



# YMÜ228 Yazılım Tasarım ve Mimarisi

## Çevik Yazılım Geliştirme

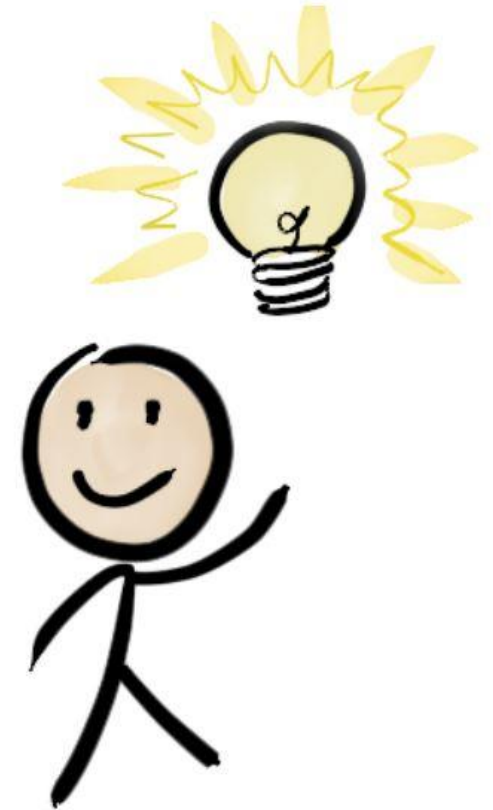
Doç. Dr. Fatih ÖZYURT  
Fırat Üniversitesi Yazılım Mühendisliği Bölümü

Bölüm-2

# 1.Günümüzde Yazılım Projelerinin Durumu

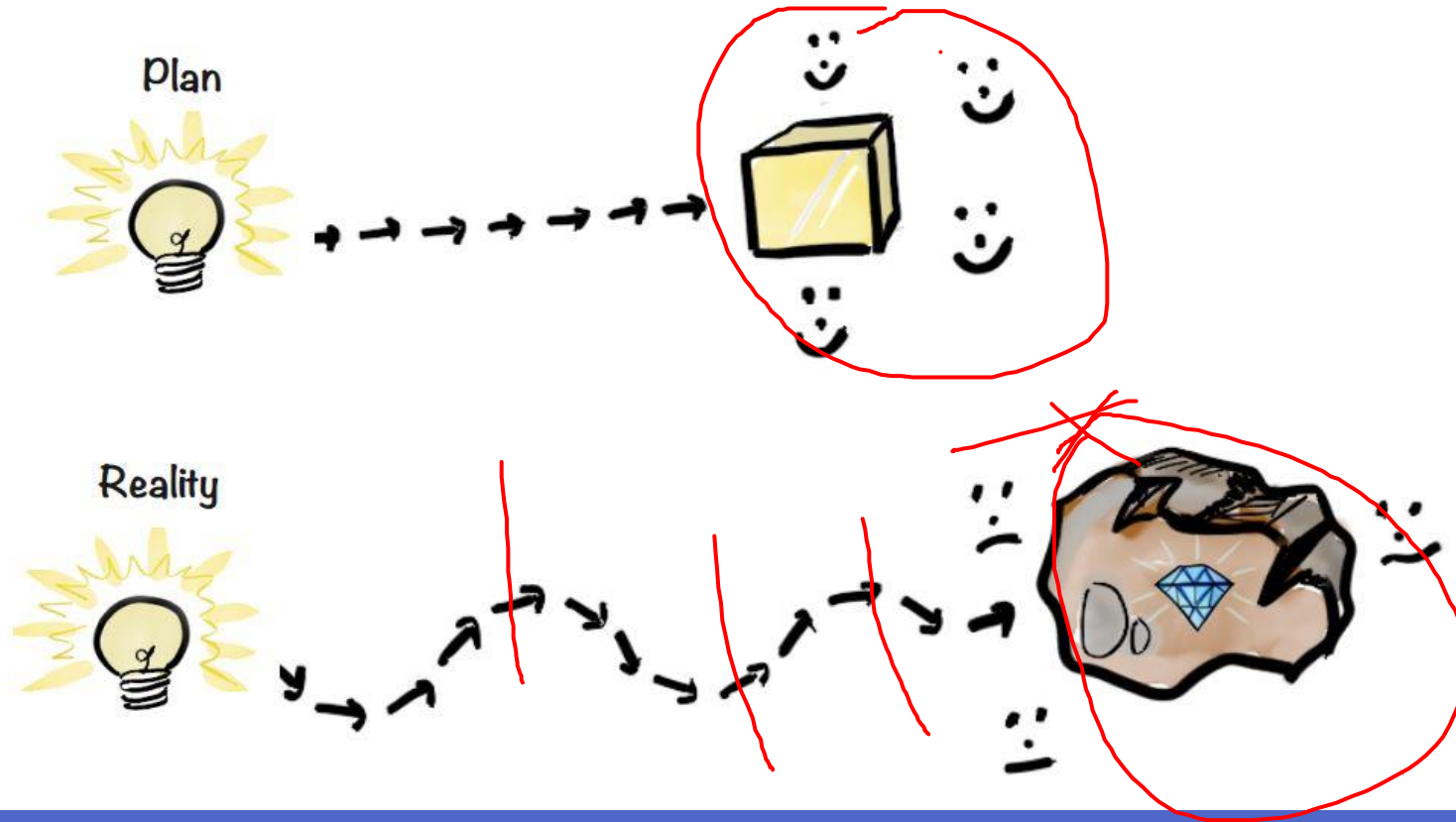
---

- Birçok proje harika bir fikir ile başlar!



# 1.Günümüzde Yazılım Projelerinin Durumu

- Bu projelerin büyük bir kısmının başarısız olması muhtemeldir!

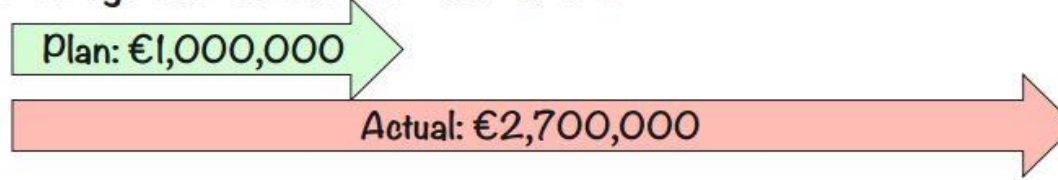


# 1. Günümüzde Yazılım Projelerinin Durumu

- Birçok Bilgi Teknolojisi projesi başarısız olmuş veya gecikmiştir. The Standish Group, 10 yıl içerisinde 40.000'den fazla proje üzerinde çalışmıştır.

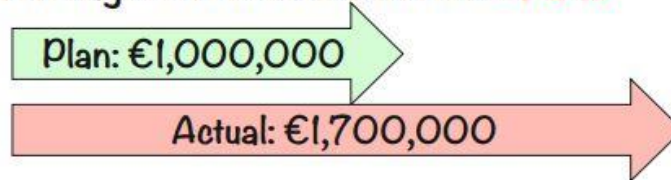
IT project success rate 1994: **15%**

Average cost & time overrun: **~170%**



IT project success rate 2004: **34%**

Average cost & time overrun: **~70%**



# 1.Günümüzde Yazılım Projelerinin Durumu

---

Ülkemizde durum nasıl?

Durum	Oran
Tam başarılı	%4-5
Kısmen başarılı	%45-50
Çöpe gidenler	%50

Agile Turkey

# 1.Günümüzde Yazılım Projelerinin Durumu

---

- Başarısızlığın ana sebepleri:
- Müşterinin isteklerini doğru analiz edememek;
- Proje için uygun ekibi kuramamak;
- Yanlış teknoloji ve mimari seçimleri;
- Geleneksel yöntemlerin eksiklikleri;
- Müşteriyle iletişimden kaçınmak vs.



# 1.Günümüzde Yazılım Projelerinin Durumu

# Peki ne yapmalıyız?



## 2. Çevik Yazılım Yöntemi



- **Tekrarlamalı** ve **artımsal** bir ürün geliştirme yöntemidir.
- **Bireyler** ve **etkileşimi**, süreç ve araca tercih eder.
- **Çalışan bir yazılımı**, detaylı ürün belgelendirmeye tercih eder.
- **Müşteri ile işbirliğini**, sözleşmedeki kesin kurallara tercih eder.
- **Değişikliklere uyum sağlayabilmeyi**, belirli bir plana tercih eder.



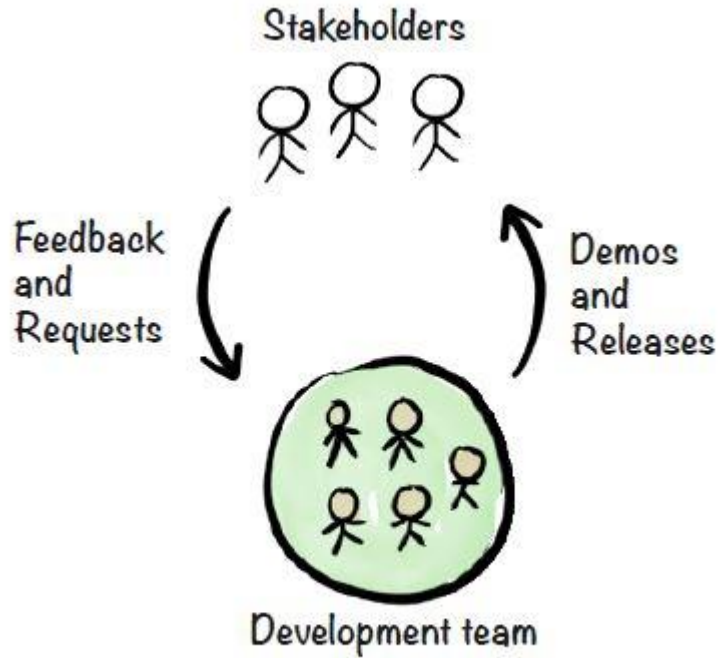


## 2. Çevik Yazılım Yöntemi

- Hızlı, devamlı ve kullanışlı yazılım üreterek müşteri memnuniyeti sağlamayı amaçlar.
- Geliştiriciler ile iş adamları arasında günlük ve yakın işbirliği bulunmalıdır.
- Çalışan yazılım gelişimin en önemli ölçüsüdür.
- Taleplerdeki geç değişikliklerin de memnuniyetle karşılanır.
- Yüz yüze görüşme iletişimin en güzel yoludur.
- Kendi kendini organize eden takım yapısı gereklidir.
- Basitlik önemlidir.

## 2. Çevik Yazılım Yöntemi

---



- Çevik yazılım metodu, kısa vadeli planlar ve küçük parçalar halinde yazılımın geliştirilmesini ön görür. Yazılımın geliştirilmesindeki geri dönüş (**feedback**) ve değişikliklere uyum sağlamak son derece önemlidir. Her yapılan yineleme yazılımı hedeflenen adıma bir adım daha yakınlaştırır. İstenilen sonuca ulaşmak adına birden çok yineleme gereklidir.

## 2. Çevik Yazılım Yöntemi

---

- Örnek:

2007



2008

- App Store
- 3G



2009

- Copy/Paste
- Search



2010

- Multitasking
- Video calls

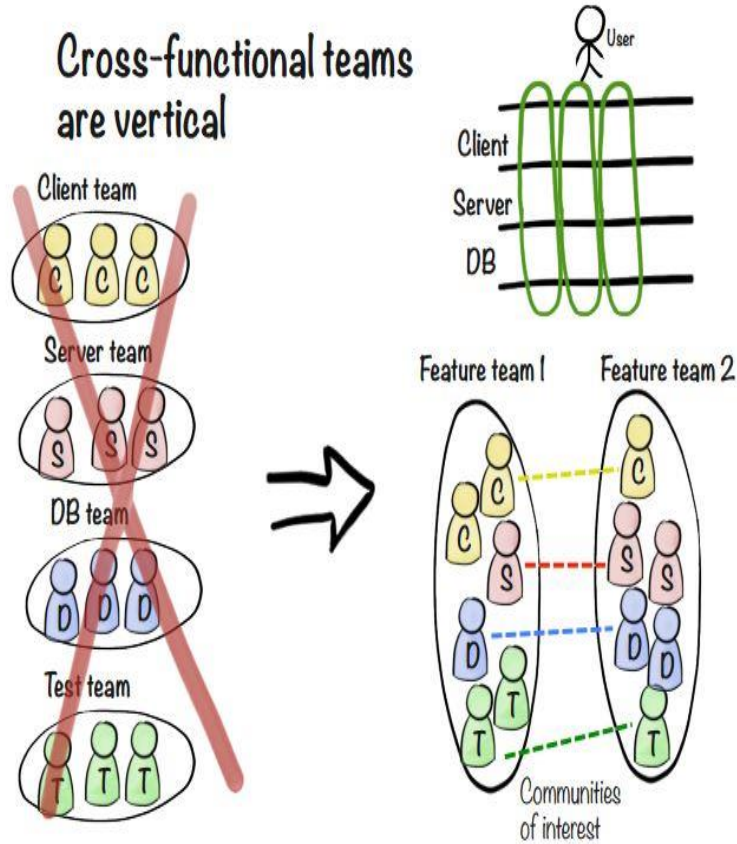


## 2. Çevik Yazılım Yöntemi

---

- Temel prensipler:
- Müşteriyi memnun etmek
- Değişen ihtiyaçları karşılamak
- Sık aralıklarla ürün teslimi yapmak
- Yüz yüze iletişime önem vermek
- Sürdürülebilir gelişmeyi desteklemek
- Teknik mükemmeliyete, iyi dizayna ve sadeliğe odaklanmak
- **Kendi kendine organize olan takımlar kurmak ?**

## 2.1. Çevik Model Takımları



- Bir araya gelmiş,
- Kendi kendilerine organize olan,
- Çapraz fonksiyonlu,
- İşine odaklanmış,
- Hedefleri net olan,
- Teslim edilebilecek düzeyde ürün ortaya koyabilen
- Küçük (3-7 kişilik) gruplar.

## 2.1. Çevik Model Takımları

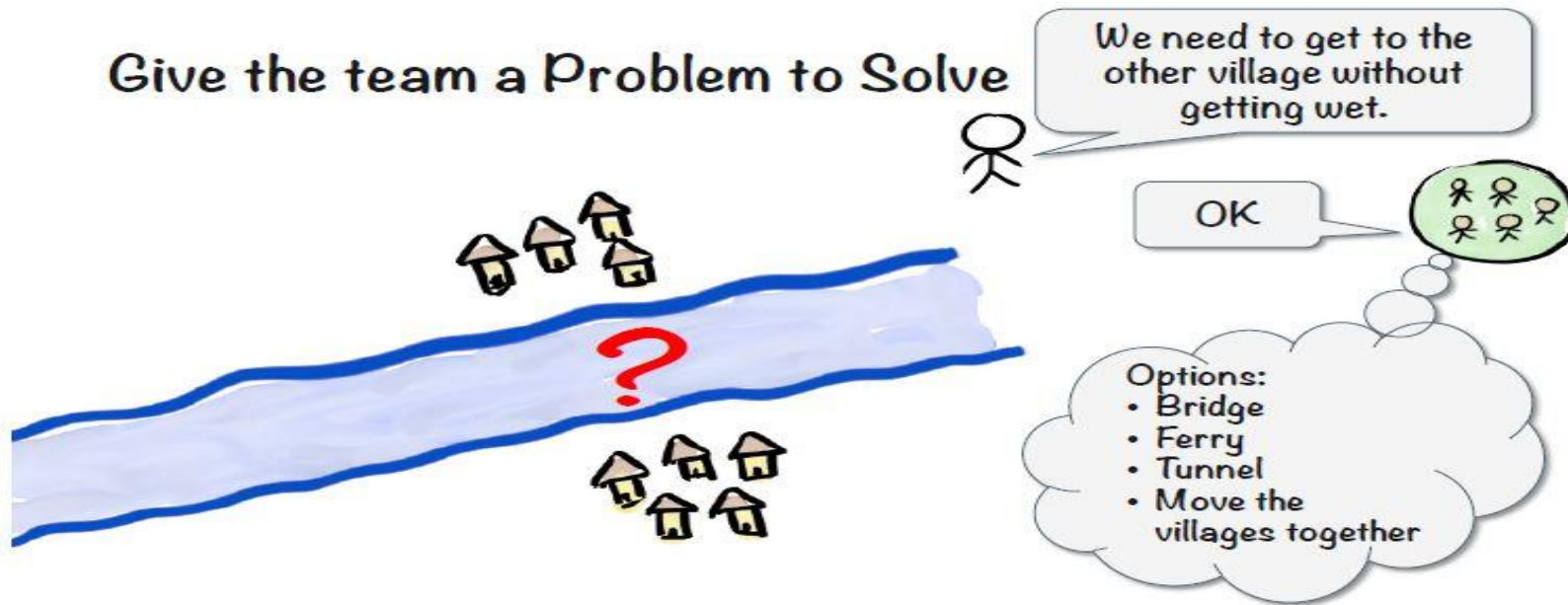
Takımlara çözümü söylemeyin!

Don't give the team a Solution to Build



## 2.1. Çevik Model Takımları

Sorunu söyleyin, onlar çözümü üretsin!





# 3. Yazılım Projelerinin Temel Adımları

---

## Yazılım Projelerinin Temel Adımları ve Metodoloji İlişkisi

Yazılım projelerinin temel adımları aşağıdaki gibidir:

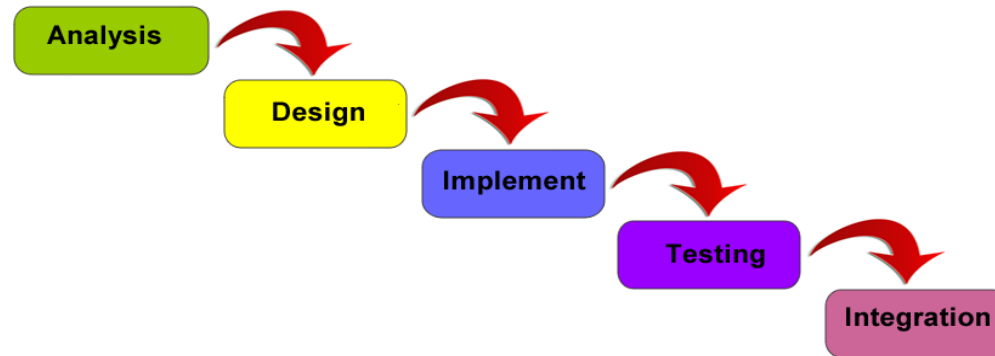
- 1. Planlama:** Yazılım yaşam döngüsünün başlangıç evresidir. Projeyi sipariş eden veya projenin hedeflediği kullanıcıların istekleri ve ihtiyaçları dikkate alınarak proje ihtiyaçları tespit edilir. Proje boyunca takip edilecek olan yol, projede kullanılacak olan donanımların maliyeti, proje sürecinde personeller arasında iş bölümünün nasıl olacağı ve proje adımlarının zamana nasıl yayılacağı gibi detaylar değerlendirilir.
- 2. Analiz:** Geliştirilecek olan üründen tam olarak ne istendiği net bir şekilde kağıda dökülür. Proje somutlaşmaya başlamıştır.
- 3. Tasarım:** Yazılım sisteminin temel yapısı bu aşamada oluşturulur. Yazılım ürününün mantıksal ve fiziksel tasarımı üzerine çalışmalar yapılır.
- 4. Geliştirme:** Temelleri oluşturulmuş olan ürünün tam olarak kodlanması, kurulması ve test edilmesi bu aşamada gerçekleştirilir.
- 5. Teslim ve Bakım:** Test aşamalarının tamamlanması; ürünün hazır olduğu ve müşteriye teslim edilebileceği anlamına gelir. Müşterinin yazılımı aktif olarak kullanması ve kullanım sonucu ihtiyaç duyulan bakımların yapılması bu aşamada gerçekleşir.



### 3. Geleneksel Model vs. Agile

---

- Çağlayan modeli 2008 yılında dahi geçerliliğini koruyan bir modeldir ve çevik modellemeden farklılık gösterir. Bu model yazılım projesini baştan sona planlar. Gelişim, sunulabilir işler açısından ölçülür: talep açıklamaları, tasarım dokümanları, test planları, kod incelemeleri vb. Bu durum belli aralıklara bölünmeye uygun değildir ve ilerideki değişikliklere uyum gösterilemez.



Waterfall Model

# 3. Geleneksel Model vs. Agile

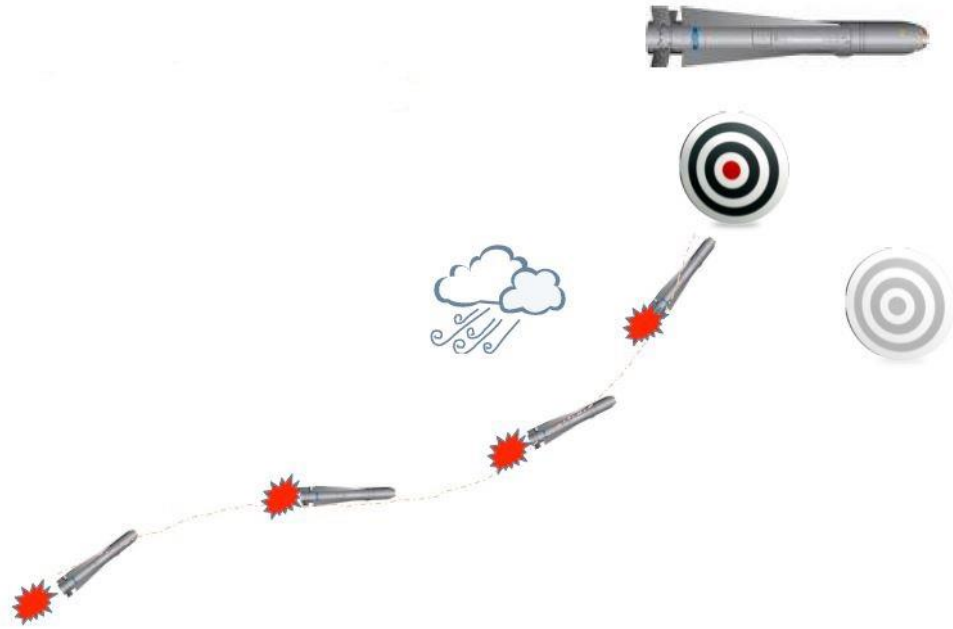
---



- Geleneksel Yöntemler
- Müşteriler ne istediğini iyi bilir.
- Geliştiriciler neyi, ne şekilde üreteceklerini iyi bilir.
- Bu yol boyunca hiçbir şey değişmeyecektir.

# 3. Geleneksel Model vs. Agile

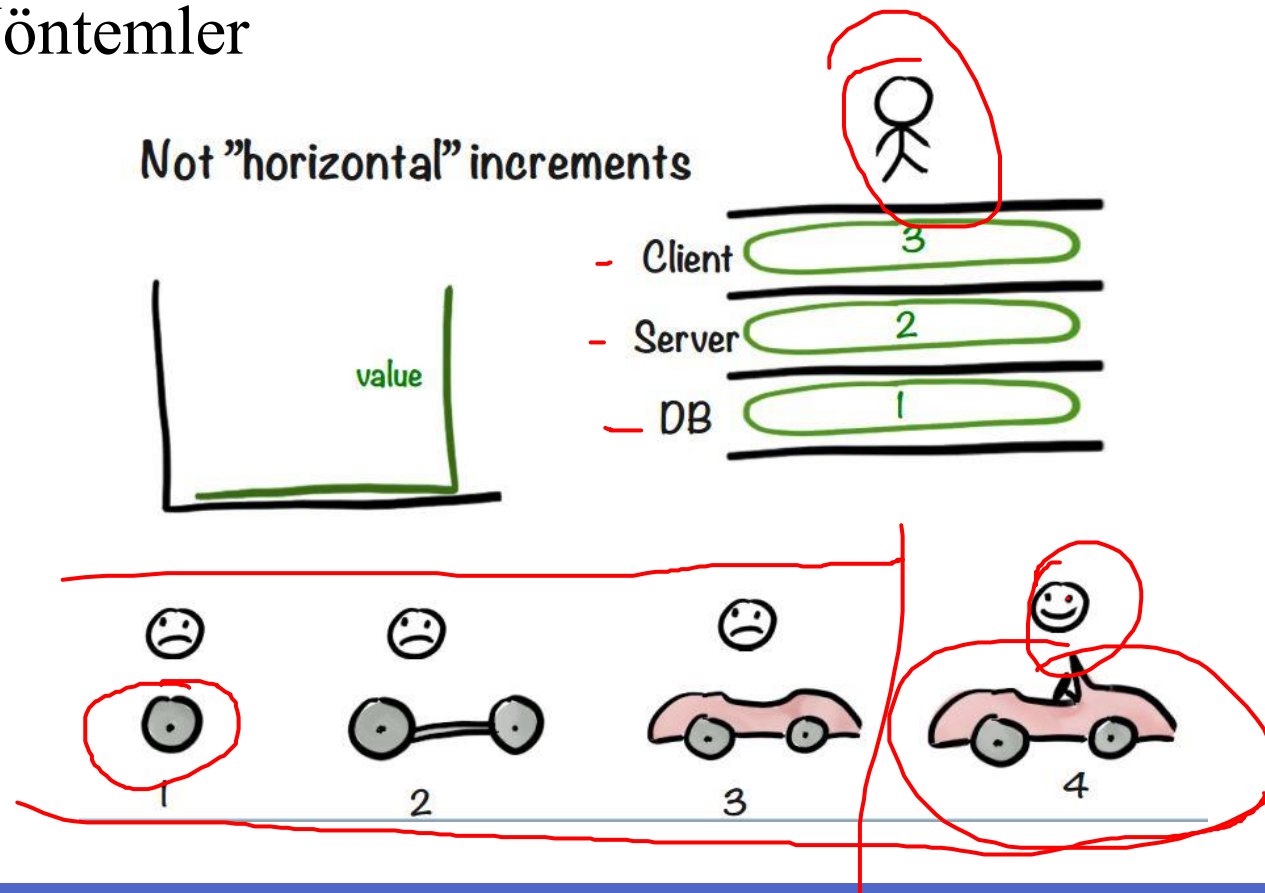
---



- Çevik Yöntemler
- Müşteriler ne istediğini keşfeder.
- Geliştiriciler neyi nasıl üreteceğini keşfeder.
- Bu yol boyunca bir çok değişiklik yapılabilir.

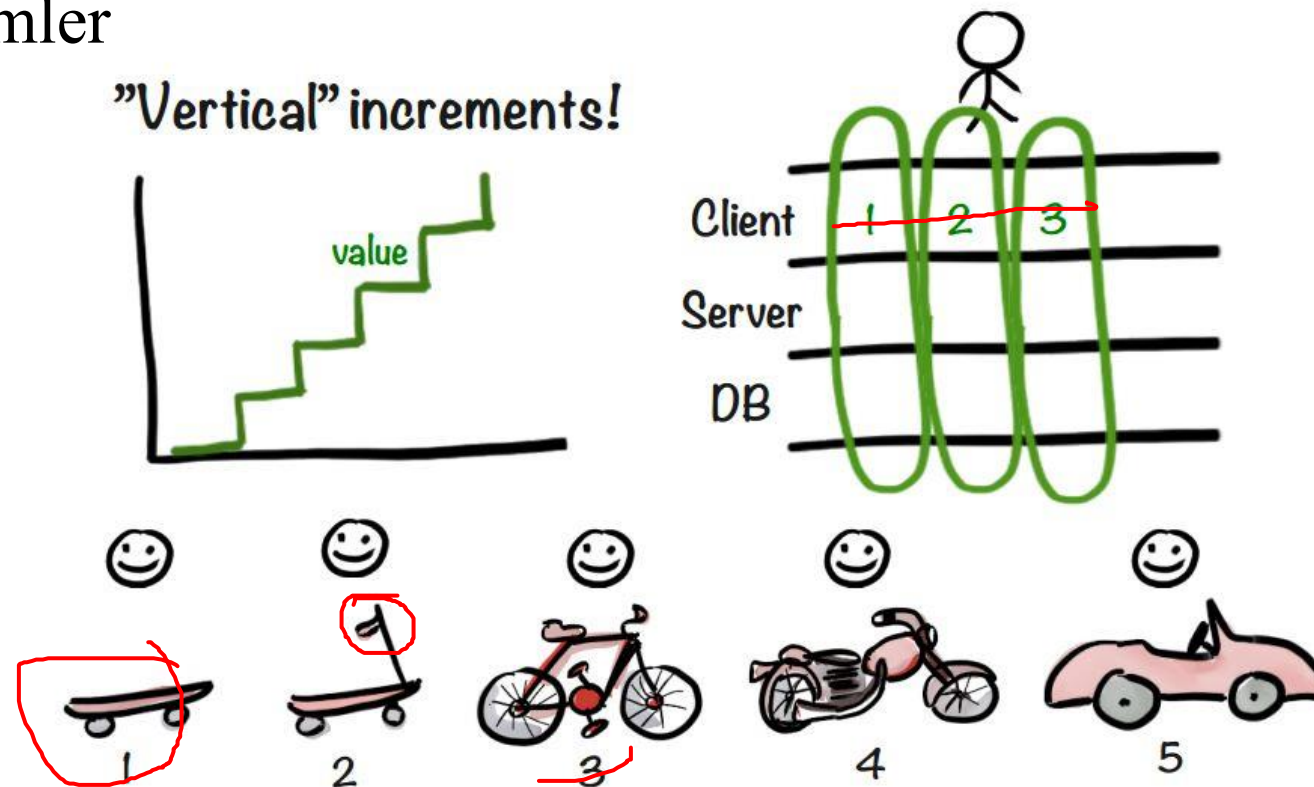
# 3. Geleneksel Model vs. Agile

## Geleneksel Yöntemler

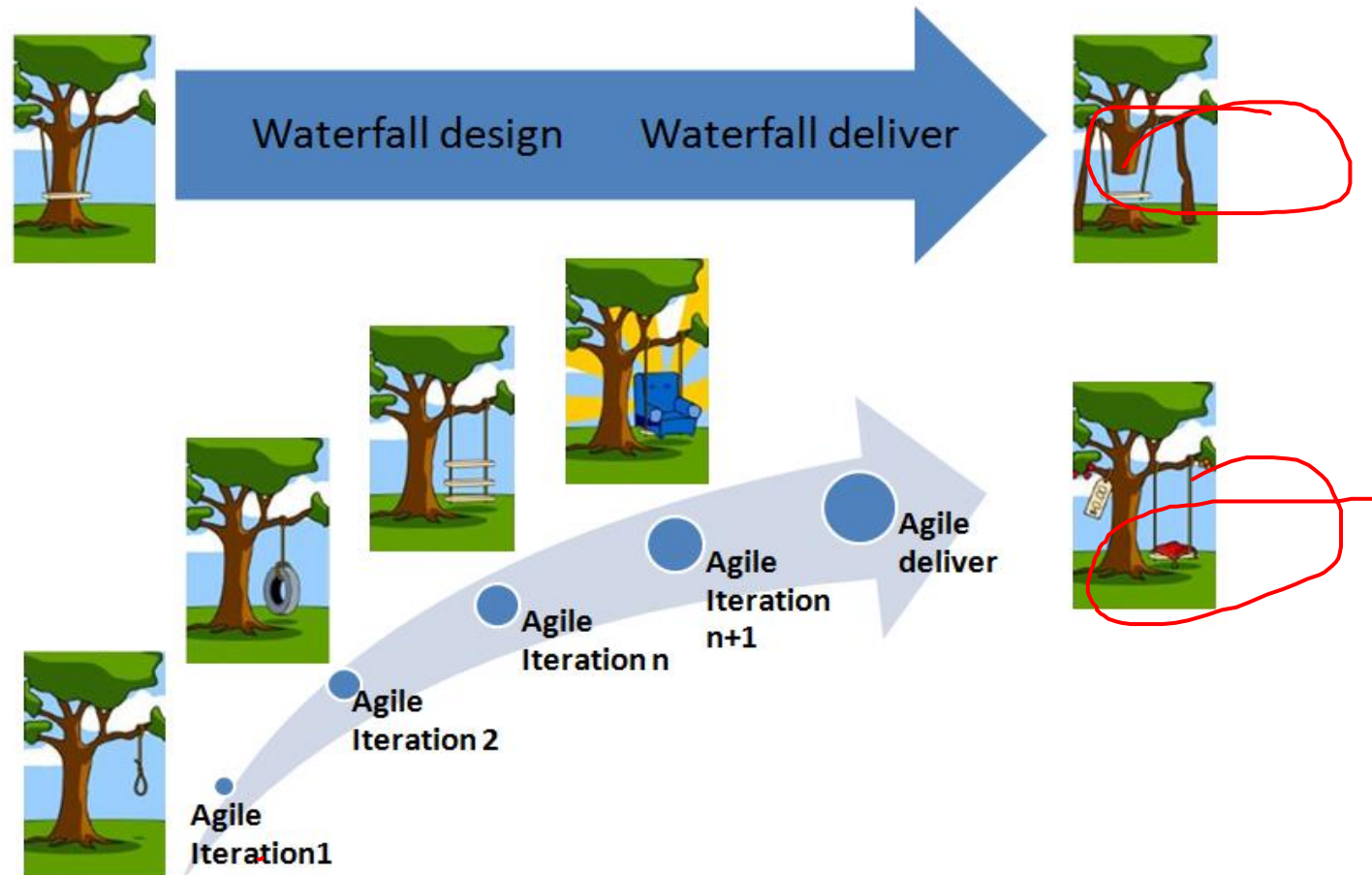


# 3. Geleneksel Model vs. Agile

## Çevik Yöntemler

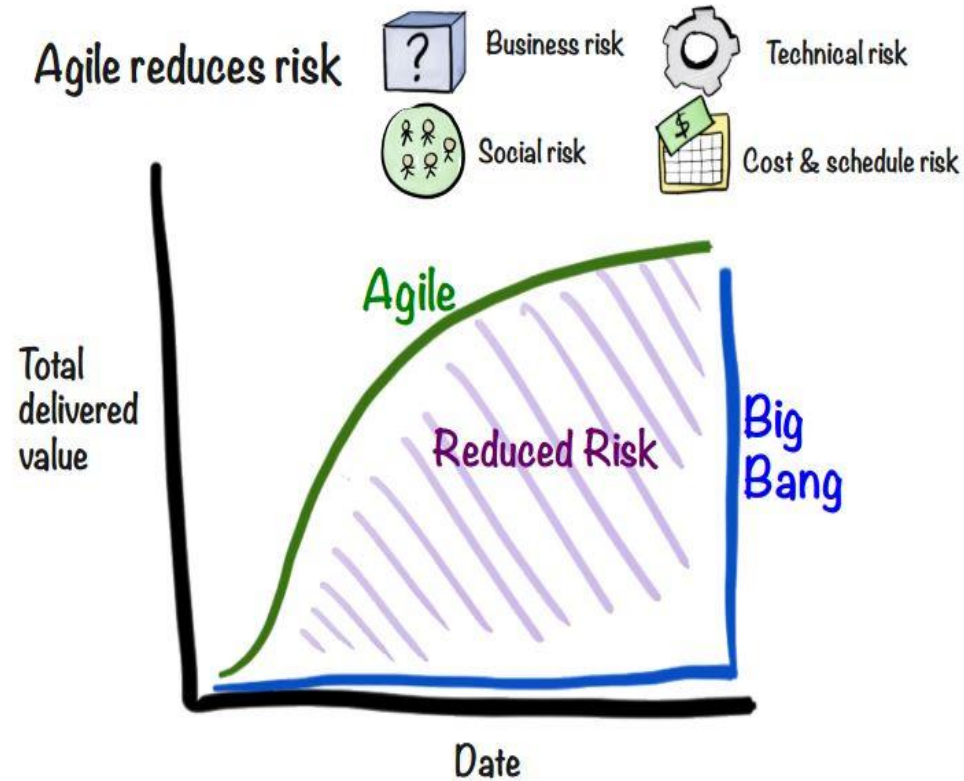


### 3. Geleneksel Model vs. Agile



# 3. Geleneksel Model vs. Agile

Çevik modeller riski azaltır!



# 3. Geleneksel Model vs. Agile

For 2011 to 2015, the overall breakout of success, challenged and failure is shown below for agile and waterfall, with Agile projects being about 3X more likely to succeed.

PROJECT SUCCESS RATES - AGILE VS WATERFALL

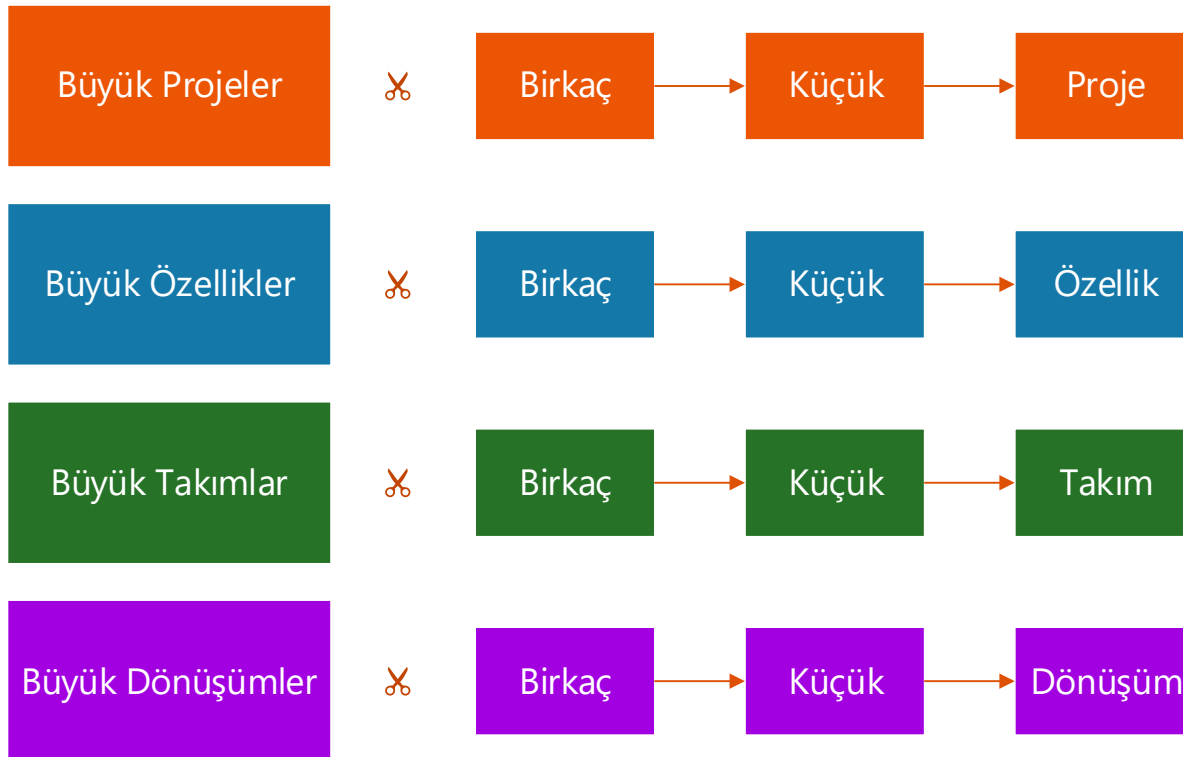




### 3. Geleneksel Model vs. Agile

Ölçüm	Çevik Modelleme	Çağlayan Modeli
Planlama ölçeği	Kısa dönemlik	Uzun dönemlik
Müşteri ile geliştirici arasındaki mesafe	Kısa	Uzun
Özelleştirme ve uygulama arasındaki zaman	Kısa	Uzun
Sorunları keşfetmek için zaman	Kısa	Uzun
Proje tamamlanma riski	Düşük	Yüksek
Değişikliklere uyum yeteneği	Yüksek	Düşük

# 4. Değerlendirme



## Sonuç

- Büyüklük kötüdür, onu parçalara ayırın. Böylece daha başarılı projeler geliştirebilirsiniz.

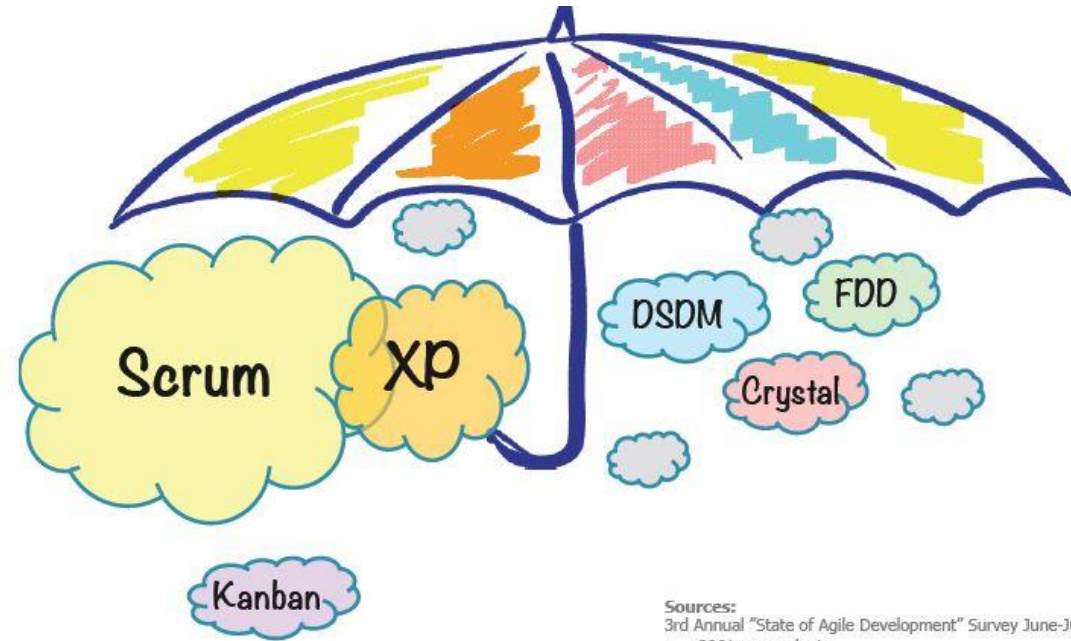
# 4. Değerlendirme

---

- 3 somut değişiklik yapın!
  1. Gerçek takımlar oluşturun
    - Küçük, çapraz fonksiyonlu, kendi kendine organize olabilen
  2. Sık sık teslimat yapın
    - Normal olarak ortalama her 3 haftanın sonunda
    - Ek olarak projenin tüm çeyreklerinin sonunda
  3. Gerçek kullanıcıları dahil edin.
    - Takım ve kullanıcılar arasında doğrudan ve hızlı geri dönüşler

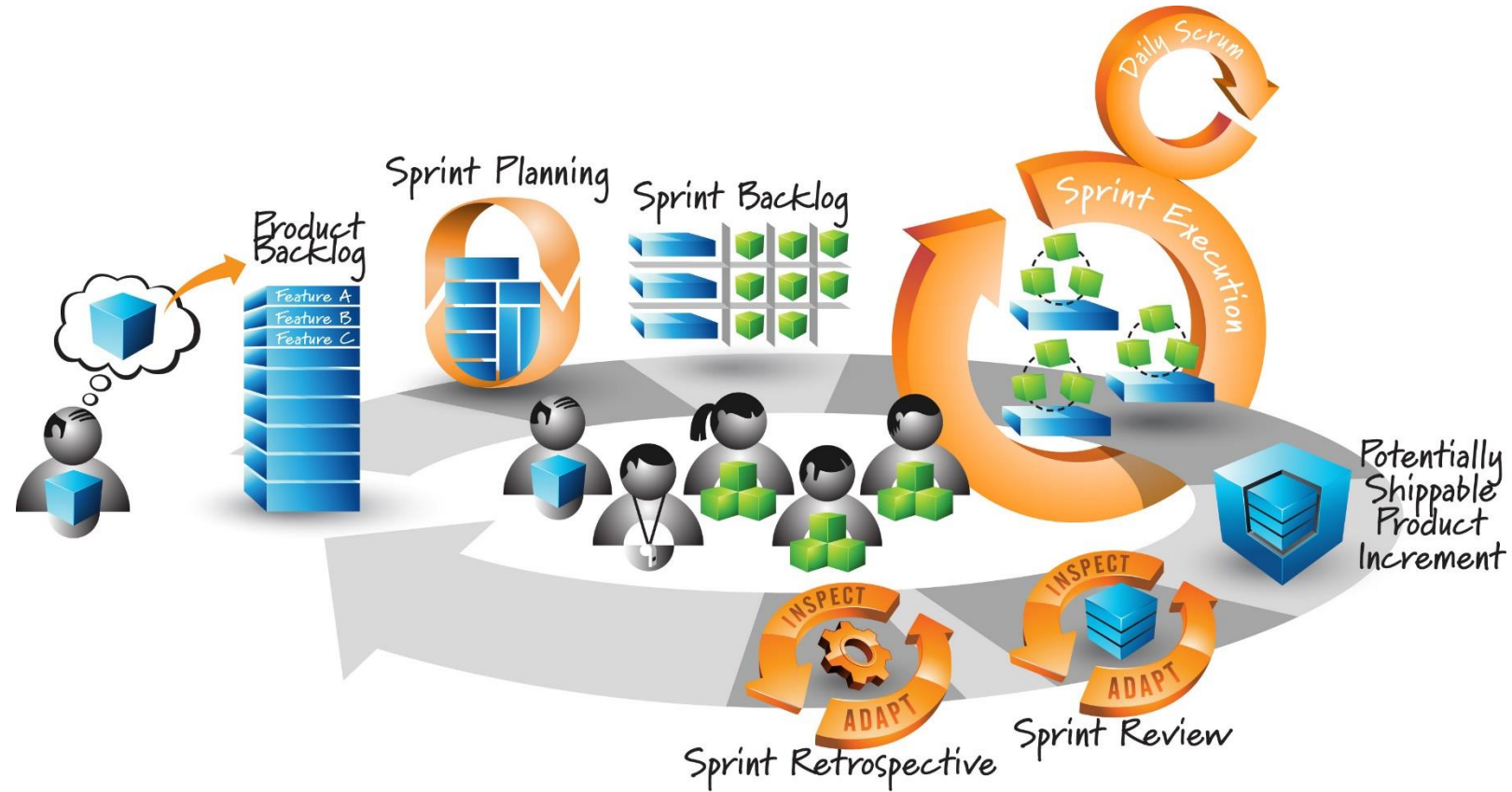
# 5. Çevik Yazılım Şemsiyesi

---



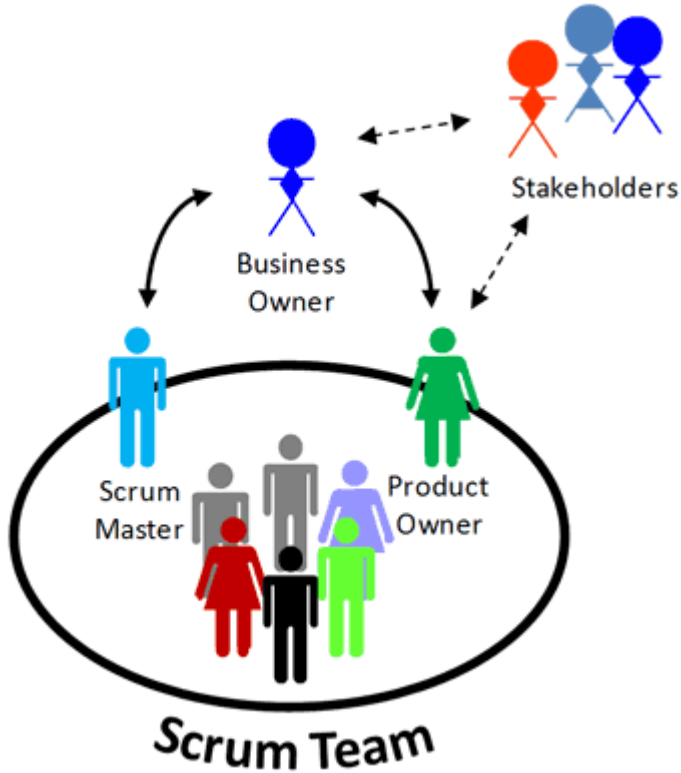
Sources:  
3rd Annual "State of Agile Development" Survey June-July 2008  
• 3061 respondents  
• 80 countries

# 6. Scrum Modeli



Scrum Süreci

# 6. Scrum Modeli



- Scrum Takımı: Ürün Sahibi, Geliştirme Ekibi ve Scrum Master'dan oluşur. Takım kendi kendini örgütler. Böylece kendi içerisinde uyum içinde olan takımlar daha başarılı sonuçlar alırlar. Scrum takım modeli esneklik, yaratıcılık ve verimliliği optimize etmek için tasarlanmıştır.

# Scrum Roller

## Şeffaflık

Sürecin tümleriyle geliştiriciye görünür olmasıdır. Böylece ortak bir görüş ortaya konur. Örneğin, “bitti” takımdakilerin artık işin sonuna geldiğinde hep bir ağızdan söyledikleri bir ortak bir cümledir. Çalışılan sürecin bittiği anlamına gelir.

## Denetim

Scrum, kullanıcılarına yönelik istenmeyen sapmaları tespit etmek için sürecin ve programın sürekli denetim altında tutulması gerekir. Scrum karmaşık ürün gelişimini basite indirmek için yapılandırılmış bir çerçeve olarak düşünebiliriz.

## Uyum

Yazılan programların talebe uygun olup olmaması durumudur. Herhangi bir saptama söz konusu ise bu açıklar giderilir ve talebe uyumlu hale getirilir. Saptamaları aza indirmek kısa bir sürede yapılmalıdır.

## Scrum Takımı

Ürün Sahibi, Geliştirme Ekibi ve Scrum Master’dan oluşur. Takım kendi kendini örgütler. Böylece kendi içerisinde uyum içinde olan takımlar daha başarılı sonuçlar alırlar. Scrum’ın takım modeli esneklik, yaratıcılık ve verimliliği optimize etmek için tasarlanmıştır.

## Ürün Sahibi (Product Owner)

Ürünün değerinden ve gerikaydından (Backlog) sorumlu olan kişidir . Ürün Backlog öğeleri açık olmalıdır. Bunuda sağlayacak olan kişi Product Owner’dır. Sorumluluğun büyük bir kısmı Product Owner’dadır. Takımda Product Owner tektir. Product Owner’in başarılı olması için, tüm organizasyonun kendi kararlarına saygı göstermesi gerekir. Geliştirme Ekibi, Product Owner’in kararlarına göre hareket eder .

# Scrum Roller

## Sprint

Scrum çerçevesinde her bir iterasyon Sprint olarak isimlendirilmektedir. Sprintleri paketlenmiş zaman aralıkları olarak düşünebilirsiniz. Yeni Sprint önceki Sprint sonucundan hemen sonra başlar. Sprintlerimiz başlangıç ve bitiş tarihleri bulunmaktadır, ve ekibin hızını ayarlamak için her bir sprint'in aynı sürede tutulması önerilmektedir. Scrum yöntemi ile yönetilen projeler “sprint” denilen fazlara ayrılır. Her bir “sprint”in süresi genellikle (en fazla) 30 gündür. İdeal olan proje boyunca tüm “sprint”ler aynı süreli olur.

## Sprint İnceleme

Burada katılımcılar, gelecek hakkında iş birliği yapabilir. Gayriresmi bir toplantı olup iş birliğini daha iyi gerçekleştirmek için toplanılır. Bu, bir aylık sprint için dört saatlik zamanı kapsar. Sprint İnceleme sonucu bir sonraki Sprint için muhtemel Ürün Backlog öğelerini tanımlar.

## Sprint Planlama Toplantısı

Sprint ile yapılacak çalışma Sprint Planlama Toplantısı ile planlanmıştır. Bu plan tüm Scrum Ekibinin ortak çalışmasıyla oluşturulur. Örneğin, iki haftalık Sprint'te dört saatlik Sprint Planlama Toplantıları vardır. Bu toplantılarda artımın nasıl sağlanacağı konuşulur.

## Günlük Scrum (Daily Scrum)

15 dakikalık toplantılardan oluşur. Bu günlük Scrum'larda planlama yapıldığı gibi bir sonraki gün için tahminler yapılır. Bu toplantılar her gün düzenlenir ve amaç karmaşıklığı azaltmaktır. Bu toplantılarda her bir üye aşağıdaki sorulara cevap verir :

- Son toplantıdan bu yana neler tamamlandı ?
- Sonraki toplantıda neler yapılacak ?
- Engeller ne şekildedir ?

Kısaca hedefe doğru ilerlemeler değerlendirilir.



# Scrum Rollerleri

---

## Sprint Backlog

Takım belirlenir ve “Sprint Planning” dediğimiz en fazla 4 hafta sürecek olan küçük döngülerle “Sprint” ile işe başlanır. Her döngü için ürün gerikaydından önemli gereksinimler seçilerek sprint gerikaydı (Sprint Backlog) oluşturulur ve sprint boyunca bu gereksinimler geliştirilir.

## Product Backlog

İlk önce müşteriye yani ürün sahibine (Product Owner), istediği ürün gereksinimlerinin neler olduğu sorulur ve bu ihtiyaçlar çıkartılır (Product BackLog). Product BackLog içerisinde maddeler en önemlisinden en aza doğru sıralanır. Belirli bir periyot içerisinde belirtilen gereksinimleri karşılayan tam olarak bir portatif müşteriye teslim edilir .

## Sprint Retrospective

Sprintte bir sonraki aşama için iyileştirme fırsatı yakalanır. Bu bir aylık sprint için toplamda 3 saatlik toplantıdır. Burada takım işini kolaylaştırmak için planlar oluşturulur. Böylece gelişim süreci ve sonraki Sprinti daha etkili ve zevkli hale getirmek için uygulamalar geliştirme ortamı yaratılır. Bu toplantılar sonucunda ürün kalitesini artırma yolları planlanmış olur.

# Scrum Rollerleri

## Scrum Ustası(The Scrum Master)

Scrum'dan sorumlu olan kişidir. Scrum Master, Scrum Takım teorisi ve uygulamaları kurallara uyarak sağlar. Scrum Master takım için liderdir. Scrum Master, ekip tarafından yaratılan değeri maksimize etmek için çalışır. Scrum Master, çeşitli yollarla Ürün Sahibine yardım eder . Bunlar:

- Ürün Backlog yönetimi bulma teknikleri
- Geliştirme ekibinin vizyon ve hedeflerini belirlemek
- Açık ve özlü Ürün Backlog öğeleri oluşturmak için geliştirme ekibine temel yolların öğretilmesi
- Anlaşılabilir bir ortamda uzun vadeli ürün planlamaları yapmak
- Çevikliği anlama ve uygulama

### **Scrum Master, geliştirme ekibinde hizmet eder:**

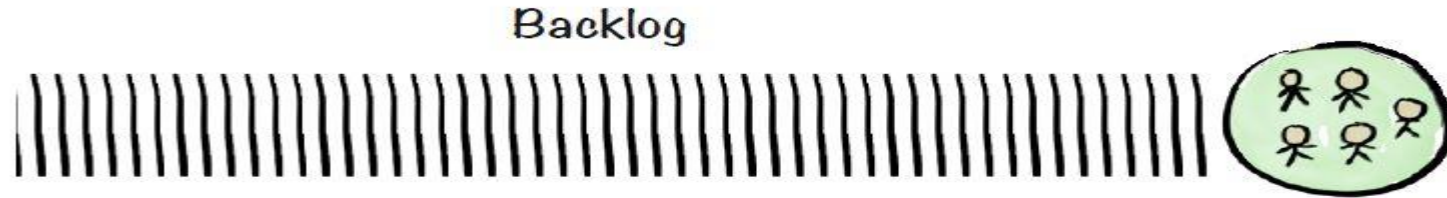
- Örgütlenmeye koçluk eder
- Geliştirme ekibi için değeri yüksek ürünler oluşturmak
- Geliştirme ekibinin ilerlemesine engel olacak şeyleri ortadan kaldırmak
- Talebe göre Scrum olaylarının kolaylaştırılması

### **Scrum Master organizasyon içinde çalışır:**

- Scrum'ın belirlenmesinde koçluk eder
- Örgüt içinde planlama yapar
- Takımın verimliliğini artırır

# 6. Scrum Modeli

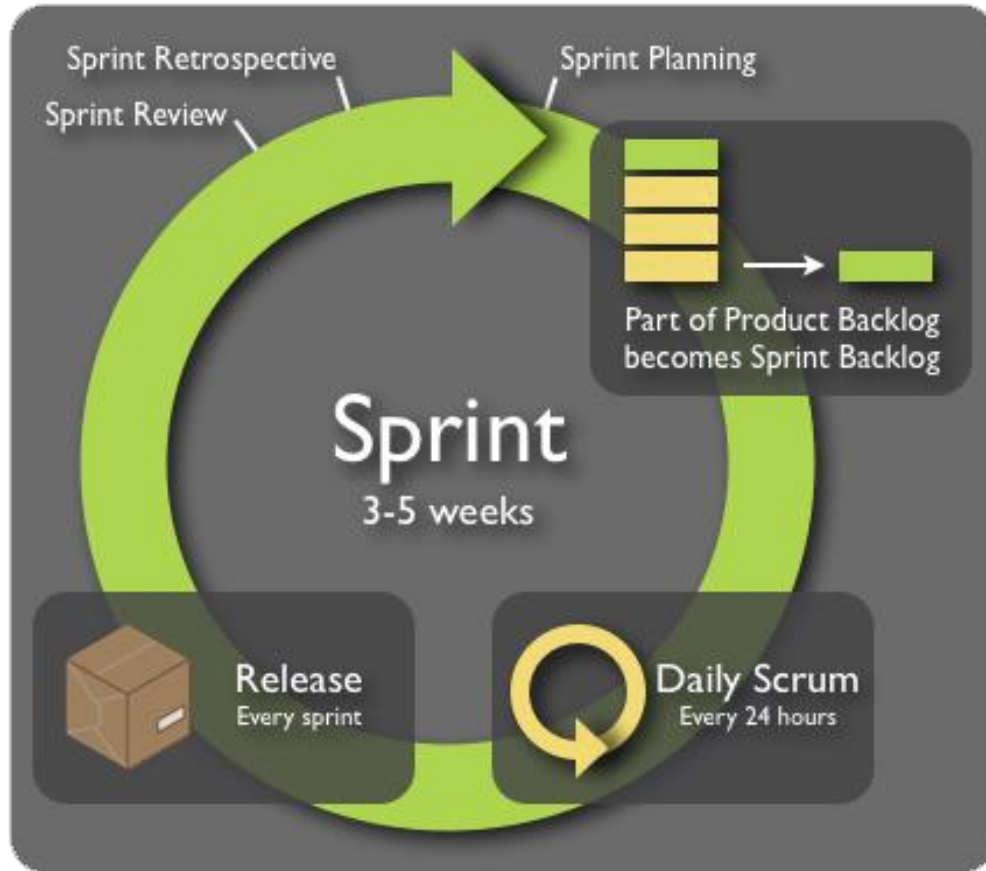
---



## **Backlog:**

- Müşteriden ve son kullanıcıdan gelen gereksinimleri içerir.
- "Ne yapacağız" sorusunun yanıtını içerir.
- Herkese açık ve herkes tarafından müdahale edilebilir.
- Risk, iş değeri, zaman gibi kavramlara göre ürün sahibi tarafından sıralandırılır.
- User Story'lerden oluşur.

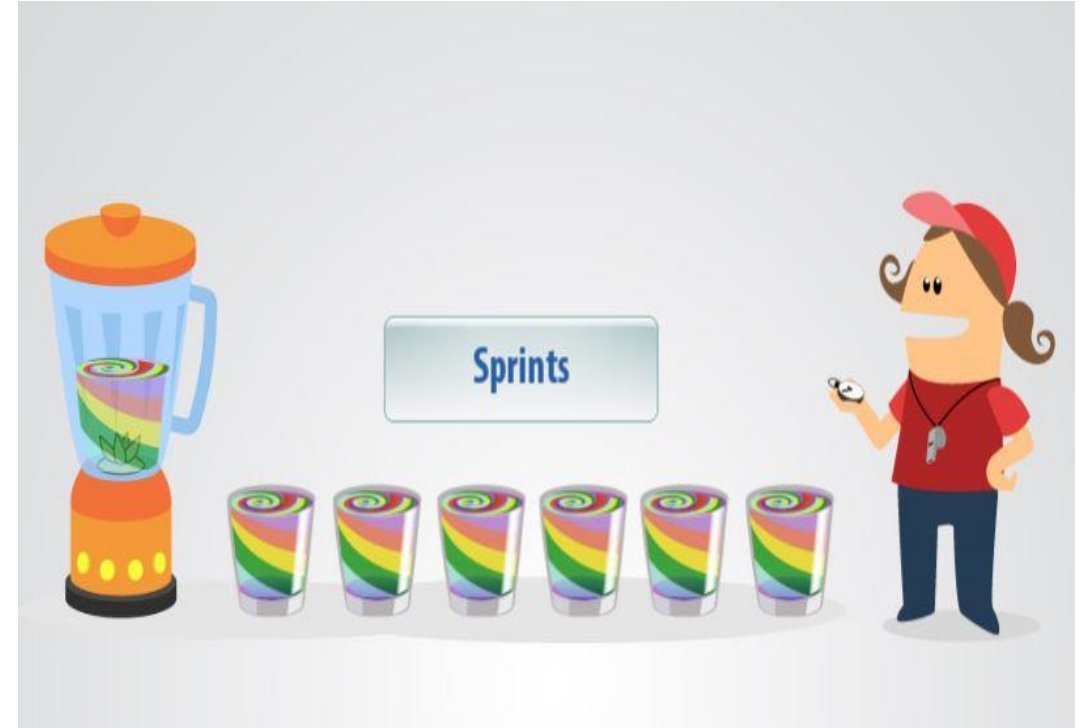
# 6. Scrum Modeli



## Sprint

- Belirli bir süreye sahiptir.
- Sonunda ortada değeri olan bir çıktı olmalıdır.
- Toplantılarla içerik belirlenir.
- Sprint süresi boyunca her gün toplantılar yapılır.

## 6. Scrum Modeli

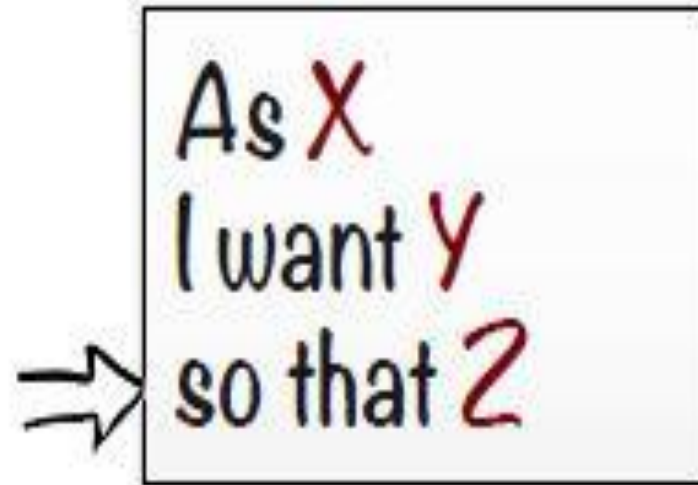


Sprint Gösterimi

## 6. Scrum Modeli

---

- **User Story:** Müşteri, son kullanıcı veya ürün sahibi için değerli olan ve anlam ifade eden genellikle fonksiyonel özelliklerin belirtildiği ifadelerdir.



## 6. Scrum Modeli

---

- Örnek User Story: Online alışveriş yapan biri olarak, alışverişe daha sonra devam edebileyim diye, alışveriş kartımın kaydedilmesini istiyorum.

As online buyer  
I want to save my shopping cart  
so that I can continue shopping later

## 6. Scrum Modeli

Fact: Features have different sizes

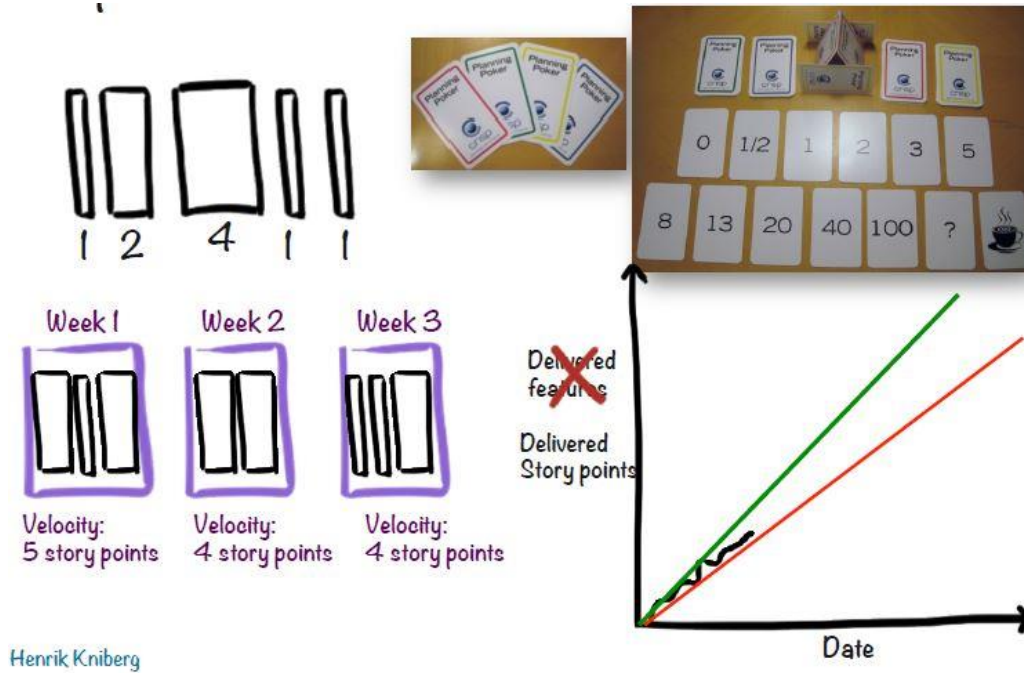


- Her bir user story farklı bir boyuttadır. Somut olarak bakarsak, bir projedeki her bir gereksinim için gereken iş gücü ve zaman aynı değildir. Bu sebeple ürün backlogları sprintlere bölünürken, user storylerin boyut ve öncelikleri göz önünde bulundurulur. Örneğin bir sprint 3 user story içerirken diğeri daha küçük boyutlarda 5 user story içerebilir. Peki boyutları nasıl belirleyeceğiz?

İş görüldüğü gibi değil!

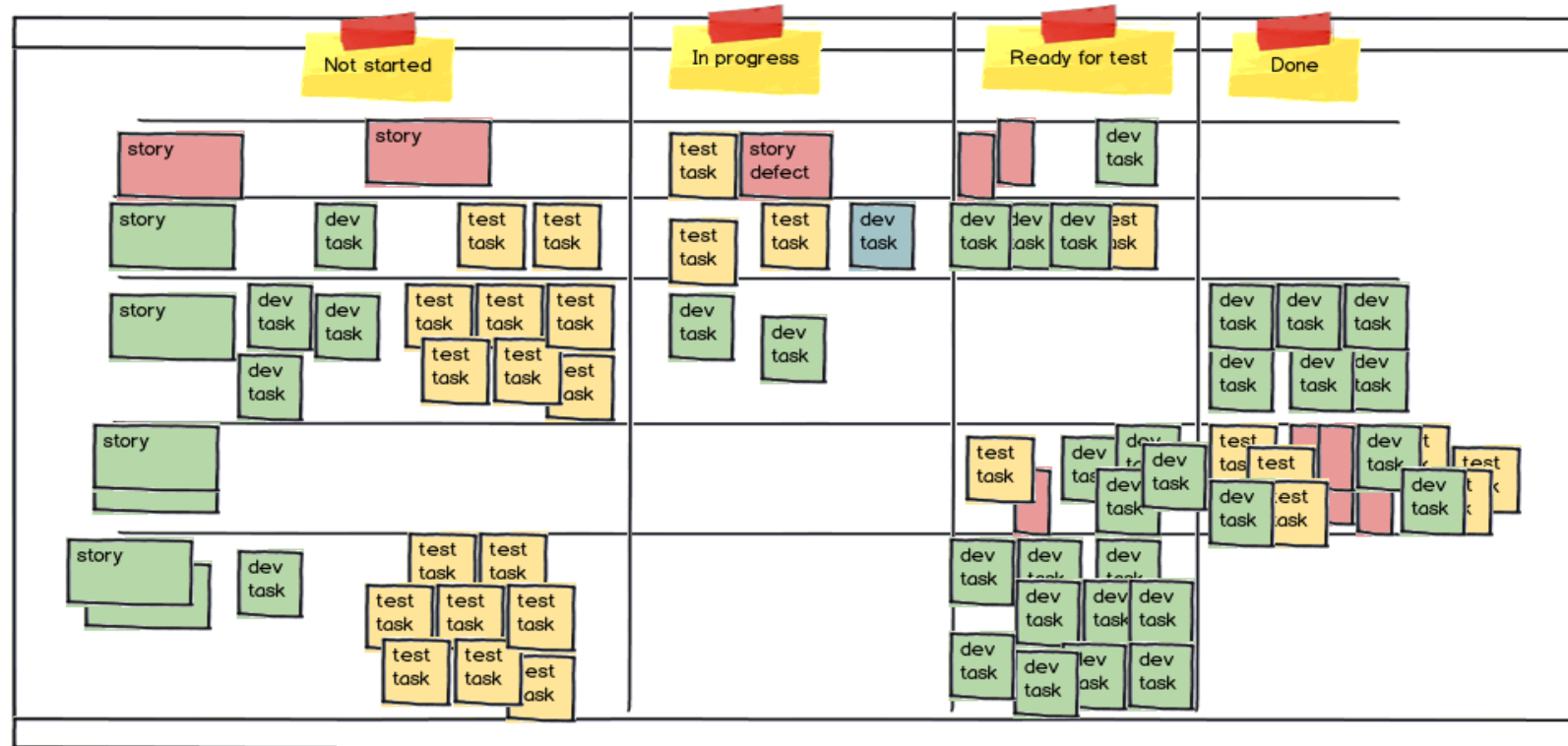


# 6. Scrum Modeli



- **Poker Kartları:** Scrum takım üyeleri bir araya gelir. Scrum master bir user story okur. Takımdaki her bir üye user story için uygun gördüğü poker kartlarından birini seçer. Herkes kartları seçtikten sonra tüm kartlar açılır ve değerlendirilir. Böylece herkesin ortak görüşü sonunda user story'lerin büyüklüğü belirlenir.

# 6. Scrum Modeli



created with Balsamiq Mockups - [www.balsamiq.com](http://www.balsamiq.com)

Scrum Board

# 6. Scrum Modeli

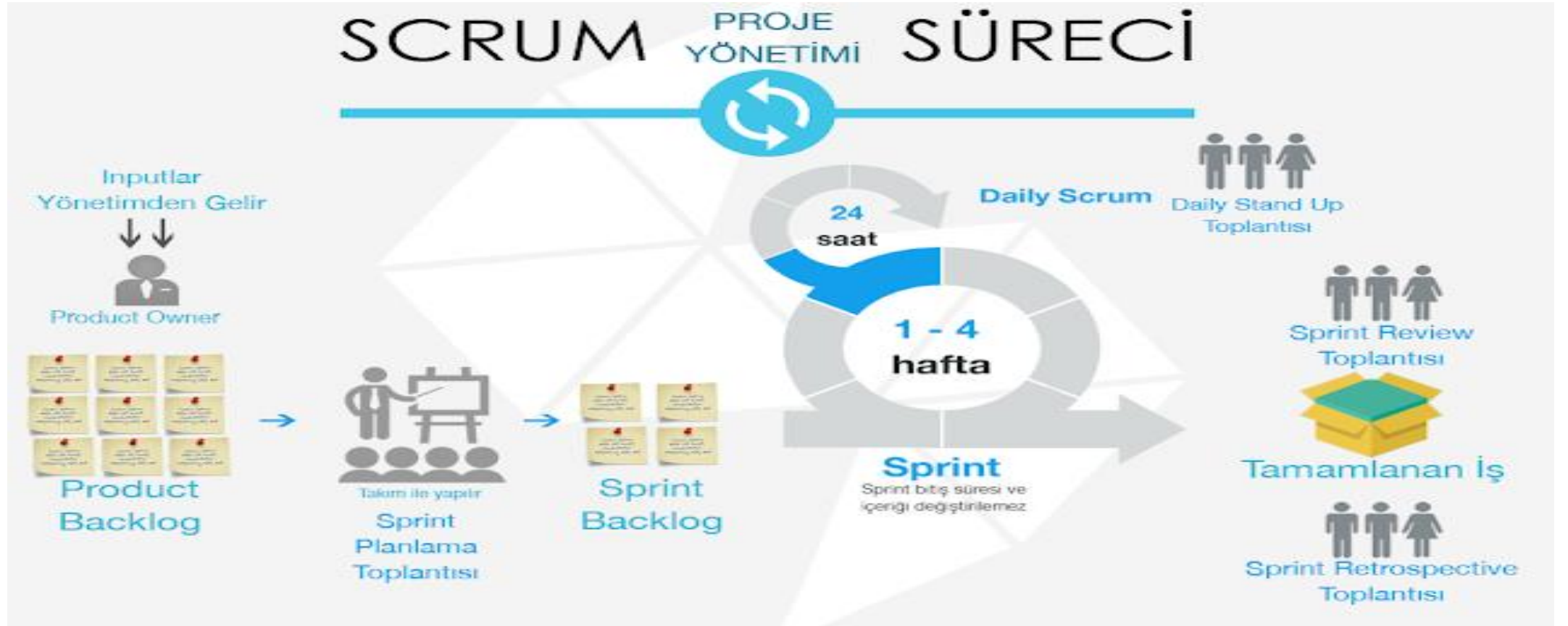
## SCRUM

- Sprint (2 hafta-1 ay)
- Sprintler en son halini aldıktan, toplantı yapıldıktan sonra değişmez.
- Özellikler geliştiriciler tarafından derecelendirilir.
- Herhangi bir mühendislik pratiği tanımlamaz.

## XP (EXTREME PROGRAMMING)

- Sprint (1 yada 2 hafta)
- Sprintler değişebilir.
- Özellikler ürün sahibi tarafından derecelendirilir.
- Mühendislik pratikleri tanımlar. Eşli programlama, otomatik test, basit dizayn vs.

# 6. SCRUM MODELİ



Agile is...

**Early delivery of business value**

**Less bureaucracy**

# Sorularınız

---



# Çalışma Soruları

---

1. Agile modelinin geliştirilmesine neden ihtiyaç duyulmuştur? Açıklayınız.
2. Agile takımları kimlerden oluşur ve iş paylaşımları nasıldır?
3. Agile modelinin diğer modellerden farkları nelerdir? Üzerinde durduğu temel noktalar nelerdir?
4. User story, backlog ve sprint kavramlarını açıklayınız.
5. Agile ile geliştirilmiş büyük yazılım projelerine örnekler veriniz.
6. Scrum haricindeki diğer çevik yazılım geliştirme yöntemlerini açıklayınız.

# Kaynaklar

---

- [1] Martin, Micah, and Robert C. Martin. *Agile principles, patterns, and practices in C#*. Pearson Education, 2006.
- [2] Kniberg, Henrik. "Scrum and XP from the Trenches." *Lulu. com* (2007).
- [3] Kniberg, Henrik. " What is agile" , 2013
- [4] [http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development)
- [5]<http://www.mshowto.org/microsoft-visual-studio-team-foundation-server-nedir.html>
- [6] [www.kurumsaljava.com/download/10/](http://www.kurumsaljava.com/download/10/)