

YAZILIM GELİŞTİRME SÜRECİ

Her ürün geliştirme işlemi belirli aşamalar içerir.Yazılımda bir ürün olduğu için geliştirme aşaması vardır.Yazılımın hem üretim,hem de kullanım süreci boyunca geçirdiği tüm aşamalar yazılım geliştirme yaşam döngüsü olarak adlandırılır.Bir döngü şeklinde olmasının nedeni yazılım işlev ve ihtiyaçlarının sürekli değişmesi ve gelişmesidir.Döngü içerisinde herhangi bir evreye geri dönebilir.Döngünün bazı temel adımları vardır. Bunlar; Gereksinim,Analiz,Tasarım,Gerçekleştirme,Bakım olarak sıralanır.

Gereksinimler

Müşteri gereksinimlerinin elde edildiği ve fizibilite çalışmasının yapıldığı aşamadır.

Analiz

Sistem gereksinimlerinin ve işlevlerinin ayrıntılı olarak incelendiği aşamadır.Temel sorunlar ortaya çıkarılır.

Tasarım

Yazılım projesinin nasıl sürdürüleceğinin tasarlanması.

Gerçekleştirme

Yazılım projesinin kodlama ve test kısmıdır.

Bakım

Yazılımın ürün olarak sunulduktan sonra güncelleme,bakım olaylarının yapılmasıdır.

Gelişigüzel Model

Model veya yöntem gözetmeksizin yapılır.İlgi duyan ve geliştiren kişiye bağımlıdır.Takip edilebilirliği ve bakımı oldukça zordur.60'lı yıllarda kullanılmış bir model olup artık kullanılmamaktadır.Tek kişilik üretim ortamı vardır.Çok basit programlamaya sahiptir.

Barok Modeli

Bu model yaşam döngüsü adımlarının doğrusal bir şekilde geliştirildiği modeldir.70'li yıllarda kullanılmış ve günümüzde artık tercih edilmeyen bir modeldir.Bu model belgelemeyi ayrı bir süreç olarak ele alır.Halbuki belgeleme yapılan işin doğal bir ürünüdür. Adımlar arası geri dönüşlerin nasıl yapılacağı da tanımlanmamıştır.Gerçekleştirim evresine daha çok ağırlık veren bir model olduğundan bugün tercih edilmeyen bir modeldir.

Çağlayan (Şelale) Modeli

Bu model diğer yaşam döngü modellerinin temelidir.Geleneksel bir model olup geçmişteki en popüler modeldir diyebiliriz.Bu modelde her aşama en az bir kez tekrar edilir.Şelale modeli çok iyi tanımlanmış ve kısa sürede bitebilecek projelerde kullanılır.Eğer barok ile kıyaslayacak olursak şelale modelinde belgeleme sürecin içerisinde bir adım tamamlanmadan diğer bir adıma geçilemez.

Öncelikle gereksinimlerin tanımlanması yapılır. Sonrasında sistem ve yazılım tasarımı gerçekleştirilir.Kodlama ve parça test etme etapları gerçekleştirilir.Yapıyı birleştirme ve sistemi test etme yapıldıktan sonra da sistemin bakım ve idamesi sağlanır.

Bilişim projeleri için çok uygun bir yönetim sistemi değildir.Ama en basit,en etkili ve hepsinin temeli olan bir yöntem olduğu için bilmekte fayda vardır.Küçük projeler için uygulanabilir.Hangi aşamada hata var ,problem kimden kaynaklanıyor veya problemi nerede çözebiliriz gibi bilgileri verir.Genelde burada geri dönüşler de söz konusudur.Tasarımda bir problem olduğunda analize,uygulamada bir problem olduğunda tasarıma dönülmesi gibi geri dönüşler yapılabilir.Hataların maliyeti de giderek artar.Analiz aşamasındaki bir hatanın düzeltilmesi çok düşük maliyeti olurken,bu problem test aşamasında yakalanırsa tekrar analize dönülüp bunun telafisi tekrar bütün süreçlerin sıfırdan ele alınmasını gerektirir ve maliyet artışını beraberinde getirir.

Bu modelin bazı sorunları vardır.Bütünde yazılımın kullanıcıya ulaşma zamanı uzundur.İhtiyaç tanımlamaları çoğu kez açık bir şekilde yapılamadığından ötürü,hataların düzeltilme ve eksikleri giderme maaliyetleri artırır.Üst düzeydekilerin ürünü görme müddetinin uzun olması, projenin bitmeyeceği ve devamlı gider merkezi boyutuna geldiği düşüncesini artırmaktadır.

Fıskiye Modeli

Fıskiye modeli şelale modelinden öykünerek oluşturulmuş bir modeldir.Henderson-Sellers and Edwads tarafından geliştirilmiş bir modeldir. Şelale modelinden farkı , döngüleri vardır.Döngüler şeklinde her bir aşamada belli döngüler elde edilir.Tasarımaşamasında tekrar koddan tasarıma dönülmesi, tasarımdan istek analizlerine geri dönülmesi ve operasyona geçtikten sonra testlere geri dönülmesi gibi döngülere sahiptir. Maintenance (bakım) ve evolution (kriterler) belirleniyor.Bu yaklaşımda şelale modelindeki adımlar görülebilir.

V Süreç Modeli

Adımlar V şeklini oluşturduğu için V şeklinde model denilmiştir.Aslında şelale modeline benzer ama farkları da vardır.Yine planlama, ihtiyaç belirleme, üst seviye ve alt seviye tasarımlar vardır. Üst seviyede daha genel bakışa göre tasarım yapılır.Üst seviye tasarımlarında daha sonra daha detaylı alt seviye tasarımlarının yapıldığı görülür.Daha alt seviyelerinin açıldığı , bunların girdilerini-çıktılarını ve beklentilerinin yazıldığı söylenebilir.Sonrasında kodlamaya geçilir.Buraya kadar şelale modeli ile benzerlikler söz konusudur.Aşağı doğru akan bir proje yönetimi görülür.Buradan sonra yukarı doğru çıkılmaya başlanır.Birim testler yapılır.Her üretilen modülün testi yapılır.Daha sonra bu modüllerin alt seviye tasarım ürünlerinin birbirleri ile çalışma durumu entegrasyon testleri yapılır ve kabul testleri aşamasında müşterinin kabul edip etmemesi ile ilgili müşteriye gidilir.Müşteri uygulamayı test eder sonra bakım başlar.

Arttırımlı Model

Üretilen her bir yazılım sürümü diğerini kapsayacak ve giderek fazlalaşan adette işlev içerecek şekilde modernleştirilir.

Proje parçalara bölünür ve kullanıcının önceliğine göre bu parçalar sıralanır.Sıralanan bu parçalar bittiğinde sırasıyla birer ara ürün geliştirilmiş olur ve bu ara ürünler de kullanıcı tarafından kullanılır.Ara ürünler her seferinde bir öncekinin üstüne bir şeyler katarak çıkartılır. Yani bu modelde bir taraftan üretim kısmı sürerken diğer tarafta ise kullanım kısmı sürer.Artımlı gelişim modeli uzun zaman alabilecek ve ürünün eksik işlevsellikle çalışabileceği türdeki yazılımlar için uygundur.Bu model ile sistemin başarısız olma olasılığı azalır , ara ürünler yazılımın geliştirilmesinde önemli bir yere sahip olur.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.

Kodla ve Düzelt Yaşam Döngü Modeli

Bir kaç yüz satırdan oluşan programlar için kuullanılabilir.İlk safhada yazılım ürünün ilk sürümünü geliştiriyoruz. Direkt olarak yazılım ürünü gerçekleştirilir.Sistem en son istenilen şelke gelinceye kadar devamlı geliştirilir.Bakım

safhası vardır ama çok zordur.Çünkü sisteme ait dökümantasyon yoktur.Ayrıca emeklilik safhası vardır.

Yazılım geliřtirmenin en kolay ama en pahalı yoludur.Yazılım geliřtirmek kolay olduđu için ne yazıkki küçük firmalardaki bir çok yazılım projesinde bu model kullanılır.

Helozonik (Spiral) Model

Modelin geçirdiđi safhaları sıralamak gerekirse ; önce planlama gelir.Ardından risk analizi ve üretimi saymalıyız.Son olarak da kullanıcı değeriendirilmesi olarak bitirilmelidir.

Planlama : Her aşamada olan ara ürün için bir planlama yapılır.

Risk Analizi : Risklerin araştırılması, belirlenmesi ve çözülmesi.

Üretim : Ara ürünün/ürünün üretilmesi.

Kullanıcı Değeriendirmesi : Oluřturulan ara ürünün sonucunda kullanıcıdan alınan geri dönütlerin değeriendirilerek diğeri aşamaya geçilmesi.

Bu modelin avantajları vardır. Kullanıcı katkısı en ön plandadır.Yönetici bakışı çok önemlidir.Bir de yazılım geliřtirici (mühendis) bakışı gibi avantajlar vardır.

Helezonik Modeli ifade etmek için önce söylenecek şey ; Risk Analizi olgusu ön plana çıkmıştır.Her bir döngü bir fazı tasvir eder.Direkt tanımlama ,tasarım ve saire gibi bir faz yoktur.Artımsal bir yineleme yaklaşım vardır.Ayrıca bu modelde prototip yaklaşım mevcuttur.

Helezonik Modelin dezavantajları ; küçük ve düşük riskli projeler için çok pahalı bir sistem olması ,karmařık bir içeriđe sahip olması, uzun sürmesi ve fazla dökümantasyondan oluşması olarak sıralanabilir.

Çevik (Agile) Modelleme

Yazılım tabanlı sistemlerin dökümantasyonu ve modellemesi için uygulama tabanlı bir yöntemdir.Agile modellemenin amacı ; geleneksel modelleme yöntemlerinden daha esnek bir tavır içinde , yazılım geliřtirme projeleri üzerinde uuygulanabilecek değeri, ilkeler ve uygulamalarda birleřmektir.Agile Modelleme diğeri agile metodolojileri için tamamlayıcıdır.Modelleme ve dökümantasyonun nasıl yapılması gerektiđini tanımlar.

Çevik Modellemenin başlıca özelliđi veri modelleri ve ara yüzü modelleri gibi modelleme tekniklerinin neler olduđunu ve bunların ayrıntılarını söylemek yerine bu tekniklerin nasıl uygulanması gerektiđini söylemesidir. Mesela ; yapılan projelerin test edilmesi gerektiđini belirtse bile nasıl test hazırlanacağına değeriinmemesi gibi.

Kullanıldıđı Durumlar

Projenin yazılım evresinde müşteriiden gelebilecek talep değerişikliklerinin tahmin edilemez olduđu durumlarda kullanılabilir.

Projenin parçalarının önce tasarlanıp ardından hemen geliřtirilmesinin gerekmesi ve önceden ne yapılacağına , detaylı yol haritasını ve taarımını tahmin etmenin çok güç olduđu durumlarda kullanılabilir.

Analiz,tasarım ve test etme süreçlerinin ne kadar zaman alacağını önceden bilinmemesi durumunda kullanılabilir.

Yazılım ekibinin birlikte çalışmak , hiyerarřiye önem vermemek, sağlam iletiřim kurmak gibi özelliklere sahip olması gibi durumlarda da kullanılabilir.

Çok kişinin dahil olduğu projelerde çevik metotlar ile proje geliştirmek mümkün olmayabilir.

Çevik metotlar tam bir yazılım süreci değildir ama kapsamlı yazılım geliştirme yöntemlerini tamamlayıcı niteliktedir. Başlıca çevik süreç modelleri : Sınırsal programlama , Çevik Birleştirilmiş Süreç, Scrum , Test GÜdümlü Geliştirme, Çevik Bilgi Metodu, Özellik GÜdümlü Geliştirme.

Biz şimdi Scrum'u ele alacağız.

Scrum

Bir uygulama geliştirme yöntemidir. Bu geliştirme yönteminin temel özelliği gözlemci,geliştirmeci ve tekrara dayalı olmasıdır.Birçok modern yazılım projesinin oldukça karmaşık olduğu ve en baştan tümünü planlamanın zor olacağı şeklindeki bir varsayımdan hareket eder.Bu karmaşıklığı üç ilke ile azaltmaya çalışır.

1- Şeffaflık : Projedeki ilerlemeler ve sorunlar günlük olarak tutulur ve herkes tarafından izlenebilir olması sağlanır.

2-Denetleme : Ürünün parçaları ya da fonksiyonları düzenli aralıklarla teslim edilir ve değerlendirilir.

3- Uyarılma : Ürün için gereksinimler en baştan bir defalığına belirlenmez,bilakis her teslimat tekrar değerlendirilir ve duruma göre uyarlamalar yapılır.

Amaç başlangıçta hayal edilen ve tasarlanana uyan bir ürünün ,hızlı,ucuz ve kaliteli bir şekilde üretilmesidir.Tasarlanan ürünün gerçekleştirilmesi,müşteri/kullanıcı tarafından mümkün olduğunca detaylı şekilde hazırlanmış bir talepler listesinin aşama aşama gerçekleştirilmesi biçiminde yapılmaz.Bunun yerine müşteri/kullanıcı tarafından istenilen ve tanımlanan işlevler,iki ya da dört haftalık "Sprint" adı verilen dönemler içerisinde geliştirilir ve yeniden gözden geçirilir. Bu kullanıcı bazlı gereksinim tanımı Kullanıcı Hikayesi olarak nitelenir ve özellikler defterinde yer alır. Her Sprint sonunda yazılımın fonksiyonel bir parçası bitmiş ve müşteriye teslim edilebilir bir durumda olur.Scrum Çevik yazılım geliştirme prensiplerini hayata geçiren bir yöntemdir.Scrum karmaşık projelerin yönetimi için bir çatıdır.

Scrum Süreç-Modeli net olarak üç çeşit rolü tanıır : Ürün Sahibi,Takım ve Scrum Ustası Dış roller ya da yan rollerolarak da: işletme,müşteri,kullanıcı ve pazarlama. Takım kendi içinde üç ayrı yetenek barındırır: Yazılımcı,Testçi ve Tasarımcı.

Roller

Ürün Sahibi:Projenin iş değeri açısından geri dönüşü ile sorumludur.

Scrum Yöneticisi: Takımın Scrum'un temel değerlerine,pratiklerine ve kurallarına bağlı kalmasını garanti altına alır. Takımı ve organizasyonu Scrum'a adapte eder.

Scrum Takımı: Scrum takımı, devamlı iletişim halinde olan ve tek bir hedefe ulaşmak için mücadele eden kişilerden oluşur.

Toplantılar:

Sprint Planlaması

Geniş kapsamlı gereksinim listesinin çıkarılması

Başarılı geliştirme için uygun dağıtım gereksinimlerinin belirlenmesi

Dağıtımlar için gereksinimlerin eşleştirilmesinin yapılması

Dağıtımlar için takımların belirlenmesi

Risk deęerlendirmesi ve risk kontrollerinin belirlenmesi

Yönetimi ve destekleri gözden geçirme ve onaylanması

Sonuç

Bu çalışma kapsamında yazılım yaşam döngü kavramları ve yazılım yaşam döngü modelleri açıklanmıştır. Ayrıca agile modellemeye de değinilerek scrum kavramı üzerinde durulmaya çalışılmıştır.

KAYNAKÇA

Ömer Harun Çetin medium hesabı yazılım yaşam döngü modelleri

Hayri Rıza Çimen medium hesabı yazılım yaşam döngü modelleri

Ybsansiklopedi Cilt 2, Sayı 3,Eylül 2015

İrem KUMLU

190601017

