YÖNETİM BİLİŞİM SİSTEMLERİ

**YAZILIM SÜREÇ MODELLERİ VE SWOT ANALİZİ**

Ümeyir ATASOY

Danışman: Dr. Öğr. Üyesi Muhammed Fatih ALAEDDİNOĞLU

Erzurum

Kasım, 2022

İçindekiler

[Giriş 3](#_Toc119942196)

[1. Kodla ve Düzelt (Code and Fix) Modeli 3](#_Toc119942197)

[1.1 Kodla ve Düzelt (Code and Fix) Modeli | SWOT Analizi 4](#_Toc119942198)

[2. Çağlayan | Şelale (Waterfall) Modeli 5](#_Toc119942199)

[2.1 Çağlayan - Şelale (Waterfall) Modeli | SWOT Analizi 6](#_Toc119942200)

[3. V | V- Shaped Modeli 7](#_Toc119942201)

[3.1 V- Shaped Modeli | SWOT Analizi 8](#_Toc119942202)

[4. Evrimsel Geliştirme (Evolutionary Development) Modeli 9](#_Toc119942203)

[4.1 Evrimsel Geliştirme (Evolutionary Development) Modeli | SWOT Analizi 10](#_Toc119942204)

[5. Prototipleme (Prototyping) Modeli 11](#_Toc119942205)

[5.1 Prototipleme (Prototyping) Modeli | SWOT Analizi 12](#_Toc119942206)

[6. Spiral | Helezonik Model 13](#_Toc119942207)

[6.1 Spiral | Helezonik Model | SWOT Analizi 14](#_Toc119942208)

[7. Formal Sistem Geliştirme (Formal System Development) 15](#_Toc119942209)

[7.1 Formal Sistem Geliştirme (Formal System Development) | SWOT Analizi 16](#_Toc119942210)

[8. Yeniden kullanıma yönelik geliştirme (Re-use based development) 17](#_Toc119942211)

[8.1 Yeniden kullanıma yönelik geliştirme (Re-use based development) | SWOT Analizi 18](#_Toc119942212)

[9. Artımlı Geliştirme (Incremental Development) 19](#_Toc119942213)

[9.1 Artımlı Geliştirme (Incremental Development)| SWOT Analizi 20](#_Toc119942214)

[10. Birleşik Süreç (Unified Process) 21](#_Toc119942215)

[10.1 Birleşik Süreç (Unified Process) | SWOT Analizi 22](#_Toc119942216)

[11. Çevik Modeller (Agile Programming) 23](#_Toc119942217)

[11.1 Çevik Modeller (Agile Programming) | SWOT Analizi 24](#_Toc119942218)

[KAYNAKÇA 25](#_Toc119942219)

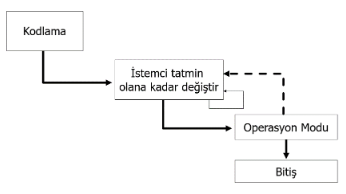
# Giriş

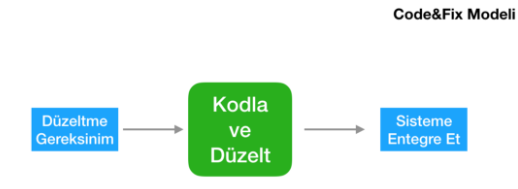
Yazılım süreç modelleri, yazılım geliştirirken karşılaşacağımız zorluk ve karmaşık durumları seçtiğimiz modele göre belli bir plana göre yapmamıza yarayan modellerdir. Bu modeller bizim yazılım geliştirirken yapacağımızı sistemi daha sistematik yapmamıza yarayacak bu yüzden çoğu şirket bu modellerden birini kullanır. Eğer bu modeller herhangi bir sistem yazılımını yaparken kullanılmazsa plansızlık, tasarımsızlık ve birçok hatalarla karşılaşılacaktır. Bu yüzden modeller önemli ve bu modelleri anlamak ve kullanmak daha önemli. Günümüzde kullanılan yaklaşık 50’den fazla SDLC modeli vardır. Ben bunlardan en popülerlerini anlatacağım.

# Kodla ve Düzelt (Code and Fix) Modeli

Basit bir yöntem olarak da söyleyebiliriz, en basit ürün geliştirme(Cowboy Coding) olarak adlandırılan bu modelde en kısa sürede hedefe gitme amaçlanır. Planlama ve analiz yapma gibi kısımlar ile çok vakit kaybetmeden ana problem üzerine yani bireysel belirlenen problem üzerine uğraşılır. İsminden de anlaşılacağı gibi belirlenen projeyi kodla ve düzelt şeklinde ilerlenir. Bu modeli genellikle öğrenciler ve bireysel geliştiriciler kullanır, takım arkadaşları için faydasızdır.

Modelimizi kullanılan projelerde genellik bitiş süresi belli olmaz genellikle de ve proje yarım kalır. Eğer büyük bir projeye kodla ve düzelt modeli ile başlıyorsak kodları düzeltmek çok uğraştırıcı olabilir. Herhangi bir planlamaya ihtiyaç duyulmaz.

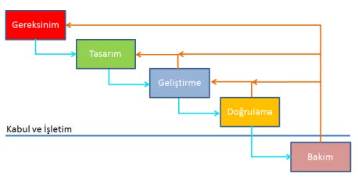




## 1.1 Kodla ve Düzelt (Code and Fix) Modeli | SWOT Analizi

# Çağlayan | Şelale (Waterfall) Modeli

Şelale modelinde yazılımı geliştirme sürecini gereksinim, tasarım, geliştirme ve en son bakım olarak sınıflandırabiliriz. Kullanımı ve yöntemi basit ve kolaydır. Büyük projelerde kullanımı zordur çünkü belli bir statik içinde lineer olarak ilerler ve büyük projelerde değişiklikler olacağından bu model büyük projeler için uygun olmayabiliyor. Bu modelde az önce sınıflandırdığımız sınıflara çok geri dönülüp işlem yapılır tekrar tekrar sınıflara geri dönülüp üzerinden geçilir.  İhtiyaç tanımlamaları çoğu kez açık bir şekilde yapılamadığından ötürü, hataların düzeltilme ve eksikleri giderme maliyetleri artırır. Eski askeri yazılımlarda bu model sık kullanılmıştır.

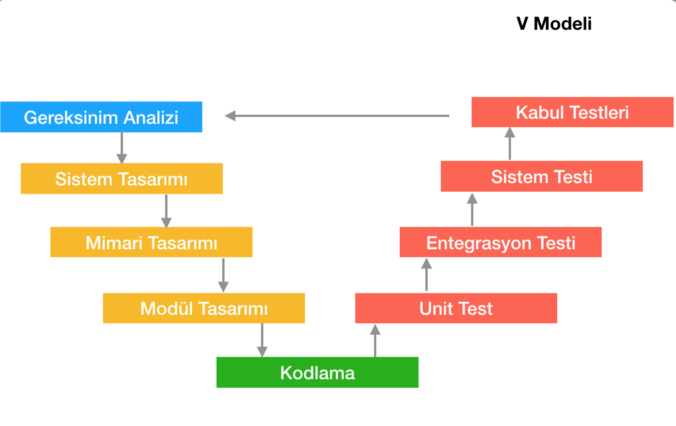
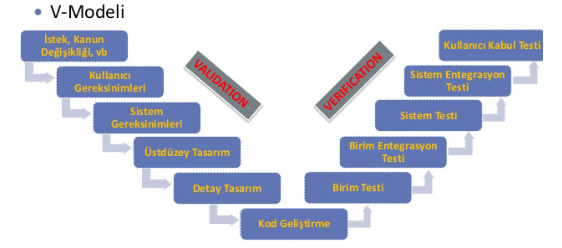
## 2.1 Çağlayan - Şelale (Waterfall) Modeli | SWOT Analizi

# V | V- Shaped Modeli

Çağlayan modelinin gelişmiş hali olarak düşünebiliriz V modelini. Şelale modeline Verification (Doğrulama) ve Validation (Onaylama) mekanizmasının eklenmiş halidir.

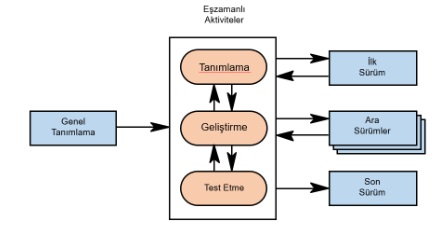
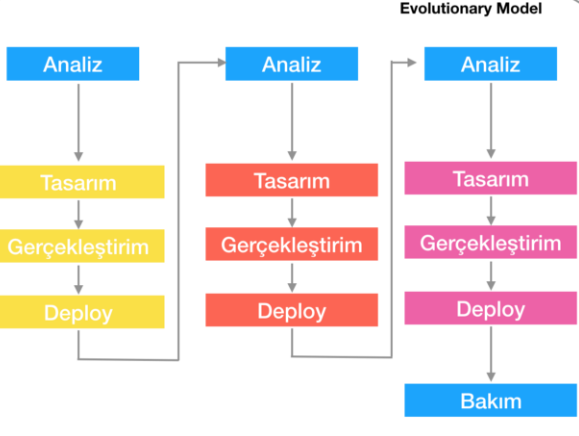
Sol taraf üretim, sağ taraf ise sınama bölümüdür. Üst ve alt seviye tasarım vardır. Üst seviye daha genel bir tasarımdır. Alt seviye de ise girdiler-çıktılar, beklentiler gibi daha detaylı bir tasarım mevcuttur. Bu yöntemde proje takibi kolaydır. Modelin kullanımı genel olarak basittir. Fakat aşamalar arası tekrarlamaları kullanmaz. Risk çözümleme ile ilgili aktiviteleri içermez. Bu modelde de şelale modeli gibi tekrara düşme vardır çünkü en son adımda müşteri kabul testini geçmezse başa döner ve müşterinin isteği gerçekleşir ve bu böyle sonsuza kadar gidebilir.

* Unit Test geçen Kodlarınız başarılı çalışıyor demektir.
* Entegrasyon Testi başarılıysa Modül ve Mimari Tasarımınız doğrudur.
* Sistem Testlerinden geçiyorsanız Sistem tasarımınız doğrudur.
* En son olarak müşteri kabul testlerinden geçiyorsanız Gereksinim analizini doğru yaptınız anlamına gelir.

## 3.1 V- Shaped Modeli | SWOT Analizi

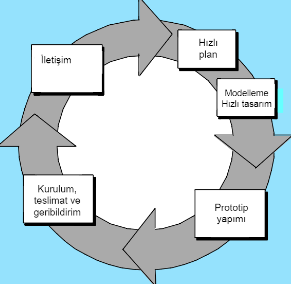
# Evrimsel Geliştirme (Evolutionary Development) Modeli

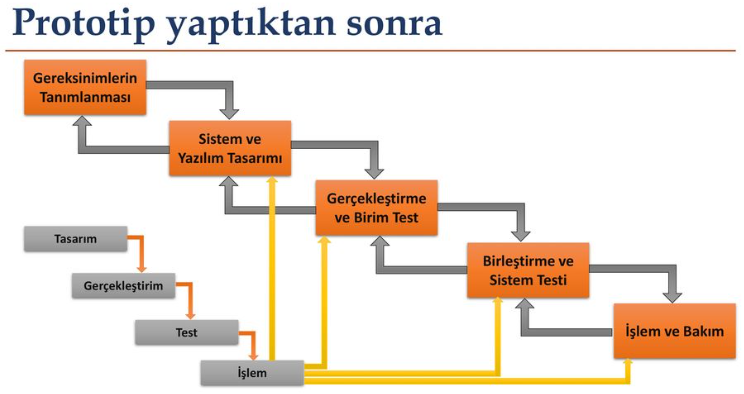
Genellikle banka uygulamaları için önerilmektedir. Bu model ilk tam ölçekli olan modeldir. Projenin başarıya ulaşması ilk evrimin başarısına bağlıdır. Kötü yanıysa değişiklik denetimine sahip olmamasıdır bu sebeple bakımı zor ve maliyetlidir. Müşteri geri bildirimleriyle ilerlediği için geliştirilmesi yavaştır. Küçük sistemler için veya büyük sistemlerin belirli parçaları(Kullanıcı arayüzleri) için kullanılabilir. İki çeşit evrimsel geliştirme vardır: keşifçi geliştirme (exploratory development) ve atılacak prototipleme (throw-away prototyping). Sistemler sıkla iyi yapılandırılmaz (sürekli değişiklik yazılımın yapısına zarar verir).  

## 4.1 Evrimsel Geliştirme (Evolutionary Development) Modeli | SWOT Analizi

# Prototipleme (Prototyping) Modeli

Bu modelde yazılım müşteri ve geliştirici arasında periyodik bilgi gidip gelmeleri sonucunda gelişir. İsminden de anlaşılacağı gibi prototip üretip üzerine katarak geliştirilir. Hızlıca gereksinimler elde edilir ve bütün çıktılara, bu çıktılar için gerekli girdilerin nasıl sağlanacağına, nasıl korunacağına, hangi işlemlere uğrayacağına karar verilir. Bu modelde, gereksinim analizi ve prototipleme için tasarım yapıldıktan​​ sonra, geliştirme​​ süreci başlatılır.​​ Prototipleme yaratıldıktan sonra, müşteriye değerlendirme için verilir.​​ Müşteri paketi test eder ve düşüncelerini, ürünü müşterinin tam beklentilerine göre düzenleyen geliştiriciye iletir.​​ Hedefler net değilse güvenlik açığı problemleri ortaya çıkabilir.





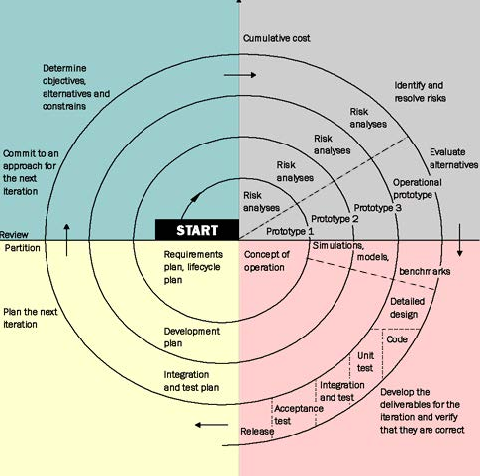
## 5.1 Prototipleme (Prototyping) Modeli | SWOT Analizi

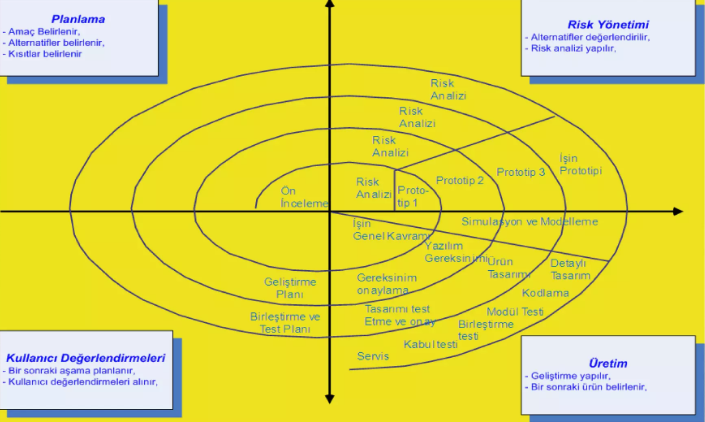
# Spiral | Helezonik Model

 Planlama, risk analizi, üretim ve kullanıcı değerlendirmesi olarak 4 ana aşamadan oluşur. Ve bu

aşamalar dönerek tekrar eder. Şelale modelinde yok sayılan riskleri göz önünde bulundurur. Proje çevrimlere ayrılır ve her bir çevrimin riskleri ayrı ayrı ele alınır. Çağdaş modellere son derece yakın bir modeldir. Riske duyarlı yaklaşımı potansiyel zorlukları engeller. Hataları erken gidermeye odaklıdır. Amacı riski seviye seviye azaltarak projenin başarılı bir şekilde tamamlanmasını sağlamaktır. Risk önemli bir faktördür bu model için.

1. **Hedeflerin Belirlenmesi ve Alternatif Çözüm Yollarının Bulunması:**Bu aşamada müşteriden elde edilen gereksinimler sonucunda hedefler belirlenir ve bu hedeflere göre çözüm önerileri üretilir.
2. **Çözüm Yolu Seçilir ve Risk Ortadan Kaldırılır:**Bu aşamada çözüm yolları değerlendirilir ve en iyi çözüm yolu belirlenip bu konuda prototip geliştirilir.
3. **Ürünün Bir Sonraki Aşaması için Geliştirme Yapılır:**Bu aşamada bu bulunan çözüm yoluna göre geliştirme ve testleri yapılır ve ürün bir sonraki aşamaya hazırlanır.
4. **Bir Sonraki Faz Planlanır:**Bu aşamada bir sonraki fazın planlaması yapılır. Operasyonel planlamalar, geliştirme planlamaları, entegrasyon ve test planlaması gibi.

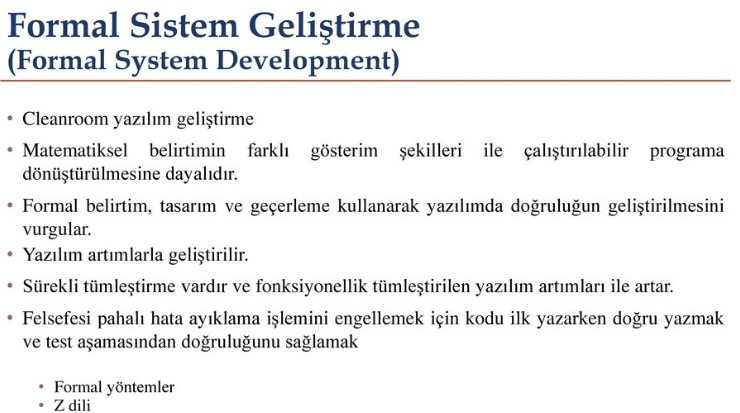




## 6.1 Spiral | Helezonik Model | SWOT Analizi

# Formal Sistem Geliştirme (Formal System Development)

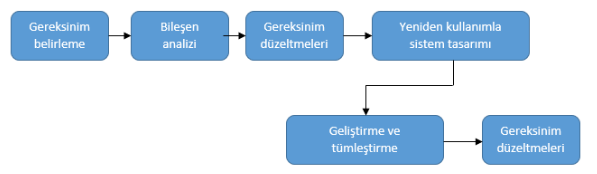
Bu model matematiksel bir modeldir. Karmaşık sistemleri geliştirmeye destek olur. Formal Sistem Geliştirme Metodu​​ kullanıcı sistemi kullanmaya başladığında karşısına çıkan belirtim hatalarını minimize eder.  Bu yüzden hatasız yazılım geliştirme imkanları sunar. Kodları ilk yazarken doğru yazmak test aşamasında doğruluğu kolaylaştırır.



## 7.1 Formal Sistem Geliştirme (Formal System Development) | SWOT Analizi

# Yeniden kullanıma yönelik geliştirme (Re-use based development)

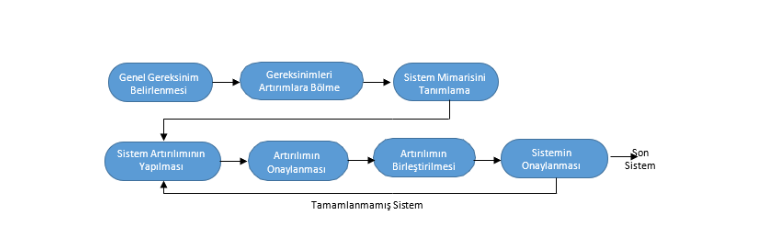
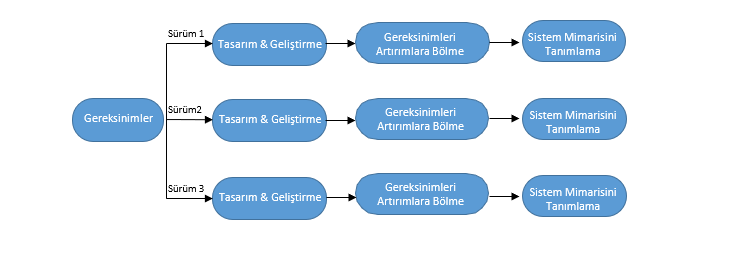
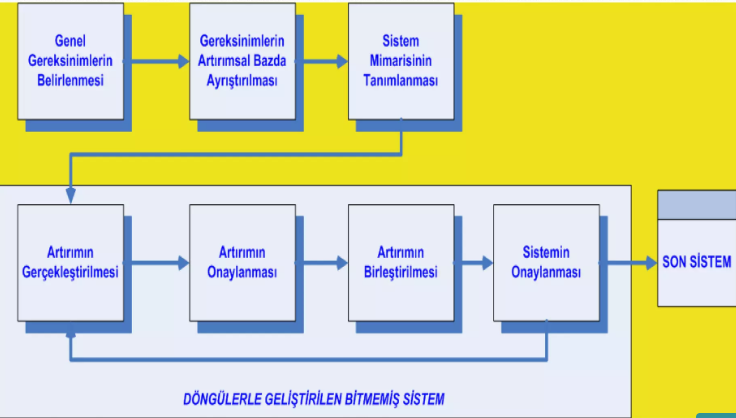
Daha önce hazırlanmış yazılımların kullanılması ile geliştirme yapılması son yıllarda popülaritesi artan bir yaklaşımdır. Bu tür yazılımlar geliştirilerek ilerler. Bu modeli kullanabilmek için öncesinde yazılmış farklı yazılımlara ve tecrübeye sahip olmak gerekir. Önceden oluşturulmuş classlarını kullanmak bu modelde bize avantaj sağlar. Ancak kullanılan yazılımı bir başka kurum oluşturduğu için anlaşılmakta zorluk çekilebilir.



## 8.1 Yeniden kullanıma yönelik geliştirme (Re-use based development) | SWOT Analizi

# Artımlı Geliştirme (Incremental Development)

Adı üstünde artımlı şekilde geliştirilir yazılım sistemi. Bir müşterinin ürünlerinde değişiklikler ihtiyaçlar varsa bu model o müşterinin değişikliklerine ayak uydurur. Artırımsal model bir takvime bağlı olarak yazılımı kesim kesim geliştirip teslim etmeye dayanır. Her bir yeni kesim öncekinin üstüne bazı ek işlevlerin eklenmesini öngörür. Gereksinim belirlemeye gerek yoktur çünkü müşteri gereksinimleri belirler. Sadece müşterinin istediği gereksinimler gerçekleştirileceği için risk azalır. Bu model için tecrübeli, deneyimli olmak gerekir.

## 9.1 Artımlı Geliştirme (Incremental Development)| SWOT Analizi

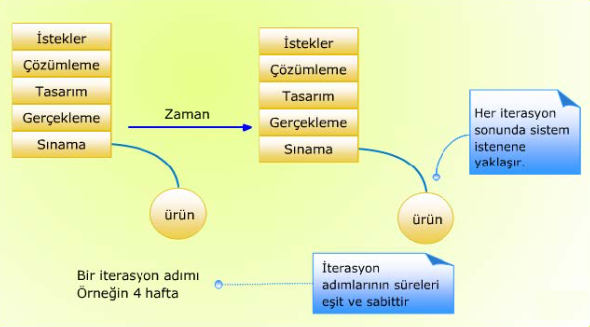
# Birleşik Süreç (Unified Process)

Nesne tabanlı (OOP) yazılım geliştirmek için var olan yöntemlerin kullanılmasıyla edinilen deneyimlerle bu süreçlerin en iyi özellikleri bir araya getirilerek bütünleştirilmiş bir yazılım geliştirme sürecidir. Yinelemeli, arttırmalı ve evrimsel aynı zamanda da risk güdümlüdür.

Yinelemeli (Iterative): Problemdeki istekler bir bütün olarak görülmez. Problemdeki hedeflerin bir kısmı ele alınır ve sınanmış bir ürün ortaya konulur. Her ürün oluşumunun sonunda bir sonraki yinelemeye (iterasyon) geçilir ve yeni istekler ele alınarak geliştirilir. Her iterasyon sonunda hedeflenen ürüne yaklaşılır. Her bir iterasyona ayrı bir proje gibi bakılır ve tüm aşamalardan geçirilerek sınanmış ve çalışır bir ürün elde edilir.

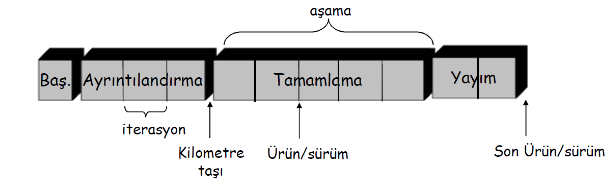
Artırmalı ve Evrimsel (Incremental and Evolutionary): Her yinelemenin ardından farklı ve yeni istekler ele alındığı için elde edilen ürünlerin özellikleri artarak gelişir ve istenilen ürüne daha fazla yaklaşılmış olunur.

Risk Güdümlü ( Risk-Driven): İlk iterasyonda en riskli kısımlar ele alınmalıdır. Böylece proje daha ilerlemeden oluşabilecek büyük problemler görülür ve gerekli önlemler alınabilir. Örneğin bütçe, personel miktarı veya zaman planı gibi değişkenler ayarlanabilir.



Birleşik süreçte yazılım geliştirme aşamaları şunlardır: başlangıç, ayrıntılandırma, tamamlama, yayımdır.

* Başlangıç: Vizyon kararı, fizibilite çalışması, tamam ya da devam kararı
* Ayrıntılandırma: Daha gerçekçi çözümleme, çekirdek yapının ve yüksek riskli kısımların yinelemeli olarak oluşturulması.
* Tamamlama: Daha az riskli ve düşük öncelikli kısımların yinelemeli olarak gerçeklenmesi.
* Yayım: Beta testleri, piyasaya sürme çalışmaları.



## 10.1 Birleşik Süreç (Unified Process) | SWOT Analizi

# Çevik Modeller (Agile Programming)

Günümüzde en çok kullanılan yazılım geliştirme süreci ise ‘’Agile (Çevik)’’ modeldir. Modelin ismine çevik adı vermelerinin sebebi bana göre müşterinin istediklerini hızlıca yapmaları ve hataları, sorunları hızlı bir şekilde giderilmesidir. Takım çalışmaları kullanır genellikle bu model ve bu model müşterinin gereksinimlerini hızlı bir şekilde tamamlar. Kaliteli yazılımları hızlı bir biçimde geliştirilmesini hedefler. “Heavyweight” olarak da bilinen eski, yavaş ve bürokratik sistemlere tepki olarak geliştirilmiştir. Agile metotları bir yazılım geliştirme yaklaşımı değil, geliştirme süreçleri topluluğudur. Merkezinde yinelemeli gelişim, yoğun iletişim ve erken müşteri geri bildirimi vardır**.**

Agile çalışma modeli yazılımcının isteklerini ön plana çıkarır ve geri bildirimlerden beslenir. Dünyada en hızlı ve en güvenli model olarak kabul edilir. Dünyada birçok yazılım şirketi bu modeli kullanır.  Bu modelin bu kadar popüler olmasını sağlayan özellikleri şöyle sıralanmaktadır:

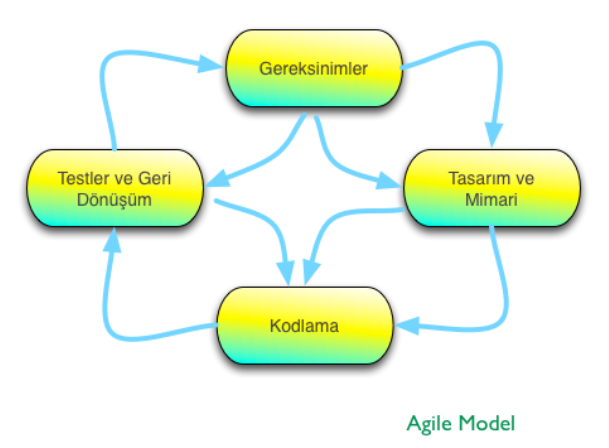
     1. Araçlar yerine, bireylerle olan etkileşimi arttırması

     2. Detaylı bilgilendirmeler yerine, yazılımın çalışmasına odaklanması

     3. Müşteri ile yapılan sözleşmeler yerine, müşteri ile iletişime odaklanması

4. Belirli ve değişimi kabul etmeyen bir çalışma planı yerine, değişime açık bir plan olması

     Bu modelde en önemli şeylerden bir tanesi de Zaman kavramıdır. Müşterinin isteğini en hızlı şekilde yapıp teslim etmek en önemli faktörler arasında yer alıyor. Günümüzde ise bir sistemi 1 yıl sonra teslim etmek ile 1 ay sonra teslim etmek arasında çok büyük bir fark var nedeni ise günümüzde rekabet çok fazla ve en hızlı gelişen ve geliştirilen yazılımlar tutulmaktadır. Bu modeli kullanan yazılımcılar “İlerlemenin en iyi göstergesi, çalışan yazılımdır” prensibiyle ilerlemektedir bu yüzden devamlı surette müşteriye bir ürün sunulur. Müşteriye sistemin teslimatını hızlı ve aralık yaparak müşteri memnuniyeti sağlar. Bu model için en iyi iletişim yolu yüz yüze görüşme kabul edilir.



## 11.1 Çevik Modeller (Agile Programming) | SWOT Analizi

# KAYNAKÇA

Yazılım Süreç Modelleri

https://www.linkedin.com/pulse/yaz%C4%B1l%C4%B1m%C4%B1n-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-beyza-nur-karako%C3%A7/?originalSubdomain=tr

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

https://hayririzacimen.medium.com/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-ve-s%C3%BCre%C3%A7-modelleri-70fdfb2f8f77

https://medium.com/architectural-patterns/yaz%C4%B1l%C4%B1m-geli%C5%9Ftirme-modelleri-62915545c51e

https://talentgrid.io/tr/yazilim-gelistirme-modelleri/

https://www.arge24.com.tr/yazilim-gelistirme-surec-modelleri/

https://www.codex.com.tr/yazilim-gelistirme-modelleri

https://letsbecool.com/tr/2020nin-populer-yazilim-gelistirme-modelleri/

https://www.slideshare.net/aoguzhany/yazilim-mimarileriaoy

https://medium.com/@secilcor/scrum-nedi%CC%87r-6a4326951dd8

https://enprobilisim.com/yazilim-gelistirme-sureci-modelleri-sdmp/

Swot Analizi

https://btm.istanbul/blog/swot-analizi-nedir-nasil-yapilir

https://www.albertsolino.com/blog/swot-analizi-nedir/

https://www.ideasoft.com.tr/swot-analizi-nedir-nasil-yapilir/