

Bazı Yazılım Yaşam Döngü Modelleri Hakkında Bilgiler

Ecem Şimşek, 220601071

Özet

Yazılım yaşam döngüsü modelleri diğer bir deyişle SDLC, yazılım geliştirme sürecinin farklı aşamalarının belirlenmesi ve yönetilmesi için kullanılan yöntemlerdir. Bu yazıda bazı SDLC modelleri incelenmiştir. Bunlar prototipleme, hızlı uygulama geliştirme modeli, çevik yazılım geliştirme modeli ve scrum metodu ile waterfall modelidir. Modellerin tanıtımları yapılacak, avantajları ve dezavantajları hakkında bilgi verilecek ve bazı modeller birbiriyle karşılaştırılacaktır.

Anahtar Kelimeler: scrum, RAD, çevik yazılım, prototipleme, waterfall model, proje yönetimi

Giriş

Yazılım yaşam döngüsü modelleri (Software Development Life Cycle), yazılımın tasarımından başlayarak geliştirilmesi, test edilmesi, dağıtılması ve bakımı aşamalarını kapsayan bir dizi işlemi ifade eder. Bu modellerin amaçları, yazılımın kalitesini arttırmak, projelerin zamanında ve bütçeye uygun olarak tamamlanmasını sağlamak ve müşteri ihtiyaçlarını karşılamaktır. Genellikle SDLC kabaca şu adımları içerir: İlk olarak ihtiyaç analizi yapılır. Bu aşamada, müşterinin ihtiyaçları belirlenir ve yazılımın hangi özellikleri içermesi gerektiği belirlenir. Proje planlanır ve gereksinimler belirlenir. Sonra tasarım aşamasına geçilir. İhtiyaç analizi tamamlandıktan sonra, yazılımın tasarımı yapılır. Bu aşamada, kullanılacak teknolojiler belirlenir ve yazılımın mimarisi tasarlanır. Sonraki aşama geliştirme aşamasıdır. Tasarım aşamasından sonra, yazılımın kodlama işlemi gerçekleştirilir. Bu aşamada, yazılımın tüm işlevleri ve özellikleri geliştirilir. Ardından test aşamasına geçilir. Yazılım geliştirme tamamlandıktan sonra, yazılım test edilir. Bu aşamada, yazılımın doğru çalışıp çalışmadığı kontrol edilir ve hatalar düzeltilir. Test aşamasından sonra, yazılımın canlı ortama uygulanması gerçekleştirilir. Bu uygulama aşamasında, yazılım kullanıma hazır hale getirilir. Yazılımın canlı ortamda kullanımı sırasında meydana gelen hataların düzeltilmesi, güncellemelerin yapılması ve yazılımın sürekli olarak güncel tutulması için de daha sonra bakım işlemleri gerçekleştirilir. SDLC aşamaları kabaca bu şekilde olsa da her model kendi içinde farklılıklar barındırır. Modellerin birbirlerine göre avantaj ve dezavantajları da vardır.

PROTOTİPLEME MODELİ

Prototipleme modeli, en popüler SDLC modellerinden biridir. Bu model, müşteriler önceden proje gereksinimlerini tam olarak bilmediğinde oldukça kullanışlıdır. Bu modelde önce son ürünün bir prototipi geliştirilir, test edilir ve saklanır. Bu durum son aşamada ortaya çıkacak ürünün modeli müşterinin isteklerine uyuncaya dek devam eder. Bu modelde müşteri geri bildirimleri oldukça önemlidir. Müşterinin istekleri doğrultusunda bir prototip oluşturulur ve müşteriye sunulur. Müşteri beğenmediği veya beğendiği yönleri geri bildirir ve prototip bu bilgiler ışığında güncellenir. Bazen prototip hiç beğenilmez ve en baştan yeni bir model oluşturmak gerekebilir. Bu modelde oluşturulan prototip gerçek programın çok kaba bir versiyonudur. Gerçek yazılıma kıyasla genellikle sınırlı işlevi, düşük güvenilirliği ve verimsiz performansı vardır. Yalnızca üründen beklenenler hakkında fikir vermek amacıyla kullanılır.

Prototipleme Modelinin Adımları

Prototipleme modeli Őu aŐamaları sırasıyla takip eder: Gereksinim toplama ve analiz, Hızlı karar ve planlama, Prototip oluŐturma, Kullanıcı deęerlendirmesi, Prototip iyileŐtirme, Son ürün.

Prototiplemenin Bazı Avantajları

Sistemdeki hatalar daha erken tespit edilir. Bakım maaliyetleri azaltılmıŐ olur. Erken ürün pazarlaması dolayısıyla müşteri memnuniyeti artar. OluŐturulan prototip gelecekte baŐka projeler için de kullanılabilir. Yazılımın son hali müşterinin isteklerine daha uyumlu olur.

Prototiplemenin Bazı Dezavantajları

Prototip müşteriye her sunulduęunda müşteri çok fazla deęiŐim talebinde bulunabilir. Müşteri ilk prototipi beęenmezse hayal kırıklıęına uğrayıp ilgisini kaybedebilir. Müşteri prototipten son ürün performansı bekleyebilir ve prototipi gördükten sonra son ürünün yakın tarihte teslim edilmesinde ısrar edebilir. Prototip sürekli beęenilmezse para ve zaman kaybedilir. Acele edildięi için dokümantasyonu olmayan hızlı ve kirli prototipler üretilebilir. Düzeltme adımı atlanırsa düşük performans gösterebilir. Prototipleme modeli müşteri odaklı ve yenilikçi projelerde tercih edilebilir. Ayrıca karmaŐık sistemlerin gereksinimlerini daha iyi anlamak için kullanılabilir. Prototipler tamamlanmıŐ sistemi tasarlamadan önce sistemin gereksinimlerini ve iŐlevlerini anlamak için kullanılabilir.

HIZLI UYGULAMA GELİŐTİRME MODELİ (RAD)

Bu modelde proje küçük modüllere ayrılarak çeŐitli ekiplere dağıtılır. Modüller oluŐturulduktan sonra nihai ürünü oluŐturmak için birleŐtirilir. Her modülün geliŐtirilmesinde analiz, tasarım, kodlama ve test etme gibi temel adımlar kullanılır. Bu modelin çarpıcı özellięi projenin kısa sürede teslim edilmesidir. Ürünün üretiminde JAVA, C, C++, Visual BASIC, XML gibi güçlü geliŐtirme araçları kullanılır.

RAD Modelinin Adımları

İlk aŐama gereksinimlerin belirlenmesidir. Bu aŐamada projeyi geliŐtirirken ortaya çıkacak ihtiyaçları, gereksinimleri, zorlukları ve hedefleri içeren bir plan oluŐturulur. İkinci aŐama kullanıcı katılımı aŐamasıdır. Bu aŐamada kullanıcı geri bildirimleri ile geliŐtiriciler bir prototip oluŐtururlar. İlk adımda toplanan verilerin kullanıcının geri bildirimleri ışıęında yeniden deęerlendirilmesi yapılır. Üçüncü aŐama inŐaat aŐamasıdır. Prototip iyileŐtirilir ve teslim edilir. Bu aŐama kullanıcı sonucundan memnun oluncaya kadar tekrarlanır. En son aŐama sonlandırmadır. Bu aŐamada ayrı ekipler tarafından geliŐtirilen baęımsız modüller detaylı Őekilde test edilir. Ürün kullanıcıya sunulur ve onun da test etmesi istenir.

RAD Modelinin Avantajları

GeliŐtirme için harcanan süre azdır. Ürün kısa sürede teslim edilir. Daha az developer ve güçlü geliŐtirme araçları kullanıldıęı için maliyet azdır. Kısa yenileme süre aralıklarına sahiptir. Dolayısıyla deęiŐen gereksinimleri karŐılamak daha kolaydır. Esnektir, deęiŐikliklere uyarlanabilir. Genel proje riskini azaltır.

RAD Modelinin Dezavantajları

Güçlü geliştirme araçları iyi eğitilmiş personel gerektirir. Küçük projeler için kullanılamaz. Büyük takımlar gerektirir. RAD modeli gereksinimler bilindiğinde ve 2-3 ay gibi kısa süreli bir sistem üretilmesi gerektiğinde kullanılabilir. Ayrıca bu model halihazırda var olan bir sistemin bileşenleri başka ve yeni bir sistem geliştirmede kullanılabilir haldeyse de kullanılabilir.

RAD ve Prototipleme Arasındaki Farklar

Prototiplemede projeler kısa sürede bitmezken RAD modelinde kısa sürede biter. Prototip model yüksek riskli projeler için uygunken RAD teknik riskler yüksek olan projeler için uygun değildir. Prototipleme risk analizi üzerinde fazla durmazken RAD risk analizine önem verir. Prototiplemede prototipin oluşturulması için zaman gerekli olduğundan hızlı ilk incelemeler yapılamaz. RAD ise buna olanak tanır. RAD otomatik kod yazılımına olanak tanıdığı için minimum kod yazımı ile sonuçlanırken, prototipleme otomatik kod yazılımına uygun değildir. Prototiplemede büyük takımlar olmadan da proje çıkarılabilirken RAD için büyük takımlar gerekir.

SCRUM

Scrum çevik proje yönetimi stratejisine ait bir yaklaşım ve metottur. Scrumu daha iyi anlamak için öncelikle çevik yazılım yaşam döngüsüne bakmak daha sağlıklı olacaktır.

Çevik Proje Yönetimi (Agile) Nedir?

Yazılım ürününün hızlı bir şekilde teslim edilmesiyle süreç uygunluğuna ve müşteri memnuniyetine odaklanan yinelemeli ve artımlı süreç modellerinin bir kombinasyonudur. Agile modelde ürün küçük artımlı yapılara bölünür. Bu yapılar iterasyonlar halinde oluşturulur ve yapı özellikleri açısından artımlıdır. Her iterasyon bir ila üç hafta arasında sürer. Her iterasyonun ardından çalışan bir yazılım derlenir. Her derleme özellikler açısından artımsal olarak geliştirilir. Son derleme müşteri tarafından talep edilen tüm özelliklere sahiptir. Çök düşünce süreci, yazılım geliştirmenin başlarında başladı ve esnekliği ve uyulanabilirliği nedeniyle zamanla popüler olmaya başladı.

Çevik Yöntem ile RAD Arasındaki Farklar

İkisi çok benzer gibi görünse de aralarında bazı farklılık vardır. RAD hızlı prototiplemeye dayanırken çevik yöntemde prototip kullanımı yoktur. RAD modelinde uygulama geliştirmenin ilk aşamalarında kaliteye odaklanılmaz Bu nedenle her birini zaman içinde daha kaliteli kodlara dönüştürmeden önce tüm temel özellikleri kabataslak olarak oluşturulur. Agile ile çözüm özelliklerine göre ayrılır. Bu özellikler daha sonra kademeli olarak geliştirilir. Ayrıca çevik yöntem RAD'a göre daha esnek çalışma imkanı sunar ve müşteri ile daha fazla iletişim halindedir. Fakat farklılıklara rağmen bu iki yöntem de hızlı bir şekilde projeyi tamamlama ihtiyacına yöneliktir.

Scrum Nedir?

Scrum, Agile proje yönetim metotlarından biridir. Düzenli planlama ve geri bildirim ile projeyi tamamlamayı hedefler. Karmaşık projeler üstünde çalışırken esnek, verimli ve hızlı olmayı sağlar. Özellikle yazılım geliştirme sürecindeki değişikliklere ve belirsizliklere yanıt verme açısından oldukça yararlıdır. İteratif ve artımlı bir yaklaşımı benimser ve takım üyeleri

sıkı bir iş birliği içinde çalışır. Belirli sürede tamamlanması gereken hedefler belirlenir ve bunlar sprint denen kısa zaman dilimlerine bölünerek üzerinde çalışılır. Scrum üç temel prensip üzerine kurulmuştur. Bunlardan ilki şeffaflıktır. Projenin ilerleyişi, problemler ve gelişmeler herkes tarafından takip edilir şekilde olmalıdır. İkincisi denetlemedir. Projenin ilerleyişi düzenli olarak takip edilir. Bu takip günlük Scrum toplantısını ve sprint gözden geçirme toplantısını da içerir. Sonuncusu adaptasyondur. Proje yapılabilecek değişimlere uyum sağlayabilecek özellikte olmalıdır.

Scrum Aşamaları

Öncelikle projenin ana hedefleri belirlenir. Hedeflere ulaşmak için gereken işler listelenir. Daha sonra sprint planlaması yapılır. Takımlar sprint boyunca yapacakları işleri belirler. Sonra sprint aşamasına geçilir ve ekip üyeleri belirlenen hedefler doğrultusunda çalışmaya koyulurlar. Takım üyeleri her gün bir araya gelerek geliştirdikleri işleri paylaşırlar. Bu toplantılara günlük toplantılar denir. Sprint boyunca takım üyeleri belirlenen işleri tamamlar ve bunları bildirir. Bu şekilde herkesin projedeki ilerlemesi rahatça takip edilebilir. Bu aşamadan sonra bir sprint boyunca gerçekleştirilen işler değerlendirilir ve gelecek sprintler için hedefler belirlenir. Sprint değerlendirme toplantısı adı verilen bir toplantı yapılır. Bu toplantıda takım üyeleri bitirilen işleri ve sprint boyunca öğrendiklerini paylaşırlar. Müşterinin geri bildirimi alınır. Ayrıca ekip sprint boyunca nelerin iyi gittiğini ve nelerin geliştirilmesi gerektiğini belirlemek için bir retrospektif toplantısı yapar. Takım gelecekte yapılacak işlerde sprint sürecini iyileştirmek için yapılabilecek şeyleri masaya yatırır. Tüm bu aşamalar scrumun temel iş akışını oluşturmakla birlikte sürekli bir geliştirme çemberi şeklinde ilerler.

Scrumun Avantajları

Scrum, değişen müşteri gereksinimlerine ve pazar koşullarına hızlı bir şekilde uyum sağlamayı mümkün kılar. Sık sık geri bildirim sağlayarak ve hızlı bir şekilde yeni özellikler ekleyerek müşteri memnuniyetini artırabilir. Scrumda sürekli geri bildirimler ve testler yapılır dolayısıyla olası risk ve hatalar erkenden tespit edilir. Proje küçük parçalara bölünerek ilerletildiği için ürün müşteriye daha hızlı şekilde teslim edilir.

Scrum Dezavantajları

Takım üyeleri kararlı ve işbirlikçi değilse proje başarısız olabilir. Bu modelde çalışanların scrum modelinde tecrübeleri olması önemlidir. Günlük toplantılar ekip üyelerini yorabilir ve üyelerden biri ayrılırsa proje üzerinde büyük bir olumsuzluk yaratır. Çok ~~g~~iş çaplı projelerde scrumun uygulanması zor olabilir. Çünkü büyük ~~ö~~ekte iyi bir eğitim ve hassas bir koordinasyon gerektirir. Scrum büyük projelerde uygulanamaz değildir sadece uygulanması zordur. Sürekli ayarlamalar gerektiren projeler için genellikle iyi bir çözümdür fakat Scrum yaklaşımı öngörülebilir ve iyi tanımlanmış bir plan gerektiren projeler için uygun olmayabilir. Bu tip projelerde hem klasik yöntemin hem de scrumun avantajlarından yararlanabilmek için hibrit bir model kullanılabilir. Projenin sonu belli değildir. Proje müşteri memnun olana kadar devam eder.

WATERFALL MODELİ

Aynı zamanda doğrusal-sıralı yaşam döngüsü modeli olarak da adlandırılır. Türkçeye çağlayan model, şelale model gibi isimlerle geçmiştir. Anlaşılması ve kullanılması çok basittir. Bir şelale modelinde, bir sonraki aşama başlamadan önce her aşamanın tamamlanması

gerekir ve aşamalarda çakışma olmaz. Şelale modeli, yazılım geliştirme için kullanılan en eski SDLC yaklaşımıdır. Bu modelde bir aşamanın çıktısı, sonraki aşamanın girdisidir. Şelale modelinin uygulanabilmesi için kullanılacak teknolojinin dinamik olmaması gerekir. Aynı zamanda gereksinimler çok iyi belirlenmiş şekilde olmalı ve müşteri ne isteyip ne istemediğinin farkında olmalıdır. Bu modelde her aşamada dokümantasyon oluşturulur ve proje bu şekilde ilerler. Kullanıcı başlangıç safhasında katılım gösterebilir, bundan sonraki aşamalarda kodlama aşaması bitene kadar katılım göstermez.

Waterfall Modelin Aşamaları

İlk aşama analiz aşamasıdır. Geliştirilecek sistemin tüm olası gereksinimleri tespit edilir ve belgelendirilir. Sonraki aşama tasarım aşamasıdır. Gereksinimlere uygun olarak sistem tasarımı yapılır. Üçüncü aşama kodlama aşamasıdır. Sistem tasarımından gelen girdilere uygun olarak kodlama yapılır. Daha sonra entegrasyon ve test aşamasına geçilir. Ortaya konan çalışma test edildikten sonra sisteme entegre edilir. Entegrasyon tamamlandıktan sonra da herhangi bir arıza oluştu mu diye kontrol etmek için tekrar test yapılır.

Waterfall Modelin Avantajları

Model basittir, anlaşılması ve kullanılması kolaydır. Aşamalar tek tek işlenir ve tamamlanır. Küçük projeler için faydalıdır. Görevleri düzenlemek kolaydır. Hem süreç hem de sonuç kapsamlı şekilde kaydedilir.

Waterfall Modelin Dezavantajları

Bu modelde aşamalar birbirlerini sıralı şekilde takip eder. Fakat bazı projelerde bu durum mümkün değildir, önceki aşamaya veya aşamalara geri dönmek zorunda kalınabilir. Müşteriler gereksinimleri detaylı şekilde ifade etmelidir. Eğer müşteri proje ilerlerken isteklerinde değişime giderse bu durum süreyi ve maaliyeti arttırır. Çünkü bu model sonradan yapılacak değişikliklere uyum sağlama konusunda iyi değildir. Değişiklik yapabilmek için bazı aşamaların sil baştan yapılması gerekebilir. Başlangıçta yapılan hataların tespiti zordur ve ilerleyen safhalarda tespit edilirse düzeltilmesi yüksek maliyetlidir.

Waterfall Model ve RAD Arasındaki Farklar

Waterfall model yüksek risklidir, RAD daha düşük riske sahiptir. Waterfall modelde amaç yüksek güvenceli yazılım geliştirmekken RAD'da amaç hızlı şekilde yazılım geliştirmektir. Yapılacak değişiklik en waterfall modelde önceki aşamalarda olmalı yoksa maliyet çok artar. Fakat RAD modelde değişiklik her aşamada yapılabilir. Waterfall modelde ürün döngünün son aşamasında teslim edilirken RAD'da ürün daha erken teslim edilir ve güncelleme yapmak için geri bildirim ister. Waterfall modelde müşterinin yönetim üzerindeki kontrolü RAD modele göre daha azdır.

Waterfall Model ve Scrum Arasındaki Farklar

Scrum her aşamada müşteriyi dahil ederken waterfall modelde müşteri katılımı sadece projenin başlarında ve sonlarındadır. Scrumda değişiklikler her aşamada yapılabilirken, waterfall modelde değişikliklerin yapılması için bazı aşamalar sil baştan yapılmalıdır. Waterfall modelde bir önceki aşamaya dönmek neredeyse imkansızdır. Sürecin başında tespit edilemeyen bir hata son aşamada tespit edilirse projeye yeniden başlamak gerekir. Scrum daha

esnektir, birden fazla eş zamanlı döngülerden yararlanır. Dolayısıyla hata tespit edildiğinde düzeltmek daha kolaydır. Scrum daha hızlı şekilde projeyi tamamlamayı öngörürken waterfall modelde bu süreç daha uzun sürebilir. Scrumda sürecin ilerlemesi daha karmaşık yapıdadır. Waterfall modelde ise daha basit ve doğrusal olarak ilerler.

Sonuç

Yapılacak her projenin ve ekibin kullanabileceği metotlar birbirinden farklıdır. Yazılım sürecinde ekibe ve projeye uygun modeli seçmek oldukça önemlidir. Daha hızlı ve dinamik bir proje ortaya çıkarmak için RAD, agile gibi modeller tercih edilebilirken; daha stabil sonuçlar için waterfall model tercih edilebilir. Müşterinin sunacağı gereksinim bilgileri de model seçiminde önemli bir etkidir. Ne istediğinden tam olarak emin olmayan müşterilerin projeleri için prototipleme kullanılan modeller seçmek daha etkili olabilir. Model seçiminde bütçe planlamasının ve takımın büyüklüğü ile tecrübesine de dikkat etmek gerekir.

Kaynakça

Agile vs Waterfall vs Scrum: Similarities and Differences. (tarih yok). TCGen:

<https://www.tcgen.com/agile/agile-vs-waterfall-vs-scrum/> adresinden alındı

Difference between Prototype Model and RAD Model. (2022, Ağustos 25). geekforgeeks:

<https://www.geeksforgeeks.org/difference-between-prototype-model-and-rad-model/> adresinden alındı

Nanda, V. (2021, Kasım 30). *Comparison between Agile Model and other models.*

tutorialspoint: <https://www.tutorialspoint.com/comparison-between-agile-model-and-other-models> adresinden alındı

Yazılım Süreç Yönetim Modelleri ve Karşılaştırılması. (2018, Eylül 3). Fikir Jeneratörü:

<https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/> adresinden alındı

GitHub Hesabım: <https://github.com/simsekecem>

LinkedIn Hesabım: <https://www.linkedin.com/in/ecem-%C5%9Fim%C5%9Fek-90b883169/>

Medium Hesabım: <https://medium.com/@simsekecem>