BATCH : 146 - 149

LESSON : **SDLC**

DATE : 11.07.2023

SUBJECT : **SDLC - 2**

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ

SDLC

Software || System Development Life Cycle

2. Ders

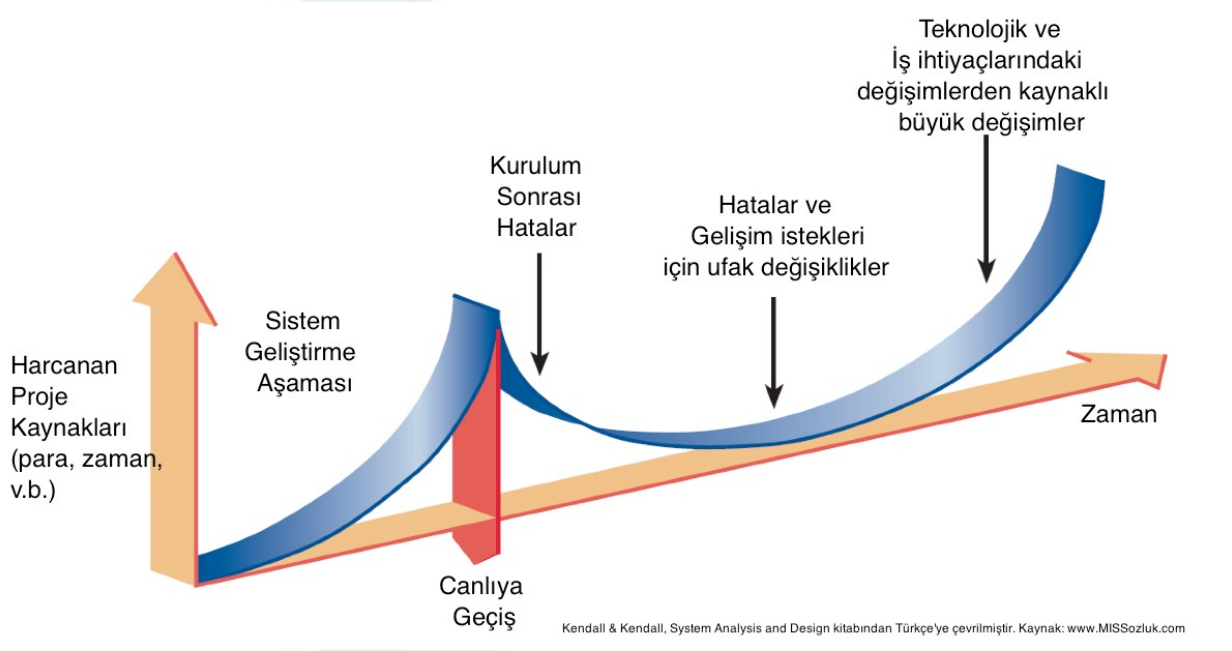
11.07.2023

B149 AWS & DevOps

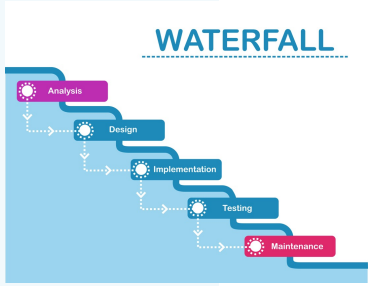
B146 Cyber Security

Bugün ne yapıyoruz?

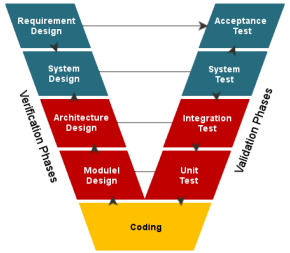
- SDLC Metodları nelerdir?
  - Waterfall
  - V Model
  - Spiral
  - Prototip
  - Agile
    - Scrum nedir?
    - Scrum Team
      - Product Owner (PO)
      - Scrum Master
      - Development Team
- Kahoot



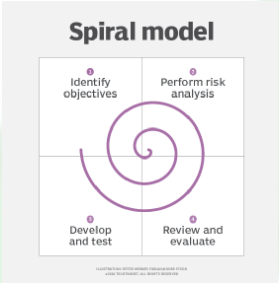
SDLC Metodları



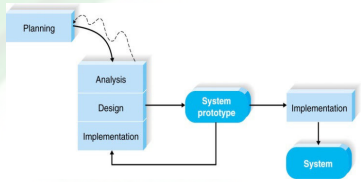
Waterfall
(Şelale)



V Model



Spiral



Prototip

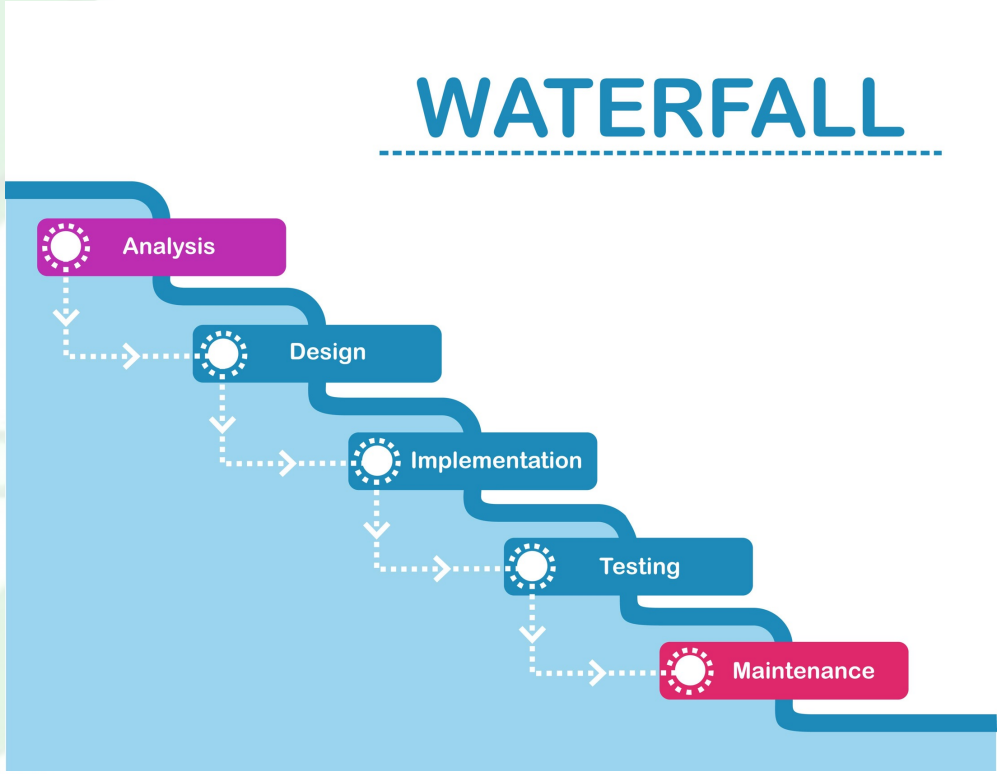
Agile
(Çevik)

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.

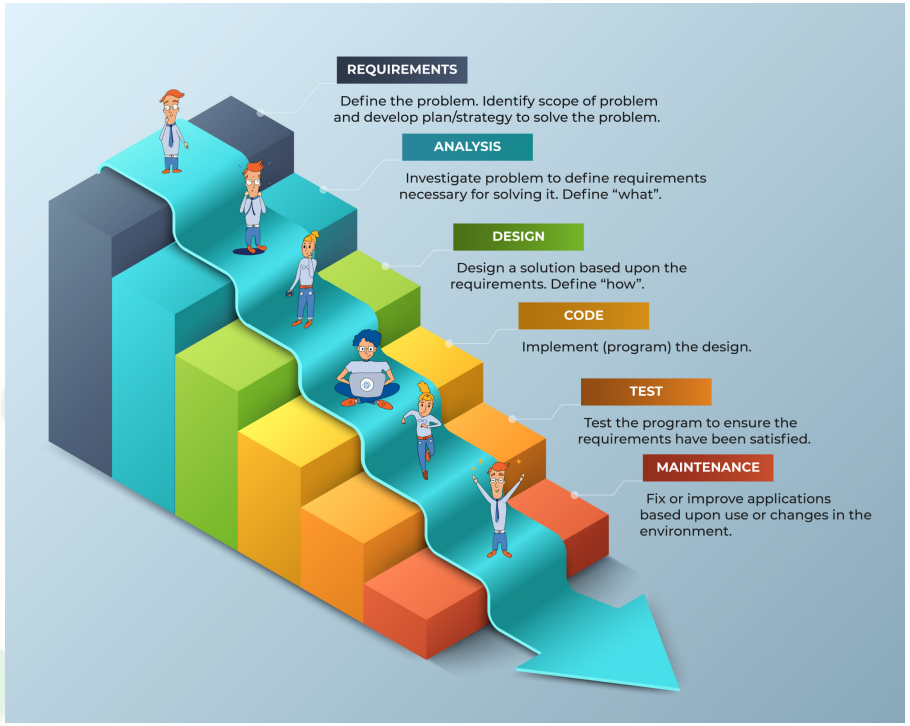
Waterfall (Şelale) Methodology

- Şelale modeli (Waterfall) proje yönetim süreci; analiz, tasarım, yazılım, test, yayın gibi fazlardan oluşur.
- Geleneksel bir yöntemdir; süreçler tıpkı 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 adımında, tüm yazılım gereksinimleri net bir şekilde belirlenerek analiz dokümanı üretilir.
- Daha sonra, tasarım adımında yazılımın arayüz, veritabanı, sınıf vb. tasarımları yapılarak tasarım dokümanı üretilir.
- Bir sonraki kodlama adımında yazılım analiz ve tasarım dokümanlarında belirtilen şekilde kodlanır.
- Test adımında 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

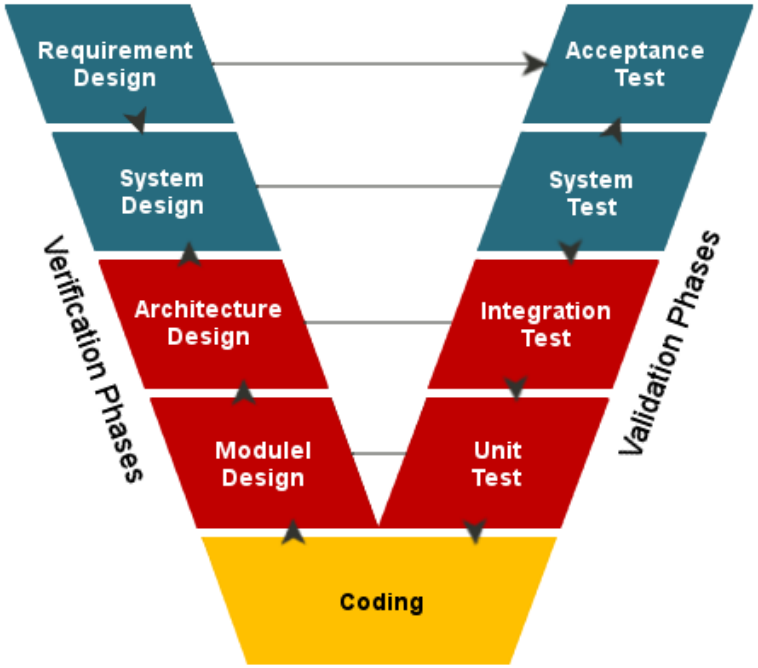
- Değişim ve yenilik **zordur**.
- Müşteri öngörü ve önerileri **önemsenmez**.
- Projenin bitimine kadar çalışan **ürün yok**.
- Beklenmedik riskleri kolayca **ele alınmaz**.

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 tasarımı gibi test faaliyetleri kodlamadan önce gerçekleştirildiği için proje içerisinde çok **zaman kazandırır**.
- Bu nedenle şelale modeline göre daha **yuksek 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 Methodology

Avantajlar

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

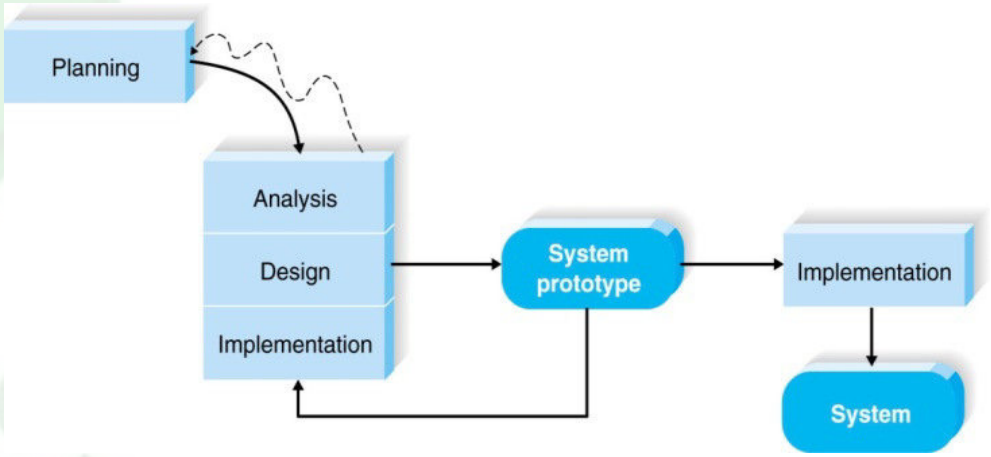
Dezavantajlar

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

Prototip Methodology

- Bu modelde çabuk tasarım, prototip geliştirme ve müşteri değerlendirilmesinden 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, pratiğe 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 yayınlamışlar, 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

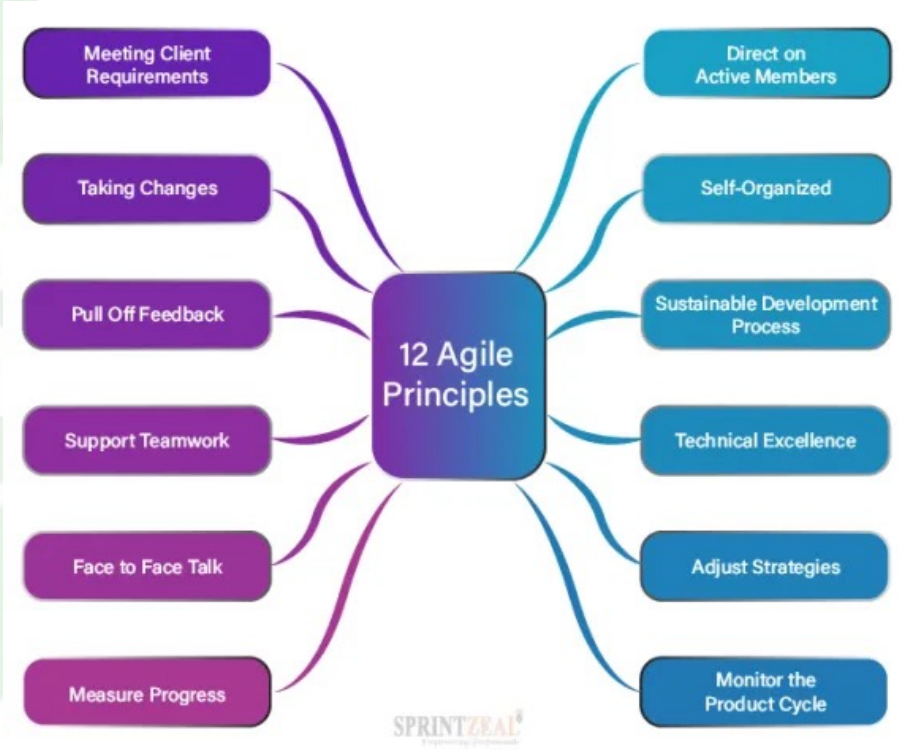
Agile Manifesto

- Önceliğimiz **kaliteli yazılımı geliştirerek** bunu müşteriye **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.
- Motive olmuş bireyler etrafında projeler oluşturun.
- Bir geliştirme ekibine ve içinde bilgi aktarmanın en verimli ve etkili yöntemi yüz yüze görüşmedir.

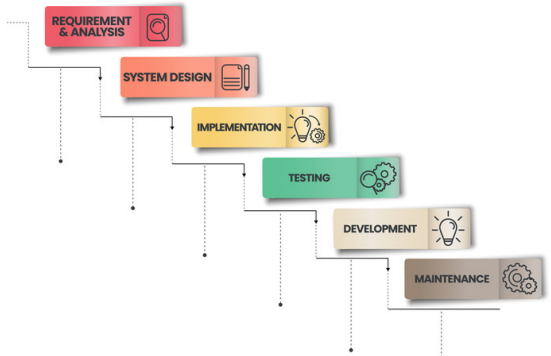
Agile (Çevik) Methodology

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**.

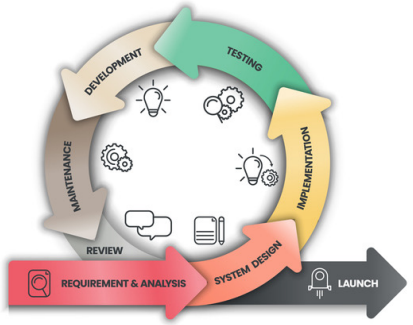


Waterfall vs Agile



WATERFALL
MODEL

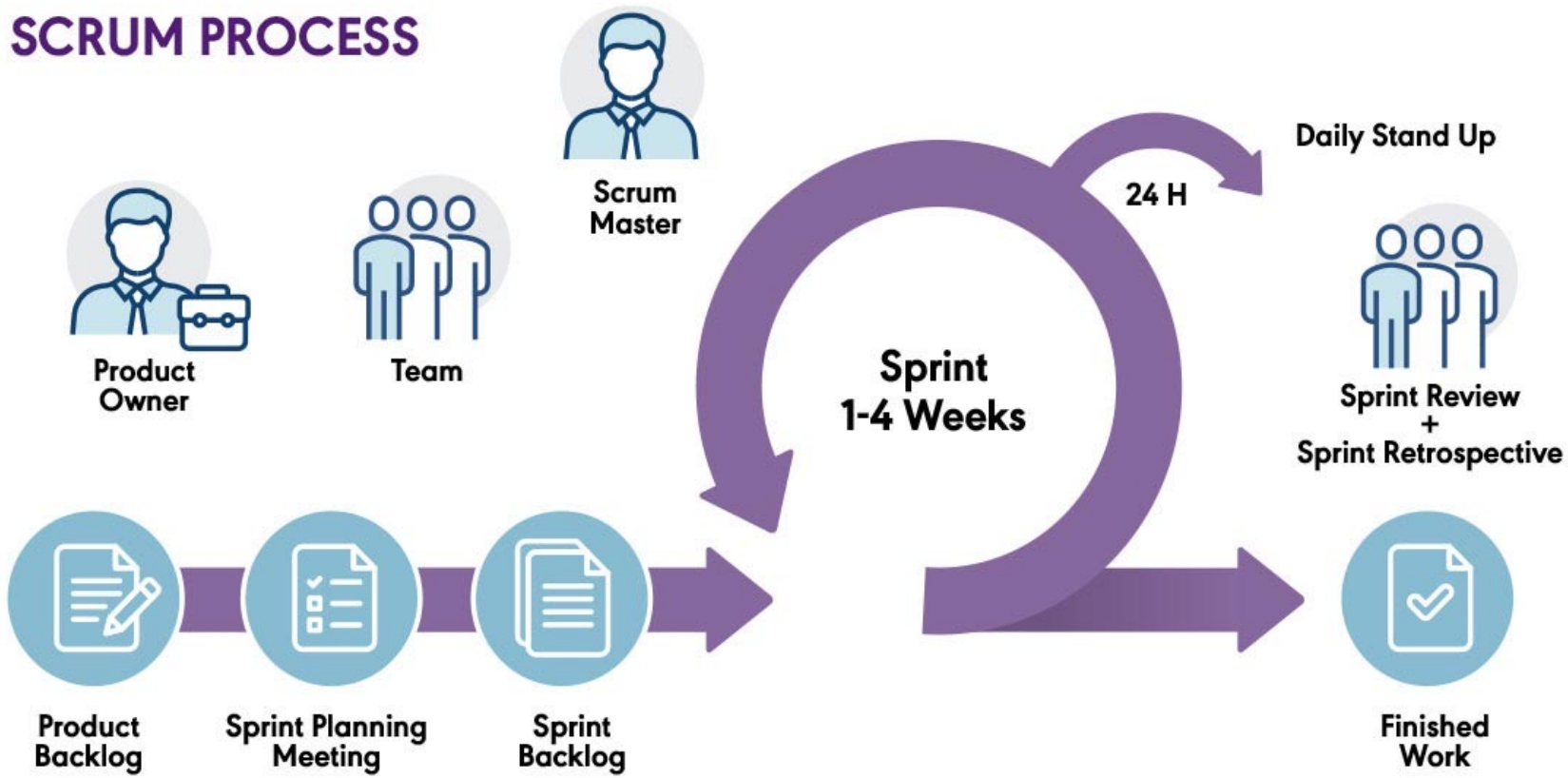
VS



AGILE
MODEL

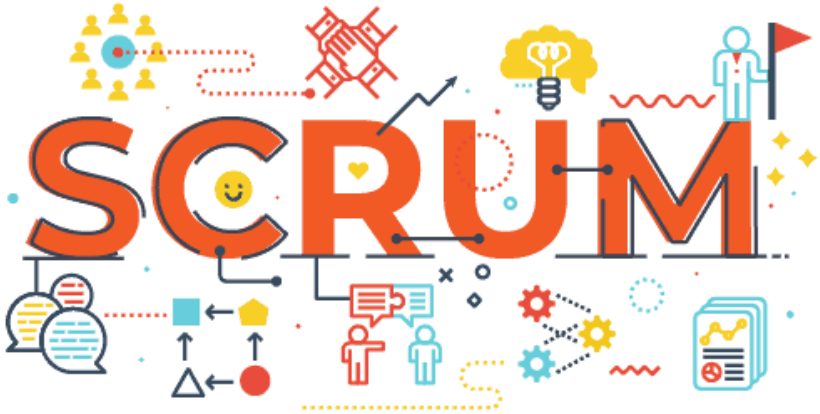
Method	Successful	Challenged	Failed
Agile	42%	50%	8%
Waterfall	26%	53%	21%

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

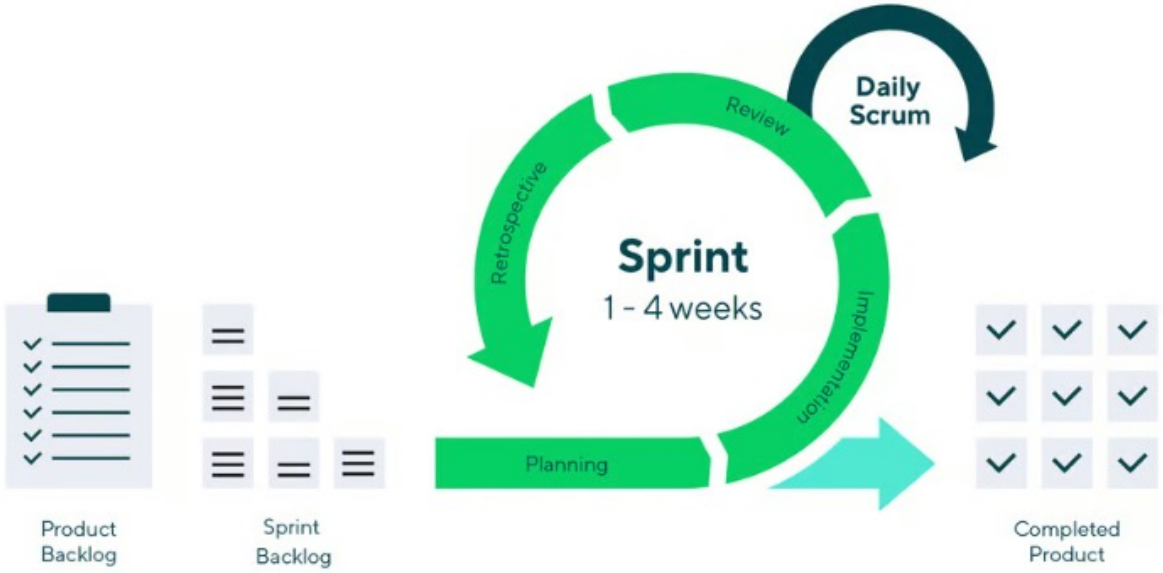
Sprint – Rugby Yaklaşım

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

Scrum

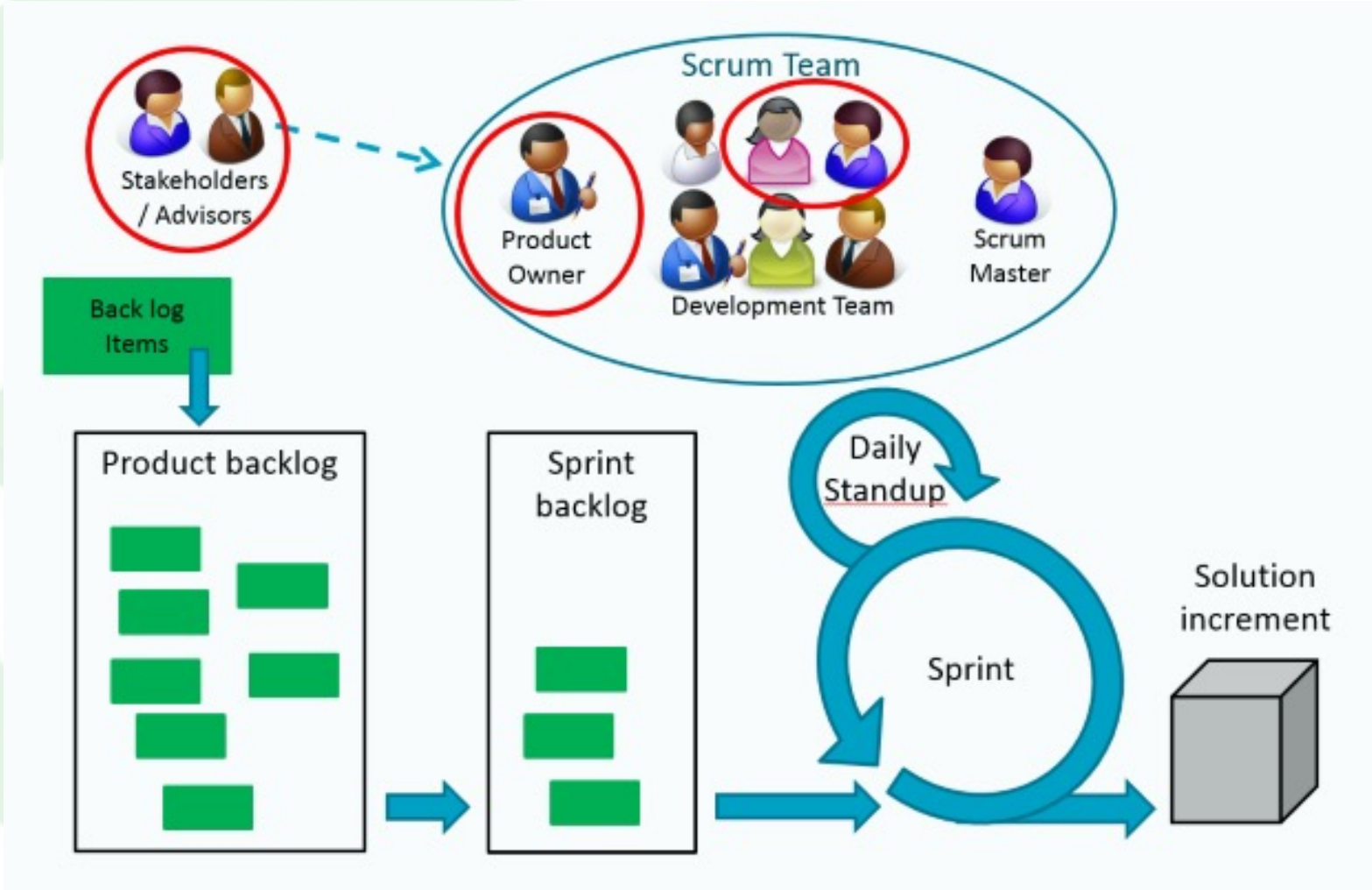
Sprint

- 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 tasarlanır, kodlanır ve test edilir.
- Yapılacak tüm işler, **Product Backlog** da biriktirilir, **Product Owner**' un belirlediği önceliğe göre Sprint Backlog' una alınır ve bir sprintte bitirilerek ürünün demosuna eklenir.



Scrum Team

- Product Owner
- Scrum Master
- Development Team



Product Owner

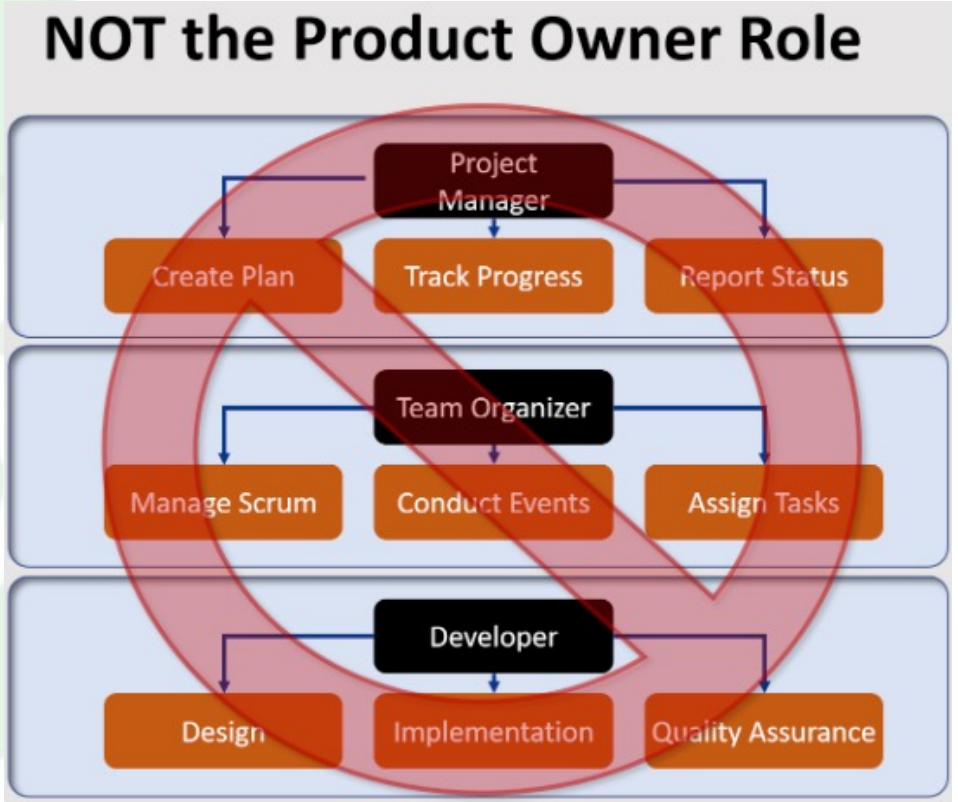
Kimdir ?

- Geliştirme takımı ve müşteri arasındaki iletişimi sağlar.
- Projenin önceliklerine göre **Product Backlog (iş listesi) oluşturur**.
- Ekipte **Stakeholders**' ı temsil eder. Geliştirme Takımı ve Stakeholders arasındaki iletişimi gerçekleştirir.
- Her Sprint' te hangi user story' lerin sprinte dahil edileceğine 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

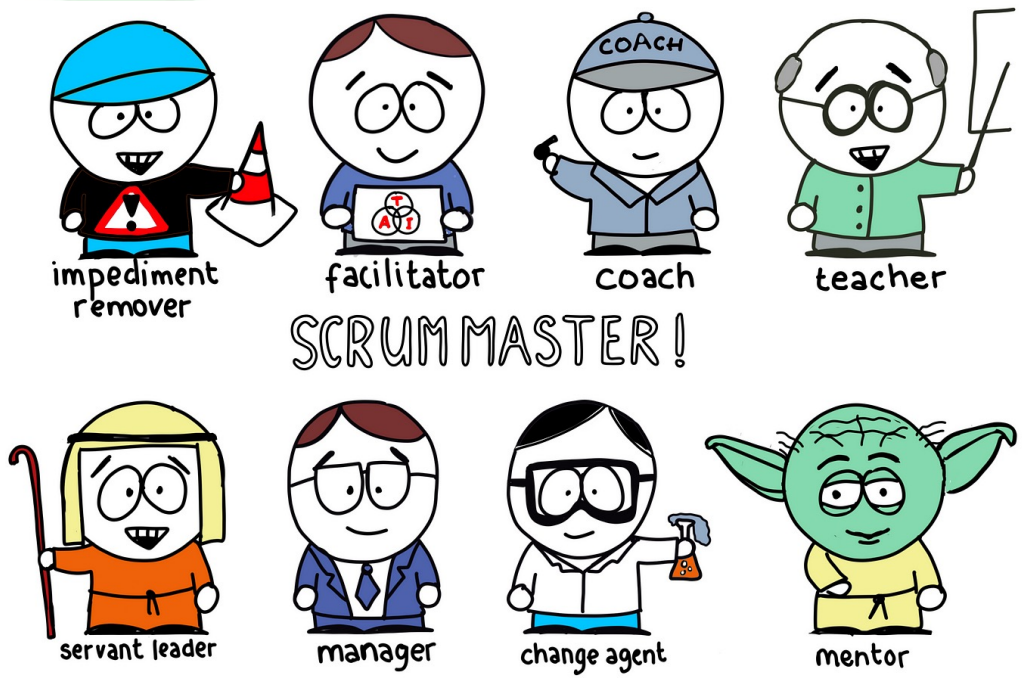
Kim değildir ?

- 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 karışabileceği **anlamına gelmez**.
- PO, iş-görev ataması **yapamaz**.
- İşin nasıl yapılacağına **karışamaz**!



Scrum Master

- Scrum kurallarını, teorilerini ve pratiklerini iyi bilir ve takımın bu kurallarını uygulamasından sorumlu kişidir.
- Takımın yöneticisi değildir.
- 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, takımı ve organizasyonu Scrum' a adapte eder."

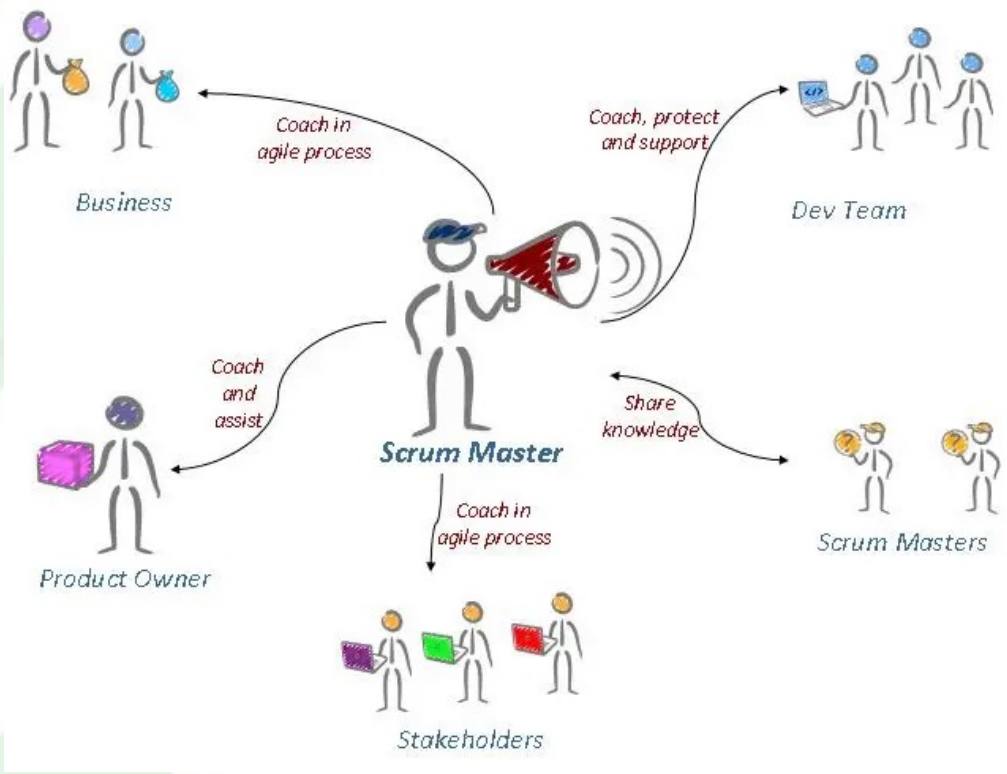


Scrum Master

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

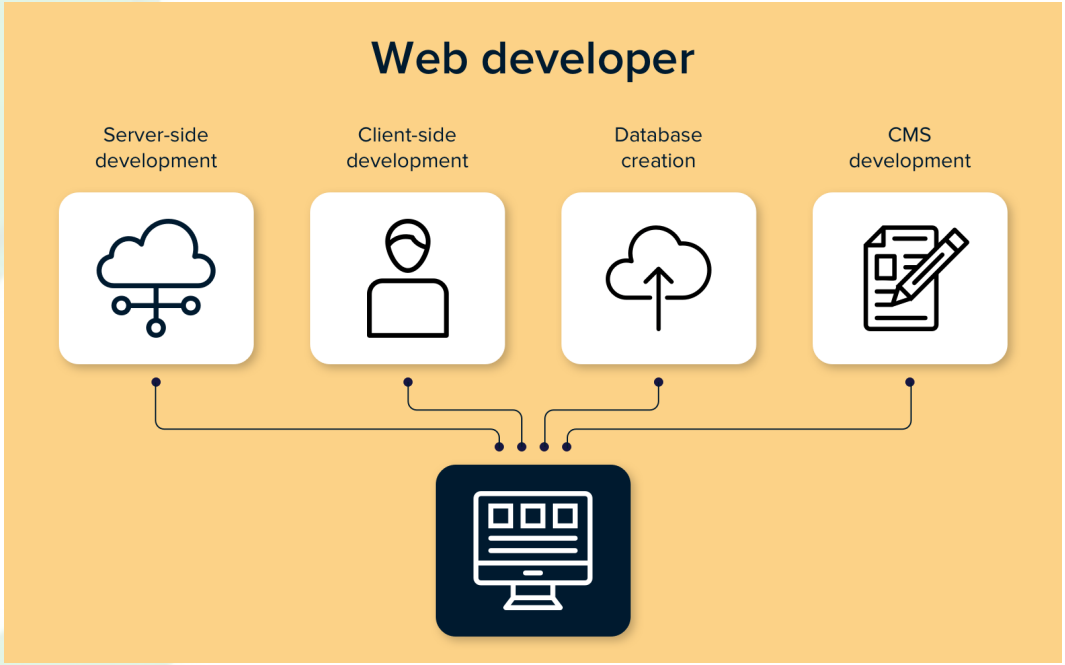
Scrum Master

- ScrumMaster, takıma rehberlik ve koçluk eder, karşılaşılan engelleri ortadan kaldırmalarına yardımcı olur.
- Takım içi harmoniyi, ekip elemanları arasındaki uyumu, iletişimi arttırmak 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 önceliğe göre yapan geliştirme takımıdır.
- Teknik konularda sorumluluk Development Team' e aittir.
- PO ve SM Development Team' in yapacağı işlerin önceliğini 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' ı Product Owner' ın 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.