

## Nedir Bu Yazılım Yaşam Döngüsü?

Nasıl ki doğada var olan her bir ürünün yaşam süreci bulunuyorsa aynı şekilde yazılımlarında bir yaşam süreçleri vardır. Yazılım projelerinin hayata geçirilmesi düşüncesiyle birlikte izlenilecek bir yol haritası oluşur. Bu haritayı planlama, analiz, tasarım, gerçekleştirme ve bakım süreci olarak ifade edebiliriz. Yazılımın nasıl geliştirileceği, sürdürüleceği ve daha iyi hale nasıl getirileceğini açıklayan bir plandan oluşmaktadır.

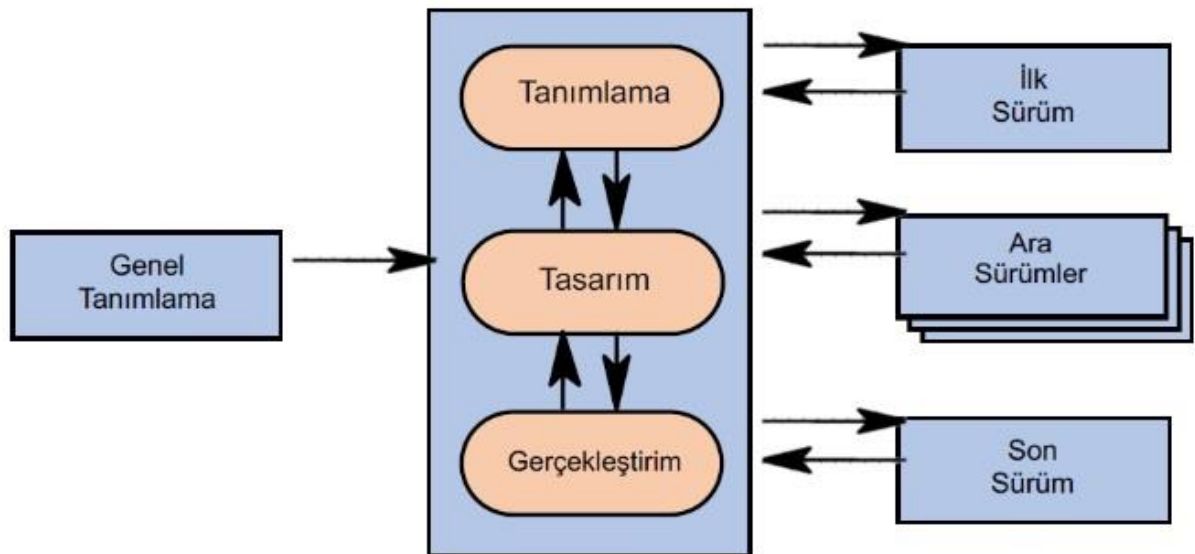
Bu aşamalardan kısaca bahsedecek olursak;

- İlk olarak temel ihtiyaçların belirlendiği yazılım projesinin planlaması ve görev dağılımı yapılır.
- Daha sonra yaşam döngüsünün en önemli aşamalarından biri olan analiz sürecinde projenin ne kadar süreceği, hangi risklerin olabileceği gibi temel sorunlar belirlenir.
- Analiz sonucu ortaya çıkan detaylarla yazılım projesinin nasıl sürdürüleceği tasarlanır.
- Yazılım projesinin kurulum çalışmalarının yapıldığı gerçekleştirme aşamasında kodlama ve test kısmı ele alınır.
- Yazılım ürünü tüm aşamaları tamamlayıp sunulduktan sonra bakım aşaması başlar. Bakım aşamasında genelde güncellemeler yapılır.

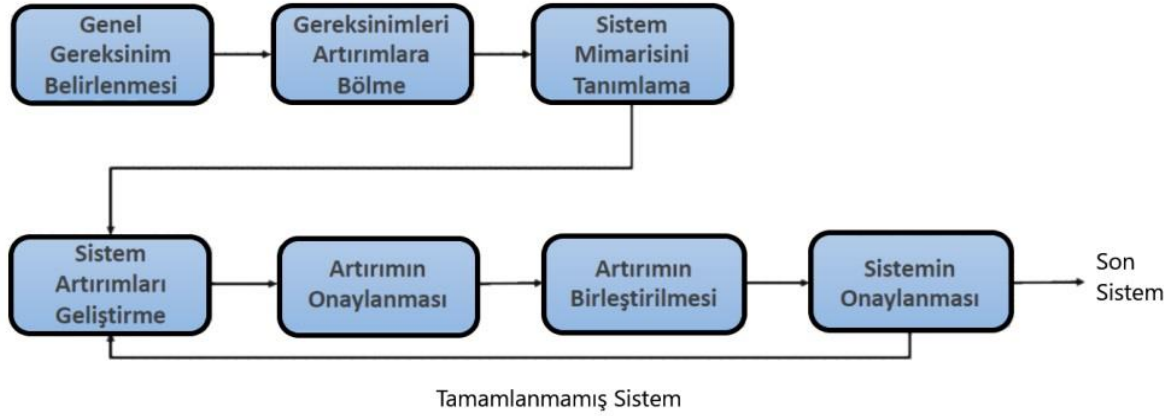
## Yazılım Yaşam Döngüsü Modelleri

Yazılım yaşam döngüsünde projenin büyüklüğü, kimler için hangi amaçla kullanılacağı gibi durumlardan dolayı tek bir model değil de daha fazla model kullanılmaktadır. Bazı yazılım geliştirme yaşam döngüsü modellerine bakalım.

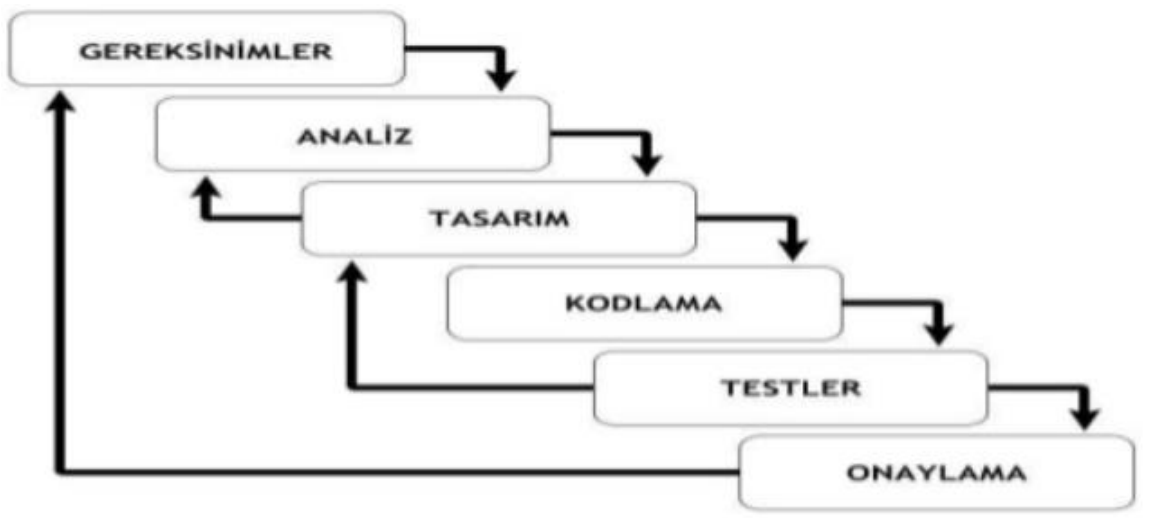
**Evrimsel Model :** Evrimsel süreç modeli, daha çok coğrafik olarak geniş alana yayılmış, çok birimli organizasyonlar, büyük firmalar için önerilir. Her aşamada üretilen ürün tam işlevselliğe sahiptir. Modelin başarısı ilk evrimin başarısına bağlıdır. Evrimsel geliştirme modelinde kullanıcılar gereksinimleri daha iyi anlarlar, hatalar azalır ancak sürecin görünürlüğü azdır ve bakımı zordur.



**Artımsal Geliştirme Süreci Modeli :** Sistemi tek seferde teslim etmek yerine, geliştirme ve teslim parçalarına bölünüp kullanıcının önceliğine göre bu parçalar sıralanır. Kullanıcının gereksinimleri önceliklendirilir. Üretilen her yazılım sürümü birbirini kapsayacak ve giderek artan sayıda işlev içerecek şekilde geliştirilir. Bir taraftan kullanım, diğer taraftan üretim vardır. Olası değişiklikler sonraki teslimlerde ele alınır. Bu model ile sistemin başarısız olma olasılığı azalır, ara ürünler yazılımın geliştirilmesinde önemli bir yere sahip olur. Uzun zaman alabilecek ve sistemin eksik işlevlikle çalışabileceği türdeki projeler bu modele uygun olabilir.



**Çağlayan Model :** Geleneksel yazılım geliştirme modeli olarak bilinir, aşamaların en az 1'er kez tekrarlanmasıyla geliştirilir. Çok iyi tanımlanmış ve üretimi az zaman gerektiren projeler için uygun bir model olsa da günümüzde kullanımı gittikçe azalmaktadır. İşler aşama aşama yapılır biri bitmeden diğerine geçilmez. Her safhada dökümantasyon yazılmalıdır. Her şeyin dökümanınin olması gerekir. Bir safhada dökümantasyon veya test olmamışsa o safhanın tamamlandığı kabul edilemez.

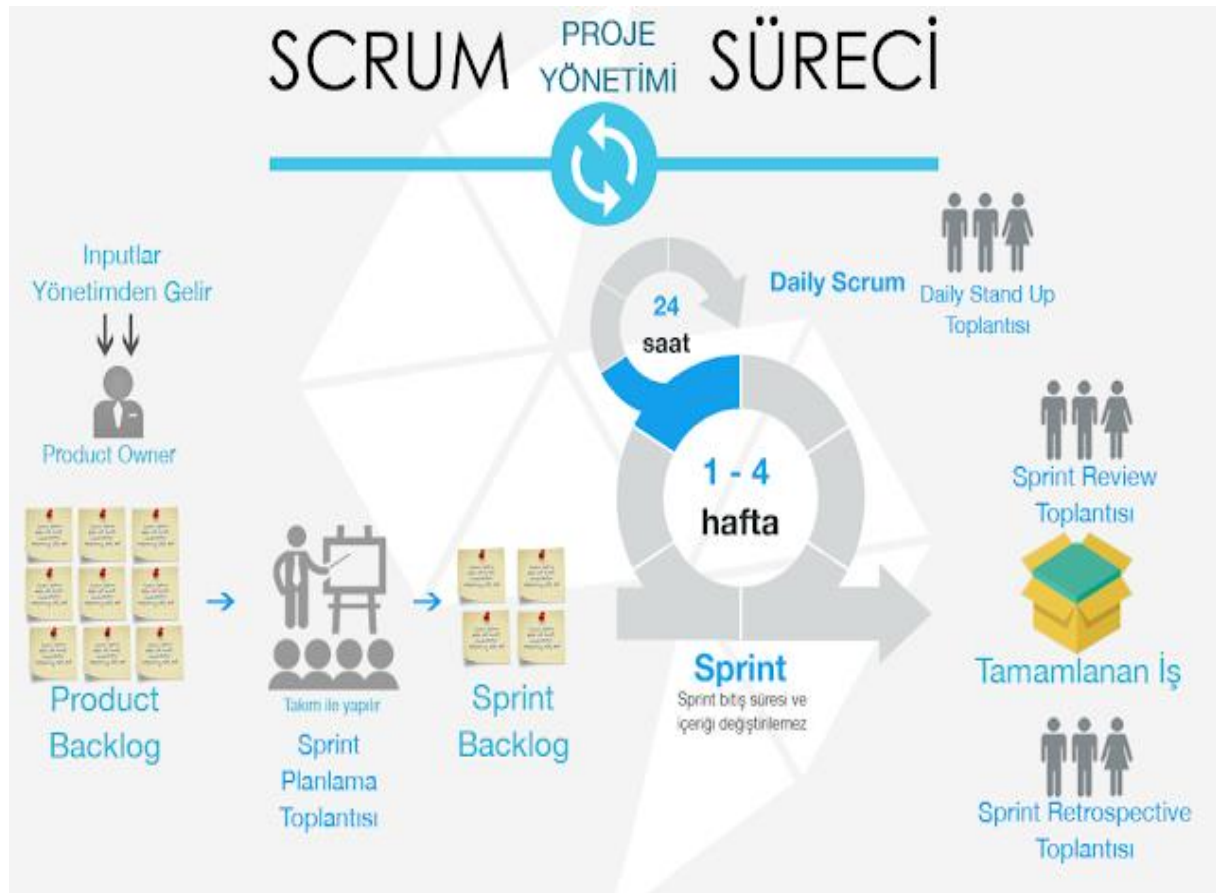


**SCRUM** : Scrum ilk olarak, Ken Schwaber ve Jeff Sutherland tarafından 1990'ların başında geliştirilmiştir. Daha sonra yüksek performanslı takımlar oluşturulurken Scrum topluluğu dünya çapında büyümeye başlar. Topluluğun büyümesiyle Scrum Alliance kurulur ve dünyada Scrum uygulayanlar ise bu organizasyon ile bir araya gelirler. Scrum modeli yazılım süreçlerinin yönetimi için kullanılmaktadır, yani sadece yazılım geliştirmeye değil her projeye uygulanabilir. Karmaşık yazılım işlerini küçük birimlere bölerek geliştirmeyi sağlar. Ayrıca düzenli geri bildirim ve planlamalar ile hedefe ulaşmayı sağlar. Bu açıdan ihtiyaca yönelik ve esnek bir yapıya sahiptir. Müşteri ihtiyacına yönelik olduğu için geri bildirimlere göre yapılanmayı sağlar. Scrum modeli sürecinde iletişim ve takım çalışması oldukça önemlidir. Günlük kısa toplantılarla sürekli iş takibi yapılır. SCRUM 'da müşteri tarafından istenilen ve tanımlanan işlevler 2-4 haftalık "Sprint" adı verilen dönemler içerisinde geliştirilip gözden geçirilir. Her sprint sonunda yazılımın fonksiyonel bir parçası bitmiş ve müşteriye teslim durumunda olur. Bu metod 3 temel prensip üzerine kurulmuştur;

**Şeffaflık:** Projenin ilerlemesi ve problemler, gelişmeler herkes tarafından görülebilir olmalıdır.

**Gözlemleme:** Projenin ilerlemesi düzenli olarak kontrol edilmelidir.

**Uyarılama:** Proje, uygulanacak değişiklikler ile uyumlu olmalıdır.



SCRUM günümüzde en çok kullanılan yazılım geliştirme yöntemidir. Hatta sadece yazılım geliştirmede değil birçok sistemin geliştirilmesinde de kullanılır.

## Modellerin Karşılaştırılması

1. Scrum karmaşık görülen ve gereksinimleri tam belirlenmemiş projeler için idealken, çağlayan model çok iyi tanımlanmış ve üretimi en az zaman gerektiren projeler için uygundur.
2. Çağlayan modelde test aşaması geliştirme sürecinin en sonunda yapılır hatalar önemli yeniden tasarım gerekliliğini oluşturur. Artımsal Geliştirme modelinde her teslimde müşterinin geri dönüşü alındığı için işlevsellik erken ortaya çıkar, proje daha güvenli bir hal alır.
3. Evrimsel modelde sistemler sıklıkla iyi yapılandırılmaz sürekli değişiklik yazılımın yapısına zarar verir. Artımsal Geliştirme Modelinde böl ve yönet yaklaşımı uygulanarak en önemli özellikleri daha fazla sınaıma imkanı bulunur.
4. Evrimsel modelde kullanıcılar gereksinimleri daha iyi anlarlar, hatalar azalır ancak sürecin görünürlüğü azdır ve bakımı zordur.
5. Çağlayan modelinin kullanılması ve uygulanması basit, iş bölümü ve iş planlaması detaylıdır. Artımsal modelde deneyimli personel gerekir.

## Hangi Projede Hangi Modeli Kullanmalıyız?

- Büyük kitlelere ulaşacak projelerde evrimsel geliştirme modeli kullanılabilir.
- Artımsal model büyük, maliyetli ve uzun süren projelere uygundur.
- Küçük ve özellikleri iyi tanımlanmış projelerde çağlayan modeli kullanılabilir.
- Büyük ya da birbirine bağımlı projelerde, yeni ya da var olan bir projelerde scrum kullanılabilir.

## SCRUM günümüzde neden popüler?

Ekip çalışmasını ön planda tuttuğu için ekip içi iletişimi geliştirerek sorunların kısa sürede çözümlenmesine olanak sağlıyor. Bu gibi etmenler günümüzde projelerinin başarı oranının %80 olmasını sağlamıştır. Scrum yaklaşımı en popüler yöntemidir. Büyük ve karışık yazılım süreçlerinin yönetilmesinde tercih edilen Scrum bütünü parçalamaya ve tekrara dayalı bir yöntemdir. Projenin açık ve net olması hem zaman kazandırır hem de projenin başarılı sonuçlanmasını sağlar. Projenin anlaşılabilir ve yönetilebilir parçalara ayrılması olası sorunları hızlı bir şekilde tespit etmekte ve düzeltmekte zaman kazandırır. Bütün ekibin, projenin tüm akışından haberdar olması takım içindeki iletişimi artırır. Müşteri ile geliştiriciler arasında güven oluşur ve böylelikle projenin başarılı sonuçlanması beklenir.

Mehmet Salih Akçiçek

220601083