

Yazılım Yaşam Döngü Modelleri

Günümüzde bir yazılım geliştirmek geleneksel bir ürün geliştirmekten daha farklı ve komplikedir. Bu yüzden yazılım geliştirirken yazılım yaşam döngü modelleri kullanılır. Birçok modelin farklı yaklaşımları vardır fakat genellikle şu adımları içerir: Planlama, analiz, tasarım, kodlama, test etme ve bakım. Planlama aşamasında yazılımın gereksinimleri belirlenir ve bu gereksinim doğrultusunda hedefler, bütçe ve bunun gibi şeyler belirlenir. Analiz aşamasında yazılımın kullanacağı teknolojiler belirlenir. Örneğin bir web uygulaması için uygulamanın single page (tek sayfa) olması mı yoksa multipage (çok sayfalı) olması mı, kullanılacak kütüphaneler(paketler) veya kullanılacak veri tabanı, nerede host edileceği gibi. Kodlama aşamasında bu gereksinim ve tasarımlar kullanılarak yazılımın kodu yazılır. Test etme aşamasında yazılım test edilir. Bakım aşamasında ise yeni güncellemeler sayesinde yazılım geliştirilmeye devam edilir. Bu yazımda sizlere Scrum, kodla ve düzelt, çağlayan ve arıtsal geliştirme modeli ile ilgili bilgi vereceğim ve bu modellerin karşılaştırması ve neden popüler olduğuyla ilgili de bilgilendirmeler yapacağım.

Kodla ve Düzelt

Aslında isminden de anlaşıldığı gibi basit ve kolay uygulanabilir bir yöntemdir. Fakat bakımı çok maliyetlidir. Dokümantasyon yapılmaz. Öncelikle ilk versiyon kodlanır ve müşterinin gereksinimleri karşılanana kadar düzenlenir. Daha sonra teslim edilir ve teslim sonrası oluşan hata veya yeni gereksinimler için bakımı yapılır. Fakat dokümantasyon olmadığı için maliyetlidir. Özellikle tecrübesiz şekilde yazılan kod daha sonra birçok sıkıntıya sebep olabilir. Test yoktur. Kodlamaya başlamadan önce tasarım analiz gibi adımları bulunmadığından gelişigüzel bir iş yapılır. Büyük projeler için çok kullanışsız ve yanlış bir yöntemdir. İki aşamadan oluşur. Basitliği nedeniyle küçük deneyimsiz ekipler tarafından kullanılır. Küçük işlevli 100 200 satırdan oluşan hobi gibi projelerde kullanılabilir.

Çağlayan Yazılım Geliştirme Modeli

Çağlayan yazılım geliştirme modeli günümüzde kullanımı azalsa bile daha öncesinde sıkça kullanılmıştır. Geleneksel yazılım geliştirme modeli olarak da bilinmektedir. Emeklilik aşaması da bulunur. Daha çok iyi tanımlanmış gereksinimleri ve müşteri istekleri tam olan ve üretimi az zaman gerektiren projeler için daha uygundur. Aşamalar arası en az 1 tekrar olur. Bu modelde projenin adımları belli olup sırayla tamamlandığı (biri bitmeden bir sonrakine geçilmez) bir yazılım geliştirme modelidir. Her aşamadan sonra dokümantasyon ve test yapılır. Yapısı çağlayana benzediği için de adına çağlayan modeli denmiştir. Çağlayan yazılım geliştirme modeli 5 temel adımdan oluşur. Bu adımlar: İhtiyaç analizi, tasarım, geliştirme, test ve bakım olarak sıralanabilir. Bir adım bitmeden diğer adıma geçilmez ve adımlar arası atlama yapılmaz. Her adım arasında güçlü bir dokümantasyon bulunur ve testler gerçekleştirilir ve eğer dokümantasyon ve test gerçekleştirilmediyse bu adım tamamlanmış sayılmaz. Bu yüzden bakımı daha kolaydır. İhtiyaç analizi çağlayan yazılım geliştirme modelinin en önemli ve en dikkat edilerek yapılması gereken aşamasıdır. Analiz aşamasında müşteri ile beraber uygulamanın ihtiyaçları tanımlanır. Ürün müşterinin istekleri doğrultusunda şekillenir. Geliştirme başlamadan önce müşterinin istekleri raporlanır. Tasarım aşamasına geçilmeden önce analiz aşamasının doğru bir şekilde yapılması çok önemlidir. Çünkü yanlış bir tasarım sonucu

aşamaların tekrar etmesi gerekebilir. Bu yüzden en önemli aşama analiz aşamasıdır. Tasarım aşamasında kullanılacak teknolojiler projenin işleyişi ile ilgili detaylı bir tasarım oluşturulur ve dokümantasyon ve testler sonrası sonraki aşamaya geçilir. Geliştirme aşamasında isminden de anlaşıldığı gibi tasarımda ortaya çıkan tasarım geliştirilir. Daha sonra uygulama(test) aşamasına geçilir. Bu aşamada ürün testlere tabi tutularak gereksinimleri karşılayıp karşılayamadığı kontrol edilir. Kullanıcı testleri yapılır. Bakım aşaması yazılımın bakımı yeni güncelleştirmelere yapılır. Ortaya çıkan hatalar ve sorunlar düzeltilir, yamanır. Çağlayan modelinin birçok avantajı vardır. Kontrollü bir şekilde ilerlediğinden geliştirme aşamasında bir hata çıktığında veya tasarımsal bir hataya rastlandığında çok daha hızlı bir şekilde yamandır. Her adım arası dokümantasyon ve testler yapıldığından çok kontrollü bir yazılım geliştirme yaşam döngüsü olduğu söylenebilir. Projenin adımlarının planlanması daha karmaşık modellere göre çok daha kolaydır. Tüm aşamalar önceden belirlenir. Ve sırasıyla en az bir tekrar olacak şekilde gerçekleştirilir. Bu nedenle planlanması çok daha kolay bir yazılım geliştirme modelidir. Avantajlarına ek olarak birçok dezavantajı da bulunmaktadır. Analiz ve tasarım aşamasında yapılan hatalar çok daha maliyetli sonuçlara yol açar. Müşterinin analiz aşamasında kendini yanlış anlatması sonucu ürünü ancak test aşamasında görebileceği için birçok tekrara ve maliyete yol açabilir. Bu yüzden gereksinimler iyi anlaşılmalıdır. Yazılım ekiplerinde bir an önce kod yazma hevesi olduğu için analiz tasarım gibi adımlara daha az önem verilmektedir. Bunun sonucunda maliyetler çok ağır olabilir. Kullanıcı modelin içinde yer almadığından sadece yazılımın sonucunda geri dönüş yapılabilir bunun sonucu olarak da birçok aşamada tekrara düşülebilir. Her aşamada dokümantasyon ve test yapıldığından maliyeti yüksektir. Günümüzde de kullanılabilirliğini yitirmekte olan çağlayan modeli yine de gayet kullanılabilir bir modeldir. Fakat en önemli aşaması olan analiz ve tasarım aşamalarına çok dikkat edilmelidir.

V Süreç Modeli

Günümüzde de sıkça kullanılan bir model olan V süreç modeli sürecin V harfine benzemesinden ötürü v süreç modeli olarak adlandırılmıştır. Bu modelde kullanıcının projeye katkısını arttırmak ve yazılım geliştirme sürecinin eksiksiz bir şekilde takip edilerek kontrollü bir şekilde ürün üretmek amaçlanmaktadır. Üç modelde tanımlanır. Bunlar: Kullanıcı modeli, mimari model, gerçekleştirim modelidir. Öncelikle ürün üretilir daha sonra testler yapılır. Bu modelde her adımın sonucu bir sonraki adımda kullanılır. Kullanımı kolay ve kalite kontrolünün iyi sağlandığı bir model olduğu için büyük ve karmaşık projelerde çokça tercih edilmektedir. Sekiz aşamadan ve iki evreden oluşmaktadır. Öncelikle gereksinim analizi yapılır. Bu evrede gereksinimler kullanıcından toplanır ve dokümanite edilir. Kullanıcı kabul testleri de bu aşamada tasarlanır. Daha sonra sistem tasarımı aşamasında geçilir. Bu aşamada sistem mühendisleri gereksinimlerden yola çıkarak sistem tasarlanır ve dokümantasyon yapılır. Sistem testi için de bu aşamada dokümantasyon hazırlanır. Mimari tasarım aşamasında ise daha çok teknik tasarım yapılır. Hangi veri tabanının kullanılacağı gibi birçok tasarım bu aşamada yapılır. Daha sonra modül tasarımı gelir. Modül tasarımı düşük seviyeli tasarım olarak da adlandırılabilir. Burada veri tabanı tabloları sözde kod ve akış şemaları ile ürün tasarlanır. Daha sonra bu tasarım aşamaları kodlamaya başlanılır. Daha sonra v sürecin diğer tarafı olan sınaama tarafına geçinilir. Öncelikle birim testleri(unit testing) yapılır. Daha sonra bir üst olan alt sistem testleri ve daha sonra sistem testleri yapılır. En sonunda ise oluşan sisteme

kullanıcı testleri yapılır. Bu testlerin de sonucunda ürün ortaya çıkar. V süreç modelinin birçok avantajı bulunmaktadır. Kullanıcının projeye katkısı çağlayan modeline göre daha fazladır. Test ve verifikasyonlar erken aşamada tasarlandığından tüm ürünlere uygulanır. V süreç modelinin dezavantajlarına ise fazlar arasında tekrarlama olamaması ve risk çözümleme aktiviteleri olamaması gösterilebilir. Testler sırasında da gereksinimlerin farklı gösterebilir. V süreç modeli çağlayan yazılım geliştirme modeline göre testi çok daha ön planda tutar ve her aşamanın test edilmesi sağlanır. Çağlayan modelinde ise test ayrı bir aşama olarak tutulmaktadır. Çağlayan modeli v modeline göre çok daha doğrusal ilerleyen bir yazılım geliştirme modelidir. İkisinin de uygulanması kolaydır ve büyük karmaşık projelerde kullanılabilir. Dokümantasyon olduğundan bakımları kodla ve düzelt modeline göre maliyetsizdir.

SCRUM

Günümüzde de pek çok büyük firmanın kullanmakta olduğu, karmaşık ve büyük takımlar için çok kullanışlı olan SCRUM modeli 1986 yılında Ken Schwaber ve Jeff Sutherland tarafından tasarlanmıştır. Scrum çevik yazılım geliştirme modellerinden biridir. Bu model özellikle yazılım geliştirme alanında kullanılır. Fakat başka alanlarda da kullanılmaktadır. Karmaşıklığı azaltmak için üç ilke kullanılır. Bu ilkeler şeffaflık, denetleme ve uyumlama. Şeffaflık projedeki ilerlemeler ve sorunlar günlük olarak herkese açık bir şekilde günlük olarak tutulur ve kaydedilir. Bu sayede ilerde karşılaşılan bir sorun daha önce de karşılaşıldıysa çözümü kolaylıkla bulunur. Gözlem aşamasında ise yapılan ilerlemeler gözlenir ve hedeften sapmalar tespit edilir. Uyumlama ise gözlem sonucu elde edilen bilgilerin hayata geçirilmesidir. Üç ana rolü bulunur. Bu roller ürün sahibi, scrum yöneticisi, scrum takımıdır. Scrum takımı ürünü oluşturan ekiptir. Kendi içinde bir yöneticiye ihtiyaç duymaz. Ekip içerisinde çapraz görev dağılımı yapılabilir. Bu sayede ürün hakkında tüm scrum takımı bilgi sahibi olmuş olur. Bu sayede risk de azaltılır. Scrum yöneticisi scrum yapmayı iyi bilen, scrum kurallarını bilen ve uygulayan insanlar olmalıdır. Scrum yöneticisinin bir bakıma scrum takımını koordine eden ve oluşabilecek kaosu engelleyen kişidir. Ürün sahibi ise ürünün gereksinimlerini bildiren öncelikleri belirleyen alınan geri talepleri bildiren kişidir. Ürün sahibi tamamen kendi kafasına uygun bir şekilde geliştirme ekibini yönetebilecek bir yetkiye sahip değildir. Scrum modelinde yazılım geliştirme süreci müşteri gereksinimleri alındıktan sonra iki yada dört haftalık sprint adı verilen dönemlere bölünerek geliştirilir. Her sprint sonrası yazılımın bir bölümü daha tamamlanır. Her gün günlük toplantılar yapılarak karşılaşılan sorunlar ve projedeki ilerleme günlük olarak kaydedilir. Bu toplantılar genellikle ayakta ve 30 dakika sürecek şekilde planlanır. Sprint öncesi sprintin gereksinim listesi, başarılı olması için dağıtım gereksinimleri, takımların belirlenmesi, risk değerlendirmesi gibi sprint planlanır. Sprint süresince Sprint kalan zaman grafiği (Burndown chart) ile sprint ne kadar ilerlendiği kaydedilir. Kalan zaman grafiği ile maksimum şeffaflık elde edilmeye çalışılır. Sprint incelemesi, Sprint sonunda gerçekleşir ve geliştirme ekibi, tamamlanan işleri gösterir. Scrum, yazılım geliştirme projelerinde kullanılan birçok yöntem arasında popülerdir. Scrum'un avantajları arasında, proje ekibinin daha iyi bir şekilde takip edilmesi, esnekliğin artması, işlerin daha iyi bir şekilde önceliklendirilmesi, müşteri geri bildirimlerinin daha hızlı alınması ve ekip üyelerinin daha yüksek bir motivasyon seviyesine sahip olmaları bulunur. Bununla birlikte, Scrum modelinin bazı dezavantajları da vardır. Örneğin, proje süresi boyunca değişen

gereksinimlerin yönetilmesi zor olabilir. Ayrıca, Scrum yönteminin uygulanması zor olabilir ve ekip üyeleri arasında iş birliği eksikliği yaşanabilir. Günümüzde de Google gibi dev teknoloji firmalarında da dahill olmak üzere birçok firma Scrum modelini kullanmaktadır kullanılmaktadır. Büyük takımlar için idealdir. İş birliği ve takım çalışması ön plandadır. Scrum modeli çağlayan ve v modeline göre çok daha esnetir. Doğrusal bir akış yerine geribildirimlerde önem verilir. Scrum işi sprintlere bölüp daha optimize bir şekilde projeyi ilerletir. Çağlayan modeli ise işi tek bir bütün olarak alarak sadece akışı optimize etmeye çalışır. Fakat scrum daha çok büyük ve karmaşık projeler için idealdir. Çağlayan modelinin aksinde ürün parça parça oluşur ve hatalardan geri dönmek daha kolaydır. V modeli geliştirme süresinde gereksinimlerin değişmesini kontrol etmek zor olduğundan scrum v modeline göre çok daha esnek bir yapıya sahiptir. Bu yüzden scrum karmaşık projelerde çok daha fazla tercih edilebilir. V modeli ise testin daha çok gerektiği alanlarda kullanılabilir. Scrum günümüzde çok popülerdir. Bunun bazı sebepleri şunlardır. Kolay implemente edilebilir. Esnek ve adaptasyon çok daha kolaydır. Ekip ağırlıklı bir modeldir. Her sprintde testler yapıldığından ve ortaya çıkan ürün parçası kontrol edildiğinden yüksek kaliteli ürün elde edilir. Günümüzde de pek çok firma tarafından kullanılmaktadır.

Okuduğunuz için teşekkür ederim.

Aksel Saatçı

[www.github.com/akselsaatci](https://github.com/akselsaatci)

<https://medium.com/@aksel.saatci1>

<https://www.linkedin.com/in/aksel-saatc%C4%B1-2855a8244/>