

## YAZILIM YAŞAM DÖNGÜSÜ

Yazılım kavramını açıklamak gerekirse elektronik cihazların hangi işlevleri yerine getirmesini söyleyen , kullanıcının cihazı yönetebilmesini sağlayan komutlar dizisidir. İnsanlar ile elektronik donanımlar arasındaki bir yardımcı gibi düşünebiliriz. Günümüzde hayatın hemen her alanında yazılım teknolojileri bulunmaktadır. Akıllı telefonlar , bilgisayarlar , ulaşım araçları, elektronik ev eşyaları gibi daha devam eden yüzlerce örnek sayabiliriz. Hayatımızla artık bu kadar iç içe olan bu yazılım teknolojisini oluşturmak zor ve karmaşık olabileceği için birçok metod ve prensipler geliştirilmiştir. Bunlara yazılım geliştirme yaşam döngüleri adı verilmektedir. Gereksinimlerin doğru belirlenmesi en uygun bütçeyle en verimli en işlevsel yazılımın oluşturulması için yazılımdaki hataları en aza indirmek ve daha sonrasında bakımını kolaylaştırmak için bu yazılım yaşam döngüleri adı verilir. Yazılım yaşam döngülerinin temel basamakları vardır. Bunlar gereksinim , analiz , tasarım , gerçekleştirme ve bakımdır. Bu aşamalar kesinlikle tek yönlü ve doğrusal olarak düşünülmemelidir.

### 1.GEREKSİNİM AŞAMASI:

Müşterinin istekleri belirlenir. Yazılım oluşturma sürecinde çalışacak personel ve kullanılacak donanımlar belirlenir. Fizibilite çalışmaları yapılır.

### 2.ANALİZ:

Temel sorunların ortaya çıkarıldığı , ne gibi risklerle karşılaşılacağı belirlenir. Sistemin detaylı bir biçimde incelenir. Gereksinimler tekrar gözden geçirilir. Gerekirse müşteri ve yazılımı oluşturan grup birlikte bir değerlendirme yapar.

### 3.TASARIM:

Gereksinimler göz önüne alınarak yazılım nasıl yapılacağı tasarlanır. Yazılımın temel yapısı oluşturulur. Akış diyagramı , ekran tasarımları gibi işler yapılır. Mimari tasarım ve detaylı tasarım olarak iki aşamaya ayrılır.

### 4.GERÇEKLEŞTİRME:

Artık kodlama işlemine başlanır. Tüm yazılım ekibinin anlayabileceği bir şekilde temiz kod yazmak önemlidir. Ayrıca bu sahfada yazılan kodlar belli aralıklarla test edilerek ilerde oluşabilecek hataların önüne geçilir ve gereksiz masraftan kurtarır.

### 5. BAKIM:

Yazılım ürünü müşteriye teslim edildikten sonra oluşabilecek hataların çözümü ve yeni gereksinimler için yazılımın güncellenmesi gibi işlemlerin gerçekleştirildiği bir aşamadır.

## YAZILIM YAŞAM DÖNGÜLERİ

### 1-GELİŞİGÜZEL MODEL:

Gelişigüzel model kişiye göre değişebilen bir modeldir. Net bir yöntem veya prensiplerden oluşmaz. Daha basit programlarda kullanılabilir. Sürecin takibi ve bakımını yapmak zordur.

## 2-BAROK MODEL:

Barok model sürecindeki adımlar doğrusal bir şekilde ilerler. Adımlar arasında nasıl geri dönüş yapılabileceği belli değildir. Modern modellere göre en temel farkı dökümantasyonu ayrı bir süreç olarak ele alır. Günümüzde pek kullanılmamaktadır.

## 3-ÇAĞLAYAN ŞELELE MODELİ:

Geleneksel model olarak bilinir. Popülerliğini kaybetmektedir. Gereksinimler çok net olmalı ve kısa süren bir proje olmalı. Aksi takdirde bu modeli uygulamak işi zorlaştırır. Barok modelden farklı olarak dökümantasyonu ayrı bir süreç olarak görmez, aşamalar en az bir kere tekrar edilir. Aşamalar doğrusal bir şekilde izlenir. Bir aşama tamamlanmadan diğer aşamaya geçilmez. Değişime fazla elverişli değildir. Yeni gereksinimler için uygun olmayabilir. Günümüzde eskisi kadar popüler olmasa da yine de kullanılmaktadır. Gereksinimleri iyi belirlenmiş projelerde kullanılabilir. İş dağılımı ekip için kolay olur. Bazı istenmeyen yanları da vardır. Müşteri yazılım üretilirken sürecin içinde bulunmadığı için süreç sonunda tam olarak istediği ürüne ulaşamayabilir. Müşterinin istekleri doğru belirlenmezse yanlışların düzeltilmesi zordur ve maliyeti artırır.

## 4-V SÜREÇ MODELİ

Çağlayan yaşam döngüsünde olduğu gibi doğrusal bir ilerleme yoktur. Süreç V harfine benzeyecek şekilde ilerler. Sol tarafında üretim sağ tarafında ise test işlemleri yapılır. Sağ tarafta belirlenen hata onun hizasına denk gelen sol taraftaki iş sürecinde bir sorun olduğunu göstermektedir. Hatalar böylece erkenden bulunur. Zaman ve bütçe kaybından kurtarır. Gereksinimlerin eksiksiz bir biçimde belirlendiği projeler için uygundur. 3 temel çıktısı vardır.

### Kullanıcı Modeli:

Yazılım sürecinin kullanıcıyla ilgili olan kısımlarını oluşturur.

### Mimari Model:

Sistemin sınaama işlevlerini oluşturan bölümdür.

### Gerçekleştirim Model:

Yazılım parçalarının kodlanması ve test edilmesi gibi işlevleri barındırır bölümdür.

## 5-HELEZONİK (SİRAL) MODEL:

İçten dışarıya açılan döngülerden oluşan bir modeldir. Her döngü ayrı bir aşamayı temsil eder. Yinelemeli artımsal yani yeni şeylerin eklenerek sürecin ilerlediği bir modeldir. Planlama , risk analizi üretim ve kullanıcı değerlendirmesi gibi 4 kriterden oluşur. Her döngü bu kriterlere uğramak zorundadır. Risk analizi olgusu ön plandadır. Her aşama öncesi risk analizi yapılarak olası hataların önüne geçilir. Müşteri önceki modellerde olduğu gibi ürünü en son

görmez. Prototip ürünleri kullanma imkanı bulur. Hataların erkenden giderilmesi , kodlama işleminin erkenden başlaması avantajlarındandır. Dökümantasyonun fazla olması , sözleşme tabanlı projelere uymaması ve döngülerin karmaşıklığı bu modelin dezavantajlı yönleridir.

## 6-ARTIMSAL GELİŞTİRME SÜREÇ MODELİ:

Yazılımı tek seferde müşteriye iletmek yerine parçalara ayırarak üretim yapılır. Her parça için gereksinimler belirlenir. Bir parça kodlanmaya başlandığında gereksinimler artık alınmaz. Yeni gereksinimler olursa bir sonraki parçada eklenir. Her parça kendisinden bir önceki parçayı kapsayacak şekilde büyür. Yazılım geliştirme sürecinde hem ürünün kullanılabilirliği hemde üretilmesi sağlanmış olur. Her parçada müşteriden geri dönüşler alındığı için yazılım çok verimli bir hale gelir. Bazı işlevleri olmasada temel omurgayla çalışabilecek yazılımlar için ve uzun soluklu yazılımlar için uygun bir modeldir.

## 7-KODLA DÜZELT MODEL:

Belgelendirme işlemlerinin bulunmadığı modeldir. Bu yüzden bakımını yapmak zordur. Hemen yazılım ürünü geliştirilir. Direkt olarak kodlama işine başlandığı için en kolay gerçekleştirilen yazılım modelidir. Ancak bakımı ve değiştirilmesi zor olduğu için pahalı olabilecek bir modeldir. Çoğunlukla küçük projelerde kullanılır.

## ÇEVİK YAZILIM GELİŞTİRME:

Yazılım sektörünün gün geçtikçe daha çok alana yayılmalı ve popülerliğinin artması sonucunda bu sektör hızla büyümektedir Bu büyümeyle beraber yazılım talepleri artıyor ve istenilen yazılımlar da daha kompleks hale gelmeye başlamaktadır. Yazılım mühendislerinin diğer birçok mühendislik dalıyla birlikte uyumlu bir ekip çalışması yapması artık kaçınılmaz bir hale gelmiştir. Bunun üzerine çevik yazılım geliştirme metodları geliştirilmiştir. Bu metodların amacı daha hızlı bir şekilde yazılım üretmek , hata oranını en aza indirmek ve yeni gereksinimleri hemen yazılıma entegre edebilmektir. Çevik yazılım geliştirmede yazılım iterasyon adı verilen parçalara ayrılır. Her iterasyon süreci 2 ila 4 hafta sürmektedir. Her iterasyon bittiğinde müşteriye bu konuda bilgi verilmekte varsa yeni gereksinimler not edilmektedir. Projenin böyle küçük iterasyonlara ayrılması daha sonra bir hata farkedildiğinde hatayı tüm yazılımda düzeltmek yerine iterasyonda düzeltmemizi sağlar. Böylece hatalar kısa sürede giderilir. Çevik yazılım geliştirmede ekibin birbiriyle sürekli iletişim halinde olmasında yazılım sürecini hızlandıran faktörlerden birisidir. Çevik yazılım geliştirme metodlarının en popülerlerinden biri olan scrumu şimdi inceleyelim.

## SCRUM:

Scrum sadece yazılım geliştirmede değil bir ekiple oluşturulan her projenin yönetiminde kullanılabilir. 3 temel ilkesi bulunur.

Şeffaflık: Yazılım projesindeki süreç yazılım ekibi , müşteri ve ekip yöneticisi tarafından rahatlıkla anlaşılabilir olmalıdır.

Denetleme: Yazılım projesinin sprint adını verdiğimiz parçaların düzenli olarak kontrol edilmesidir.

Uyarılama: Yazılımın gereksinimleri toptan belirlenmez her sprint için ayrı ayrı belirlenir.

Scrum uzun soluklu , karmaşık ve gereksinimlerin net bir şekilde belirlenemediği kordineli bir ekip çalışması isteyen projeler için çok uygundur. Yazılım projesi iterasyon adı verilen küçük parçalara ayrılır ve bu parçalar 2 ila 4 haftada bitirilmesi beklenir. Her sprint aşaması sonucunda bazı fonksiyonlarını yerine getirebilen yazılım omurgası teslim edilmiş olur. Scrum üç temel kavramdan oluşur.

### 1-Roller:

Ekip rolleri de kendi içerisinde 3'e ayrılır. Müşteriyi temsil eden product owner , yazılım ekibini organize eden scrum master , ve sürekli iletişim halinde olan ekip yani scrum team. Scrum da herhangi bir proje yöneticisi bulunamaz. Scrum alanında uzmanlaşmış scrum expert bulunmalıdır. Aksi halde scrum doğru bir şekilde uygulanamaz.

### 2-TOPLANTILAR:

Ekip üyeleri günlük olarak toplantılar yapar. Sürekli iletişim halindedirler. Bu toplantılarda gereksinimler belirlenir , görev dağılımı yapılır , olası riskler saptanır , hedefler belirlenir.

### 3-BİLEŞENLER:

Bileşenlerde kendi arasında 3'e ayrılır. Bunlar ürün gereksinim dökümanı , sprint dökümanı ve sprint kalan zaman grafiğidir. Gereksinim dökümanı statik değil dinamik bir dökümandır. Değişime açıktır. Sprint dökümanı ise her sprintin ayrıntılarının , hedeflerinin belirlendiği belgedir. 2-4 haftalık süreçte yapılacak işleri kapsar. Sprint kalan zaman grafiğinde ise projenin ne kadarının bittiği ne kadar zaman kaldığı hangi aşamada zaman kaybedildiği gibi saptamalar yapılır, zamanın etkin kullanılması için gereklidir.

Scrum metodu ekip üyelerinin birbiriyle uyumunu artırır , zor ve karmaşık gözükten projeleri kolaylaştırır , hata oranını en aza indirir ve belirlenen hataların daha kısa sürede düzeltilmesini sağlar. Scrum bu yüzden günümüzde oldukça popüler hale gelmiştir.

### KAYNAKÇA:

- 1.<https://medium.com/@denizkilinc/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-temel-a%C5%9Famalar%C4%B1-software-development-life-cycle-core-processes-197a4b503696>
- 2.<https://tr.wikipedia.org/wiki/Scrum>
3. Dr. Öğr. Üyesi Zekeriya Anıl GÜVEN hocanın ders sunumları
4. <https://medium.com/@omerharuncetin/yaz%C4%B1l%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BC-modelleri-543c7879a742>
- 5.<https://akademiksunum.com/index.jsp?modul=document&folder=a93e3a2fccf8eb56a557c55c5f0d5cf10789abe2>

HAZIRLAYAN: Atakan BAYRAM 210601695

Medium hesabım: <https://medium.com/@atakanbayram45>

Github hesabım: <https://github.com/AtakanBayramceng>

Linkedin hesabım: <https://www.linkedin.com/in/atakan-bayram-b5961326b/>