Yazılım Geliştirme Yaşam Döngüsü  
  
Yazılım Geliştirme Yaşam Döngüsü (Software Development Lifecycle), Bir yazılımı oluştururken geçilecek aşamalardan oluşan bir süreçtir. Bu süreç yazılımı iyi bir şekilde geliştirmeye ve ortaya çıkan hataları gidermeye yardımcı olur. Bu süreci oluşturan temel aşamalar şunlardır:  
  
1. Planlama: Yaşam döngüsünün ilk aşaması olan planlama evresinde yazılımın nasıl başarılı olacağı konuşulur, fizibilite çalışması yapılır ve bir görev dağılımı oluşturulur.

2. Analiz: Projenin tamamlanma süresi, risk durumları ve bazı temel sorunlar gözler önüne serilir.  
  
3. Tasarım: Projenin nasıl sürdürüleceği tasarlanır, yazılım sisteminin temel yapısı oluşturulur.

4. Gerçekleştirme: Yazılım projesi kodlanır, test edilir, eksik ve hatalı kısımlar gün yüzüne çıkar.  
  
5.Bakım: Gün yüzüne çıkan eksik ve hatalı kısımlar değerlendirilir ve düzeltilir.  
  
  
Yazılım Geliştirme Yaşam Döngüsü Modelleri

Yazılım projelerinde gereksinimler her zaman aynı olmaz projelerin büyüklükleri ve amaçları gereksinimleri değiştirebilir. Gereksinimlerin farklı olmasından dolayı projenin yapılış biçimleri değişebilir, bu da yeni modelleri ortaya çıkarır.   
  
1. Gelişigüzel Model: Belirlenmiş bir model veya yöntem yoktur, kişiye bağlı olarak değişir, bu yüzden de takip edilmesi ve bakım yapılması zorlaşır. Genellikle basit bir programlamaya sahiptir, 1960’lı yıllarda kullanılmıştır.  
  
2. Barok Modeli: 1970’li yıllarda ortaya çıkan bu model temel yaşam döngüsü adımlarını doğrusal olarak uygular, günümüzde yapılan işin bir parçası olan belgeleme işlemi ayrı bir süreç olarak ele alınır, gerçekleştirme evresine odaklanır, aşamalar arası geri dönüşler belirsizdir, günümüzde kullanılabilirliğini yitirmiştir.

3. Çağlayan (Şelale) Modeli: Temel yaşam döngüsü adımlarını doğrusal bir şekilde izler, Barok modelinden farklı olarak geri dönüşlerin nasıl yapılacağı belirlidir ve belgeleme işlemi süreç içinde ele alınır, aşamalar en az bir kez tekrar edilir. Ancak Çağlayan Modeli genel olarak değişime fazla elverişli değildir, yani yeni düşüncelerin ve gereksinimlerin ortaya çıkabileceği ortamlarda iyi sonuçlar vermeyebilir, bu yüzden de daha çok küçük projeler için uygundur. Geleneksel model olarak da bilinir, günümüzde etkisini yitirmektedir.  
  
4. V Süreç Modeli: Çağlayan ve Barok modellerindeki doğrusal ilerleyiş yerine burada sol tarafta üretim, sağ tarafta ise test işlemlerinin olduğu bir ‘V’ şekli vardır. Aynı zamanda bu model test sonrasında ortaya çıkan hatalardan sonra nereye dönülmesi gerektiğini de belirtir, bu da zaman kaybını önler ve sorunlar daha kolay çözülebilir. Test bölümünde bulunan bir hata, bulunduğu yerin karşısında yer alan üretim süreciyle ilişkilidir, bu sayede hatanın nedeni de anlaşılır. Bu modelin 3 temel çıktısı vardır:  
  
Kullanıcı Modeli: Geliştirme sürecinin kullanıcı ile olan ilişkilerini tanımlar. Üretim ve test açısından kullanıcıdan beklenenleri ortaya koyar.  
  
Mimari Model: Sistemin sınama işlemlerine ilişkin işlevleri barındırır.  
  
Gerçekleştirim Modeli: Yazılım modüllerinin kodlanması ve testiyle alakalı fonksiyonları barındırır.

5. Helezonik (Spiral) Model: Bu model temel olarak dört ana kısımdan oluşur; planlama, risk analizi, üretim ve kullanıcı değerlendirmesi. Planlama evresinde maliyet, zamanlama ve kaynaklar düşünülür, sistem gereksinimleri tahmin edilir. Risk analizi evresinde, riskli durumlar araştırılır ve riskler belirlenir. Üretim evresinde, bir ara ürün ileride başka ürünlerle birleştirilecek şekilde üretilir. Kullanıcı değerlendirmesi evresinde, Ara ürünü kullanan kişinin yaptığı test ve değerlendirmeler alınır.

Bu modelde risk analizi ön plandadır. Her döngü öncesinde bulunulan evrenin risk analizi yapılır ve o evre için planlanmış olan prototip oluşturulur.  
  
6. Artımsal Geliştirme Süreç Modeli: Yazılımı parça parça geliştirmeye dayanır her yeni eklenen parça ile yazılım biraz daha geliştirilir, yazılımın parça parça geliştirilmesi sayesinde geliştirilen her parça ile çalışanlar tecrübe kazanmış olur, bu yüzden de daha iyi işler çıkması mümkün hale gelir. Projeyi oluşturan parçalardan projenin çıkaracağı sonuç hakkında bilgi edinilir ve projenin işlevselliği öğrenilmiş olur. Uzun bir zaman diliminde geliştirilen ve tamamlanmamış bir sistemle de çalışabilecek durumda olan projeler için uygundur.  
  
7.Kodla ve Düzelt Modeli: Oluşturulan sistem istenilen özellikleri karşılayana kadar sürekli olarak geliştirilir. Bu modelde direkt olarak yazılım ürünü oluşturulur, belgelendirme bulunmaz bu yüzden de bakım işlemleri zor hale gelir. Bu modelle yazılım geliştirmek kolaydır, ancak proje sonunda yazılım ürününde değişiklikler yapmak bu modelle oluşturulan ürünlerin maliyetini arttırır, bu nedenle de bu model daha çok kısa projeler için uygundur.

Scrum Neden Bu Kadar Popüler?   
  
Yazılım projelerinin karmaşık bir hal alması, teknolojinin hızlıca değişmesi, müşterinin gereksinimlerini belirleyememesi gibi nedenlerden dolayı projelerin başarı oranları azaldı, projenin yönetilmesi önemli bir hal aldı, bu projelerin başarılarını arttırmak için “çevik yazılım geliştirme” kavramı ortaya çıktı. Bu kavramın metotlarından biri olan Scrum’un temel özelliği gözlemci, geliştirmeci ve tekrara dayalı olmasıdır. Yazılım projelerinin karmaşıklığının önüne geçmeye çalışan üç ilkeye sahiptir:

1-Şeffaflık: projede oluşan sorunlar ve projenin ilerleyişi günlük olarak tutulur, böylece proje herkes tarafından izlenebilir bir hale gelir.

2-Gözlem: Oluşturulacak ürünün parçaları düzenli aralıklarla teslim edilir ve değerlendirilir.

3-Uyarlama: Ürünün gereksinimleri hemen belirlenmez, değerlendirilen her teslimden sonra değişiklikler yapılır.

Scrum; disiplinli bir metot olup, projede yer alan diğer ekip üyelerinin yaptığı işlerin anlaşılmasına olanak sağlar, Scrum sayesinde alınan geri bildirimlerle hızlı ve kaliteli projeler oluşturabilir, bu da Scrum’u popüler hale getirmiştir.

Yararlanılan Kaynaklar

* Deniz Kılınç,” Yazılım Mühendisliğine Giriş Ders Notları”, İzmir Bakırçay Üniversitesi
* <https://www.argenova.com.tr/scrum-nedir>
* <https://tr.wikipedia.org/wiki/Scrum>
* <https://tr.wikipedia.org/wiki/V-Model_(Yaz%C4%B1l%C4%B1m_geli%C5%9Ftirme)>
* <https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742>

Ferhat Zeki Mataraci  
210601065

İzmir Bakırçay Üniversitesi