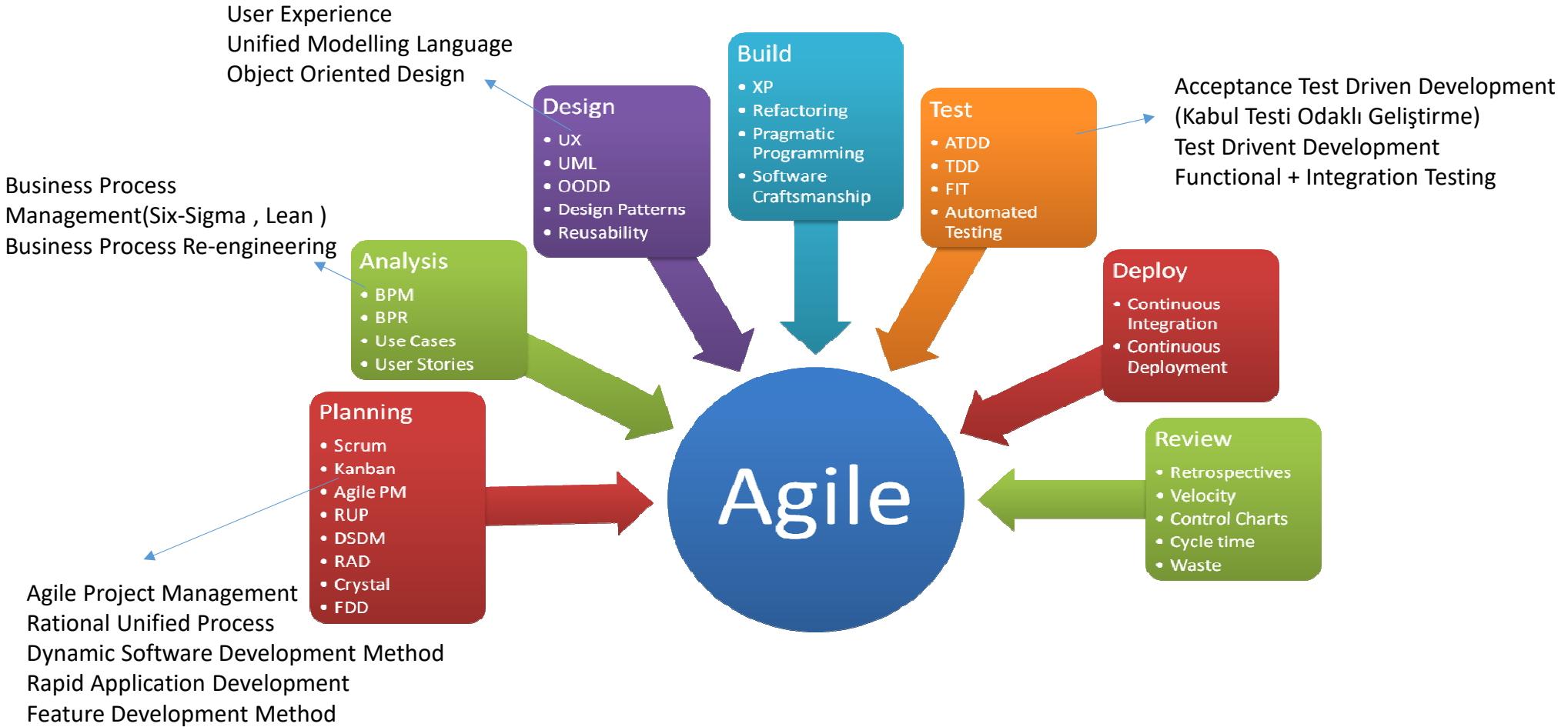


Yazılım Mühendisliğine (YM)  
Giriş

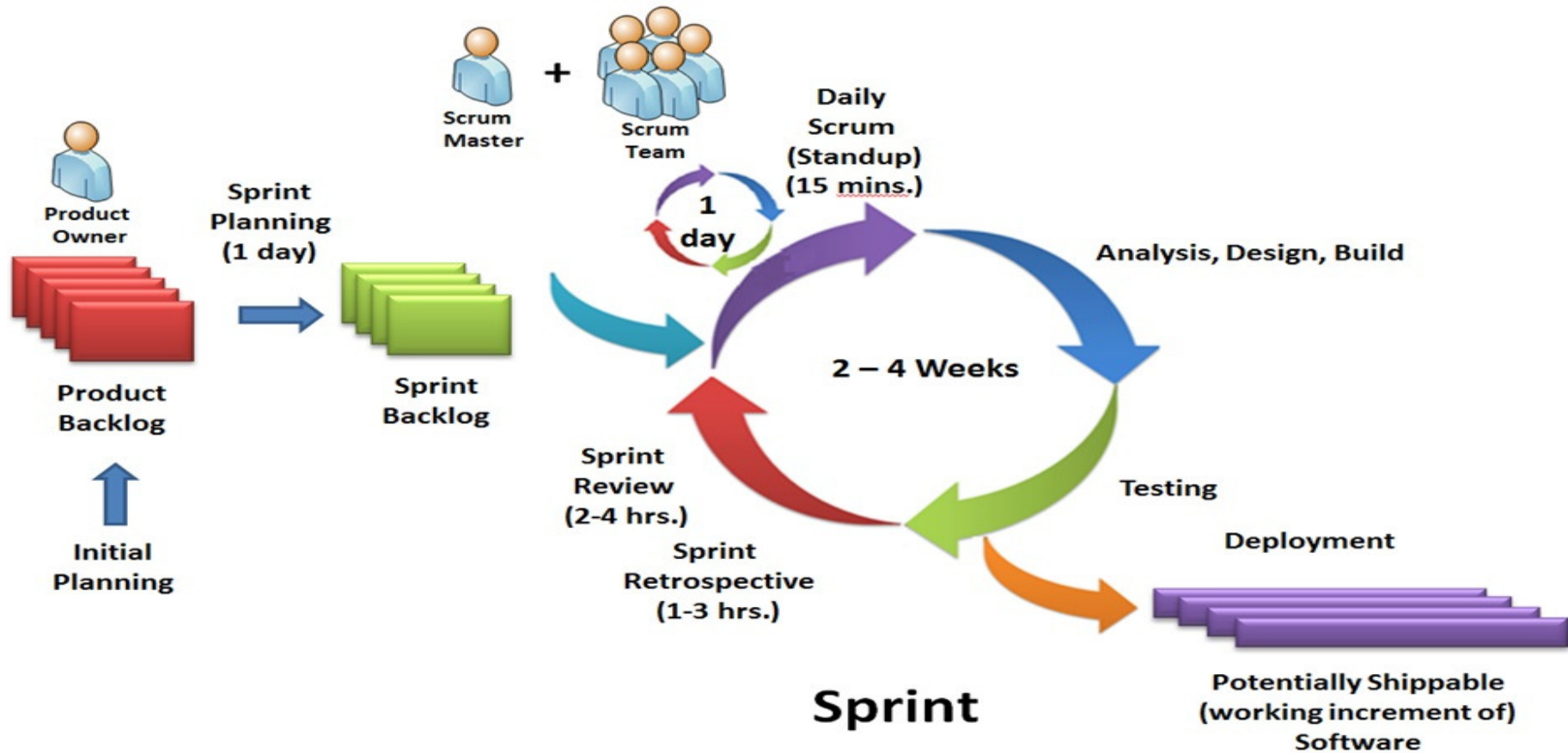
# Çevik Yöntemler

Ders Notu XI  
2023

# Çevik (Agile) Yazılım Geliştirme Süreçleri



# Çevik (Agile) Yazılım Geliştirme Örneği : Scrum





Visual Paradigm\* İnternet sayfası Agile dokümanı

\* UML tool community edition ücretsiz indirilir.

## Agile (Çevik) Nedir?

- ❑ Yazılım ürünü geliştirmeye yönelik yinelemeli bir yaklaşımdır.
- ❑ Proje ekiplerinin müşterilerine daha hızlı ve daha iyi değer sunmasına yardımcı olur.
- ❑ Çevik ilkelerin anahtarlarından biri, işi küçük, test edilebilir ve yönetilebilir ölçekte teslim etmektir.
- ❑ Agile, gereksinimleri önceden belirlemek yerine, bir proje boyunca gereksinimlerin sürekli olarak keşfedilmesini ve tanımlanmasını vurgular.
- ❑ Çevik ekipler değişime daha hızlı ve daha kolay tepki verebilir.

# Yazılım Projesinin Vizyonu (Hedefleri ve Amaçları) Ne Demektir?

- Bir yazılım projesinin vizyonu çalışmanın ilham kaynağıdır; projeye odaklanılmayı sağlar.
- Yazılım projesinin vizyonu projenin hedeflerini ifade eder.
- Tüm ekibin tüm proje boyunca aynı vizyonu kavramış olması, bunu paylaşması ve bu vizyon doğrultusunda çalışması önemlidir.

## Projenin Vizyon Nasıl Belirlenir?

- ❑ Projenin vizyonu, «paydaş(lar)» (stakeholders), «ürün sahibi» (product owner) ve «scrum master» içeren bir «proje vizyonu» toplantısı ile belirlenir.
- ❑ Bu toplantıda , etkili bir proje vizyon beyanı oluşturulur.
  - ❖ Projenin geliştirilmesi sırasında belirlenmesi önemli olan işin bağlamı (business context), iş gereksinimleri (business requirements) ve paydaş beklentileri (stakeholder expectations) belirlenir.
- ❑ Software Craftsmanship değilseniz bu toplulukta yeriniz şaibelidir.

# Scrum Projesinin Yönetimindeki Birimler



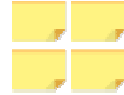
Project Vision



Stakeholder



Use Cases



Epics



Product Backlog



Sprint Backlog



Release Mgmt.



Scrum Board



Product Owner



Scrum Master



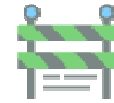
Scrum Team



Daily Scrum



Burndown



Impediment Log



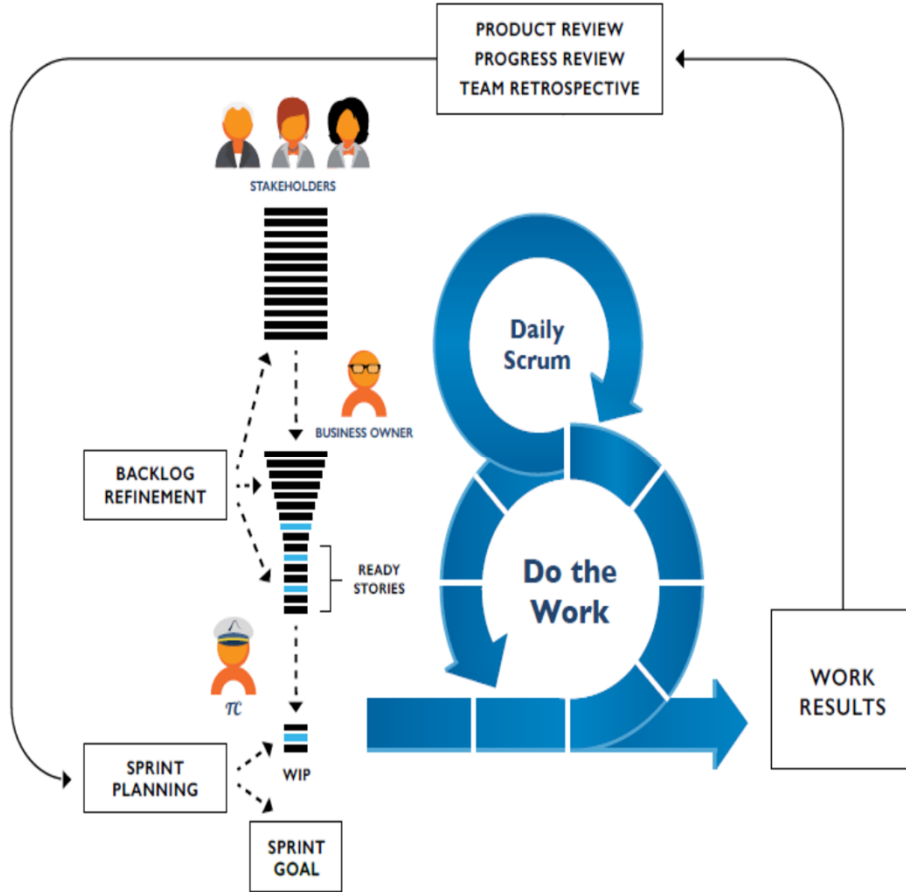
Sprint Review



Project Retrospective



# Scrum Yapısı / Scrum Artifact



❑ **Sprint Backlog (iş listesi)** : Çalışma sırasındaki ilerlemelerden oluşur. Sprint aşamasına getirilen kullanıcı hikayeleri ile ilgili olmalıdır.

- ❖ Bu hikayelerin çalışmaya eklenip eklenemeyeceğine ya da nasıl dahil edilebileceğine karar verilir.

# Stakeholder (Paydaş/İştirakçi)

Ürüne katkı veren herhangi bir kişidir.

- Ürün (product), süreç(process) veya ekipte(team) rolü olan bir kişi olabilir.
- Scrum ekibinin fazla önemsemediği, ama ürüne katkı sağlayan birisi de olabilir.
- Ürünün incelenmesi, yani gözden geçirilmesi (product review) sırasında ekibin geliştirdiği kısmı (ekibin çalışmasını) inceleyen kişi de olabilir.
- Takımın içinde olan, takımın çalışmalarından etkilenen veya takım üzerinde etkisi olan bir kişi olabilir.

Örneğin:

Paydaşlar, sprint süresince özellikle ürünün incelenmesi (Product Review) sırasında Backlog üzerine eklemeler yapabilir

Paydaşlar takımın önceliklerini değiştirmek istediklerinde Product Owner ile birlikte hareket etmelidirler.

## Sprint nedir?

- ❑ Bir geliştirme ekibinin incelemek üzere ürünün bir parçasını ürettiği sabit süre (bir aydan az).
- ❑ Sprint uzunluğu, ürün incelemeleri (ProductReview) arasındaki aralıkla tanımlanır
  - ❖ Bu aralıklar genellikle ardışık iki sprint arasında tutarlıdır ve Sprint başladıktan sonra değiştirilmemelidir.

# Product Review (Ürün İncelemesi)

❑ Sprint sonunda, geliştirme takımının (product owner ve paydaşlar ile birlikte) bundan sonra ne yapılacağı, neyin daha iyi yapılacağı vb. hakkında geri bildirim almak için ürünün mevcut parçalarını gözden geçirdiği bir geliştirme takımı oturumudur.

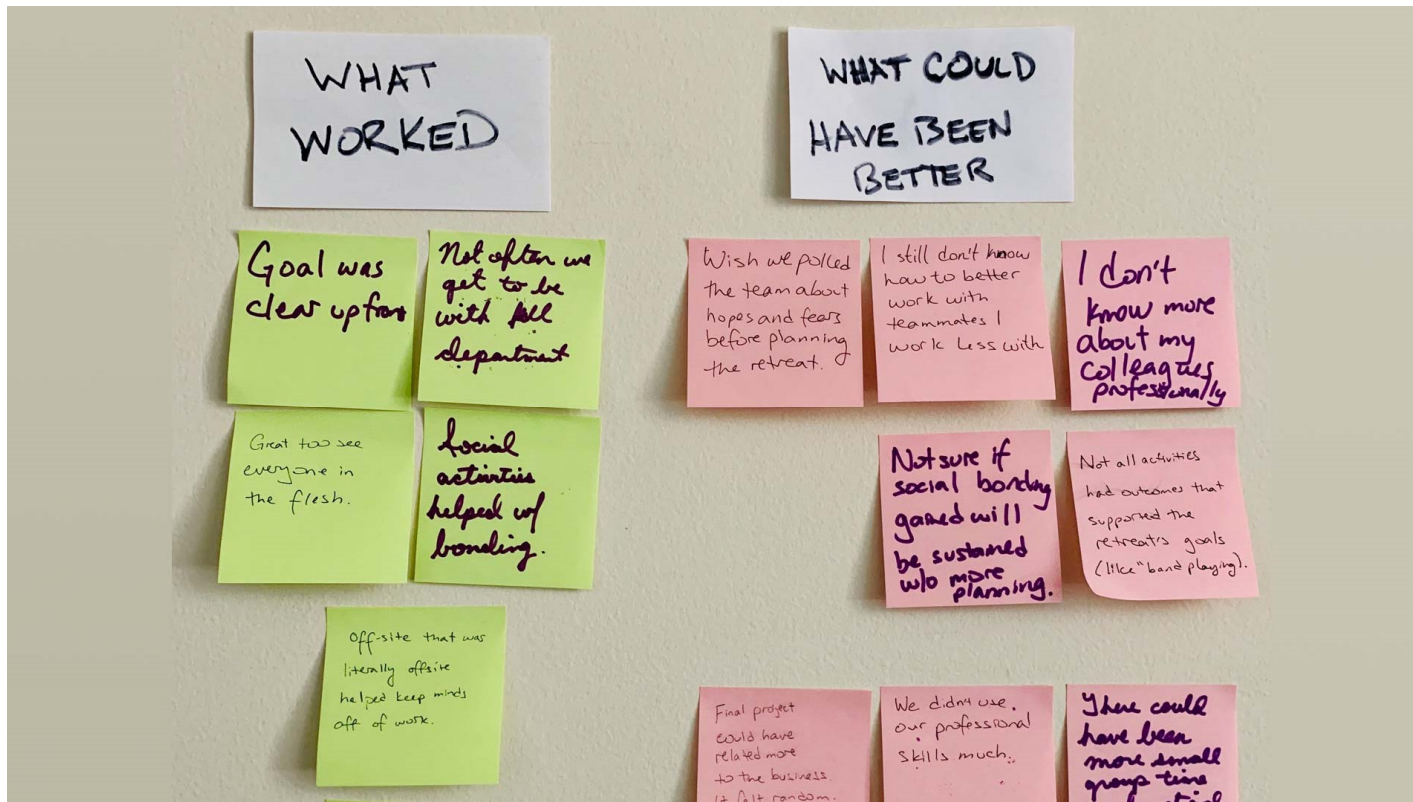
Örneğin:

- ❑ Sprint boyunca ve özellikle de ürün incelemesi (product review) sırasında, paydaşlar iş listesine (backlog) yeni öğeler (items) ekler.
- ❑ Ürün incelemesinden sonra, geliştirme ekibinin performansını ve uygulamalarını analiz etmek için dahili bir Retrospektif uygulanır.
- ❑ Uygulamada, ürün incelemesi (product review) genellikle hem ürün hem de ilerleme incelemelerini içerir
  - ❖ Gerçek uygulamalarda bu bir hata doğurabilir, yalnızca ürüne odaklanmak esas olmalıdır.
- ❑ Ürün incelemesinin (product review) amacı, ekibin müşteriler/kullanıcılar için ürünün ilgili parçası hakkında anlamlı geri bildirim sağlamasıdır.

# Product Owner (PO)

- ❑ Çevik yazılım geliştirme metodolojisinde olması gereken bir pozisyondur .
- ❑ Ürünün vizyonunu açık olarak anlaşılır yapan kişidir.
  - ❖ Ürünün vizyonu müşteri, son kullanıcı ve iştirakçiler bakış açısından sağlanır.
- ❑ Ürünün ticari ilişkilerini gerçekleştirir.
- ❑ Ürün «backlog» larının yönetiminden ve ürüne ait backlog görünürlüğünden sorumludur.
  - ❖ Backlog çevik yazılım geliştirme metodolojisinin vazgeçilmez terimidir.
  - ❖ Kısaca yapılacakların listesini verir.
  - ❖ Geliştirme takımından yapması istenilen her şeyi içerir.
    - ✓ Özetle geliştirilecek ürün ile ilgili her ayrıntıyı verir.

# Retrospective

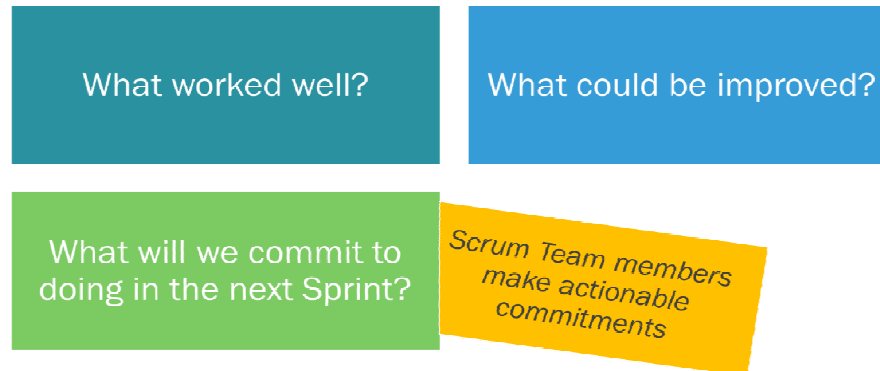


- What went well (keep doing these things)
- What could be improved (went OK, but could be better)
- What went badly (don't do these things again)
- Focus for next period/sprint/month/quarter (One or two things to focus on)

# Sprint Retrospective Toplantıları

- ❑ «**Sprint Retrospective**» toplantıları her bir «sprint» sonunda yapılır
- ❑ Takımlar her bir sprint implementasyonu tamamlanıp «**release**» işlemi gerçekleştirileceği zaman bu toplantıları yapar.
- ❑ Tüm takım elemanları, Product Owner ve Scrum Master katılımı ile gerçekleşir.
- ❑ Genellikle 1 saatlik toplantılardır.

## A Typical Sprint Retrospective Model





# Retrospective

**Coaching Team Retrospective – 15 July**

R&D Team Coaches ENT Team Visible

### Purpose and Context (5 mins)

#### Purpose

Purpose: share open and honest feedback so that we can identify how we can improve

#### Scope

Scope: our last Sprint across coaching projects and initiatives

#### Agenda

Agenda (60 mins total)

- \* Set purpose and context (5 mins)
- \* Review our ground rules
- \* What went well? (25 mins)
- \* What needs improvement? (25 mins)
- \* Actions (5 mins)

### Ground Rules

- Be as open and honest as possible
- Switch off and be present
- Ask clarifying questions
- Leave space in the Zoom for others
- Always assume positive intent
- Take turns sharing (top 3)
- + Any others?

Trello shortcuts: ADD YOURSELF: hover + space / VOTE: hover + v

### What went well? (10 mins)

What went well?

- Really enjoyed working as a cross-functional team with our Craft Learning teammates
- Strong alignment to our purpose and mission as a team
- LOVED our project kickoff – super productive and energetic way to start a project
- Really appreciate everyone's respect towards work/life boundaries
- We had really productive, but tough conversations we needed to have at the right time!

### What needs improvement? (10 mins)

What needs improvement?

- Priorities aren't super clear at the moment, which is challenging because we're getting so many requests for support
- We don't know how to say no
- Seems like we're facing some bottlenecks in our decision making
- Still some unclear roles and responsibilities as a leadership team

### Actions (5 mins)

Capture Actions (WHO will do WHEN)

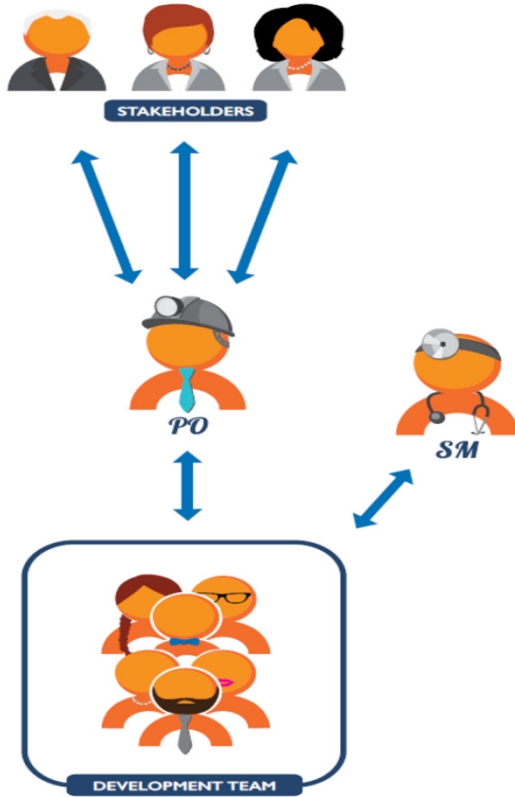
+ Add another card

Video call participants:

- Participant 1: Woman with glasses and a grey top.
- Participant 2: Woman with glasses and a purple top.
- Participant 3: Man with glasses and a black top.
- Participant 4: Man with a beard and a white top.
- Participant 5: Woman with glasses and a white top.



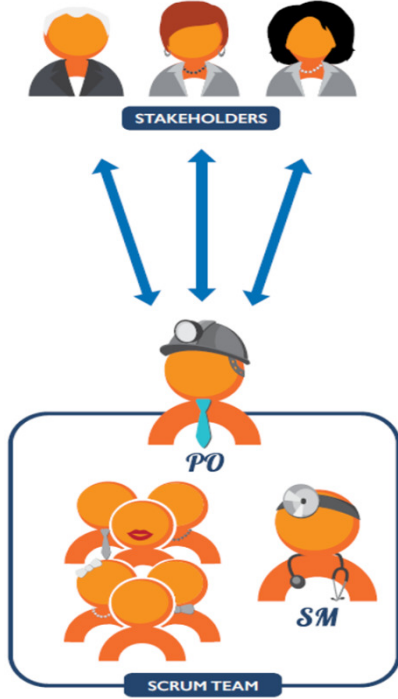
# Scrum 1.0



- ❑ **Product Owner** geliştirme takımının dışında olarak gereksinimlerin belirlenmesine sadece **Sprint** başında katkı sağlayabilir.
  - ❖ Geliştirme süresince Product Owner katkısı sadece başlangıç aşamasında olacaktır.
- ❑ Günümüzde uygulamaları devam etmektedir.
  - ❖ Fonksiyonel gereksinimlerin çok sık değişiklik göstermediği problemlerin çözümünde uygulanmaktadır.

# Scrum 2.0

❑ **Product Owner** her zaman çevik (agile) olmak zorundadır.

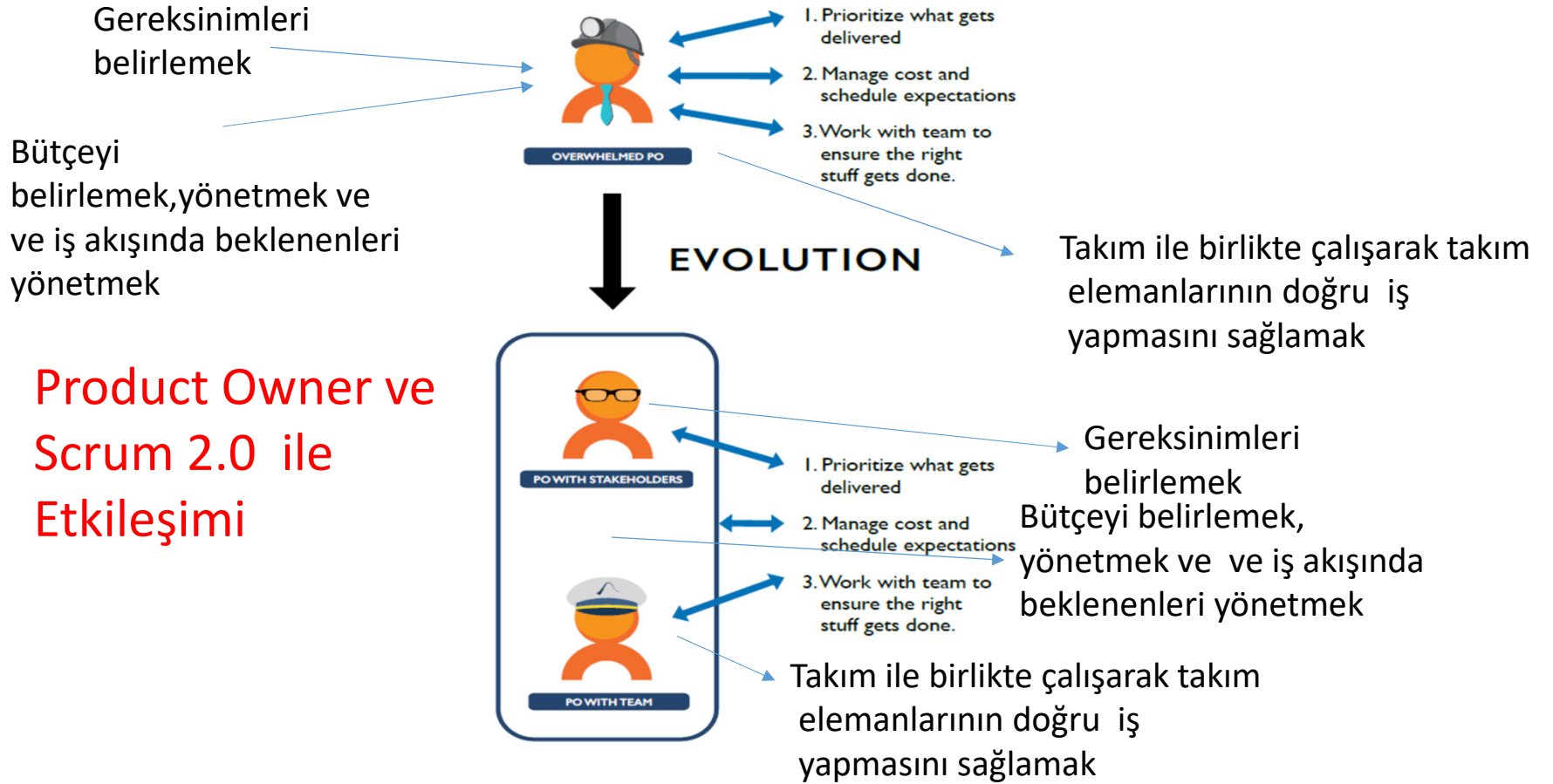


Çünkü :

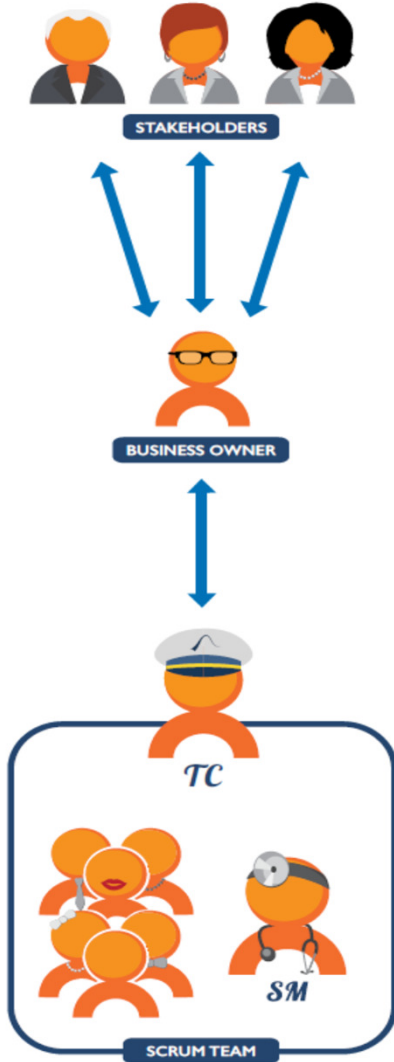
❑ Ürünün geliştirme takımının görevi sadece yazılım geliştirmek değil, geliştirilen ürünün devamlılığını da sağlamaktır.

❑ **Product Owner** yazılım ürününün geliştirilmesi aşamasında ürünün fonksiyonelliği ile ilgili karar vermek zorundadır.

- ❖ Buradan, Product Owner her zaman çevik olmak zorundadır.
- ❖ **Product Owner** bu işlevleri gerçekleştirirken iş (business) tarafında fazla karar verebilme olanağı bulamaz.
- ❖ Zira, geliştirme takımı **Product Owner** 'ı tümüyle geliştirme takımına çekmek ister.
  - ✓ Çünkü sürekli olarak değişen gereksinimler bunu gerektirir.



## Scrum 3.0



- ❑ Product Owner iki kişiliğe sahiptir. Business Owner ve Team Captain
- ❑ Stakeholders ile etkileşimi sağlayan Product Owner bu versiyonda Business Owner olarak adlandırılır.
- ❑ Product Owner, Stakeholders isterlerini sürekli olarak geliştirme takımına aktarma görevi ile zamanın çoğunu çevik olarak takım içinde geçirir.
  - ❖ Bu rolü ile **Team Captain** rolünü gerçekleştirir.

# Scrum 1.0 & Scrum 2.0 & Scrum 3.0

## *Scrum 1.0*

- ❖ Product Owner, stakeholders (paydaşlar / iştirakçiler) ile birlikte.
- ❖ «Stakeholders» (katılımcılar) isteklerini geliştirme takımına aktarır.
- ❖ **Sprint** süresince öncelikleri değiştiremez.

## Scrum 2.0

- ❖ Product Owner geliştirme takımının bir elemanıdır ve «stakeholders» (katılımcılar) ile iletişimi devam ettirir.
- ❖ **Product Owner** geliştirme takımı ile birlikte çalışabilir ve **Sprint** süresince öncelikleri değiştirebilir.

## Scrum 3.0

- ❖ Product Owner geliştirme takımında Team Caption rolünde, geliştirme takımının dışında Business Owner rolündedir.
- ❖ Geliştirme takımı ile Team Caption olarak çalışarak **Sprint** süresince öncelikleri değiştirebilir.



- ❑ DevOps is a methodology in the software development and IT (Information Technology) industry.
- ❑ Used as a set of practices and tools (Bir dizi pratikler, yani uygulamalar ve araçlar kullanılır).
- ❑ DevOps integrates and automates the work of software development (Dev) and IT operations (Ops)
  - ❖ Amacı sistemlerin geliştirilme yaşam döngülerini kısaltmak ve daha gelişmiş (çağdaş) hale getirmektir.
- ❑ DevOps önemli bir kültürel değişimdir.
- ❑ Geliştirme ve dağıtım (teslim) süreçleri sırasında operasyonların, geliştiricilerin ve test uzmanlarının işbirliği yapma biçimini dönüştürüyor.
  - ❖ Bu grupların uyumlu bir şekilde çalışmasını sağlamak, DevOps'un kurumsal bir kimlik olarak kabulünde kritik bir zorluktur.

# Özetle DevOps

Dev takımının sorumlulukları:

Uygulamaya ait planların yapılması,

Uygulamanın kodlanması (oluşturması),

Uygulamanın versiyonları (release) ve yayınlanması (publish) süreçlerinin gerçekleştirilmesi,

Uygulama iyileştirmesi (update),

Uygulamanın test süreçleri.

Ops takımı sorumlulukları:

Oluşturulan uygulamaların kullanılacağı ortamın tasarlanması,

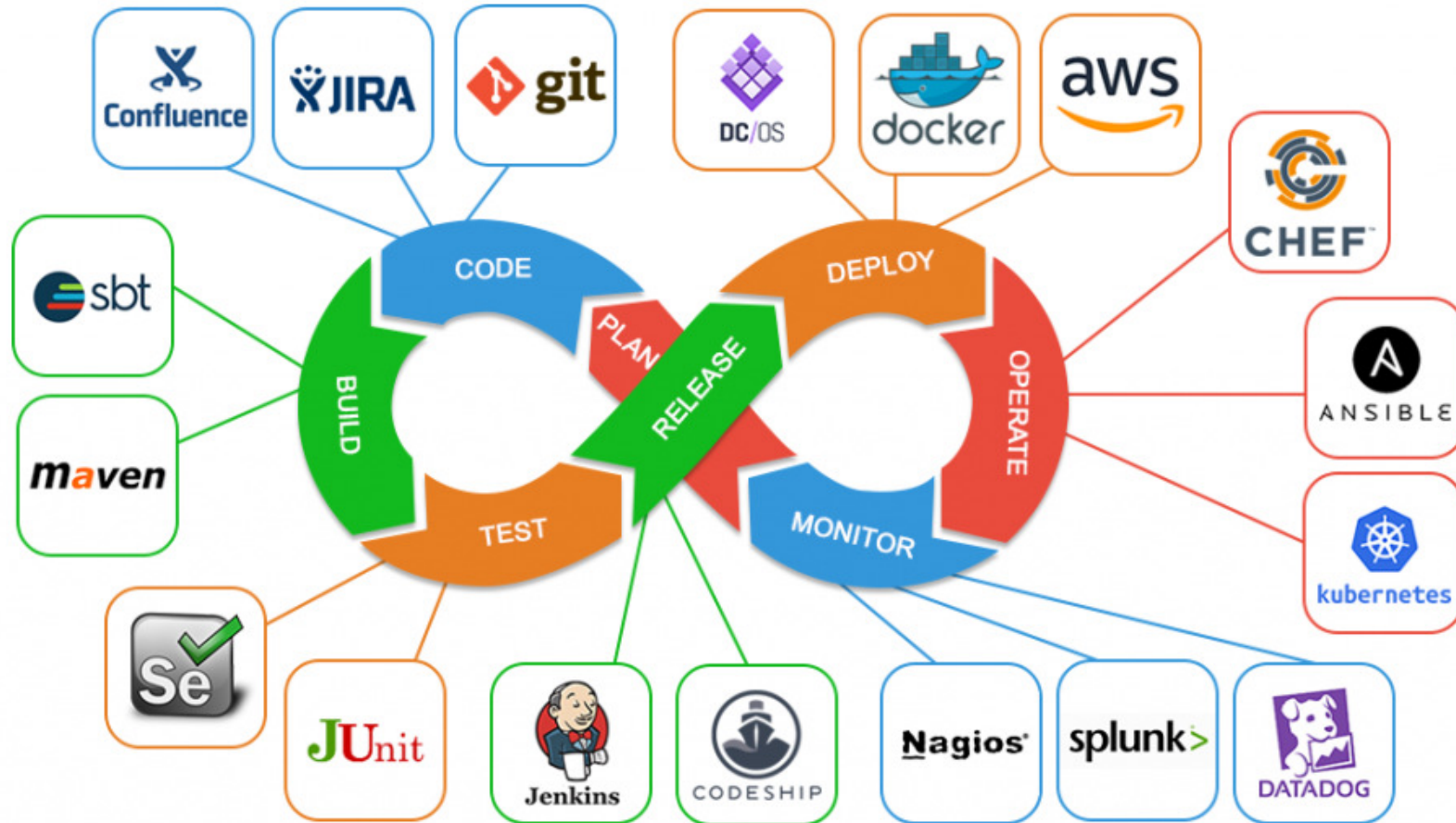
Uygulamaların çalışması için gerekli sistem bileşenleri ile iletişime geçebilmeleri için gerekli ağ ve güvenlik yapılandırmalarının temini,

Uygulamanın kaynak kullanımının belirlenmesi,

Uygulamanın izleme (monitoring) araçları ile takibinin sağlanması,

Uygulamanın sistem kaynaklarını kullanım düzeyine göre kaynak artırımının sağlanması (scale up ve scale down).

# DevOps Araçları (tools)

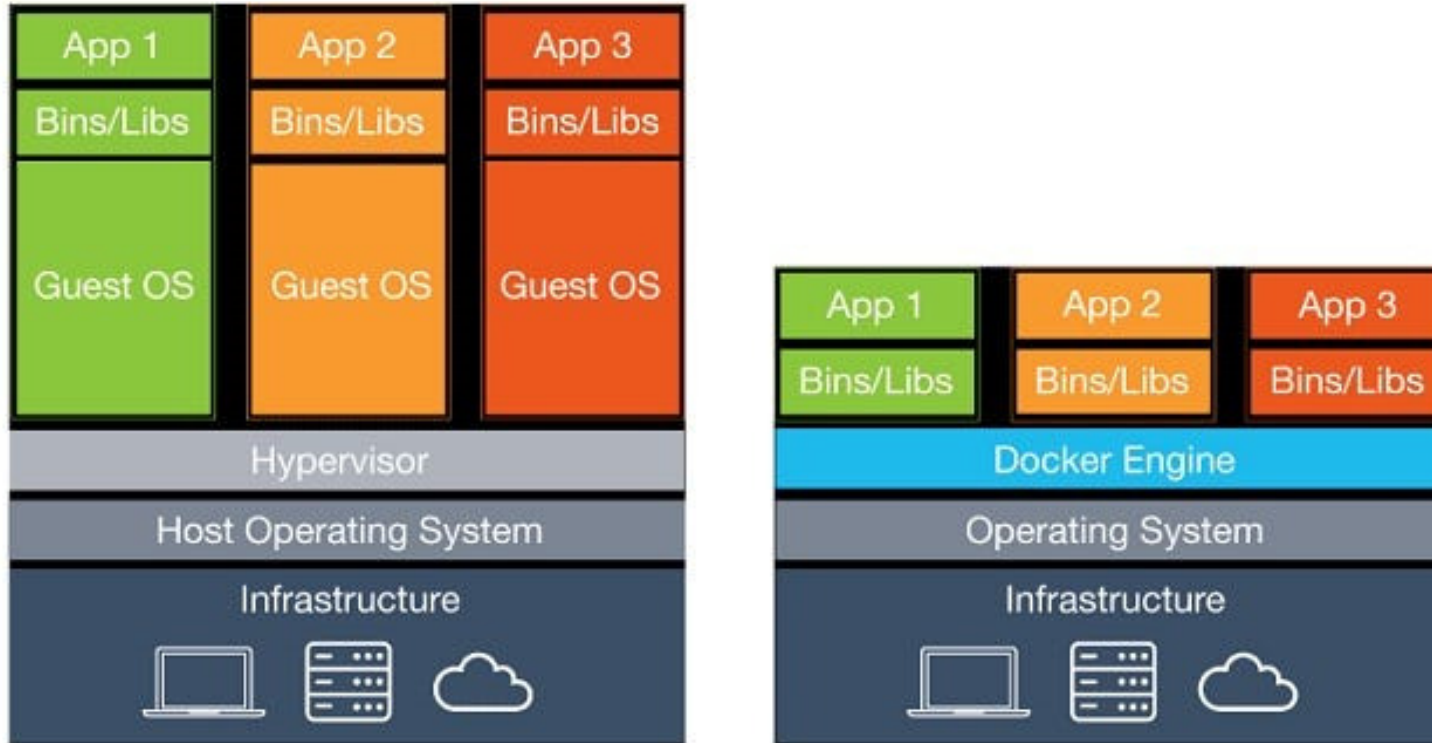




# .NET – A unified platform



## Sanal Sunucu ve Container Mimarisi



Kubernetes Google tarafından GO dilinde geliştirilmiş konteyner haline getirilmiş uygulamaları otomatik deploy etmek, sayılarını arttırıp azaltmak gibi işlemler ile birlikte yönetilmesini i sağlayan bir Konteyner kümeleme (container cluster) aracıdır.