

**YAZILIM YAŞAM DÖNGÜSÜ MODELLERİ(SDCL)**

**ÖĞRENCİ NO:220601028**

**İZMİR BAKIRÇAY ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ 1. SINIF**

**ARDA  
ÇAY**

# YAZILIM YAŞAM DÖNGÜSÜ MODELLERİ

## YAZILIM YAŞAM DÖNGÜSÜ NEDİR?

Herhangi bir yazılımın, ihtiyaç durumunda gerçekleştirilmesi ve sonrasında bakımında kadar geçen tüm süreçtir. Bir yazılım projesinin geliştirilmesi sadece kodlamaktan oluşmamaktadır. Basitçe bir proje geliştirilirken projenin planlama, analiz, tasarım, üretim ve test aşamaları yer almaktadır. Yazılım işlevleri ve ihtiyaçları sürekli değiştiği ve geliştiği için mutlaka bir döngü biçiminde düşünülmelidir. Bu aşamalar bir döngü halinde düşünülmesi gerekmektedir. Yazılım yaşam döngüsünün temel adımlarını genel olarak şu şekilde sınıflandırabiliriz:

**1.PLANLAMA:** İlk olarak gereksinim analizi yapılmasıyla başlar. Gereksinim analizi müşterinin istekleri doğrultusunda ortaya çıkacak olan yazılım hakkında ilk bilgilerin alındığı kısımdır. Diğer adımların başıdır, işin projelendirildiği fikrin ortaya çıktığı ve tartışıldığı kısımdır.

**2.ANALİZ(ÇÖZÜMLEME):** Bu aşamada gereksinimler detaylı bir şekilde incelenir ve projede nelerin istenildiği ile ilgili analiz çalışmaları da yapılır. Müşterinin istediğinin ne olduğu ve temel analiz üzerinde konuşulacak kavramların tanımlandığı aşamadır. İstenilenin tam olarak ne olduğu anlaşılmaya çalışır müşterinin fikriyle uyumuyorsa ileride daha büyük sorunlar ortaya çıkartabilir. Analiz sürecinde projenin tüm detayları ortaya çıkartılır.

**3.TASARIM:** Analiz çalışması sonucu ortaya çıkan proje detaylarına göre proje içerisinde yapılacak işlemler adım adım belirlenir. Tasarım mantıksal ve fiziksel olarak ikiye ayrılır. Mantıksal tasarımda sistemin yapısı anlatılır ve olası değişiklikler önerilir. Fiziksel tasarımda yazılımı içeren bileşenler ve bunların ayrıntılarını içerir. Tasarım dokümanının amacı, yazılım geliştiricinin yazılımını geliştirirken referans alacağı ve proje sürecinde projeye dahil olacak yeni yazılımcıların projeyi daha kolay anlayabilmesini sağlayabilmektir.

**4.GERÇEKLEŞTİRİM:** Bu aşama 3e ayrılır kodlama, test etme ve kurulum olarak. Kodlama işlemi planlama veya çözümleme aşamasında belirtilen programa dili geliştirme ortamı ve teknoloji yardımıyla yapılır.

**5.BAKIM:** Projenin gerçekleştirilmesinden sonra oluşabilecek hatların, yazılımının iyileştirilmesinin ve yeni özelliklerin eklenmesi sürecidir. Bu aşamada projedeki hatalar düzeltilir.

Yazılım yaşam döngü sürecinde yapım aşamasında geçirilen süreçler modellere ayrılmıştır. Geliştirme aşamasında nasıl bir yol izleneceği ve hangi sırada yapılacağı hakkında bilgi vermektedir. Bu modeller iki Kısıma ayrılmıştır geleneksel ve çevik yazılım süreci olarak. Geleneksel modeller: kodla ve düzelt, şelale(çağlayan)modeli, barok modeli, V süreç modeli, helezonik(spiral) modeli, gelişigüzel model, prototipleme, evrimsel geliştirme modeli, araştırma tabanlı model, formal sistem geliştirme, artırımsal geliştirme modeli, bileşen tabanlı geliştirme. Çevik yazılım süreci: scrum,extreme programming,

## YAZILIM YAŞAM DÖNGÜSÜ MODELLERİ İNCELENMESİ:

**HELEZONİK(SPIRAL) MODEL:** Waterfall ve tekrarlı modelin kombinasyonudur. Risk yönetimi için destek sağlayan en önemli modellerden biridir. Spiralin ilmek sayısı

projeden projeye göre değişmektedir. Sarmalın her bir döngüsü, sürecin her bir aşamasını ifade eder. Gereksinimlerin toplanması ve analizinden tasarım, uygulama, test ve bakıma kadar eksiksiz bir yazılım geliştirme döngüsünü temsil eden sarmalın her yinelemesi bir sarmal fikrine dayanmaktadır. Spiral Model, risk güdümlü bir modeldir, yani yazılım geliştirme sürecinin birden fazla yinelemesi yoluyla risk yönetimine odaklanılır. Spiral şu aşamalardan oluşur:

**PLANLAMA:** Spiral modelin ilk aşamasıdır, projenin kapsamının belirlendiği ve sarmalın bir sonraki yinelemesi için bir planın oluşturulduğu aşamadır.

**RİSK ANALİZİ:** Proje ile ilgili riskler değerlendirilir.

**MÜHENDİSLİK:** Bu aşamada önceki aşamalarda toplanan gereksinimlere göre geliştirilir.

**DEĞERLENDİRME:** Yazılımın müşteri gereksinimlerini karşılayıp karşılamadığı ve kalitesinin nasıl olduğu değerlendirilir.

**PLANLAMA:** Sarmalın bir sonraki yinelemesi, değerlendirmenin sonuçlarına göre yeniden planlanmasıyla başlar.

### **NEDEN HELEZONİK(SPIRAL) MODEL KULLANMALIYIZ?**

Risk yönetimi: Projenin her adımında riskler belirlenir analiz edilir ve yönetilir.

Sürekli gelişim: Helezonik modeli projenin her aşamasında ürün gelişimine odaklanır ve sürekli gelişim sağlar.

Esneklik: Hedeflerine ve değişen koşullara uygun esnek bir yapısı vardır.

İterasyonel yaklaşım: Proje geliştirme sürecini tekrarlayan döngüler halinde yönetir, değişiklikler yapılmasında ve ürünü iyileştirmesinde izin verir.

### **HELEZONİK(SPIRAL) MODELİN AVANTAJLARI:**

- Geliştirmesi zaman almaz hızlıdır
- Geliştirmesi sürekli olduğu için risk yönetimi kolaydır
- Her döngü başında risk analizi yapıldığı için zaman ve maliyet gereksiz harcanmaz
- Birçok yazılım modelini barındırır
- Projeyi küçük parçalara böler

### **HELEZONİK(SPIRAL) MODELİN DEZAVANTAJLARI:**

- Sonsuza kadar gidebilir
- Karmaşıktır
- Küçük projeler için pahalıdır
- Dokümantasyon fazladır
- Bütçeyi aşma riski barındırır

**V SÜRECİ MODELİ:** Waterfall modelinin gelişmiş hali olarak düşünülebilir. İşlemin V şeklinde sıralı bir şekilde yürütüldüğü SDLC modelidir. Doğrulama modeli olarak da bilinir. Her geliştirme aşaması için bir test aşamasının ilişkilendirilmesine dayanır. Aşamaları şu şekildedir:

**GERKESİNİM TOPLAMA VE ANALİZ:** V modelin ilk aşamasıdır, projenin kapsamını belirlemek için yazılım gereksinimlerinin toplandığı ve analiz edildiği aşamadır.

**TASARIM:** Üst düzey tasarım ve detaylı tasarım dahil olmak üzere yazılım mimarisi ve tasarımı geliştirilir:

**UYGULAMA:** Yazılımın tasarıma uygun olarak oluşturulmasını sağlandığı evredir:

**TEST ETME:** Test aşamasında yazılım, müşterinin gereksinimlerini karşılayıp karşılamadığı kontrol edilir ve kalitesinden emin olmak için test edilir.

**DAĞITIM:** Yazılımın kullanıma açıldığı evredir.

**BAKIM:** Müşterinin isteklerinin karşılanmasının devam ettirilmesi için yapılan düzenlemeler ve hata tespitleridir.

**KULLANIM ALANLARI:** Havacılık ve savunma sistemleri gibi güvenlik açısından riskli projelerde kullanılır. Gereksinimlerin açıkça tanımlandığı projeler için kullanılabilir. Basit projelerde kullanılabilir. Belirsiz ve tanımsız bir gereklilik yok ise kullanılabilir.

**V SÜREÇ MODELİ AVANTAJLARI:**

- Kullanımı kolaydır
- Planlama ve test tasarımı kodlamadan önce gerçekleştiği için zaman kazandırır
- Hatalar erken aşamalarda bulunur
- Hatalar bir sonraki aşamaya geçmesi engellenir

**V SÜREÇ MODELİ DEZAVANTAJLARI:**

- Kuralları katıdır
- Yazılım geliştirme aşamasında geliştirilir, bu yüzden prototipleri önceden üretilemez
- Dosyada değişiklik olursa tüm belgelerin güncellenmesi gerekmektedir

**ŞELE(WATERFALL)MODELİ:** Şelale modeli tanıtılan ilk ve en eski sdcı modelidir. Kullanımı aşırı basittir. Şelale modeli geliştirme sürecinde doğrusal bir sırada akış gösterir yani geliştirme sürecinde bir adıma geçebilmek için önceki adımın tamamlanmış olması gerekmektedir ve aşamalarda kesişme olmaz. Tüm yazılım geliştirme süreci aşamalara bölünmüş şekildedir. Şelale modelinin adımları şu şekilde ilerlemektedir:

**GEREKİNİM ANALİZİ:** Sistemin geliştirilecek tüm gereksinimleri bu aşamada toplanır ve belgelenir.

**SİSTEM TASARIMI:** İlk aşamadaki gereksinimler toplandıktan sonra bu aşamada incelenir ve tasarım hazırlanır. Bu tasarım donanım ve sistem gereksinimlerinin belirlenmesini sağlar. Genel sistem mimarisinin anlaşılmasına yardımcı olur.

**UYGULAMA:** Sistemin geliştirilmeye başlandığı adımdır. Her birim, birim testi olarak adlandırılan işlevselliği açısından geliştirilir ve test edilir.

**ENTEGRASYON VE TEST:** Gerçekleştirme aşamasında geliştirilen tüm birimler, her birimin test edilmesinden sonra bir sisteme entegre edilir. Entegre sonrası tüm sistem herhangi bir hata ve arıza var mı diye kontrol edilir.

**SİSTEMİN DAĞITIMI:** Testler sonucu ürün müşterilerin karşısına çıkartılır ve piyasaya sunulur.

**BAKIM:** İleride oluşturabileceği sorunlar için yeni güncellemelerin yapılması ve ürünü geliştirmek için daha iyi sürümlerin çıkartılması sağlanır ve piyasaya sürülür.

### **ŞELALE(WATERFALL)MODELİ KULLANIM ALANLARI:**

- Gereksinimleri çok iyi belgelenmiş ve açıktır olan projelerde
- Ürün tanımı sabittir olan işler
- Belirsiz gereksinimler olmayan projelerde
- Ürünü desteklemek için yeterli uzmanlığa sahip geniş kaynakları olan işlerde

### **ŞELALE(WATERFALL)MODELİ AVANTAJLARI:**

- Basit ve anlaşılması kolaydır
- Sağlam olduğu için yönetimi kolaydır, her aşamanın belli çıktıları ve inceleme süreci vardır.
- Aşamalar teker teker işlenir
- Küçük projeler için iyi iş yapar
- Görevleri düzenlemek kolaydır
- Süreç ve sonuçlar iyi bir şekilde belgelenmiştir

### **ŞELALE(WATERFALL)MODELİ DEZAVANTAJLARI:**

- Yüksek miktarda risk vardır
- Karmaşık projeler için iyi bir model değildir
- Gereksinimlerin değişme riski yüksek olduğu için büyük projelere göre değildir
- Değişen gereksinimleri karşılayamaz

**SCRUM MODELİ:** Çevik yazılım sürecinin bir üyesidir. Çevik yazılım yinelemeli iş dizilerini kullanır. Ürünü döngülere ayırır ve çok hızlı bir şekilde çalışan ürün sunar. Bu metot bir dizi yayın üretir her sürümün test edilmesi, bir sonraki sürüme dahil edilen bilgileri geri besler. Üretkenlik ve ürün sunma yaratıcılığı yüksektir ve karmaşık sorunların ele alabileceği bir yöntemdir. Verimli ve etkili bir yöntem olduğu için bir rapora göre yazılım ekiplerinin yüzde 70 i scrum modelini kullanmaktadır. Bu metot temel olarak yüksek geliştirme süreci ve paylaşımların çok olduğu yerlerde görülür. Sorumluluk ekip çalışması ve iyi tanımlanmış bir iş hedefine doğru ilerlemeye yöneliktir.

Ürün sahibi: Ürünün gereksinimlerini belirleyen kişidir

Geliştirme ekibi: Ürünü geliştiren ve hedefe ulaşmak için çalışan grup

Scrum master: Takım içi düzeni ve işleyişi sağlar

### **SCRUM TEMEL KAVRAMLARI:**

**ROLLER:** Ürün sahibi: Ürünün gereksinimlerini belirleyen kişidir

Geliştirme ekibi: Ürünü geliştiren ve hedefe ulaşmak için çalışan grup

Scrum master: Takım içi düzeni ve işleyişi sağlar

**TOPLANTILAR:** Sprint planlama toplantısında gereksinimlerin listeleri çıkartılır uygun görev dağılımı yapılır, risk değerlendirmeleri yapılır, olası gereksinim değişiklikleri ve pazarlama maliyetleri hesaplanır; gözden geçirme toplantılarında ise takım üyeleri gereksinim listesinin en üstünden başlayarak sprint sonunda gerçekleştirilecek sprint gereksinim listesini oluştururlar bu scrumun anahtar uygulamasıdır. Günlük toplantılarda ise 15 dakikalık her iş gününde belirli saatlerde gerçekleştirilir ve tüm takım katılır. Takımın ilerleyişi ve karşılaştıkları engelleri görmek sağlar.

**BİLEŞENLER:** Ürün gereksinim dokümanları proje boyunca yapılması gereken iş elemanlarının listesidir ve kullanışlı olması için devamlı bakım gerektirir. Sprint doküman olağan sprint için product backlogdan elde edilen iş ve görevleri kapsar. Amacı son ürünün bir parçası olan bir işlevselliği ya da çalışabilir bir parçayı elde etmektir. Kalan zaman grafiği işlerin ne kadarının yapıldığını ve normalde ne kadarının yapılması gerektiğinin karşılaştırılmasını sağlar.

**SCRUM MODELİ GÜNÜMÜZDE NEDEN POPÜLER:** İletişimde şeffaflık sağlar ve sürekli geri bildirim ile düzenli ilerleme ortamı sağlar. Karmaşıklığa bakmaksızın her tür endüstride ve her tür projede ürün geliştirmeyi destekleyecek şekilde yapılandırmıştır.

**SCRUM MODELİ NERELEDE KULLANILIR:** Yazılım, ürün geliştirme, pazarlama, finans gibi alanlarda stratejilerin planlanması ve yürütülmesinde kullanılabilir.

#### **SCRUM MODELİ AVANTAJLARI:**

- Hızlıdır ve para açısından da gayet verimlidir
- Büyük ürünü küçük alt parçalara bölerek çalışır
- Müşteri memnuniyeti çok önemlidir
- Kısa bir sprinte sahip olduğu için uyarlanabilir bir modeldir-
- Sürekli geri bildirim dayandığı için ürün kalitesi daha hızlı artar

#### **SCRUM MODELİ DEZAVANTAJLARI:**

- Sprintlerde değişikliğe izin vermez
- Tam olarak tanımlanmış bir model değildir
- Net bir tanımlı olmayan projeyi planlamak çok zor olabilir
- Günlük scrum toplantıları ve sık yapılan gözden geçirmeler ciddi miktarda kaynak gerektirir.

## KAYNAKÇA

<https://www.geeksforgeeks.org/>

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

<https://www.finoit.com/blog/sdlc-models-methodologies/>

<https://www.scrum.org/>

<https://medium.com/>

## HESAPLAR

<https://www.linkedin.com/in/arda-%C3%A7ay-69280026b/>

<https://medium.com/@ardacay3>

<https://github.com/Ardacay>