Kerem Levent

Bilgisayar Mühendisliği 1.Sınıf

210601039

**YAZILIM YAŞAM DÖNGÜ MODELLERİ**

Yazılım ürününün geliştirilmesi sürecinde içerisinden geçtiği her türlü süreç ve adımlara yaşam döngüsü modeli denir.Bu modeldeki safhaların sayısı ve modeli kullanılan modele göre değişir. Yaşam döngüsünün temel adımları; analiz, çözümleme,tasarım, gerçekleştirme ve bakımdır.

Gereksinim:Müşterinin gereksinimleri belirlenir.  
Analiz: Gereksinimlerin detaylıca belirlendiği aşamadır.Temel hatalar ve sorunlar ortaya çıkarılır.   
Tasarım: Gereksinimlere yanıt verecek yazılım tasarımı burada yapılır.  
Gerçekleştirme: Kodlama ve test etme işlemleri gerçekleştirilir.  
Bakım: Teslim sonrası hata ve eksiklikleri giderme aşamasıdır.  
**Yazılım Geliştirme Süreç Modelleri**  
Yazılım yaşam döngüsü modelleri belirtilen süreçlerin geliştirme aşamasında, hangi düzen ya da sırada, nasıl uygulanacağını tanımlayan modellerdir.  
Bu modellerden bazıları şunlardır;  
• Gelişigüzel Model  
• Barok Modeli  
• Çağlayan Yaşam-Döngü Modeli  
• V-Süreç Modeli  
• Helezonik (Spiral) Model  
• Formal Sistem Geliştirme  
• Artırımsal Geliştirme Süreç Modeli  
• Kodla ve Düzelt Yaşam Döngüsü Modeli  
**Kodla ve Düzelt**  
Birkaç yüz satırdan oluşan projeler için uygun olan bu model direkt yazılım geliştirilerek başlanır. Yazılan yazılım dinamik olarak geliştirilerek istenilen hale getirilmeye çalışılınır.   
**Çağlayan Modeli**  
 Bu yöntem analiz, tasarım, kodlama, test, sürüm ve bakım gibi safhalardan oluşur. Bu modelde bir sonraki aşamaya geçmek için, bir önceki aşamanın tamamlanması gerekir. Her aşamanın sonunda bir doküman oluşturulur. Analiz adımında, tüm yazılım gereksinimleri net bir şekilde belirlenerek analiz dokümanı üretilir. Tasarım adımında yazılımının arayüz, veritabanı, sınıf vb tasarımları yapılarak tasarım dokümanı üretilir. Kodlama adımında yazılım; analiz ve tasarım dokümanlarında belirtilen şekilde kodlanır. Test adımında; analiz ve tasarım dokümanlarındaki tüm fonksiyonel ve fonksiyonel olmayan gereksinimler ve tasarımlar için test senaryoları yazılır ve bu test senaryoları denenerek yazılımın testleri yapılır. Testlerde bulunan hata hangi adımdan kaynaklı ise o adımı geri dönülerek o hata düzeltilir. Daha sonra kodlama düzeltilir ve test aşamasına geçilir. Test adımı sonunda, yazılımda herhangi bir hata bulunmaz ise, sürüm adımıyla sonlanır,ürün alıcıya teslim edilir.  
Şelale modeli, daha çok askeri vb sektörlerde, uzun süreli, analiz ve tasarımın çok önemli olduğu, hata yapmanın maliyetinin çok yüksek olduğu, kritik projelerde kullanılır.  
**Evrimsel Geliştirme**  
Bu model ilk tam ölçekli modeldir. Coğrafik olarak geniş alana yayılmış, çok birimli organizasyonlar için önerilmektedir.Bu modelde sistem zaman içinde kazanılan anlayışa göre gelişir(sürümler şeklinde geliştirilir). Müşteri tarafından talep edildikçe yeni özellikler eklenerek sürüm ilerletilir. Bu model müşterinin ne istediğini bilmediği zaman kullanılır.  
**Prototipleme**  
Müşterilerin ayrıntılı girdi, süreç ve çıktı gereksinimlerini daha iyi tanımlayabilmelerini sağlamak için kullanılan bir modeldir. Hızlıca yapılan bir tasarım ile yazılımın kullanıcıya yansıyacak yönünü aktaran bir ilk örnek üretilir. Prototip kullanıcının kullanımına ve değerlendirilmesine sunulur. Bu değerlendirmelere bakılarak ilk örnek üzerinde gerekli değişiklikler yapılır.   
**Spiral Model**  
Dört aşamadan meydana gelir: hedeflerin belirlenmesi, risk değerlendirme ve azaltma, geliştirme ve doğrulama ile planlamadır. Bu modelde risk analizi ön plana çıkmıştır.Spiral modelin, risk hareketli bir model olması, oldukça esnek oluşu, daha az dokümantasyon gerektirmesi ve prototiplemeyi kullanması gibi avantajları vardır  
**Formal Sistem Geliştirme**  
Bu model matematiksel belirtinin farklı gösterim şekilleri ile çalıştırılabilir programa dönüştürülmesine dayalıdır.Belirtim hatalarını minimize eder. Zaman alan pahalı bir yöntemdir Bu model hatasız yazılım imkanı sunar.  
**Artımlı Geliştirme**  
Sistem fonksiyonel birimlere ayrılıp, teslimatlar artımsal fonksiyonel birimler halinde yapılır.Müşteri, değer almak için tüm sistemin bitmesini beklemez. Kritik ihtiyaçlarını karşılayan ilk geliştirme ile yazılımı kullanmaya başlar. Başarısızlıkların tüm projeye yansıması engellenmektedir.  
**Birleşik Süreç**  
 Bu model değişken isteklere uygun bir modeldir. Erken geri besleme, risklerin erken giderilmesi, erken ürün elde etme gibi avantajları vardır. Risk yönetimi düşüktür. Dokümantasyon yükü ağır olduğundan pahalı olabilir ve karmaşık bir modeldir.  
Birleşik süreçte yazılım geliştirme aşamaları şu şekildedir :  
• Başlangıç: Vizyon kararı, fizibilite çalışması, tamam ya da devam kararı.  
• Ayrıntılandırma: Daha gerçekçi çözümleme, çekirdek yapının ve yüksek riskli kısımların yinelemeli olarak oluşturulması.  
• Tamamlama: Daha az riskli ve düşük öncelikli kısımların yinelemeli olarak gerçeklenmesi.  
• Yayım: Beta testleri, piyasaya sürme çalışmaları.  
**V Modeli**  
Şelale modelinden farklı olarak yazılım geliştirmeye başlamadan test planı oluşturulur.V modelin sol tarafı yazılım geliştirme faaliyetlerini sağ tarafta ise test aşamaları yapılır. Yazılım geliştirme faaliyetlerinden iş gereksinimi başlandığında test aşamalarından da kabul testlerine başlanır.  
**Çevik Modelleme**  
Çevik yazılım metodu, kısa vadeli planlar ve küçük parçalar halinde yazılımın geliştirilmesini ön görür.İstenilen sonuca ulaşmak adına birden çok yineleme gereklidir.  
**Scrum**; Agile proje yönetim metodolojilerinden biridir. İlk bakışta çok basit kuralları olan bir yönetimsel modeldir. Gereksinimleri açıkça belirli olmayan, değişime açık, karmaşık yazılım projelerinin yönetimi için uygulanmaktadır. Scrum, detaylı bir şekilde projede izlenmesi gereken adımları belirtmemekte, onun yerine basit ama önemli birkaç olmazsa olmaz kuralıyla esnek bir yönetim sunmaktadır.  
Scrum 3 temel prensip üzerine kurulmuştur;  
• Şeffaflık; Projenin ilerleyişi, sorunlar, gelişmeler herkes tarafından görülebilir olmalıdır.  
• Denetleme; Projenin ilerleyişi düzenli olarak kontrol edilir.  
• Uyarlama; Proje, yapılabilecek değişikliklere uyum sağlayabilmelidir.  
**Scrum süreç modeli üç farklı rol üzerinde şekillenir**. Bunlar; ürün sahibi, yazılım geliştirme takımı ve Scrum yöneticisi (ustası) olarak ayrılır. Yazılım geliştirme takımı da kendi içinde üç farklı role ayrılır. Bunlar; Yazılımcı, Testçi ve Tasarımcı’dır. Dış roller veya yan roller de diyebileceğimiz kategoride de müşteri, işletme, kullanıcı ve pazarlama vardır.  
**Ürün Sahibi:** Kullanıcı istek ve durum önerilerini ayrıntılı olarak yazılım geliştirme takımı ile paylaşır. İsteklerini Sprint dediğimiz 2 haftalık süreçler halinde takıma iletir.  
**Yazılım Geliştirme Takımı:** Gereksinim ayrıntılarını, istenilen ve elde edilen verileri, hangi zaman çizelgesi üzerinden projenin gerçekleştirileceğini, sonuç verilerini ürün sahibi ile paylaşır.  
**Scrum Yöneticisi:** Yazılım Geliştirme Takımı ve Ürün Sahibi arasında köprü görevini üstlenen kişidir. Ürünü belli standart düzeyinde tutmak, scrum sürecindeki oluşabilecek negatif durumlarla ilgilenmek, takımın önündeki engelleri kaldırmak ile ilgilenir. Yazılım Geliştirme Takımı ile birlikte çalışır ama takıma tabii tutulmaz.  
Sprint (Toplantılar) Toplantılar da proje çerçevesinde istenilen ürün geliştirilir. Planlama süresi boyunca belirtilen ürün işlevleri Sprint’te teslim edilir. Yazılım Geliştirme Takımı’nın çalışmaları Taskboard’a göre şekillenir ve Ürün Sahibi’nin belirttiği kullanıcı isteklerine göre aşama kateder.