Bu makalede sizlere yazılım yaşam döngü modelleri hakkında elimden geldiğince bilgi vermeyi amaçlıyorum. Öncelikle yazılım nedir ve neden bu kadar önemlidir ? Yazılım , elektronik aygıtların görevlerini yapmasını , haberleşmesini sürekli etkileşim içinde bulanabilmesini sağlayan komutlara verilen isimdir. Yazılım fark etmesek de hepimizin için ciddi bir öneme sahiptir ve birçok yerde karşımıza çıkabilir. En yaygın yazılım türleri; masaüstü yazılımlar, mobil yazılımlar ve web yazılımlardır. Yazılımların en büyük önemi, az zamanda tamamlanması olası olmayan işlerin, bilgisayar tarafından otomasyon ile yapılmasını sağlamaktır.

**Yazılım Yaşam Döngüsü**

Üzerinde çalıştığınız bir yazılım projesinin planlamasından ,analizinden başlayarak teslimatına, bakımına kadar geçirdiği evrelere ve bu evrelerde oluşan döngüye, Yazılım Yaşam Döngüsü denir. Bir proje geliştirilirken projenin planlama, analiz, tasarım, üretim ve test aşamaları yer almaktadır. Bu aşamalardan sonra ise bakım evresi vardır , olası bir hata, bozulma söz konusu sürekli bakıma dönüş olacağı ve ardından aynı evrelerden geçileceği için hepsi bir arada birbiriyle döngü içerisinde düşünülmelidir. Şimdi bu adımlara göz atalım.

**Planlama**: Bu aşama yazılım yaşam döngüsünün ilk adımıdır. Bu adım ile uygulamadaki ilk fizibilite testleri yapılır. İhtiyaçlar , imkanlar gözden geçirilir. Personel ve kaynak durumları ile ilgili değerlendirmeler yapılır.

**Analiz**: Bu aşamada oluşturulacak sistemin işlevleri belirlenir, sistem ile ilgili sorunlar listelenir ve çözüm sistemleri araştırılır. Bu adımda müşteri, ürün yöneticisi, yazılım mühendisi gibi işi direkt ilgilendiren bireyler arasında toplantılar yapılır ihtiyaçların net bir şekilde anlaşılıp anlaşılmadığı gözden geçirilir.

**Tasarım**: Sistemdeki tüm gereksinimler ve planlamaların tamamlanması ile birlikte tasarım aşamasına geçilir. İhtiyaçlara karşılık veren bilgi ya da yazılım sistemi kurulur. Gereksinimlerin müşterinin isteklerini karşılaması beklenir. Ara yüzler ve sistemin belirlenir. Tasarım mantıksal ve fiziksel olarak ikiye ayrılır.

**Üretim**: **“**Kodlama, Test, Kurulum**”**bu aşamada yapılır. Tasarımdan sonra kodlama aşaması vardır. Kodlamadaki amaç temiz ve okunabilir kod oluşturmaktır.

**Test**: Buradaki test tüm sistemi etkileyecek olan genel sistem testidir. Çünkü test her aşamanın sonunda yapılır, her aşama kendine ait test kısmı içerebilir. Buradaki test işlemi sistemin tamamına yapılmakta genel problemlerin tespiti ve giderilmesi üzerinde çalışılmaktadır. Bu aşamadan sonra ürün teslim edilir.

**Bakım**: Bakım aşaması yazılımın kullanıldığı var olduğu süre boyunca devrededir aslında. Kalite bakımın bir sonucu olduğundan bakımın kalitesi uygulamanın kalitesi ile doğru orantıda olacaktır.

**Yazılım Süreci Modelleri**

1. **Gelişigüzel Model:** Aslında tam olarak bir model değildir belirli bir gidişatı yoktur. Kişiye bağlı yazılım şeklinde yapılır , yeteri kadar doküman olmazsa izlenebilirlik ve bakım oldukça zor hale gelir. Basit programlar için genellikle tek bir kişinin izlediği yöntemdir.
2. **Çağlayan Modeli:**

Sistem ve Yazılım

Tasarımı

Gereksinimlerin

Tanımlanması

Birleştirme ve Sistemi test etme

Sistemin Bakım ve İdamesi

Kodlama ve Modül

test etme

Yaşam döngüsü temel adımları bu modelde baştan sona en az bir kez gerçekleştirilir . Kısa zamanda tamamlanması gereken projeler için uygun bir modeldir. Yazılım tanımlamada belirsizlik yok ise ve yazılım üretimi çok zaman almayacak ise uygun bir süreç modelidir.

1. **V Süreç Modeli**: Sol taraf üretim, sağ taraf sınama işlemleridir.

V süreç modelinin temel çıktıları;

Kullanıcı Modeli

Geliştirme sürecinin kullanıcı ile olan ilişkileri tanımlanmakta ve sistemin nasıl kabul edileceğine

ilişkin sınama belirtimleri ve planları ortaya çıkarılmaktadır.

Mimari Model

Sistem tasarımı ve oluşacak alt sistem ile tüm sistemin sınama işlemlerine ilişkin işlevler.

Gerçekleştirim Modeli

Yazılım modüllerinin kodlanması ve sınanmasına ilişkin fonksiyonlar.

1. **Barok Modeli:** Yaşam döngüsü temel adımlarının doğrusal bir şekilde uygulandığı bir modeldir.

Belgeleme bu modelde diğerlerine istinaden daha önemlidir.

Belgelemeyi ayrı bir süreç olarak ele alır, ve yazılımın geliştirilmesi ve testinden hemen sonra yapılmasının öngörür.

1. **Evrimsel Geliştirme Süreç Modeli:** Geniş alanlara dağılarak yapılacak projeler için önerilir. Her aşamada üretilen ürünler, üretildikleri alan için tam işlevselliği içermektedirler. Bu modelde kullan, test et, güncelle prensibi vardır.
2. **Artırımsal Geliştirme Süreç Modeli:** Üretilen her yazılım sürümü birbirini kapsayacak ve giderek artan sayıda işlev içerecek şekilde geliştirilir.Uzun zaman alabilecek ve sistemin eksik işlevlikle çalışabileceği türdeki projeler bu modele uygun olabilir.
3. **Helezonik Model:**

Planlama

Üretilecek ara ürün için planlama, amaç belirleme, bir önceki adımda üretilen ara ürün ile bütünleştirme

Risk Analizi

Risk seçeneklerinin araştırılması ve risklerin belirlenmesi

Üretim

Ara ürünün üretilmesi

Kullanıcı Değerlendirmesi

Ara ürün ile ilgili olarak kullanıcı tarafından yapılan sınama ve değerlendirmeler.

Her döngü farklı bir fazdır. Prototip ve artımsal bir yaklaşım vardır.

Gelişi güzel modelin izlenebilirliği ve bakımı çok zordur eğer tek kişilik bir çalışma değilse bu model tavsiye edilmez. Barok modeli ise gerçekleştirim aşamasına çok fazla ağırlık vermiştir günümüzde çok kullanılmamaktadır. Çağlayan modeli ise Barok modeline göre daha iyi geri dönüş verir. Fazla zaman almayacak tanımlanması sıkıntılı olmayan işler için biçilmiş kaftandır. V süreç modeli ise kararlaştırılamayan konuların olmadığı , işlerin kesin bir şekilde seçilmiş olduğu BT projeleri için oldukça uygundur. Spiral yöntem çağlayana kıyasla daha büyük, kapsamlı projelerde kullanılır.

**Neden Scrum ?**

[Scrum](http://www.acm-software.com/acmblog/scrum-nedir/), kelime olarak rugby oyununda oluşturulan küçük ekiplere verilen isimdir. Bu yöntem 90’li yıllarda oluşturulmuş, günümüze sürekli bir gelişme halinde gelmiştir.

[Scrum](http://www.acm-software.com/acmblog/scrum-nedir/), ilk bakışta çok basit kuralları olan bir yönetimsel modeldir. Gereksinimleri açıkça belirli olmayan, değişime açık, karmaşık yazılım projelerinin yönetimi için uygulanmaktadır. [Scrum](http://www.acm-software.com/acmblog/scrum-nedir/), detaylı bir şekilde projede izlenmesi gereken adımları belirtmemekte, onun yerine basit ama önemli birkaç olmazsa olmaz kuralıyla esnek bir yönetim sunmaktadır.

[Scrum](http://www.acm-software.com/acmblog/scrum-nedir/)’ın sunmakta olduğu en önemli çıktı sürecin şeffaf bir şekle getirilerek süreç içerisinde aksayan noktaların açığa vurulmasıdır. [Scrum](http://www.acm-software.com/acmblog/scrum-nedir/) böylelikle proje ekibini ortaya çıkan aksaklıkları çözümleyerek sürekli iyileştirme yapması yönünde motive eder. Açıkçası scrum için bir tablo çizmek doğru olmaz. scrum çünkü belli başlı toollardan oluşan bir süreç değildir. aksine kendi içinde evrimleşebilen gelişebilen ve ihtiyaçlar doğrultusunda esnekleşebilen bir yapısı vardır. Scrum meetingler çok önemldir, çalışanın önemi burada ortaya çıkar. Eğer doğru şekilde uygulanabilirse bir şirkete gerçekten çok fazla verim , kalite katacağını düşünmekteyim. Müşterilerin ihtiyaçları ön plandadır. Takım çalışmasının önemini gösterir. Az kişiyle kısa süreli toplantılarla yapılmalıdır. Sorunlar planlar gidişat paylaşılmalıdır. Başarıyı yakalamak, rakebet edebilmek için scrum uygulanması gereken bir yöntemdir. Çünkü teknoloji takipçisi, yenilikçi, büyük firmalar scrumla gerçek başarıya ulaşabilmişlerdir.

Yiğit Ali TUFAN

180601047