



YMH 311-YAZILIM TASARIM VE MİMARİSİ

Dr. Öğr. Üyesi BİHTER DAŞ

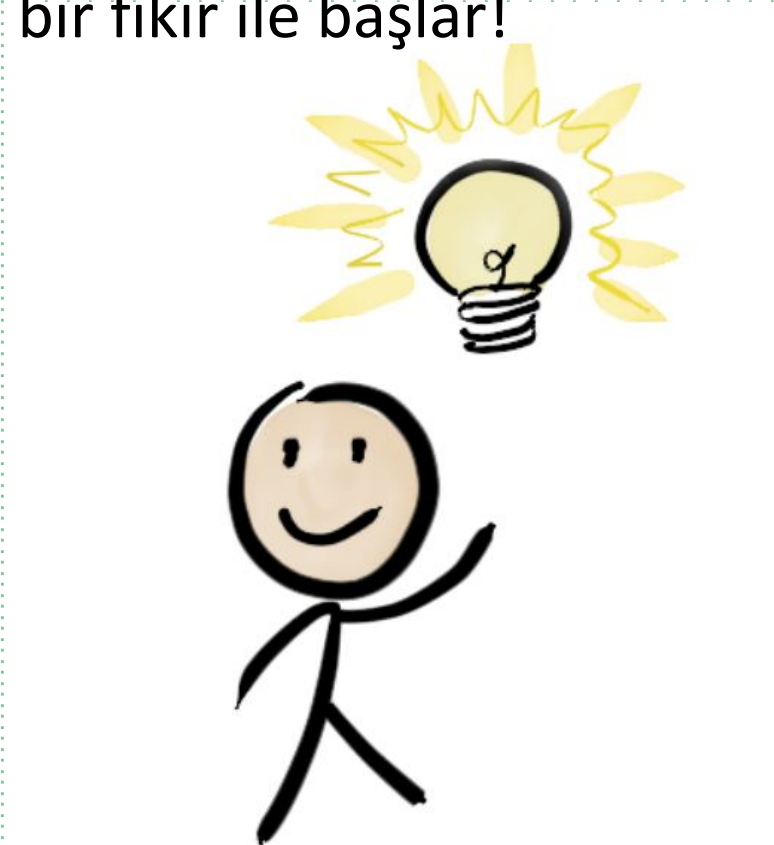
Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü

3. HAFTA:

ÇEVİK (AGİLE) YAZILIM GELİŞTİRME
SÜREÇ MODELLERİ

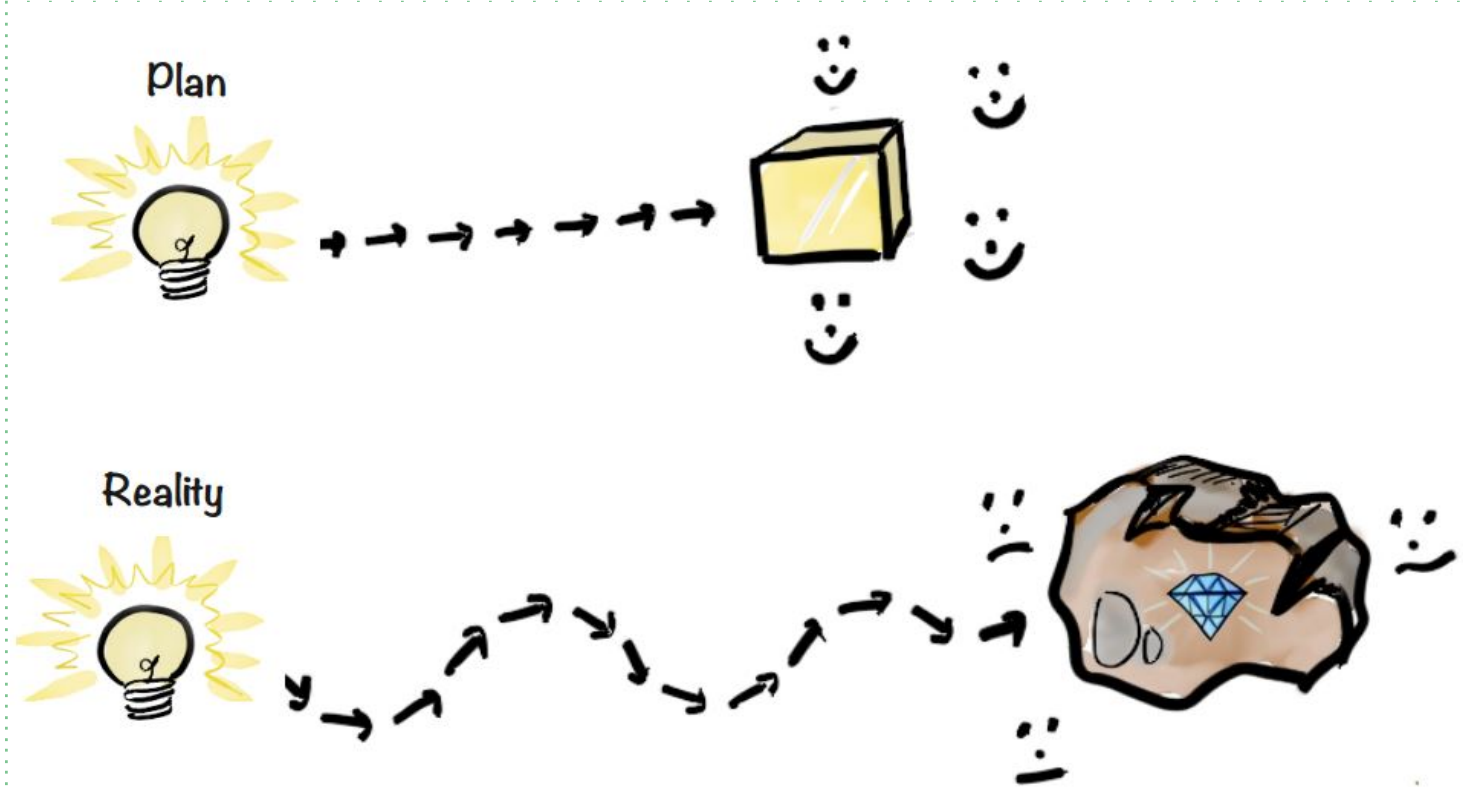
1.Günümüzde Yazılım Projelerinin Durumu

- 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%**

Plan: €1,000,000

Actual: €2,700,000

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

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

Plan: €1,000,000

Actual: €1,700,000

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:

- Hızlı değişen müşteri gereksinimlerinin doğru olarak belirlenememesi;
- Proje yönetim metotlarının uygulanmadan geliştirilmesi;
- Süreçlerin iyi takip edilememesi;
- Değişikliklerin esnek uygulanamaması
- Proje için uygun ekibi kuramamak;
- Yanlış teknoloji ve mimari seçimleri;
- Geleneksel yöntemlerin eksiklikleri;
- Maliyet zaman aşımı;
- Müşteriyle iletişimden kaçınmak vs.

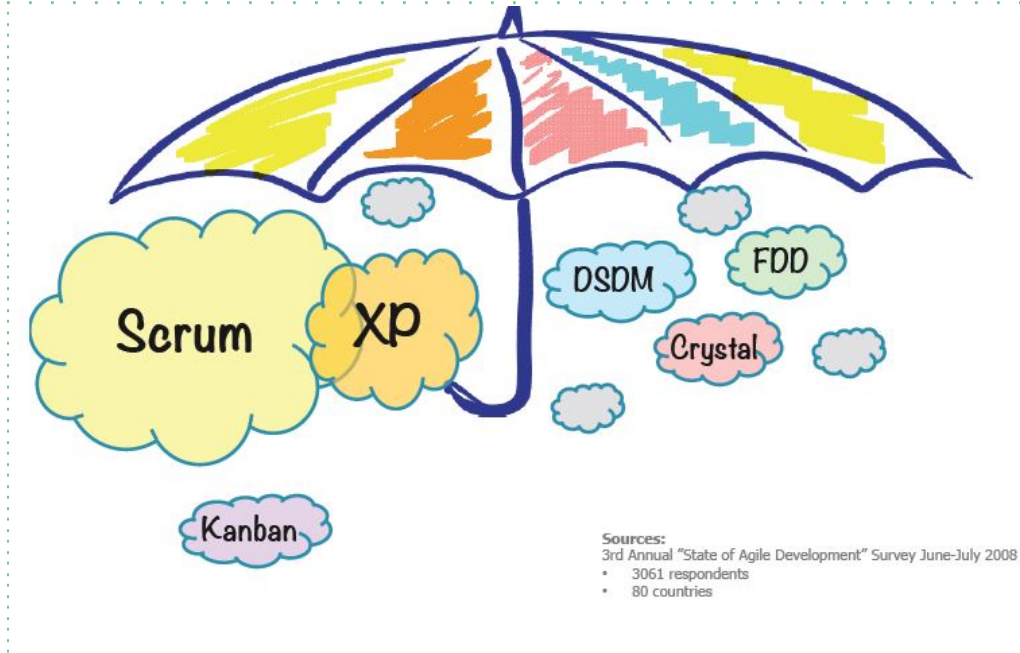


1.Günümüzde Yazılım Projelerinin Durumu

Peki ne yapmalıyız?



5. Çevik Yazılım Şemsiyesi



Agile Manifesto

4 Temel değer

- ❑ Süreçler ve araçlardan ziyade **bireyler ve etkileşimlere**
- ❑ Kapsamlı dokümantasyondan ziyade **çalışan yazılıma**
- ❑ Sözleşme pazarlıklarından ziyade **müşteri iş birliğine**
- ❑ Bir plana bağlı kalmaktan ziyade **değişime karşılık vermeye**

12 destekleyici ilke

- ❑ En önemli önceliğimiz değerli yazılımın erken ve devamlı teslimini sağlayarak müşterileri memnun etmektir.
- ❑ Değişen gereksinimler yazılım sürecinin son aşamalarında bile kabul edilmelidir. Çevik süreçler değişimi müşterinin rekabet avantajı için kullanır.
- ❑ Çalışan yazılım, tercihen kısa zaman aralıkları belirlenerek birkaç haftada ya da birkaç ayda bir düzenli olarak müşteriye sunulmalıdır.

- ❑ İş süreçlerinin sahipleri ve yazılımcılar proje boyunca her gün birlikte çalışmalıdırlar.
- ❑ Projelerin temelinde motive olmuş bireyler yer almalıdır. Onlara ihtiyaçları olan ortam ve destek sağlanmalı, işi başaracakları konusunda güven duyulmalıdır.
- ❑ Bir yazılım takımında bilgi alışverişinin en verimli ve etkin yöntemi yüz yüze iletişimdir.
- ❑ Çalışan yazılım ilerlemenin birincil ölçüsüdür.
- ❑ Çevik süreçler sürdürülebilir geliştirmeyi teşvik etmektedir. Sponsorlar, yazılımcılar ve kullanıcılar sabit tempoyu sürekli devam ettirebilmelidir.
- ❑ Teknik mükemmeliyet ve iyi tasarım konusundaki sürekli özen çevikliği artırır.
- ❑ Sadelik, yapılmasına gerek olmayan işlerin mümkün olduğunca arttırılması sanatı, olmazsa olmazlardandır.
- ❑ En iyi mimariler, gereksinimler ve tasarımlar kendi kendini örgütleyen takımlardan ortaya çıkar.
- ❑ Takım, düzenli aralıklarla nasıl daha etkili ve verimli olabileceğinin üzerinde düşünür ve davranışlarını buna göre ayarlar ve düzenler.

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.

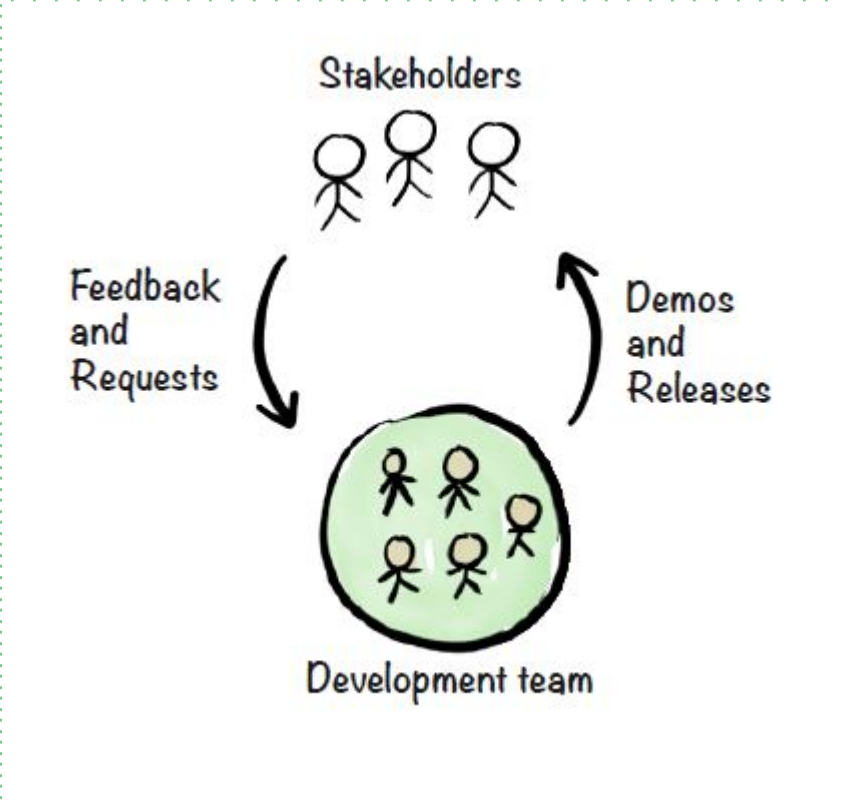
Taleplerdeki geç değişikliklerin de memnuniyetle karşılanır.

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

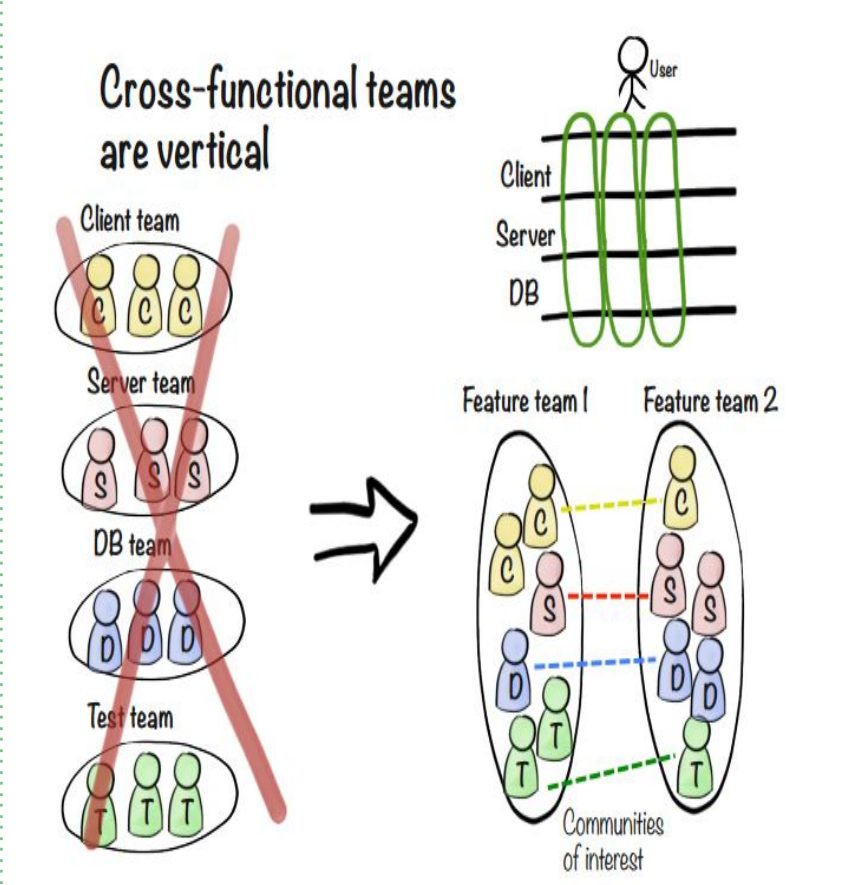
- Multitasking
- Video calls



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



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

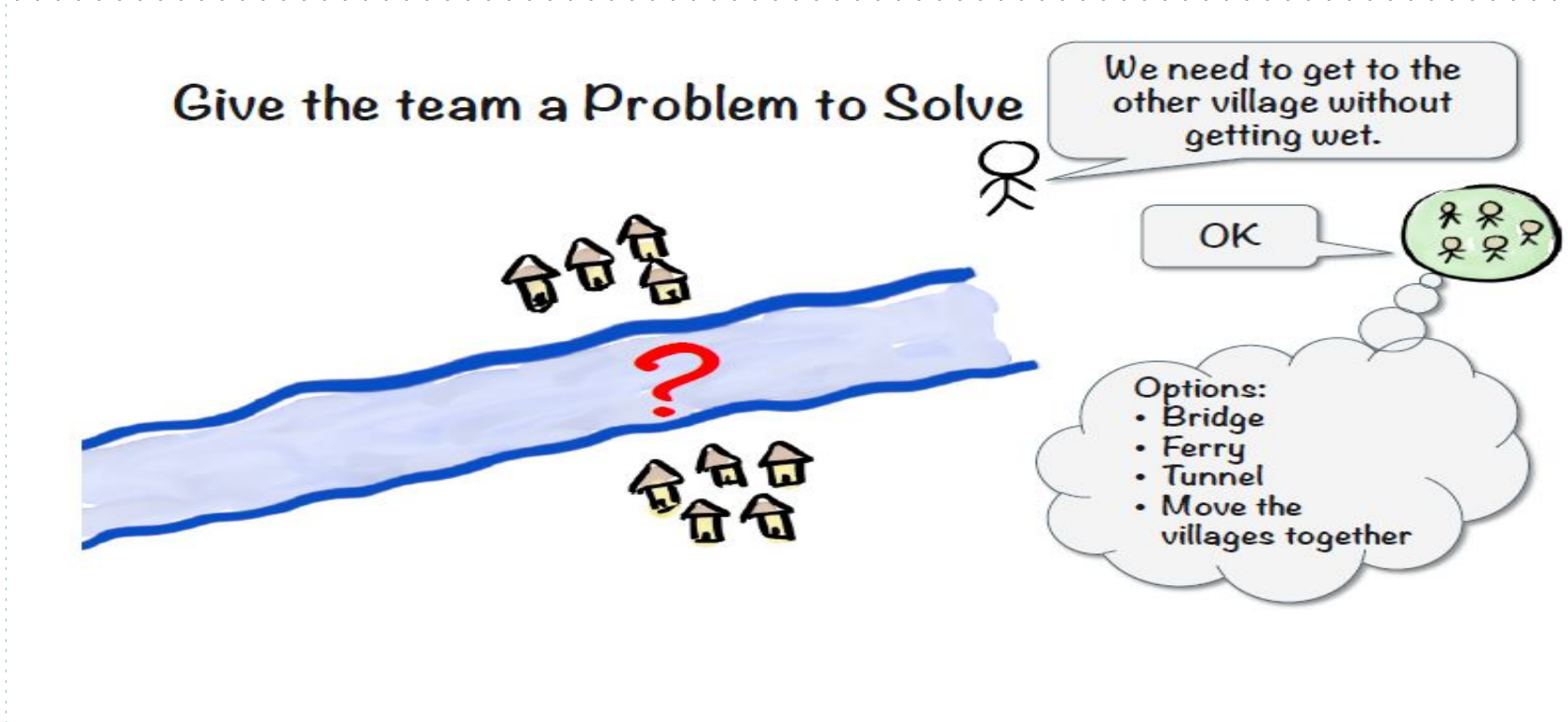
2.1.Çevik Model Takımları

Takımlara çözümü söylemeyin!



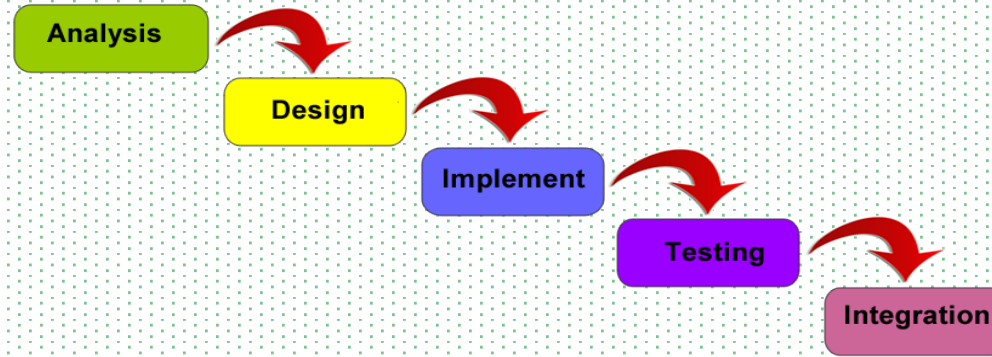
2.1.Çevik Model Takımları

Sorunu söyleyin, onlar çözümü üretsin!



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.



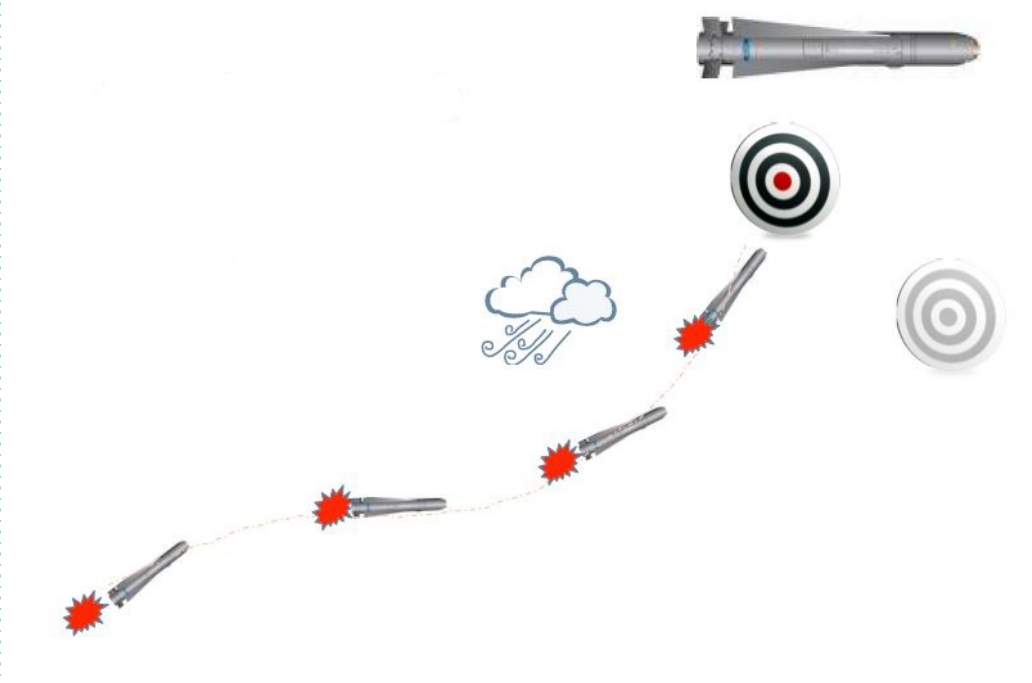
Waterfall Model

3. Geleneksel Model vs. Agile



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

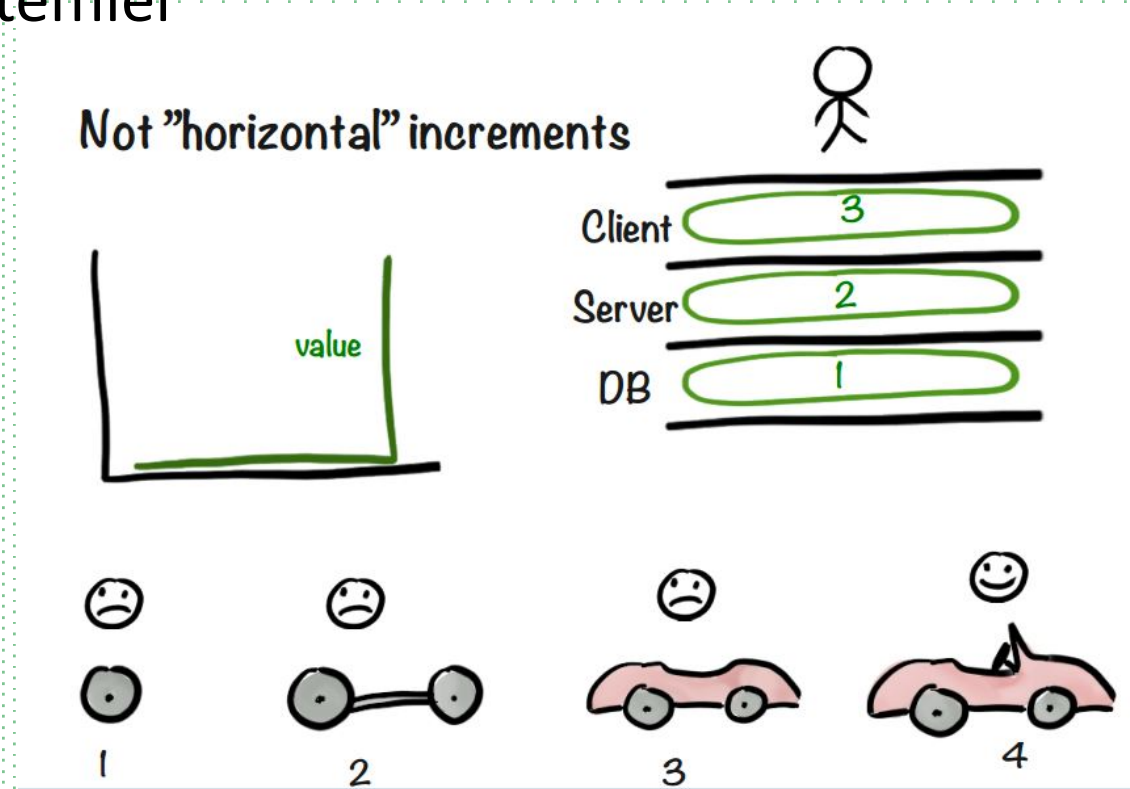
3. Geleneksel Model vs. Agile



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

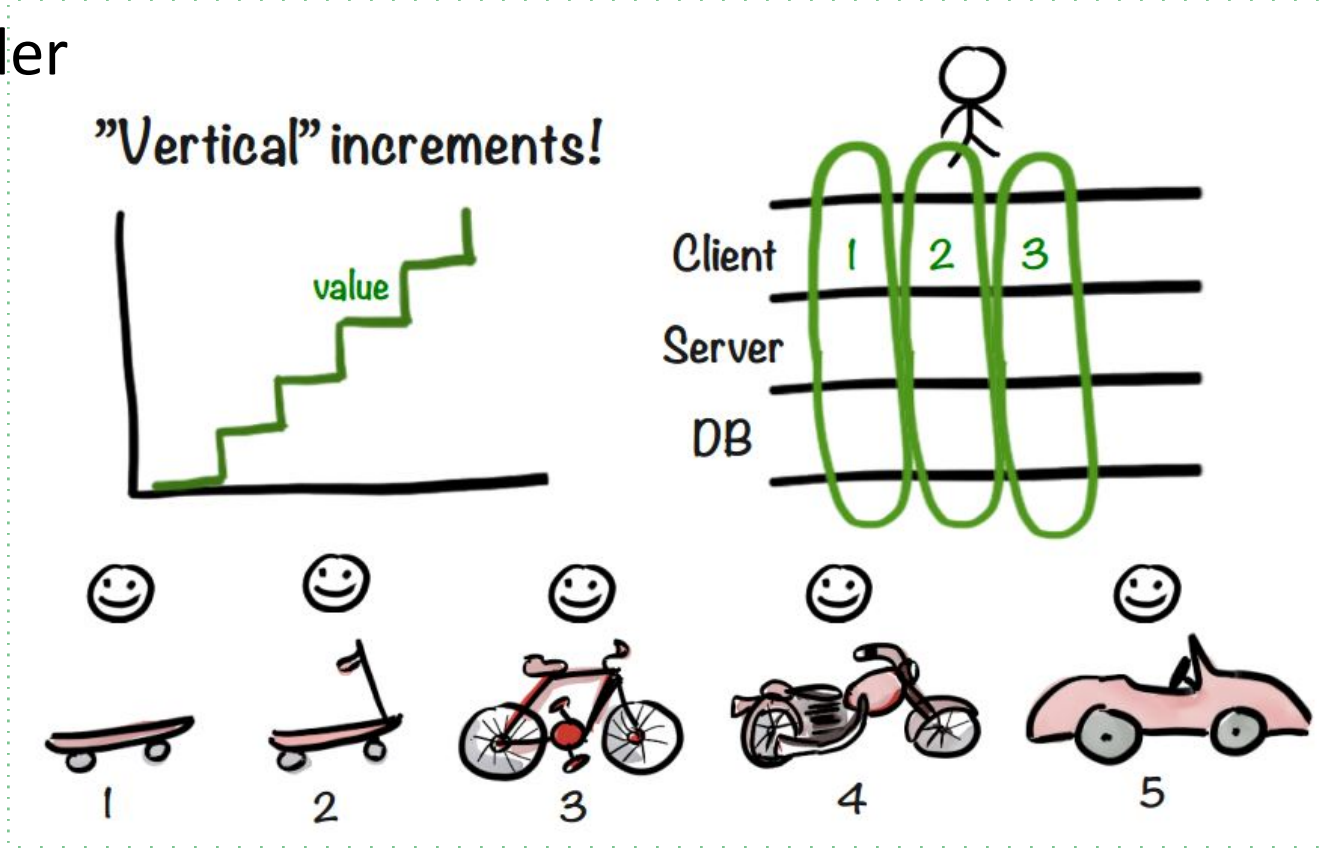
3. Geleneksel Model vs. Agile

Geleneksel Yöntemler

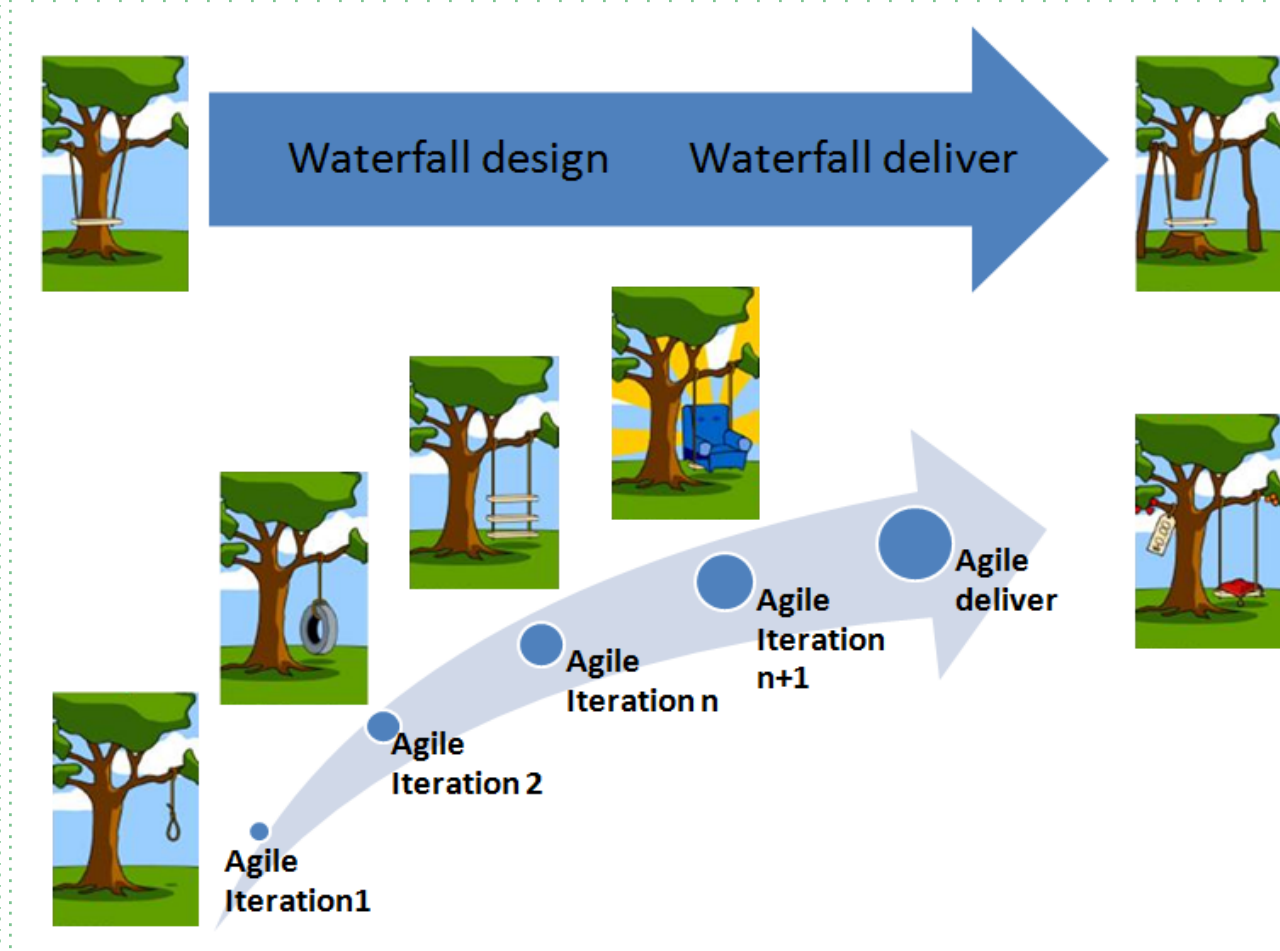


3. Geleneksel Model vs. Agile

Çevik Yöntemler

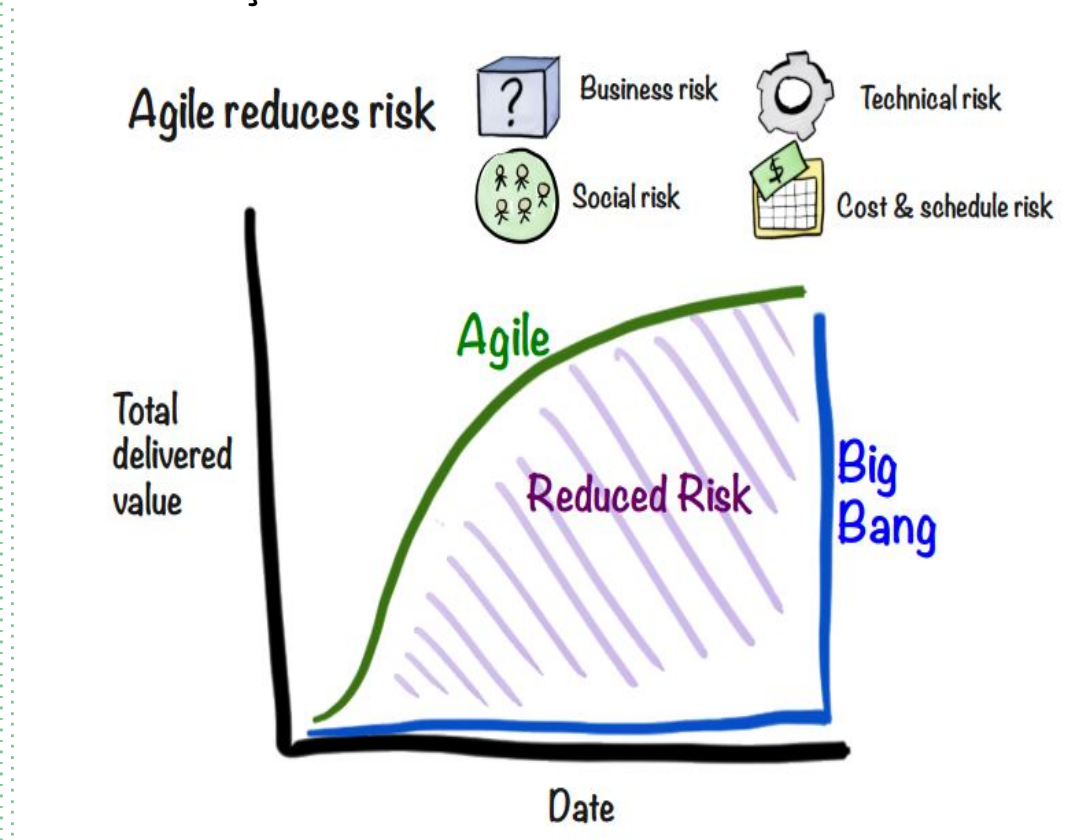


3. Geleneksel Model vs. Agile

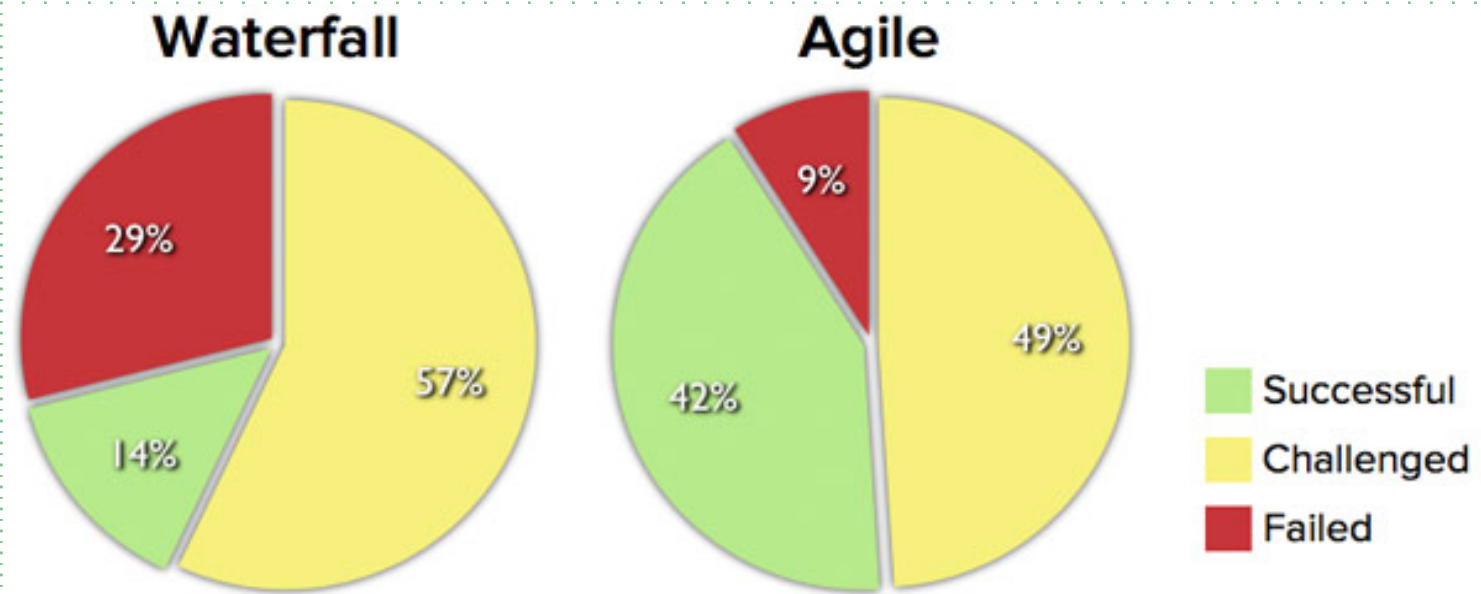


3. Geleneksel Model vs. Agile

Çevik modeller riski azaltır!



3. Geleneksel Model vs. Agile

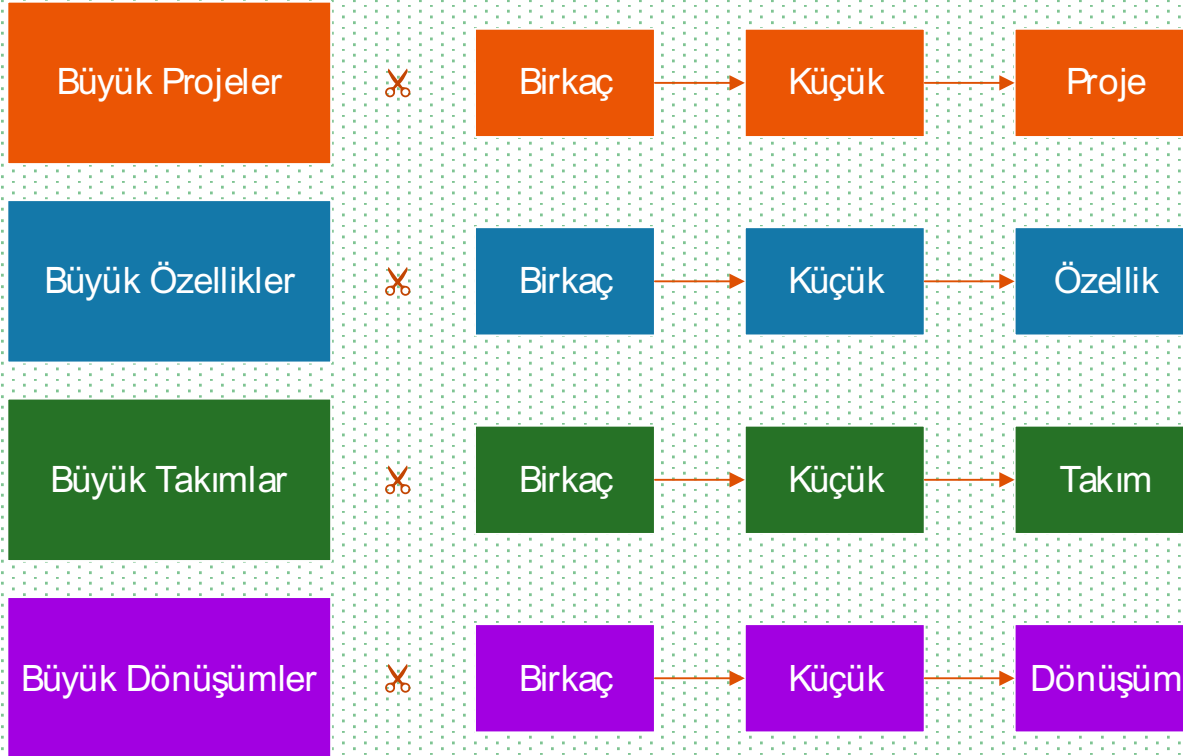


Source: The CHAOS Manifesto, The Standish Group, 2012.

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

6. Scrum Modeli

3 Roles

- Product Owner
- Developers
- Scrum Master

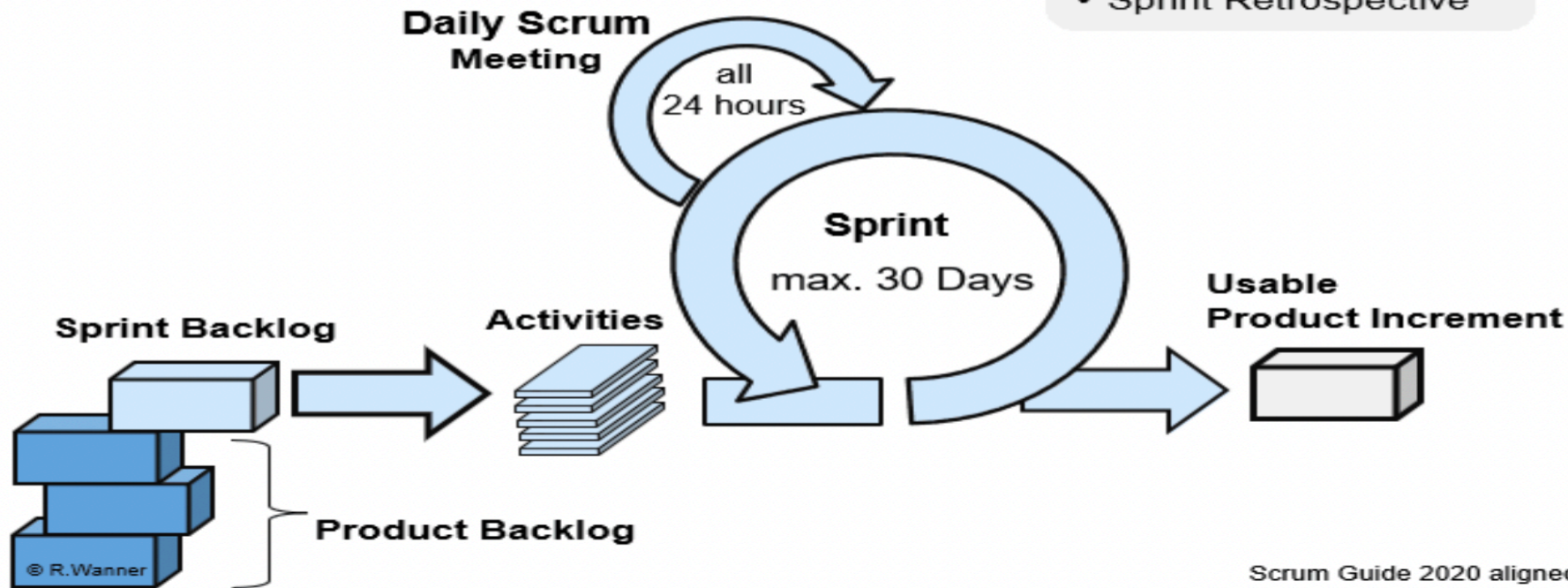
Scrum Team

3 Artifacts

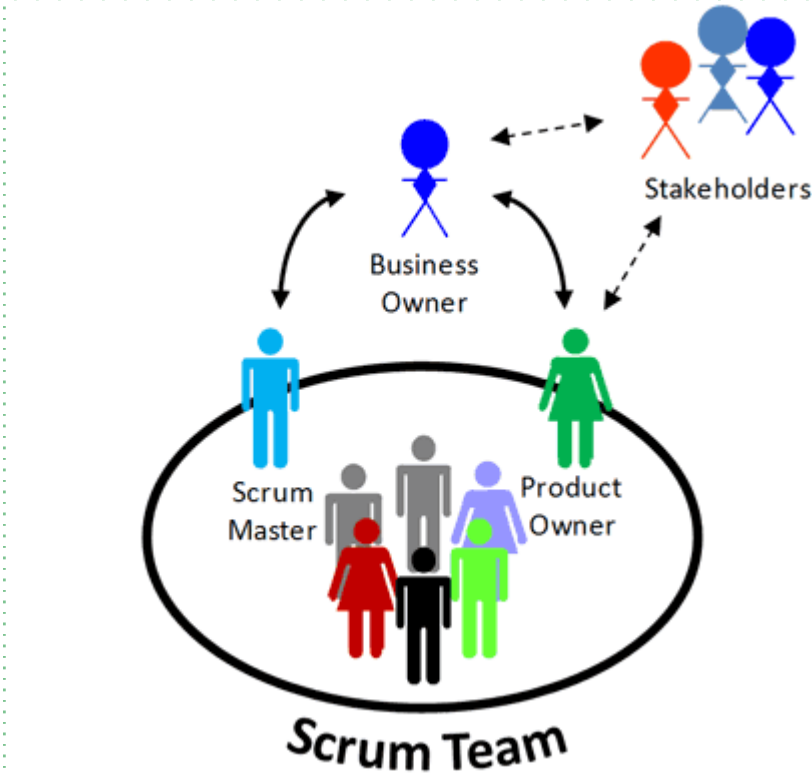
- Product Backlog
- Sprint Backlog
- Product Increment

5 Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective



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.

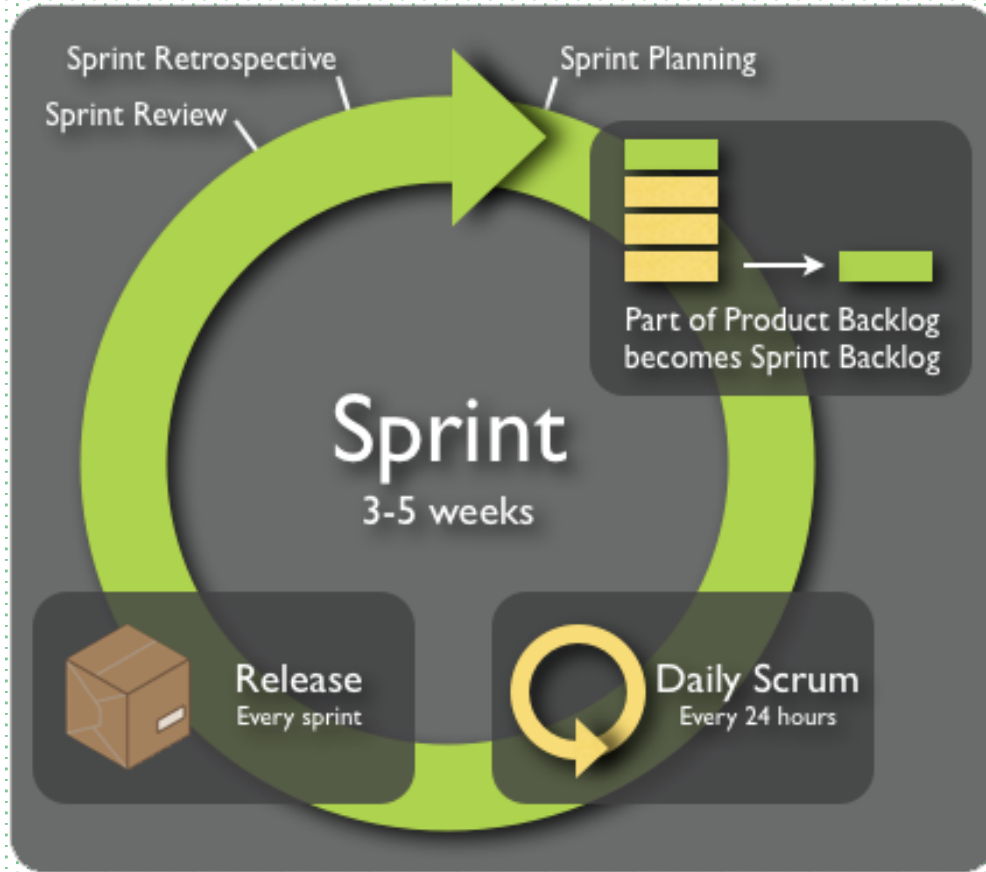
6. Scrum Modeli



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.

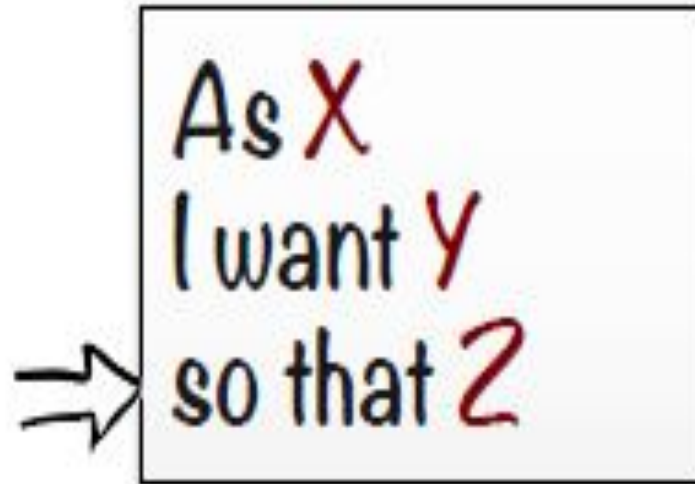
6. Scrum Modeli



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.



As X
I want Y
so that Z

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

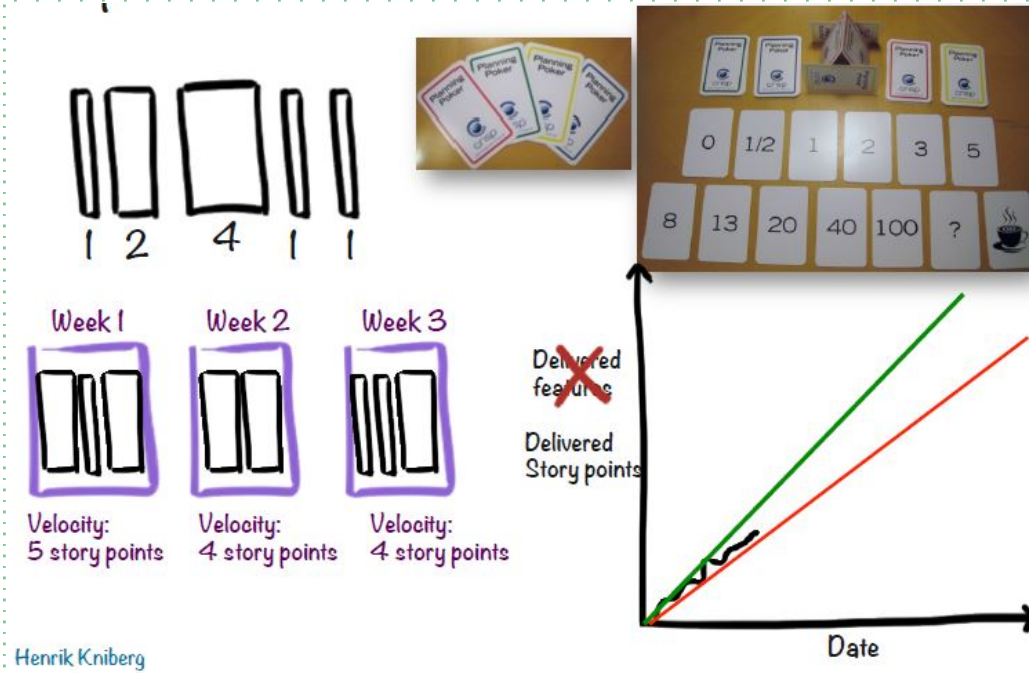
Fact: Features have different sizes



➤ 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?**

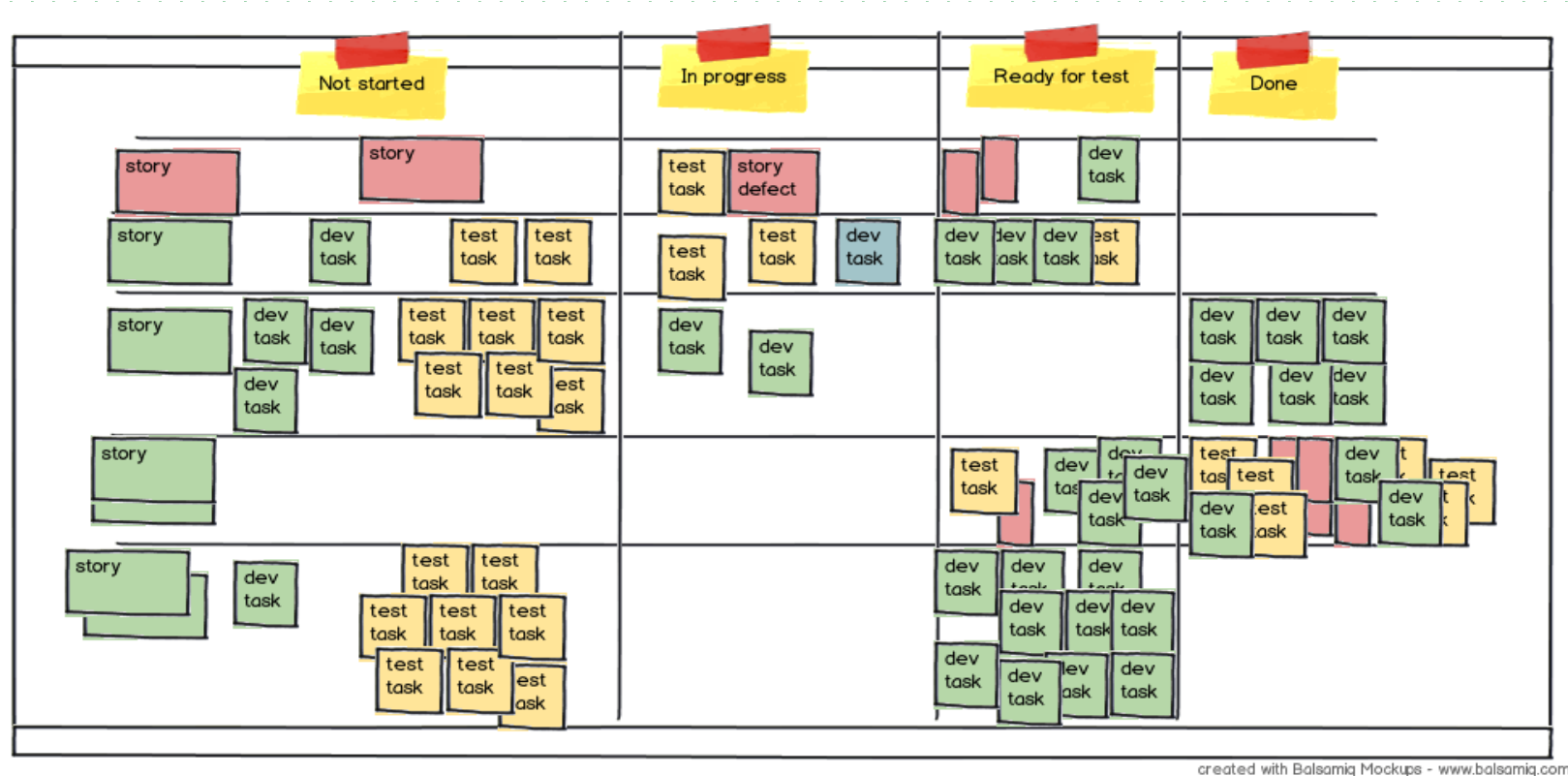
İş görüldüğü gibi değil!

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

Scrum Board

6. Scrum Modeli

SCRUM

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

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.

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
- [5] <http://www.mshowto.org/microsoft-visual-studio-team-foundation-server-nedir.html>
- [6] www.kurumsaljava.com/download/10/