

BİL 102 Yazılım Mühendisliği
Temelleri Ödevi
(15.03.2023)

Damla nur alper

220601017

Yazılım yaşam döngüsü nedir?

Yazılım aslında bir üründür ve bu ürünlerin yaşam süreci vardır. Bu anlamda herhangi bir ürün geliştirilmesi için çeşitli aşamalardan bahsediliyorsa yazılım için de aynı aşamalar mevcuttur. Genel olarak ürünlerin tamamını birer sistem olarak görmek sistem geliştirme aşaması altında incelemek gerekir.

SDLC kavramı ilk başlarda yazılım yaşam döngüsü olarak kabul ediliyordu. Fakat sistem geliştirme yaşam döngüsü (system development life cycle) ismi daha çok kullanılmaya başlandı. Aslında her sistemin geliştirilmesi için kullanılan bir yapıya sahip. Dolayısıyla herhangi bir sistem geliştirirken de bu yaşam döngüsüne bakmakta fayda vardır. Her kurumda benzer aşamalardan geçen yazılım geliştirme süreçleri ve bilişim sistemleri kavramlarını farklı kelimeler ve aşamalarla anlatmak mümkün olsa da genel olarak aşağıdaki modeller adı altında incelemek mümkündür.

- Planlama
- Tanımlama
- Tasarım
- Geliştirme
- Gerçekleştirme
- Uygulama
- Bakım

Bu 4 aşama aslında bir dairenin merhaleleridir. Yani bütün aşamalar tamamlandıktan sonra tekrar başa dönülür ve her şey yeniden devam eder. Aslında her yazılımın da bir ömrü vardır ve bu ömür tamamlanıncaya kadar yazılımda gerekli yinelemeler ve değişiklikler ile yeni yazılım müşterinin veya kurumun hizmetine sunulur, bu süreç bir süreklilik arz eder.

Planlama aşamasında yazılım projesinin planlaması ve görev dağıtımı yapılır. Bu aşama planlamanın bütün gerekliliklerini kapsar. Kaynak ayırımı, personel, kurumun bilişim seviyesi, ve yeni sistem kullanılabilirliği gibi çok sayıda işlem bu aşamada incelenir. Bu aşamanın tamamlanmasının ardından bir uygunluk raporu hazırlanarak kurumdaki yönetime sunulur ve bilişim sisteminin yapacağı olumlu katkıların maliyet analizine göre katma değeri olduğu kesinleştirilir. Bu plan aşaması aslında proje yöntemlerinin en önemli aşamasıdır. Diğer adımların da başlangıç aşamasıdır, işin projelendirildiği, fikrin ortaya çıkarıldığı ve fikrin tartışıldığı aşamadır.

Tanımlama aşamasında planlama aşamasından sonra planlanan ürünün tanımları burada yapılır. Genelde tartıştıkları fikrin ne olduğu ve temel tanımlar üzerinde konuşulacak kavramların tanımlanması da bu aşamada olur. Eğer aynı fikirden bahsedilmiyorsa farklı sonuçlara varılır.

Tasarım aşamasında yazılımın veya sistemin tasarımları yapılır, projeleri çizilir, yazılım projesinin ne kadar süreceği, ne gibi risklerin olacağı belirlenir. Sistemin bütün bileşenleri ve genel yapısı tasarlanır. Çok çeşitli tasarım yöntemleri ve aşamaları var ve genelde mantıksal bir tasarım aşamasından geçilerek sistemin hedefleri ve etkileşimde olduğu alt bileşenler arasındaki uyum gözlenir. Planlama ve tanımlamaya göre bir tasarım çizilir, kararlar verilir,

seimler yapılır, yazılım bileşenleri, modülleri burada tasarlanır. Geliştirme aşamasına bütün kararlar verilmiş olarak geçilmesi gerekir, herhangi bir kafa karışıklığı bırakılmaz.

Geliştirme modelinde bu projeler için kodlama aşamasıdır ve sistemi yaşatmaya başlanılan ilk örneklerin çıktığı aşama olarak kabul edilir. Burada programlama dilleri, geliştirme ortamları, kodlama standartları ve diğer araçlar kullanılır. Tasarımda karar verilen ortama uygun olarak yine orada verilen kararlara göre projenin gerçekleştirilmesine başlanır. Kısaca daha önceden projesi tamamlanan ürünün gerçekten yapılmaya başladığı aşamadır.

Gerçekleştirme sistemin gerçek hayata uygulandığı aşamadır.

Uygulama aşamasındaysa proje artık yaşamaya başlar. Bu yapılan proje artık kullanılır ve bunlardan doğan problemlere karşılık tartışılır, yeni fikirler ortaya konur ve bunlar değerlendirilir. Bu değerlendirme sonrası son olarak bakıma geçilir.

Son olarak bakım aşaması projenin farklı aşamalardan geçtiği evredir, yeni yeni tanımlar ve fikirler, tasarımlar yapılır. Bu aşamalar bittiğinde planlama aşamasında yeni bilgi alınır ve devam edilir bu döngü hep devam eder.

Yazılım geliştirme yaşam döngüsünün SDLC içinde kullanılan diğer modeller olarak düşünebiliriz. Burada bir yaşam döngüsü, geliştirme aşamaları, ve entegrasyon aşamaları vardır. Bunlar SDLC 'nin içinde kullanılan alt proje yönetim modelleri olarak tanımlanır. Yaşam döngüsünü düşünürsek bunun içinde çok farklı yöntemlerle geliştirilmiş epey projeden söz edebiliriz, bu farklı yazılım geliştirme aşamalarından ilki;

Waterfall (şelale) modeli için genelde proje modelleri bununla başlar. Mühendislikte anlatılan ilk modellerdendir. Bu model statik olarak tanımlanır. Bu model planlama ile başlar ve analizlerle devam eder; durum analizi yani bu yazılımın hangi ortamda gerçekleştirileceği gibi sorulara cevap veren analizler yapılır ve sonra da istek analizi yani “bizim bu projedeki amacımız ne? Ne istiyoruz?” soruları sorarak analizler yapılır. Sonra da sırasıyla yukarıdaki aşamaların hepsi bu modelde uygulanır. Planlama ile başlayıp testlere kadar sisteme dokunamayız ama yeni bir şey yapılacaksa eğer planlamaya geri dönüp tekrar eline alınması gerekir ki bu uzun ömürlü projeler için büyük tehdit olarak algılanır. Çünkü bu proje ne kadar sürüyorsa o kadar beklemek zorundayızdır ki karışma şansı yoktur. O yüzden bu model bilişim projeleri için çok kullanışlı değildir. Ama diğerleri arasında en basit en temel en etkili yöntem olduğundan önemlidir ve küçük projeler için birebirdir. Bu hataların maliyeti de fazlaştığından bu durumu şelale modelinin dezavantajı olarak görebiliriz.

Bu model tarihsel olarak ilk başta üretim ve inşaatla kullanılmıştır. Bu sektörde isteklerin zamanla değişmesinin maliyetli olduğu zamanlarda üretim tamamlanıncaya kadar değişimin olmadığı zamanlarda uygulanıyordu. Bu üretimlerin tasarımından sonraki talepler hiç karşılanamadığı için şelale modeli de bu yapıyı desteklemiştir. Bu model 1983 yılında Benington tarafından kazandırılmıştır. Royce da Benington gibi bu modeli şelale ismi kullanılmadan önce bu metottan bahseden ve en çok atıfta bulunduğu için önemli bir yere sahiptir.

Diğer bir model olan V shaped model için adımlar aynı V şeklini oluşturduğu için bu adı almıştır. Yine aynı şelale modeli gibidir ama belirli farkları vardır. Yine Planlama, ihtiyaç belirleme, seviyeli tasarımları vardır ama üst seviye tasarımda daha genel bir tasarım yapılır. Sonra da kodlamaya geçilir. bu kısma kadar şelale ile çok benzerdir. V model aslında alttan yukarı çıkan ve bu aşamaların karşılığı olan bir modeldir. Modeldeki aynı hizada olan aşamalar birbirinin karşılığıdır. Üst seviye ve alt seviyenin uyumlu olup olmadığı da kontrol edilir. Bu model şelale modelinin geliştirilmiş halidir, belirsizliklerin az olduğu projelerde kullanılması uygundur. V şeklinde sol taraf üretim, sağ taraf sınaama olarak geçer. Döngü üç modelden oluşur:

- Kullanıcı model
- Mimari model
- Gerçekleştirim model

Kullanıcı modelde istekler ve proje tanımları kullanılır ve bitmiş hali teslim edilir.

Mimari modelde ise tasarım ve bunları deneme yanılma işlemleri yapılır.

Gerçekleştirim modeldeyse kodlama ve bunların denemesinden oluşur.

V modelin dezavantajlarından biri de riskler için çözümleme yeri ve aşamalarda tekrar bulunmaz.

Diğer bir modelimiz kodla ve düzelt modeli çok küçük projeler ve ömrü kısa projelerde kullanılabilir çünkü bu modelde herhangi bir planlamaya ihtiyaç duyulmaz. Sadece uzmanların kullanabileceği bir model değildir aksine herkes bunu rahatlıkla kullanabilir. Program aşamaları da hızlı hızlı ilerler. Hemen ürün geliştirilir ve sunulur, büyük projeler için kullanılmaz. Bakım aşaması yeterli değildir ve zordur. Diğer modellerde olduğu gibi dezavantajları mevcuttur; ne zaman biteceği belli değil ve kaynak planlaması yoktur. Ayrıca kontrollü değildir ve kodları düzeltmek maliyetli olabilir.

SCRUM

Scrum yazılım geliştirmede kullanılan bir uygulamadır. Atık yazılım geliştirme yöntemi olarak çevik yönetim ve proje yönetiminde karmaşık bir ortamda ürünleri geliştirmek, sunmak ve sürdürmek için bir çerçevedir. Bunu temel özelliği gözlemci geliştirmeci ve tekrarlanabilmesidir. Asıl temelde karmaşık projeleri bazı koyduğu ilkeler ile karmaşıklığını azaltmaya çalışmaktır. Scrum büyük projeleri böler ve bunların her birine “sprint” adı verir. Bu teker teker böldüğü sprintleri geliştirir. Bu yüzden ekipte iletişim, paylaşım çok önemli ve gereklidir ve bunu sağlayan “scrum meetings” adlı toplantılar kurularak fikirler tartışılır. Scrum ın en önemli özelliği başta hayal edilen ve tasarlanan projenin hızlı ,ucuz ,kaliteli yani en iyi şekilde üretilmesidir. Bu taleplerin gerçekleştirilmesi diğerleri gibi aşama aşama olmaz bunun yerine sprint dönemlerinde geliştirilir. Her sprint sonu yazılımın fonksiyonel kısmı bitmiş ve müşteriye sunulabilir şekilde olur. Bu yöntem Çevik yazılım geliştirme prensibini hayata geçirir . scrumdaki üç temel kavram her şeyi belirler:

- Görevler
- Toplantılar
- Araçlar

Bu görevler ürün sahibi, scrum yönetici, scrum takımından oluşan bir aşamadır. Sahip projenin başıdır, en önemli rol ondadır. Yönetici ise takımı kontrol eder, takım da birbirleriyle iletişimi önemli tutan yazılım geliştirme ekibine denir.

Toplantılar da scrumın olmazsa olmazı, mihenk taşıdır. Özellikle her gün bu toplantılar yapılır ve bu toplantılarda yazılım geliştiricilerin neler yaptığı neler ortaya koyduğu, ve proje aşamasında ne sorunlarla karşılaştığı ve daha ne yapacaklarını konuşurlar ve her şeyi aşamalı aşamalı gözden geçirirler.

Araçlar için ürün gereksinim dökümanı oluşturulur ve bunda projede yapılması gereken şeyler yazılmıştır. Kalan zaman grafiği de proje sırasında yardımcı bir doküman olup yapılan işin ne seviyede olduğu ve nerede olduğunu belirlemek için hazırlanır.

Scrum günümüzde en çok kullanılan popüler bir modeldir. Sadece bu projeler için değil birçok sistem için de kullanılır. Bunların en büyük sebebi avantajlarının çok olmasıdır. Mesela zamandan ve maddiyattan çok tasarruf edilmesi, güncel kalabilmesi, her şeye uyum sağlayabilmesi, ne olduğu belirsiz, karmaşık, çözülemeyen projeler için birebir olması gibi avantajlarından bahsedilebilir. Ayrıca diğer metotlar gibi yinelemelidir. Değişken durumlara hızlı bir karşılığı olması da cabası.

Bu yazılım geliştirme modelleri için hepsinin farklı bir durumu söz konusudur. Farklı projelerde farklı metotlar kullanılır. Küçük ve basit projeler için şelale modeli kullanılması uygundur. V modeli şelalenin gelişmiş versiyonu olduğundan daha farklı projelerde de kullanılabilir. Belirsizliklerin az olduğu projeler için v süreç modeli uygundur. Scrum da klasik yöntemlerle başlayan bir projeyi kurtarmaya uygundur. Aynı şekilde büyük projelerde de kullanılabilen scrum çok kullanışlıdır.

KAYNAKÇA:

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

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

<https://www.academia.edu/>

<https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742>

<https://ybsansiklopedi.com/>

https://www.tutorialspoint.com/sdlc/pdf/sdlc_v_model.pdf