

# Çevik Uygulama Kılavuzu

- Ne zaman çevik uygulanmalı
- Çevik manifesto ve düşünce yapısı
- 12 temel prensip
- Bütünsel takım yaklaşımı
- Erken ve sık geribildirim
- Günlük toplantı
- Retrospektifler
- Sürüm ve yineleme planlama
- İşbirlikçi kullanıcı hikayesi oluşturma
- Gösteriler ve gözden geçirmeler
- Sürekli tümleştirme
- Liderlik
- SCRUM
- KANBAN
- XP

## Yaşam döngüsü seçimi

- Öngörülse (predictive) yaşam döngüsü
  - Geleneksel «çağlayan» yaklaşımı
  - Başlangıçta çok yoğun planlama aktiviteleri
  - Sonrasında tek ve sıralı bir süreç takibi
- Tekrarlayan (iterative) yaşam döngüsü
  - Bitmeden işi iyileştirmek için geribildirim alarak değiştirmeye dayalı yaklaşım
- Artırmalı (incremental) yaşam döngüsü
  - Müşterinin hemen kullanabileceği tamamlanmış teslimatlar sunan yaklaşım
- Çevik (agile) yaşam döngüsü
  - Hem tekrarlayan hem artırmalı
  - İş ürünlerini iyileştirip sık teslimat sağlayan yaklaşım

## Yaşam döngüsü özellikleri

| Yaklaşım                | Gereksinim | Aktiviteler                                   | Teslimat                 | Amaç  |
|-------------------------|------------|---|--------------------------|---|
| Öngörülse (predictive)  | Sabit      | Tüm proje için tek sefer gerçekleştirilir     | Tek teslimat             | Maliyet yönetimi                                    |
| Tekrarlayan (iterative) | Dinamik    | Doğruya ulaşına dek tekrar edilir             | Tek teslimat             | Çözümün doğruluğu                                   |
| Artırmalı (incremental) | Dinamik    | Her bir artım için tek sefer gerçekleştirilir | Sık ve küçük teslimatlar | Hız   |
| Çevik (agile)           | Dinamik    | Doğruya ulaşına dek tekrar edilir             | Sık ve küçük teslimatlar | Sık teslimat ve geri bildirimler ile müşteri değeri |

## Yaşam döngüsü özellikleri

| Yaklaşım                | Gereksinim | Aktiviteler                                   | Teslimat                 | Amaç  |
|-------------------------|------------|---|--------------------------|---|
| Öngörüsöl (predictive)  | Sabit      | Tüm proje için tek sefer gerçekleştirilir     | Tek teslimat             | Maliyet yönetimi                                    |
| Tekrarlayan (iterative) | Dinamik    | Doğruya ulaşana dek tekrar edilir             | Tek teslimat             | Çözümün doğruluğu                                   |
| Artırmalı (incremental) | Dinamik    | Her bir artım için tek sefer gerçekleştirilir | Sık ve küçük teslimatlar | Hız   |
| Çevik (agile)           | Dinamik    | Doğruya ulaşana dek tekrar edilir             | Sık ve küçük teslimatlar | Sık teslimat ve geri bildirimler ile müşteri değeri |

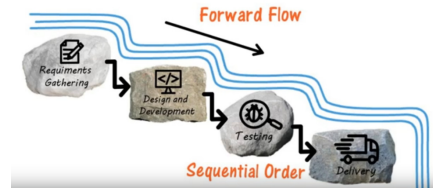
## WATERFALL

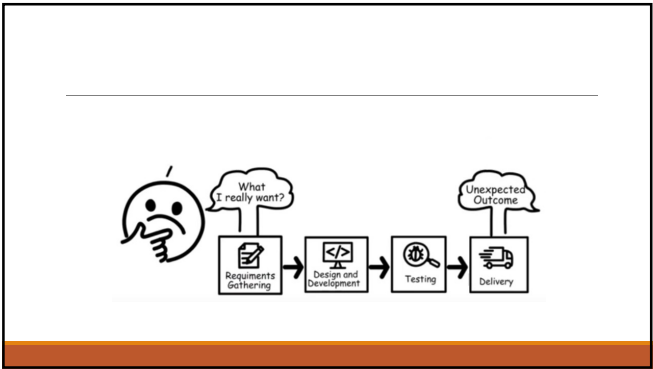
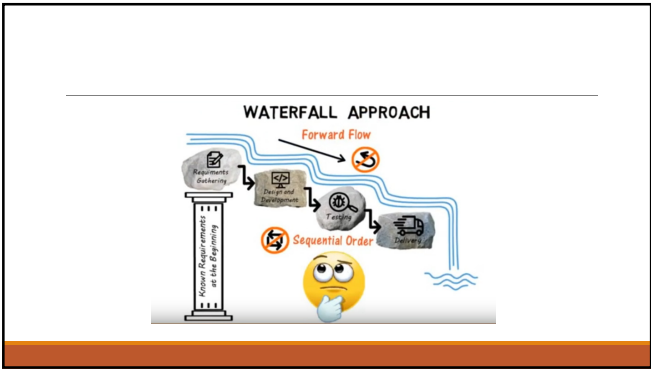
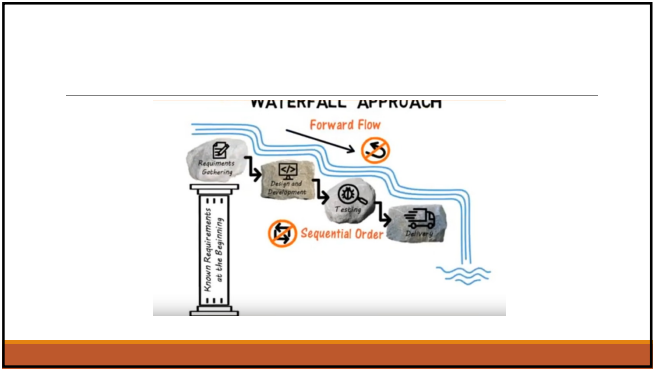
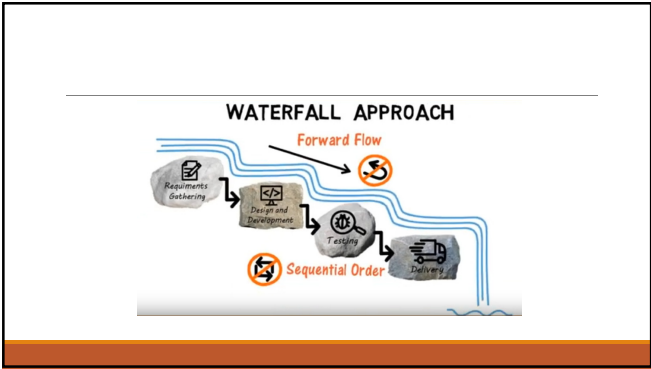


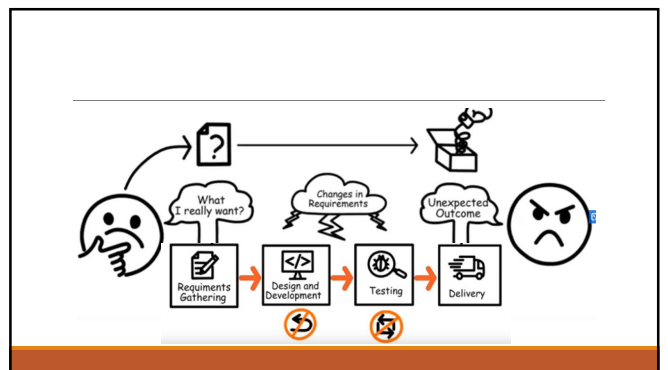
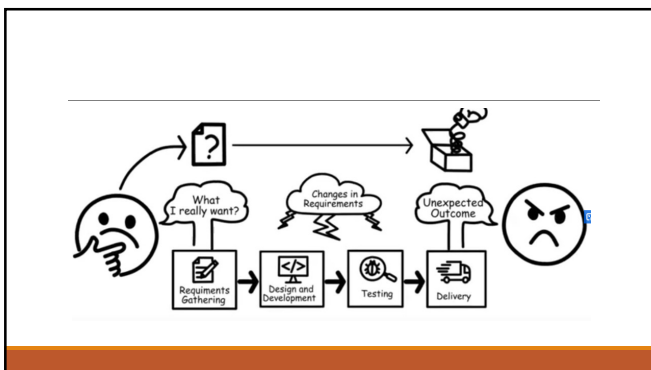
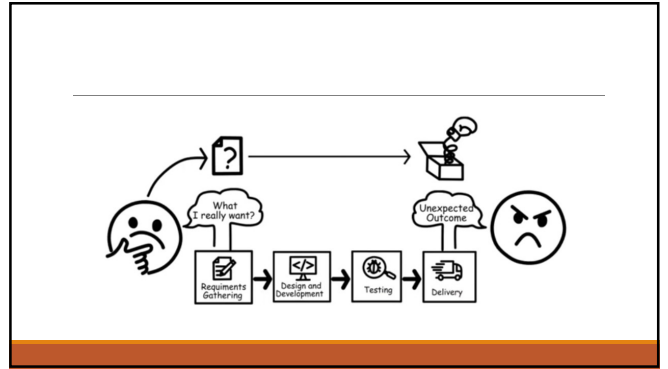
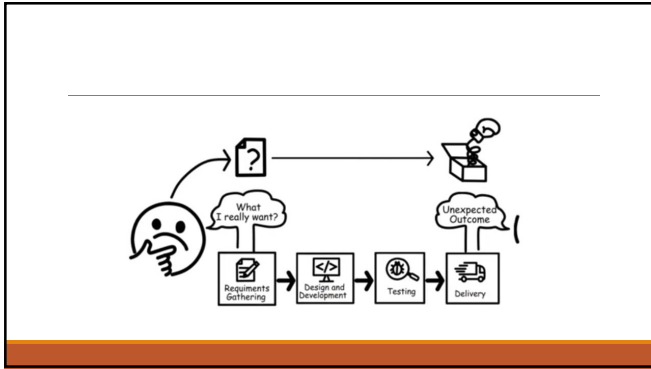
## WATERFALL APPROACH



## WATERFALL APPROACH





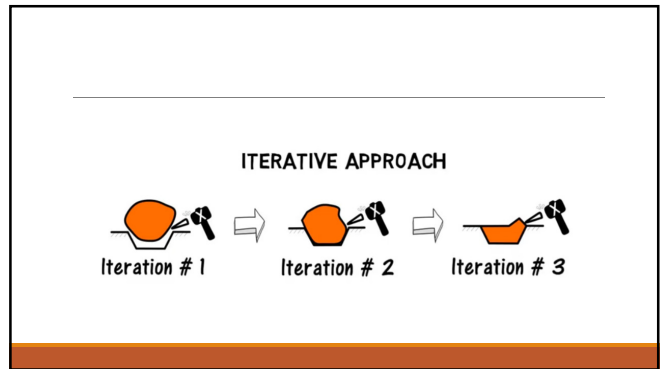
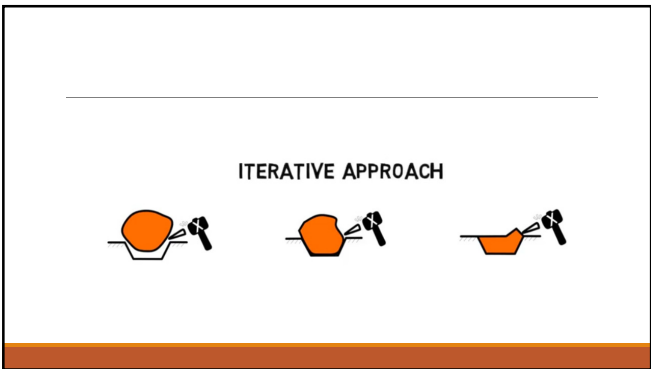
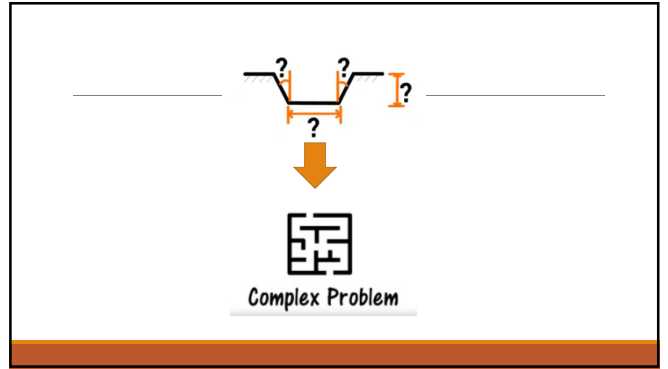




## Öngörüşel (predictive) yaşam döngüsü

- Geleneksel «çağlayan» yaklaşımları
  - Gereksinimler
  - Sabit takım
  - Düşük risk
- } gibi konularda kesinlik
- Bunun sonucunda proje aktiviteleri doğrudan seri bir şekilde uygulanır
  - Takım detaylı bir plan doğrultusunda ilerler, kapsam, zaman ve bütçeyi etkileyebilecek değişiklikleri izler ve kontrol eder
  - Projeler, projenin sonunda kadar iş değeri ortaya koyamaz





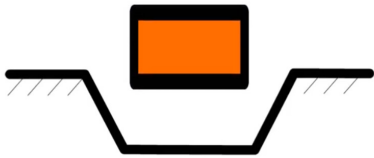
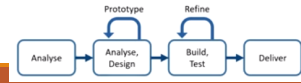
## Yaşam döngüsü özellikleri

| Yaklaşım                       | Gereksinim     | Aktiviteler                                   | Teslimat                 | Amaç  |
|--------------------------------|----------------|---|--------------------------|---|
| Öngörüsöl (predictive)         | Sabit          | Tüm proje için tek sefer gerçekleştirilir     | Tek teslimat             | Maliyet yönetimi                                    |
| <b>Tekrarlayan (iterative)</b> | <b>Dinamik</b> | <b>Doğruya ulaşana dek tekrar edilir</b>      | <b>Tek teslimat</b>      | <b>Çözümün doğruluğu</b>                            |
| Artırmalı (incremental)        | Dinamik        | Her bir artım için tek sefer gerçekleştirilir | Sık ve küçük teslimatlar | Hız   |
| Çevik (agile)                  | Dinamik        | Doğruya ulaşana dek tekrar edilir             | Sık ve küçük teslimatlar | Sık teslimat ve geri bildirimler ile müşteri değeri |

## Tekrarlayan (iterative) yaşam döngüsü

### Birbirini izleyen prototipler

- **Kavram kanıtları (proof of concepts)** ile ürün ya da sonucu iyileştirir
- Her yeni prototip yeni bir paydaş geribildirimi ya da takım bilgilerine yol açar
- Takımlar bir tekrarlama için **zaman-sınırlı (time-boxed) 2-4 haftalık süreler** kullanır
- Prototip yapar, geri bildirim alır ve bunu bir sonraki tekrarlama kullanır
- Özellikle **karmaşıklık çoksa** ve **kapsam değişime açık** ise kullanışlıdır
- Tekrarlayan yaşam döngüleri **hızlı teslimattan ziyade öğrenmeye optimize** edildiklerinden daha uzun sürer



## INCREMENTAL APPROACH

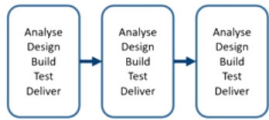


## Yaşam döngüsü özellikleri

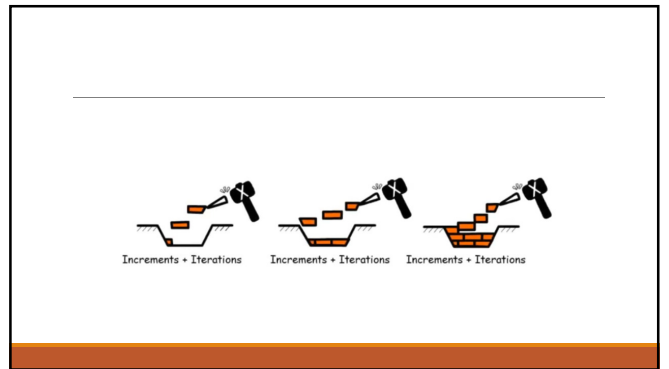
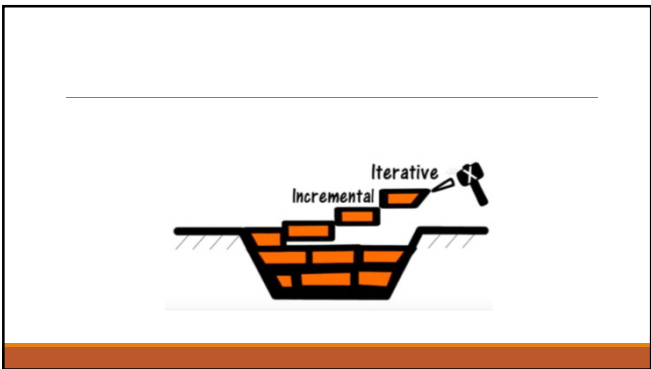
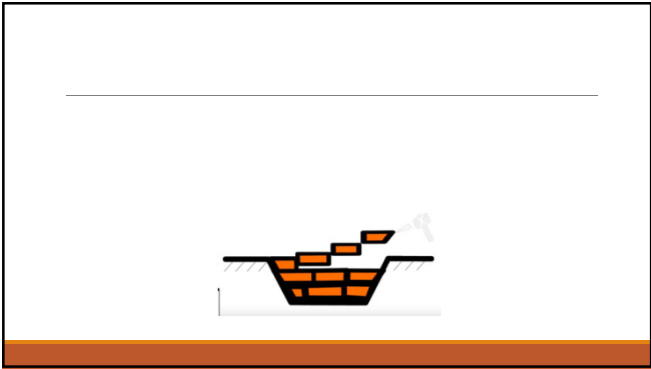
| Yaklaşım                | Gereksinim | Aktiviteler                                     | Teslimat                 | Amaç  |
|-------------------------|------------|---|--------------------------|---|
| Öngörülse (predictive)  | Sabit      | Tüm proje için tek sefer gerçekleştirilir       | Tek teslimat             | Maliyet yönetimi                                    |
| Tekrarlayan (iterative) | Dinamik    | Doğruya ulaşana dek tekrar edilir               | Tek teslimat             | Çözümün doğruluğu                                   |
| Artırmalı (incremental) | Dinamik    | Her bir artırım için tek sefer gerçekleştirilir | Sık ve küçük teslimatlar | Hız   |
| Çevik (agile)           | Dinamik    | Doğruya ulaşana dek tekrar edilir               | Sık ve küçük teslimatlar | Sık teslimat ve geri bildirimler ile müşteri değeri |

## Artırmalı (incremental) yaşam döngüsü

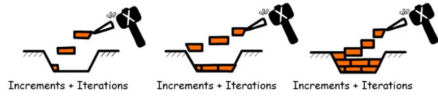
- Her şeyin **tamamlanmasını beklemeyi karşılayamayacak** projeler
  - Çözümün tamamı yerine **küçük bir kümesine** karşılık gelen **sık ve küçük sürümler**
- Her bir seferinde tek bir özellik sunma ya da ekleme
- Bu yöntemde de artırımlar yine **zaman sınırlıdır (time-boxed)**







## AGILE APPROACH



## Yaşam döngüsü özellikleri

| Yaklaşım                | Gereksinim     | Aktiviteler                                   | Teslimat                        | Amaç   |
|-------------------------|----------------|---|---------------------------------|--|
| Öngörüsöl (predictive)  | Sabit          | Tüm proje için tek sefer gerçekleştirilir     | Tek teslimat                    | Maliyet yönetimi   |
| Tekrarlayan (iterative) | Dinamik        | <b>Doğruya ulaşana dek tekrar edilir</b>      | Tek teslimat                    | Çözümün doğruluğu  |
| Artırmalı (incremental) | Dinamik        | Her bir artım için tek sefer gerçekleştirilir | <b>Sık ve küçük teslimatlar</b> | Hız  |
| <b>Çevik (agile)</b>    | <b>Dinamik</b> | <b>Doğruya ulaşana dek tekrar edilir</b>      | <b>Sık ve küçük teslimatlar</b> | <b>Sık teslimat ve geri bildirimler ile müşteri değeri</b> |

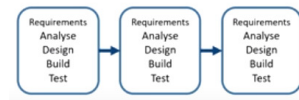
## Çevik (agile) yaşam döngüsü

- Takım gereksinimlerin değişmesini bekler
- Tekrarlanan ve artırmalı yaklaşımlar beraber kullanılır -
  - Bir sonraki tekrarlamayı planlamak için **geri bildirim** sağlanır (Yine 2-4 haftalık tekrarlamalar)
  - Artımlar kullanarak da **gizli ve yanlış anlaşılan gereksinimler** ortaya çıkarılır
- Kanban ya da akış tabanlı çevik yöntemlerde ilerleyen iş (Work In Progress) ya da tahtadaki özellik sayısı tamamlanana kadar tekrarlama kullanılır
  - Takım tahtadaki backlog (ürün iş listesi) sütunundan kendi kapasitesine göre bir özellik seçer



## Çevik (agile) yaşam döngüsü

- Erken ve sürekli **değer teslimi** ile **müşteri memnuniyeti** artar
- İlerlemenin ana ölçümü değer yaratan artımlı ve **fonksiyonel teslimatlar**dır

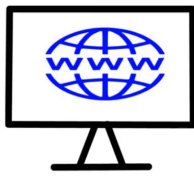
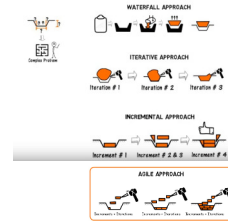


## Hibrit yaşam döngüsü

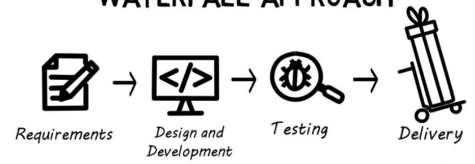
➤ Öngörülebilir, tekrarlanan, artımlı ya da çevik yaklaşımları bir arada kullanan yaklaşımlardır.

### ➤ Örnek

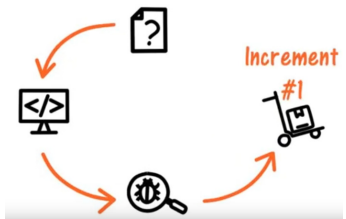
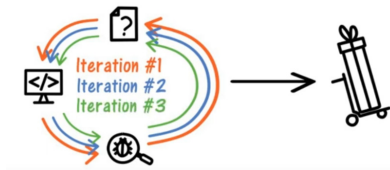
- Bir miktar öngörülebilir yoğunlukla çevik
  - Farklı bir tedarikçi tarafından geliştirilen bir bileşeni entegre etme durumunda → teslim edilen bileşen tek bir tekrarlama ile gelebilir
- Bir miktar çevik yoğunlukla öngörülebilir
  - Basit bir proje geliştirirken yeni bir bileşenin olması durumu
- Birleşik öngörülebilir ve çevik yaklaşım
  - En sonunda bir proje ürünü teslim edilecek olsa bile görevlerin Kanban ile takip edildiği veya güncellemeler için günlük Scrum toplantılarının kullanıldığı lineer bir proje



## WATERFALL APPROACH



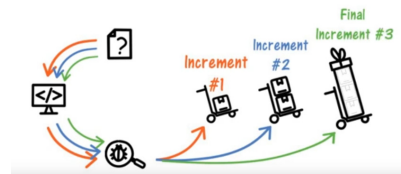
### ITERATIVE APPROACH

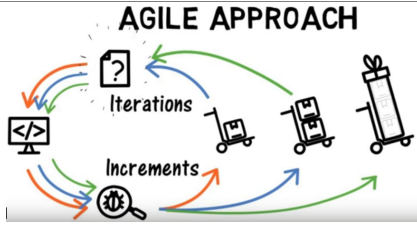


### INCREMENTAL APPROACH



### INCREMENTAL APPROACH





## Çevik Manifesto ve düşünce yapısı

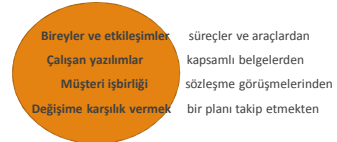
### Agile Manifesto

2001 yılında bir grup bağımsız yazılım geliştirici hantal olmayan yazılım geliştirme metodolojilerini bir araya getirerek günümüzde Agile Manifesto olarak bilinen ortak değer ve prensipler kümesini ortaya koymuşlardır

<https://www.agilealliance.org/>

### Agile Manifesto

Yazılımı geliştirerek ve başkalarının bunu yapmasına yardımcı olarak yazılım geliştirmenin daha iyi yollarını keşfediyoruz. Bu çalışma sayesinde:



değerlidir.

Yani, soldaki öğeler sağdaki öğelerden daha fazla değer verdiğimiz konulardır.

## 12 Çevik Prensip

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
1. En yüksek önceliğimiz, değerli yazılımların **erken ve sürekli teslimi** ile müşteriyi memnun etmektir.
2. Geliştirmenin geç aşamasının bile değişen gereksinimlere hoş karşılanır. Çevik süreçler, müşterinin rekabetçi avantajı için **değişimi** kullanır.
3. Çalışan yazılımı sıklıkla ve tercihen birkaç haftadan birkaç aya kadarlık **kısa zaman aralıklarında** teslim edin.
4. İş insanları ve geliştiriciler proje boyunca **günlük olarak birlikte çalışmalıdır**.

## 12 Çevik Prensip

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
5. Projeleri motive olmuş bireyler etrafında oluşturun. Onlara ihtiyaç duydukları ortamı ve desteği verin ve işi yapmaları için **onlara güvenin**.
6. Bir geliştirme ekibine ve ekibin içinde bilgi aktarmanın en etkili ve etkili yöntemi **yüz yüze görüşmedir**.
7. **Çalışan yazılım** ilerlemenin birincil ölçüsüdür.
8. Çevik süreçler sürdürülebilir geliştirmeyi teşvik eder. Sponsorlar, geliştiriciler ve kullanıcılar süresiz olarak **sabit bir hızı** korumalıdır.

## 12 Çevik Prensip

9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
9. Teknik mükemmelliğe ve iyi tasarıma **sürekli dikkat**, çevikliği artırır.
10. **Basitlik** - yapılmayan iş miktarını en üst düzeye çıkarma sanatı - esastır.
11. En iyi mimariler, gereksinimler ve tasarımlar, **kendi kendini düzenleyen ekiplerden** ortaya çıkar.
12. Düzenli aralıklarla ekip, nasıl daha etkili olabileceğini **düşünür**, ardından davranışını buna göre ayarlar ve uyarlar.