



İzmir Bakırçay Üniversitesi  
Mühendislik ve Mimarlık Fakültesi  
Bilgisayar Mühendisliği

## Yazılım Yaşam Döngü Modelleri

Hazırlayan:

Sudenaz Onaran:220601060

## Yazılım Yaşam Döngü Modeli Nedir?

Yazılım yaşam döngüsü modeli yazılım geliştirme sürecini yönetmek ve kontrol altında tutmak için kullanılan bir yöntemdir. Bu model yazılımın tasarımından başlayarak geliştirilmesi, test edilmesi, kullanıma sunulması ve bakım gibi tüm aşamaları kapsar. Yaşam döngü modelinin aşamalarına bakacak olursak öncelikle gereksinimler analizi aşaması vardır, bu aşamada yazılımın geliştirilmesi için ihtiyaç duyulan gereksinimler belirlenir. Bu aşamada kullanıcının ihtiyaçları, yazılımın işlevleri ve performans kriterleri tanımlanır. Bu aşamada müşteri, kullanıcı ve yazılım geliştirme ekibi arasındaki iletişim çok önemlidir. Gereksinimler analizinde, işlevsel gereksinimlerin yanı sıra yazılımın fonksiyonel gereksinimleri de belirlenir. Gereksinimler belirlendikten sonra ise tasarım aşamasına geçilir. Bu aşamada yazılımın mimari tasarımı, kullanıcı arayüzü, veritabanı tasarımı, algoritmalar ve veri yapıları gibi detaylar belirlenir. Tasarım aşaması, sistem bileşenlerinin ve modüllerinin birlikte nasıl çalışacağına dair bir plan hazırlanmasıdır. Yazılım geliştirme aşaması, yazılım tasarımına dayanarak kodlama yapma sürecidir. Bu aşamada yazılım geliştiricileri, yazılımın tasarımına göre bunun doğrultusunda kodlama sürecine başlarlar. Kodlama sürecinde programlama dili ve yazılım geliştirme platform seçimleri önemlidir. Test aşaması ise yazılımın hatalarının tespit edilmesi ve giderilmesi aşamasıdır. Test sürecinde yazılımın doğruluğu ve kalitesi belirlenir. Test süreci: yazılımın işlevsellik testleri, performans testleri, kullanılabilirlik testleri, güvenlik testleri, sistem testleri ve entegrasyon testlerinden oluşur. Dağıtım aşaması yazılımın kullanıma sunulma aşamasıdır. Bu aşamada yazılımın kurulumu, konfigürasyonu ve kullanıcıların eğitimi gerçekleştirilir. Yazılımın kullanılabilir olması kullanıcılar açısından oldukça önemlidir. Bakım aşamasında yazılımın kullanım sürecinde oluşabilecek hataların giderilmesi, yazılım güncellemeleri ve iyileştirmeler yapılır. Bakım aşaması yazılımın devamlı olarak geliştirilmesi sürecidir. Yazılım yaşam döngü amaçları birbiriyle bağlantılı ve sıralıdır. Bağlantıda veya sırada bir sorun bir bozulma yaşanır, yazılımın kullanımında birçok sorun çıkacaktır. Aşamaların doğru bir şekilde gerçekleştirilip yerine getirilmesi gereklidir. Yazılım yaşam döngüsü modeli farklı şekilde uygulanabilir. Örneğin scrum modeli, barok modeli, arıtsal geliştirme modeli, şelale yaşam döngü modeli, v süreç modeli, helezonik model gibi farklı modeller uygulanabilir. Her modelin kendine özgü avantajları ve dezavantajları vardır ve önemlisi seçilen model proje gereksinimlerine uygun olmalıdır.

## Yazılım Yaşam Döngü Modelleri

### Scrum Modeli

Seçtiğimiz modelde en önemlisi proje gereksinimlerine uygun olmasıdır. Öncelikle scrum modelinden bahsedecek olursak bu model yazılım geliştirme sürecinde kullanılan bir çevik modeldir. Çevik modeller yazılım projelerindeki başarı oranlarını ve ekip içindeki iletişimi arttırmak üzere ortaya çıkmıştır. Scrum modeli ekip çalışmasına dayalı bir yaklaşım kullanır ve esnek bir süreçtir. Scrum büyük projeleri parçalara bölerek çalışır ve her birine “sprint” adını verir. Her sprinti teker teker geliştirir. Scrum modeli bir çok aşamadan oluşur. Öncelikle ilk aşamamız ürün backlog oluşturma bu aşamada ürün sahibi, müşterinin isteklerini toplayarak ürünün geri bildirimini ve özelliklerini belirler. Bu özellikler ile bir liste oluşturulur. Sonraki

aşama sprint planlamadır.Bu aşamada ürünün listesinden yani backlogundan alınacak işleri seçer ve bunları gerçekleştirmek için bir sprint planı yapar.Sprint geliştirme aşamasında,sprint planlamada belirlenen özellikleri ve işleri gerçekleştirir.Geliştirme ekibi günlük toplantılarla bir araya gelerek,hedeflere ulaşmak için yapacaklarını planlarlar.Bu süreçte ürün sahibi işin ilerlemesini takip eder.Sonraki aşamada geliştirme ekibi;geliştirilen özellikleri,işleri ve sprint hedeflerini inceler.Bu süreçte ürün sahibi ve müşteriler yapılan işleri görür ve geri bildirim verirler.Sprint retrospektifi aşamasında sprint sonrasında alınacak önlemleri değerlendirir ve bu süreçte ekibin kendini geliştireceği alanlar değerlendirilir.Scrum günümüzde en çok kullanılan yazılım geliştirme yöntemidir.Sadece yazılım geliştirme değil birçok sistemin geliştirilmesinde de kullanılır.Scrum modelinin birçok avantajları vardır;zamandan ve paradan büyük ölçekte tasarruf edilmesi,yüksek teknolojiler ve son gelişmelere kolaylıkla uyum sağlayabilmesi,karmaşık gözüken ve gereksinimi tam olarak belirlenmemiş projeler için ideal olması,ekip içi iletişimin yüksek olması ve bu sayede hataların erken farkedilip düzeltilmesi,kullanıcıdan sürekli geri bildirim alınması ve hataların düzeltilmesi,diğer yazılım geliştirme modelleri gibi yinelemeli olması,gelişen gereksinimlere hızlı bir şekilde tepki verilmesidir.Her model gibi scrum modelinin avantajları olduğu gibi dezavantajları da vardır;scrumun esnek yaklaşım sunması planlama ve tahmin yapmaktan zorlanan bazı takımlar için dezavantaj olabilir.Birçok takım üyesinin sürekli bir araya gelmesi gerektiğinden büyük takımlar için koordinasyon ve zamanlama açısından zor olabilir.Scrum risk yönetimini yeterince ele almaz bu takımların belirsizliklere ve riske maruz kalmasına neden olabilir.Scrum teknik konulara yeterince odaklanamaz bu yüzden bazı teknik sorunlar ortaya çıkabilir.Scrum çok fazla popüler olsa da bunun gibi bazı dezavantajları da vardır.Scrum modeli özellikle karmaşık ve değişkenlik gösteren projelerde kullanılır.Scrum modeli,yazılım geliştirme projelerinde sıklıkla kullanılan bir modeldir.Yeni bir ürün geliştirilmesinde,mevcut bir ürünün geliştirilmesinde ve yeniden tasarlanmasında,karmaşık yazılım projelerinde,yeni teknolojilerin kullanıldığı projeler,belirsizliklerin ve risklerin bulunduğu projelerde kullanılabilir.

## Barok Modeli

Yaşam döngü adımları doğrusal bir şekilde ele alınır ve geliştirilir.Geleneksel yazılım geliştirme modellerinden farklı bir yaklaşım sergiler ve kullanıcının ihtiyaçlarına daha iyi cevap verir. Bu model, kullanıcıların ihtiyaçlarını doğrudan anlamaya ve yazılımın bunları karşılamasına odaklanır.Döngü yoktur ve dökümantasyon diğer modellerden farklı olarak ayrı bir süreç olarak ele alınır.Barok modelinin avantajları:Kullanıcının ihtiyaçları doğrudan modelleme sürecine dahil edildiği için, sonuçta ortaya çıkan yazılımın kullanıcıların ihtiyaçlarını daha iyi karşılaması beklenir.Esnek bir yaklaşım sergiler ve gerektiğinde değişikliklere kolayca adapte olabilir.Bu, geliştirme sürecindeki gereksinim değişiklikleri ile başa çıkmayı kolaylaştırır.Dezavantajları:Zaman ve kaynak açısından maliyetli olabilir ve karmaşık projeler için uygun değildir.

## Aritımsal Geliştirme Modeli

Aritımsal geliştirme modelinde proje parçalara bölünür ve kullanıcının önceliğine göre bu parçalar sıralanır.Sıralanan bu parçalar bittiğinde sırasıyla birer ana ürün geliştirilmiş olur ve

bu ara ürünler de kullanıcı tarafından kullanılır.Bu süreçte işlevsellik bir dizi artımlı adımda gerçekleşir.Bu model yazılımın tüm işlevselliğini tek bir aşamada gerçekleştirmek yerine bir dizi artımlı aşamada gerçekleştirerek yazılımın işlevselliğini arttırır.Her bir artım bir öncekinden geliştirilmiş ve test edilmiş bir işlevsellik sağlar.Bu modelin bir kısmında üretim kısmı dönerken diğer kısmında kullanım kısmı sürer.Aritımsal geliştirme modeli uzun zaman alabilecek yazılımlar ve ürünün eksik işlevsellikle çalışabileceği türdeki yazılımlar için uygun bir modeldir.Bu model müşteri tarafından beklenen işlevselliği tamamlamak için üç aşamadan oluşur;planlama aşamasında müşterinin ihtiyaçlarının analiz edilmesi,önceliklendirilmesi ve işlevselliğinin artımlı aşamalarda geliştirilmesi için bir plan oluşturulur.Geliştirme aşamasında yazılımın artımlı aşamalarda geliştirilmesi,test edilmesi ve entegre edilmesi sürecidir.Değerlendirme aşaması ise artımlı aşamaların tamamlandığı ve yazılımın müşteri beklentilerine uygun olarak tanımlandığı aşamadır.Bu model de scrum modeli gibi kullanıcıların ihtiyaçlarına göre esneklik sağlar.Aritımsal geliştirme modelinin avantajları;tüm projenin başarısız olma ihtimalini azaltır,ara ürünler yazılımın geliştirilmesinde önemli bir yere sahip olur,gereksinimlerin daha iyi anlaşılmasını sağlar,en önemli sistem özellikleri daha fazla test edilme olanağı bulur,müşteri memnuniyeti yüksektir.Bu modelin dezavantajları ise; her bir parçanın kendi içinde tekrar etmesine izin verilmez bu yüzden de bir ara ürünün bitip diğeri başlayana kadar herhangi bir değişiklik yapılamaz, parçaları oluşturmak için de bu sistemin detaylı bir şekilde tanımlanması lazımdır,her artımın tamamlanması için ekstra bir zaman ve maliyet gerekir,her artım yazılımın karmaşıklığını arttırır,artımları tamamlamak için tüm sistemi tamamlamak gerekir,artımların kendi içinde tekrar etme özelliği yoktur.

## Şelale Yaşam Döngü Modeli

Bu model diğer güncel yaşam döngü modellerinin temelidir denilebilir. Şelale yaşam döngü modeli, yazılım geliştirme sürecinde kullanılan en eski ve en yaygın modellerden biridir. Bu model, adını şelale benzeri bir şekilde ilerleyen adımlardan almaktadır. Her bir adım, bir önceki adımdan sonra başlar ve tamamlandıktan sonra bir sonraki adıma geçilir. Bu modelde, her bir adım, ayrı ayrı planlanır, geliştirilir ve tamamlanır. Bu modelde her aşama en az bir kez tekrar edilir.Şelale modelinin aşamaları;gereksinim analizi aşamasında yazılımın gereksinimleri belirlenir, analiz edilir ve dokümanite edilir. Bu aşamada, müşteri ile iletişim kurulur ve ihtiyaçları anlaşılır.Tasarım aşamasında yazılımın tasarımı yapılır. Bu aşamada, yazılımın yapısı, bileşenleri, arayüzleri, algoritması, veri yapıları ve diğer ayrıntılar belirlenir.Geliştirme aşamasında yazılımın kodlanması yapılır. Bu aşamada, yazılımın programlama dili kullanılarak kodlanması gerçekleştirilir.Test aşamasında yazılımın işlevselliği, performansı, doğruluğu ve güvenilirliği test edilir. Hata ve kusurlar tespit edilir ve düzeltilir.Dağıtım aşamasında yazılım, kullanıma hazır hale getirilir ve müşteriye teslim edilir. Bu aşamada, yazılımın kurulumu, entegrasyonu, eğitimi ve bakımı gerçekleştirilir. Her bir adım, bir önceki adımın tamamlanmasını gerektirir. Şelale modelinde geri dönüş yoktur ve her adım tamamlandıktan sonra bir sonraki adıma geçilir. Şelale modeli çok iyi tanımlanmış ve kısa sürede bitebilecek projelerde kullanılır.Şelale Yaşam Döngü Modelinin avantajları:Doğrusal ve sistematik bir yaklaşım sunar.Her bir adım ayrı ayrı belgelenir, bu nedenle yazılımın dokümantasyonu sağlanır.Her aşama, önceki aşamanın tamamlanmasını gerektirdiği için, hatanın erken tespiti ve düzeltilmesi sağlanır.Müşteri, yazılımın

tasarımından başlayarak sürece dahil edilir, bu nedenle müşteri ihtiyaçlarına daha iyi cevap verilir. Her model gibi avantajları olduğu gibi dezavantajları da vardır: Model doğrusal bir yaklaşım öngördüğü için, geri dönüşler zor veya imkansız hale gelebilir, bundan dolayı yazılım maliyeti artar. Yazılım gereksinimleri ve tasarımı tamamlanana kadar diğer adımlara geçilemez, bu nedenle süreç zaman alır ve genelde yazılımların hazırlanması ve kullanıma sürülmesi uzun bir süreçtir fakat şelale döngü modeli bunun için uygun değildir. Genelde yazılımcılar bir an önce programı kodlama ve çalıştırma amacıyla oldukları için ve bu modelde kodlama çok az bir kısımda olduğu için motivasyon kaybı yaşanabilir. Şelale yaşam döngüsü modeli, genellikle büyük ve karmaşık yazılım projelerinde kullanılır. Bu projeler, yazılımın birçok farklı bileşeninin bir araya getirilmesini gerektiren ve tamamlanması uzun zaman alan projeler olabilir. Özellikle savunma, havacılık, uzay, telekomünikasyon ve büyük işletmelerde kullanılan yazılım projeleri, şelale yaşam döngüsü modeline uygun olabilir. Bu projelerde, yazılımın tasarımı ve geliştirilmesi genellikle önceden belirlenmiş bir plana göre gerçekleştirilir ve tamamlanması gereken birçok farklı adım vardır.

## V-Süreç Modeli

V- süreç modeli şelale modelinin gelişmiş hali olarak düşünülebilir. V-süreç modeli, yazılım geliştirme yaşam döngüsü modeli olarak kullanılan ve aşağıdan yukarıya doğru bir yaklaşım sergileyen bir modeldir, sol taraf üretim, sağ taraf ise sınamadır. Bu döngü esas olarak 3 modelden oluşmaktadır. Kullanıcı Modeli, Mimari Model, Gerçekleştirim Modeli. Kullanıcı Modeli yazılımın gereksinimlerinin belirlendiği aşamadır. Bu aşamada, kullanıcının beklentileri, ihtiyaçları ve gereksinimleri toplanır ve bir gereksinim belgesi hazırlanır. Bu belge, sonraki aşamalarda yazılımın geliştirilmesi için temel bir kaynak oluşturur. Kullanıcı modeli aşamasında, beklentileri ve ihtiyaçları karşılamak için kullanılacak teknolojiler ve gereksinimlerin tanımlanması için bir çözüm tasarlanır. Mimari Model aşamasında kullanıcı belirlenen gereksinimlerin karşılanması için bir çözüm tasarlar. Bu tasarım, yazılımın mimarisini, tasarım özelliklerini, fonksiyonel ve teknik gereksinimleri, yapılandırma ve bağımlılık ilişkilerini içerir. Mimari model aşamasında, gereksinim belgesi ve çözüm tasarımı arasındaki uyum kontrol edilir. Bu aşama, geliştirme sürecinin gelecekteki adımlarının temelini oluşturur ve aynı zamanda bir referans noktası olarak kullanılır. Gerçekleştirme Modelinde yazılım geliştirilir, kodlanır ve test edilir. Bu aşamada, yazılımın gerçekleştirilmesi için kullanılacak programlama dili, araçlar ve platformlar belirlenir. Kodlama ve test işlemleri gerçekleştirilir ve yazılımın geliştirme süreci tamamlanır. Bu aşamada, her bir geliştirme aşaması için uygun test aşamaları da gerçekleştirilir. Test aşamaları, yazılımın gereksinim belgesi ve çözüm tasarımı ile uyumlu olmasını sağlar. V-süreç modelinin avantajları: Geliştirme sürecinde tüm aşamaları doğru bir şekilde test edildiği için ürünün kaliteli ve verimli olması sağlanır. Doğru bir şekilde sıralandığı için hataların erken tespiti yapılır ve doğru şekilde çalışmasını sağlar. Model kullanıcının projeye katkısını artırır. Proje yönetimi ve takibi kolaydır. V-Süreç Modelinin dezavantajları: Esnek olmayan bir yaklaşım olduğu için değişikliklerin yapılması zor olabilir. Bu, geliştirme sürecindeki değişikliklerin yönetimini zorlaştırabilir ve ekstra zaman ve kaynak gerektirir. V-süreç modeli, karmaşık projeler için uygun değildir. Büyük projelerde, V-süreç modelinin doğru bir şekilde uygulanması ve yönetilmesi daha zor olabilir. Fazlar arasında tekrarlama yoktur. Risk çözümleme aktivitesi yoktur. Bu model belirsizliklerin az olduğu, iş tanımlarının ise belirgin

olduđu projelerde kullanılır. Bu tür projelere örnek olarak Bilgi Teknolojileri için geliştirilen projeler verilebilir.