Bu yazımızda; yazılım geliştirme yaşam döngüsünü, bu kavrama bağlı olan yazılım geliştirme modellerini ve Scrum’ ı derinlemesine inceleyeceğiz.

**Yazılım Yaşam Döngüsü Nedir ?**

Yazılım belirli bir yaşam döngüsü olan bir üründür ve bu ürünün oluşturulması-geliştirilmesi sürecinde birçok aşama vardır. Yani bu ürünün geliştirilmesinden teslim edilme sürecine ve hatta teslim edildikten sonra bile devam eden aşamalar topluluğuna ve bu aşamalardan oluşan döngüye, yazılım yaşam döngüsü diyoruz. Öncelikle temel adımları açıklayalım.

1. Gereksinim

Müşteriden gereksinimlerinin elde edildiği ve fizibilite çalışmalarının yapıldığı aşamadır

1. Analiz

Bu aşamada sistem gereksinimleri ve işlevleri ayrıntılı olarak incelenenir. Projenin yapım sürecinin ne kadar uzun olabileceği tahmin edilir ve proje sürecinin ne gibi risklere sahip olacağı belirlenir.

1. Tasarım

İlk aşamamızda belirlemiş olduğumuz gereksinimlere uygun yazılım sisteminin temel yapısının oluşturulduğu safhadır.

1. Gerçekleştirme

Yazılım projesinin kodlanıp birleştirilip ve dahilinde test edildiği kısmıdır.

1. Bakım

Sunduğumuz yazılım ürününün sunulduktan sonraki güncelleme, bakım, eksik giderme olaylarının yapılmakta olduğu aşamadır.

Yazılım sektöründe yaygın olarak kullanılan ve genel olarak bilinen yazılım başlıca yazılım-yaşam döngü modelleri:

* Çağlayan Yaşam-Döngü Modeli,
* Gelişi-Güzel Model
* Barok Modeli
* Kodla ve Düzelt Yaşam-Döngü Modeli,
* V Süreç Modeli,
* Helezonik (Spiral) Model,
* Artımsal Geliştirme Yaşam-Döngü Modeli.

Yazılım geliştirme; bir işe sıfırdan başlayarak gereksinimler, analiz, tasarım ve gerçekleştirme safhalarını doğrusal bir biçimde ilerleyerek gerçekleştirmektir. Elbette pratikte göründüğü kadar basit değildir ve oldukça farklıdır. Bu farklılığın sebebi ise yazılım geliştiriclerinin insan olup hata yapma ihtimallerinin bulunması ve yazılım geliştirilme aşamasında müşterinin gereksinimlerinin değişme ihtimali de vardır. Bu iki ihtimalin gerçekleştiğini varsaydığımızda yazılım geliştirme adımlarının doğrusallıktan çıktığını ve bazı durumlarda da tekrar ettiğini fark edebiliriz.

**1-) Gelişi-Güzel Model**

1960'lı yıllarda uygulanmakta olan bu yöntem, genellikle basit programlama içermekte olan ve çoğunlukla da tek bir yazılımcının ürün üretimini yaptmakta olduğu yöntemdir. Aslında bu yöntemi bir model olarak adlandırmamamız gerekmetedir çünkü bu geliştirme yönteminde herhangi bir belirlenmiş bir model ya da yöntem bulunmamaktadır. Bu yöntem kişiye bağlı yazılım geliştirme şeklinde yapılmakta olduğu için geliştirme sürecinde yazılımın izlenebilirliği ve bakım yapılabilirliği olduça zor olmaktadır.

**2-) Barok Model**

70'li yıllarda ortaya çıkmış olan barok modeli belgeleme konusunda dönemin yazılım geliştiricilerinin dikkatini üstüne çekmiştir çünkü Bu modelde, dokümantasyon günümüz modellerinden farklı olarak ayrı bir süreç olarak ele alınmakta ve yazılımın geliştirme ile test faaliyetleri tamamlandıktan sonra dökümantasyonun yapılması öngörülür.

Model yazılım yaşam-döngüsünün temel adımları doğrusal bir şekilde ele alınıp geliştirilir. Aşamalar arasında da gereken geri dönüşlerin nasıl yapılacağı tanımlı değildir.

\*\* Günümüzdeki yazılım geliştirme projelerinde uygulanan bir model olmaktan çıkmıştır.

**3-) Çağlayan Yaşam-Döngü Modeli**

Çağlayan yaşam modeli geçmişteki en popüler yazılım geliştirme modellerinden biridir.Çokça tanınmış olup en temel ve en eski modeldir.Bu model tıpı moderm yazılım dillerindeki C dilinin bir temel niteliğinde olması gibi yaşam döngü modellerinde de “Çağlayan Yaşam-Döngü Modeli” bir temel niteliği oluşturmuştur. Bu model yazılımcılar arasında geleneksel yazılım geliştirme modeli oalrak ta bilinebilmektedir. Bu modelde yazılım, modelin içerdiği aşamaların en az birer kez tekrarlanması ile geliştirilir. Üretimi fazla zaman gerektirmeyen projeler için oldukça uygun bir model olmakla birlikte günümüzdeki kullanımı gittikçe azalmaktadır. Çağlayan yaşam-döngü modelinde:

* İşler sırasına göre aşama aşama yapılır.
* Bir aşama bitmeden asla diğer aşamaya geçilemez.
* Her aşama esnasında dokümantasyon yazılmalıdır.
* Süreçte bulunan her şeyin dokümanının bulunması gereklidir.
* Eğer herhangi bir safhada dokümantasyon ve dahilinde test olmamışsa, o aşamanın tamamlandığı kabul edilmez.

\*\* Çağlayan yaşam-döngü modeli, Barok modelinden farklı olarak proje içerisindeki dokümantasyonu ayrı bir süreç olarak değil üretimin doğal bir parçası olarak ele alır.

\*\* Çağlayan yaşam-döngü modelinde aşamalar arasındaki geri dönüşlerin nasıl olacağı da tanımlıdır. Barok modelinde tanımlı değildi.

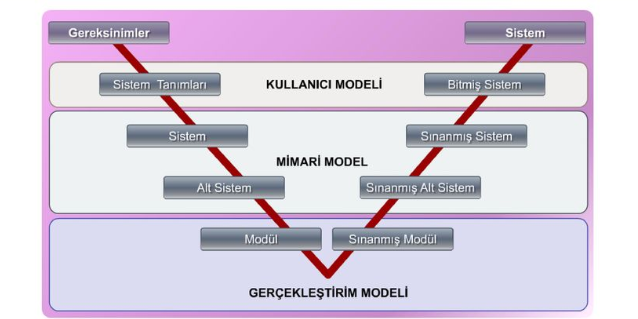
\*\* Model içerisindeki aşamalar arasında geri dönüşler yapılabilse de analiz aşamasına gelindiğinde mümkün olan tüm detayların projenin tasarımına yansıtılabilinmesi için müşteri ve sistem gereksinimlerinin en ince ayrıntısı dahi belirlenmelidir.

Modelde Tasarım aşaması yazılım ürününün tüm gereksinimlerini kapsayacak bir biçimde detaylı bir çalışma gerektirmektedir. Bu nedenle Çağlayan yaşam-döngü modelini kullanmakta olan proje ekipleri zamanlarının çoğunluğunu bu iki aşamada harcamak zorunda kalırlar. Detaylı çalışmalara ve harcanan emeğe rağmen uzun zamana yayılmakta olan projelerde gereksinimlerin değişmesi neredeyse kaçılmazdır. Bu değişikliklerin kodlama veya test aşamasında ürüne yansıması maliyeti oldukça yükseltir.

Bu modelde genellikle yazılımın müşteriye ulaşma süreci oldukça uzundur. Eğer gereksinim tanımlamaları net bir şekilde yapılamaz ise ilerleyen süreceçlerde yazılımdaki yanlış düzeltme ve eksiklerin giderilme maliyetleri oldukça yüksek olur.

**4-) “V” Süreci Modeli**

V- modeli şelale modelinin gelişmiş hali olarak düşünülebilir. Bu model belirsizliklerin az, iş tanımlarının ise belirgin olduğu projelerde kullanılır. Bu model kullanıcının projeye katkısını arttırmaktadır

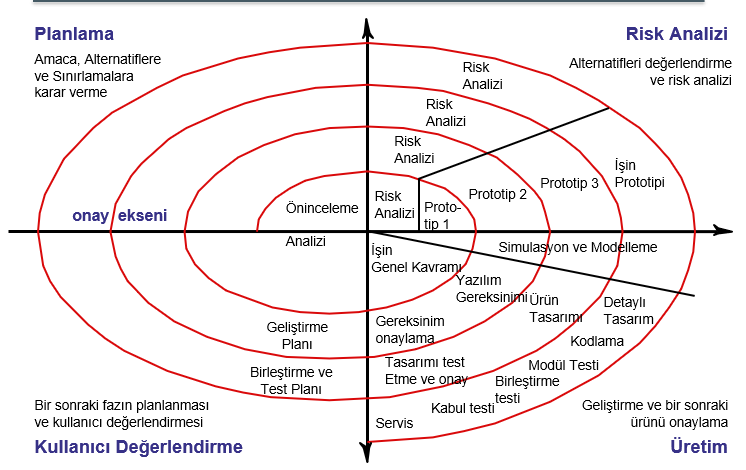


Modelde sol taraf üretim, sağ taraf ise sınamadır. Bu döngü esas olarak 3 modelden oluşmaktadır.

V süreç modelinin temelleri;

* Kullanıcı Modeli: Geliştirme sürecinin kullanıcı ile olan ilişkileri tanımlanmakta ve sistemin nasıl kabul edileceğine ilişkin sınama belirtimleri ve planları ortaya çıkarılmaktadır.
* Mimari Model: Sistem tasarımı ve oluşacak alt sistem ile tüm sistemin sınama işlemlerine ilişkin işlevler.
* Gerçekleştirim modeli: Yazılım modüllerinin kodlanması ve sınanmasına ilişkin fonksiyonlar.

\*\* V- modelinde aşamalarda tekrar bulunmaz ve risk çözümleme için ayrılan bir yer yoktur

**5-) Helezonik Model**

Helezonik modeli diğer modellerden ayırmakta olan özellik; risk analizinin ön planda olması ve prototip oluşturulmasıdır. Risk analizi ön planda olması ortaya çıkan hatalar için erken giderilebilme imkânı sağlayabilir. Prototip oluşturma ise her aşamada bulunduğundan dolayı kullanıcı her aşamada yazılım ürününün bir parçasını görme imkanına sahip olur ve bu sayede sorunların azalması sağlanılabilir.

Helezonik model 4 temel aşamadan oluşur:

* Planlama: Üretilecek olan ara ürün için planlama yapılır, amaç belirlenir ve bir önceki adımda üretilmiş olan ara ürün ile bütünleştirme.
* Risk Analizi: Risk seçeneklerinin araştırılması ve risklerin belirlenmesi.
* Üretim: Ara ürnün üretilmesi.
* Kullanıcı Değerlendirmesi: Ara ürün ile ilgili olarak kullanıcı tarafından yapılan sınama ve yorumlamalar.

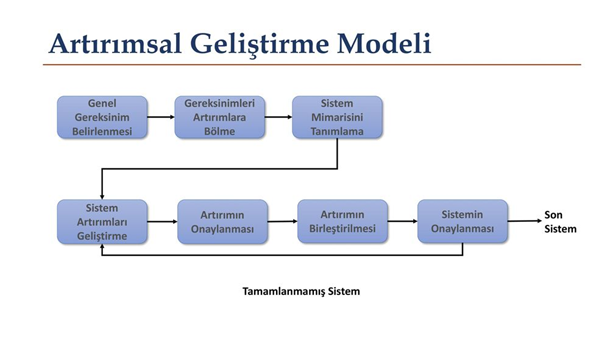
Helezonik modelde risk analizi ön plandadır. Modeldeki her döngü bir faz idafe etmektedir. Doğrudan analiz,tasarım gibi bir fazı bulunmamaktadır. Prototip ve yinelemeli artımsal bir yaklaşım bulunmaktadır.

Helezonik yaşam-döngü modelinin avantajları:

Üretim süresince ara ürünü üretme ve ara ürünün kullanıcı tarafından denenmesi/yorumlanması temelindedir.

Yazılım kullanıcılarının bu sürece erken katılması ilerleyen süreçte oluşabilecek istenmeyen hataları/durumları engeller. Yazılım ürününün kodlanma ve sınanma işlemleri daha erken başlar.

**6-) Artımsal Geliştirme Süreç Modeli**



Artımsal geliştirme yaşam-döngü modelindeki temel süreç akışı yukarıda verilen şekildeki gibidir.

Artımlı Geliştirme Modelinde proje parçalara bölünür ve kullanıcının önceliğine göre bu parçalar sıralanır. Sıralanan bu parçaların üretimi bittiğinde sırasıyla birer ara ürün geliştirilir ve bu ara ürünler kullanıcı tarafından kullanılmaya başlanır. Ara ürünler her defasında kullanıcı deneyimleri ve eksikler/hatalar dikkate alınarak bir önceki sürüme göre giderek artan sayıda işlev içerecek şekilde eklenti yapılarak çıkartılır. Yani bu modelde bir taraftan üretim yapılırken diğer taraftan da ürün kullanıcı tarafından kullanılır. Yazılım geliştirme iteratif bir süreçtir. Her süreçte sonuca biraz daha yaklaşılmaktadır.

Artımsal Geliştirme Süreç Modelindeki avajlarımız:

\*\* Her teslimle birlikte müşteriye görünen bir değer döndüğünden, sistemin işlevselliği erken aşamalarda ortaya çıkar.

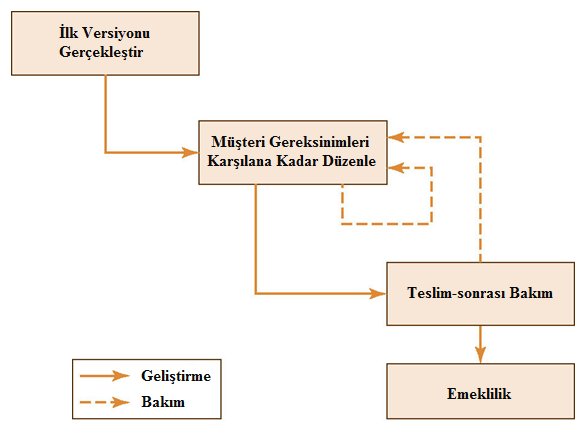
\*\* Erken teslimler, sonraki teslimler için gereksinimleri çıkarmada prototip vazifesi görür.

\*\* Projenin tümden batması riskini azaltır.

\*\* Öncelikli gereksinimleri karşılayan sistem işlevleri daha çok test edilir.

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

Birkaç yüz satırdan oluşan küçük programlar için kullanılabilir, direkt ürün gerçekleştirilir ve emeklilik safhası vardır. Direkt olarak yazılım ürünü oluşturulup sistem, en son istenilen şekle getirilinceye kadar devamlı olarak geliştirilir. Modelde bakım safhası bulunmakta fakat sisteme ait bir dökümantasyon olmadığından dolayı çok zordur ve ürünün emeklilik safhası da bulunmaktadır.



Yazılım geliştirmenin en kolay yoludur fakat kolay olduğu kadar da en pahhalı yoludur.

Kodlamadan sonra bir yazılım ürünündeki değişikliklerin maliyetini düşünürsek, bu maliyet çok yüksektir.

Ayrıca döküman bulunmadağından dolayı ürünün bakımını yapmak oldukça zordur.

Yazılım geliştirmek kolay olduğundan dolayı ne yazık ki tecrübesiz yazılımcılar projesinde bu modeli kullanır.

**Extreme Programming (**XP**)**

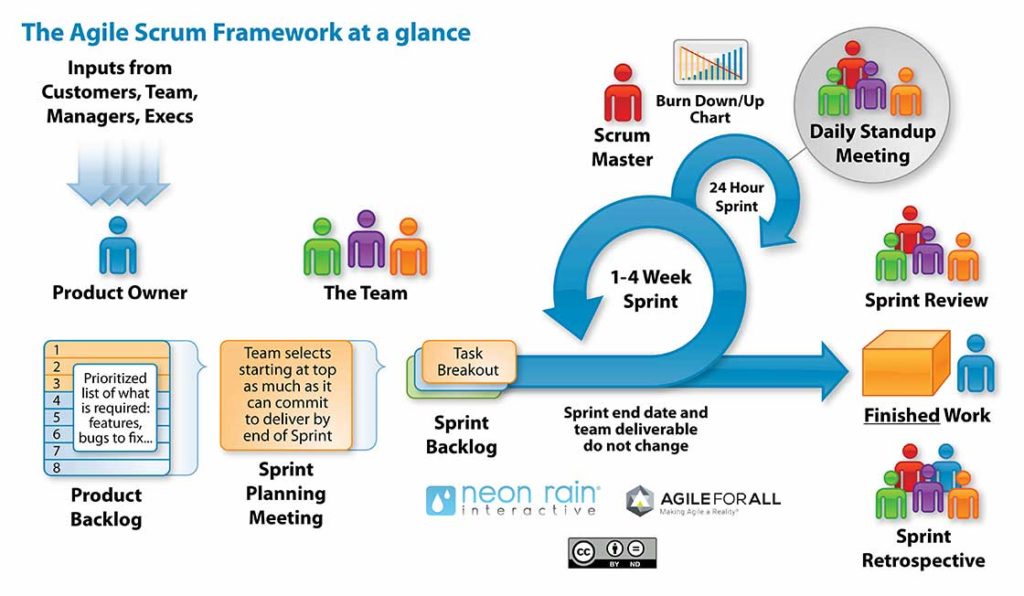
Kent Beck ve arkadaşları tarafından 1996 yılında kurulmuştur. XP kolay, grup içi iletişime önem veren, geri dönüşlerin fazla olmasına imkân sağlayan bir yazılım geliştirme yöntemidir. 4 temel maddeden oluşur: Basitlik, Cesaret, Geri Dönüş, İletişim.

* Basitlik: yazılmış olan kodun ve yapılan ürünün sade, anlaşılır ve basit bir yapıda yapılmasını gerektirir. Dokümantasyonun uzun olmasından kaçınılır.
* Cesaret: Yapılan işte cesur olunmalı, projelerde korkmadan ilerlenmeli, gerekirse bir kodu tamamen silip yeniden yazma girişiminde bulunabilinmelidir.
* Geri Dönüş: Geri dönüşler ile oluşabilecek hatalar azaltılır. Müşteri ile yazılım ekibi her daim birbirleriyle iletişim halindedir.
* İletişim: İletişim, projelerdeki en önemli sorunlardan birisidir. XP ise bu sorunu aşmaya çalışmaktadır. Ekip içi iletişime önem verir ve artırılması için çalışır.

**SCRUM**

SCRUM, Jeff Sutjerland ve Ken Schawaber tarafından 1990’ların ortalarında geliştirilen, çevik yazılım geliştirme metodolojileriyle uygulanabilecek bir proje yönetim yaklaşımıdır. Yani SCRUM sadece yazılım geliştirmede değil, her alanda da uygulanabilir. SCRUM da büyük projeler parçalara bölünerek ve her bir parçaya da “sprint” adı verilerek sonrında her bir sprint in taker taker geliştirildiği bir proje yönetimidir.

Gereksinimlerin kolaylıkla tanımlanamadığı ve kaotik durumların beklendiği projeler için en uygun metodolojidir SCRUM.

Bu metodolojide bir yinelemenin tamamlanması 30 günden fazla sürmemekte ve günlük 15 dakikalık toplantılarla sürekli iş takibi yapılmaktadır. Bu da hep geliştiriciler arası iletişimi sağlamlaştırıp sıklaştırmakta ve projenin her bir adamından her gelişticinin haberdar olmasını sağlayarak iş takibi yapılmasını da sağlar.  - Roller: Scrum yöneticisi, scrum takımını ve ürün sahibinden oluşmaktadır. Scrum yöneticisi,scrum takımını belirlenmiş olan sınırlar çerçevesince yönetir düzeni sağlar. Scrum takımının boyutu genelde 5–9 kişi aralığında olmaktadır ve bu takım içerisi iletişim oldukça sıkıdır.

- Toplantılar: SCRUM’ın neredeyse en önemli olan kısımı toplantılardır. Her gün geliştiriciler ve yöneticiler arasında toplantı yapılır ve bu toplantılarda geliştiricilerin önceki günlerde neler yaptıkları, geliştirme süreci esnasında karşılaştıkları sorunlar ve bugün yazılım üzerinde neler yapmaları gerektiği hakkında bir planlama yaparlar. Sprintlere bakılarak ne kadar ilerleme kaydedildiğine de baklır.

- Bileşenler: Ürünün Gereksinimler dokümanı oluşturulur. Dökümana proje süresince neler yapılması gerektiği basit bir şekilde kaydedilir. Sprint için de döküman oluşturulur. Sprint dokümanı ve ürün dokümanının ayrı ayrı oluşturulmasının amacı her sprintin ürün gereknsinimler dökümanına uygun ayarlanmasıdır . Sprint kalan zaman grafiği bulunmaktadır ve bu grafik yapılmış olan işlemlerin ne seviyede bulunduğunu ve planlanan zamanın neresinde olunduğunu belirlemek için hazırlanır.

SCRUM un günümüzde oldukça popüler ve en çok kullanılan yazılım geliştirme yöntemidir. Scrumun oldukça popüler olasını sağlayan yanları:

* Proje süresince geçen zamandan ve paradan büyük ölçekte tasarruf edilmesi,
* Her teknolojiye uyum sağlayabilip. Teknolojideki son gelişmelere uyum sağlayabilmesi,
* Karmaşık olan ve gereksinimleri tam belirlenememiş projeler için ideal bir yapı olması,
* Ekip içerisindeki iletişim düzeyinin yüksek olmasını sağlaması ve bu sayede hataların erken fark edilip düzeltilmesi,
* Kullanıcıdan sürekli olarak geri bildirim gerektirmesi ve bu geri bildirim ile beraber sorunların çözülüp azaltılabilbesini sağlaması,
* Diğer yazılım geliştirme metodolojileri gibi yinelemeli ve sürekli aktif olması,
* Değişen gereksinimlere hızlı bir şekilde tepki verilmesine ortam hazırlamış olması,
* Büyük parçaları küçülterek hızlıca ilerlemeyi sağlayan bir yapıya sahip olması.

**Kaynaklar**

* Doç. Dr. Deniz Kılınç, Bakırçay Üniversitesi Yazılım Mühendisliğine Giriş Dersi hafta 2-3
* Doç.Dr.Resul Daş, Yazılım Yaşam Döngüsü ve Süreç Modelleri bölüm 2
* <https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742>
* [*http://ybsansiklopedi.com/wp-content/uploads/2015/08/Yaz%C4%B1l%C4%B1m-Geli%C5%9Ftirme-Modelleri-Yaz%C4%B1l%C4%B1m-Ya%C5%9Fam-D%C3%B6ng%C3%BCs%C3%BCSDLCYBS.pdf*](http://ybsansiklopedi.com/wp-content/uploads/2015/08/Yaz%C4%B1l%C4%B1m-Geli%C5%9Ftirme-Modelleri-Yaz%C4%B1l%C4%B1m-Ya%C5%9Fam-D%C3%B6ng%C3%BCs%C3%BCSDLCYBS.pdf)
* [*https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/*](https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/)