



BATCH :
LESSON :
DATE :
SUBJECT :

146 - 149

SDLC

10.07.2023

SDLC - 1

ZOOM GİRİŞLERİNZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ





SDLC

Software || System Development Life Cycle

1. Ders
10.07.2023

B149 AWS & DevOps
B146 Cyber Security

Bugün ne yapıyoruz?

- SDLC nedir?
- SDLC' nin önemi ve faydaları nelerdir?
- SDLC' nin temel aşamaları nelerdir?
- SDLC takımı kimlerden oluşur?
- SDLC takımı görevleri nelerdir?
- Cyber Security için SDLC
- DevOps için SDLC
- Kahoot



SDLC

High-Quality

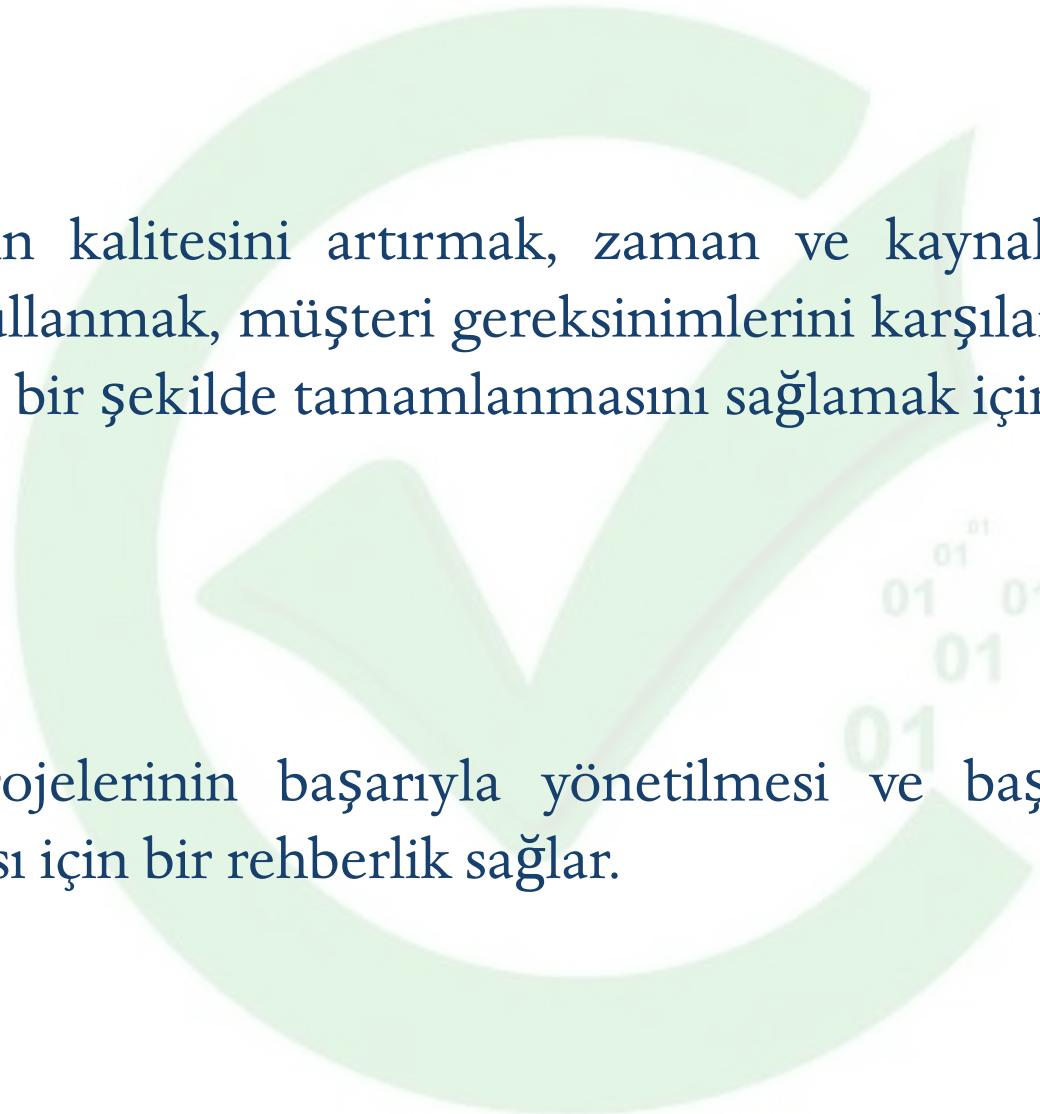
User Expectation



- Yüksek Kaliteyi (High-Quality) ve Kullanıcı Taleplerini (User Expectation) karşılayan Yazılım ürünlerini geliştirmek için planlanan yazılım geliştirme sürecidir. (Process)
 - SDLC' nin ana hedefi, bir yazılım projesinin etkili ve sistematik bir şekilde yönetilmesini sağlamaktır.

SDLC

- Bu süreç, yazılımın kalitesini artırmak, zaman ve kaynakları etkin bir şekilde kullanmak, müşteri gereksinimlerini karşılamak ve projenin başarılı bir şekilde tamamlanmasını sağlamak için bir çerçeve sunar.
- SDLC, yazılım projelerinin başarıyla yönetilmesi ve başarılı sonuçlara ulaşılması için bir rehberlik sağlar.



SDLC



Knights \$440 million hatası

Yazılımda bulunan bir hata sonucu, Amerikan borsasının en büyük firmalardan bir tanesi 30 dk içerisinde 440\$ kayıp yaşadı. Firma hisse senetleri 2 gün içerisinde %75 değer kaybetti.



NASA – The Mars Climate Orbiter

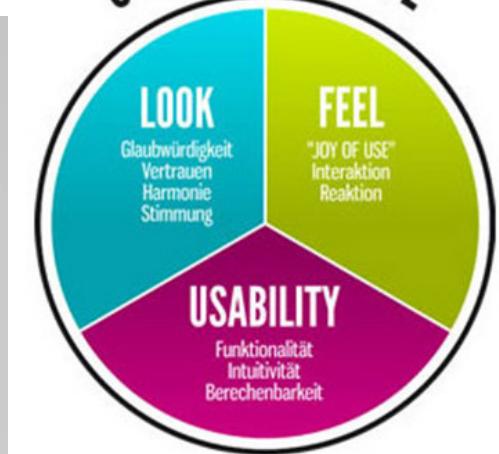
Bu \$125 milyon dolarlık uzay projesi, yazılımın Amerikan metriklerini Avrupa metriklerine doğru çeviremediğinden dolayı uzayda kaybolmuştur.



SDLC

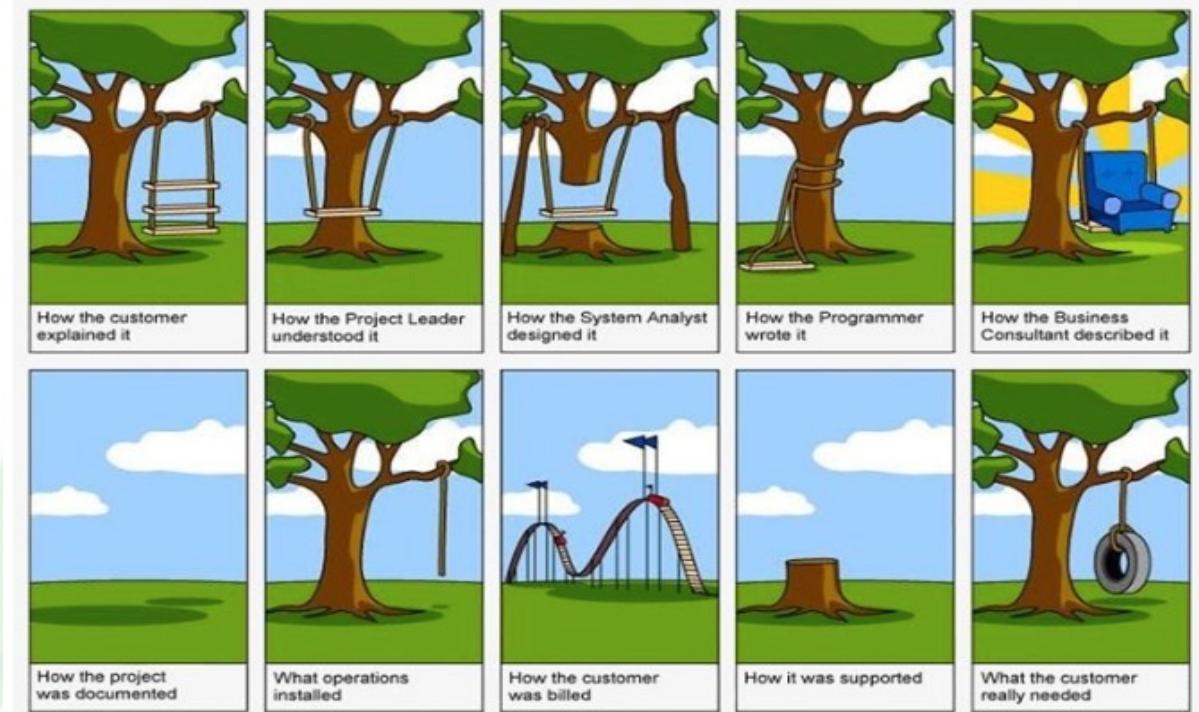


USER EXPERIENCE



Önemi ve Faydaları

- Proje Yönetimi
- Yazılımın Kalitesi
- Zaman Çizelgesi
- Bütçe Yönetimi
- SDLC' de her bir aşama, belirli görevler, çıktılar ve girdilerle birlikte çalışan ekipler ve roller içerir.



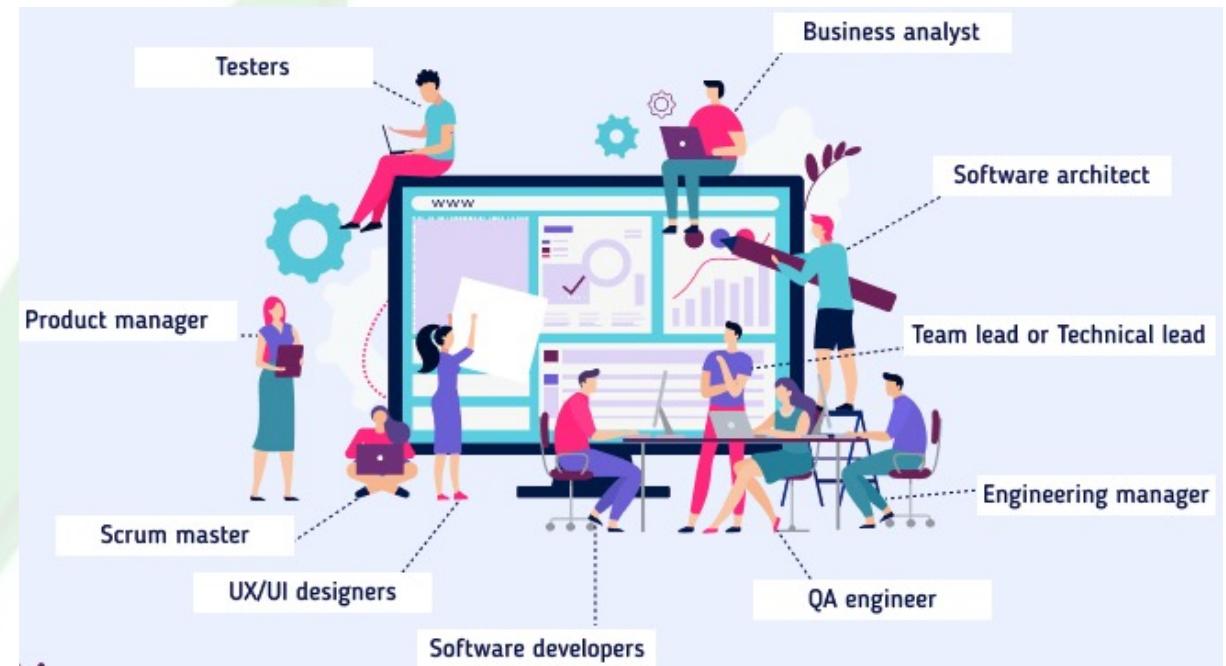
Temel Aşamaları

- **Planlama** : Beklentileri Tespit Etme
- **Analiz** : Amaç ve Gereksinimleri
- **Tasarım** : Proje Tasarımı
- **Uygulama** : Geliştirme
- **Test** : Hedef ve Geliştirme Doğrulama
- **Bakım**: Yayınlama ve Geri Dönüşler ile Güncellemeler



Takım Üyeleri

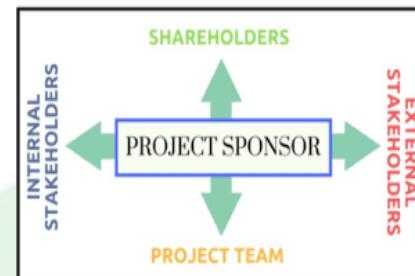
- Project Manager (PM) : Proje Yöneticisi
- Business Analyst (BA) : İş Analisti
- Developer (DEV) : Yazılımcılar
- Quality Analyst (QA) : Kalite Analisti Tester



Project Manager (PM)



Proje Planlanması



İletişim Yönetimi



Risk Yönetimi



Kaynak Yönetimi

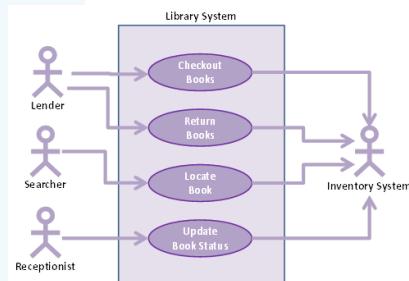


Ekip Yönetimi

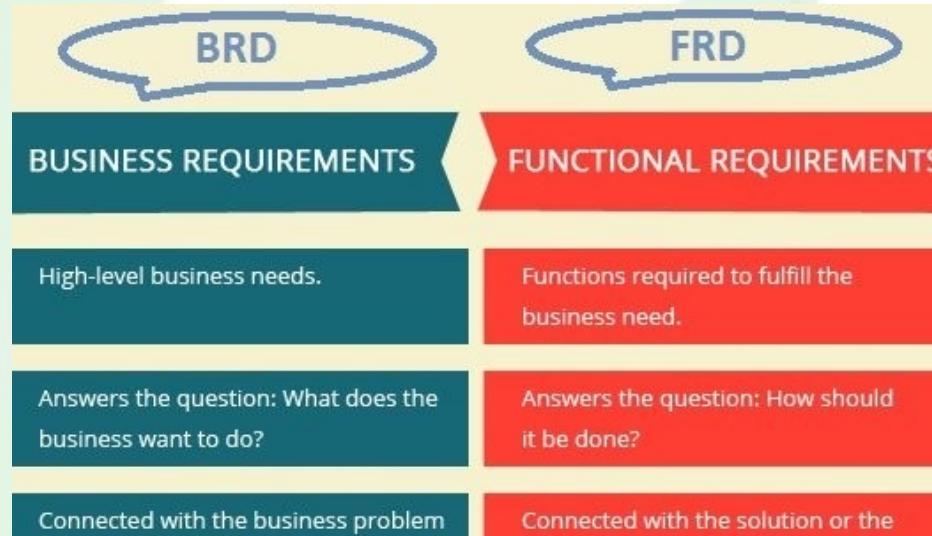
Business Analyst (BA)



Requirements Yönetimi



Use Cases Oluşturma



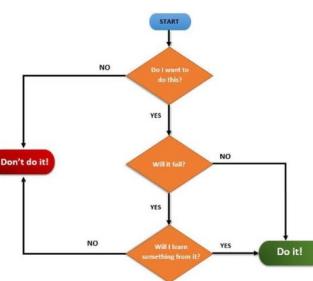
FRD
Functional Requirement Document
Hazırlama



Business Requirements Document

BRD

Business Requirement Document
Hazırlama



Akış Şeması Oluşturma

Developer (DEV)



Kodlama



Yazılım Tasarımı



Hata Ayıklama



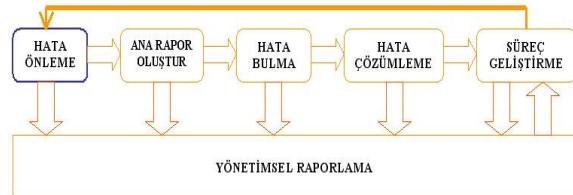
İşbirliği

Öncelikli amaçları son kullanıcı odaklı olarak yeni yazılımlar oluşturmak ya da hali hazırda var olan yazılımların geliştirilmesini sağlamaktır.

Quality Analyst (QA)



Test Planlama



Hata Tespiti
Raporlama



Kalite Standartlarının
Uygulanması



Otomasyon Yönetimi

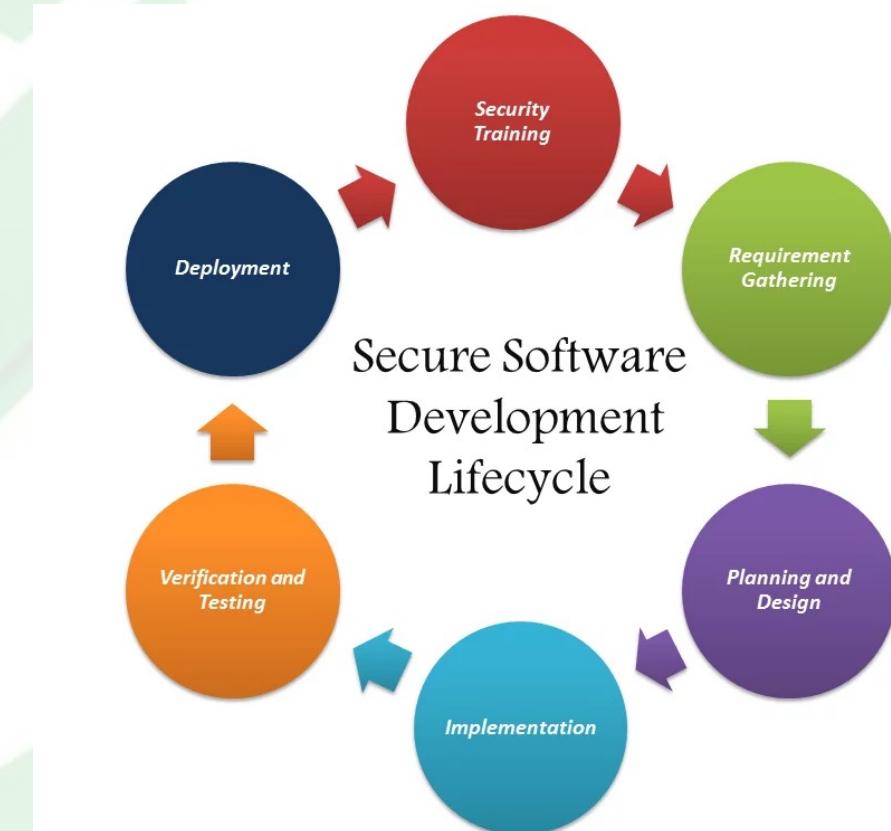
Kalite kontrolü sağlayacak şekilde Software Testing Life Cycle'in bütün prensiplerini uygulamakla yükümlüdür.

Cyber Security

Özellikle güvenlik odaklı bir yaklaşım benimsenen projelerde Cyber Security' nin katılımı daha belirgin olur.

- **Planlama** : projenin başlangıcında güvenlik gereksinimlerini belirlemek ve analiz etmek için katkıda bulunabilir.
- **Tasarım**: Güvenlik açıklarını belirlemek, riskleri azaltmak ve güvenlik önlemleri önermek için tasarım incelemeleri yapar.
- **Uygulama**: Bu aşamada sızma testleri, zafiyet taramaları ve kod analizleri gibi güvenlik kontrolleri gerçekleştirilebilir.
- **Test**: güvenlik açısından zayıf noktaların belirlenmesi ve düzeltilmesi için işbirliği yapar.
- **Bakım**: güvenlik olaylarına hızlı bir şekilde yanıt verir ve güvenlik politikalarının uygulanmasını sağlar.

Cyber Security mühendisi, SDLC' nin tüm aşamalarında güvenlik perspektifini göz önünde bulundurarak, projenin güvenlik açısından sağlam bir temel üzerine inşa edilmesine katkıda bulunur.

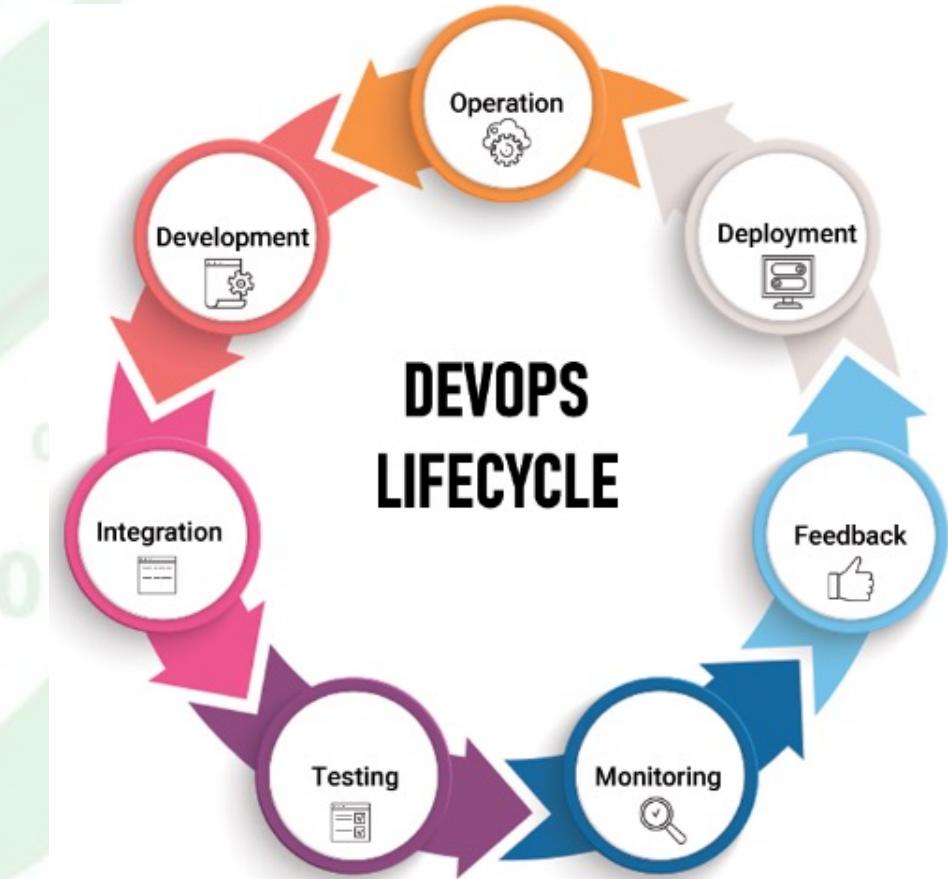


DevOps

DevOps mühendisi, SDLC' nin farklı aşamalarında aktif olarak yer alır ve bu aşamalar arasında koordinasyon sağlar.

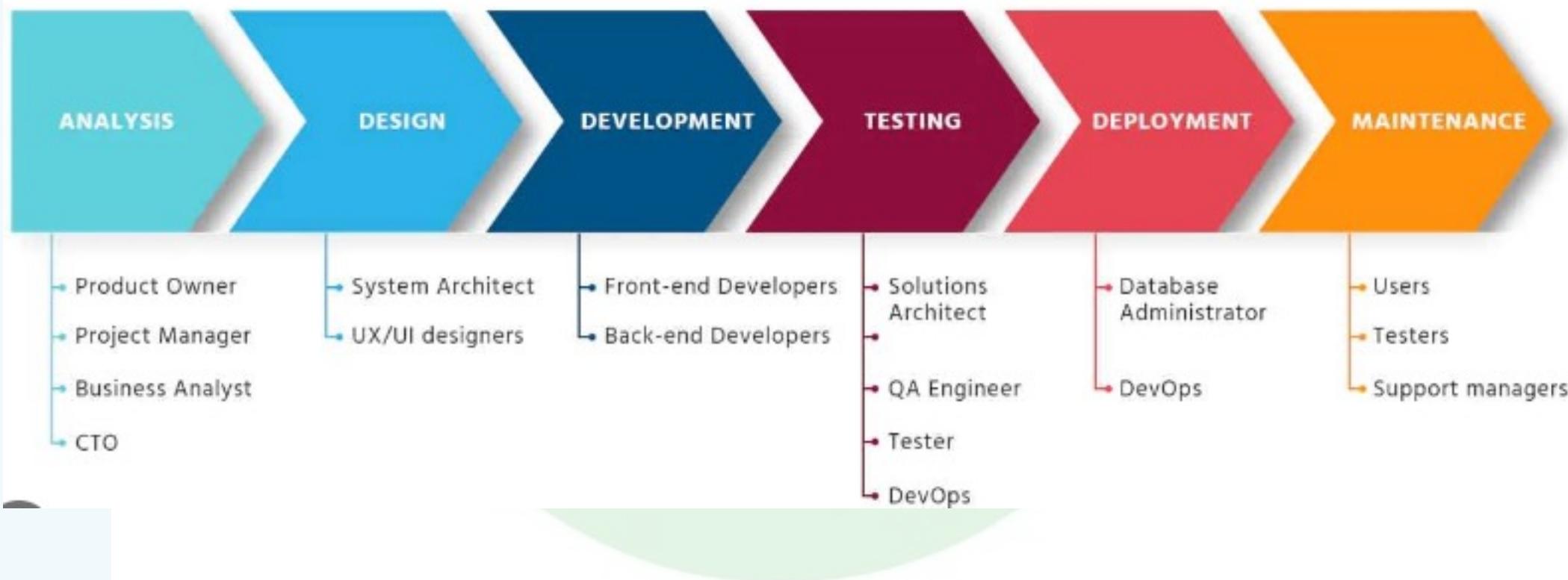
- **Planlama** : Geliştirme ekibine işletme perspektifi sağlayarak, operasyonel gereksinimlerin ve kısıtlamaların proje tasarımlına yansıtılmasını sağlar.
- **Tasarım** : Otomatikleştirilmiş dağıtım süreçlerinin oluşturulması, konteynerizasyon, yapılandırma yönetimi ve izleme araçlarının seçimi gibi konularda rol alır.
- **Geliştirme** : Sürekli entegrasyon (CI) ve Sürekli dağıtım (CD) süreçlerinin oluşturulmasında önemli bir rol üstlenir.
- **Test** : Birim testleri, bütünlük testleri ve performans testleri gibi çeşitli testlerin otomatik olarak gerçekleştirilmesi için altyapıyı sağlar.
- **Bakım** : Yazının canlı ortama güvenli ve sürekli bir şekilde dağıtılmmasını sağlar. Sürekli dağıtım süreçlerinin otomatik olarak çalışmasını ve hataların hızlı bir şekilde tespit edilmesini sağlar.

DevOps mühendisi, yazılımın hızlı bir şekilde geliştirilmesi, teslim edilmesi ve işletilmesi süreçlerinde ekipler arasında koordinasyon sağlayarak, sürekli iyileştirmeye odaklanır ve yazılımın kalitesini artırmak için operasyonel etkinlikleri optimize eder.



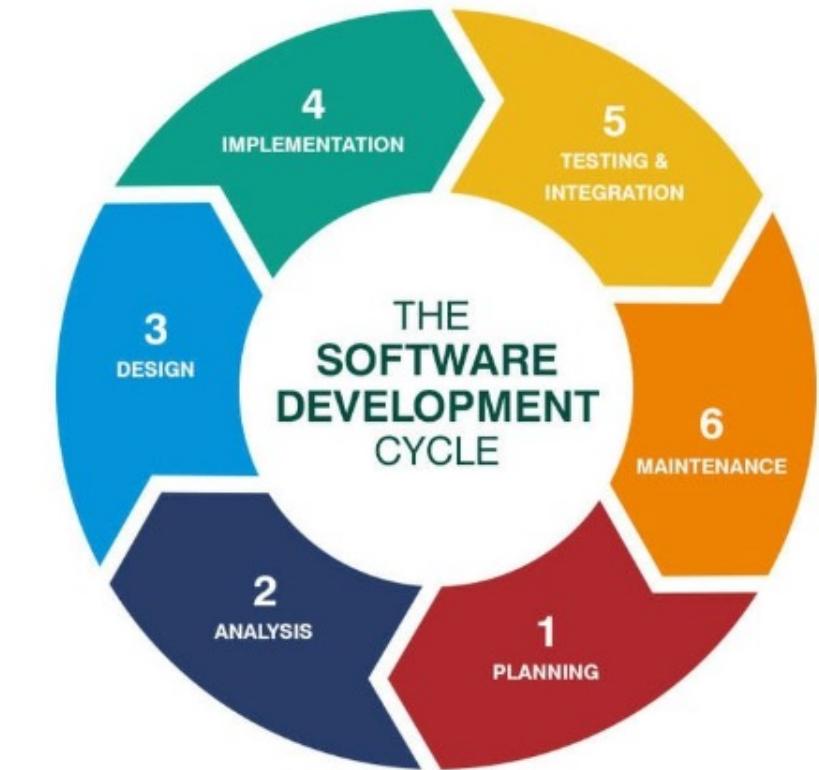
Summary

6 PHASES OF THE SOFTWARE DEVELOPMENT LIFE CYCLE



Temel Aşamaları

- **Planlama** : Beklentileri Tespit Etme
- **Analiz** : Amaç ve Gereksinimleri
- **Tasarım** : Proje Tasarımı
- **Uygulama** : Geliştirme
- **Test** : Hedef ve Geliştirme Doğrulama
- **Bakım**: Yayınlama ve Geri Dönüşler ile Güncellemeler



Planlama (Planning)

- İlk toplantıda **HIGH LEVEL** hedefler belirlenir.
- Müşteriden gelen fikirler de göz önünde bulundurularak ekibin kıdemli üyeleri (**expert**) tarafından gerçekleştirilir.
- Kalite güvence gerekliliklerinin planlanması ve projeye ilişkili risklerin belirlenmesi de planlama aşamasında yapılır.



Analiz (Defining)

- Alınan kararların DOCUMENT edilerek raporlandırma süreci yapılır ve stakeholder onayına sunulur.
- BA tarafından proje yaşam döngüsü boyunca tasarlanacak ve geliştirilecek;
 - tüm ürün gereksinim dokümanı “BRD” ve
 - teknik ihtiyaçlar dokümanı olan “FRD” raporları yazılır.
- BRD ve FRD en küçük User Case’ lere kadar hazırlanır.



Tasarım (Designing)

- BRD' de belirtilen gereksinimlere dayanarak, ürün mimarisi için genellikle birden fazla tasarım yaklaşımı taslağı oluşturulur.
- Bir **Tasarım Belgesi Spesifikasyonu** yazarak; **DDS (Design Document Specification)** 'nda belgelendirilir.
- **System Architecture** ve **UI/UX Designer** aktif rol alır.



Geliştirme (Developing)

- SDLC' nin bu aşamasında **Dev Team** tarafından gerçek geliştirme aşamalarında geçilerek ürün inşa edilir.
- Kodlama için FRD raporu baz alınır, Developerlar gereken Funcionality' leri oluştururlar.
- Bu aşamada **Developer** aktif rol alır.



Test (Testing)

- Ürün BRD' de tanımlanan kalite standartlarına ulaşıncaya kadar, ürün kusurlarının rapor edildiği, izlendiği, düzeltiği (**fixing**) ve tekrar test edildiği aşamadır.
- Ürün iş beklentilerini de karşılamalıdır.
- Bu aşamada **QA Tester** aktif rol alır.
- **DevOps Team** de dolaylı olarak dahil edilebilir.



Bakım (Deploying & Maintenance)

- Ürün test edildikten ve onaylandıktan, resmi olarak uygun görülen şekilde release edilir.
- Ürün piyasaya sunulduktan sonra mevcut müşteri tabanı için monitoring süreçleri ile bakımı yapılır.
- Müşteriden (**End-User**) gelen **feedback** ve **teknolojik gelişmeler** ile ihtiyaçlar yeniden belirlenir ve döngü yeniden başlatılır.
- Bu aşamada **DevOps Team** aktif rol alır.
- **Cyber Security Team** de dolaylı olarak dahil edilebilir.





BATCH :
LESSON :
DATE :
SUBJECT :

146 - 149

SDLC

11.07.2023

SDLC - 2

ZOOM GİRİŞLERİNZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ





SDLC

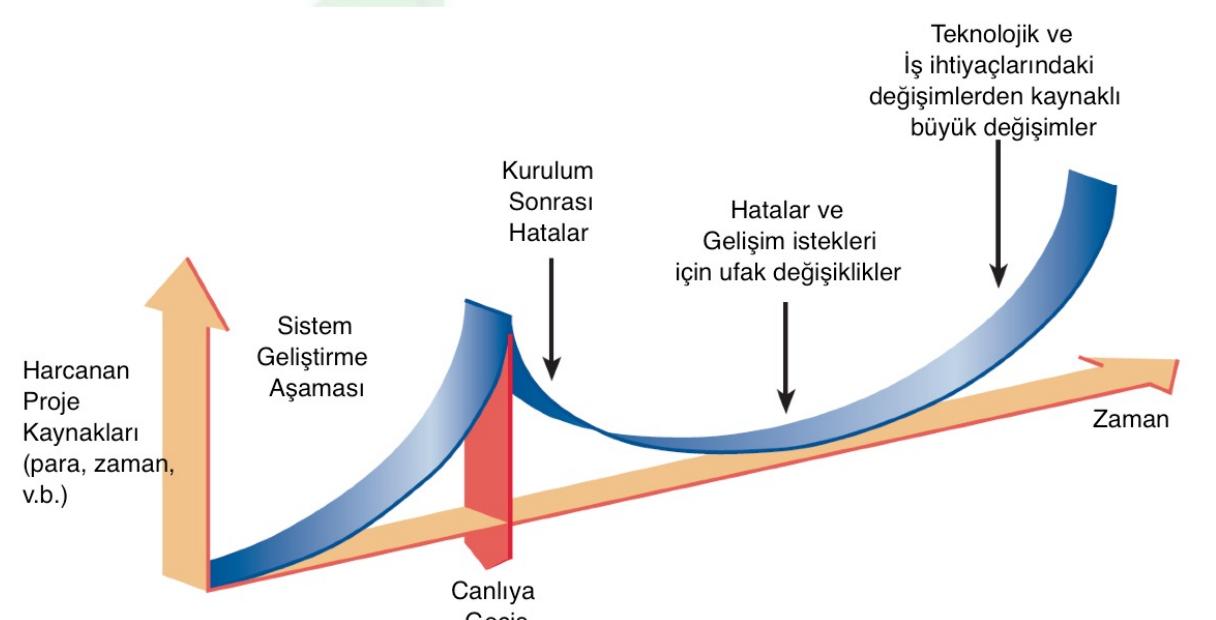
Software || System Development Life Cycle

2. Ders
11.07.2023

B149 AWS & DevOps
B146 Cyber Security

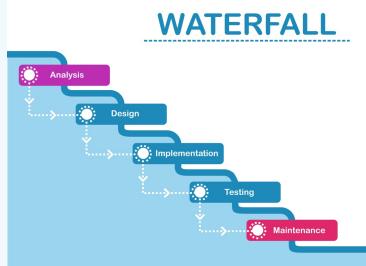
Bugün ne yapıyoruz?

- SDLC Metodları nelerdir?
 - Waterfall
 - V Model
 - Spiral
 - Prototip
 - Agile
 - Scrum nedir?
 - Scrum Team
 - Product Owner (PO)
 - Scrum Master
 - Development Team
- Kahoot

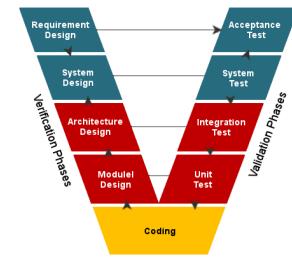


Kendall & Kendall, System Analysis and Design kitabından Türkçe'ye çevrilmiştir. Kaynak: www.MISSozluk.com

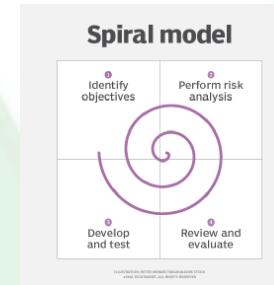
SDLC Metodları



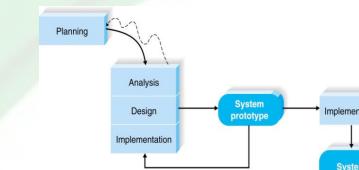
Waterfall
(Şelale)



V Model



Spiral



Prototip



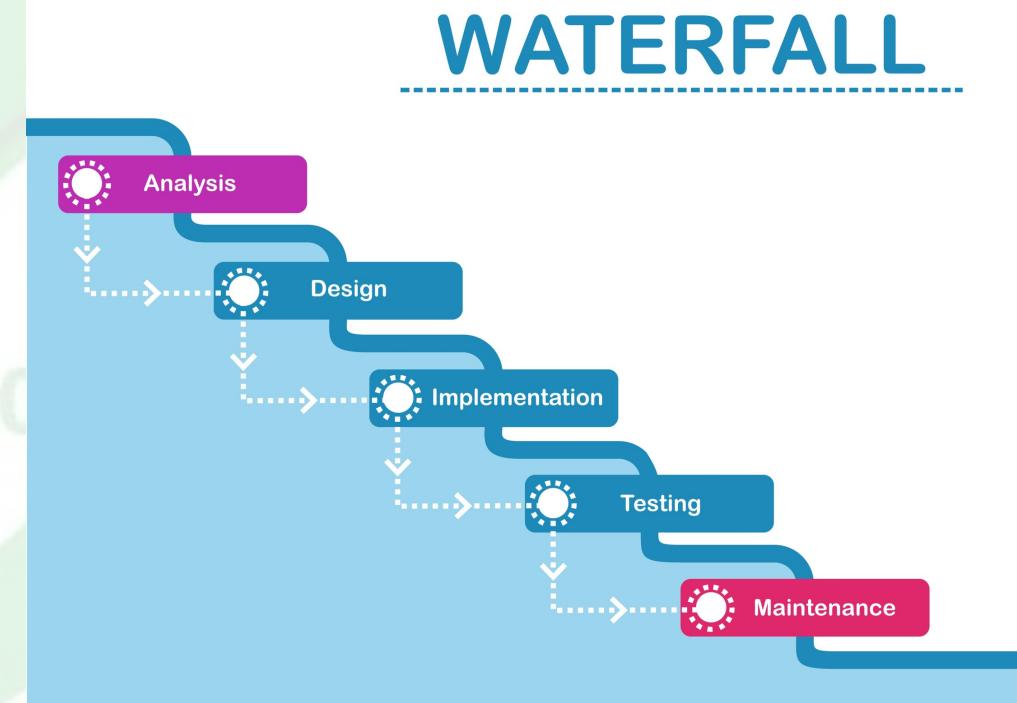
Agile
(Çevik)

Yazılım geliştirme modeli yazılımın gerçekleştirilebilmesi için gerekli stratejiyi ifade eder ki bu strateji bir dizi aktiviteyi ve olayları içermektedir.

Dikkat edilmesi gereken nokta çözüm istenen ürüne ve sürece uygun modelin seçilebilmesidir.

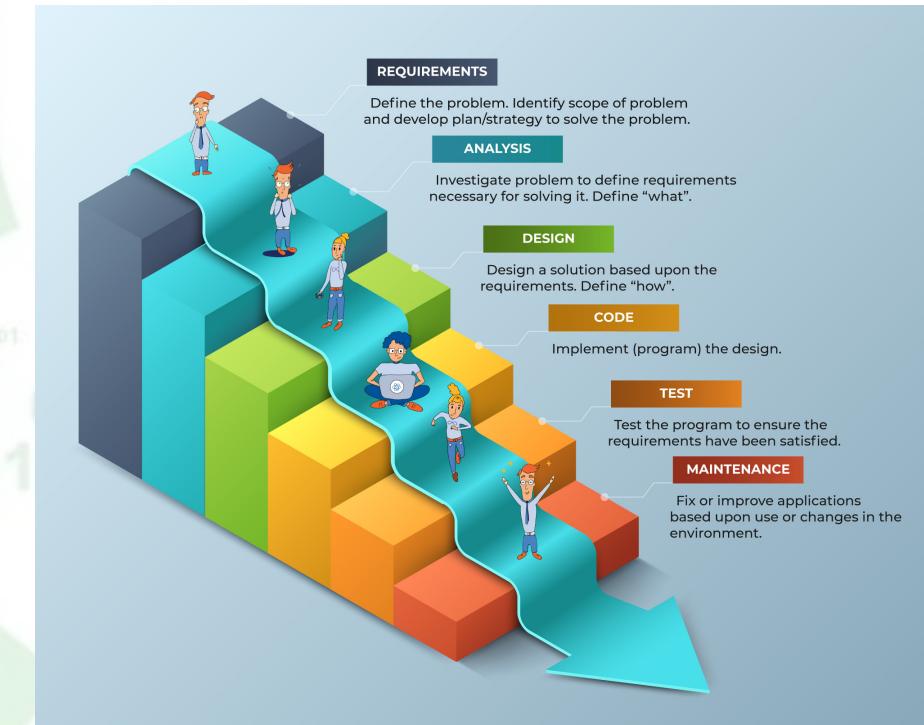
Waterfall (Şelale) Methodology

- Şelale modeli (Waterfall) proje yönetim süreci; analiz, tasarım, yazılım, test, yayın gibi fazlardan oluşur.
- Geleneksel bir yöntemdir; süreçler tipki bir şelale gibi yukarıdan aşağıya doğrusal olarak işler.
- Bir faz tamamlanıp yenisine geçildiğinde, bir önceki faza geri dönülmez.
- Stakeholder, **proje tamamlandıktan sonra** ürünü görebilir.



Waterfall (Şelale) Methodology

- Analiz adımı ile başlar. Analiz adımda, tüm yazılım gereksinimleri net bir şekilde belirlenerek analiz dokümanı üretilir.
- Daha sonra, tasarım adımda yazılımın arayüz, veritabanı, sınıf vb. tasarımları yapılarak tasarım dokümanı üretilir.
- Bir sonraki kodlama adımda yazılım analiz ve tasarım dokümanlarında belirtilen şekilde kodlanır.
- Test adımda analiz ve tasarım dokümanlarındaki tüm fonksiyonel ve fonksiyonel olmayan gereksinimler ve tasarımlar için test senaryoları yazılır ve bu test senaryoları icra edilerek yazılımın testleri yapılır.
- Test adımı sonunda, yazılımda herhangi bir hatası bulunamaz ise, entegrasyon adımına geçilir ve yazılım, canlı ortama entegre edilerek müşterinin kullanımına açılır.



Waterfall (Şelale) Methodology

Avantajlar

- Proje bilgisini aktarmak daha kolaydır.
 - Projeyi yönetmek daha kolaydır.
 - Küçük projeler için daha iyidir.
 - Görevler mümkün olduğunca sabit kalır.

Dezavantajlar

- Değişim ve yenilik zordur.
- Müşteri öngörü ve önerileri önemsenmez.
- Projenin bitimine kadar çalışan ürün yok.
- Beklenmedik riskleri kolayca ele alınmaz.



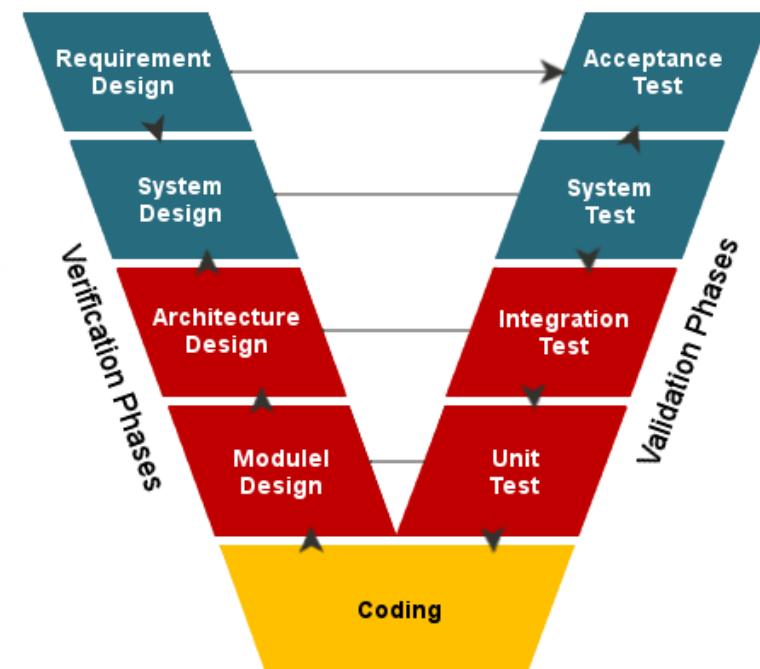
Waterfall (Şelale) Methodology

- Şelale modelinde analiz ve tasarım aşamaları oldukça detaylı yapıldığından, bu adımlar uzun sürmektedir.
- Ancak, analiz ve tasarım aşamalarında gereksinimlerin ve tasarımın net bir şekilde ortaya konulmasından dolayı, kodlama ve test aşamaları çok kısa sürmektedir.
- Test aşamasında çıkan hata sayısı azdır.



V Methodology

- V modeli, Doğrulama (**verification**) ve Onaylama (**validation**) modeli anlamına gelir.
- Tıpkı şelale modelinde olduğu gibi yazılım yaşam döngüsü adımları V şeklinde sıralı bir şekilde uygulanır.
- Bu modelde de her aşama bir sonraki aşama başlamadan önce tamamlanmalıdır.
- V Modelinin en temel özelliği “ürünün test edilmesi kendisine karşılık gelen geliştirme aşamasına paralel olarak planlanmaktadır.”.



V Methodology

Avantajlar

- Basit ve kullanımı kolaydır.
- Planlama ve test tasarıımı gibi test faaliyetleri kodlamadan önce gerçekleştirildiği için proje içerisinde çok zaman kazandırır.
- Bu nedenle şelale modeline göre daha yüksek başarı şansı vardır.
 - Hataların bulunması erken aşamada olur.



Dezavantajlar

- Uygulama şekli oldukça katı, kesin kurallara bağlıdır.
 - Yazılım şelalede olduğu gibi geliştirme aşamasında geliştirilir, bu nedenle yazılımın erken prototipleri üretilmez.
 - Herhangi bir aşamada gereksinimler üzerinde değişiklik olursa, test belgelerinin de diğer belgeler ile birlikte güncellenmelidir.

Spiral Methodology

- Tasarımı doğrusal bir süreç olarak gören diğer modellerin aksine, bu model spiral bir süreç olarak görür. Bu, yineleyici tasarım döngülerini genişleyen bir spiral olarak temsil ederek yapılır.
- Genellikle iç çevrimler, gereksinim tanımının rafine edilmesi için prototipleme ile birlikte ihtiyaç analizinin erken evresini ve dış spiraller yazılım tasarımını aşamalı olarak temsil eder.
- Her helezonda, tasarım çabalarını ve bu yineleme için ilgili riski değerlendirmek için bir risk değerlendirme aşaması vardır.
- Her spiralin sonunda, mevcut spiralin gözden geçirilebilmesi ve bir sonraki aşamanın planlanabilmesi için gözden geçirme aşaması vardır.

Spiral model



ILLUSTRATION: DETER HOMAS FRIBERG/DOE/STOCK
©2004 TECHTARGET. ALL RIGHTS RESERVED

Spiral Methodology

Avantajlar

- Risk analizi yapmaktadır.
 - Bu yazılım tasarım modeli, büyük yazılım projelerinin tasarlamak ve yönetmek için daha uygundur.

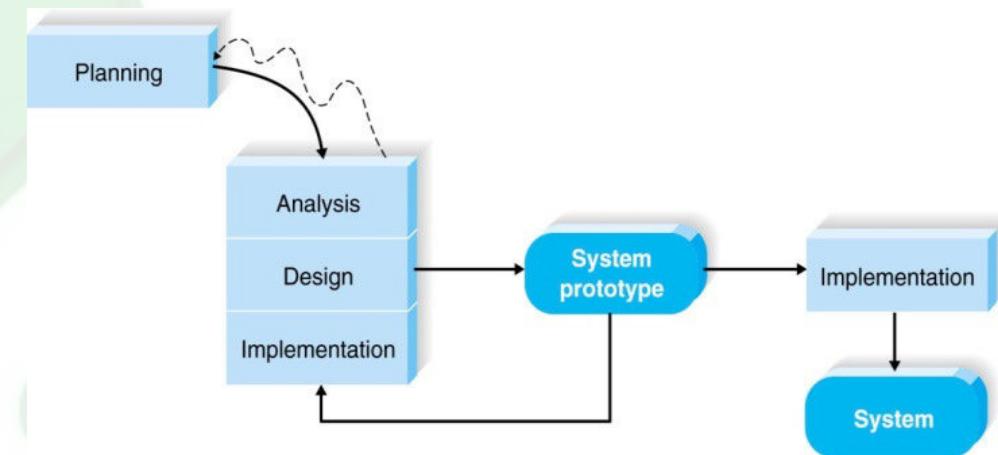
Dezavantajlar

- Risk analizi yüksek uzmanlık gerektirir.
- Kullanması **pahalı model**
- Küçük projeler için **uygun değil**.



Prototip Methodology

- Bu modelde çabuk tasarım, prototip geliştirme ve müşteri değerlendirmesinden sonra Prototip iyileştirilip referans ürün ortaya konur.
- Müşteriye sunulan ön ürün; ilk ürün olarak kabul edilir, yada iptal edilip en baştan yapılabilir.



Agile (Çevik) Methodology

- Agile Metodoloji (Çevik Metodoloji) yazılım sistemlerini etkili ve verimli bir şekilde modellemeye ve dokümantasyonunu yapmaya yönelik, pratige dayalı bir yöntemdir.
- Aşırı kuralcı klasik yazılım süreç modellerine tepki olarak ortaya çıkmıştır.
- Yazılımlar daha yüksek maliyetli ve daha yavaş geliştirilmekteydi. Yazılım geliştirme sürecini hızlandırmak, daha etkin kullanmak ve gerektiğinde dokümante etmek amacıyla bir çok yaklaşım ortaya çıkmıştır.
- 2001 yılında yazılım dünyasının önde gelen isimlerinden 17 arkadaş “Agile(Yazılım Geliştirme Manifestosu” ve “Agile (Yazılımın Prensipleri” ni yayımlamışlar, bu oluşumu ve gelişimini desteklemek için “Agile Alliance” adıyla, kar amacı gütmeyen bir organizasyon kurmuşlardır.
- Manifesto, nasıl daha iyi bir yazılım geliştirdiklerini ve bunu yapmak isteyenlere yol gösterecek 12 maddeden oluşmaktadır.



Agile (Çevik) Methodology

Agile Manifesto

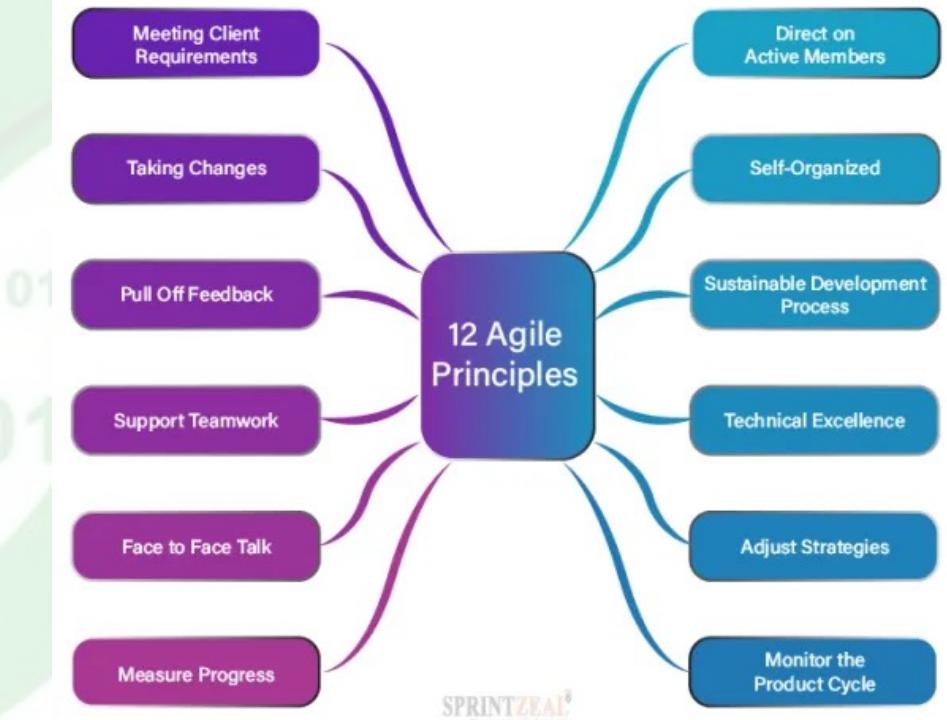
- Önceliğimiz kaliteli yazılımı geliştirerek bunu müşterisiye en hızlı şekilde teslim edebilmektir.
- Tüm değişiklikler projenin ileriki aşamalarında dahi olsa kabul edilir.
- Çok kısa aralıklarla yazılım teslimleri yapılır.
- Alan uzmanları yazılımcılar, testçiler günlük olarak birlikte çalışırlar.
- Motiv olmuş bireyler etrafında projeler oluşturun.
- Bir geliştirme ekibine ve içinde bilgi aktarmanın en verimli ve etkili yöntemi yüz yüze görüşmedir.



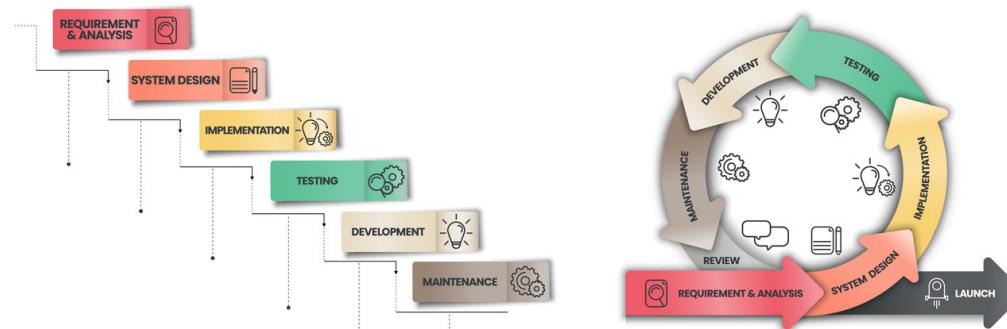
Agile (Çevik) Methodology

Agile Manifesto

- Çalışan yazılım, ilerlemenin **birincil** ölçüsüdür.
- Agile processes sürdürülebilir gelişmeyi **teşvik** eder.
- Teknik mükemmelliğe ve iyi tasarıma verilen sürekli dikkat, **çevikliği** artırır.
- Sadelik **esastır**.
- En iyi mimariler, gereksinimler ve tasarımlar **kendi kendini organize eden ekiplerden** ortaya çıkar.
- Ekip, düzenli aralıklarla nasıl daha etkili olunacağını düşünür, ardından davranışını **buna göre** ayarlar.



Waterfall vs Agile

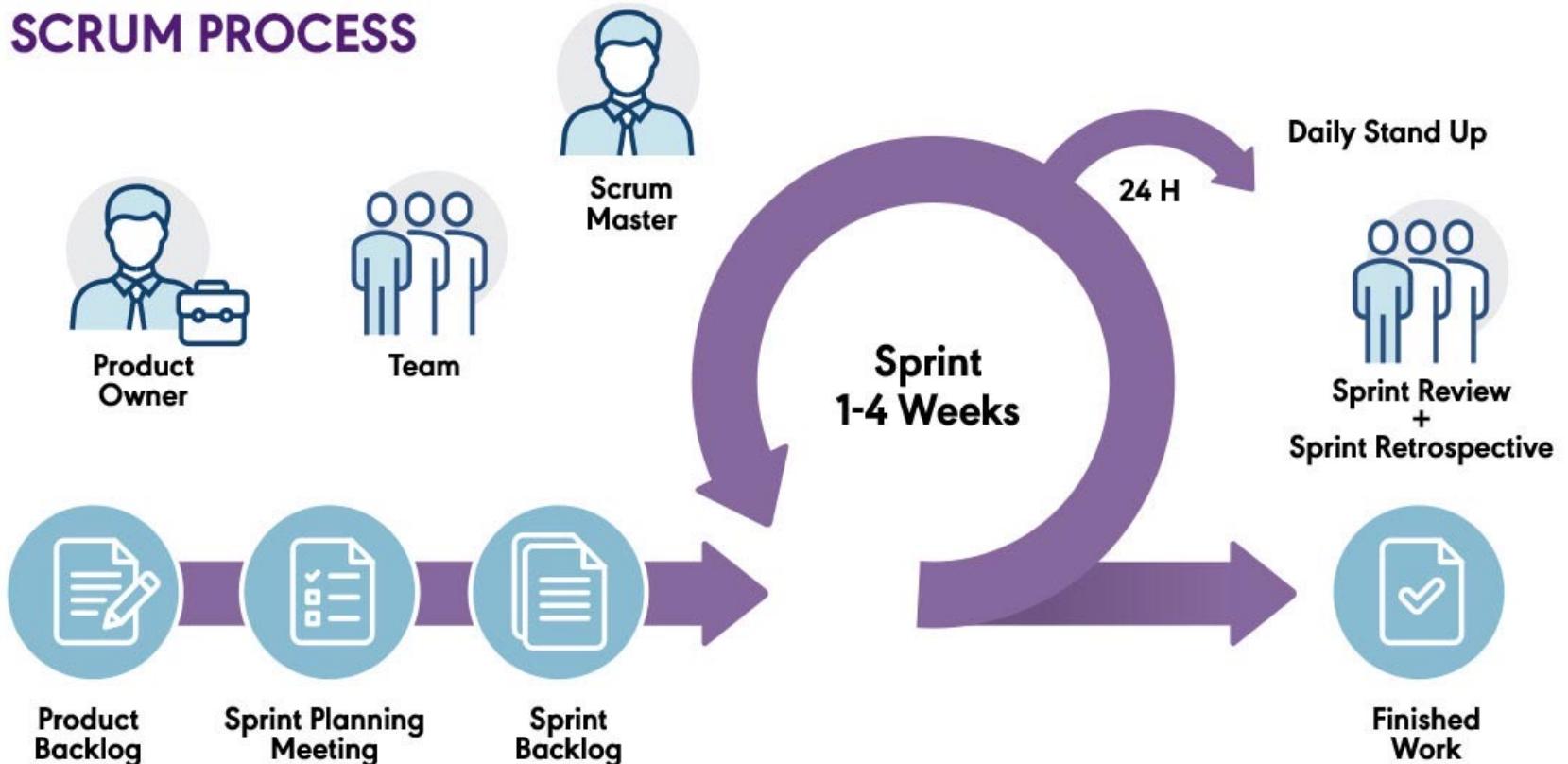


**WATERFALL
MODEL**

**AGILE
MODEL**

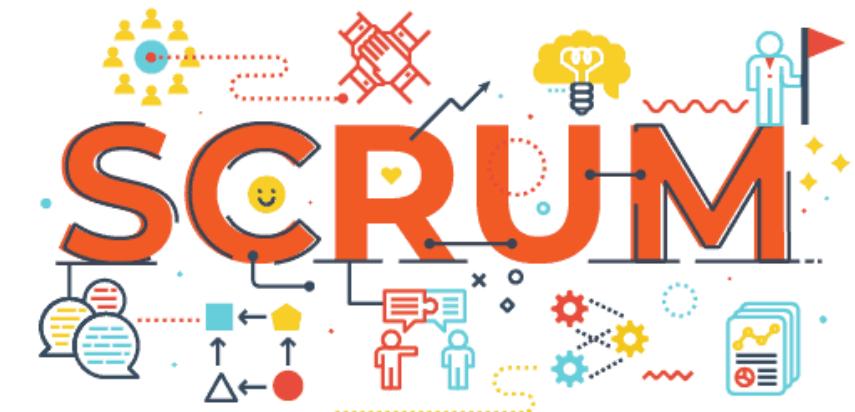
Method	Successful	Challenged	Failed
Agile	42%	50%	8%
Waterfall	26%	53%	21%

Scrum Process



Scrum

- Agile proje yönetim metodolojilerinden biridir.
- Kompleks yazılım süreçlerinin yönetilmesi için kullanılır.
- Bunu yaparken bütünü parçalayan tekrara dayalı bir yöntem izler.
- Düzenli geri bildirim ve planlamalarla hedefe ulaşmayı sağlar.
- Bu anlamda **ihtiyaca yönelik ve esnek bir yapısı** vardır.
- Müşteri ihtiyacına göre şekillendiği için **müşterinin geri bildirimine göre** yapılanmayı sağlar.
- İletişim ve takım çalışması **çok önemlidir**.
- Scrum teorisi 3 temel aşamaya dayanır;
 - **Şeffaflık – Denetleme - Adaptasyon**



Scrum

Sprint – Rugby Yaklaşım

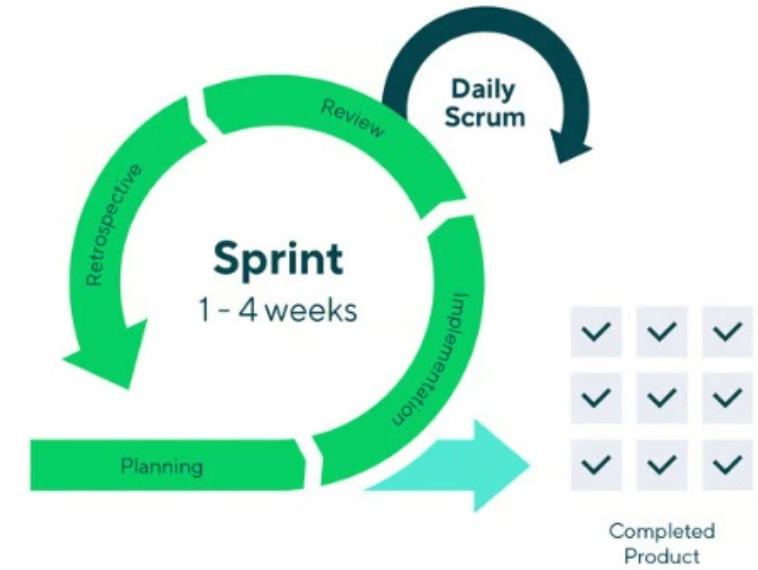
- “Rugby” yaklaşımı : “takımın, mesafenin tümünü hep beraber, bir birim halinde topu ileri geri atarak kat etmesidir.



Scrum

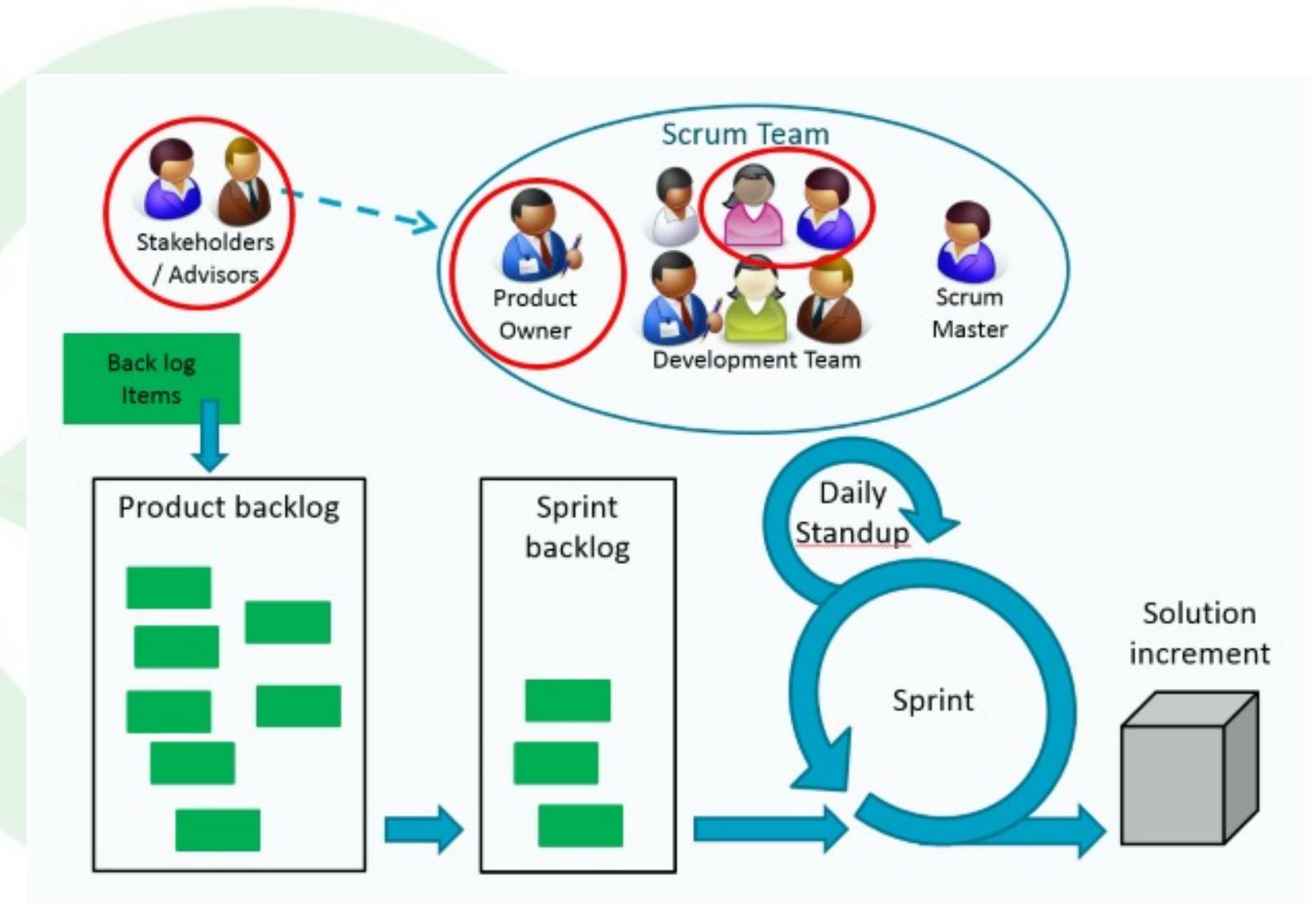
Sprint

- Her sprint genellikle **2 - 4 hafta** veya en fazla bir takvim ayı sürer.
- Ürünleri her seferinde küçük bir parça oluşturmak, üretkenliği teşvik eder ve ekiplerin geri bildirime ve değişime yanıt vermesini ve tam olarak gerekli olanı oluşturmasını sağlar.
- Scrum' da ürün sprintte tasarılanır, kodlanır ve test edilir.
- Yapılacak tüm işler, **Product Backlog** da biriktirilir, **Product Owner**' un belirlediği öncelike göre Sprint Backlog' una alınır ve bir sprintte bitirilerek ürünün demosuna eklenir.



Scrum Team

- Product Owner
- Scrum Master
- Development Team



Product Owner

Kimdir ?

- Geliştirme takımı ve müşteri arasındaki iletişimini sağlar.
- Projenin önceliklerine göre Product Backlog (iş listesi) oluşturur.
- Ekipde Stakeholders' ı temsil eder. Geliştirme Takımı ve Stakeholders arasındaki iletişimini gerçekleştirir.
- Her Sprint' te hangi user story' lerin sprinte dahil edilemeyeceğine karar verir.
- Sprint değerlendirme toplantılarının sahibidir.
- İş listesini (Product Backlog) yönetir.
- Ürünle ilgili yapılacak bütün geliştirme Product Backlog' da bulunur.
- Product Backlog' un sahibidir.

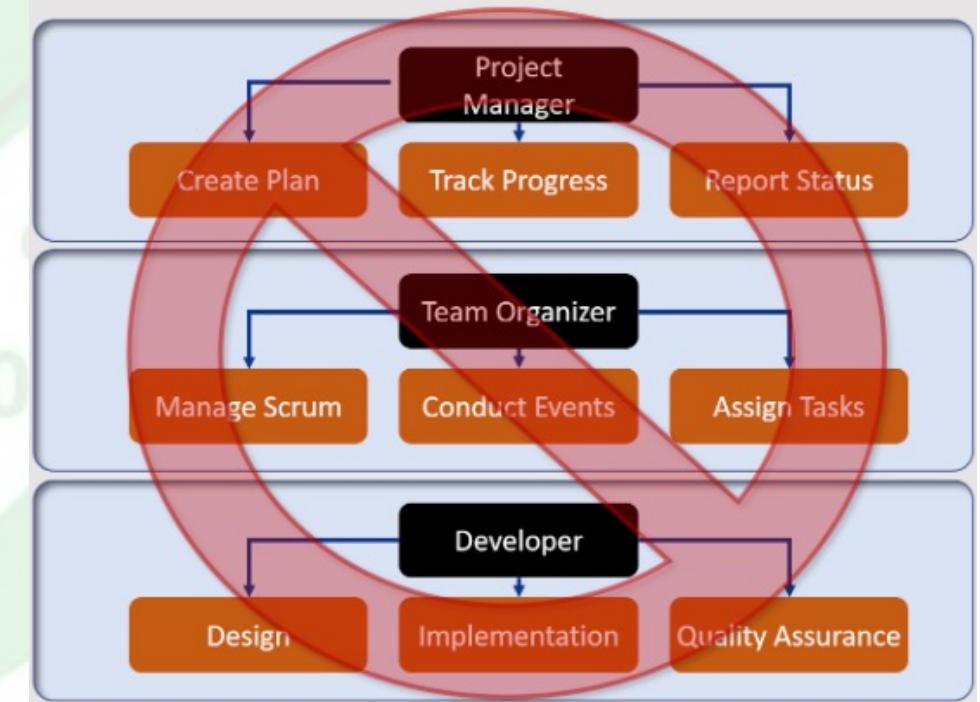


Product Owner

Kim değildir ?

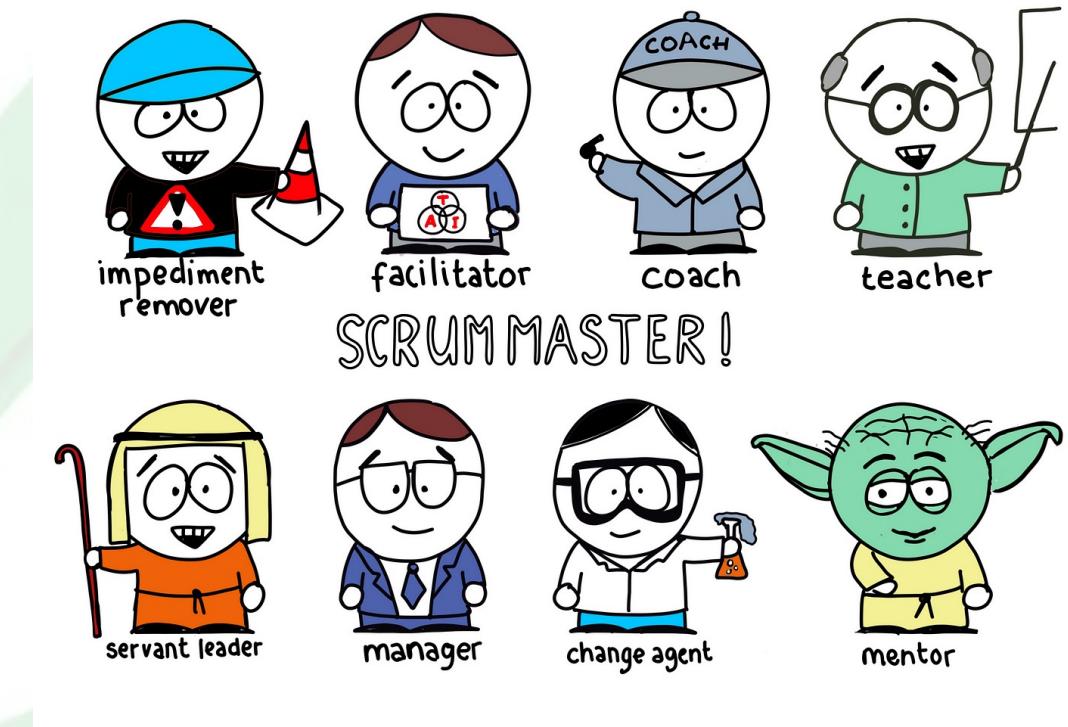
- Development Team' in ve Scrum' in **yöneticisi değildir.**
- Geliştirme Takımı teknik konularda **kendi kendini yönetir.**
- PO, teknik bir geçmişe sahip olabilir.
- Bu, hangi işi kimin yapacağına karışabileceğinin anlamına gelmez.
- PO, iş-görev ataması **yapamaz.**
- İşin nasıl yapılacağına karışamaz!

NOT the Product Owner Role



Scrum Master

- Scrum kurallarını, teorilerini ve pratiklerini iyi bilir ve takımın bu kurallarını uygulamasından sorumlu kişidir.
- Takımın yönetici değilidir.
- Takımı rahatsız eden, verimli çalışmalarını engelleyen durumları ortadan kaldırır.
- “Scrum Master, takımın Scrum değerlerine, pratiklerine ve kurallarına bağlı kalmasını garanti altına almakla sorumludur.
- Scrum Master, takımını ve organizasyonu Scrum’ a adapte eder.”



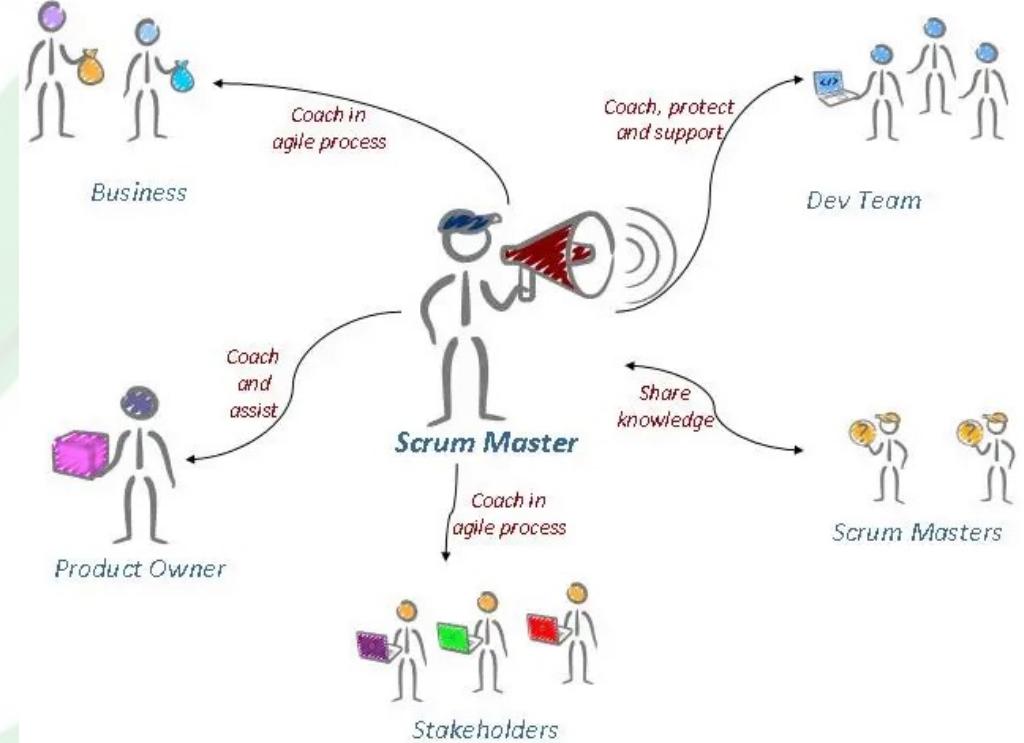
Scrum Master

- Scrum takımının üyesidir.
- İş birlikçidir.
- Koruyucudur.
- Yardımcıdır.
- Problem çözücüdür.
- Kararlı ve ulaşılabilir.
- Bilgilidir.



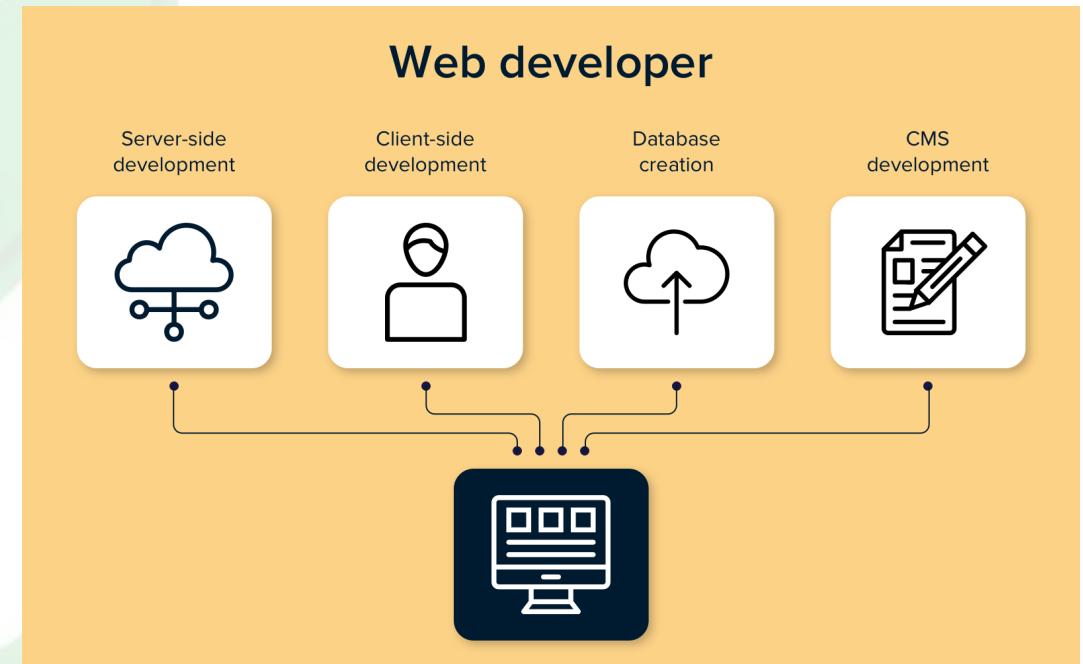
Scrum Master

- ScrumMaster, takıma rehberlik ve koçluk eder, karşılaşılan engelleri ortadan kaldırılmalarına yardımcı olur.
- Takım içi harmoniyi, ekip elemanları arasındaki uyumu, iletişimini artırmak için çabalar.
- Sprint Planlama, Sprint Retrospektif, Günlük Scrum ve Sprint Review gibi Scrum ritüellerini ve toplantılarını kolaylaştırır.
- ScrumMaster takımın güvenli ve sorunsuz bir ortamda çalışabildiğinden emin olmalı, gerektiğinde takım elemanlarına bireysel koçluk da dahil olmak üzere bir çok hizmetini sunmalıdır.



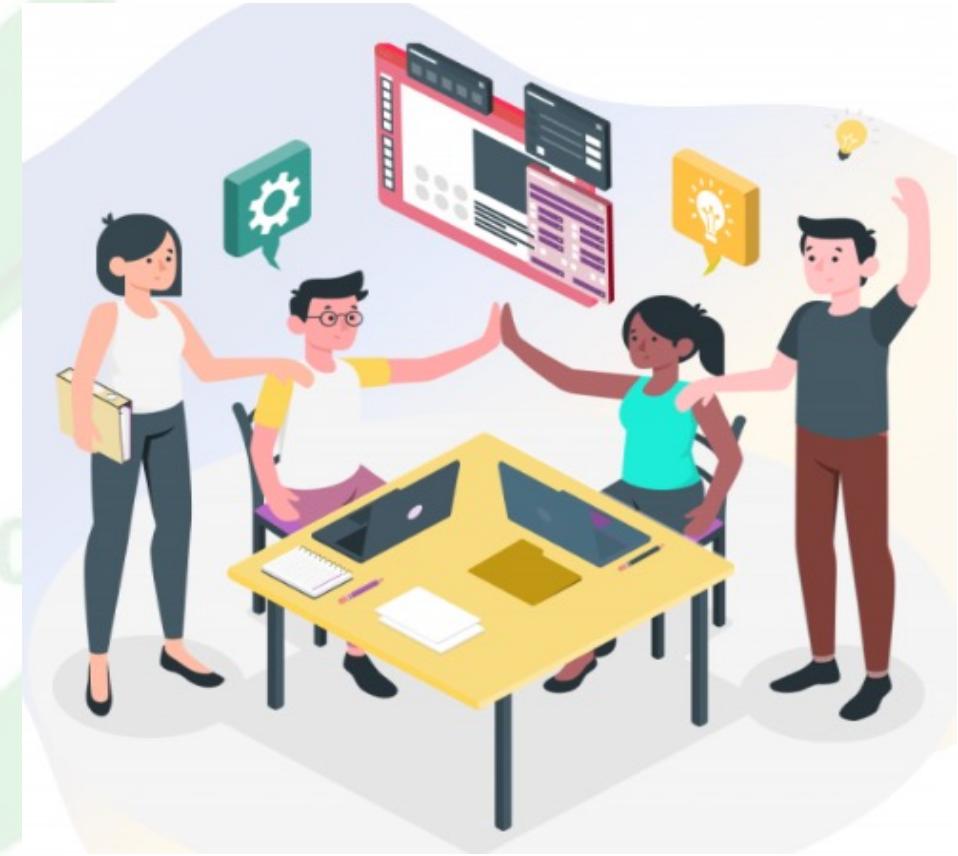
Development Team

- Front End Developer, Back End Developer, DevOps, QA Tester UI/UX Designer vb. gibi teknik becerilere sahip kişilerden oluşur.
- Product Owner'ın yapılacak işler listesi olan Product Backlog'tan belli bir sürede yapabileceği kadar işi, Product Owner'ın belirlediği öncelikle göre yapan geliştirme takımıdır.
- Teknik konularda sorumluluk **Development Team'** e aittir.
- PO ve SM Development Team'ın yapacağı işlerin öncelğini belirleyebilir ama neyi nasıl yapacaklarına karışmazlar.



Development Team

- Takım içerisindeki kişilerin rolleri ve yetenekleri ne olursa olsun, dışarıya karşı bir işin tamamlanmasından tüm takım sorumludur.
- Bir Sprint' e alınan bütün işleri tamamlayacak özelliklere sahip kişilerdir.
- Sprint Backlog' u oluştururlar.
- Kendi kendini yönetir İşin verilmesini beklemezler, işi kendileri alır ve geliştirirler.
- Development Team, Product Backlog' tan çektiği bir Product Backlog Item' i Product Owner' in önünde çalışan bir kod parçacığı olarak koymakla yükümlüdür.
- “Self Organize” olmalıdır. Development Team, sorumluluğunu aldığı işlerin yapılması için bir iş tanımlamasına veya bir iş takipçisine ihtiyaç duymaz.





BATCH :
LESSON :
DATE :
SUBJECT :

146 - 149

SDLC

12.07.2023

SDLC - 3

ZOOM GİRİŞLERİNZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ





SDLC

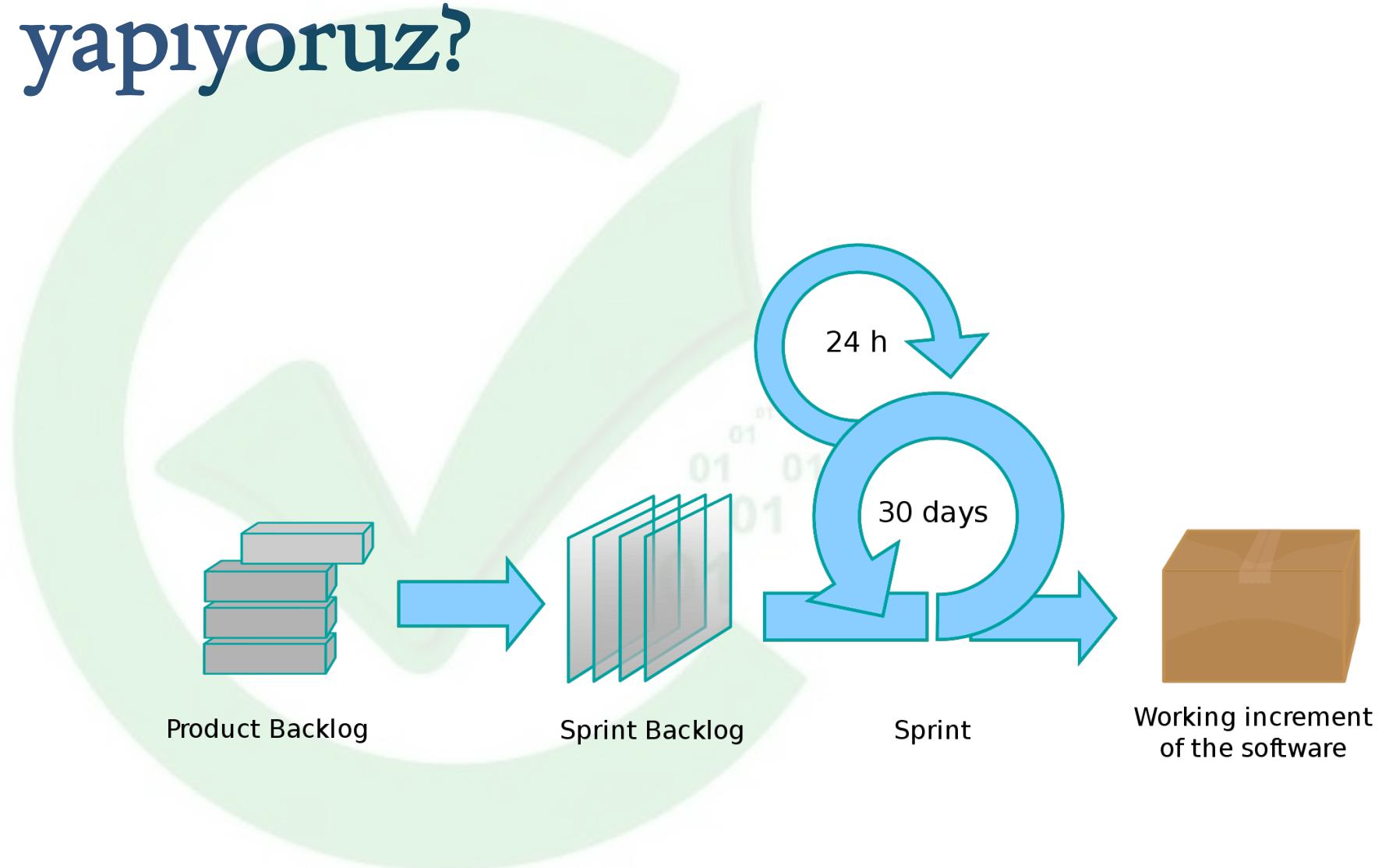
Software || System Development Life Cycle

3. Ders
12.07.2023

B149 AWS & DevOps
B146 Cyber Security

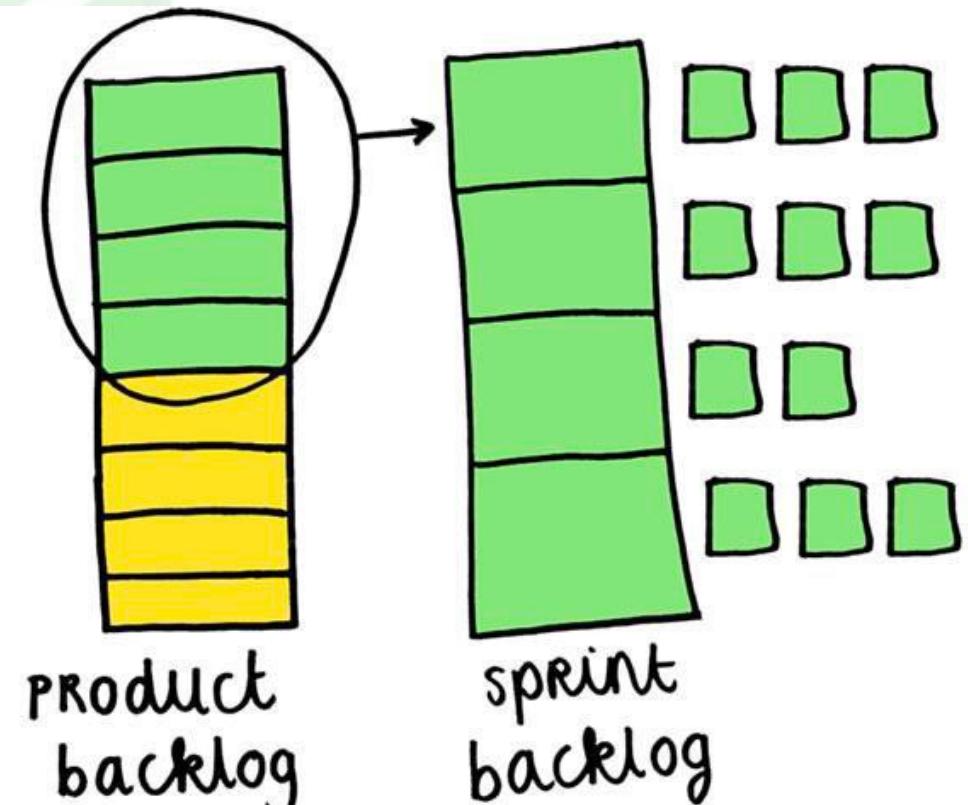
Bugün ne yapıyoruz?

- Product Backlog
- Sprint Backlog
- Scrum Meets
 - Sprint Planning
 - Daily Scrum
 - Sprint Review
 - Sprint Retrospective
- Scrum Board
- Kanban Board
- Feature
- Epic
- User Story
- Scrum Tools
- Kahoot



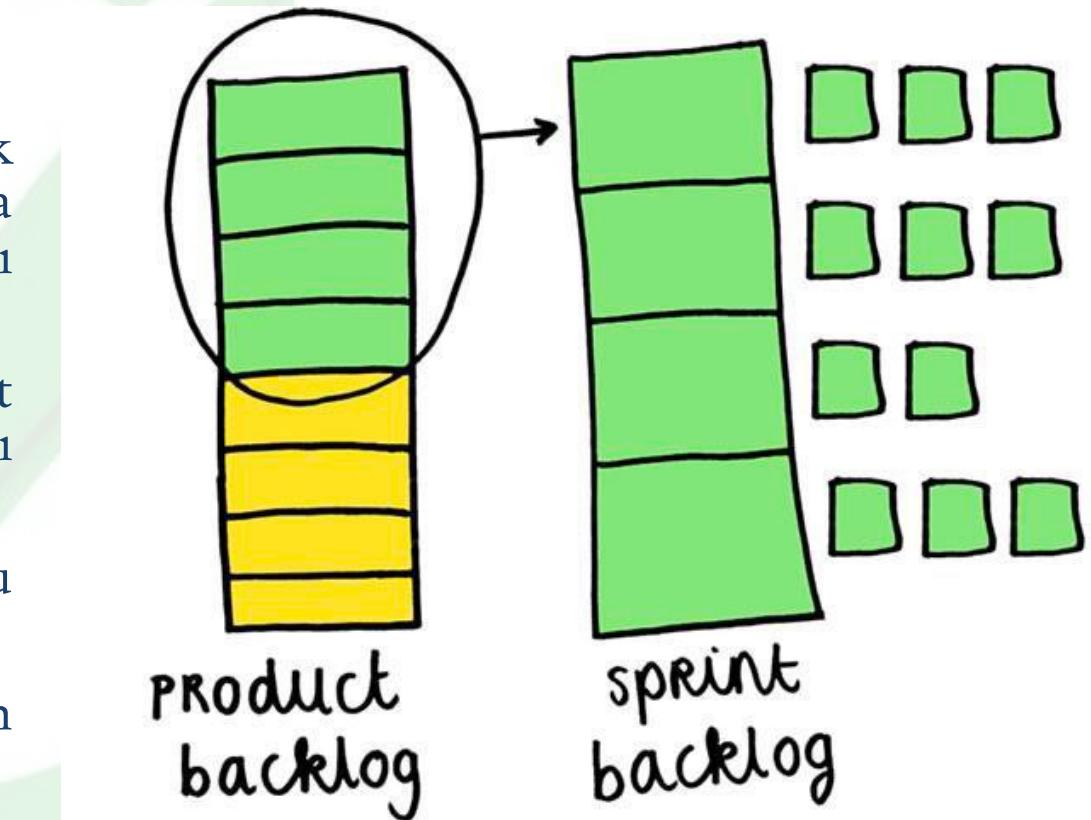
Product Backlog

- Proje için gerekli olan gereksinimler listesidir. Proje sonunda “Ne üretilmek isteniyor?” sorusuna cevap aranır.
- Product owner tarafından müşteriden gereksinimler alınır, öncelik sırasına göre sıralanır.
- Product owner, değişen ihtiyaçlara göre Product Backlog'a ekleme veya çıkarma yapabilir.
- Böylece değişim, projenin her aşamasında projeye kolayca entegre edilebilir.
- **Product Backlog Item:** Product backlog içindeki her bir gereksinime verilen isimdir.



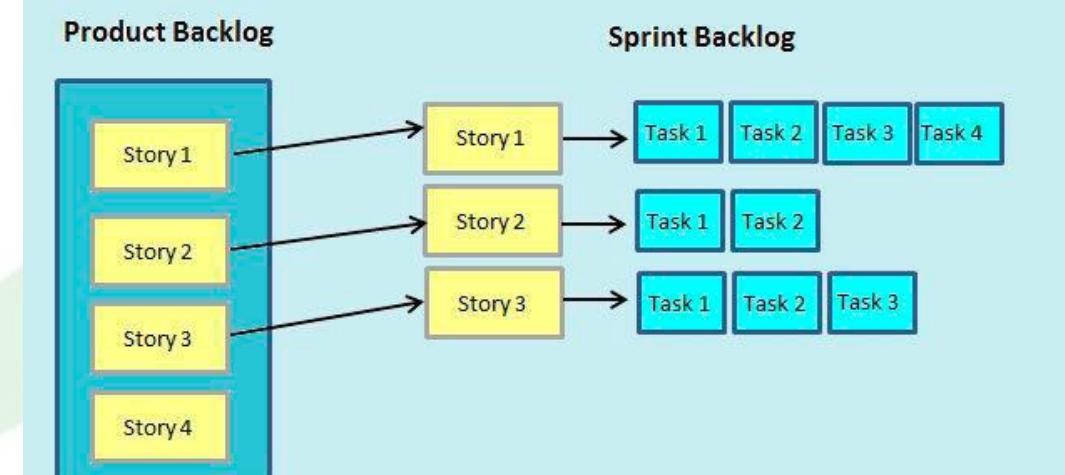
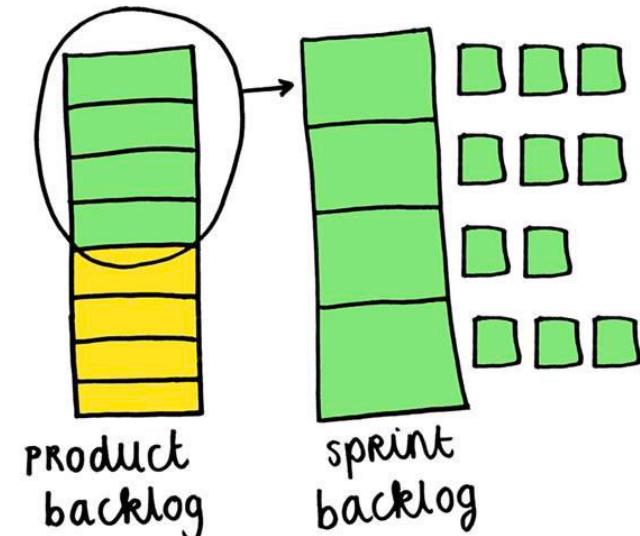
Product Backlog

- Product Backlog yaşayan bir dokümanıdır.
- Çevresel koşullar, müşteri talepleri, teknolojik gelişmeler ve geliştirme yetkinliğinde meydana gelebilecek değişimlerden en yüksek faydayı sağlayacak şekilde sürekli değişir.
- İyi bir Product Backlog takımın önündeki 2-3 sprint için net anlaşılmış ve yeterince küçük PBI'ları barındırmalıdır.
- Product Owner (Ürün sahibi), Product Backlog' u yöneten yegane kişidir.
- Bu işi yaparken paydaşlardan, geliştirme takımından ve Scrum Master' dan destek ve yardım alabilir.



Sprint Backlog

- Sprint Backlog takımın Sprint boyunca yapacağı işlerin adımlandırılmış detaylandırılmış, saatlendirilmiş içeriğini teşkil etmektedir.
- Sprint Backlog Development Team' in oluşturduğu, Development Team' in Sprint' te almış olduğu işleri ve bu işleri nasıl tamamlayacaklarına ilişkin stratejilerini içeren bir çıktıdır.
- Sprint Backlog' a alınacak user story' ler belirlenirken de Product Backlog' daki öncelik sırasını gözetilir. Düşük öncelikli olanları ilk sprint' lere koymazlar.
- Sprint Backlog' a alınacak user story miktarı takımın verdiği point' lere göre değişir.



Scrum Meets



Sprint Planning Meets

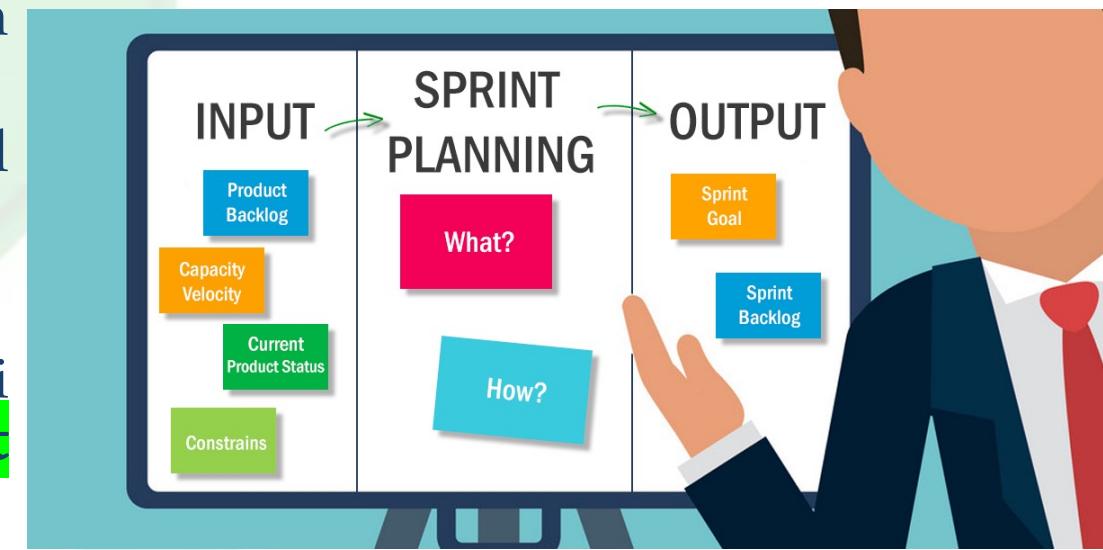
- Her Sprint' i **başlatan** etkinliktir.
- Product Owner ve Development ekibinin Sprint' e hangi Ürün İş Listesi öğelerinin PBI' ler dahil edileceğini tartıştığı yerdir.
- Product Owner Sprint' e potansiyel olarak dahil edilmek üzere her PBI' ya öncelik verme hakkına sahip olsa da, Development Team yanıt vermeye, sorunları gündeme getirmeye ve gerektiğinde geri itmeye teşvik edilir.
- Development Team daha sonra hız, kaynaklar ve mevcut zaman ve kaynakları etkileyebilecek faktörler hakkında bilgi sahibi olduklarında Sprint' te kaç item dahil tahmin eder.
- Sprint Planlama toplantısının sonucu, herkesin kabul ettiği gerçekçi ve ulaşılabilir bir Sprint Hedefi ve Sprint İş Listesi elde etmektedir.

Sprint Planning Meeting



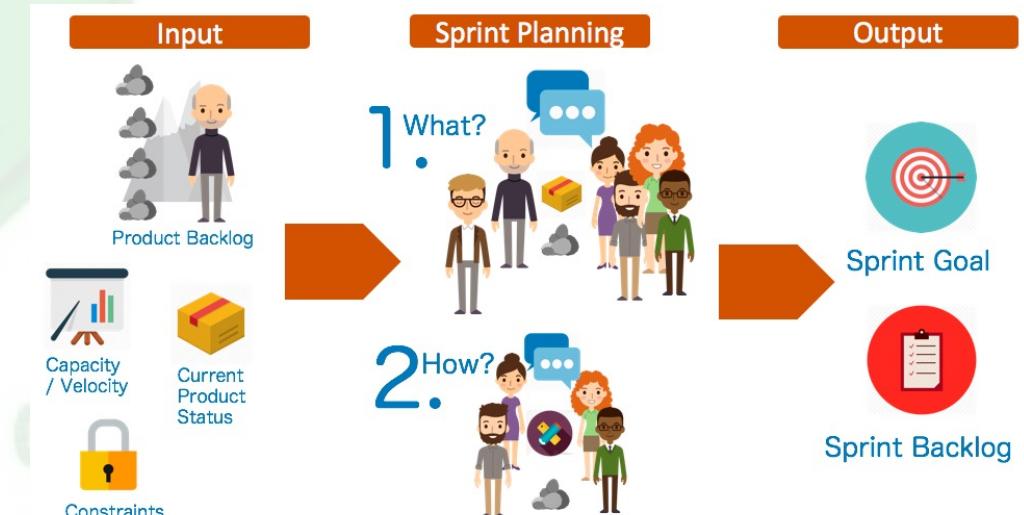
Sprint Planning Meets

- Sprint planlamada şu sorulara cevap verir;
 - Başlayan Sprint' te task olarak ne teslim edilebilir? (Ne Yapacağınız?)
 - Uygulamayı teslim etmek için gerekli olan iş nasıl başarılıacak? (Nasıl Yapacağınız?)
- 2. madde için gerekirse Sprint süresi içerisinde eski adıyla **Grooming** yeni adıyla **Refinement** (detaylandırma) aktivitesi gerçekleştirilebilir.
- **Scrum Master**, toplantıının yapılmasını ve Development Team' in etkinliğin amacını anlamasını sağlar.



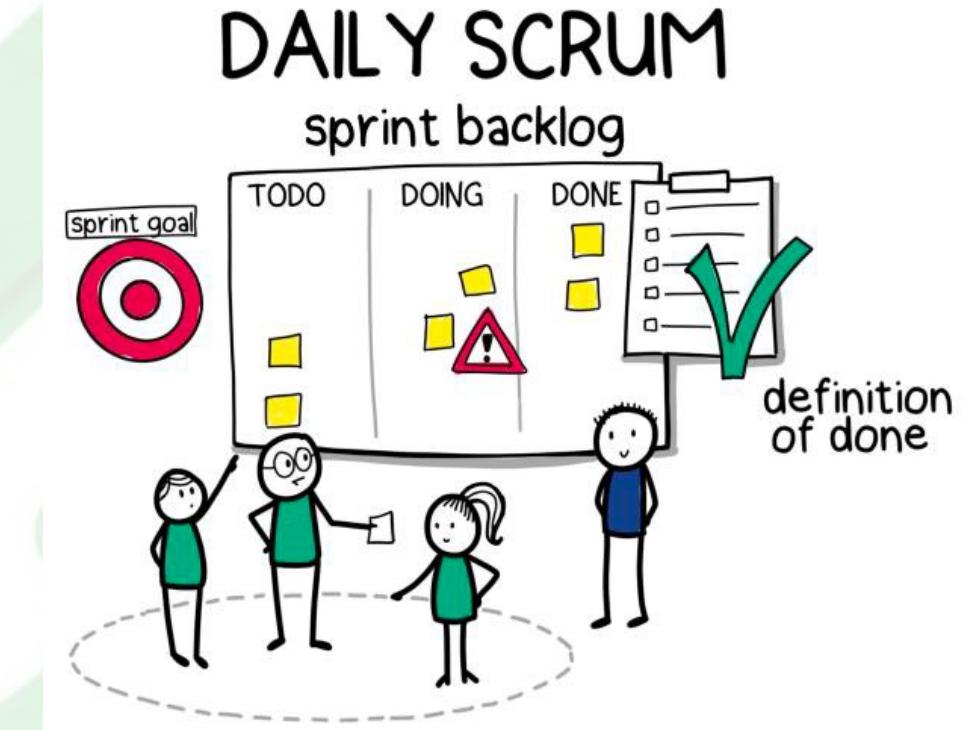
Sprint Planning Meets

- Bir önceki Sprint Retrospektif Toplantısı'nda alınan kararlar içinde Sprint Planlama Toplantısını etkileyen aksiyonlar varsa bunları yerine getirildiği kontrol edilir.
- Herşeyin başlangıcı iyi bir planlamadır.
- Sprint'e iyi başlarsanız başarılı bir şekilde bitirme şansınız yüksek olur.
- İyi Scrum Master, Scrum Takımı'nın iyi planlama yapmasını sağlar.
- İş Listesi Maddeleri'nde kabul kriterleri belirtilmiş mi?



Daily Stand-Up Meets

- Scrum zamanınızı ve kaynaklarınızı verimli bir şekilde kullanmayı amaçlar.
- Yaklaşık 15 dakika kadar sürer. Ayağa kalkmak zorunlu değildir Ancak, birçok takım bu toplantıyı kısa ve öz tutmak için yararlı bir teknik olarak bulmaktadır.
- Geliştirme Ekibinin oto kontrol yapması, Sprint Hedefi'ne ulaşma yolundaki ilerlemeyi değerlendirmesi ve önümüzdeki 24 saat boyunca faaliyetlerini gözden geçirmesi ve planlaması için bir fırsattır.
- Genelde dün neler yaptık? Bugün ne üzerinde çalışacağız? Herhangi bir sıkıntı veya engel ile karşılaştık mı?



Sprint Review (Demo) Meets

- Genellikle Sprint' in son gününde yapılır ve Stakeholder' lara (yönetim ve ilgili ve ilgilenilen diğer herkes) "done" (yapılan işi gösterme fırsatı verir).
- Sprint sırasında üretilen çalışma özelliklerini göstermenin yanı sıra, gelecekteki sprint' ler için çalışmayı yönlendirmeye yardımcı olabilecek Ürün İş Listesi' ni ekleyebileceğiniz yararlı geri bildirimler de alıyorsunuz.
- Sprint Review(Demo) Toplantısı' ni **Product Owner liderlik eder.**
- Ancak aktarımı **QA Tester ekibi** yapar.



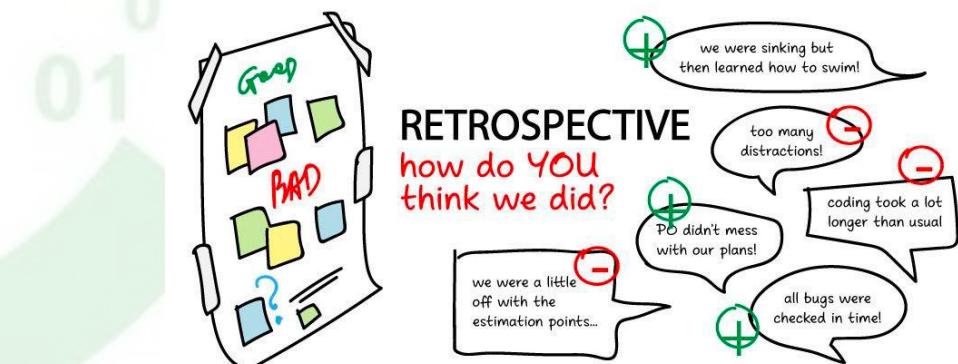
Sprint Review (Demo) Meets

- Burada yapılması gereken şey stakeholders' a “ürünün onların ürünü” olduğunu vurgulamaktır. Onlara anlatacaklarınızla, göstereceklerinizle sizden istediklerinin örtüşüp örtüşmediğinin bu toplantıda anlaşılacağını anlatmaktadır.
- Stakeholders çalışan ürünü görerek bu ürüne eklenmesi istedikleri yeni özellikler yada değiştirilmesini istedikleri özelliklerin hangi niteliğe, görselliğe, yetkinliğe sahip olması gerektiğini yüz yüze, birinci ağızdan söyleyebilirler.

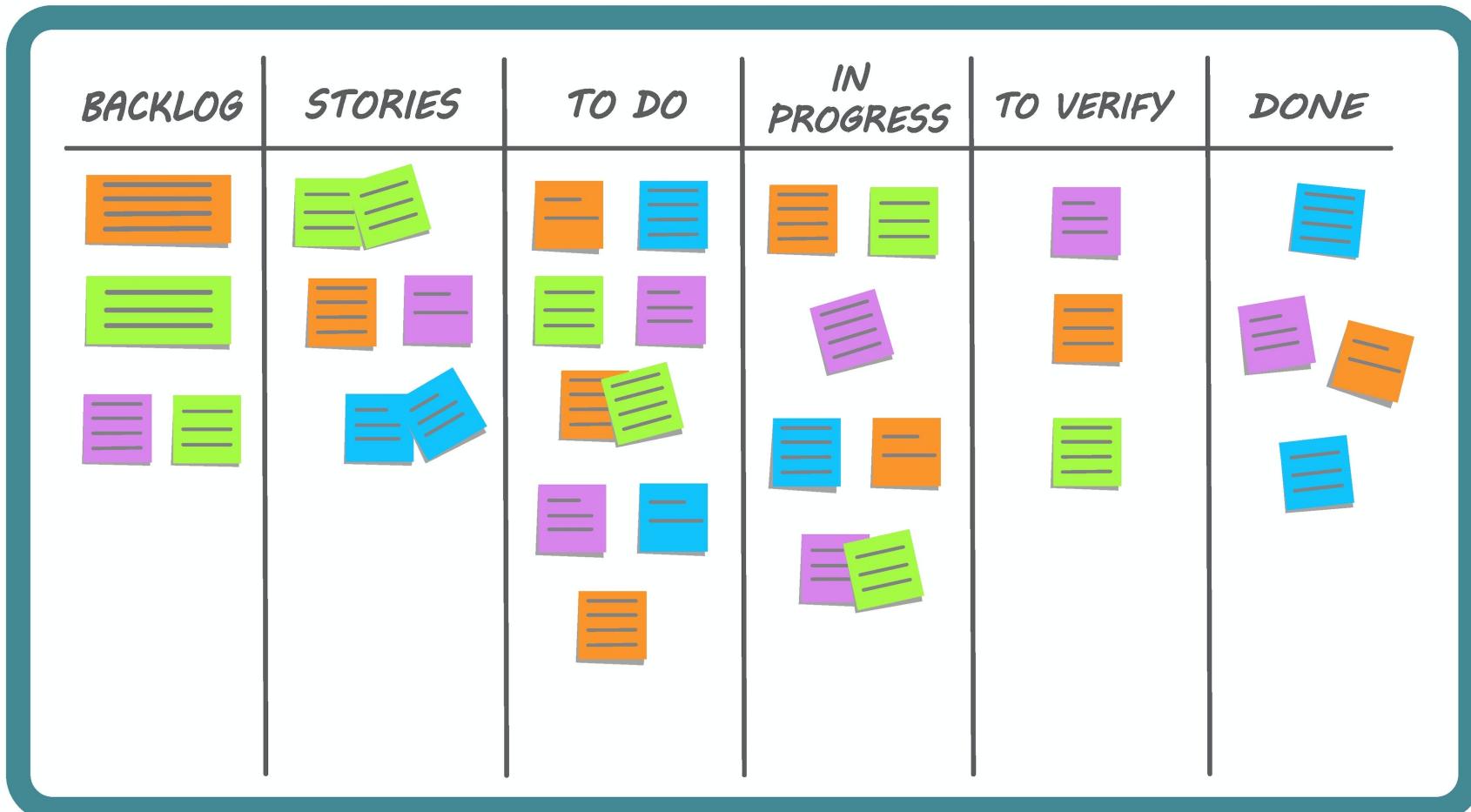


Sprint Retrospective Meets

- Retrospective toplantıları bir sonraki Sprint' te iş süreçlerini iyileştirmek için geçmiş Sprint' in incelendiği ve "nasıl daha iyi performans gösterebiliriz?" sorusuna cevap aranan toplantılardır. Bu toplantıyı geliştirme ekibi, scrum master ve product owner katılır.
- Sprint' teki son toplantıdır ve "sprint review toplantısının hemen ardından yapılır.
- Scrum ekibi gelecekteki Sprint' ler için nelerin geliştirilebileceğini ve nasıl yapmaları gerektiğini gözden geçirir.
- Ne tür engellerle karşılaşırlar ve hangi fikirlerin ve güncellemelerin daha fazla gelişim sağlamalarına katkıda bulunduğunu değerlendirirler.



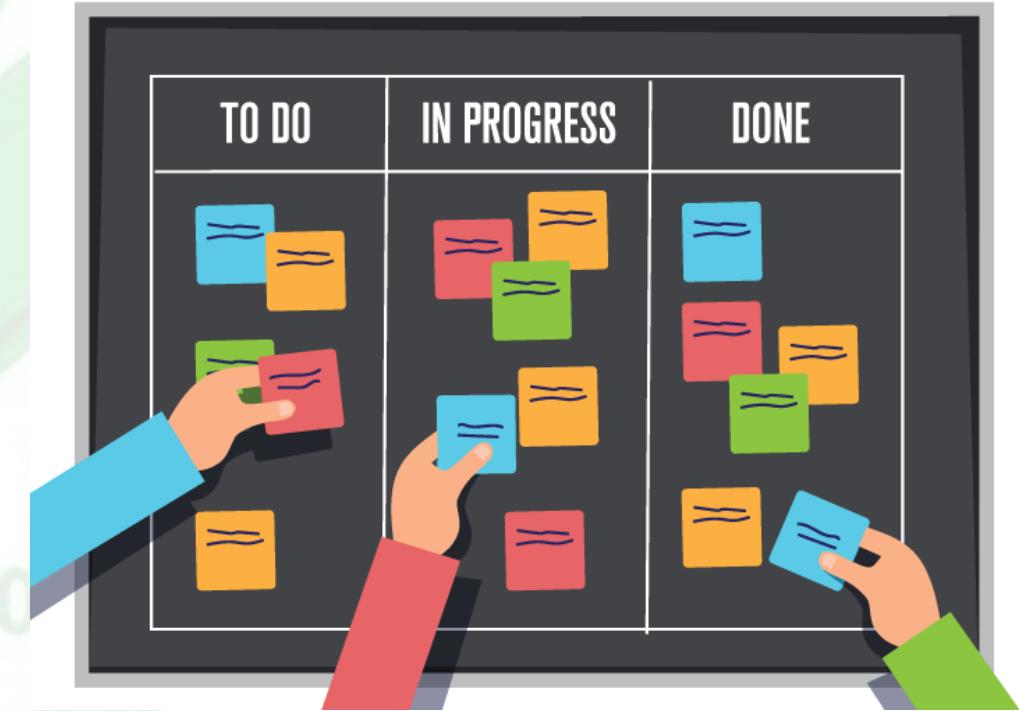
Scrum Task Board



Kanban Task Board

Başarılı Kanban uygulaması için
gözlemlemeniz gereken **6 (altı) temel**
uygulama vardır.

- İş akışını görselleştirin
- Devam eden çalışmayı sınırla
- Akışı yönetin
- Süreç politikalarını açık hale getirin
- Geri bildirim döngülerini kullanın
- İşbirliği yaparak geliştirin

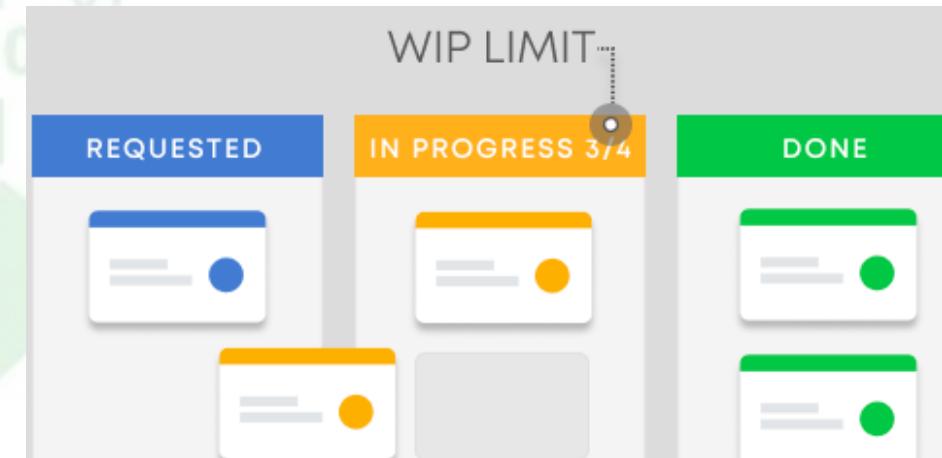
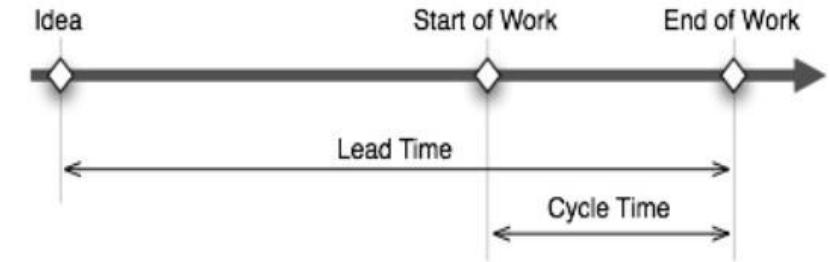


Kanban Task Board

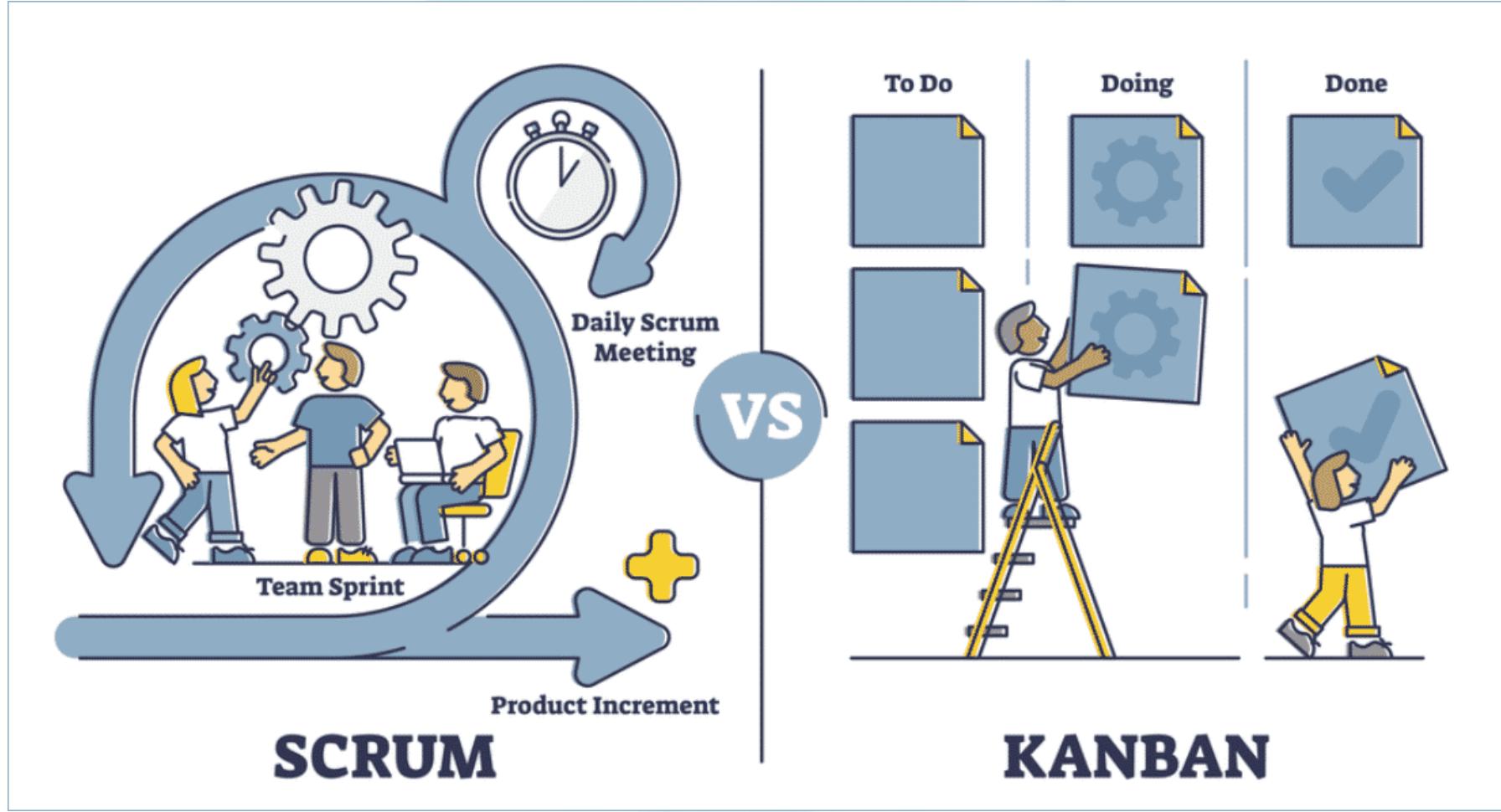
Kanban uygulaması proje istenilen kısmında müşteri isteği ile kolayca geçilebilmektedir.

Kanban Board' da kullanılan terimler;

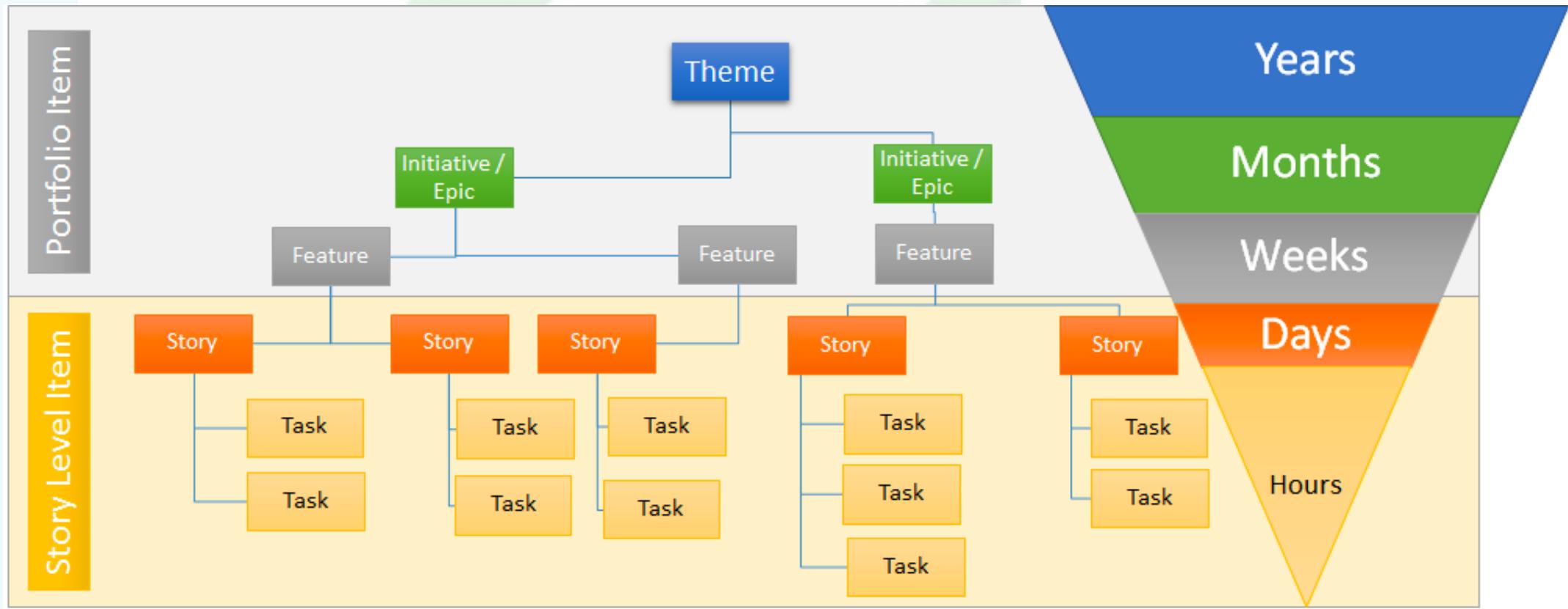
- WIP Limit : Work Process adeti belirtir.
- Lead Time : Task fikri ile bitiş arasındaki süreyi belirtir.
- Cycle Time : İşe başlangıçdan bitişe kadar olan süreyi belirtir.



Scrum vs Kanban



EPIC, Feature, User Story, Task



EPIC

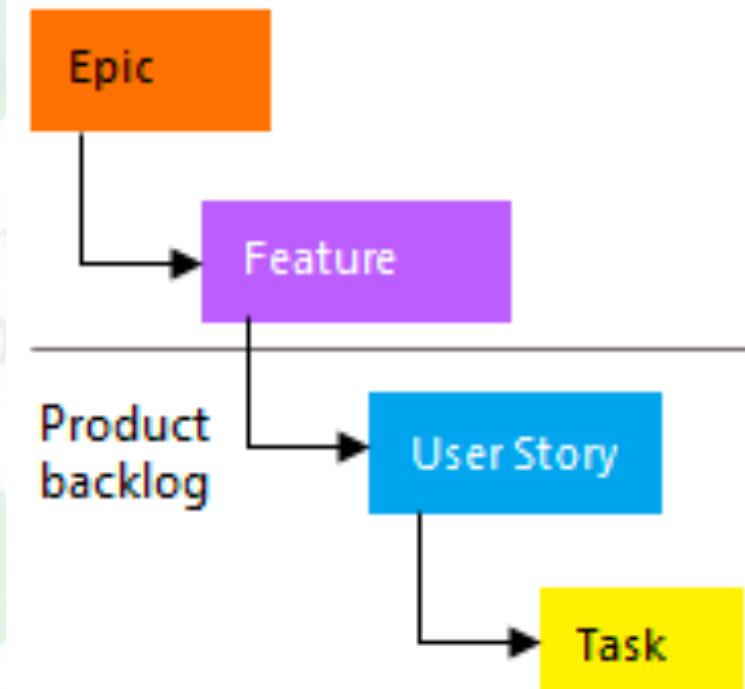
- Belirli bir vizyona ulaşabilmek için şirketlerin metrikler ile tanımlı, ulaşması gereken hedefleri vardır. Ekipler ise bu hedeflere ulaşabilmek için yeni girişimler/özellikler ile gelir ve sahip oldukları ürünü/ürünlerde yeni güncellemeler yayınlarlar.
- Epics, yapılması planlanan bir veya birden fazla yeni fonksiyonların yahut özelliklerin gruplanmasıdır.
- Product ekibinde yapılan her şey son kullanıcıya sunulmaz veya yeni bir özellik olarak sonuçlanmaz. Bu yüzden özellikler(features) olarak değil epics olarak adlandırılırlar. Ayrıca bir Sprint' ten daha uzun süren çalışmalar da “ olarak adlandırılmaktadır.



Feature

- Epicler içerisinde PO tarafından seçilerek Product Backlog' a eklenen, birden fazla user story içeren, birden fazla haftada tamamlanacak modüllerdir.
- Son kullanıcıya tamamlanmış ürün olarak ulaşır.

Portfolio backlog



User Story

- User Storyler 3 bölümünden oluşur. Bunlar;

WHO – WHAT - WHY

- Kullanıcının hikaye tanımı (Value statement)
- Kabul kriterleri (Acceptance criterias)
- Bitti tanımı (Definition of done)

- User Stories, Agile metodolojisinin bir parçasıdır.
- Son kullanıcıya sunulacak herhangi bir özelliği veya ürünü açıklama yollarından biridir.



User Story

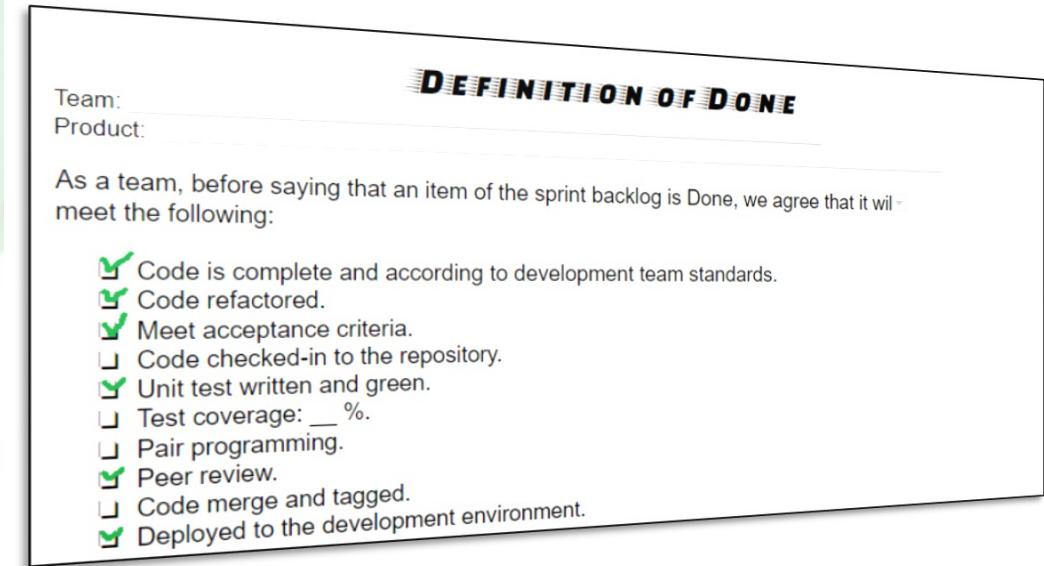
- User Stories 3 ortak özelliği barındırır. Bunlar;

- Kısa ve Açıklayıcı Olma
- Kullanıcı Tipini belirlemek
- Problemi Tanımlama

User Story ID	Description	Acceptance Criteria
US 0001	ATM kullanıcısı olarak, nakit çekebilmem için PIN kodumu girmek istiyorum	PIN dört basamaklı olmalıdır
		PIN özel karakterlere (sembollere) izin vermemelidir
		PIN'in 30 saniye içinde girilmesi gereklidir, aksi takdirde işlem iptal edilir
US 0002	Facebook login (Giriş sayfası) geçersiz kimlik bilgileriyle erişilmemelidir	Gecersiz kullanıcı adı ile erişim sağlanamaz
		Geçersiz şifre ile erişim sağlanamaz
		Geçersiz kullanıcı adı ve şifre ile erişim sağlanamaz

Definition of Done (DoD)

- Development Team, kendilerinden tamamlanması beklenen işleri Product Owner'a sunmadan önce bir kalite kontrol süreci gibi kontrol listesi hazırlar.
- Bu liste tüm User Storyler için ortaktır ve biten tüm işler bu prosedürden geçmesi gereklidir.
- Buna kısaca "veya "DoD Definition of Done (Bitti Tanımı)" denir.
- Bu liste genellikle şu kontrolleri içerir;
 - Unit Testleri
 - QA Testleri
 - User Story' deki tüm dokümanların güncellenmesi
 - Kodun Review edilmesi
 - User Story kabul kriterlerinin karşılanması
 - PO Onayı
- DoD checklist, Sprint veya Release sonunda belirli tamamlama kriterlerinin karşılanıp karşılanmadığını kontrol etmek için de kullanılmak üzere hazırlanabilir.

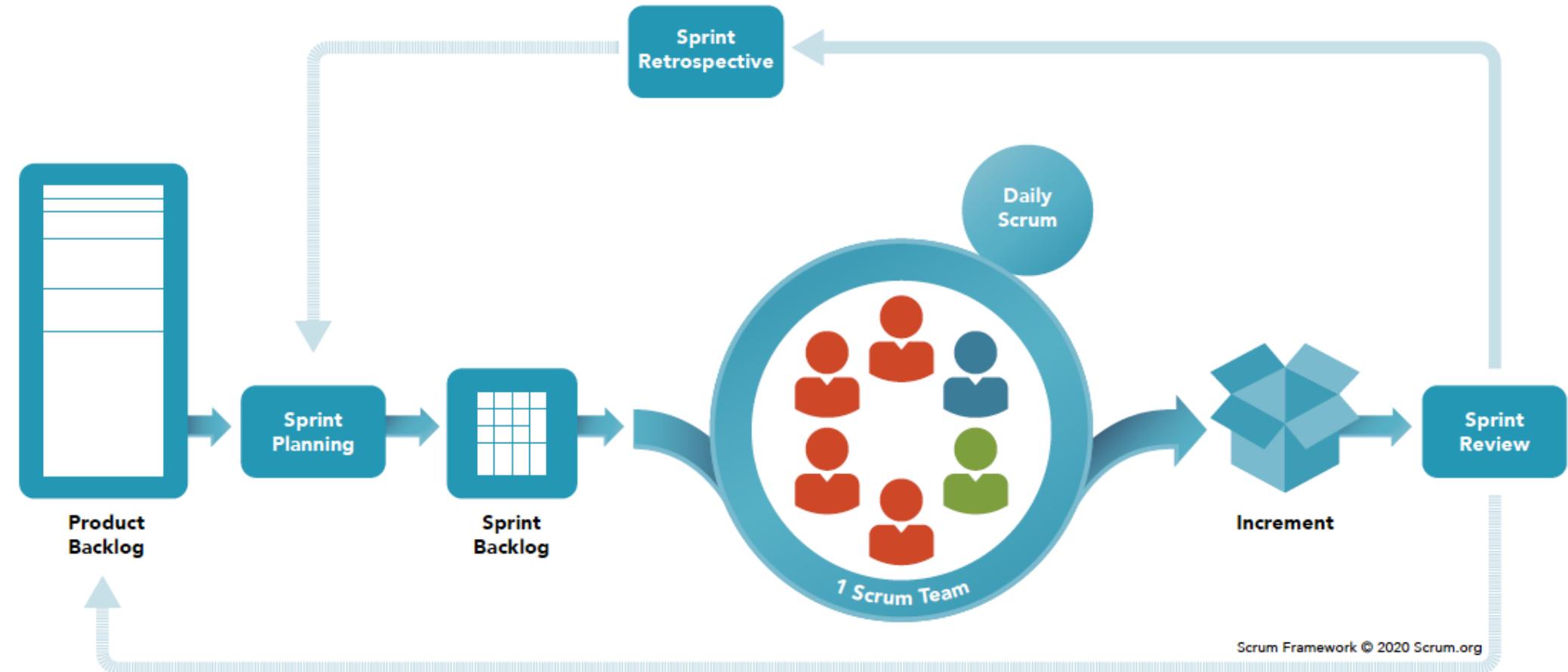


Sprint Planning Puanlama

- User Storyler Sprint Backlog 'a alınırken takım tarafından puanlanır. (POINT)
- Sprint başlangıç toplantısı veya refinement toplantılarında User Story okunur ve development team' in tüm üyeleri user story' e point verir.
- Verilen pointler kartlara yazılarak gösterilir.
- Pointler Fibonacci Serisindeki sayılara göre verilir.
 - 1 1 2 3 5 8 13 21 34 ...
- Her üye niçin bu point' i verdigini açıklar ve team' in ortak kararıyla user story' nin point' i belirlenir.
- Team' deki developer ve QA sayısına gore takım team kapasitesi capacity belirlenir.
- 1 point 1 günlük çalışma saatine göre belirlenir. (Ör: 8 saat)



SCRUM Process



Best Scrum Project Management Tools

