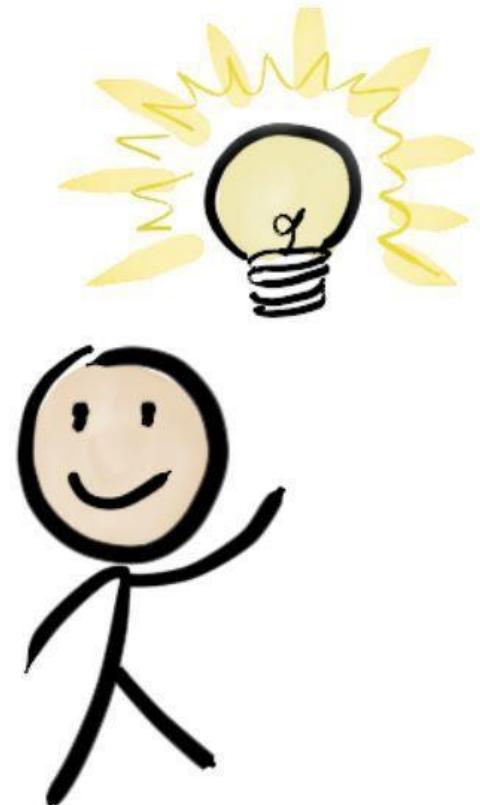
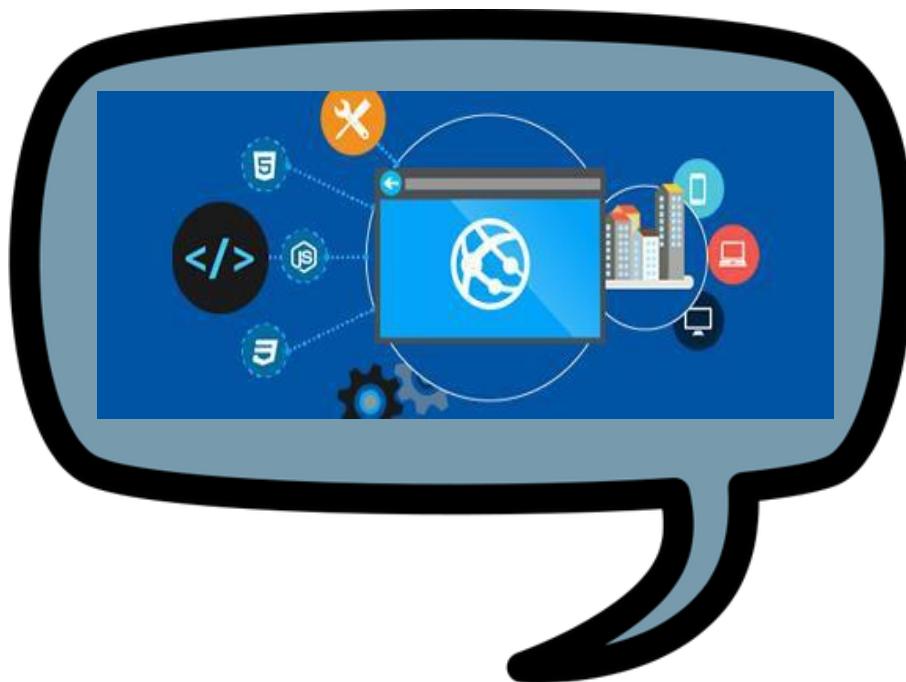


İçindekiler

1	Günümüzde Yazılım Projelerinin Durumu.....	3
2	Çevik Yazılım Geliştirme Yöntemi.....	9
3	Geleneksel Model vs. Agile.....	17
4	Değerlendirme.....	26
5	Çevik Yazılım Şemsiyesi.....	28
6	Scrum Modeli.....	29

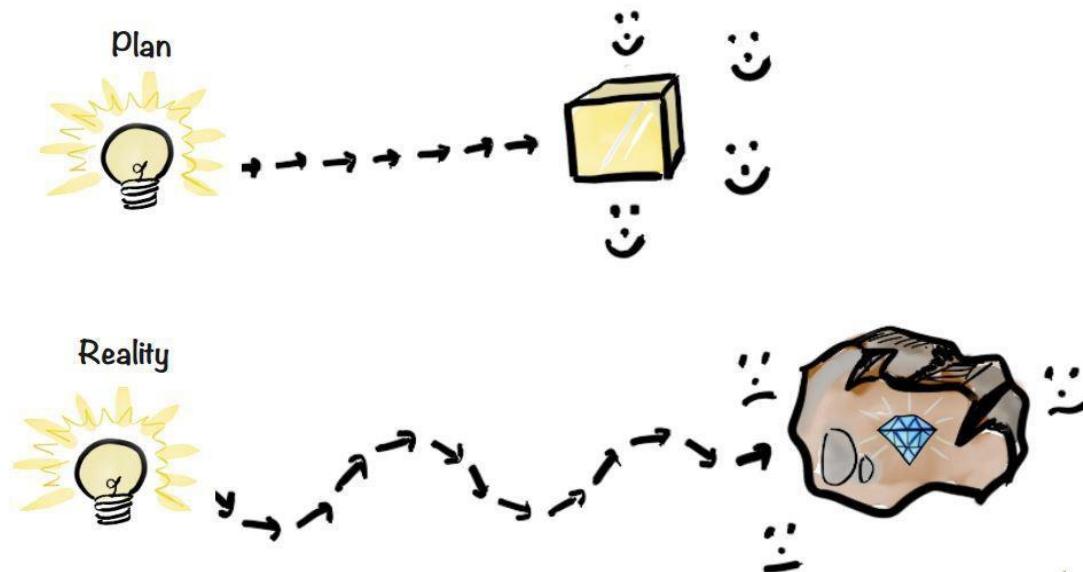
1.Günümüzde Yazılım Projelerinin Durumu

- Birçok proje harika bir fikir ile başlar!



1.Günümüzde Yazılım Projelerinin Durumu

- Bu projelerin büyük bir kısmının başarısız olması muhtemeldir!

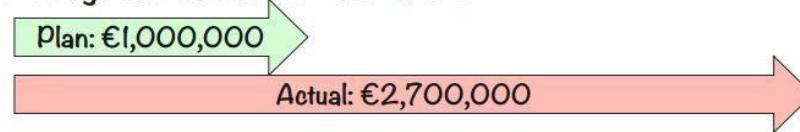


1.Günümüzde Yazılım Projelerinin Durumu

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

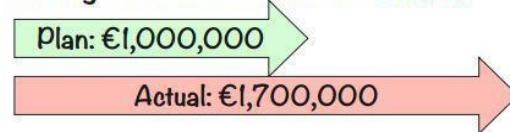
IT project success rate 1994: 15%

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

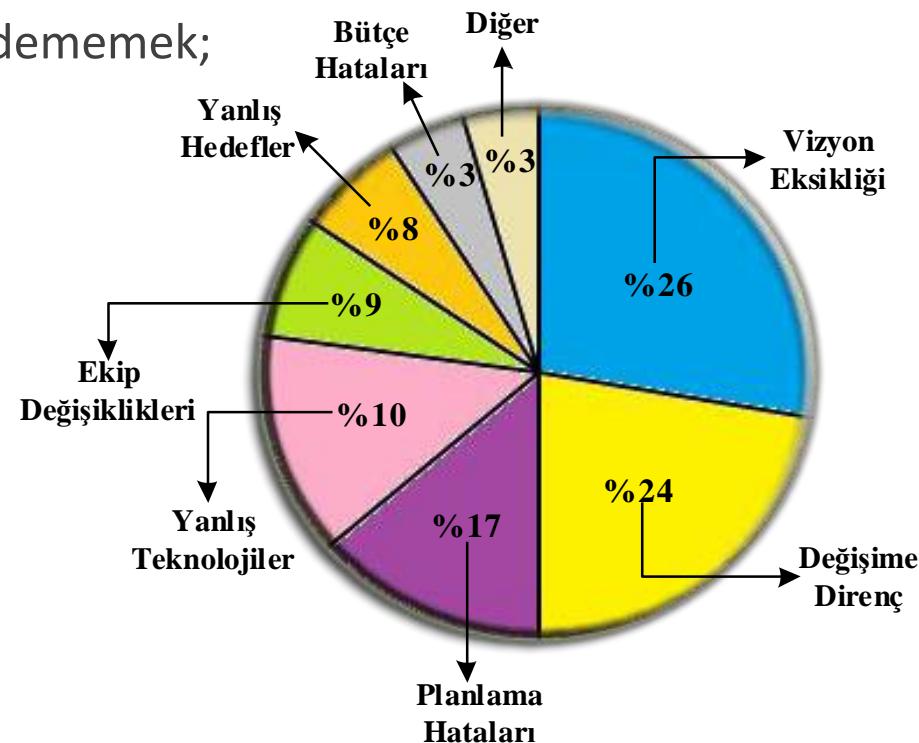
Örneğin, ülkemizde geliştirilen projelerin başarıya ulaşmasına katkı sağlamak amacıyla, Araştırma Destek Programları Başkanlığı (ARDEB) tarafından desteklenen projelerin çıktı, sonuç ve etkilerini nicelik ve nitelik olarak artırmak amacıyla yüksek başarı ile sonuçlanan projelerin yürütücü ve araştırmacılarını ödüllendirmek için TÜBİTAK tarafından belirlenen ölçütler ve değerlendirme yöntemine göre hesaplanarak, proje ekibine (yürüttü ve araştırmacılara) TÜBİTAK Proje Performans Ödülü (PPÖ), denilen bir teşvik ödülü verilmektedir.

1.Günümüzde Yazılım Projelerinin Durumu



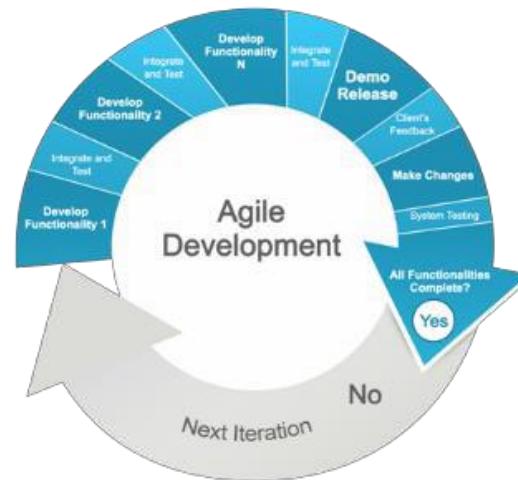
➤ Başarısızlığın ana sebepleri:

- 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

Bireylerle Etkileşim



Süreç Ve Araçlar

Çalışan Bir Yazılım



Detaylı Belgelendirme

Müşterilerle İşbirliği



Sözleşmedeki
Kurallar

Değişikliklere Uyum
Sağlama



Belirli Bir Plan

Çok Önemli

Az Önemli

➤ **Tekrarlamalı** ve **artımsal** bir ürün geliştirme yöntemidir.

■ **Bireyler ve etkileşim i**, süreç ve araca tercih eder.

■ **Çalışan bir yazılımı**, detaylı ürün belgelendirmeye tercih eder.

■ **Müşteri ile işbirliğini**, sözleşmedeki kesin kurallara tercih eder.

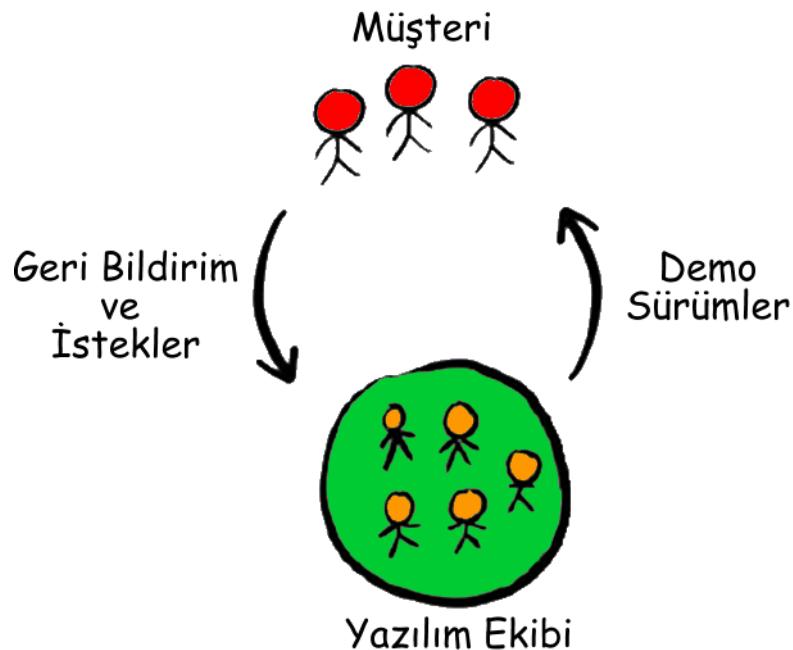
■ **Değişikliklere sağlayabilm** **uyum** belirli bir plana **tercih** eder.

2.Çevik Yazılım Yöntem



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

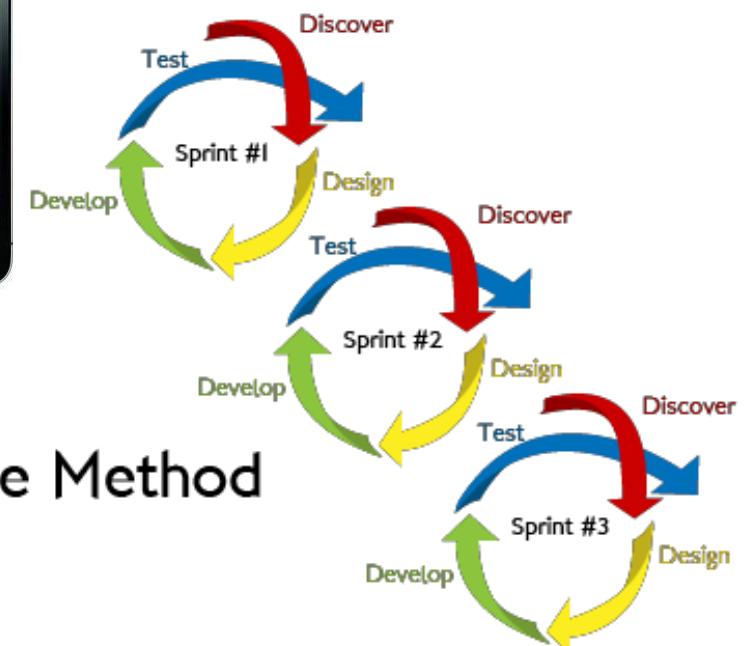
2.Çevik Yazılım Yöntemi



➤ 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:

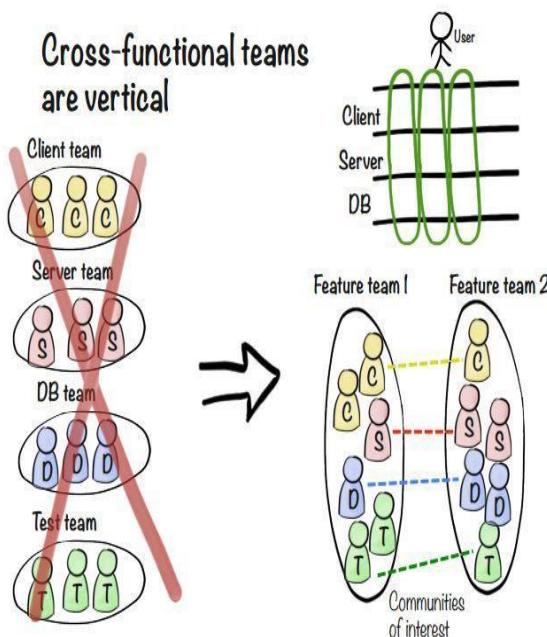


2.Çevik Yazılım Yöntemi

- 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

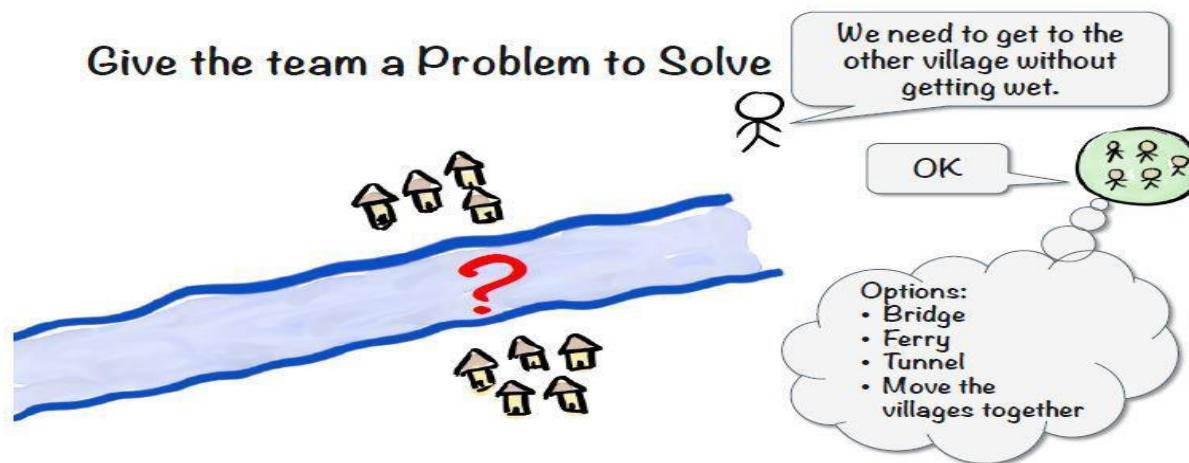
Don't give the team a Solution to Build



Takımlara çözümü söylemeyin!

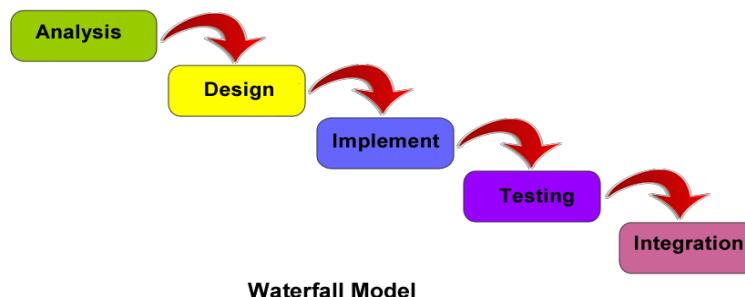
2.1.Çevik Model Takımları

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.



3. Geleneksel Model vs. Agile



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

3. Geleneksel Model vs. Agile

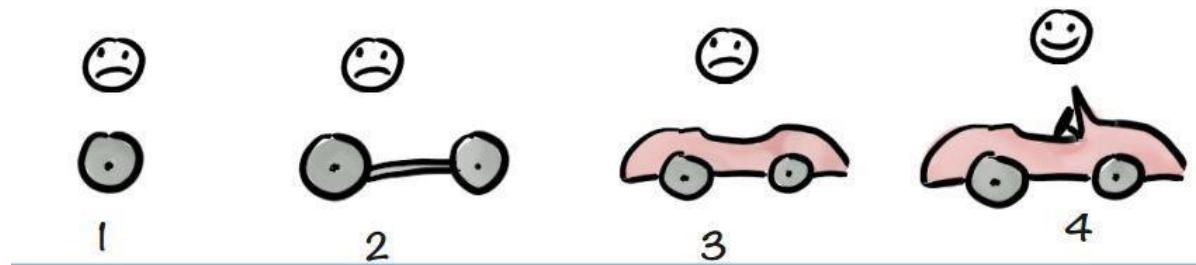
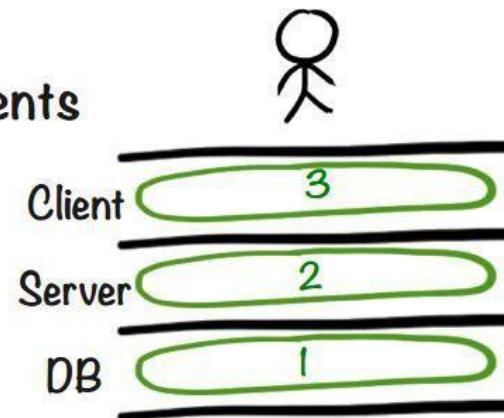
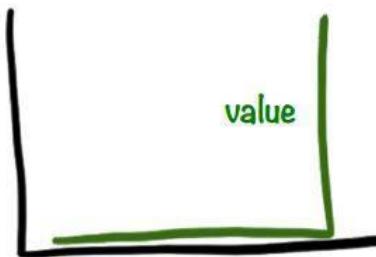


- Çevik Yöntemler
- Müşteriler ne istediğini keşfeder.
- Geliştiriciler neyi nasıl üreteceğini keşfeder.
- Bu yol boyunca bir çok değişiklik yapılabilir.

3. Geleneksel Model vs. Agile

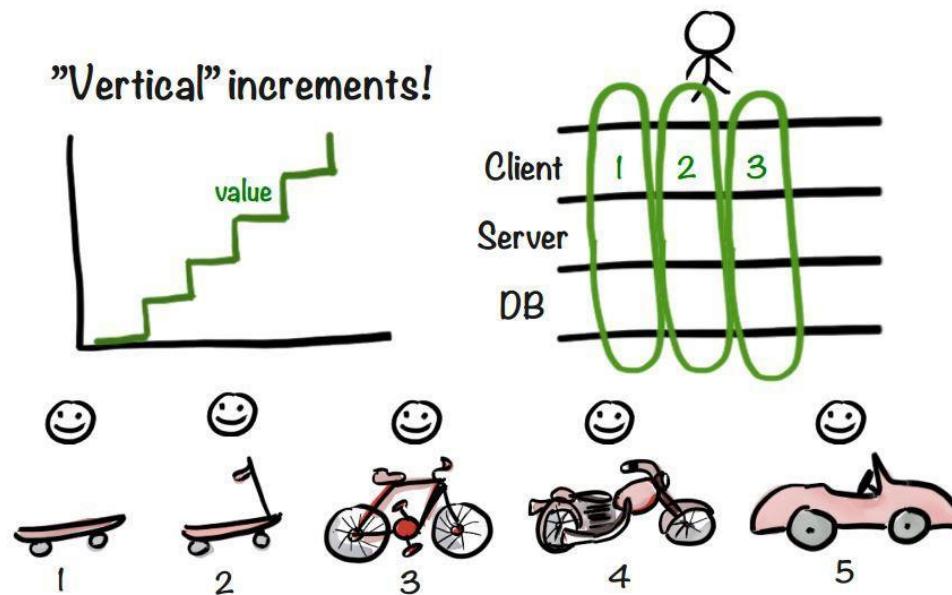
Geleneksel Yöntemler

Not "horizontal" increments

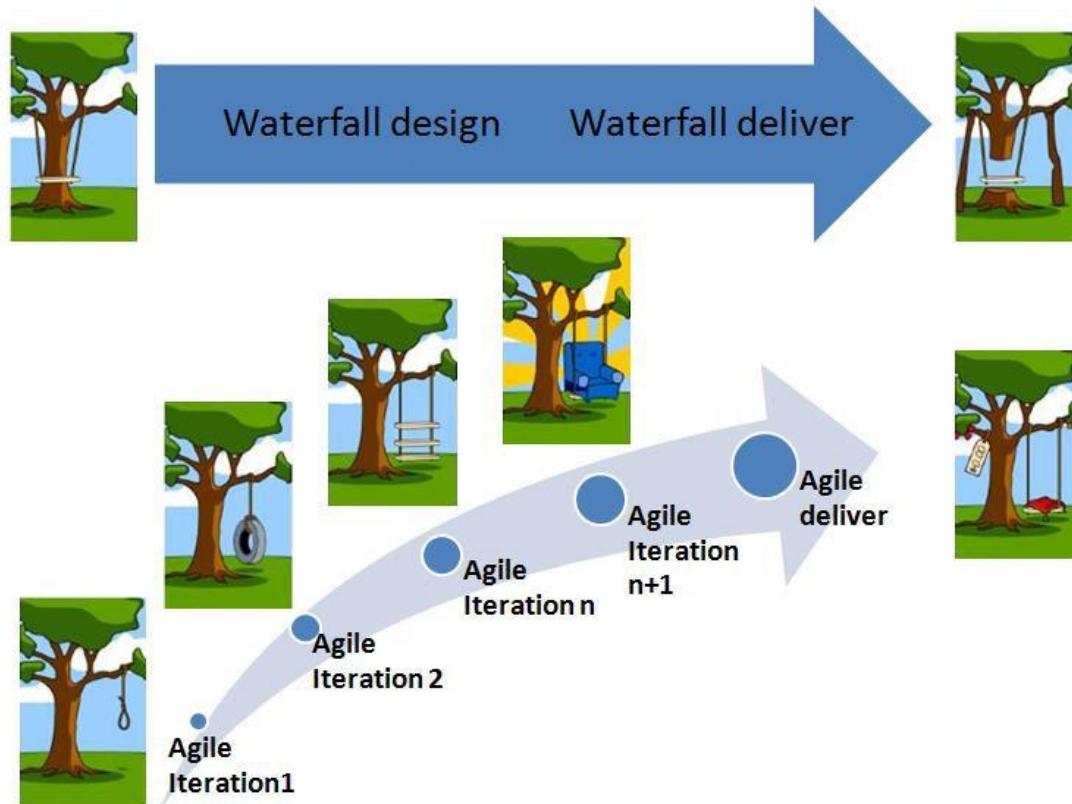


3. Geleneksel Model vs. Agile

Çevik Yöntemler

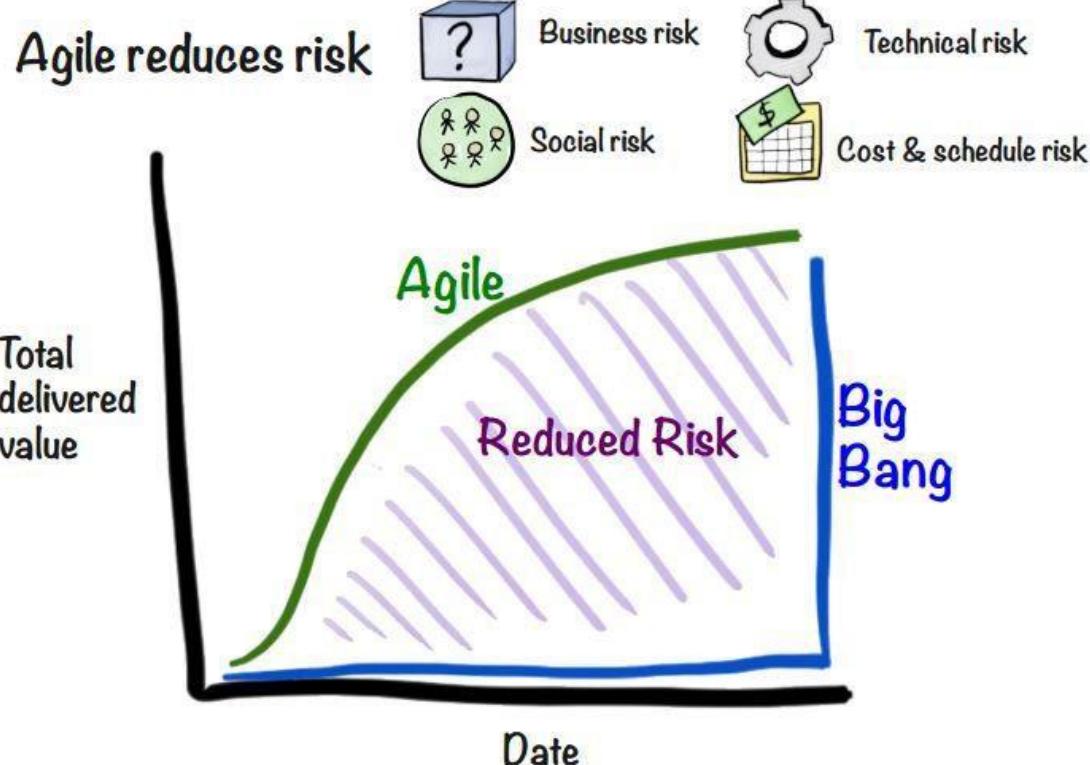


3. Geleneksel Model vs. Agile

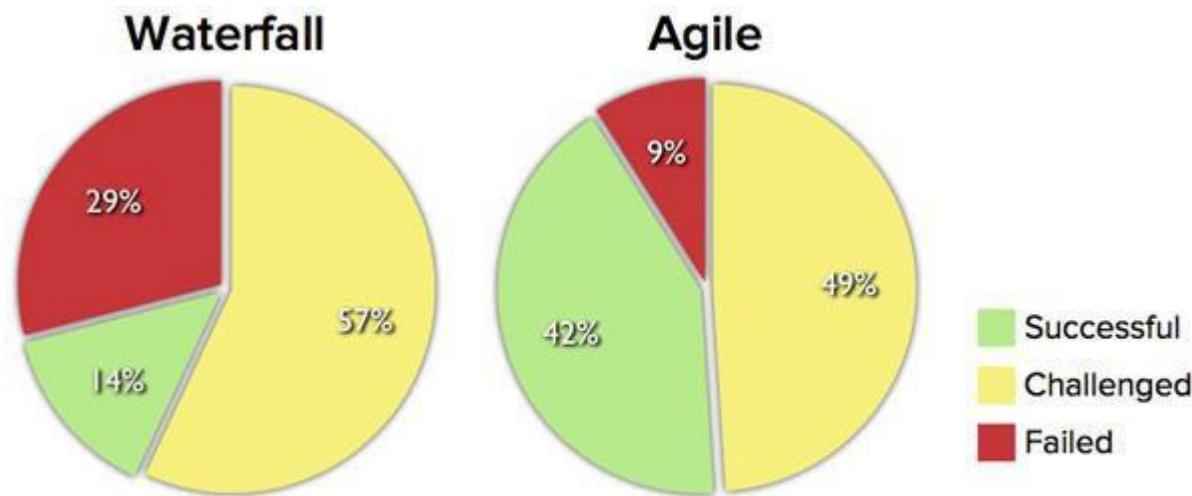


3. Geleneksel Model vs. Agile

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

Büyük Projeler



Büyük Özellikler



Büyük Takımlar



Büyük Dönüşümler



Sonuç

➤ Büyüklük kötüdür, onu parçalara ayırin. Böylece daha başarılı projeler geliştirebilirsiniz.

4. Değerlendirme

➤ 3 somut değişiklik yapın!

➤ 1. Gerçek takımlar oluşturun

- Küçük, çapraz fonksiyonlu, kendi kendine organize olabilen

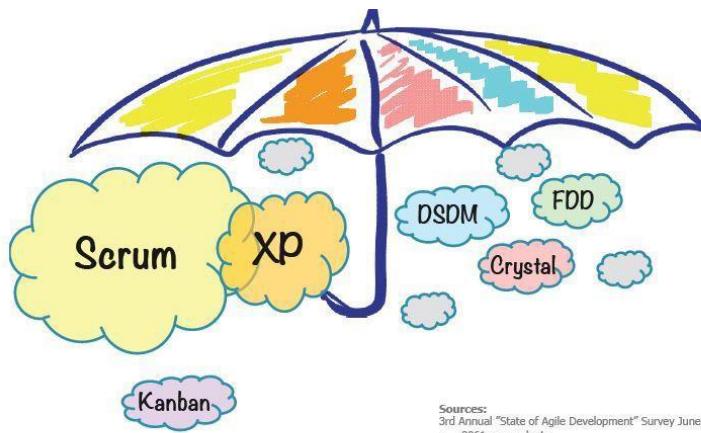
2. Sık sık teslimat yapın

- 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



FDD: Feature-Driven Development

RUP : Rational Unified Process

DSDM : Dynamic System Development Method

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

FDD TANIMI

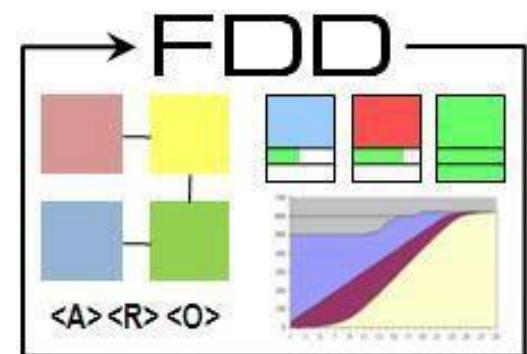
➤ FDD, Avustralyalı Jeff De Luca tarafından geliştirilmiş ve Singapur projesinde beraber çalıştıkları Peter Coad tarafından modifiye edilmiştir. Singapur projesi 50 kişi ile 15 ayda tamamlanan bir proje olmuştur. Daha sonra FDD'nin uygulandığı başka bir proje ise 250 kişi ile 18 ayda tamamlanmıştır. FDD değişik boyutlara büyüyebilen, tekrarlanabilir bir süreçtir. Aşağıdaki noktalara odaklanır.

1. Sistemi hazırlamak için gereken sistem büyüyebilir olmalıdır. Büyük projeler için de kullanılabilir olmalıdır.

2. Basit iyi tanımlanmış sistem iyi çalışır.

3. Süreç adımları basit olmalıdır.

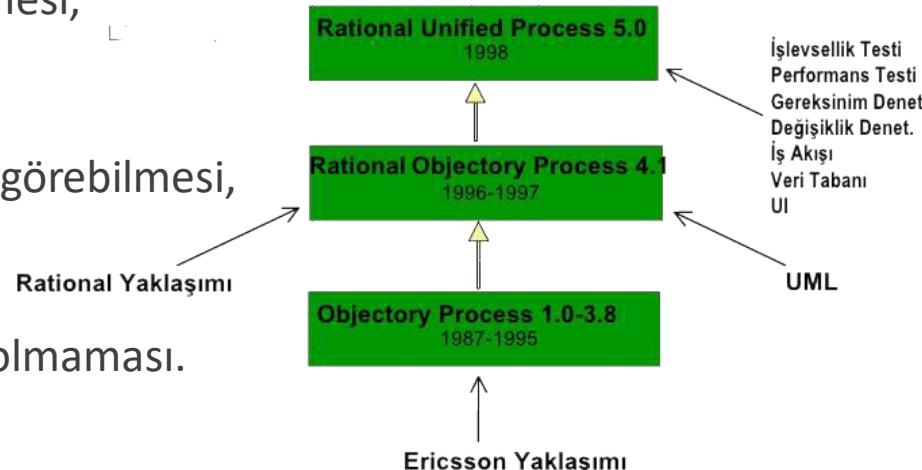
4. İyi süreç arka plana saklanır ve insanlar sonuçlara odaklanabilir.



5. Kısa, iteratif, özellik yaklaşımı yaşam döngüleri en iyi sonucu verir

RUP ("Rational Unified Process")

- 2003 yılından beri IBM'in bir bölümü tarafından oluşturulan bir iteratif yazılım geliştirme süreci çerçevesidir. Başarısız bir yazılımdaki sorunların aşılıp başarılı yazılım oluşturmak için gerekli adımları saptayarak oluşturulmuş bir süreçtir.
 - Başarısız bir yazılımdaki özelliklerini yazımızın devamında okuyabilirsiniz. RUP şirketlere yazılım geliştirme aşamasında bir yön sağlar. RUP use-case ve teknolojileri tabanlı; tekrarlanan (iterative) yazılım geliştirme ve ış modelleme yöntemidir. RUP'un verebileceği özellikler şunlardır;
- RUP'un Gelişimi**
1. Müşteriyi ve yazılımcıyı organize edebilmesi,
 2. Standart tanımlı adımları olması,
 3. Oluşacak yazılımdaki sık değişiklikleri öngörebilmesi,
 4. Basit olması,
 5. Proje yönetim aktivitelerinin çok fazla olmaması.



Uç Programlama (Extreme Programming XP) Nedir?

- Uç Programlama (XP), yazılım geliştirme süreci boyunca son derece kaliteli olmak koşuluyla çalıştırılabilir kod üretmeye odaklanmış bir yazılım geliştirme metodolojisidir. Yazılım geliştirme sürecinin en temel, en önemli ve final çıktısı ya da ürünü çalıştırılabilir kod olduğundan, XP metodolojisi sürecin en başından itibaren çalıştırılabilir kodu sürecin merkezinde tutmaktadır. İşte bu yüzden bu metodolojinin adı XP'dir.

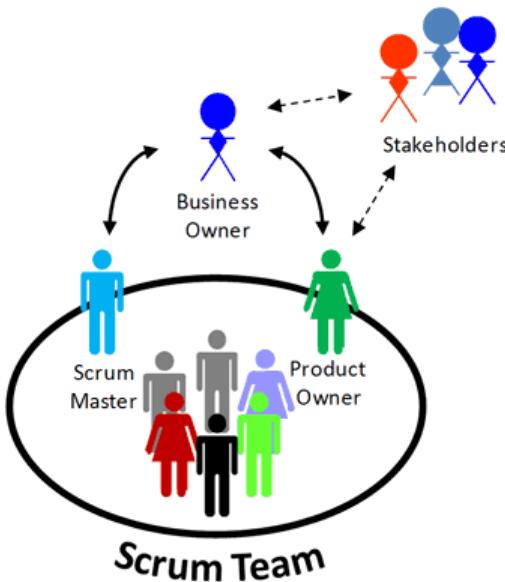
Uç Programlama'nın özünde aşağıdaki uygulamalar yer alır:

- Sık ve küçük sürümler
- Basit tasarım:
- Önce test
- Haftada 40 saat çalışma
- Müşteriyle yakın iletişim
- Kodlama standartları

6. Scrum Modeli

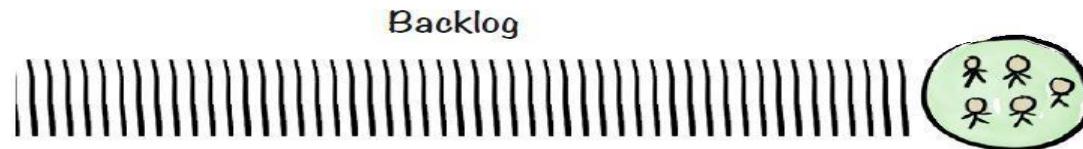


6. Scrum Modeli



➤ **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.

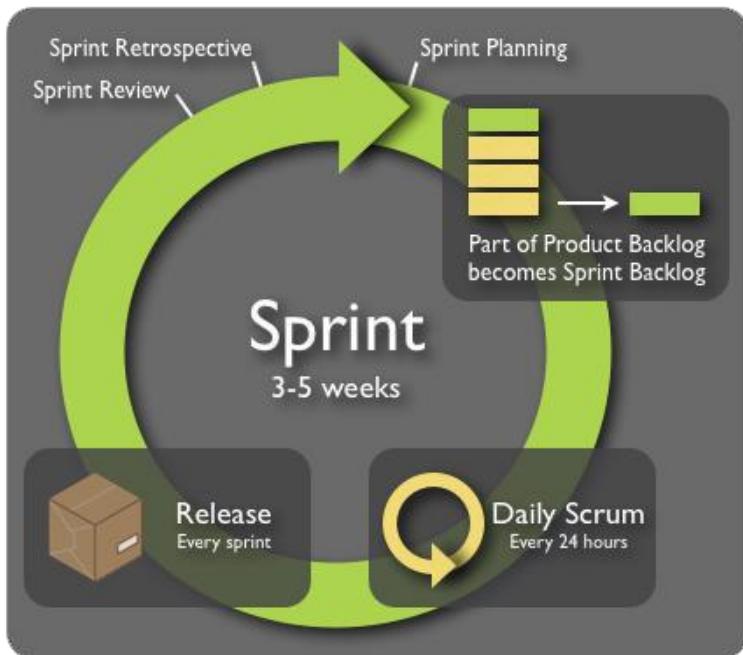
6. Scrum Modeli



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 kavamlara göre ürün sahibi tarafından sıralandırılır.
- User Story'lerden oluşur.

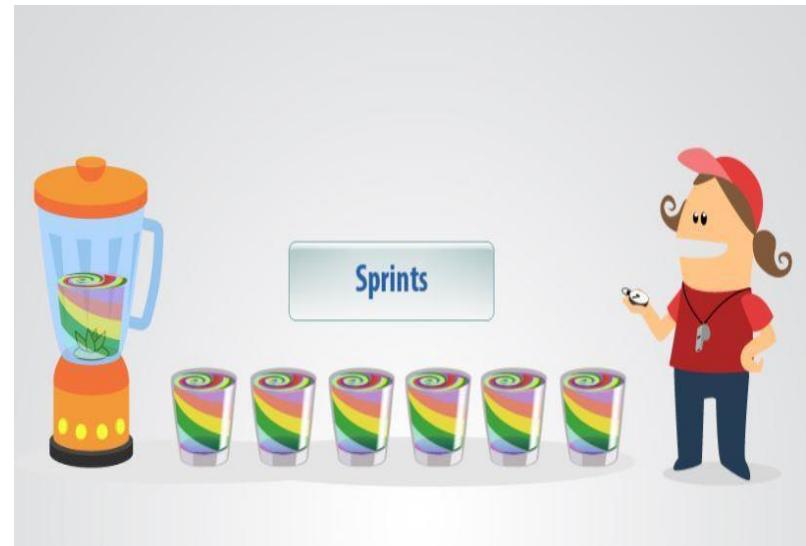
6. Scrum Modeli



➤ Sprint

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

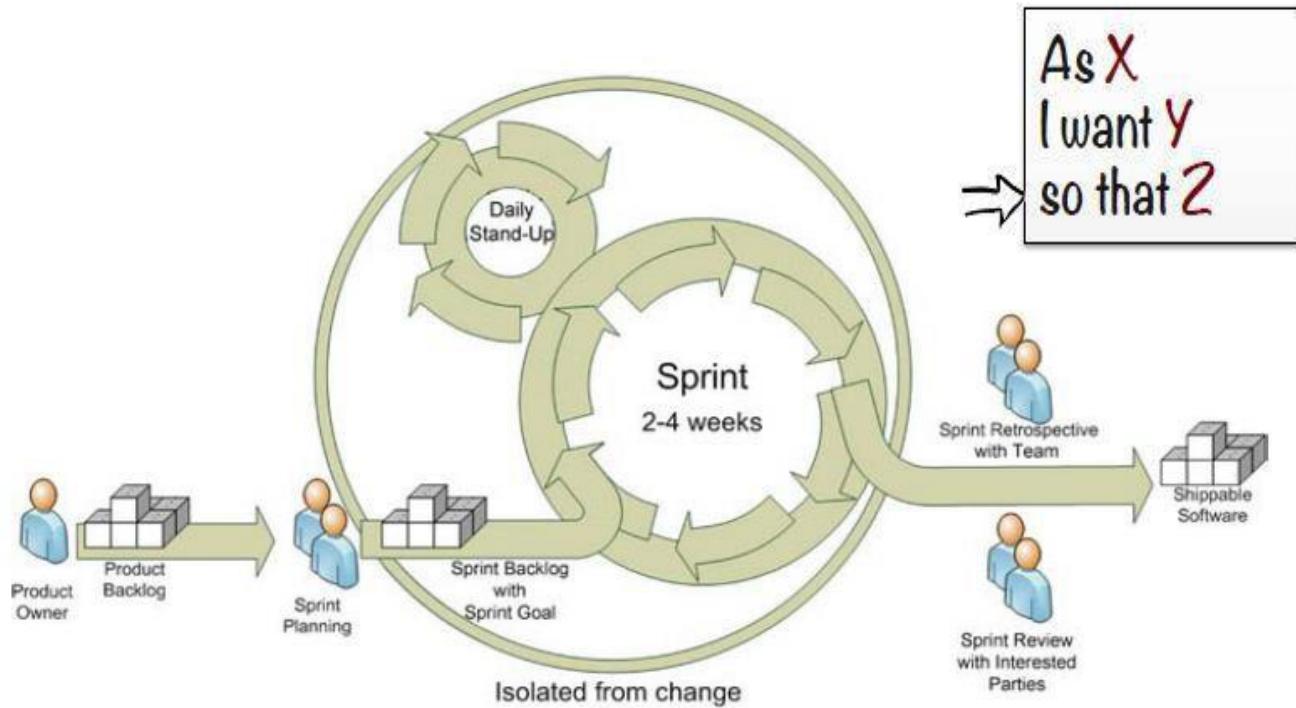
6. Scrum Modeli



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.



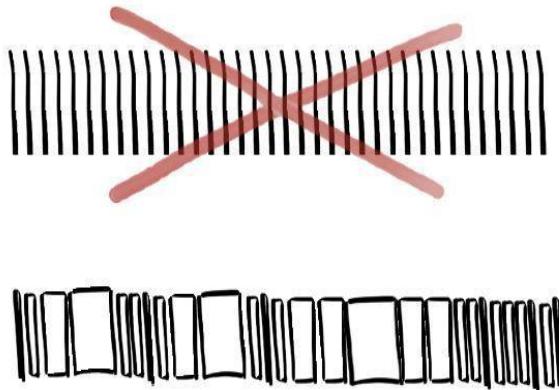
6. Scrum Modeli

➤ **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

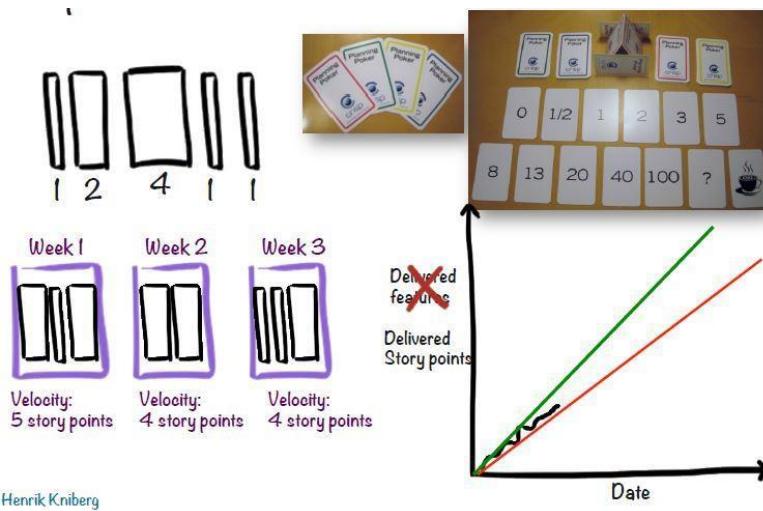
Fact: Features have different sizes



➤ 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 diğer 5 user story içerebilir. **Peki boyutları nasıl belirleyeceğiz?**

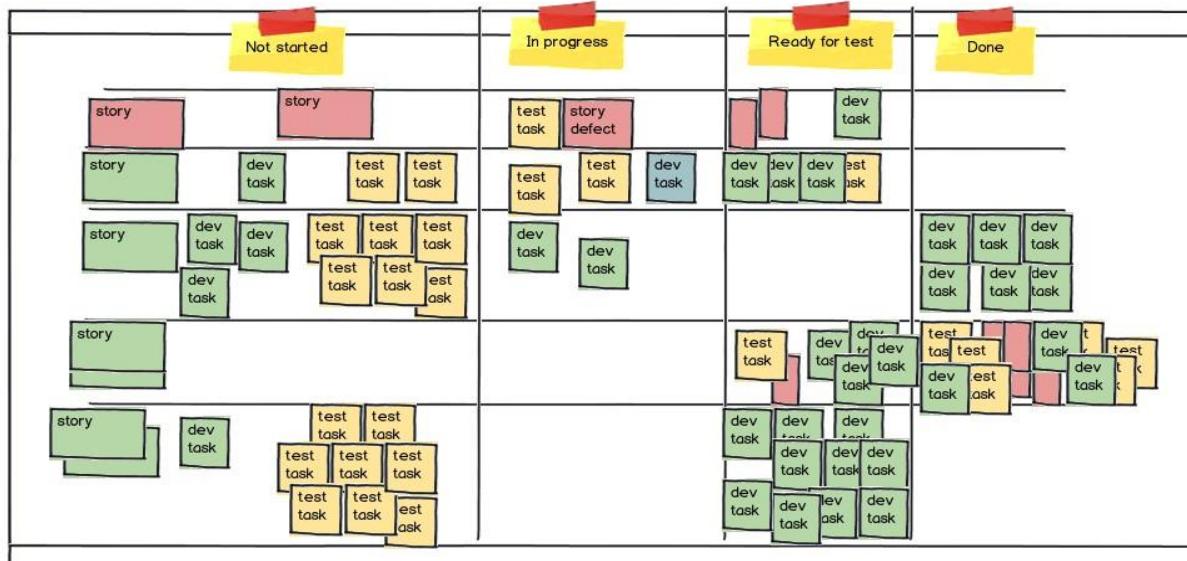
İş göründüğü gibi değil!

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 seçikten sonra tüm kartlar açılır ve değerlendirilir. Böylece herkesin ortak görüşü sonunda user story'lerin büyüklüğü belirlenir.

6. Scrum Modeli



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)

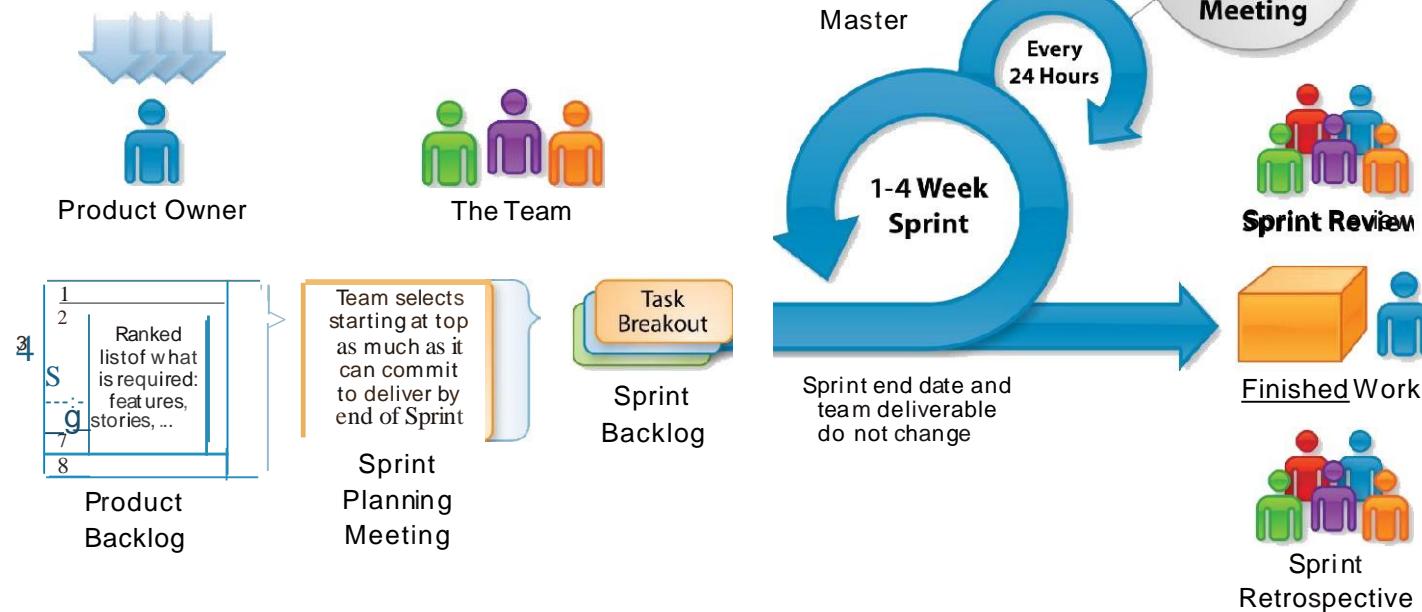
- Sprint (1 yada 2 hafta)
- Sprintler değişebilir.
- Özellikler ürün sahibi tarafından derecelendirilir.
- Mühendislik pratikleri tanımlar. Eşli programlama, otomatik test, basit dizayn vs.

Early delivery of business value

less bureaucracy

The Agile: Serum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Çalışma Soruları

1. Agile modelinin geliştirilmesine neden ihtiyaç duyulmuştur? Açıklayınız.
2. Agile takımları kimlerden oluşur ve iş paylaşımıları 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
- [5] <http://www.mshowto.org/microsoft-visual-studio-team-foundation-server-nedir.html>
- [6] www.kurumsaljava.com/download/10/
- [7] <https://www.tubitak.gov.tr/tr/destekler/akademik/uygulamalar-ve-yonergeler/icerik-proje-performans-odulu-ppo-uygulamasi>
- [8] http://images.slideplayer.biz.tr/8/2395426/slides/slide_27.jpg
- [9] <http://www.bayramucuncu.com/wp-content/uploads/2013/04/Ads%C4%B1z.png>
- [10] <http://antasya.com/Images/is-alanlarimiz/yazilim-sistemleri/agile-method.png>
- [11] <http://volkansel.com/wp-content/uploads/2014/07/agile-scrum.jpg>

İçerik

Yazılım Proje Yönetimi

Agile Nedir?

Agile (Çevik) yaklaşım ile Scrum yöntemi

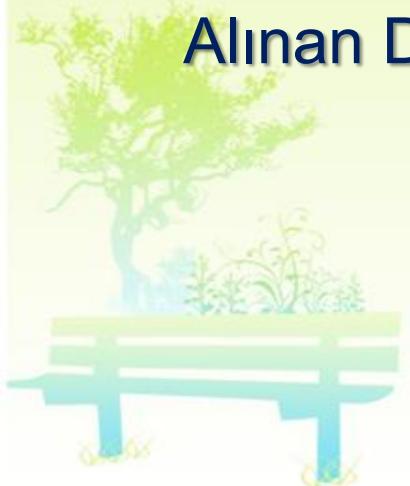
Scrum yöntemi

Scrum Kullanan Firma Anket Değerlendirmeleri

Scrum Proje Örnekleri

Özet ,Sonuçlar

Alınan Dersler ve Geliştirilecek Alanlar

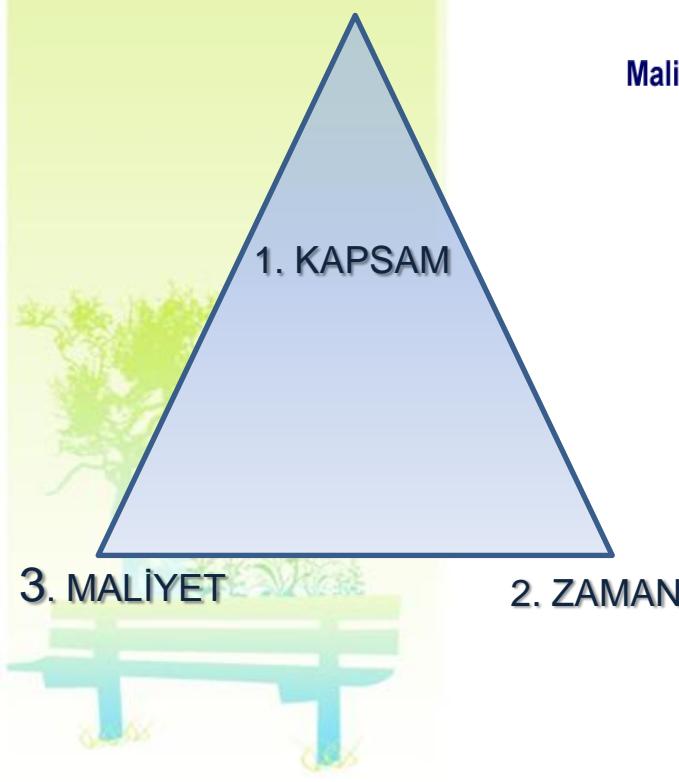


AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

Yazılım Proje Yönetimi

Uzun yıllardır proje yönetimi ile ilgili farklı yöntemler geliştirilmektedir. Özellikle 1990 yılından sonra yazılım proje yönetiminin önemi daha çok anlaşılmıştır. Yazılım Projelerinin 3 ana unsuru vardır.

- KALİTE



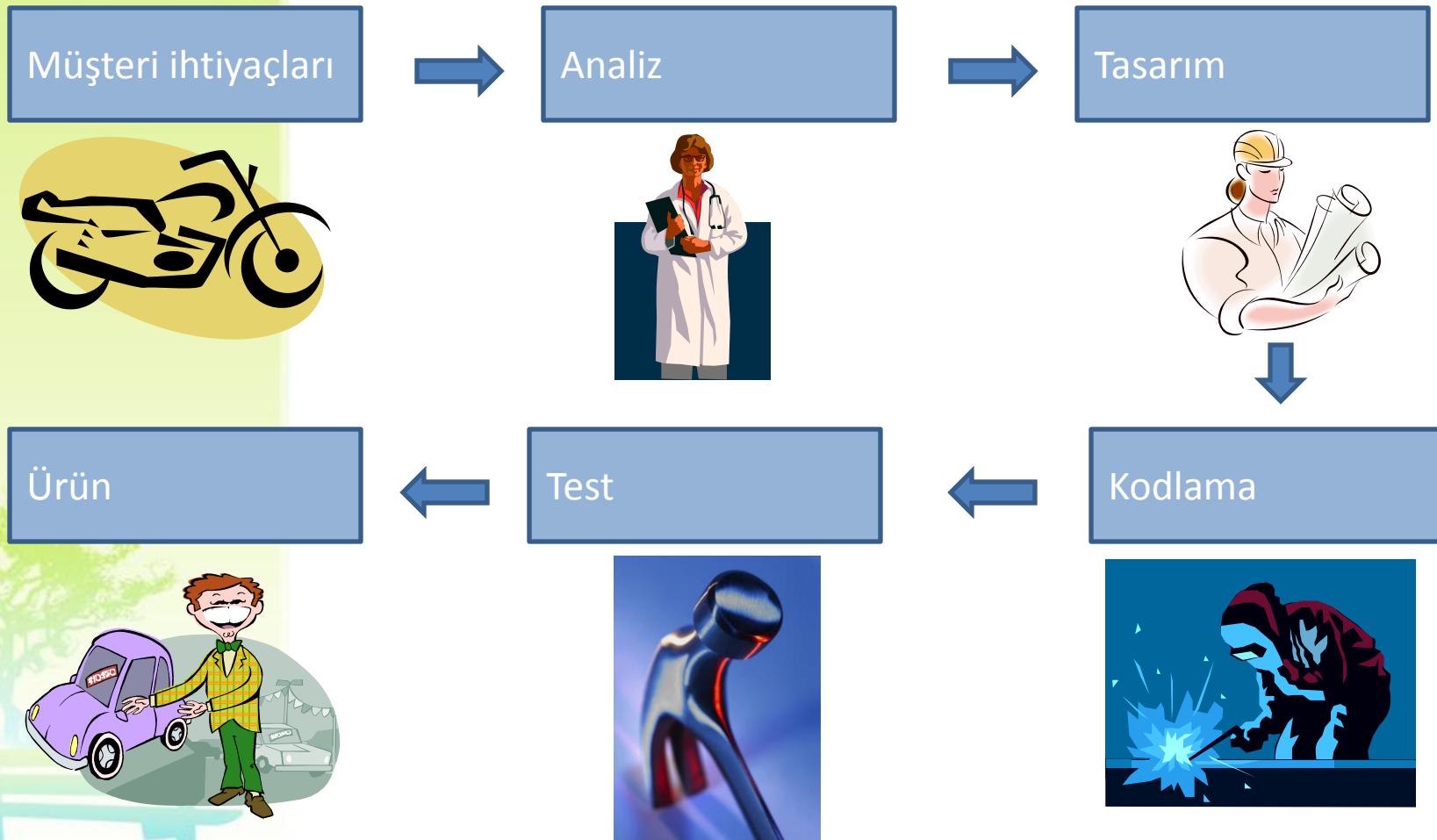
Maliyet ve Personel



Proje yaşam sürecinde zaman ve maliyet tablosu aşağıdaki gibidir.

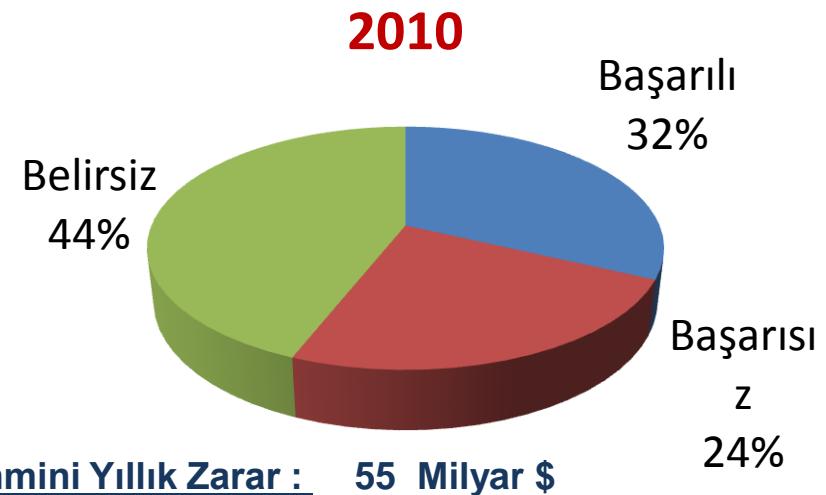
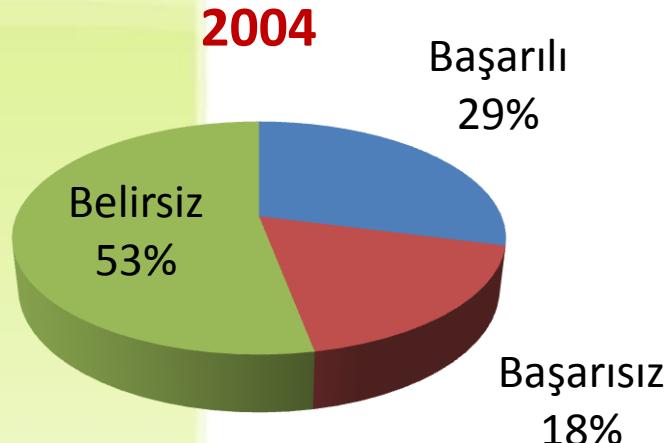
Yazılım Proje Yönetimi

Yazılım projelerinin ana katmanları

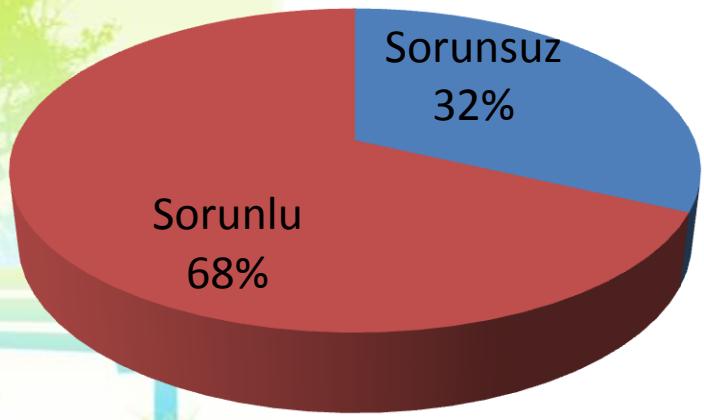


Yazılım Proje Yönetimi

Standish Group Chaos raporları;



Proje Yönetimi açısından;



Yazılım Proje Yönetimi

Başarısızlık dağılımı



SÜRE

%62



BÜTÇE

%49



BAKIM

%47



KATMA
DEĞER (RIO)

%41



BEKLENTİ

%33



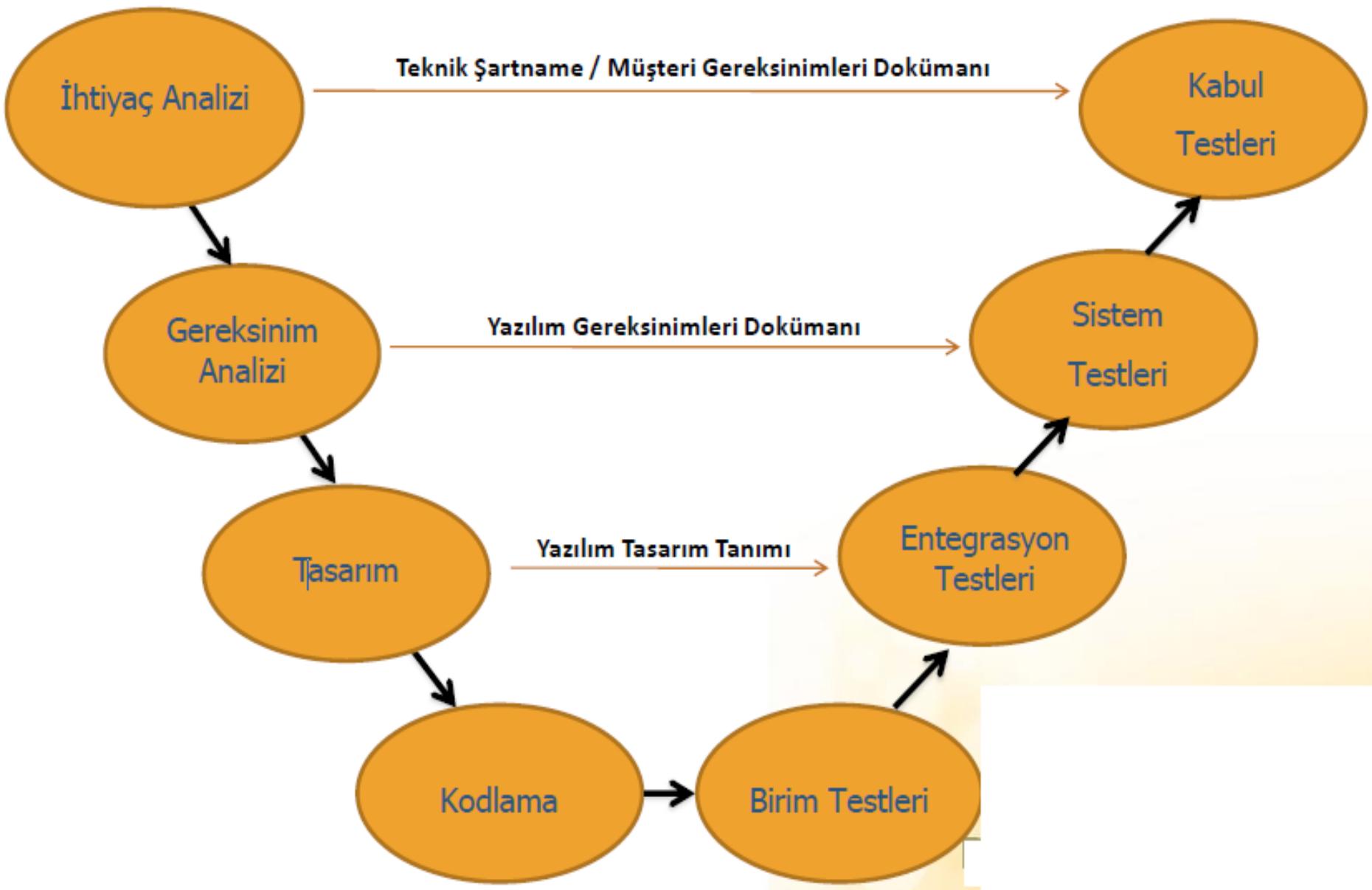
Başarısızlığın ana sebepleri :

- 1- Müşterinin isteklerini doğru analiz edememek.
- 2- Proje için uygun ekibi kurmamak / kuramamak.
- 3- Gerekli bütçe ve kaynakları ayırmamak.
- 4- Proje Yönetim metotları uygulamadan, gelişî güzel geliştirmek.
- 5- Proje süresince müşteri ile iletişimden kaçınmak.
- 6- Yanlış teknoloji ya da mimari seçimleri.
- 7- Şirketin yönetimsel sorunları.

Ülkemizde durum nasıl ? Başarılı Proje Durumu

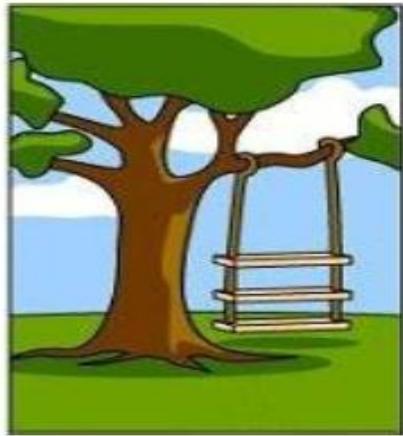
Durum	Oran
Tam Başarılı	% 4-5
Kısmen Başarılı	% 45-50
Çöpe Gidenler	% 45-50

İş Ürünleri - Proje Sonucu İlişkisi



Kontrol Edilmeyen Proje

Kullanıcının istediği :



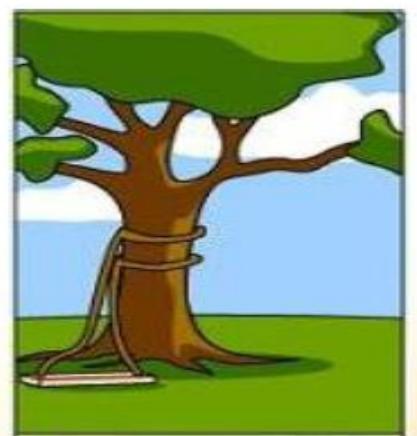
Analiz sonrasında :



Tasarım sonrasında :



Kodlama sonrasında :



Kontrollerin Gerçekleştirildiği Proje

Kullanıcının istediği :

Kullanıcının asıl ihtiyacı : Analiz sonrası :



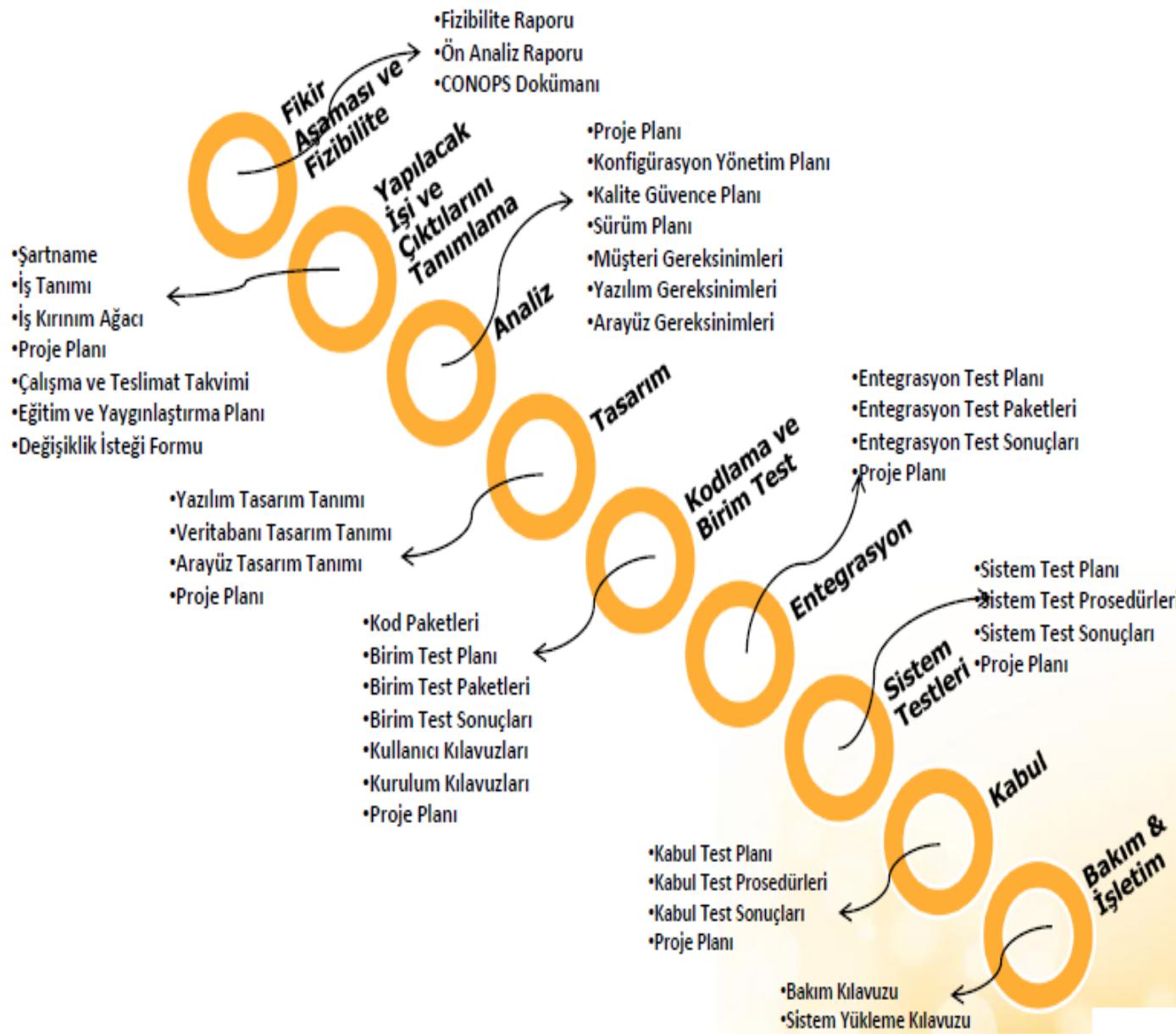
Tasarım sonrası :

Kodlama sonrası :



I'll know it when I see it!

Projenin Görünür Çıktılarının Tanımlanması



Yazılım Projelerinde Başarı Durumu- Başarısızlık Nedenleri



İşin kapsamının ve iş ürünlerinin tanımlı olmaması

(1)

Uygulanması gereken süreçlerin (Proje Yönetimi ve İzleme ile diğer süreçler) tanımlı olmaması

(2)



Yazılım ortaya çıkmadan İdare'nin değerlendirme yapamaması

(3)

Yazılım organizasyonlarının personel ve tecrübe eksikliği



Proje Gelişiminin İzlenebilmesi

- İşlerin görünürlüğü ve izlenebilirliğinin sağlanması için planlama yapılmalı ve plan izlenmelidir.

Proje Planı = f (Yazılım Geliştirme Yaşam Döngüsü, Projeye Özgү Parametreler)

- Projeye özgü parametreler
 - ❖ İşin Büyüklüğü (Adam Ay)
 - ❖ Bütçe
 - ❖ Zaman
 - ❖ Kaynak
 - ❖ Kapsam
- Yazılım Geliştirme Yaşam Döngüsü
 - ❖ İzlenecek süreçlerin sırası, etkileşimi ve çıktıları

Agile Nedir ?



- Proje Yönetim Biçimi veya frameworkü denilebilir.
- En yüksek iş değerini en kısa sürede elde etmeye odaklanır.
- Takımla beraber yapılan ürün geliştirme projelerinde çok başarılı olmuş
- En yaygın kullanım alanı ise yazılım geliştirme projeleridir.
- Çok kapsamlı olmayan ve belirsizliğin çok fazla olduğu projeler için çok kullanışlı bir yöntemdir.
- İhtiyaçların tam olarak belirlenemediği projelerde sık rastlanır.
- Müşteri ile Proje takımının esnek ilişkiler içerisinde olabilmesi kabulü ön şarttır.
- Proje Yönetimi ile Yürütme bir arada.

AGILE Prensibleri

- Bireyler ve arasındaki etkileşim, kullanılan süreç ve araçlardan daha önemlidir
- Portatif ürün, anlaşılır dökümantasyondan daha önemlidir
- Müşteri ile ilişki, müşteri sözleşmesinde yazanlardan daha önemlidir
- Değişime adapte olmak, yapılan plana ne olursa olsun bağlı olmaktan daha önemlidir
- Projeleri iterasyonlarla aşamalı olarak geliştirmeyi öngörür.
- Amacı çok kısa döngülerle, sık çıktılar üretmektir.
- Kaynağı müşteri ihtiyaçlarına ve sonuca kanalize etmeye odaklanır.
- Kalite tarifi müşterinin üründen bekłentileri karşılamasıdır.
- Risk iterasyonlardan ve maksimum etkileşimden dolayı minimumdur.

AGILE Neden ortaya çıktı ?

- Gartner Institute'un BT sektörü araştırmasına göre: BT projelerinin %74'ü başarısız ya da maliyet/zaman hedeflerini aşıyor. BT projelerinin %51'i bütçesini %200 oranında aşıyor ve hedeflenen özelliklerin %75'ini karşılayabiliyor.

Standish grubun 2000 yılında gerçekleştirdiği bir araştırmaya (Chaos in the new Millenium 2000) göre yazılım projelerinin başarıya ulaşma oranı %28 olarak veriliyor. Diğerleri ya başarısız (%23) ya da zorlanmış (%49) projelerdir. Aynı araştırma yazılım projeleri özelinde de proje maliyetlerinin tahmin edilenin üzerinde olduğu veya zaman aşımı olduğu ya da niteliklerin istenilene tam uygun olmadığını gösteriyor.

- Gartner Group'un (Technowledge SM 99 Presentation) yapmış olduğu bir araştırmaya göre BT projelerinin %70'i beklenen faydayı sağlamıyor.

Gartner Institute'un 2001 BT sektörü araştırmasına göre: Amerika'da her yıl başarısız BT projeleri için 75 milyar dolar harcanıyor .

Agile yazılım geliştirme

- İş çevrelerinin hızla değişmesinden dolayı şirketler yeni fırsatlara ve rekabete cevap verebilmelidir
- Bu yazılımı ve çabuk geliştirmeyi gerektirir, ancak teslimat yazılım sistemleri için her zaman en kritik gereksinim değildir
- Şirketler eğer yazılımın temel fonksiyonlarıyla çabuk teslimatı mümkünse düşük kaliteli yazılıma razı olabilirler

Gereksinimler

- Değişen çevreden dolayı, genellikle kararlı, istikrarlı sistem gereksinimi ayarlarına ulaşmak mümkün değildir
- O yüzden waterfall modeli kullanılan geliştirme elverisizdir, tekrarlanan tanımlama ve teslimat geliştirme tabanlı yaklaşım yazılımı hızlı teslim etmenin tek yoludur.

Özellikleri :

- Tanımlama (specification), tasarım (design) ve gerçekleme (implementation) süreçleri eş zamanlıdır. Detaylı tanımlama yoktur ve tasarım dokümantasyonu minimize edilmiştir.
- Sistem bir sıra ekleme şeklinde geliştirilir. Son kullanıcılar her artışı dener ve daha sonraki ekleme için teklifte bulunurlar.
- Sistem kullanıcı ara yüzü genellikle bir etkileşimli geliştirme sistemi kullanılarak geliştirilir.

30,000 Teams
30 Fortune 100 Companies



Alcatel-Lucent The Alcatel-Lucent logo features a purple circle containing a white stylized 'A'.

Boeing

NASDAQ

SIEMENS

Lilly

ORACLE

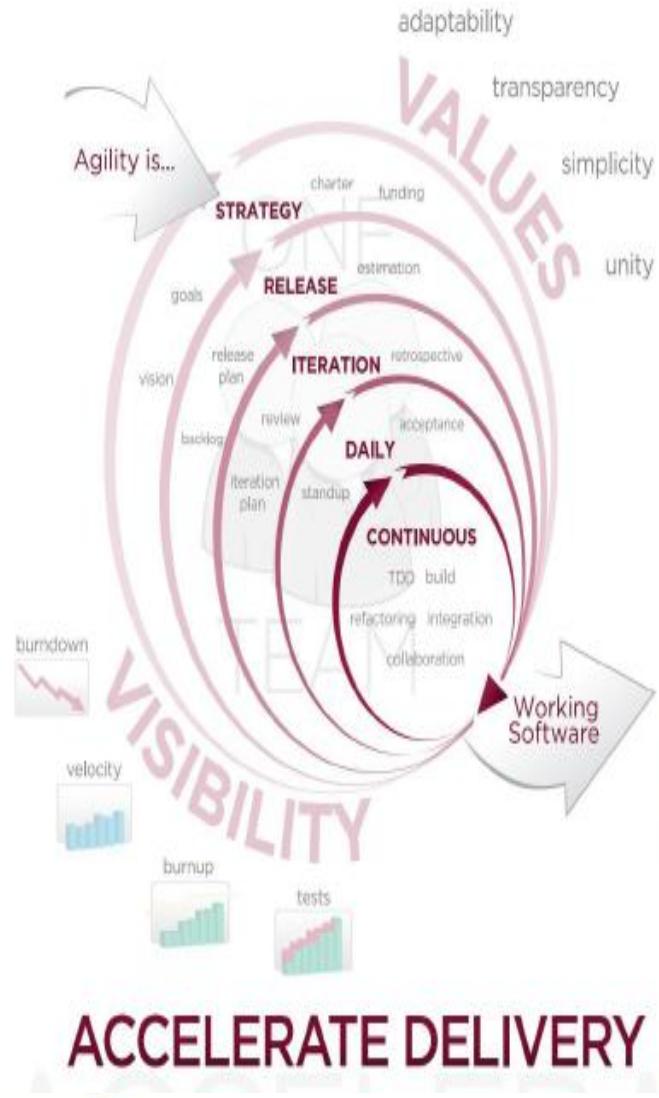
HUMANA
Guidance where you need it most

Çevik (Agile) Manifesto

Önemli	Az Önemli
Kişiler ve takım çalışması	Süreç ve araçlardan
Çalışan yazılım	Detaylı dokümantasyondan
Müşteri ile beraber çalışmak	Sözleşme ve anlaşmalardan
Değişime açık olup, uygulamak	Plana bağlı kalmaktan

Çevik Yazılım Geliştirme

AGILE DEVELOPMENT



Farklı Agile Frameworkler

- Bu frameworkler agile prensiblerini benimsemiş farklı çalışma şekilleridir.
- Scrum
- Extreme Programming
- Lean Software Development
- Feature Driven Development
- Dynamic Systems Development Method

Agile (Çevik) Yaklaşım

Bir yazılım projesi, 6 ay içerisinde tamamlanmazsa ve projeye müşterinizi dahil etmezseniz, başarıya ulaşma ihtimaliniz zayıftır.



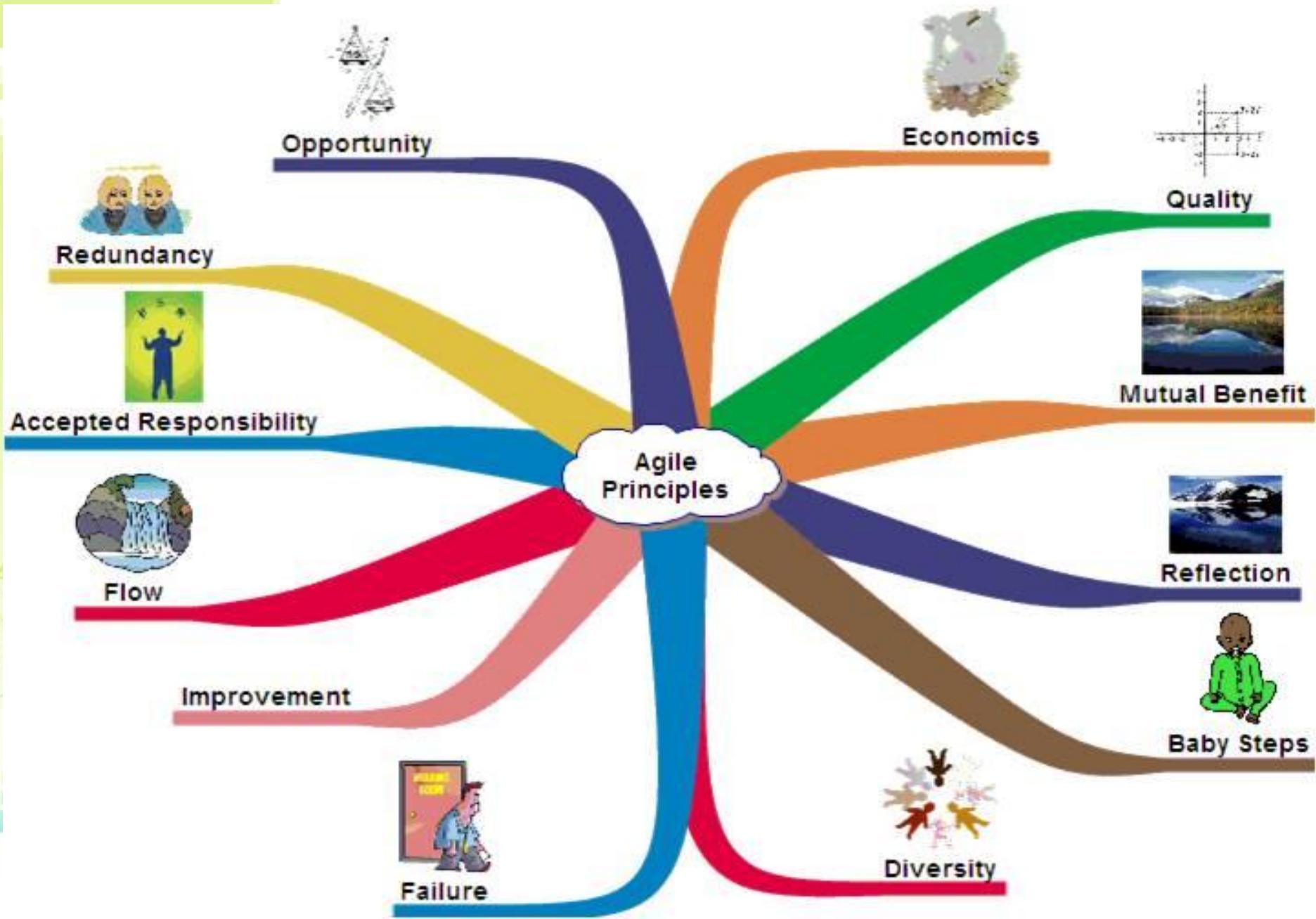
Çevik (Agile) Manifesto

2000 yılında Kent Beck ve 16 arkadaşı tarafından çevik manifesto oluşturulmuştur.

Yazılım geliştirme amacıyla üretilen bu modelleme biçimini, kapsadığı değerler, prensipler ve pratikler sayesinde geleneksel modellemelere metodlarına göre yazılımlara daha esnek ve kullanışlı biçimde uygulanabilir.



Agile (Çevik) Yazılımın Prensipleri



Agile (Çevik) Yazılımın Prensipleri:

- 1- İlk öncelik, sürekli, kaliteli yazılım teslimatıyla müşteri memnuniyetini sağlamaktır.
- 2- Proje ne kadar ilerlemiş olursa olsun değişiklikler kabul edilir. Çevik yazılım süreçleri değişiklikleri müşteri avantajına dönüştürürler.
- 3- Mمmkün olduğunda kısa zaman aralıklarıyla (2-6 hafta arası) çalışan, kaliteli yazılım teslimatı yapılır.
- 4- Analistler, uzmanlar, yazılımcılar, testçiler vs. tüm ekip elemanları bire bir iletişim halinde, birlikte çalışırlar.
- 5- İyi projeler, motivasyonu yüksek bireyler etrafında kurulur. Ekip elemanlarına gerekli destek verilmeli, ihtiyaçları karşılanarak proje ile ilgili ekiplere tam güvenilmelidir.
- 6- Ekip içerisinde kaliteli bilgi akışı için yüz yüze iletişim önemlidir.
- 7- Çalışan yazılım, projenin ilk gelişim ölçütüdür.
- 8- Çevik süreçler, mümkün olduğunda sabit hızlı, sürdürülebilir geliştirmeye önem verir.
- 9- Güçlü teknik alt yapı ve tasarım çevikliği arttırır.
- 10- Basitlik önemlidir.
- 11- En iyi mimariler, gereksinimler ve tasarımlar kendi kendini organize edebilen ekipler tarafından yaratılır.
- 12- Düzenli aralıklarla ekpler kendi yöntemlerini gözden geçirerek verimliliği artırmak için gerekli iyileştirmeleri yaparlar.

Agile (Çevik) Yaklaşım

ÇÖZÜM

Müşterini projeye dahil et



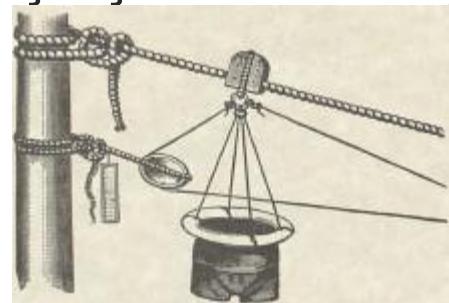
En önemli özellikleri ilk önce yap



Geri bildirimlere göre yeniden düzenle



Mümkün olan en kısa sürede çalışan bir sürüm çıkar



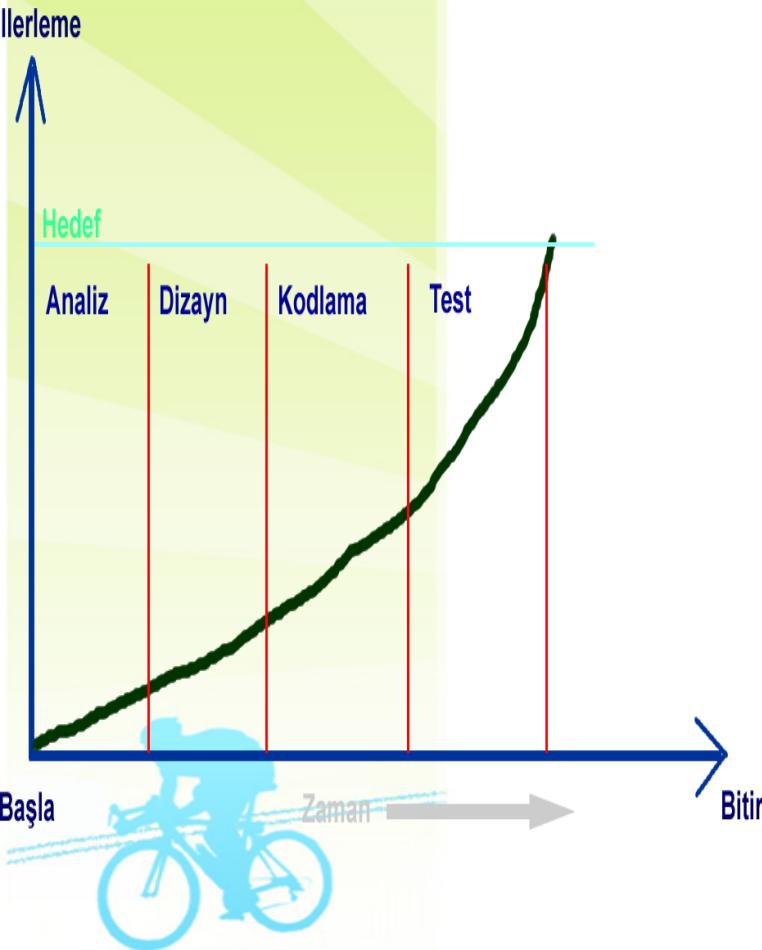
Müşteri ile değerlendir



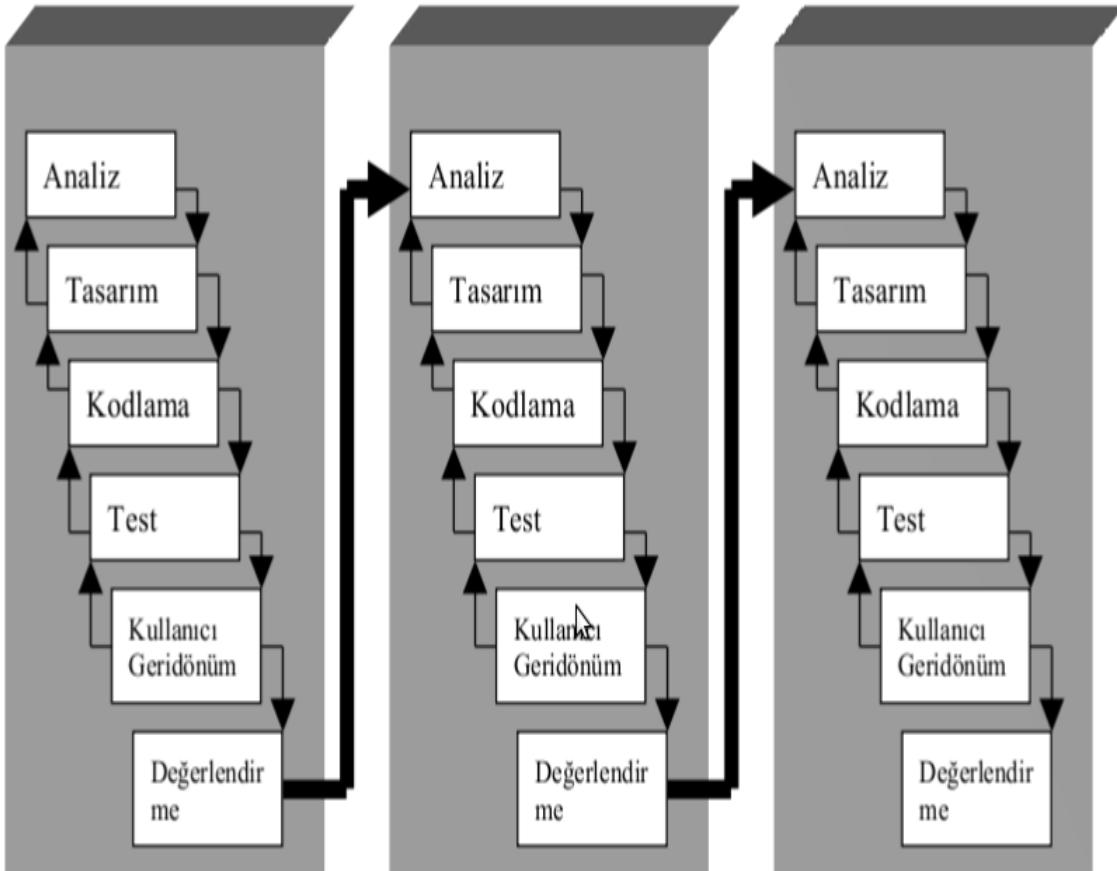
AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

Agile (Çevik) Yaklaşım

Klasik Yaklaşım



Çevik Yaklaşım



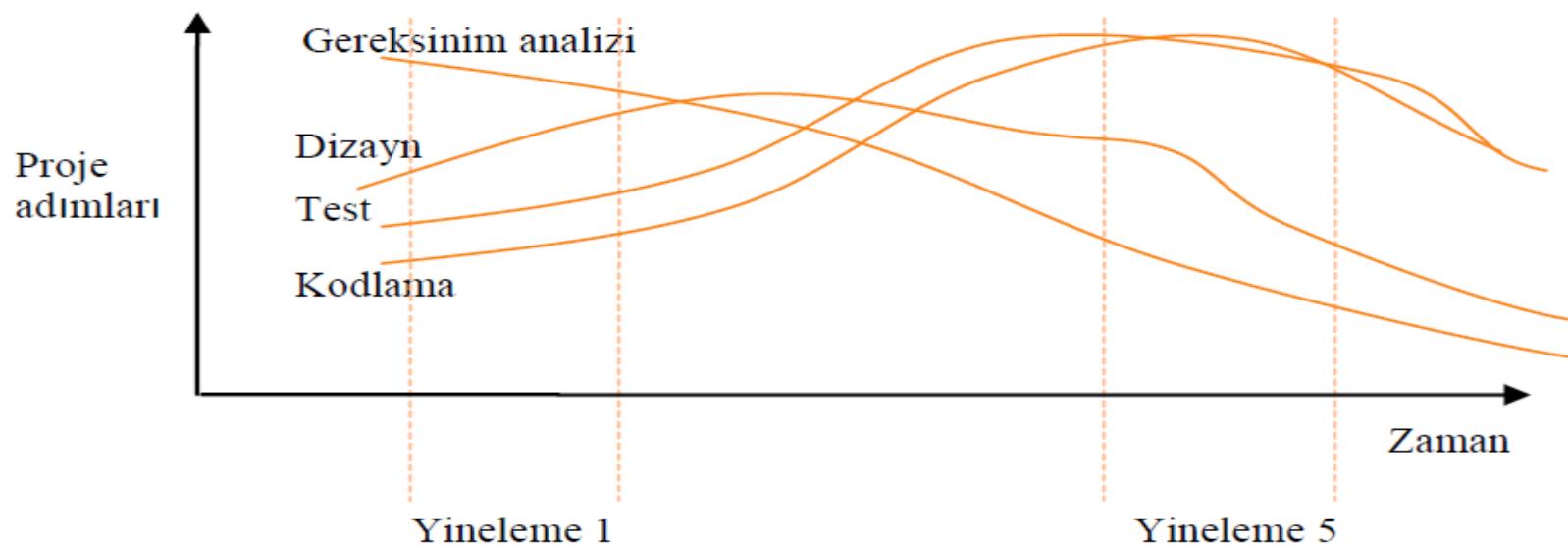
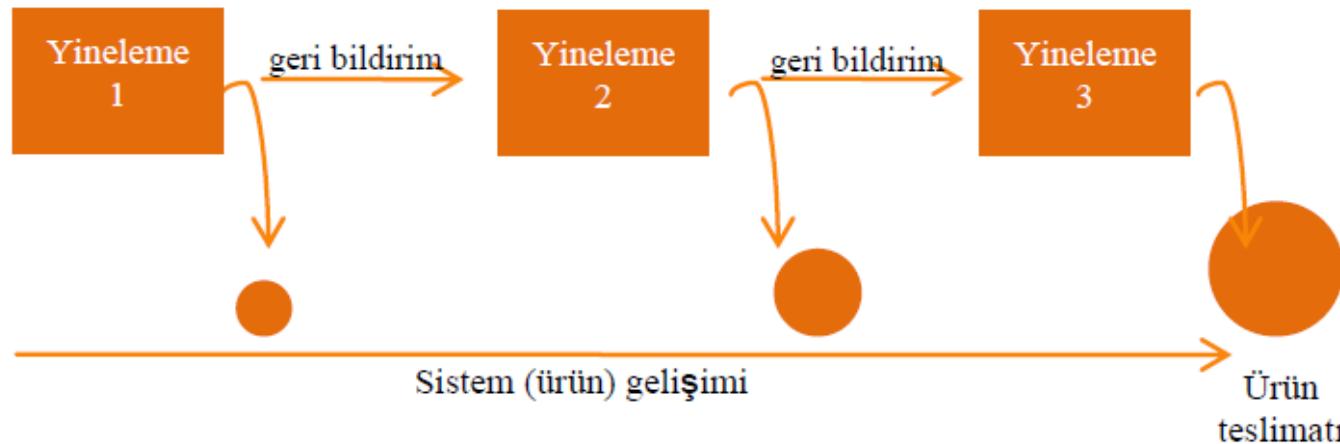
Özcan Acar - www.KurumsalJava.com

1. iterasyon

2. iterasyon

3. iterasyon

TEKRARLANAN YAZILIM GELİŞTİRME METODU



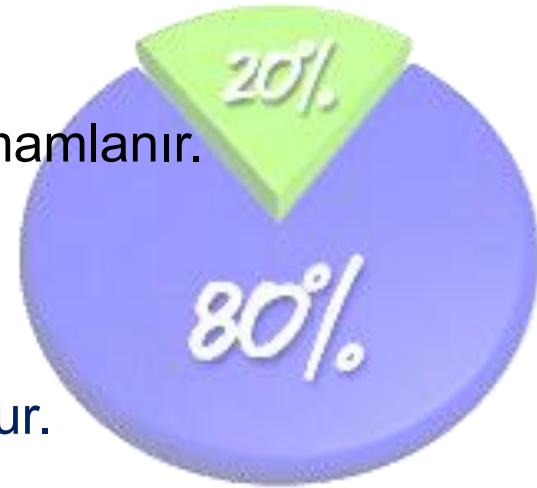
Agile metodların problemleri

- Sürece dahil edilen müşterilerin ilgisini sürekli kılmak zor olabilir.
- Takım üyeleri çevik metodları tanımlayan yoğun karışmaya uygun olmayabilirler.
- Önceliklerin değişimi birden fazla stakeholder(ortak) olması durumunda zordur.
- Sadeliği koruma fazladan iş gerektirir.
- Tekrarlanan geliştirmeye farklı yaklaşımlar olduğunda sözleşme bir problem olabilir.

80 - 20 Pareto Kuralı:

Bir projenin %80 'lik kısmı, proje süresinin %20 içinde tamamlanır.

Geriye kalan %20'lük iş ise zamanın % 80'ini alır.



Avantajları

- Yazılım ekibinin motivasyonu sürekli yüksek seviyede olur.
- Kısa sürede müşteri memnuniyeti sağlanır.
- Üretkenlikler artar.
- Yazılım kalitesi artar.
- Maliyetler düşer.
- Yazılım projelerinin başarısı %55 e kadar

Çevik (Agile) Yaklaşım Çeşitleri

- Sınırsız programlama (Extreme Programming-XP)
- Çevik Birleştirilmiş Süreç (Agile Unified Process)
- SCRUM
- Test Güdümlü Geliştirme (Test-driven Development)
- Çevik bilgi Metodu (Agile Data Method)
- Özellik güdümlü geliştirme (Feature-Driven Programming)

Scrum Nedir

- En genel ve en bilinen Agile frameworküdür.
- Adını Rugby isimli oyundan alır
- Rugby gibi takım toplanır, planlama oyunu oynar ve görevler dağıtılarak herkes tek bir hedef için çalışır.
- Kendi terminolojisi vardır.



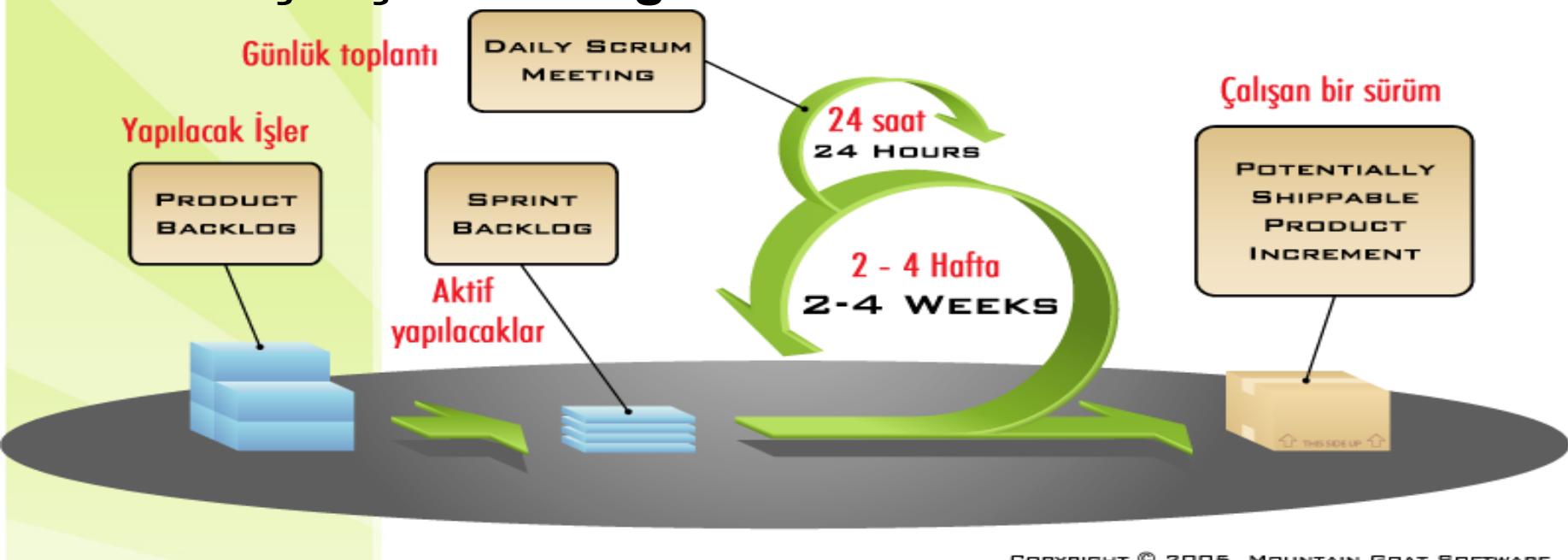
Scrum kullananlar

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce



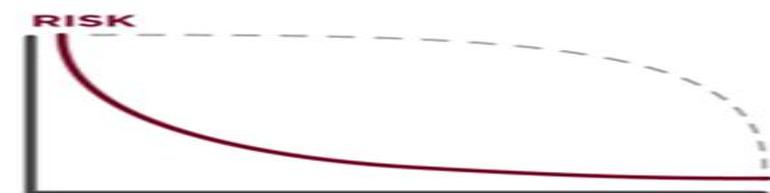
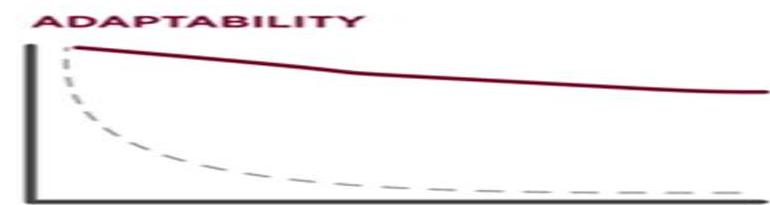
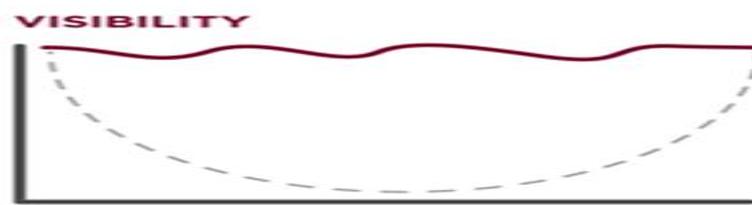
AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

Scrum'ın Çalışma Mantığı



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

AGILE DEVELOPMENT VALUE PROPOSITION



Scrum Development Process

Roles



Product Owner:
Set priorities



ScrumMaster:
Manage process,
remove blocks



Team: Develop
product



Stakeholders:
observe & advise

Key Artifacts

Product Backlog

- List of requirements & issues
- Owned by Product Owner
- Anybody can add to it
- Only Product Owner prioritizes

Sprint Goal

- One-sentence summary
- Declared by Product Owner
- Accepted by team

Sprint Backlog

- List of tasks
- Owned by team
- Only team modifies it

Blocks List

- List of blocks & unmade decisions
- Owned by ScrumMaster
- Updated daily

Increment

- Version of the product
- Shippable functionality (tested, documented, etc.)

Key Meetings

Sprint Planning Meeting

- Hosted by ScrumMaster; $\frac{1}{2}$ -1 day
- In: Product Backlog, existing product, business & technology conditions

 1. Select highest priority items in Product Backlog; declare Sprint Goal
 2. Team turns selected items into Sprint Backlog

- Out: Sprint Goal, Sprint Backlog

Daily Scrum

- Hosted by ScrumMaster
- Attended by all, but Stakeholders don't speak
- Same time every day
- Answer: 1) What did you do yesterday? 2) What will you do today? 3) What's in your way?
- Team updates Sprint Backlog; ScrumMaster updates Blocks List

Sprint Review Meeting

- Hosted by ScrumMaster
- Attended by all
- Informal, 4-hour, informational
- Team demos Increment
- All discuss
- Hold retrospective
- Announce next Sprint Planning Meeting

Development Process



Sprint:
30 days each

Sprint Planning Meeting

Daily Scrum

Daily Work

Sprint Goal

Sprint Backlog

Blocks List

Product

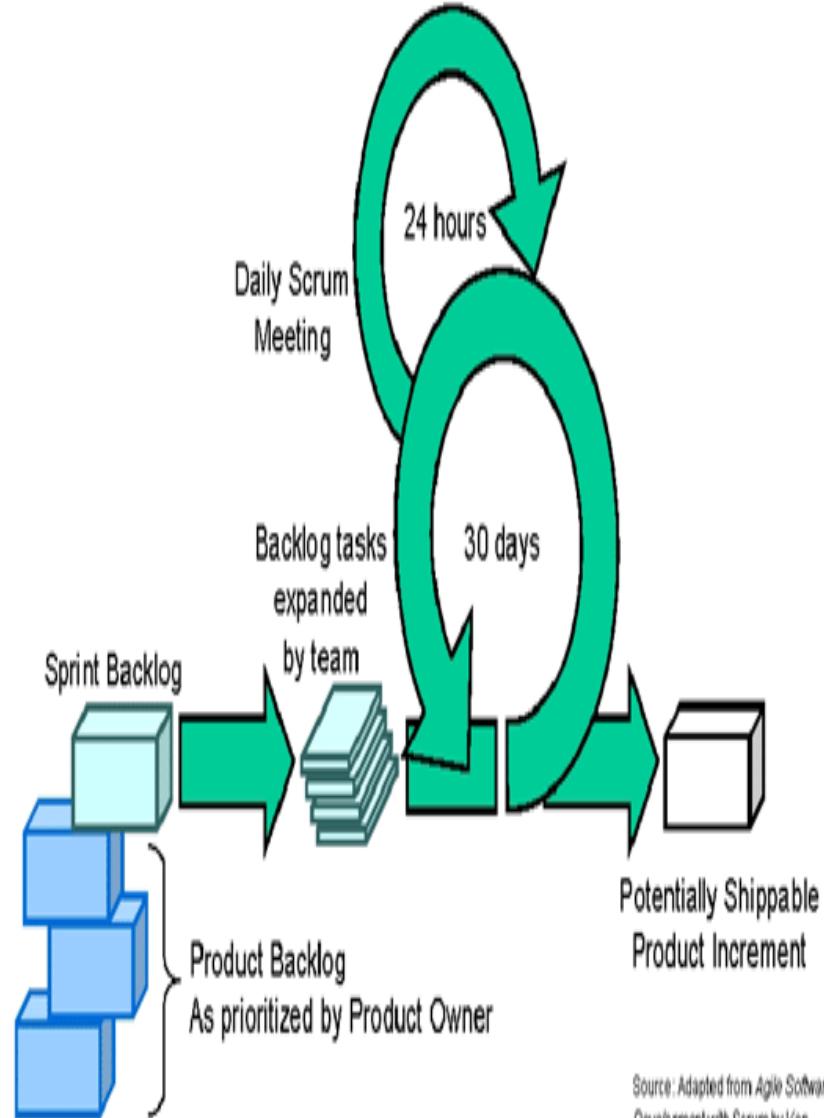
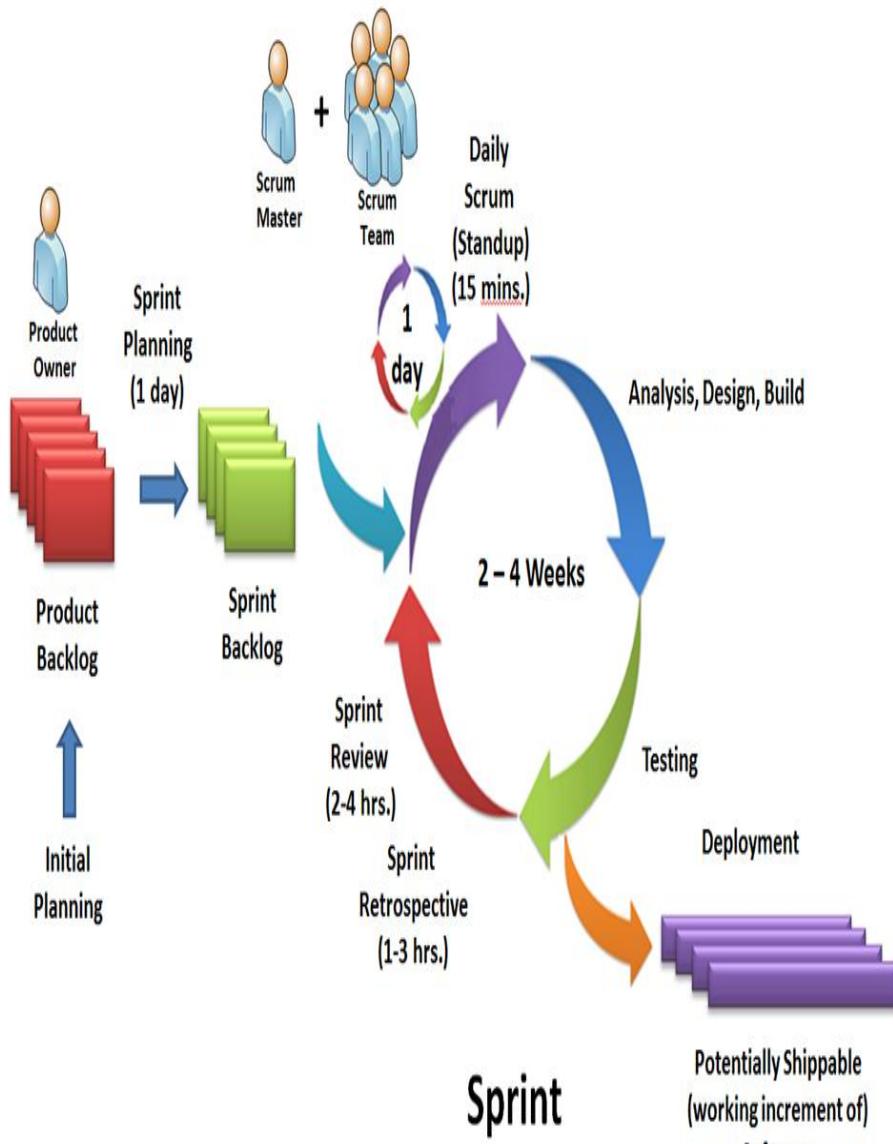
Increment

Sprint Review Meeting

Product Backlog

Scrum'ın Çalışma Mantığı

Scrum Yaşam döngüsü



Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.

AGILE RISK YÖNETİMİ



Scrum'ın Tanımı



Yıllardır yazılım geliştirmede süre gelen proje yönetimi sorunları, Scrum metodu doğru uygulandığı taktirde büyük ölçüde aşıldığı görülmektedir.
Bunun en temel nedeni «Scrum Ekibi» olusudur.

Scrum'da 3 temel kavram vardır.

- 1- Roller (Roles)
- 2- Toplantılar (Meetings)
- 3- Kavamlar / Araçlar (Artifacts)

Temel Kavramlar

Scrum'da 3 temel kavram vardır.

1- Roller (Roles)

- Ürün Sahibi (Product Owner)
- Scrum Yöneticisi (Scrum Master)
- Takım (Team)

2- Toplantılar (Meetings)

- Sprint (Koşu) Planlama (Sprint Planning)
- Sprint (Koşu) Gözden Geçirme (Sprint Review)
- Günlük Scrum Toplantısı (Daily Scrum)

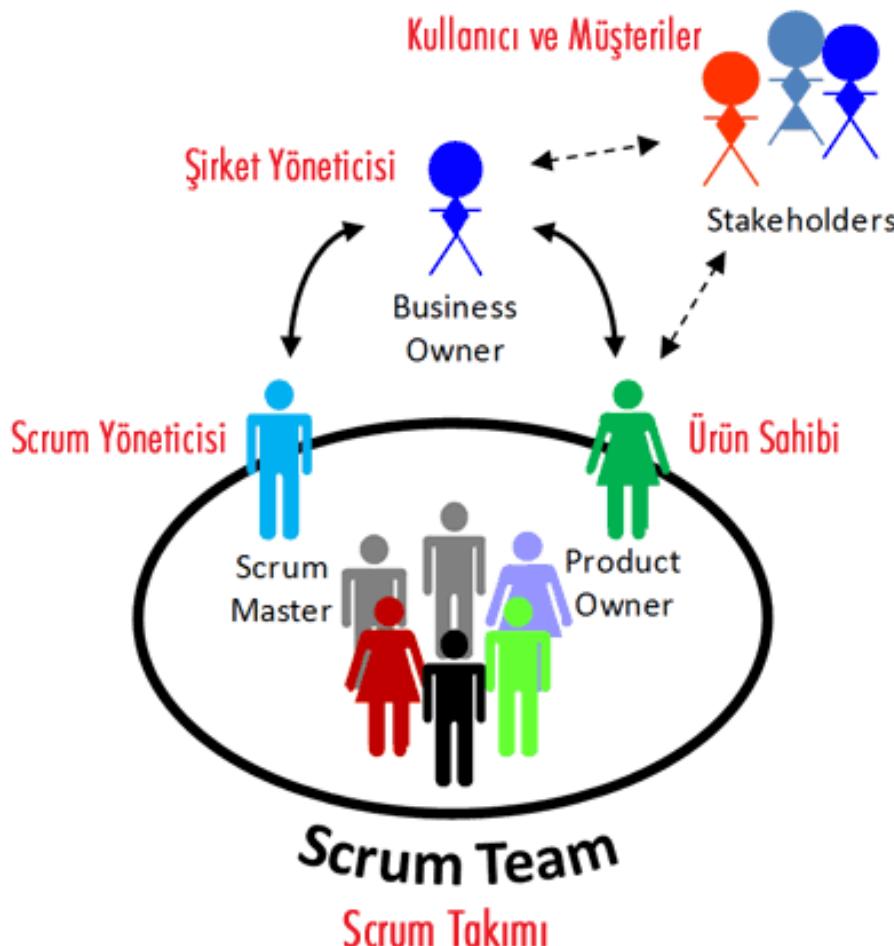
3- Kavramlar / Araçlar (Artifacts)

- Ürün Gereksinim Dokümanı (Product Backlog)
- Sprint (Koşu) Dokümanı (Sprint Backlog)
- Sprint Kalan Zaman Grafiği (Burndown Chart)



SCRUM

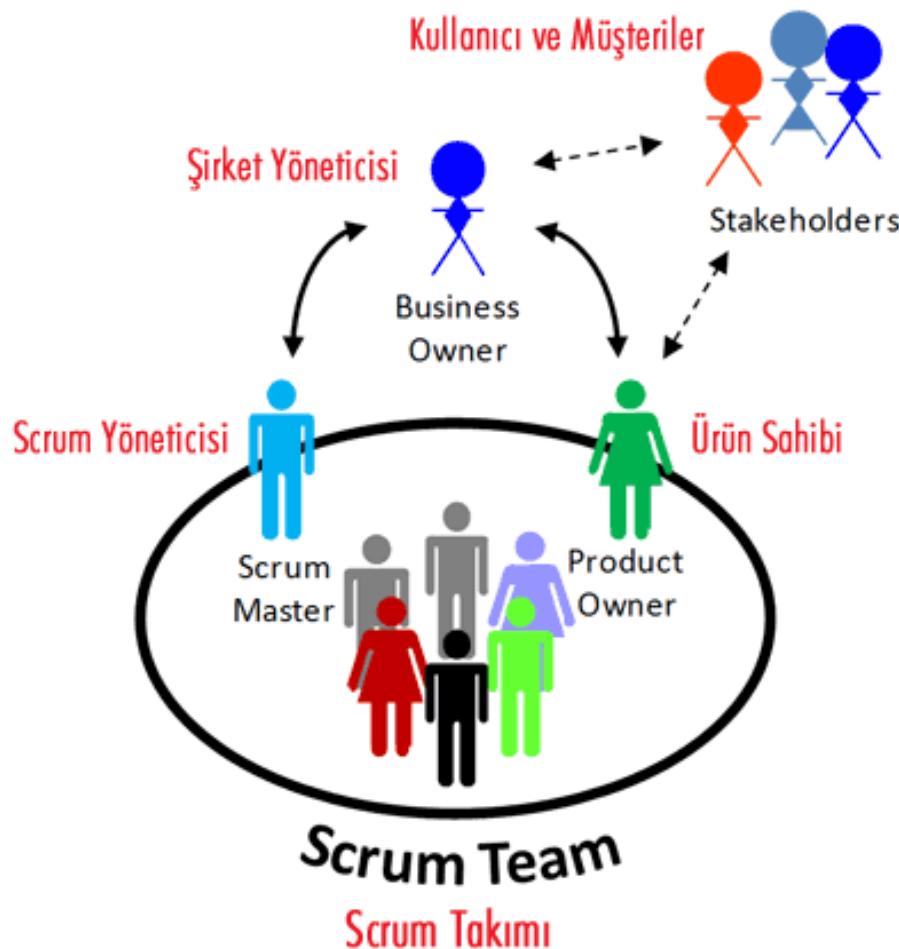
1- Roller (Roles)



- Ürün Sahibi (Product Owner)
- Scrum Yöneticisi (Scrum Master)
- Takım (Team)

SCRUM

1- Roller (Roles) - Ürün Sahibi (Product Owner)



Projenin iş değeri açısından geri dönüşü ile sorumludur. Ekibin bir parçasıdır, müşteri tarafından görevlendirilmiştir, detayları takip eder, geri dönüşler verir.

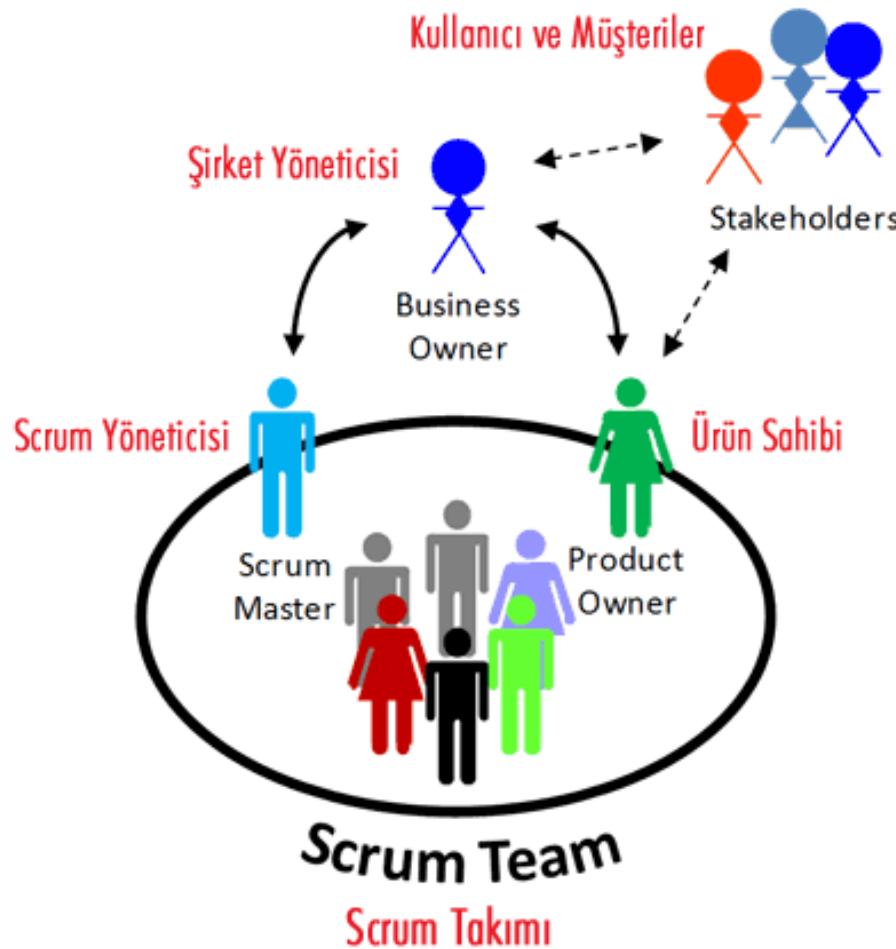
Ürün Sahibinin sorumlulukları;

- Paylaşımçı bir vizyonda çalışmak
- Gereksinimleri toplamak
- Gereksinim önceliklerini yönetmek
- Her iterasyon sonunda ürün kabulü yapmak
- Projenin yatırım geri dönüşünden sorumlu olmak (ROI-Return On Investment)

AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

- 1- Roller (Roles) - Scrum Yöneticisi (Scrum Master)



Scrum'da Geleneksel Proje Yöneticisi rolü yoktur.

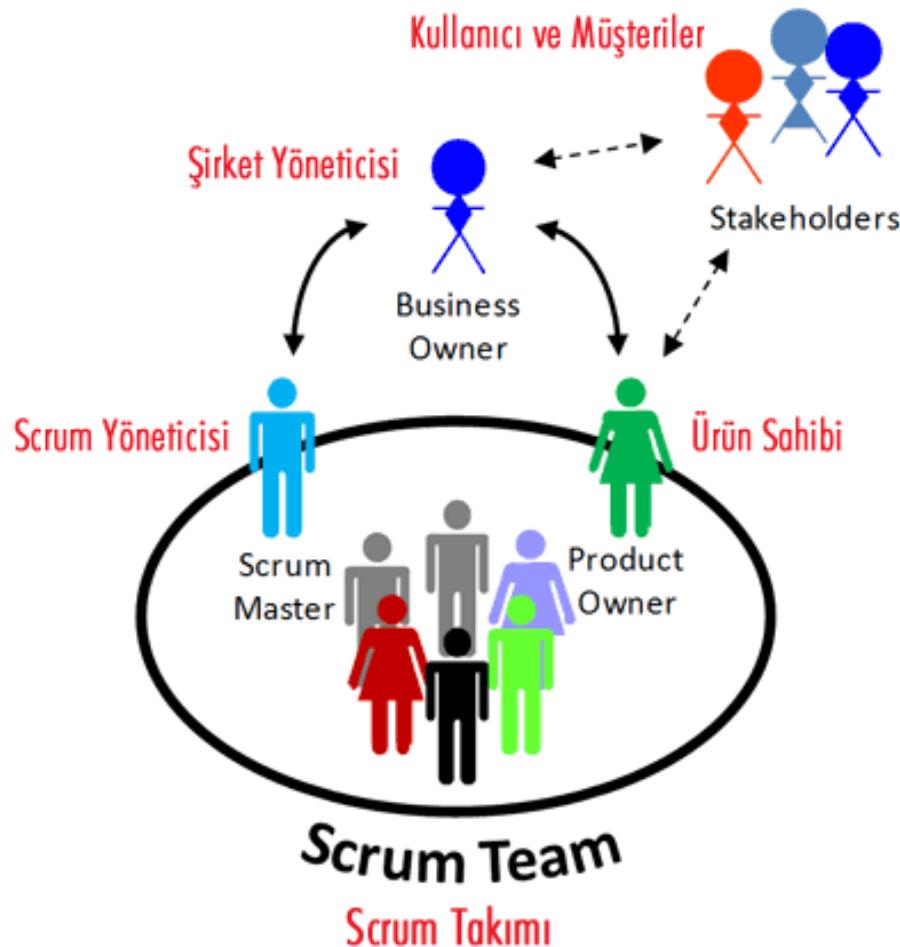
Scrum Master'ın görevleri;

- Takımın Scrum'ın temel değerlerine, pratiklerine ve kurallarına bağlı kalmasını garanti altına alır.
- Takımı ve organizasyonu Scrum'a adapte eder.
- Takımın dış etkilerden korunmasını ve sadece kendi işine yoğunlaşarak üretkenliğinin artmasından sorumludur.

AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

• 1- Roller (Roles) - Takım (Team)



Scrum Takımı, devamlı iletişim halinde olan ve tek bir hedefe ulaşmak için mücadele eden kişilerden oluşur.

Scrum'da takımların özellikleri;

- Gereksinimlerin süre tahminini yaparlar.
- 5 – 9 kişiden oluşur.
- Koşuya başlarken hedefi seçip, çalışma sonucunu belirlerler.
- Koşu hedefine ulaşmak için proje sınırları dahilinde her şeyi yapmakta serbesttirler .
- Kendi kendilerini organize ederler.
- Çalışma sonuçlarını belli aralıklar ile ürün sahibine gösterirler .

Takım üyeleri; geliştiriciler, testçiler, analistler, mimarlar, tasarımcılar ve hatta kullanıcılarından bile oluşabilir.

SCRUM

2- Toplantılar (Meetings)



- Sprint (Koşu) Planlama (Sprint Planning)
- Sprint (Koşu) Gözden Geçirme (Sprint Review)
- Günlük Scrum Toplantısı (Daily Scrum)

SCRUM

2- Toplantılar (Meetings) - Sprint (Koşu) Planlama (Sprint Planning)

Süreç adımları aşağıdakilerden oluşur;

- Geniş kapsamlı gereksinim listesinin çıkarılması
- Yapılacak dağıtımların (bir veya daha fazla) çıkış tarihinin ve fonksiyonel özelliklerinin belirlenmesi
- Başarılı geliştirme için uygun dağıtım gereksinimlerinin belirlenmesi
- Dağıtımlar için gereksinimlerin eşleştirilmenin yapılması
(Hangi gereksinim hangi dağıtımda yer alacak?)
- Dağıtımlar için takımların belirlenmesi
- Risk değerlendirmesi ve risk kontrollerinin belirlenmesi
- Gözden geçirmeler ve olası gereksinim değişikliklerinin belirlenmesi
- Geliştirme araçları ve altyapısının onaylanması
- Dağıtım maliyeti ve geliştirme, materyal toplama ve pazarlama maliyetlerinin hesaplanması
- Yönetimi ve destekleri gözden geçirme ve onaylama

Sprint (Koşu) Planlama



AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

2- Toplantılar (Meetings) - Günlük Scrum Toplantısı (Daily Scrum)



Koşu başladıkten sonra takım sürecin başka bir anahtar aktivitesi olan Günlük Scrum Toplantılarını gerçekleştirir. Bu kısa toplantı (15 dk) her iş gününde belirlenen saatte gerçekleştirilir ve tüm takım katılır (genelde sabahları). Takımın ilerleyişini ve karşılaştıkları engelleri görmek için önemli bir fırsattır.

Teker teker tüm ekip üyeleri, şu soruların cevaplarını verir;

- Dün ne yaptım?
- Bugün ne yapacağım?
- Önümde olan engeller ve karşılaştığım sorunlar neler?

Scrum Master, toplantı esnasında notlar tutar ve sorun yaşayan üyelerle toplantıdan sonra yardımcı olur.

- Sprint (Koşu) Gözden Geçirme (Sprint Review)



Ayakta Günlük Scrum Toplantısı (Daily Scrum)



SCRUM

2- Toplantılar (Meetings) - Günlük Scrum Toplantısı (Daily Scrum)



Günlük Scrum toplantıları kesinlikle bir tartışma platformu değil, işlerin ne durumda olduğu ve varsa sorunların paylaşıldığı bir toplantıdır, eğer tartışma gerekiyorsa bu toplantıdan sonra gerçekleştirilir.

Bu toplantılar çok faydalıdır ve sonuçları aşağıdaki gibidir:

- Engeller oluştu ise yönetim tarafından ortadan kaldırılır
- Daha az gereksiz tekrarlanmış efor harcanır
- Takım üyeleri arasında daha iyi birliktelik ve uyum sağlanır
- Hedef netleşir ve takım tarafından kabul edilir
- Takım iletişimini sağlar
- Takımın önünde yaptıklarını açıklayacak olmak bireyi başarılı olma yönünde teşvik eder
- Takım dinamiği ve etiği inşa edilir

AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

3- Kavramlar / Araçlar (Artifacts) - Ürün Gereksinim Dokümanı (Product Backlog)



Kullanıcı Hikayeleri (User Story)

Her tür kullanıcının, sistem içerisindeki tüm hareket ve eylemleri baz alınarak hazırlanan senaryolardır. Bir tek eylem baz alınarak, küçük parçalar halinde hazırlanması önemlidir.

Daha sonra bu senaryolara öncelik derecelendirmesi ve zorluk derecesini ifade eden hikaye puanı (story points) verilir.

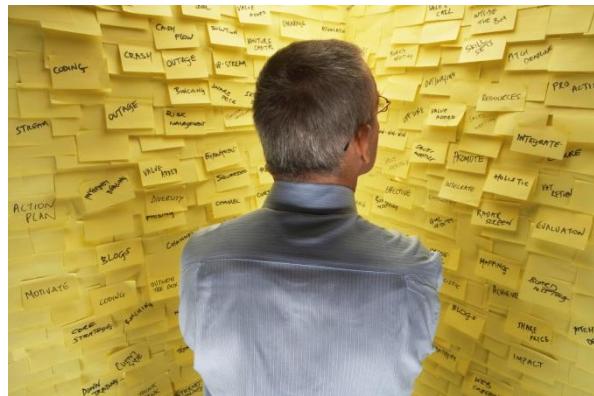
Öncelik değerleri 1 ile 10 arasında ardışık verilirken,
Puanlamalar 1, 2, 3, 5, 8, 13, 21, 34,... şeklinde verilir.

Süre bir çok etkene bağlı olarak, değişken olduğu için kullanılmaz. Scrum Ekibinin 1 haftada tamamladığı toplam hikaye puanları ile projenin toplam puanı, proje süresini zaten verecektir.

AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

3- Kavramlar / Araçlar (Artifacts) - Ürün Gereksinim Dokümanı (Product Backlog)



Product Backlog listesindeki sıra ile en öncelikli olanlardır. Hiçbir zaman önceliği düşük bir özellik veya fonksiyon önceliği yüksek bir özellik veya fonksiyondan önce geliştirilemez. Bu bağlamda bazı “sprint” lerde proje takımı, Product Backlog’dan 4 eleman, bazı sprint’lerde 25 eleman seçebilir.

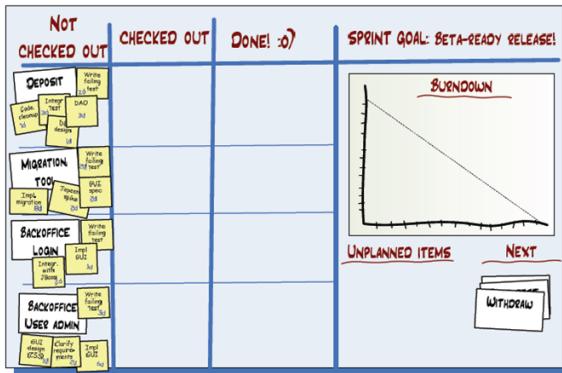
Seçilen özellik ve fonksiyonlar Sprint Backlog denilen ikinci bir listeye aktarılır. Proje takımı bir sonraki sprint başlangıcına kadar bir daha Product Backlog'a bakmaz, o sprint dahilinde sadece ilgili Sprint Backlog listesine odaklanır.

Sprint Backlog dahilindeki her özellik veya fonksiyon için maksimum 3 günlük geliştirme süresi verilir.

AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

3- Kavramlar / Araçlar (Artifacts) - Sprint Kalan Zaman Grafiği (Burndown Chart)

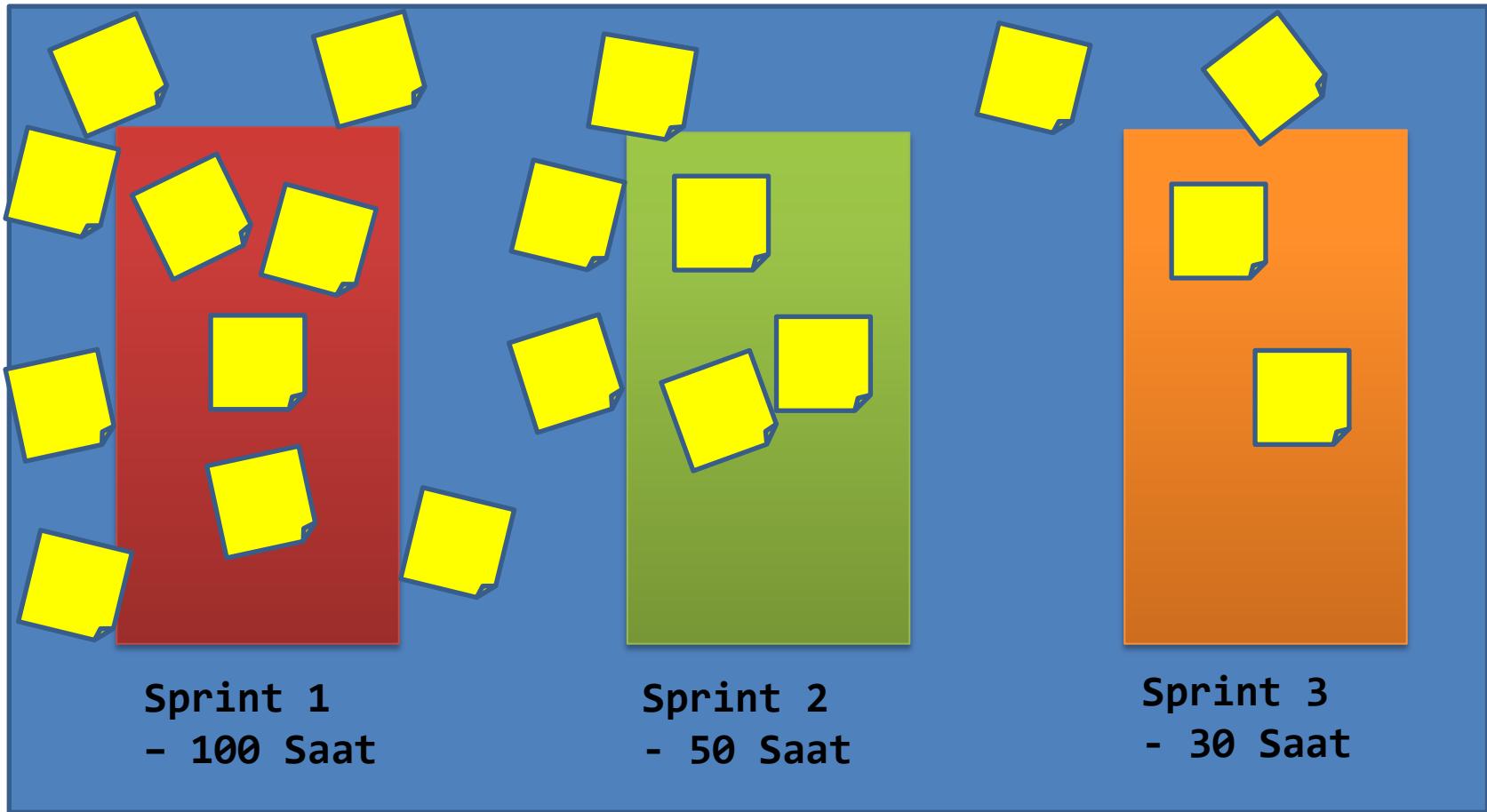


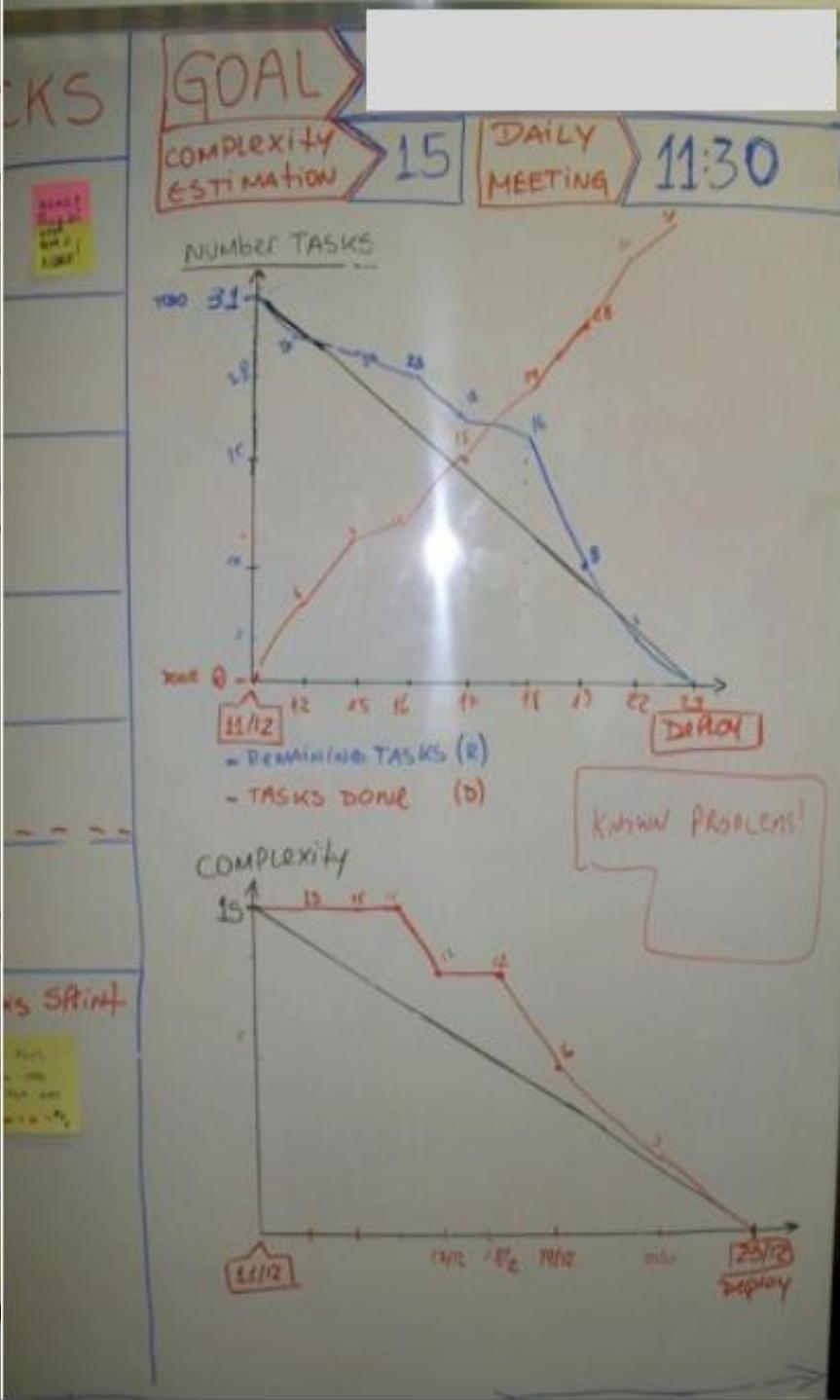
Bu grafik, iterasyon (sprint) boyunca işlerin ne kadarının yapıldığı ile normalde ne kadar yapılması gerektiğini karşılaştırılabilmesini sağlar.

Bir iterasyonun toplam 100 saatten ve 20 günden olduğunu farz edelim. Normal olarak beklenen her gün 5 saatlik bir işin yapılmasıdır. Takım elemanları her gün ne kadarlık bir iş gerçekleştikleri bilgisini girerler.

İstenen isteklerin iterasyon süresi içerisinde gerçekleşip gerçekleşmeyeceği bu şema yardımıyla izlenebilmektedir.

Release Planing





Product Backlog



AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

Örnek Bir Sprint (Koşu) Dokümanı (Sprint Backlog)

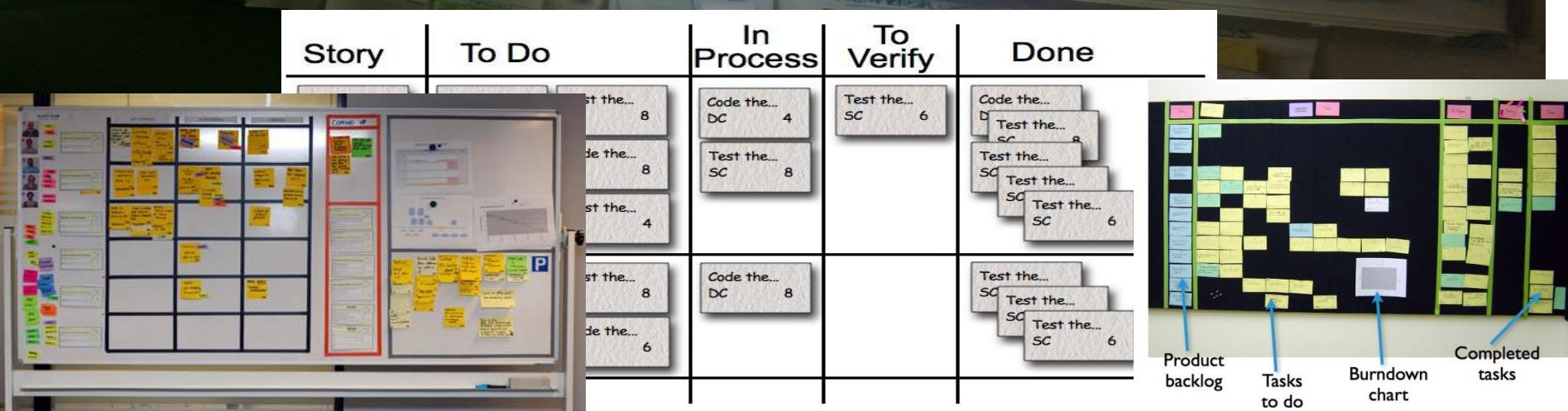
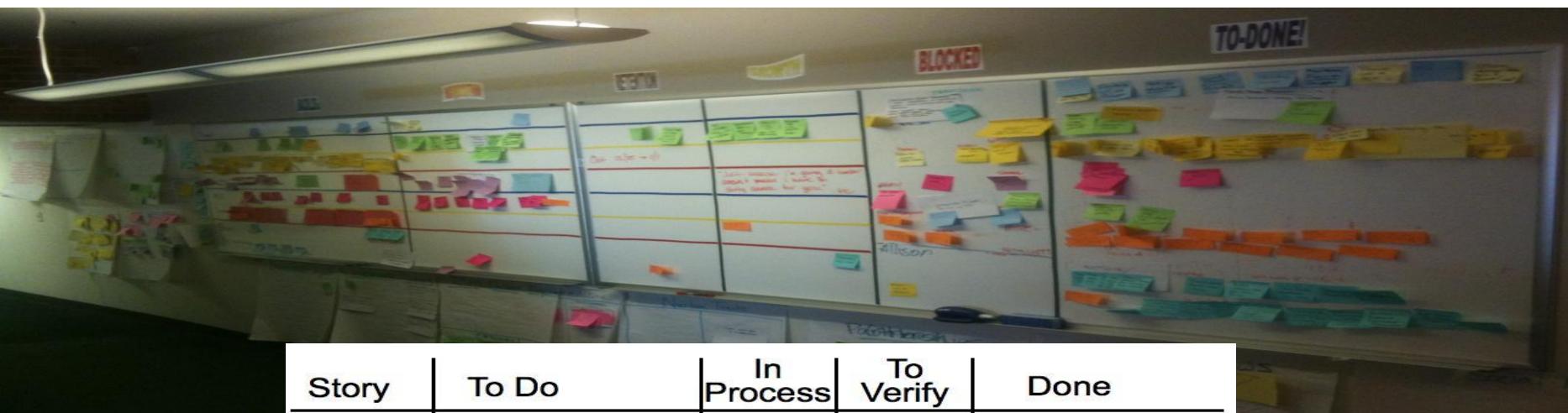
3. Koşu (Sprint) Gerçek Hava Durumu Eklentisi

Hikaye ID	Hikaye / Görev	Gün / Çaba																																										
		0	1	2	3	4	5	6	7	8	9	10	11	12	13																													
63	74	68	64	56	49	41	31	29	32	32	32	32	32	32	32																													
10	Hava sağlayıcı sisteminin bir günlük sıcaklık verilerinin getirilmesi	4	16	12	8	3	3	3	3	3	3	3	3	3	3																													
	Sunucuya bağlamak ve kimlik doğrulaması	8	7	7	7	4	0	0	0	0	0	0	0	0	0																													
	Data klasörüne bağlanmak	6	6	4	4	4	1	1	1	1	1	1	1	1	1																													
	Veriler içerisinde sıcaklık değerinin ayrıştırılması	16	16	16	16	16	16	8	2	0	0	0	0	0	0																													
	Sıcaklık verileri istemciye iletmek	4	4	4	4	4	4	4	0	0	0	0	0	0	0																													
11	Sağlayıcı, yağmur, kar, vb detayları getir	4	4	4	4	4	4	4	4	0	0	0	0	0	0																													
	Kar ve yağmurların veriler içinden alınması	4	4	4	4	4	4	4	4	0	0	0	0	0	0																													
	Kar / yağmur veri istemciye iletilmesi									3	3	3	3	3	3																													
	İstemci ekranında yenileme									4	4	4	4	4	4																													
	Refactor sunucu kodu																																											
12	Serverdan birkaç günlük verilerin getirilmesi	10	10	10	10	10	10	10	10	10	10	10	10	10	10																													
	Serverdan birkaç gün verilerin getirilmesi	3	3	3	3	3	3	3	3	3	3	3	3	3	3																													
	Istemciye birkaç günlük veriyi iletmek	6	6	6	6	6	6	6	6	6	6	6	6	6	6																													
13	Otomatik yenileme özelliği	2	2	2	2	2	2	2	2	2	2	2	2	2	2																													
	4 saatte 1 kez sunucudan verileri güncelle																																											
	İstemciye veri güncelleştirmesi yapmak																																											
Toplam Çaba																																												
<p>Scrum Burndown Chart showing remaining work over 13 days. The chart starts at 63 points on day 0 and decreases to 32 points by day 13. Points are labeled above each data point.</p> <table border="1"> <thead> <tr> <th>Gün</th> <th>Çalışma (Puan)</th> </tr> </thead> <tbody> <tr><td>0</td><td>63</td></tr> <tr><td>1</td><td>74</td></tr> <tr><td>2</td><td>68</td></tr> <tr><td>3</td><td>64</td></tr> <tr><td>4</td><td>56</td></tr> <tr><td>5</td><td>49</td></tr> <tr><td>6</td><td>41</td></tr> <tr><td>7</td><td>31</td></tr> <tr><td>8</td><td>29</td></tr> <tr><td>9</td><td>32</td></tr> <tr><td>10</td><td>32</td></tr> <tr><td>11</td><td>32</td></tr> <tr><td>12</td><td>32</td></tr> <tr><td>13</td><td>32</td></tr> </tbody> </table>															Gün	Çalışma (Puan)	0	63	1	74	2	68	3	64	4	56	5	49	6	41	7	31	8	29	9	32	10	32	11	32	12	32	13	32
Gün	Çalışma (Puan)																																											
0	63																																											
1	74																																											
2	68																																											
3	64																																											
4	56																																											
5	49																																											
6	41																																											
7	31																																											
8	29																																											
9	32																																											
10	32																																											
11	32																																											
12	32																																											
13	32																																											

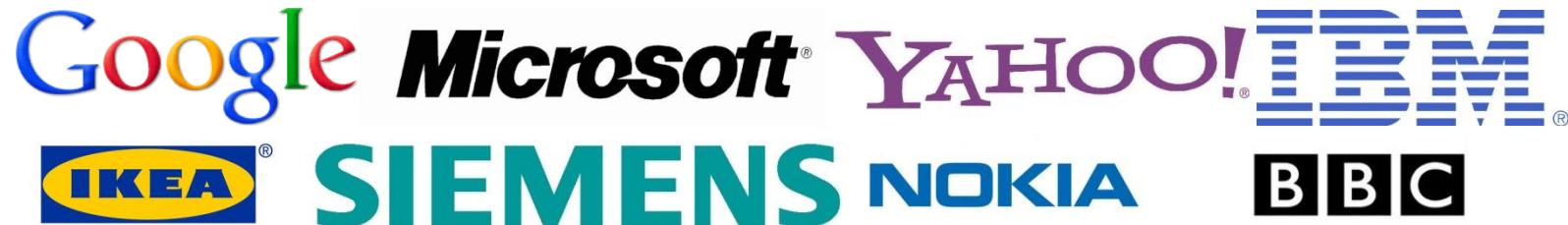
AGILE (ÇEVİK) YAKLAŞIM İLE SCRUM YÖNTEMİ

SCRUM

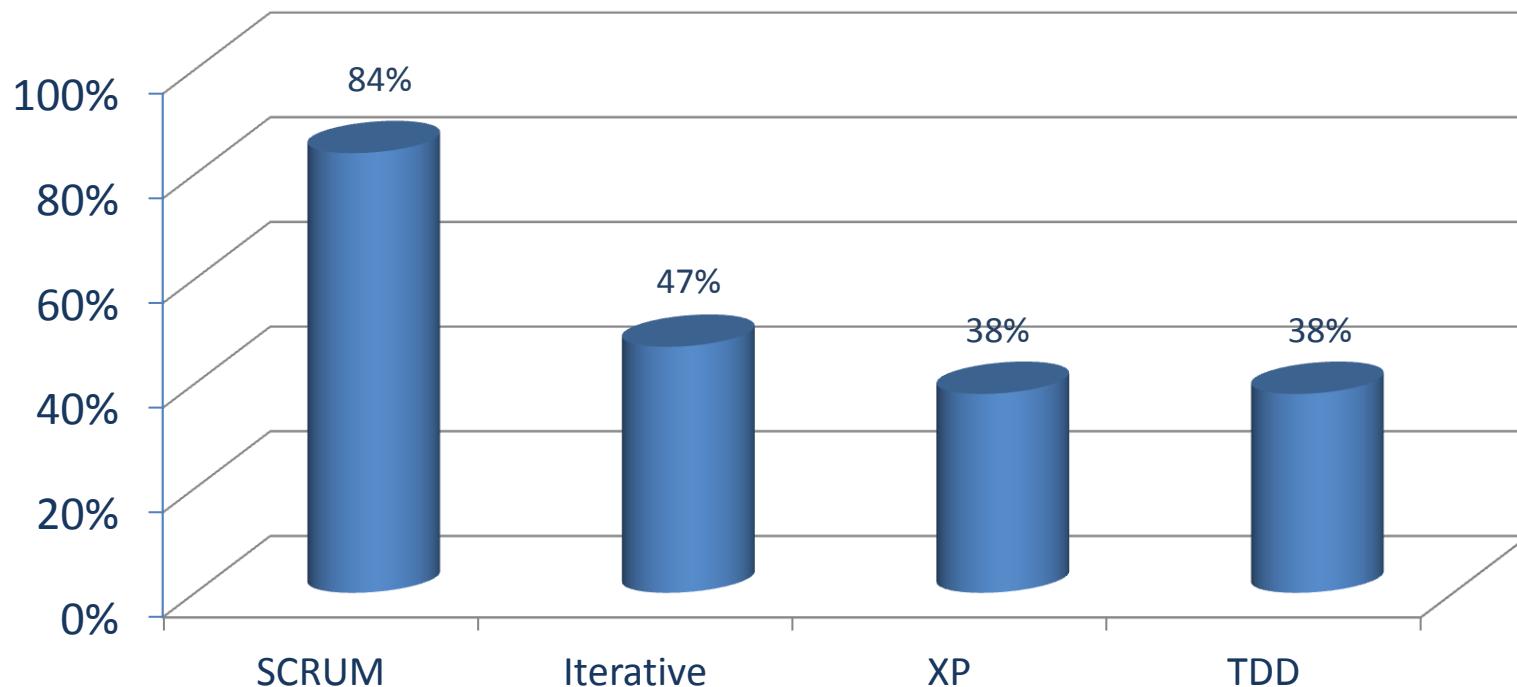
Scrum Tahtası (Scrum Board)



Kimler Kullanıyor



PY Metotları Kullanım Oranları (2008)



Agile in Avantajları

- İnsanın doğal eğilimine çok yatkındır öğrenim gerektirmez adaptasyon hızlıdır.
- Kısa döngüler dolayısı ile takım elemanlarında motivasyon çok yüksektir. Verim artışı yaşanır.
- Sık çıktı üretip geri besleme aldığından kaynağı müşteri ihtiyaçlarına ve sonuca kanalize etmeye odaklanır.
- Plan aşamasında ayrıntılı plan yerine iterasyonun planı yapılır.
- Değişime açıklık ve esneklik en üst düzeydedir.
- Sürdürülebilir Kalite
- Proje planlama ve yürütme bir arada
- Takım oyunu

AGILE in dezavantajları

- Kurumsal bir yapıda uygulaması gerçekten zor.
- Dökümantasyon hakkında ki taşları yerinden oynatan yaklaşımı.
- Sürekli değişen ihtiyaçlar dolayısı ile aşırı çalışma.
- Ürünün başarısı = projenin başarısı dolayısı ile kariyer riski
- Takım üzerindeki hedef baskısı

Agile mi ? Geleneksel Proje Yönetimi mi ?

- Belirsizlik çoksa, müşteri iletişime açıksa ve 100 metre koşusu yapıyorsanız agile kullanmalısınız.
- Maraton koşuyorsanız, karmaşık bir kaynak kullanımı varsa, müşteri ile iletişim kolay değilse geleneksel proje yönetimi kullanmalısınız.



TASK BOARD - CRUCIBLE

CRUCIBLE

Context: On the fly

[View](#) | [My Issues](#)

Views -

Touch

New card | View Version • 2.1.X

Hide Sub-tasks

[Expand all](#) | [Collapse all](#)

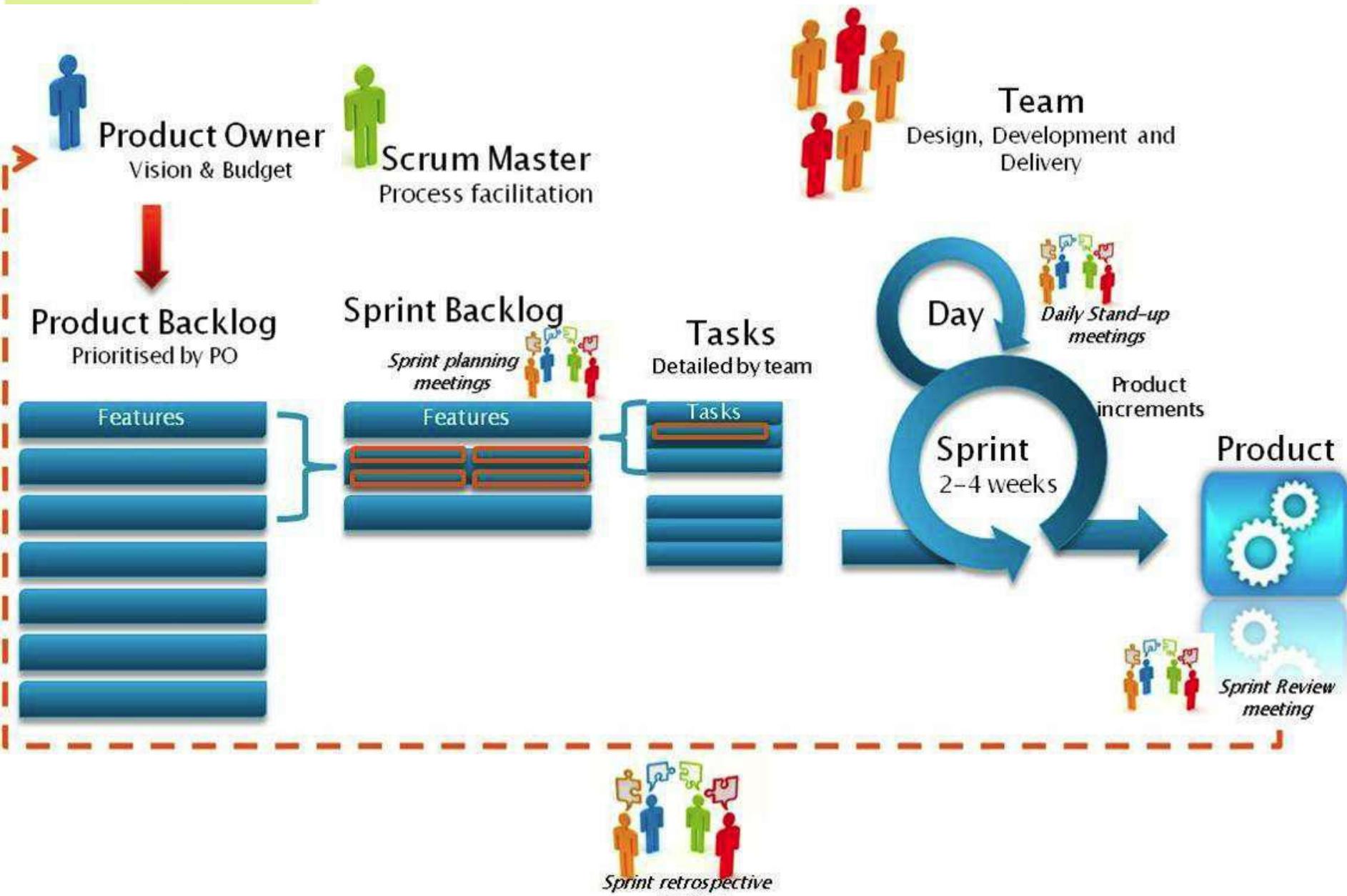
Board Search

Context [On the fly](#) Assigned to me Filters - Edit Sort by Summary  Highlight - Edit [Save context as...](#)

Save context as

TO DO - 46h	IN PROGRESS - 0h	IN REVIEW - 0h	DONE - 0h
20 Issues 12	0 Issues	0 Issues	19 Issues 12
<p>885 </p> <p>"Diff To" is ignored in "Add Files" in Lite SCM</p> <p>Adding a Confluence page to a review does not allow you to diff it to a</p> <p>Original estimate: N/A Component: Light SCM Plugin Aff. Version: 1.6.4</p> <p>Reporter: Erik van Zijst [Atlassian]</p>	<p>2095 </p> <p>Activity stream says "finished" instead of "uncompleted"</p> <p>No description</p> <p>Original estimate: N/A Component: User Interface Aff. Version: None</p> <p>Reporter: Erik van Zijst [Atlassian]</p>		<p>1385 </p> <p>Add "Change Diff" button for file revision chosen from Light SCM plugins</p> <p>Please see attached screenshot</p> <p>Original estimate: N/A Component: Light SCM Plugin Aff. Version: 2.0-M5</p> <p>Reporter: Partha Kamal [Atlassian]</p>
<p>817 </p> <p>Add Actioning User to Crucible/Fisheye events</p> <p>Verify with Tom, We are talking about events that are passed to listening</p> <p>Original estimate: 8h Story Points: 1 Component: Unknown Aff. Version: None</p> <p>Reporter: Tom Davies [Atlassian]</p>		<p>1604 </p> <p>Add fix-completion flags to REST</p> <p>Expose completion flags per fix</p> <p>Original estimate: N/A Component: API Aff. Version: None</p> <p>Reporter: Erik van Zijst [Atlassian]</p>	<p>1622 </p> <p>Document code for review update notifications</p> <ul style="list-style-type: none"> the polling period algorithm ReviewUpdatedAjax <p>Original estimate: N/A Component: Unknown Aff. Version: None</p> <p>Reporter: Nicolas Venegas [Atlassian]</p>
<p>1338 </p> <p>Add plugin point to be able to create custom fields</p> <p>Customers would like the ability to add custom fields to reviews.</p> <p>Original estimate: N/A Story Points: None Component: Plugins Aff. Version: 1.6.6</p> <p>Reporter: Partha Kamal [Atlassian]</p>	<p>1175 2 Subs </p> <p>Allow me to transition a linked jira when closing a review</p> <p>checkbox when moving from summarize to close on a reviews with linked</p> <p>Original estimate: 12h Story Points: 3 Component: Unknown Aff. Version: None</p> <p>Reporter: Brendan Humphreys [Atlassian]</p>		<p>1603 </p> <p>Due date display is too exact</p> <p>The due date field in the edit review form is</p> <p>Original estimate: N/A Component: User Interface Aff. Version: 2.0-RC1</p> <p>Reporter: Peter Moore [Atlassian]</p>
<p>874 841 </p> <p>Documentation for Web Fragment Plugin Points</p> <p>No description</p>	<p>1979 </p> <p>Enforce maximum indent level for comment threads</p> <p>As we do in the text</p>		<p>1334 </p> <p>Hot Deployment of Plugin does not work</p> <p>Get the following error:</p>
			<p>1449 </p> <p>Largish review beachball on resize and when expanding FRXs</p> <p>Observable in</p>

Scrum Dynamic Model





PinkRoccade



PHILIPS

AIRFRANCE KLM

delta lloyd



Fly
Emirates



Wolters Kluwer

Ministerie van Algemene Zaken



LIBERTYGLOBAL



De TelefoonGids
Gouden Gids

Projects using Scrum

Organizations	Ministerie van Algemene Zaken	LIBERTYGLOBAL	De TelefoonGids Gouden Gids
Industries	Government	Media & Telecommunications	Advertising
Websites	www.rijksoverheid.nl	www.upc.nl	www.detelefoongids.nl
Roles	ScrumMaster	Program Manager Online	Program Manager Marketing Online
Key elements	Content management, open standards, accessibility	eCommerce, online sales, conversion	Content, search, SEO, high volume traffic
Scrum	In-house Scrum teams Local Product Owner	Distributed Scrum team (Veenendaal) Corporate Product Owner	Distributed Scrum teams (Denver, Vienna) Local Product Owner

Roots of Scrum

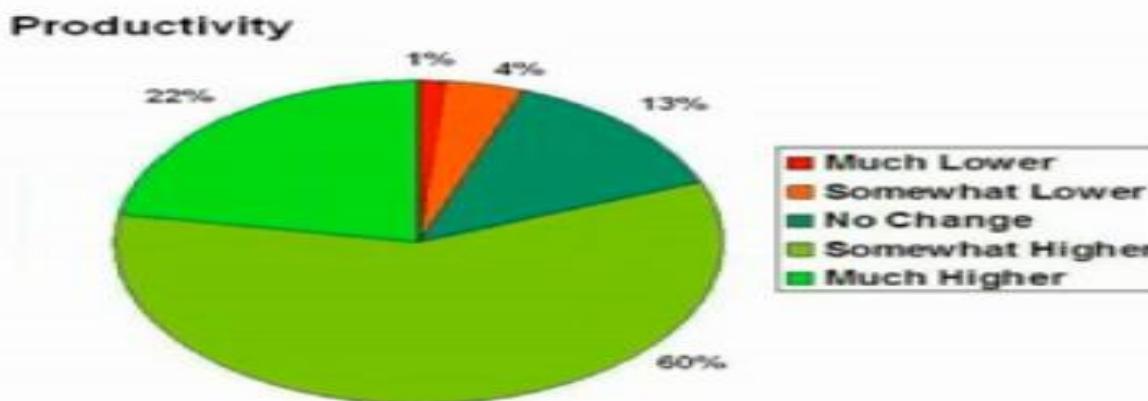


Çevik (agile) yöntemlerin en büyük hedefi müşteriye istenilen yazılımı en kısa sürede ve doğru bir biçimde teslim edilmesini sağlamaktır.

Çevik yöntemler kalite noktasında çok fazla bir şey söylemezler,
işte bu noktada yalın (lean) yaklaşımın
öğretilerini Kanban sistemiyle devreye alınabilir.



Benefits: improved PRODUCTIVITY



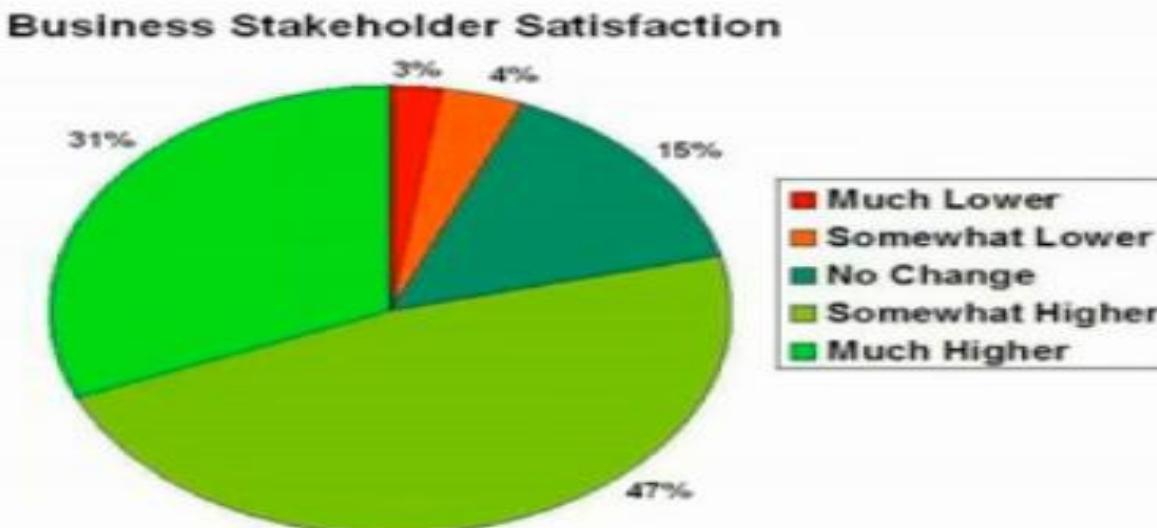
Agile ile olan değişim nasıl olduğunu gösteren anket sonucu. Katılımcıların %60 i kalitenin arttığını söylemişler.

Benefits: improved PRODUCTIVITY



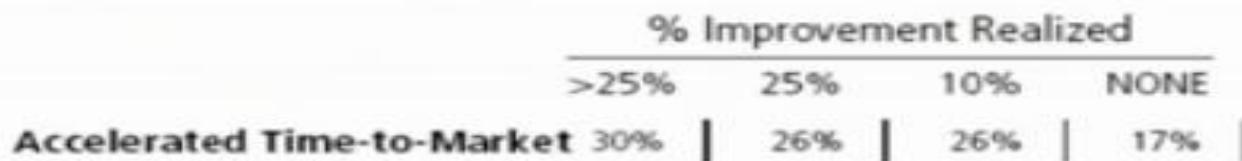
2500 kişinin katılımıyla yapılan ankette, katılımcıların %90 'ı Agile ile üretkenliklerinin büyük olasılıkla arttığını ifade etmiş. Katılımcıları %80-85 e yakını; kalitenin %10-ve üzerinde oranında arttığını söylemişler

Benefits: increased CUSTOMER SATISFACTION



Katılımcıların %80 oranında memnun olduklarını söylemişler

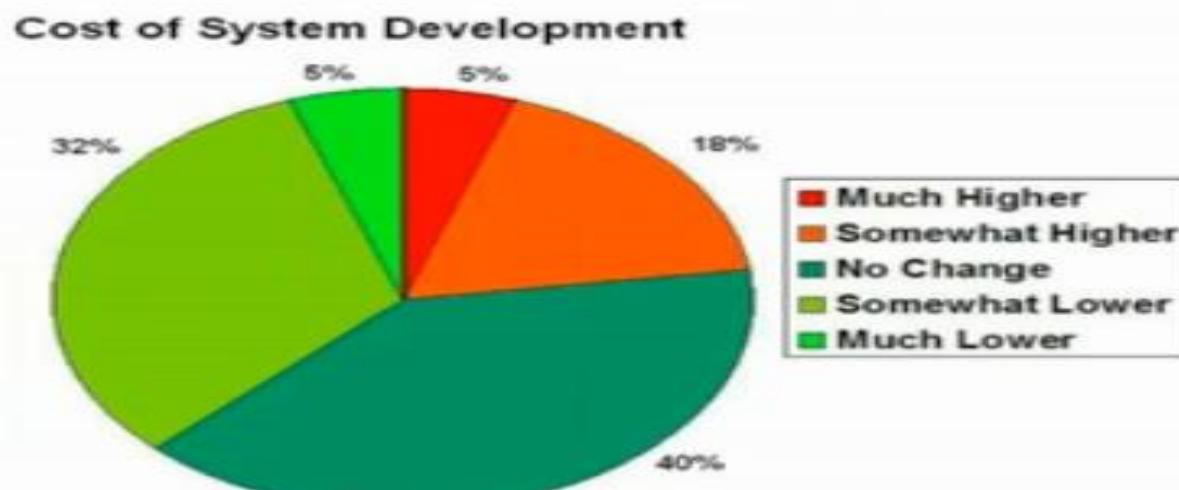
Benefits: increased CUSTOMER SATISFACTION



Copyright © 2008 VersionOne, Inc. All Rights Reserved.

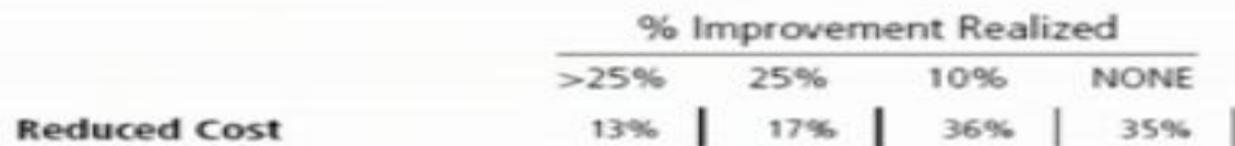
Müşteri memnuniyetinin %10 üzerinde arttığını söylüyor anket.Katılımcıların %83 olduğunu görüyoruz

Benefits: decreased COSTS



Ankette katılımcıların %72 si maliyetlerin büyük oranda azalışa geçtiğini agile projelerle birlikte söylemiştir.

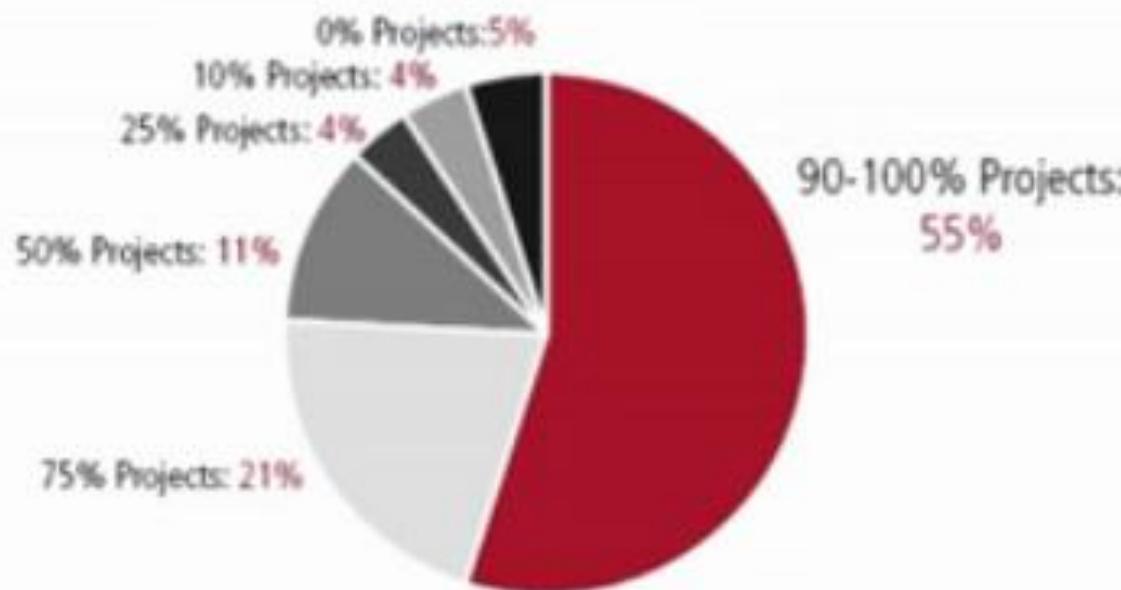
Benefits: decreased COSTS



Copyright © 2008 VersionOne, Inc. All Rights Reserved.

diğer ankette benzer yaklaşımalar söz konusudur. katılımcıları %65 'i maliyetlerin %10 ve üzerinde azaldığını beyan etmiştir.

Benefits: leads to SUCCESS



Copyright © 2008 VersionOne, Inc. All Rights Reserved.



Agile uygulamalarla projelerin başarısının %100 oranında arttığı görülmektedir.

Scrum Proje Örnekleri: Sentinel Project

**The Sentinel Project (8): 2006 yılında FBI tarafından Lockheed Martin şirketine yaptırılan bir yazılımdır.

**Kağıtta bulunan bilgilerin sisteme aktarılmasını hedeflemektedir.

**Proje 4 fazdan oluşmaktadır.

**İlk iki fazı proje bütçesinin büyük bir kısmı kullanılarak tamamlanmıştır

** fakat ikinci fazında gerçekleştirilen işlemler kullanıma açılamamıştır. (6 sene 351m \$)

**Daha sonra FBI isteği üzerine Scrum kullanılarak geliştirme işlemine devam edilmiş

** ve tahmini maliyetinin çok altında geliştirilmesi tamamlanmıştır. (Tahmini iş 65m \$)



GOV.UK yine atık felsefe ile geliştirilmiştir.

** 140 kişi ve 14 takım çalışmıştır.

** Atık yazılım geliştirmenin büyük yazılımlar için uygulanabilirliğini ölçeklenebilirliği ilgili yazda analiz edilmiştir.

**Çoğu takım Scrum kullanmıştır. Atık yazılım geliştirme ile:

- Daha fazla üretkenlik sağlanmıştır.
- Daha kaliteli ürün elde edilmiştir.
- Daha hızlı geliştirme yapılmıştır.

İngiltere Atık E-Devlet Başarısı

Atık yazılıma geçme nedenleri:

- Çok fazla tanımlayıcı ve mecburi dokümantasyon varlığı
- Kabul edilemez proje zamanları
- Organizasyon düzeyinde aşırı süreç mühendisliği
- Proje takımlarının esnek olmayan iş yorumları
- Projenin mevcut durumunun ve gidişatının açık olmayışı

Atık yazılım kullanarak şu **faydalar sağlanmıştır**:

- Hızlı başlangıç, yönetim maliyetinin azalması, paydaşlarla etkin iletişim, bakım kolaylığı

ÖZET OLARAK;

- *Scrum Framework'ün çeşitli yönleri Yalın ilkelerini destekler.
- * Scrum Takımları olgunlaşıkça ve gelişikçe, sıkılıkla Yanlı Düşünceyi yinelemeli ve artımlı geliştirmede daha fazla değer bulmak için etkili bir araç olarak görürler.
- *Belirli teknikler gelir ve gider, ancak geliştirmeye sürekli dikkat etmek sağlıklı yazılım geliştirme ekipleri sürdürmede çok önemlidir.
- *Scrum Çerçevesi, Kanban'da bulunanlara benzer Lean geliştirme yöntemlerine yer verebilecek kadar esnektir.
- *Scrum'ı Lean Düşünce üzerinden görüntülemek genellikle daha iyi kalite, yüksek verimlilik ve daha az atık sağlar.



ÖZET OLARAK;

- *Bir ekibin Scrum uygulamasını kasıtlı olarak en iyi duruma getirmek karmaşık olabilir.
- *Gelişme yolları ararken, mükemmelin yeterince iyinin düşmanımasına izin vermeyin.
- *Scrum'ın mükemmelliği, müşterilerin değer verdiği yüksek kaliteli çalışma yazılımı sağlanmasıdan çok daha az önemlidir, bu nedenle önce ürünü gerçekten iyileştiren şeylere odaklanın.
- *İteratif incremental modelin, en yüksek ölçüde yapıldığı atık yazılım geliştirme devlet projeleri için uygundur.
- *Gözle görülür bir ilerleme sağlamak hem müşteri hem de geliştiricilerin yazılıma olan inancını artırır.



Alınan Dersler ve Geliştirilecek Alanlar

Scrum Türk'lere Uygun Değil ☺

Kalitenin Kalbi “Ürün Özellik Listesi”

COBIT için Risk Yönetimi İyileştirmesi

Sürüm Planlamanın Daha İyi Yapılması

Referanslar

- Ekitaplar
 - The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process, Jeff Sutherland, Ph.D. Ken Schwaber,
 - Lean Software Development: An Agile Toolkit, Mary Poppendieck, Tom Poppendieck
 - Yonetmeler icin Dogru Sorular CMMI, Orhan Kalayci

Bağlantılar

- http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration
- <http://www.nitelik.net/>
- <http://www.extremeprogramming.org/rules.html>
- <http://agilemanifesto.org/>
- <http://www.redmine.org/>
- dminebacklogs.net/
<http://agilemanifesto.org/>
<http://www.pragprog.com/titles/pad/practices-of-an-agile-developer>
http://en.wikipedia.org/wiki/Agile_software_development
http://www.ibm.com/developerworks/rational/library/08/0701_ellingsworth/
<http://scrum-master.com/en/default.aspx>
<http://www.scrumalliance.org/>
- <http://mhazer.blogspot.com/2008/10/agile-yazlm-gelitirme-ve-scrum.html>

Scrum training

- Jeff Sutherland <http://www.scrumalliance.org/profiles/70-jeff-sutherland-phd>
- Ken Schwaber <http://courses.scrum.org/about/ken-schwaber>
- Mike Cohn <http://www.mountaingoatsoftware.com/training-available>

Scrum certification

http://www.scrumalliance.org/scrum_certification Scrum User Group Netherlands

- www.agilealliance.com
- www.scrum.org
- Agile&Iterative Development, Craig Larman, Addison-Wesley 2007
- <http://www.agilealliance.org/>
- <http://www.acm-software.com/>
- <http://vimeo.com/4587652>
- <http://www.yusufsahin.net/post/2009/12/05/Agile-Proje-Yonetimi-Scrum.aspx>
- http://en.wikipedia.org/wiki/Lean_manufacturing
- http://www.dailymotion.com/video/xec1mj_scrum-in-under-10-minutes_tech
- <http://www.mehmettargun.com/agile-proje-yonetimi-nedir/>
- <http://ccpace.com/Resources/documents/AgileProjectManagement.pdf>
- <http://objectwin.com/agile.aspx>
- http://www.versionone.com/Agile101/Agile_Benefits.asp
- <http://www.kubernetes.co.uk>
- <http://msdn.microsoft.com/en-us/magazine/dd347827.aspx>
- http://en.wikipedia.org/wiki/Microsoft_Solutions_Framework
- <http://www.mountaingoatsoftware.com/scrum-a-presentation>