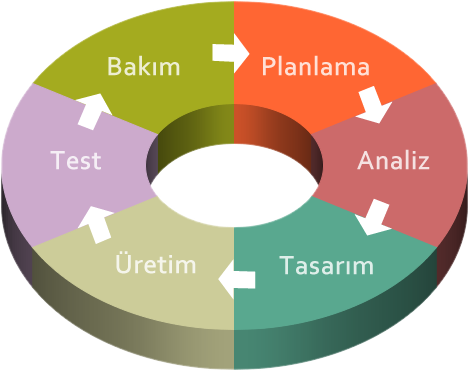
**Yazılım Yaşam Döngüsü Nedir?**

Yazılım yaşam döngüsü, geliştirdiğimiz bir yazılımın planlanmasından teslim edilmesine kadar uygulanan aşamalardan oluşan döngüye denir.Yani bir yazılım ürünün başlangıcından sonuna kadar geçen tüm sürece yazılım yaşam döngüsü diyebiliriz.Belirli bir yazılımın nasıl geliştirileceğini, değiştirileceğini, sürdürüleceğini veya iyileştirileceğini açıklayan bir süreçtir.SDLC, Yazılım yaşam döngüsünün kısaltmasıdır.

**Yazılım Yaşam Döngüsü Aşamaları**

1. PLANLAMA
2. ANALİZ
3. TASARIM
4. GELİŞTİRME
5. TEST
6. BAKIM



**1.Planlama Aşaması**

Döngünün ilk aşaması planlama aşamasıdır.Bu aşamada yapılacak proje için temel ihtiyaçların(zaman,maliyet,donanım vb.) belirlenir,gerekli analizler yapılır ve proje planı yapılır.Ardından diğer adımlar için nasıl bir yol geliştirileceği kararlaştırılır.

**2.Analiz Aşaması**

Bu aşama yazılım yaşam döngüsünün en önemli aşamalarından biridir.Bu aşamada projenin tüm fonksiyonları detaylı olarak çıkarılır,temel sorunlar ortaya çıkarılır ve projenin gereksinimlerinin kesin olarak müşteriler tarafından belirlenir.

**3.Tasarım Aşaması**

Bu aşama,analiz aşamasından sonra belirlenmiş olan gereksinimlere göre sistemin tasarımına başlanmasıdır.Bu yazılımın tasarımı müşterinin isteklerine ve ihtiyaçların hitap etmelidir.Bu aşamada mimari tasarım ve detaylı tasarım olmak üzere iki çeşit tasarım yapılır.Mimari tasarım yazılımın genel yapısını oluştururken detaylı tasarım yazılımı içeren bileşenleri ve ayrıntılarını içerir.

**4.Geliştirme Aşaması**

Bu aşama kodlamanın başladığı aşamadır.Kodlama seçilen programlama diline bağlı kalınarak yazılmaya başlanır.Bu kodlar SOLID prensibine göre oluşturulur.

**5.Test Aşaması**

Yazılımın geliştirilmesi bittikten sonra müşteriye teslim etmeden önce test ekibi tarafından projede sorunlar varsa giderilir,kodlamalarda ve tasarımlarda değişiklik yapılması gerekiyorsa yapılır.Bu gibi kontroller yapıldıktan sonra son hali teslim edilir.

**6.Bakım Aşaması**

Bu aşamada proje teslim edildikten sonra meydana gelebilecek hatalar giderilir,yazılım iyileştirilir,sisteme yeni eklentiler eklenir,projeyi kullanan kullanıcılar tarafında gelen istekler gerçekleştirilir.

**Yazılım Yaşam Döngüsü Modelleri**

Bazı yazılımlar dışardan bakıldığında aynı süreçten geçiyor gibi gözükebilir fakat çokta benzer durumlar yaşadığını söyleyemeyiz. Her yazılımın birçok nedenden dolayı kendine özgü bir süreci,döngüsü vardır bu sebeple yazılım yaşam döngüsünde birden fazla model ele alınır.Bu modeller sürecin nasıl uygulanacağını,düzenini ifade eder.

* Gelişigüzel Model
* Barok Modeli
* Şelale Modeli (Waterfall Model)
* V Modeli (V-Shaped Model)
* Evrimsel Geliştirme (Evolutionary Development)
* Prototipleme (Prototyping)
* Kodla ve Düzelt (Code and Fix)
* Artımlı Geliştirme (Incremental Development)
* Helezonik **(**Spiral) Model
* Çevik Model (Agile Model)

**Gelişigüzel Model**

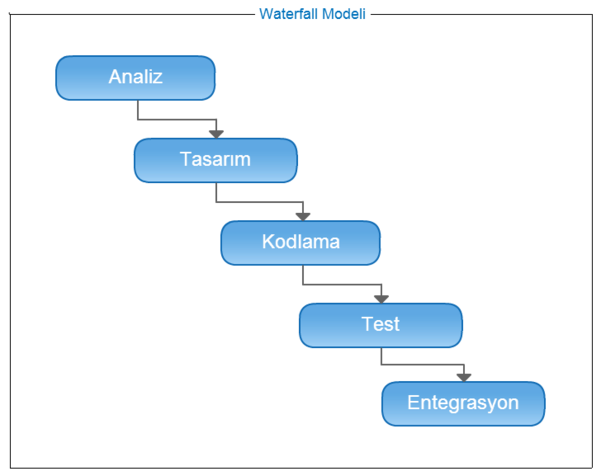
Bu model isminden de anlaşılacağı üzere herhangi bir yöntem gözetilmeksizin gelişigüzel geliştirilir.Geliştiren ya da ilgi duyan kişiye bağlıdır.Bu modelin takip edilebilirliği ve bakımı zordur.Bunu geliştiren kişi bile aradan belli bir zaman geçtikten sonra modeli takip edemeyebilir.Model basit bir programlamaya sahiptir.

**Barok Modeli**

Modelde yazılım yaşam döngüsü adımları doğrusal bir biçimde geliştirilir.Barok modelinde aşamalar arasında geri dönüşlerin nasıl yapılacağı belirli olmadığı için günümüzde bu model pek tercih edilmez.Modelin en büyük problemlerinden biri belgelendirmenin proje sonunda yapılmasıdır.Çünkü günümüzde belgelendirme işlemi döngünün başından sonuna kadar sürekli yapılmaktadır.Bu gibi sebepler hata yapma olasılığımızı arttırır bu da zaman,maliyet açısından projeyi olumsuz etkiler.

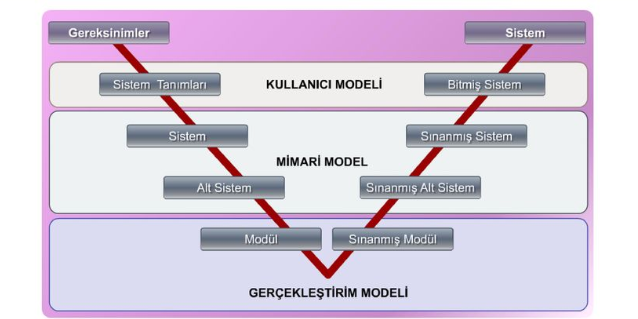
**Şelale Modeli(Waterfall Model)**

Şelale modeli doğrusal,sıralı yaşam döngüsü modeli olarak da adlandırılır.Şelale modeli yazılım geliştirme için kullanılan en eski SDLC yaklaşımıdır.Bu sebeple “Geleneksel Yazılım Geliştirme Modeli” olarak da bilinmektedir.Bu modelin anlaması ve kullanılması çok basittir.Bu modelde yazılım geliştirme süreci analiz,tasarım,kodlama,test,sürüm ve bakım gibi aşamalardan oluşur.Şelale modelinde bir sonraki aşamanın başlayabilmesi için bir önceki aşamanın tamamlanmış olması gerekir.Aşamalar arası geri dönüş pek tavsiye edilmez.Çünkü model değişikliğe açık değildir.Ayrıca başlangıçta yapılan hataların tespiti uzun sürebilir ve daha sonra bu hataların giderilmesi yüksek maliyete sebep olur.Her aşamanın sonunda bir doküman oluşturulduğundan şelale modeli doküman güdümlüdür.Şelale modeli,analiz ve tasarımın çok önemli olduğu,hata yapmanın maliyetinin yüksek olduğu kritik projelerde kullanılır.



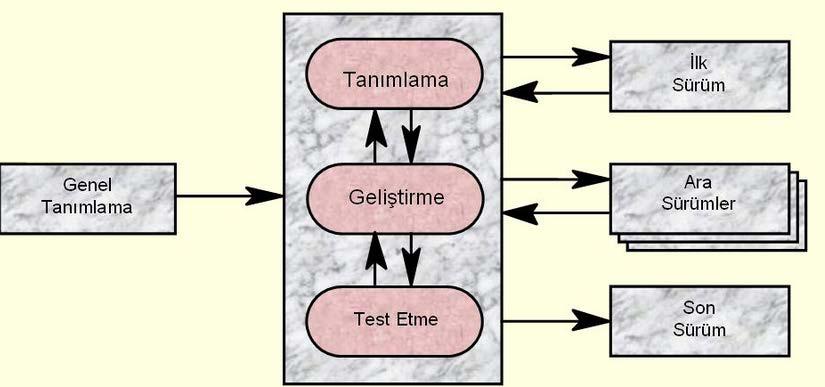
**V Modeli(V-Shaped Model)**

V modeli,yazılım sürecinin V şeklinde sıralı bir şekilde yürütüldüğü SDLC modelidir.Doğrulama ve Doğrulama modeli olarak da bilinir.SDLC’nin her aşaması bir sonraki aşama başlamadan hemen önce tamamlanmalıdır.Şelale modeli ile aynı sıralı süreci takip eder.V modeli şelale modelinin daha gelişmiş bir halidir.Bu modelin anlaşılması ve uygulanması gayet kolaydır ve modelin sadeliği yönetimini de kolaylaştırır.Fakat bu model değişikliklere karşı esnek değildir.Bu yüzden değişiklik yapılması çok pahalı hale gelir.Bu modelde yaşam döngüsü süresince geç saatlere kadar çalışan hiçbir yazılım üretilemez.Model daha öncede söylediğimiz gibi V şeklinde sıralı bir şekilde yürütülür. V yapısını simetrik olarak ortadan ayırdığımızda sol bacağı yazılım geliştirme işlemlerini sağ bacağı ise test işlemlerini temsil etmektedir.V modeli gereksinimleri açık ve net olan projelerde kullanılabilir.



**Evrimsel Geliştirme (Evolutionary Development)**

Bu model ilk tam ölçekli modeldir.Her aşamada üretilen ürünler,üretildikleri alanlar için tam işlevselliği içerir.Model daha çok birimli organizasyonlar için öneriliyor.Bu modelin izlenebilirliği ve bakımı zordur.Ayrıca sürekli değerlendirmeler yapıldığı için risk ve hataları azaltır.

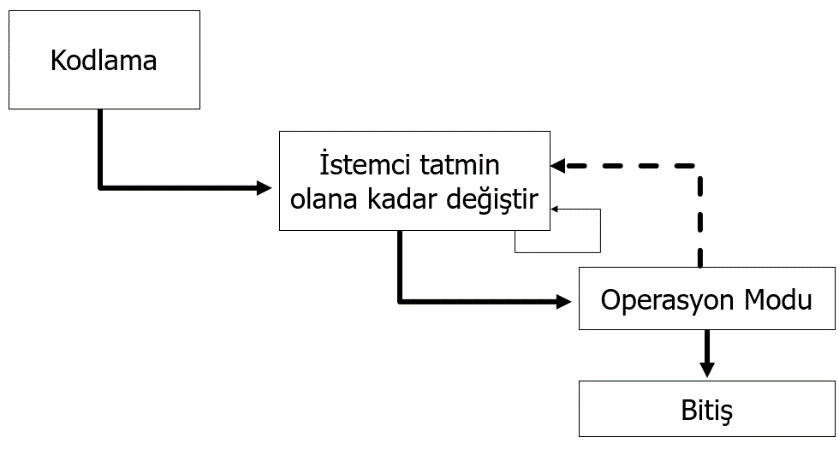


**Prototipleme (Prototyping)**

Prototipleme modeli en popüler kullanılan yazılım yaşam döngüsü modellerinden biridir.Bu modelde adından da anlaşılacağı üzere bir prototip oluşturulur ve test edilir.Ardından tüm sistemin veya ürünün geliştirilebileceği kabul edilebilir prototip elde edilene kadar,müşteri geri dönüşlerine göre tekrar tekrar test edilir,geliştirilir.Prototipleme modelinde sistem kısmen analiz aşamasında veya öncesinde uygulanır.Böylelikle müşteriler ürünü başlarda görür ve süreç müşterilerle görüşerek başlar.Model tüm proje gereksinimlerinin önceden ayrıntılı olarak bilinmediği senaryolarda en iyi sonucu verir.Müşteri prototipi kabul edene kadar bu süreç devam eder.Bu da modelin en büyük avantajlarından biridir.Böylelikle karmaşa ve yanlış anlamalarda engellenmiş olur ve risk kontrolü sağlanmış olur.Fakat düzeltme aşaması atlanırsa projede düşük performansa yol açar.

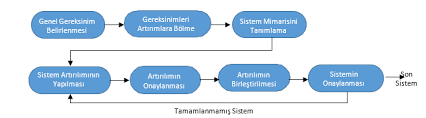
**Kodla ve Düzelt(Code and Fix)**

Bu model resmiyette bir fikir yokken başlatılır ve proje hazır hale gelene kadar süreç kodlama yaparak geçer ve müşterinin istediği olana kadar güncellenir.Bu model genellikle planlanmamış küçük,kişiselprojelerde kullanılır.Model de bir planlamaya ihtiyaç duyulmadığı gibi belirli bir kaynak planlaması da yoktur ve kontrollü değildir.Bu süreçte hataların bulunması ve doğrulanması da zordur.



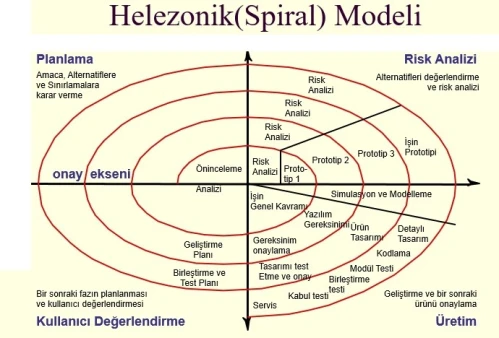
**Artımlı Geliştirme (Incremental Development)**

Bu modelde proje kısım kısım geliştirilip teslim edilir.Gereksinimler mantıksal olarak alt gruplara ayrılır ve her grup gereksinim,tasarım,uygulama ve test aşamalarından geçer.Her artımda mevcut olan ürünün üzerine yeni işlevler eklenir.Proje bitimine kadar bu süreç devam eder.Modelin her aşamasında test olduğundan hata ayıklamak ve riskleri yönetmek daha kolaydır.Müşteri her aşama için değerlendirme verebilir ve bu da ilk teslimat faaliyetini düşürür fakat bu modelin toplam maliyeti şelaleden daha yüksektir.Maliyeti büyük ve uzun süreceği düşünülen projeler için kullanılır.



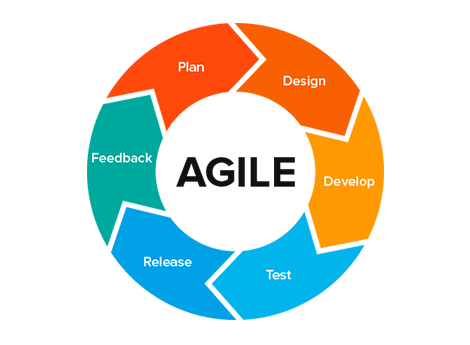
**Helezonik (Spiral) Model**

Helezonik model,bir diyagram olarak bakıldığında çok sayıda döngüye sahip bir bobine benziyor.Buradaki döngü sayısı her projeye göre değişmekle birlikte genellikle proje yöneticisi tarafından belirleniyor.Spiraldeki döngülerin her biri yazılım geliştirme sürecinde bir aşamadır.Model planlama,risk yönetimi,üretim ve müşteri değerlendirmeleri olarak 4 ana bölüm içerir.Planlama yapıldıktan sonra alternatifler değerlendirilerek risk analizi yapılır. Üretim aşamasında yazılımın tasarımı, gereksinimleri, kodlaması ve testleri yapılır.Daha sonra ürün hakkında müşterilerin değerlendirmeleri ve testleri yapılır.Çıkan sonuçlara göre aşamalar tekrarlanır.Bu modelin en iyi avantajlarından biri her aşamanın başında risk analizi yapılmasıdır.Böylelikle zaman ve maliyet unsurları kolay tahmin edilir.Modelin küçük projelerde kullanılması pek uygun değildir.Genelde proje büyük ve sürümlerin sık olması gerektiğinde bu model kullanılabilir.



**Çevik (Agile) Model**

Günümüzde en fazla tercih edilen yazılım yaşam döngü modelidir.Model geleneksel modellemelere göre daha kullanışlı ve esnektir.Çevik modelleme yazılımları daha verimli bir şekilde modellemeye ve belgelendirmeye yönelik pratik bir modellemedir.Artımlı bir model türüdür.Projenin geliştirilmesine başlamak için tüm ihtiyaçların ortaya çıkmasına gerek yoktur.Netleşmiş olan ihtiyaçlarla projeye başlanılabilir. Model de projenin boyutu fark etmeksizin proje küçük yinelemelere ayrılır ve her yineleme bir proje gibi ele alınır ve geliştirilir.Böylelikle geriye dönük hatalar daha kolay düzeltilir.Modelde insanlar ve etkileşimler daha önemlidir.Bu sebeple müşteriler,geliştiriciler sürekli iletişim halindedir.Müşteri memnuniyeti oldukça önemlidir ayrıca ekip içi motivasyonda bir o kadar önemlidir.Çevik modelin Scrum, Extreme Programming (XP), Kanban, Test Driven Development (TDD) gibi çeşitli uygulama şekilleri vardır.Kısa süreli projeler için tercih edilebilir.



**SCRUM**

Scrum günümüzde en çok tercih edilen Çevik model metodudur.Karmaşık problemler için uyarlanabilir çözümler yoluyla ekiplerin değer üretmesine yardımcı olan bir çerçevedir.Scrum,karmaşık işleri,projeleri yönetmek için bazı kurallar tanımlar.Scrum,çevik model karşısında daha katı kuralları olan bir yönetim sürecidir.Uygulanması çok kolay bir metottur ve aldığı hızlı ve kaliteli sonuçlar sebebiyle popülaritesini korumaktadır.Scrumun dayandığı 3 prensip vardır:Şeffaflık,Adaptasyon ve Denetleme.

Şeffaflık:Ekibin çalıştığı süre boyunca ortaya çıkan ürünün mevcut durumunu görebilmesi.

Adaptasyon:Ekibin sürekli olarak işlerin nasıl gittiğini araştırması ve mantıklı olmayan unsurları gözden geçirmesi.

Denetleme:Sürecin nasıl ilerlediğini görmek için sık sık denetim yapılması.

Bu süreç, fikirleri ürünlere,projelere dönüştürebilen kişilerden oluşan bir scrum takımına sahiptir.Bu takımda fikirleri toplayan değerlendiren bir ürün sahibi;scrumun kurallarına,teorilerine ve uygulamaları hakkında bilgi sahibi olan ve bunları takıma uygulatan bir scrum master ve ürünleri,projeleri istenilen sonucu üretecek şekilde tamamlayan bir geliştirme takımı vardır.



Scrum’un günümüzde bu kadar popüler olmasının sebepleri şunlardır:

* Zaman ve maliyet açısından tasarrufludur.
* Yeniliklere kolaylıkla adapte olabilir.
* Her teknolojiye uyum sağlayabilir.
* Ekip içi iletişimin çok yüksektir ve bu sebeple oluşan hatalar daha önceden çözülebilir.
* Müşteriyle sürekli bir temas halinde olması sebebiyle oluşabilecek anlamsızlıkların önüne geçilir.
* Karmaşık ve gereksinimleri tam belirlenmemiş olan projeler için güzel bir metottur.
* Sürekli yinelemeli ve aktif bir yapıdır.
* Büyük bir projeyi sprint denilen küçük parçalara ayırarak hatadan kaçınılmasını sağlar.

**KAYNAKÇA**

Doç. Dr. Deniz KILINÇ Yazılım Mühendisliği Temelleri ders notları

<https://www.linkedin.com/pulse/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-nedir-veysel-ugur-kizmaz/?originalSubdomain=tr>

<https://iskulubu.com/yazilim/yazilim-gelistirme-yasam-dongusu/>

<https://tr.wikipedia.org/wiki/Waterfall_model>

<https://www.ykcgrup.com.tr/selale-modeli-nedir/>

<http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/v-model.html>

<https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm>

<https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/>

<https://www.yazilimtestmerkezi.com/post/yazilim-gelistirme-modeli-nedir-cesitleri-nelerdir>

<http://www.aspmvcnet.com/tr/m/yazilim-muhendisligi/helezonik-tasarim-spiral-tasarim.html>

<https://www.argenova.com.tr/scrum-nedir#:~:text=Scrum%2C%20g%C3%BCn%C3%BCm%C3%BCzde%20kullan%C4%B1lan%20en%20pop%C3%BCler,kelimesi%20asl%C4%B1nda%20bir%20rugby%20terimidir>.

Medium hesap linki: <https://medium.com/@gokcenursonmez>

Linkedin hesap linki: <https://www.linkedin.com/in/gokcenursonmez-48/>

Github hesap linki: <https://github.com/gokcenur48>

**GÖKÇE NUR SÖNMEZ-210601694**