T.C

İZMİR BAKIRÇAY ÜNİVERSİTESİ



YAZILIM-YAŞAM DÖNGÜSÜ

MAKALE

Şevval ÖZDEMİR

Bilgisayar Mühendisliği Bölümü

Danışman

Doç. Dr. Deniz KILINÇ

Özet

Bu çalışmanın amacı yazılım yaşam döngüsünü hakkında bilgilendirme yapmaktır. Bu makalede yazılım

yaşam döngüsünün ne anlama geldiği, döngü içerisinde olan aşamaları, yazılım döngü modellerinin

neler olduğu ve ne işe yaradığı, bu modellerin hangi projelerde kullanılması gerektiğine dair

bilgilendirme içerir. Günümüzde çeşitli yazılımlar geliştirilmekte ve bunun için yazılımlar oluşturulurken

geçirmesi gereken aşamalar bulunmaktadır. Bu aşamaların ne olduğuna dair bilgisi olmayan

olmayan yazılımcıların yapacağı yazılımlar çok da işlevsel olmayacaktır. Bu makale herhangi bir yazılım

geliştirilmeden önce okunmalıdır.

**YAZILIM YAŞAM DÖNGÜSÜ NEDİR?**

Oluşturduğunuz bir yazılımın başından sonuna kadar gelişen aşamalar ve bu aşamaların bütününe yazılım yaşam döngüsü denir. Yazılımı yazan kişiler karşılaşacağı hatalar ve eksikliklerde döngünün başına dönebilir. Her yazılım geliştirilmeye ve değiştirilmeye açık olduğundan döngü defalarca tekrar edilebilir. Bu döngülerden yararlanıldığı için yazılımı oluşturmak daha rahat ve kolay gerçekleşir.

Yazılım yaşam döngüsünün aşamaları vardır:

**1)Planlama**

İlk adım olan planlamada oluşturulacak proje hakkında kabataslak bir planlama yapılır. Programın amacı, konusu ve ne işe yarayacağı belirlenir. Bu belirlemeler sonucunda görev dağılımları yapılır ve proje planlanmış olur.

**2)Analiz**

Analiz aşamasında projede karşılaşılacak problemler, projenin bitimi için mevcut olan zaman dilimi, projeyi yapacak grupların oluşturacakları görüşmeler vs. sağlanır. Sürecin nasıl gelişeceğine dair genel bir değerlendirme yapılan aşamadır.

**3)Tasarım**

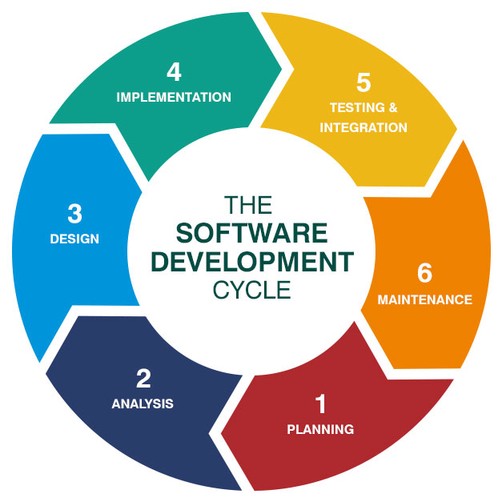
Analiz çalışması sonrasında çıkan sonuçlardan yararlanılarak projenin detaylarına inilir. İki çeşit tasarımdan söz edebiliriz: üst seviye ve mimari tasarım, detaylı tasarım. Üst seviye ve mimari tasarımda önerilen sistemin yapısı anlatılır (akış şemaları, diyagramlar ve modüller…). Detaylı tasarımda ise mimari tasarımın revize edilmiş hali ve kullanılacak programlama dilleri, algoritmalar gibi şeyler belirlenir.

**4)Gerçekleştirme**

Bu aşamada modeller kodlanır, test edilir ve kurulur. Bu aşama üretim ve test aşaması olarak da bilinir. Bir nevi de geliştirme aşamasıdır. Gerçekleştirme aşamasından sonra herhangi bir değişiklik yapılmamalı, müşteriye teslim edilecek konuma gelmelidir. Erken test etme yaklaşımına analiz aşamasında başlarsak hata yapma oranımız düşer ve daha güzel bir sonuç elde ederiz.

**5)Bakım**

Program teslim edildikten sonra kullanıcının istekleriyle doğrultulu olarak belli zaman aralıklarında güncelleme ve hata giderme yapılmalıdır. Bakım aşamasında hatalar giderilir, program iyileştirilir ve yeni işlevler eklenebilir. Bu aşama program emekliye ayrılana kadar devam edebilir.



**YAZILIM YAŞAM DÖNGÜ MODELLERİ**

Yazılım yaşam döngüsünde birden fazla model vardır. Bu modeller proje oluştururken neyin hangi sırada yapılacağı gibi etkenleri belirler ve işimizi kolaylaştırır. Bu sayede programı hazırlarken oluşabilecek karmaşa ve hata riski azalmış olur. Yazılım yaşam döngü modelleri şunlardır:

**1)Gelişigüzel Model**

Bu model 1960’larda kullanılmıştır. Herhangi bir model ya da yöntem bulunmaz. Genellikle kişiye özel yapılır ve basit programlama içerir. Kişiye bağlı olduğu için de herhangi bir bakım yapılması oldukça zordur. Günümüzde bu model kullanılmamaktadır.

**2)Barok Modeli**

1970’li yıllarda ortaya çıkan bu modelde yaşam döngüsü aşamaları doğrusal bir şekilde geliştirilir. Döngü yoktur ve günümüzden farklı olarak belgeleme işlemi ayrı olarak gerçekleştirilir. Aşamalar arasında geri dönüşlerin nasıl yapılacağı tanımlanmamıştır. Günümüzde bu model de gelişigüzel model gibi kullanılmamaktadır.

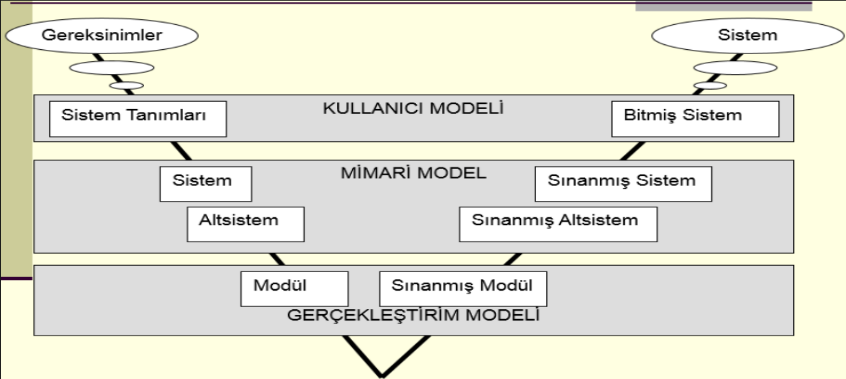
**3)Çağlayan Yaşam-Döngü Modeli**

Bu model geçmiş zamanlarda en yaygın modellerden biri olarak bilinir. En temel modeldir. Çağlayan yaşam döngü modeli çok iyi tanımlanmış ve kısa sürede bitebilecek projeler için kullanılır. Bu modelde her aşama en az bir kere tekrar edilir. Çağlayan yaşam döngü modelinde işlemler adım adım yapılır, her safha sonunda dokümantasyon yazılır; test ve dokümantasyon yanlış veya eksik olmuşsa o safha tamamlanmış olarak sayılmaz. Bunlara rağmen modelin kullanımı günümüzde gittikçe azalmaktadır. Çağlayan yaşam döngü modelinin barok modelinden farkı;

* Çağlayan yaşam modelinde belgeleme sürecin içerisindeyken barok modelinde belgeleme ayrı bir süreç içerisinde yer alır.
* Çağlayan yaşam modelinde barok modeline göre geri dönüşlerin nasıl yapılacağı tanımlıdır.
* Çağlayan yaşam modelinde analiz ve tasarım aşamalarında diğer aşamalara göre daha fazla zaman harcanır. Bu yüzden yazılımın kullanıcıya ulaşması daha uzun zaman alır.

**4) V Süreç Modeli**

Bu modelde belirsizlikler az ve iş tanımlamaları belirgindir. BT (Bilgi Teknolojileri) projeleri için uygun bir modeldir. V süreç modeli çağlayan yaşam döngü modelinin gelişmiş hali olarak düşünülebilir. Aşamalarda doğrusal ilerlenmez, kodlama evresinden sonra yukarıya doğru kıvrılarak V şeklini oluşturur.



Yukarıda gördüğünüz modelde sağ taraf üretim, sol taraf sınamayı temsil etmektedir. Eğer test işleminde hata bulunursa sol taraftaki adıma dönülerek hata giderilebilir. V süreç modelinin 3 temel çıktısı vardır:

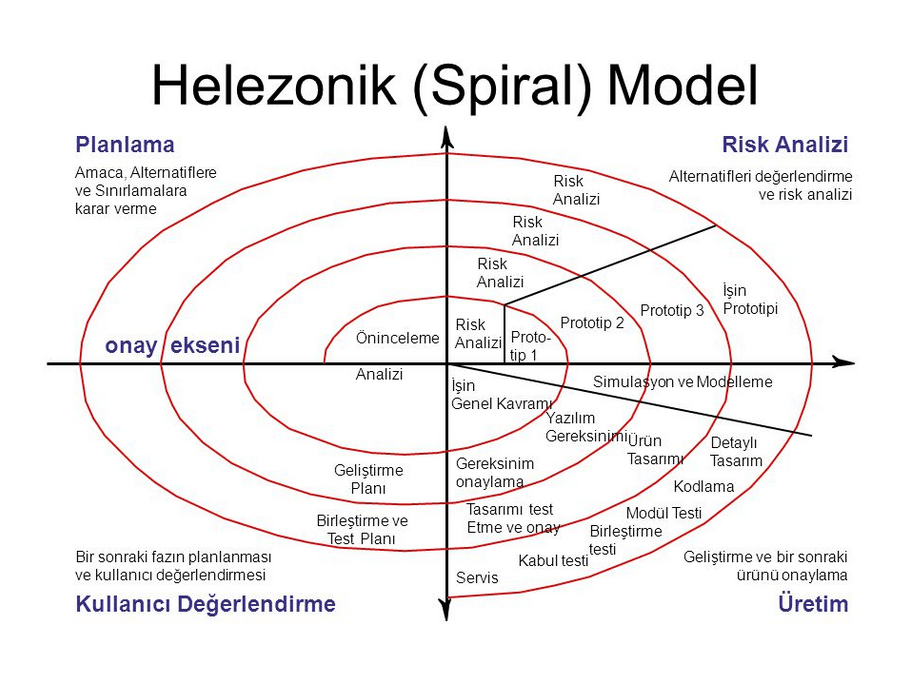
**a)Kullanıcı Modeli:** Geliştirme sürecinde kullanıcının istekleri göz önünde bulundurularak sınama ve planlanma yapılır.

**b)Mimari Model:** Sistem tasarımı, alt sistem tasarımı ve bunların bütün sistemin test işlevidir.

**c)Gerçekleştirme Modeli:** Yazılımın kodlanması ve test edilmesiyle ilgili fonksiyonlar.

**5) Helezonik (Spiral) Model**

Helezonik modelde diğer modellere göre risk analizi ön plandadır ve prototip oluşturur. Prototip oluşturma her aşamadadır ve bu yüzden kullanıcıdan alacak hatayı kısa bir süre içerisinde düzeltebilir. Helezonik modelde her döngü bir fazı ifade eder. Gereksinim, analiz.. vs. gibi bir faz yoktur. Yinelemeli artımsal bir yaklaşım vardır. Yani ilk denemede bir ürünü hızlı bir şekilde teslim etmek yerine yavaş ve hatasız bir yol izlemeyi teşvik eder.



Helezonik model 4 bölümden oluşur:

**a)Planlama:** Her aşamada olan ürün için planlama yapılır ve amaç belirlenir.

**b)Risk Analizi:** Oluşabilecek riskler belirlenir ve bunun için çözümler üretilir.

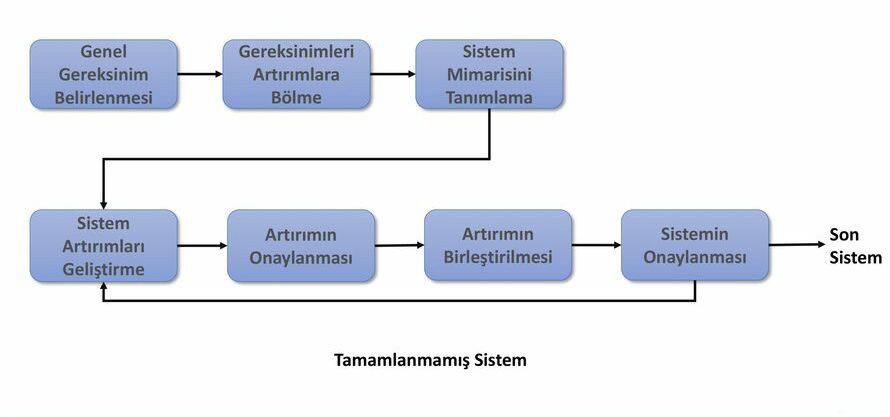
**c)Üretim:** Bu aşamada ara ürün üretilir.

**d)Kullanıcı Değerlendirmesi:** Kullanıcı ara ürün hakkında inceleme ve değerlendirme yapar.

Helezonik modelin avantajlarından bazıları; kullanıcı yazılımı kullanmaya başladığında erken tespit yapıp ileride oluşabilecek herhangi bir hata oluşumu görüp engelleyebilir. Bir yazılım mühendisi tarafından yazılımın kodlanması ve test edilmesi kısa bir süre içinde olur. Helezonik modelin olumsuz özelliklerinden bazıları ise; küçük ve riskli projeler için çok pahalı olması, kısa süreli projeler için uygun olmaması ve farklı dokümantasyondan oluşmasıdır.

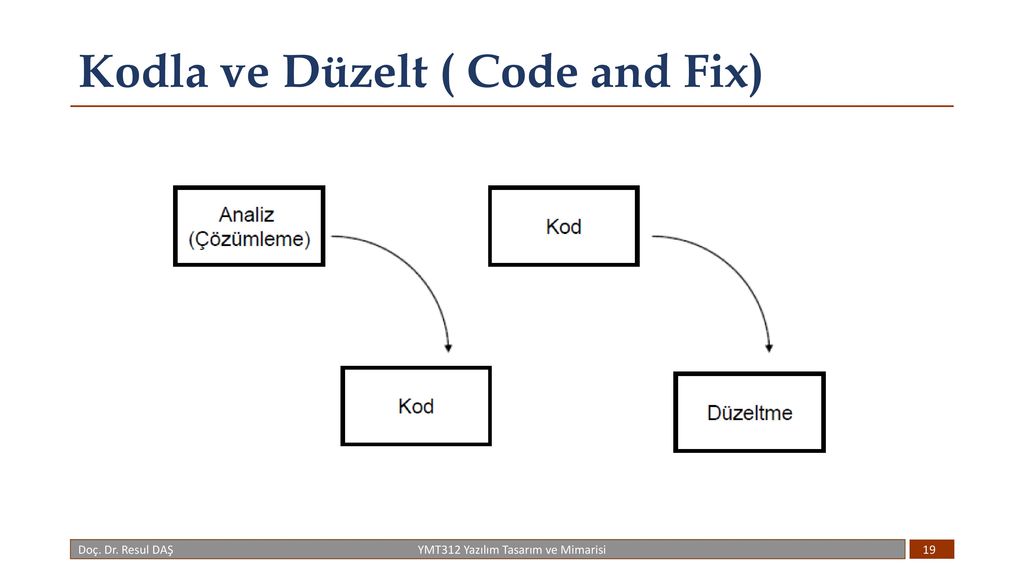
**6)Artımsal Geliştirme Süreç Modeli**

Artımsal geliştirme modelinde yapılacak proje parçalara bölünür ve yapılacak parçalar kullanıcının önceliklerine göre sıralanır. Her bir ara ürün bittiğinde kullanıcıya verilir ve kullanılır. Fakat bu ara ürünler bir sonraki ara ürün bitmeden değiştirilemez. Sadece bittikten sonra bir sonraki ara yüzde hatalar veya eksikler giderilebilir. Bu artmış olur. Artımsal projenin bu özelliği avantaj sağlarken bir yandan da uzun sürmesi, biri bitmeden diğerlerinin gereksinimlerine bakılmaması gibi dezavantajları da bulunur. Yazılım geliştirme revize edilerek olur. Her seferinde sonuca ve doğru olana biraz daha yaklaşılır. Artımsal geliştirme modeli bu yüzden yazılım geliştirme için doğru bir modeldir. Bu modelde ara ürünler için ayrı ayrı aşamalar olmak üzere ilk aşama prototip görevi görmelidir, diğer aşamalar da prototipten yararlanılarak geliştirilmelidir. Ayrıca bu modelde öncelikli olan gereksinimler, doğruluk ve kesinliğin olması için daha çok testten geçer.

Artımsal geliştirme süreç modelinin taslağı yandaki gibidir. Sürekli gelişen ve aynı zamanda kullanıcıya yazılımı kullandıran bir model olduğu için projelerde çokça tercih edilen bir model olarak bilinir.

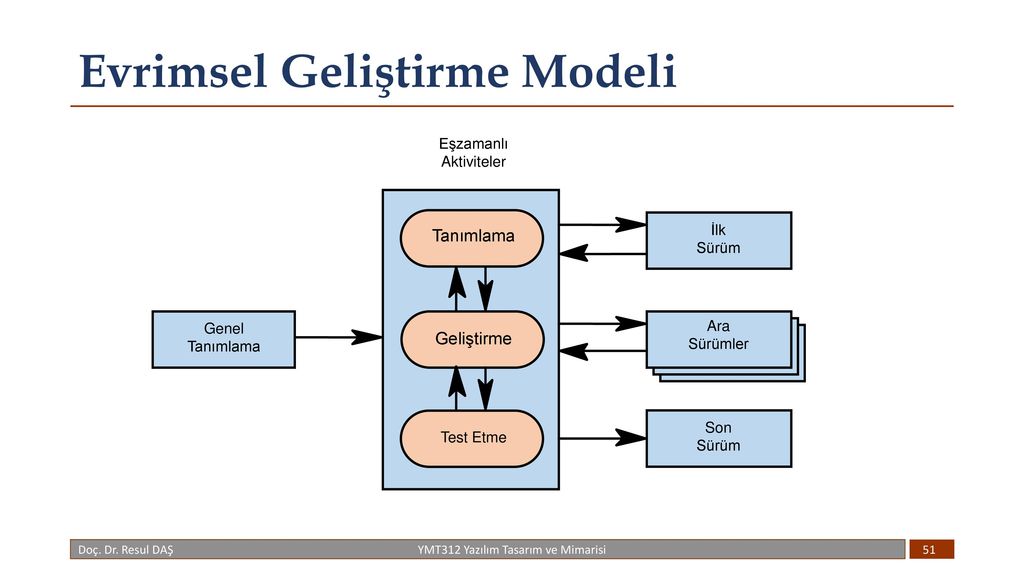
**7)Kodla ve Düzelt Yaşam Döngü Modeli**

Kodla ve düzelt yaşam döngü modeli küçük programlar için kullanılabilir, genellikle büyük projeler için kullanılmaz. Bu modelde ilk olarak yazılımın ilk safhası geliştirilir. Daha sonra gereksinim ve eksikliklere bağlı olarak değişim ve düzeltme yapılır. Bu modelin bakım evresi mevcuttur fakat çok zordur. Bunun sebebi ise bu sisteme ait dokümantasyonun bulunmamasıdır. Ayrıca bu modelin bir emeklilik safhası vardır. Uzun vadeli bir model değildir. Yazılım geliştirme bu modelde diğerlerine göre çok daha kolay ve maliyetlidir.



**8)Evrimsel Geliştirme Modeli**

Bu model ilk tam ölçekli olan modeldir. Başarısı ilk evrimin başarısına bağlıdır. Büyük firmalar için önerilen bir modeldir. Her aşama sonrasında ortaya çıkarılan ürün tam işlevselliğe sahiptir, üzerinde bir değişiklik yapılamaz.



Evrimsel geliştirme modeli coğrafik olarak geniş alanlara yayılmış, çok birimli organizasyonlar için önerilmektedir. Bu modelin eksikliği değişiklik denetimi ve konfigürasyon yönetimidir.

**Scrum Nedir?**

Scrum, yazılım geliştirme süreçlerinde yüksek oranda tercih edilen bir agile (çevik) proje yönetim metodolojisidir. Scrum 1990’ların ortalarında Jeff Sutjerland ve Ken Schawaber tarafından geliştirilmiştir. Kompleks yazılım sürecinin kolaylaşması için kullanılır. Bütünü parçalamaya yönelik ilerler. Büyük ve karmaşık projelerde problemleri parçalara ayırıp parçaya yönelik sonuçlar elde etmeyi amaçlamaktadır. Düzenli zaman aralıklarında geri bildirim yapmayı sağlar. Bu açıdan müşteri gereksinimleri ön planda tutulur ve ona göre şekillendirme yapılır. Scrum tekrara dayalı bir çerçevedir. Scrum’da takım çalışması ve toplantılar çok önemlidir. Scrum metodolojisinde bir tekrarlanmanın tamamlanması maksimum 30 gün sürmektedir ve her gün 15-20 dakikalık toplantılar sayesinde iş takibi iyi bir şekilde yapılmaktadır. Scrum’da karmaşıklığın giderilmesi için uygulanan 3 teknik vardir:

**1)Şeffaflık:** Projenin ilerleme sürecinde düzenli ve sık aralıklarla toplantılar olup projenin nasıl ilerlediğine, eksikliklerinin neler olduğuna ve sonucun nasıl olacağına dair bilgilendirme doğru ve herkese açık bir şekilde yapılmalıdır.

**2)Gözlem(denetleme):** Projede ara ürünler teslim ve gelişme halinde olmalı, projenin ilerleme süreci sık sık kontrol edilmeli ve herhangi bir hata karşısında çözüm üretilmelidir.

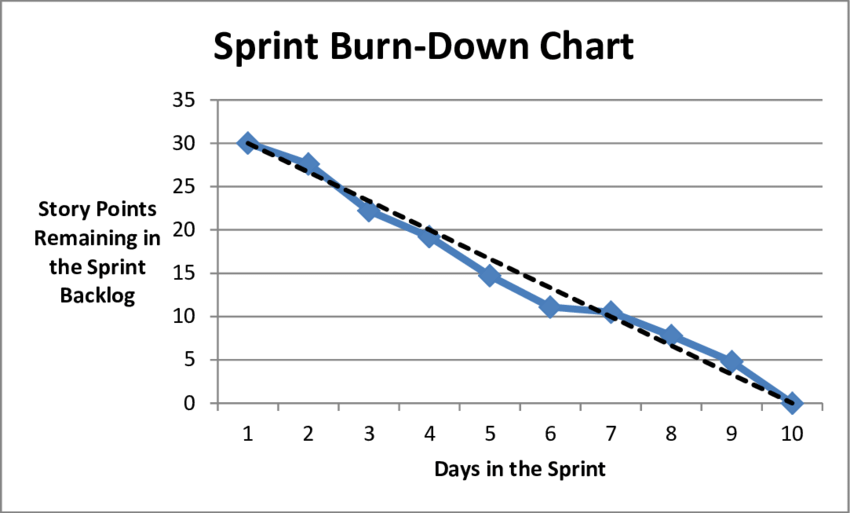
**3)Adaptasyon(uyarlama):** Projenin gidişatı her zaman istenildiği gibi olmayabilir ve bu durumda değişimlere uyumlu olmak gerekir.

**Scrum Yapısı İle İlgili Kavramlar**

**1)Product Backlog(ürün iş listesi):** Projenin içerisindeki gereksinimlerin olduğu bir listedir. Ürün sahibi bu listeyi alır ve müşteri gereksinimlerinin öncelikleri sırasına göre sıralar.

**2)Sprint(koşma):** Projenin küçük kısımlara ayrılmış haline sprint denir. Proje süreci içerisinde 1-2 haftalık zaman dilimlerine ayrılır.

**3)Scrum Board(scrum tahtası):** Projede sprint içinde yapılacak olanlar bu bölümde maddelere yer verir. Örneğin; yağılacak olanlar “to do” listesine alınır, iş test için hazırlanıyorsa “to verify” durumuna getirilir.

**4)Burndown Chart(açılış tablosu):** Bu tablo dikey eksen sprintte kalan işi; yatay eksen ise sprintin günlerini temsil eder. Bu tablo şeffaflık için katkı sağlar. Burndown tablosu aşağıdaki gibidir;

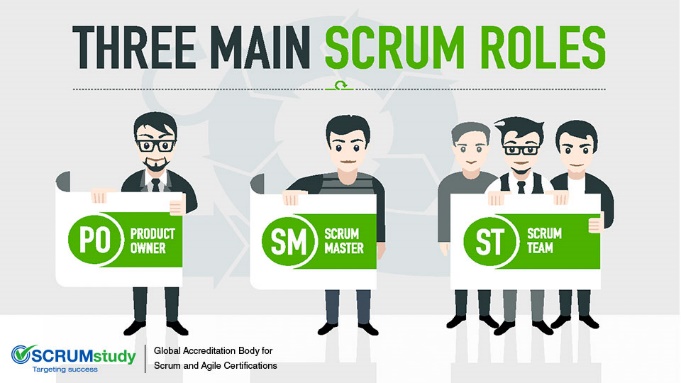
**ROLLER**

**Pig Roller:** Projeyi yapan asıl kişilerdir. Projenin her şeyiyle ilgilenirler. Bunlar üçe ayrılır:

1)Product Owner (ürün sahibi): Projede iş içerisinde geri dönüşümü sağlayan kişidir. Product backlog oluşturur. Projeyi ve projenin özelliklerini tanımlayan kişidir. Product owner aynı zamanda sprinti iptal etme yetkisine sahiptir.

2)Scrum Master(scrum yöneticisi): Scrum yöneticisi, proje yapılırken scrum kurallarını koyan kişidir. Takımı rahatsız edecek, verimi düşürecek etkenleri ortadan kaldırmakla görevlidir.

3)Geliştirme Takımı: 5-7 kişi arasında olan bir takımdır. Projede yapılacak işlerin tamamından sorumludurlar. Sürekli toplantılar düzenleyip görev dağılımıyla birlikte proje hakkında çalışma yaparlar. Kendilerine görev verilmesini beklemezler. Kendi görevlerini kendileri verirler. Projenin geliştirilmesi için sorumluluk geliştirirler.



**TOPLANTILAR (Meetings)**

**1)Sprint Planning (sprint planlaması):** Product owner, Scrum master ve Scrum Team’ın katıldığı bu toplantıda scrum team; product backlog içerisindeki gereksinimler için bu toplantıda görev dağılımı yapar. Bu toplantıda büyük gereksinimler tanımlanır ve listesi çıkartılır. Proje içinde harcanabilecek tahmini bir maliyet değeri hesaplanır. Görev dağılımları için takımlar belirlenir. Takımdaki insanlar kendine uygun olan görevleri seçer. 1-3 haftalık sprintler oluşturulur.

**2)Daily Scrum (günlük scrum toplantısı):** Her gün aynı zamanda aynı yerde yapılan kısa süreli toplantılardır. Bu toplantılarda bir gün içinde geçen durum değerlendirmesi yapılır ve sonraki bir güne kadar neler yapılacağı hakkında münakaşa yapılır. Bu toplantılar çoğunluğun gelmeme durumu harici her gün yapılmaktadır.

**3)Sprint Review (gözden geçirme):** Her sprint sonunda yapılan toplantıdır. Yapılan sprint gözden geçirilir. Herhangi bir hata veya eksik varsa giderilir. Sonrasında yapılacak sprintler ile ilgili gereksinimler belirlenir ve nasıl yapılacağına dair önbilgi oluşturulur.

**4)Sprint Retrospective:** Bu toplantıdasprint boyunca yapılanların doğruluğu, hata oranı, kalitesi ve kesinliği hakkında değerlendirme yapılır. Daha iyisini yapmak ve gelişmek için araştırmalar ve uğraşlar gerçekleştirilir.

**Scrum Günümüzde Neden Popüler?**

Scrum’ın günümüzde popüler olmasının sebebi karmaşık ve uzun projeleri çok daha kolay ve işlevsel hale getirmesidir. Scrum,takım çalışması, verimlilik, proje yapımında kolaylık sağlar. Bunun gibi bir çok olumlu etmen Scrum’ın popüler ve kullanışlı olmasını sağlamıştır.

**KAYNAKÇA**

* [**https://bilginc.com/**](https://bilginc.com/)
* [**https://www.argenova.com.tr/**](https://www.argenova.com.tr/)
* [**https://www.linkedin.com/**](https://www.linkedin.com/)
* [**https://ubys.bakircay.edu.tr/**](https://ubys.bakircay.edu.tr/)
* [**https://toptalent.co/**](https://toptalent.co/)
* [**https://www.mshowto.org/**](https://www.mshowto.org/)
* [**https://medium.com/**](https://medium.com/)