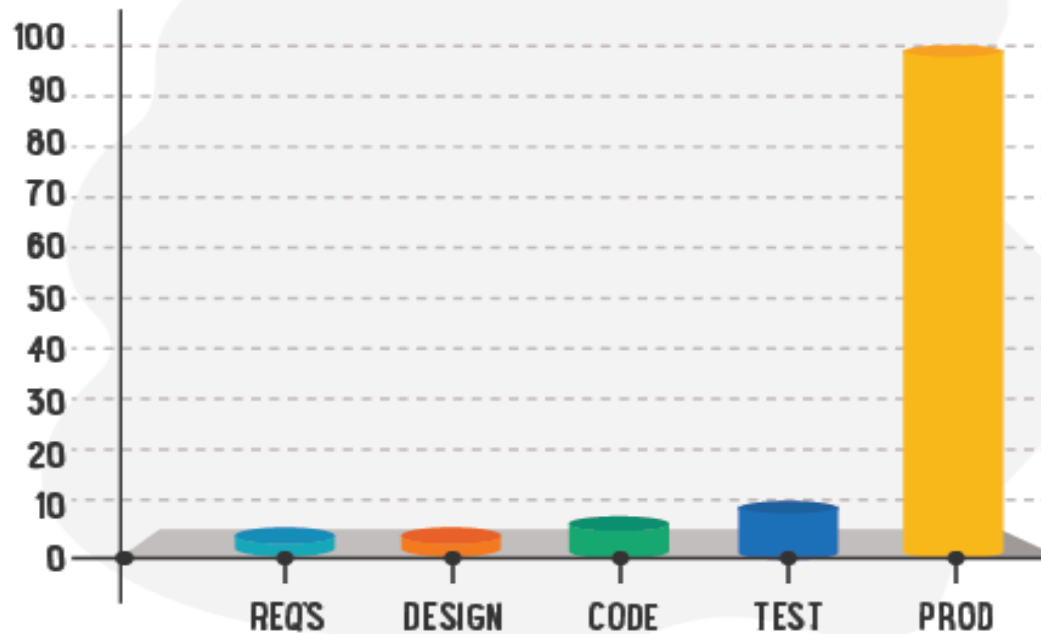


# Çevik – Şelale Karşılaştırması

- Çevik yazılım geliştirme yöntemleri ile şelale yöntemi arasındaki farklardan biri yazılım kalitesi ve testlerine yaklaşımlarıdır.
- Şelale yönteminde her zaman kodlama/geliştirme aşaması bitirilir, ondan sonra sinama/test aşaması vardır.
- Çevik yaklaşımlarda program parçaları geliştirme döngüsü içinde, yazıldıkça test edilir.

**“Kötü bir şeyi olmadan önlemek,  
olduktan sonra düzeltmekten daha iyidir.”**

## THE RELATIVE COST OF FIXING DEFECTS



ILLUSTRATED BY SEGUE TECHNOLOGIES

<https://www.seguetech.com/rising-costs-defects/>