

Ad Soyad: Cesur Akçay

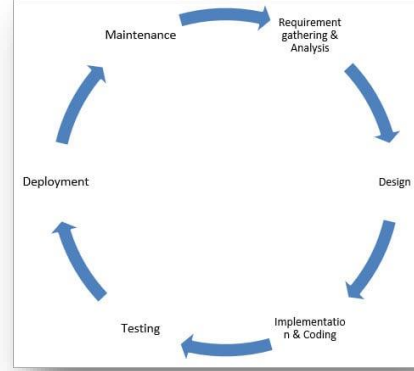
No:220601082

Yazılım Geliştirme Yaşam Döngüsü (SDLC) Nedir?

Yazılım Geliştirme Yaşam Döngüsü(SDLC), her aşamada yazılım geliştirme sürecinde yer alan oluşturma, dağıtma ve bakım adımlarını tanımlayan yapıdır. Bu adımların her biri için ayrıntılı plan kapsar. Yazılım geliştirme yaşam döngüsü sürecine bağlı kalmak, yazılımın sistematik ve disiplinli bir şekilde geliştirilmesini sağlar.

Yazılım Geliştirme Yaşam Döngüsü Aşamaları:

- Gereksinimleri belirleme ve analiz
- Tasarım
- Kodlama
- Test
- Dağıtım
- Bakım



Bir yazılım yaşam döngüsü modeli, yazılım geliştirme döngüsünün betimsel bir temsildir. Yazılım geliştirme yaşam döngüsü(SDLC) modelleri farklı bir yaklaşıma sahip olabilir, ancak temel aşamalar ve faaliyetler tüm modeller için aynı kalır. Yazılım sektöründe yaygın olarak kullanılan birkaç tane SDLC modelleri vardır. Bunlardan bazılarını örnek olarak verelim.

Çağlayan(Waterfall) Modeli

SDLC içinde kullanılmış ilk modeldir. Bu modelde, bir aşamanın çıktısı bir sonraki aşamanın girdisidir. İşlemler aşama aşama yapılır. Bir sonraki aşamanın geliştirilmeye başlanması, yalnızca bir önceki aşama tamamlandığında başlar.

- İlk olarak, Gereksinim toplama ve analizi yapılır. Gereksinim toplama aşaması bittikten sonra yalnızca sistem tasarımı başlayabilir.
- Burada oluşturulan SRS belgesi(yazılım ürününün kapsamlı bir tanımı olan döküman) Gereksinim aşamasının çıktısı olup, sistem tasarımı aşamasına girdi görevi görür.
- Sistem tasarımı aşamasında bir sonraki aşama için girdi görevi olan belgeler, yani uygulama ve kodlama yapılır.
- Uygulama aşamasında kodlama yapılır ve geliştirilen yazılım bir sonraki aşama olan test için girdi oluşturur.
- Test aşamasında, yazılımdaki kusurları tespit etmek için geliştirilen kod kapsamlı bir şekilde test edilir. Kusurlar, kusur izleme aracına kaydedilir ve düzeltildikten sonra yeniden test edilir. Hata günlüğü, Yeniden Test, Regresyon testi, yazılım canlı duruma geçene kadar devam eder.

- Dağıtım aşamasında, müşteri tarafından onay verildikten sonra geliştirilen kod üretime taşınır.
- Bakım aşamasında üretim ortamındaki herhangi bir sorun geliştiriciler tarafından çözülür.

Çağlayan Modelinin Avantajları

- Tüm aşamaların adım adım yapıldığı, anlaşılması oldukça kolay olan bir modeldir.
- Her aşamanın çıktıları iyi tanımlanmıştır ve bu, hiçbir karmaşıklığa yol açmaz ve projeyi kolayca yönetilebilir hale getirir. İş dağılımını yapmak kolaydır.

Çağlayan Modelinin Dezavantajları

- Gereksinim tanımlamaları çok net yapılmadığından, yanlışları düzeltme ve eksikleri giderme maliyeti yüksektir.
- Yazılımın kullanıcıya ulaşma zamanı uzundur ve kullanıcı sürecin içerisinde yer almaz, yazılım tamamlandıktan sonra geri dönüşleri arttırabilir.
- Zaman alıcıdır ve kısa süreli projelerde kullanılamaz, çünkü bu modelde devam eden aşama tamamlanana kadar yeni bir aşama başlatılamaz.

V Süreç(V-Shaped) Modeli

V- Modeli, üretim ve doğrulama Modeli olarak da bilinir. Bu modelde üretim ve doğrulama el ele gider, yani geliştirme ve test paralel gider. V modeli ve şelale modeli, test planlama ve testin V-Model'de erken bir aşamada başlaması dışında aynıdır. Kullanıcı modeli, Mimari Model, Gerçekleştirim Modeli olarak üç aşaması vardır. Kullanıcı modeli, gereksinim analizi ve bitmiş sistem kısımlarından oluşur. Geliştirme sürecinin kullanıcı ile olan ilişkileri tanımlanmakta ve sistemin nasıl kabul edileceğine ilişkin sınama belirtileri ve planları ortaya çıkarılmaktadır. Mimari model sistem tasarımı ve oluşacak alt sistem ile tüm sistemin test işlemlerine ilişkin işlevleri barındırır. Gerçekleştirim modeli ise kodlamadan oluşur. Yazılım modüllerinin kodlanması ve test edilmesine ilişkin fonksiyonları barındırır.

V-Süreç Modelinin Avantajları

- Model, kullanıcının projeye katkısını arttırmaktadır.
- Basit ve anlaşılması kolay bir model olup, proje yönetimini ve proje takibinde kolaylık sağlar.
- Yüksek kaliteli ürün sağlayan sistematik ve disiplinli bir modeldir.

V-Süreç Modelinin Dezavantajları

- Risk çözümleme aktiviteleri yoktur.
- İş ve ürün gereksinimleri değişkenlik gösterebilir.
- Daha sonraki aşamada gereksinim değişikliği çok yüksek maliyetli olacaktır

Spiral Modeli

İterasyonlarda sarmal model aşamaları izlenir. Modeldeki döngüler, SDLC sürecinin aşamasını temsil eder, yani en içteki döngü, planlama, risk analizi, geliştirme ve değerlendirmeyi izleyen gereksinim analizidir. Sonraki döngü, tasarım ve ardından uygulama ve ardından testtir. Spiral Model dört adet aşamadan oluşur. Bunlar planlama, risk analizi, mühendislik ve değerlendirmedir.

Planlama: Planlama aşaması, gerekli tüm bilgilerin müşteriden toplandığı ve belgelendiği gereksinim toplamayı içerir. Bir sonraki aşama için yazılım gereksinim belirtim belgesinin oluşturulduğu aşamadır.

Risk Analizi: Bu aşamada, ilgili riskler değerlendirilerek en iyi çözüm seçilir ve bir prototip oluşturularak analizin yapıldığı aşamadır. Örneğin, uzak bir veritabanından verilere erişmenin içerdiği risk, veri erişim hızının çok yavaş olması olabilir. Risk, veri erişim alt sisteminin bir prototipi oluşturularak çözülebilir.

Mühendislik: Risk analizi yapıldıktan sonra kodlama ve test yapılan aşamadır.

Değerlendirme: Müşteri, geliştirilen sistemi değerlendirir ve bir sonraki iterasyonu planlar.

Spiral Model Avantajları

- Risk Analizi, prototip modeller kullanılarak kapsamlı bir şekilde yapılır.
- Kullanıcılar sistemi erken görürler ve model hataları erken gidermeye odaklanır.
- Fonksiyonel herhangi bir geliştirme veya değişiklik bir sonraki iterasyonda yapılabilir.

Spiral Model Dezavantajları

- Nihai ürüne ulaşmak için uzun süreye yol açabilecek çok sayıda iterasyon(yineleme) gerektirebileceğinden maliyet yüksek olabilir.
- Ara adımların fazlalığı nedeniyle çok fazla dökümantasyon gerektirir
- Karmaşıktır, spiral sonsuza gidebilir.

Arttırımsal Geliştirme Süreç Modeli

Bu model, ürünü küçük parçalara böler. Örneğin her bir iterasyonda geliştirilecek özellik belirlenir ve uygulanır. Her iterasyon, gereksinim analizi, tasarım, kodlama ve test Etme aşamalarından geçer. İterasyonlarda ayrıntılı planlama gerekli değildir. Yineleme tamamlandığında, bir ürün doğrulanır ve değerlendirilmesi ve geri bildirimi için müşteriye teslim edilir. Müşterinin geri bildirimi, yeni eklenen özellik ile birlikte bir sonraki yinelemede uygulanır. Yani bir taraftan ürün kullanıldığı gibi diğer taraftan üretim devam eder. Bu nedenle, ürün özellikler açısından artar ve iterasyonlar tamamlandıktan sonra nihai yapı, ürünün tüm özelliklerini tutar.

Arttırımsal Geliştirme Süreç Modeli Avantajları

- Gereksinimdeki herhangi bir değişiklik kolayca yapılabilir ve bir sonraki iterasyonda yeni gereksinimi dahil etme kapsamı olduğu için maliyeti yoktur.
- Risk güncellemelerde analiz edilir ve tanımlanır.
- Üründeki kusurlar erken bir aşamada tespit edilir.
- Ürün daha küçük parçalara bölündüğünden, ürünü yönetmek kolaydır.

- En önemli sistem özellikleri daha fazla test edilme imkanı bulmuş olur ve tüm projenin başarısız olma riskini azaltır.

Arttırımsal Geliştirme Süreç Modeli Dezavantajları

- Bir ürünün gereksinimlerinin eksiksiz anlaşılması, aşamalı olarak yıkmak ve yapmak için gereklidir.
- Deneyimli personel gerektirir.
- Artımları tanımlamak için tüm sistem tanımlanmalıdır.
- Gereksinimler doğru artımalara atanmayabilir.

SCRUM Nedir ?

Scrum, Ken Schwaber ve Jeff Sutherland tarafından geliştirilen ve 1990'lardan beri kullanımda olan çevik bir proje geliştirme modelidir. İş, daha küçük gereksinimlere bölünmüştür ve sıkı sıkıya bağlı ekipler küçük taksitler halinde oluşturulur ve görevlerini teslim eder. Sık sık geri bildirim alınır ve üründe sık sık kısa sürümler şeklinde iyileştirmeler yapılır. Her projeye uygulanabilen çevik ve ilerlemeyi kolaylaştıran bir modeldir. Scrum'ın özellikleri:

Şeffaflık: Ekipler olan bitenin farkındadır ve paylaşmaya, birbirlerine yardım etmeye açıktır. İletişim, günlük toplantılar ve resmi olmayan kişiden kişiye etkileşimler yoluyla serbestçe akar.

Denetleme: Sık iş teftişleri, Scrum'ın başarısının anahtarıdır. Ekipler basit ve güvenilir bir şekilde tanımlanabilir, teşhis edilebilir, sorun giderebilir, düzeltebilir.

Adaptasyon: Scrum, yaptıklarının doğru olduğunu varsaymaz. Takımın gözden geçirip uyum sağladığı planlama, günlük toplantı, gözden geçirme/geriye dönük toplantılar şeklinde periyodik kontrol noktaları vardır.

Scrum takımları genellikle küçüktür (5-9) ve genellikle birbirlerinin işlerini de yapabilirler. Bir Scrum Master, geliştirici, test uzmanı içerirler.

Scrum'da Roller:

Ürün Sahibi: Ürün sahibi işlerin sahibidir. Üründen ve ürünün aldığı şekilden sorumludur. Ürün sürdürmek, genel bir ürün vizyonuna sahip olmak ve ekibin hedeflerini buna doğru yönlendirmek, bir ürün sahibinin birincil sorumluluklarıdır.

Geliştirme Ekibi: Geliştirme ekibinin herhangi bir sınırlı rolü yoktur. İşlevler arası çalışmaları ve hedefe ulaşmak için en iyi yaklaşımı seçmeleri beklenir.

Scrum Ustası: Scrum'ın doğru şekilde uygulandığından emin olmak, scrum ustasının işidir. Scrum ustası aynı zamanda tüm takım için lider olarak da adlandırılır.

Scrum en çok yazılım geliştirme ekipleri tarafından kullanılsa da, ilkeleri ile her türlü ekip çalışmasına uygulanabilir. Scrum'ın bu kadar popüler olmasının nedenlerinden biri de budur. Genellikle çevik bir proje yönetimi çerçevesi olarak düşünülen scrum, ekiplerin işlerini yapılandırmasına ve yönetmesine yardımcı olmak için uyum içinde çalışan bir dizi toplantı, araç ve rolü tanımlar.

Hangi Projede Hangi Modeller Kullanılmalıdır ?

Çağlayan modeli zaman alıcıdır ve kısa süreli projelerde kullanılamaz çünkü bu modelde devam eden aşama tamamlanana kadar yeni bir aşama başlatılamaz. Aynı zamanda bu model gereksinimin dinamik olarak değişebileceği projelerde de kullanılamaz çünkü bu modelin analiz ve gereksinimleri belirleme kısımlarında değişiklik yapmak maliyetli ve zordur. Gereksinimlerin belirlenmesi ve analizin net olduğu basit projelerde çağlayan modeli kullanılabilir.

V modelinde de aynı şekilde ilerleyen aşamalarda değişiklik yapmanın maliyeti zor olacağından devam eden projeler için iyi bir model değildir. V modeli yaklaşımı, gereksinimin tanımlandığı ve erken aşamada değişmediği daha küçük projeler için iyidir.

Büyük projelerde spiral model kullanılabilir çünkü spiral model sadece büyük projeler için uygundur. Çok büyük kitlelere uygun çalışacak bir üründe evrimsel model tercih edilebilir.

Çok büyük veya çok küçük kitleli olmayan orta derecede zaman gerektirecek süreçlerde çevik modeli kullanmak daha doğrudur.

Maliyet açısından bakıldığında en düşük maliyetli modeller artırımlı model, kodla ve düzelt modeli ve evrimsel modellerdir. En yüksek maliyeti çıkaracak model ise çevik modeldir. Çevik modelde aynı zamanda tecrübeli, uzman personel gerekliliği de yüksektir.

Risk açısından değerlendirildiğinde ise evrimsel, çağlayan ve kodla düzelt modellerinin risk oranı yüksektir. Bu modellerde büyük risk analizleri yapmak ve yoğun önlemler alarak hata riskinin düşürülmesi hedeflenir.

Burada anlatılan birçok faktör göz önüne alınarak yapılacak projeye göre hangi modelin kullanılacağı seçilebilir.

KAYNAKÇA

<https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/>

<https://www.atlassian.com/agile/scrum>

<https://www.softwaretestinghelp.com/agile-and-scrum-terminology/>

bil-102-2_hafta-yazilim-yasam-dongu-modelleri.pdf