**İZMİR BAKIRÇAY ÜNİVERSİTESİ**

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM MÜHENDİSLİĞİ TEMELLERİ DERSİ

YAZILIM YAŞAM DÖNGÜ

MODELERİ

**SEREN AKDEMİR**

**210601687**

**İÇİNDEKİLER**

**Yazılım Nedir?**

**Yazılım Çeşitleri Nelerdir?**

**Yazılım Yaşam Döngüsü Nedir?**

**Yazılım Yaşam Döngüsü Aşamaları Nelerdir?**

**Yazılım Yaşam Döngüsü Modelleri**

**Scrum Nedir?**

**Modellerin Karşılaştırılması**

**Yazılım Nedir?**

Yazılım, elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir. Başka bir deyişle, var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlamlı anlatımlar bütünüdür.

**Yazılım Çeşitleri Nelerdir?**

**1. İşletim Sistemi Yazılımları**

Bu yazılımlar bilgisayar, tablet ve akıllı telefonların çalışabilmesi için gerekli olan işletim sistemleridir. İşletim sistemi yazılımları olmazsa, donanımlar çalışmamaktadır ve yapmak istediğiniz hiçbir işlemi gerçekleştirememektedir. İşletim sistemi yazılımların örnek vermek gerekirsek; Windows 95/98/XP/7/8/9/10, Vista, Pardus, Linux, Macos gibi bilgisayar sistemleri ve İOS, Windows Mobile, Android gibi akılı telefon ve tablet işletim sistemleri yazılımları bulunmaktadır.

**2. Uygulama Yazılımlar**

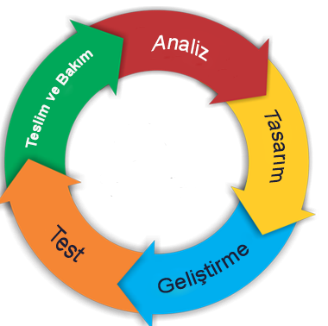
İşletim sistemi yazılımları ile uyumlu bir şekilde çalışan ve bilgisayar, akıllı telefon, tablet gibi uyumlu elektronik aletlere yüklenebilen, çalışan programlara, uygulama yazılımı adı verilmektedir. Uyumlu olduğu sistem yazılımları ile entegreli ve bağlı bir şekilde çalışmaktadırlar. Örnek vermek gerekirse; Oyunlar, Photoshop, Office Programları, winamp, anti virüs programları vb. tüm yazılım türleri uygulama yazılımlardır.

**3. Programlama Yazılımları**

İşletim yazılımlarının ve uygulama yazılımlarının oluşturulması, bu oluşturulan yazılımlar ile uyumlu, kullanılabilir elektronik veya teknolojik aletler içerisine yüklenmesi ve kullanılmaya hazır hale getirilmesi için gerekli olan programlardır. Örnek vermek gerekirse; Java, Pascal, C++, Visual Basic gibi uygulamalar programlama yazılımlarıdır.

**Yazılım Yaşam Döngüsü Nedir?**

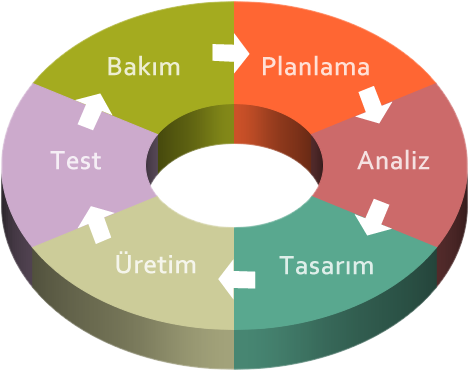
Yazılım Geliştirme Yaşam Döngüsü (Software Development Life Cycle, SDLC), yazılımları tasarlamak, geliştirmek ve test etmek amacıyla kullanılan bir süreç olarak ifade edilebilir. Yazılımın nasıl geliştirileceği, sürdürüleceği ve nasıl daha iyi hale getirileceğini açıklayan bir plandan oluşmaktadır. Buradan da yazılımın aslında bir ürün olduğu ve o ürünün de bir yaşam süreci olduğu anlamı çıkartılabilir. SDLC, müşteri isteklerini karşılayacak şekilde süre ve maliyet tahminleri dahilinde tamamlanması sağlanan kaliteli yazılım üretmeyi hedefler. Aynı zamanda SDLC, ISO/IEC 12207 dahilinde uluslararası bir standart olmayı da amaçlamaktadır.



**Yazılım Geliştirme Yaşam Döngüsü Temel Aşamaları**

Yazılım geliştirme yaşam döngüsü genel olarak 5 aşamadan oluşmaktadır. Bunlar; planlama, analiz, tasarım, test ve bakım alanlarından oluşmaktadır. Bunlar daha detaylı durumlar haline de getirilebilir. Bu aşamaları inceleyecek olursak;

1. **Aşama: Planlama:** Yaşam döngüsünün ilk aşamasıdır. Burada “Ne istiyoruz” sorusuna yanıt aranır. Başka bir ifade ile ekip bu aşamada projenin fizibilitesini ve projeyi nasıl başarılı bir şekilde faaliyete geçirebileceğini belirler.
2. **Analiz:** Projenin ne kadar sürede tamamlanacağı ve ne gibi risk durumları olduğunun belirlendiği aşamadır. Bu aşama temel UML diyagramlarının başlandığı ilk alandır. Elde edilen bilgiler belirli bir format halinde doküman haline getirilir.
3. **Tasarım:** Analiz kısmının tamamlanması sonucunda tasarım aşamasına geçilir. Burada “İstediğimizi nasıl elde edeceğiz” sorusuna cevap aranır. Proje sürecinin nasıl devam edeceği konusunda bir tasarlama işlemi gerçekleşir. Tasarımda en önemli tekniklerden bir tanesi de Soyutlama (Abstraction) tekniğidir. Bu teknik ile problemi daha basit hale getirerek problemdeki önemli kısımlara dikkat edilmesine olanak sağlar.
4. **Test:** Bu aşama projenin kodlandığı ve test edildiği aşamadır. Bu aşamada “İstediğimizi aldık mı?” sorusuna cevap aranır. Projede ortaya çıkabilecek hatalara ve eksik yanlara bu aşamada test edilerek ulaşılmış olunuyor.
5. **Bakım:** Yazılım ürünü tüm aşamaları tamamlayıp sahaya çıkartılıp teslim edildikten sonra bu aşama başlar. Bu aşama ürün ile ilgili alınan geri bildirimler sonucunda üründe yapılan yeni güncellemeler ve bakım durumlarının yapıldığı aşamadır.

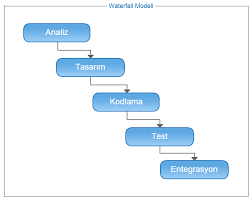


**YAZILIM YAŞAM DÖNGÜSÜ MODELLERİ**

Literatürde birden fazla Yazılım Yaşam Döngüsü modeli vardır. Her modelin avantajları ve dezavantajları da vardır. Fakat bir modelin her proje için en iyi olduğu konusunda bir olgu yoktur.

1. **Şelale Modeli (Waterfall Model)**

Şelale Modeli 1970 yılında Royce tarafından yayınlanan yazılım geliştirme sürecinin ilk modelidir. Bu model genellikle geleneksel model olarak adlandırılan klasik bir geliştirme modelidir. Ardışık bir yazılım yaşam döngüsü modelidir ve bu modeldeki gelişim ihtiyaç analizinin aşamaları olan tanımlama, tasarım, uygulama, test, entegrasyon ve bakım boyunca istikrarlı bir şekilde aşağıya doğru akar.



Bu modelin kilit noktası bir etkinlik tamamlandıktan sonra geri dönmemektedir. Dolayısıyla her etkinliği ve aşamayı gözden geçirip izlemek zorunludur. Kısaca bu model yazılım geliştirme sürecinin kullanıcı ihtiyaçlarını koda dönüştüren adım adım bir süreç olarak planlanabileceğini varsayar.

**Avantajları**

1. Proje ilerlemesinin doğru şekilde izlenmesini, oluşabilecek kaymaların erken tespitini ve ölçülebilir yazılım geliştirmeyi sağlar.
2. Projeler daha yönetilebilir ve maliyet aşımı olmadan zamanında teslim edilebilir.
3. Bütçe tahmini yapmayı kolaylaştırır.
4. Her aşamanın sonunda yapılan incelemeler kullanıcı katılımını sağlamaktadır.
5. Sistemi test etmek ve korumak için kullanılan belgeleri üretebilir.
6. Teknik açıdan deneyimsiz personel için iyi çalışır.

**Dezavantajları**

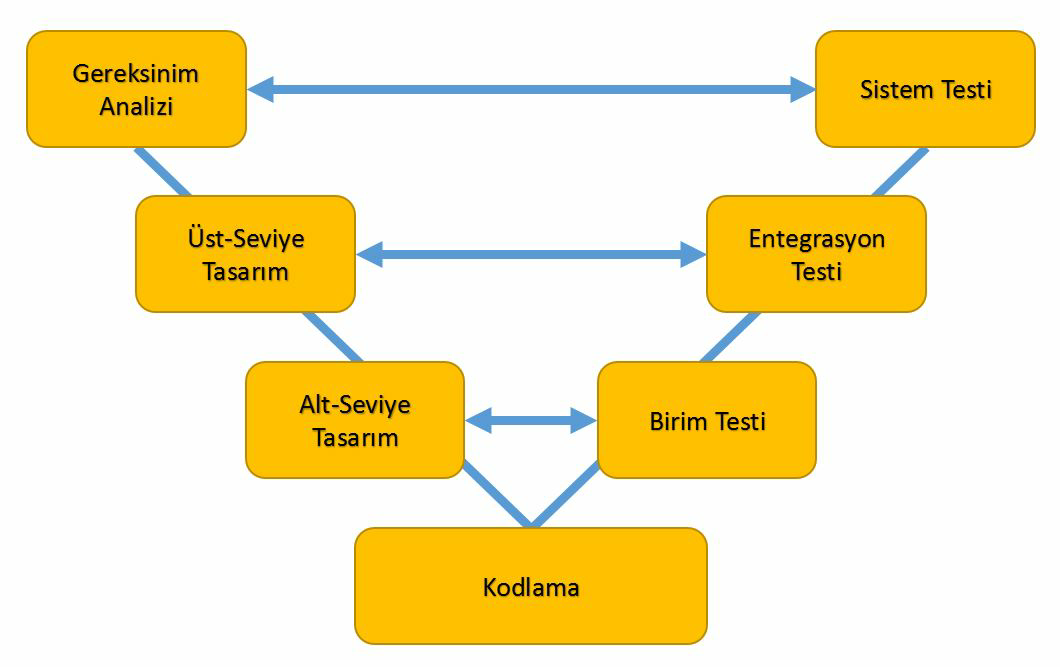
1. Planlama ve belgeleme için fazla zaman harcanır.
2. Her ihtiyacı önceden tahmin etmek zordur.
3. Esnek değildir.
4. Test aşamasına kadar fark edilmemiş hatalar olabilir.
5. Projenin sona ermesinden önce hiçbir gösterge yoktur.

**Şelale Modeli Hangi Projelerde Kullanılır?**

Şelale modeli, daha çok askeri vb sektörlerde, uzun süreli, analiz ve tasarımın çok önemli olduğu, hata yapmanın maliyetinin çok yüksek olduğu, kritik projelerde kullanılır. Örneğin bazı askeri projelerde, bu modelde analiz ve tasarım aşamaları birkaç yıl alabilmektedir.

1. **V-Şeklinde Model (V-Shaped Model)**

V modeli, süreçlerin yürütülmesinin ardışık şekilde gerçekleştiği modeldir. Bu modelde onaylama ve doğrulama vardır.



Bu modelde “V” yapısında bir yol izlenmektedir. Bu yol üzerinde sol kısım üretimi sağ kısım ise test işlemini ifade eder. Bu modelin çıktıları “Kullanıcı Modeli”, “Mimari Model” ve “Gerçekleştirim Modeli” dir. Bu modelde geliştirme sürecinin kullanıcı ile olan ilişkileri tanımlanır ve sistemin nasıl kabul edileceğine ilişkin test etme belirtimleri ve planları ortaya çıkarılır.

V Model kullanımı için uygun olan senaryolar şu şekildedir;

1. İş ihtiyaçları çok iyi tanımlanmış olmalı ve doküman haline getirilmiş olmalıdır.
2. Ürün tanıtımı sabit olup değişkenlik göstermemelidir.
3. Kullanılan teknoloji değişken olmamalıdır ve proje ekibi tarafından iyi bilinmelidir.
4. Tüm ihtiyaçlar eksiz şekilde tanımlanmalıdır.
5. Kısa süreli proje olmalıdır.

**Avantajları**

1. Disiplinlidir. Aşamalar birer birer tamamlanmıştır.
2. Gerekliliklerin çok iyi anlaşıldığı küçük projeler için iyi çalışır.
3. Basit ve kolay anlaşılırdır.
4. Yönetilmesi kolaydır.

**Dezavantajları**

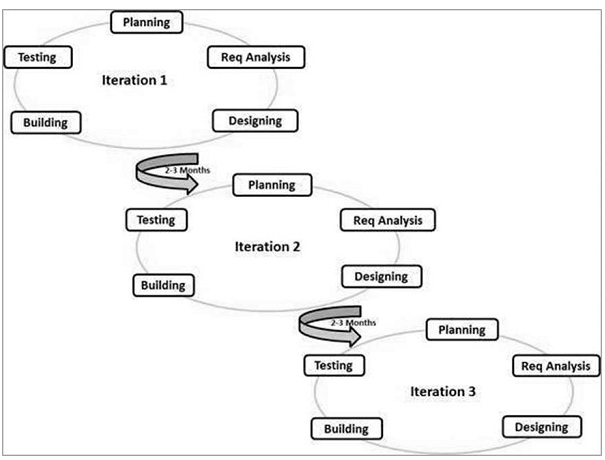
1. Karmaşık ve nesne yönelimli projeler için uygun değildir.
2. Uzun projeler için zayıftır.
3. Yaşam döngüsünün sonuna kadar çalışma yazılımları üretilmez.

**V Model Hangi projelerde kullanılır?**

Kullanım alanları; Tıbbi yazılım, havacılık filosu, yönetim yazılımı vb.

1. **Agile Model (Çevik Model)**

Çevik modelleme, yazılım sistemlerini etkili ve verimli bir şekilde modelleyen ve dokümantasyonunu yapan pratiğe dayalı yöntemlerin genel adıdır. Bu modelin başlıca özelliği veri modelleri ve ara yüzü modelleri gibi modelleme tekniklerinin neler olduğunu ve nasıl uygulanması gerektiğini söylemesidir.

****

**Agile metodunun genel prensipleri:**  
  
• Müşteriyi memnun etmek ve sürekli yazılım geliştirmek önemlidir.   
• Müşterinin rekabet avantajı için değişen gereksinimler benimsenmelidir.   
• Sık sık çalışan yazılımlar sunmaya odaklanılmalıdır. Teslimat, mümkün olan en kısa sürede yapılmalıdır.  
• Geliştiriciler ve iş adamları tüm proje boyunca birlikte çalışmalıdır.   
• Projeler motive olmuş insanlarla devam etmelidir. Onlara uygun ortam ve ihtiyaç duydukları destek sağlanmalıdır. İşlerini yapmak için güvende olmalıdırlar.   
• Yüz yüze iletişim, bir takıma bilgi aktarmanın en iyi yoludur.   
• Çalışan yazılım, ilerlemenin birincil ölçümüdür.   
• Çevik süreçler sürdürülebilir kalkınmayı teşvik ederler. Sponsorlar, geliştiriciler ve kullanıcılar belirsiz ve sürekli bir tempoyu koruyabilmelidir.   
• Teknik mükemmellik ve iyi tasarıma sürekli dikkat etmek çevikliği artıracaktır.   
• Sadelik, yapılmayan işi en üst düzeye çıkarma sanatı olarak kabul edilir ve esastır.   
• Kendi kendine organize ekipler genellikle en iyi tasarımları oluşturur.   
• Düzenli aralıklarla, takımın nasıl daha etkili olacağına dair düşünülmeli ve davranışlar buna göre düzenlemelidir.

**Avantajları**

1. Küçük ve orta çaplı projeler için uygundur.
2. Kaynak gereksinimleri minimumdur.
3. Sabit veya değişken gereksinimler için uygundur.
4. Değişken ortamlar için uygundur.
5. Yönetmesi kolay ve esnektir.
6. Erken kısmi çalışma çözümleri sunar.

**Dezavantajları**

1. Karmaşık bağımlılıkların kullanımı için uygun bir model değildir.
2. Sürdürülebilirlik için fazla risk taşır.
3. Bireysel bağımlılık çok yüksektir bunun sebebi dokümantasyonun minimum seviyede oluşturulmasıdır.

**Çevik Model Hangi Projelerde Kullanılır?**

Çevik Model’in kullanım alanları; erken geri bildirim gerektiren başlangıç girişimleri, küçük fonksiyonel parçalara bölünmesi kolay olan projeler, her yineleme aşamasında kademeli olarak geliştirilebilen projeler.

**Agile (çevik) ve geleneksel (şelale veya spiral) yöntem arasındaki farklar:**

**Temel Varsayımlar:**

**Geleneksel:** Sistemler tamamen tanımlanabilir, tahmin edilebilir, titiz ve kapsamlı bir planlama ile inşa edilebilir.

**Agile (Çevik):** Hızlı geri bildirim sağlanarak değişime dayalı sürekli tasarım geliştirilir. Test etme ilkeleri kullanılarak küçük ekipler tarafından yüksek kaliteli, uyarlanabilir yazılım geliştirilebilir.

**Kontrol:**

**Geleneksel:** Süreç merkezli   
**Agile:** İnsan merkezli

**Yönetim stili:**

**Geleneksel:** Komuta Kontrol  
**Agile:** Liderlik ve İş birliği   
  
**Bilgi yönetimi:**

**Geleneksel:** Açık   
**Agile:**Üstü kapalı   
  
**Rol ataması:**

**Geleneksel:** Birey- uzmanlaşmayı tercih eder   
**Agile:**Kendini organize eden ekipler- rol değişimini teşvik eder   
  
**İletişim:**

**Geleneksel**: Resmi   
**Agile:** Resmi değil   
  
**Müşterinin rolü:**

**Geleneksel: Ö**nemli   
**Agile:** Kritik   
  
**Proje döngüsü:**

**Geleneksel:** Görevler veya faaliyetler rehberliğindedir.  
**Agile:** Ürün özelliklerine göre yönlendirilir.  
  
**Geliştirme modeli:**

**Geleneksel:**Yaşam döngüsü modeli (Şelale, Spiral veya bazı varyasyonlar)

**Agile:** Evrimsel teslim modeli   
  
**İstenilen organizasyonel form / yapı:**

**Geleneksel:** Mekanik   
**Agile:**Organik   
  
**Teknoloji:**

**Geleneksel:** Kısıtlama yok   
**Agile:** Nesneye yönelik teknolojiyi destekler

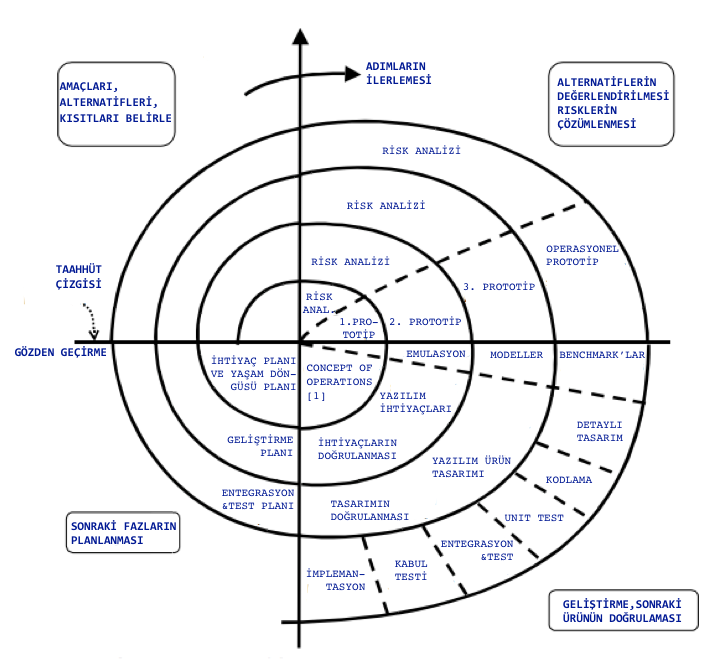
**SCRUM Nedir?**

Scrum; Agile proje yönetim metodolojilerinden biridir. Kompleks yazılım süreçlerinin yönetilmesi için kullanılır. Bunu yaparken bütünü parçalayan; tekrara dayalı bir yöntem izler. Düzenli geri bildirim ve planlamalarla hedefe ulaşmayı sağlar. Bu anlamda ihtiyaca yönelik ve esnek bir yapısı vardır. Müşteri ihtiyacına göre şekillendiği için müşterinin geri bildirimine göre yapılanmayı sağlar. İletişim ve takım çalışması çok önemlidir. 3 temel prensip üzerine kurulmuştur;

* **Şeffaflık;** Projenin ilerleyişi, sorunlar,gelişmeler herkes tarafından görülebilir olmalıdır.
* **Denetleme;**Projenin ilerleyişi düzenli olarak kontrol edilir.
* **Uyarlama;**Proje, yapılabilecek değişikliklere uyum sağlayabilmelidir.

1. **Spiral Model**

Spiral modeli şelale modelinden farklı olarak belli ve düz bir akıştan ziyade iteratif bir modeldir.



**Avantajları**

1. Değiştirme gereksinimleri karşılanabilir.
2. Gereksinimler daha doğru belirlenir.
3. Kullanıcılar sistemi başlangıçta görürler.
4. Geliştirme daha küçük parçalara bölünebilir.

**Dezavantajları**

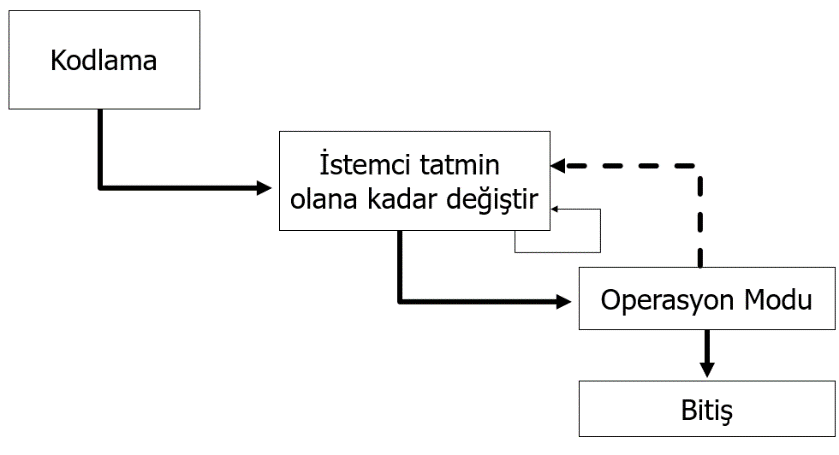
1. Yönetimi karmaşıktır.
2. Projenin sonu başlangıçta bilinmeyebilir.
3. Süreç karmaşıktır.
4. Küçük projeler için uygun değilidir.
5. Spiral süresiz olarak sürebilir.
6. Çok sayıda ara aşama ve dokümantasyon gerektirebilir.

**Spiral Model Hangi Projelerde Kullanılır?**

Spiral Model kullanım alanları; belirsiz iş ihtiyaçları olan projeler, inovatif ihtiyaçları olan projeler, büyük ve karmaşık projeler, Ar-ge faaliyeti vb.

1. **Kodla ve Düzelt (Code and Fix) Model**

Bu model genellikle resmi olmayan bir ürün fikriyle başlar ve program ürün “hazır” olana kadar ya da gerekli zaman bitene kadar kodlama yapılarak devam eder.



**Avantajları​​**

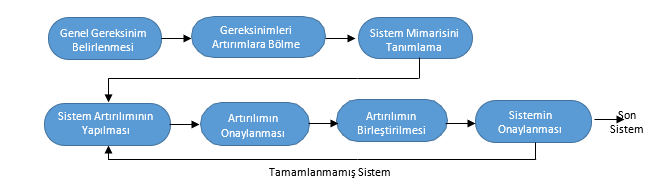
* Herhangi bir planlamaya ihtiyaç duyulmaz
* Çok küçük projelerde ya da kısa ömürlü prototiplerde uygulanabilir
* Program aşamaları çabuk geçilir​​
* Uzman görüşüne ihtiyaç düşüktür herkes bu modeli kullanabilir​​ [1]

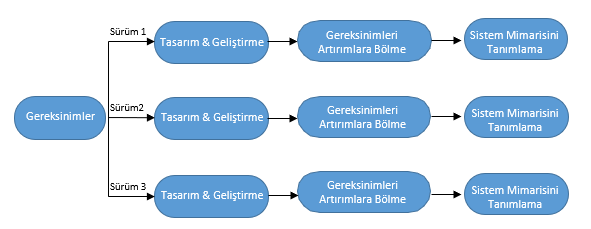
**Dezavantajları**

* Kontrollü değildir.
* Kaynak planlaması yoktur.​​
* Bitiş süresi belli değildir.
* Hataların bulunması ve doğrulaması zordur.
* Kodları düzeltmek maliyetlidir.
* Kodlar kullanıcının ihtiyacını karşılamayabilir.
* Esnek değildir, değiştirilmesi zordur​​

**Artımlı Geliştirme (Incremental Development)**

Eğer bir müşterinin ürünlerinde değişikliğe ihtiyaçları varsa, artımlı model ihtiyaç olan bu değişikliğe 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. Artırımsal model yazılım geliştirmenin kısıtlı sayıda çalışanla işin yapılmasını sağlama gibi bir üstünlüğü vardır. Ayrıca çalışanlar da her artırım geçildiğinde uygulama alanına ilişkin daha çok deneyim kazanmış olurlar. Bu modelde bir taraftan üretim bir taraftan da kullanım yapılır.​​ Önceki modellerde ürünlerdeki değişiklikler göz önünde bulundurulmaz. Bu model doğal olarak yinelemelidir.​​ Yeniden kullanılabilir bir ürün,​​ fonksiyonellik sağlamış bir şekilde tüm döngülerin sonunda ortaya çıkar.





**Avantajları**

* Sistem için gerekli olan gereksinimler müşterilerle belirlenir.
* Gereksinimlerin önemine göre teslim edilecek artımlar belirlenir
* Gereksinimlerin daha iyi anlaşılmasını sağlar
* Tüm projenin başarısız olma riskini azaltır.
* En önemli sistem özellikleri daha fazla test edilme imkanı bulmuş olur.
* Divide and Conquer (Böl ve Yönet) yaklaşımıdır​​.

**Dezavantajları**

* Artımları tanımlamak için tüm sistemin tanımlanmasına ihtiyaç vardır
* Gereksinimleri doğru boyuttaki artımlara atamak bazen zor olabilir.
* Deneyimli personel gerektirir
* Artımların kendi içlerinde tekrarlamalara izin vermez

**YAZILIM YAŞAM DÖNGÜ MODELLERİNİN KARŞILAŞTIRMASI**



Kaynakça

https://www.google.com/url?sa=t&source=web&rct=j&url=https://webdosya.csb.gov.tr/db/cbs/icerikler/salihsoylu\_tez\_v10-20180925134450.pdf&ved=2ahUKEwj5vsnB3Nn2AhXWRfEDHV93ClQQFnoECCsQAQ&usg=AOvVaw3kXO-QLYJoev5ghciyuCL1

https://www.google.com/amp/s/iskulubu.com/amp/yazilim/yazilim-gelistirme-yasam-dongusu/

https://www.google.com/amp/s/osmanozaydin.com/yazilim-yasam-dongusu-ve-agile-yazilim-gelistirme/amp/

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