

## YAZILIM YAŞAM – DÖNGÜ MODELLERİ(Software Life-Cycle Models)

Yıldırım ALADAĞ

İzmir Bakırçay Üniversitesi

Yazılım yaşam-döngüsü ,bir yazılımın gerçekleştirilmesi (üretim)ve bakımı süresince(kullanım) gerçekleşen tüm aşamalardır. Bir yaşam süreci vardır bu yüzden döngü biçimindedir. Tek yönlü ve doğrusal ilerlemez ve aşamalıdır. Bu aşamalar şunlardır:



1-Planlama: Yazılım projesinin planlandığı, gereksinimlerin(Requirements)çıkarıldığı, fizibilite çalışmasının yapıldığı aşamadır.

2-Analiz(Analysis)-Çözümleme :Yazılım gereksinim ve işlevlerinin ayrıntılı incelendiği kısımdır. İncelemede temel sorunlar ortaya çıkarılarak riskler belirlenir. Temel UML diyagramlarıda (Use Case,Activity,Class diagram vs.)çizilmeye başlanır.

3-Tasarım(Design):Belirlenen gereksinimlere karşılık verecek yazılım sisteminin temel yapısının olduğu aşamadır. Üst seviye ve mimari tasarım:Sistemin yapısı(modüller,akış şemaları...).Detaylı tasarım:Yazılım bileşenleri ve ayrıntıları(veri yapıları,ekran tasarımları...).

4-Gerçekleştirim(Implementation):Yazılım projesinin kodlandığı,test edildiği ve kurulumunun(testing and integration) yapıldığı aşamadır.

5-Bakım(Maintenance): Yazılım projesinin ürünleştikten sonra güncellemeler yapılır.Hata giderme ve yeni gereksinimlerin eklendiği(teslimden sonra) aşamadır.

### YAZILIM YAŞAM DÖNGÜ MODELLERİ

Birden fazla model bulunur.Bunun temel nedeni;proje gereksinimleri,proje büyüklüğü,yazılım projesinin kimlere hitap edeceği ,talep vs.Bu modellerin temel amacı,proje başarısı için mühendislik süreçlerini daha açık ve net hale getirmektir.Yaşam döngü modelleri:

1-Gelişigüzel Model

2-Barok Modeli

3-Çağlayan/Şelale(waterfall) Model

4-V Süreç Modeli

5-Evrimsel Model

6-Spiral(Helezonik) Model

7-Artımsal Geliştirme Süreç Modeli

8-Kodla ve Düzelt Modeli

- 9- Formal Sistem Geliştirme
- 10- Bileşen Tabanlı Geliştirme
- 11- Araştırma Tabanlı Model
- 12-Çevik Modeller

Bazı modellerden bahsedelim:

### Gelişigüzel Model

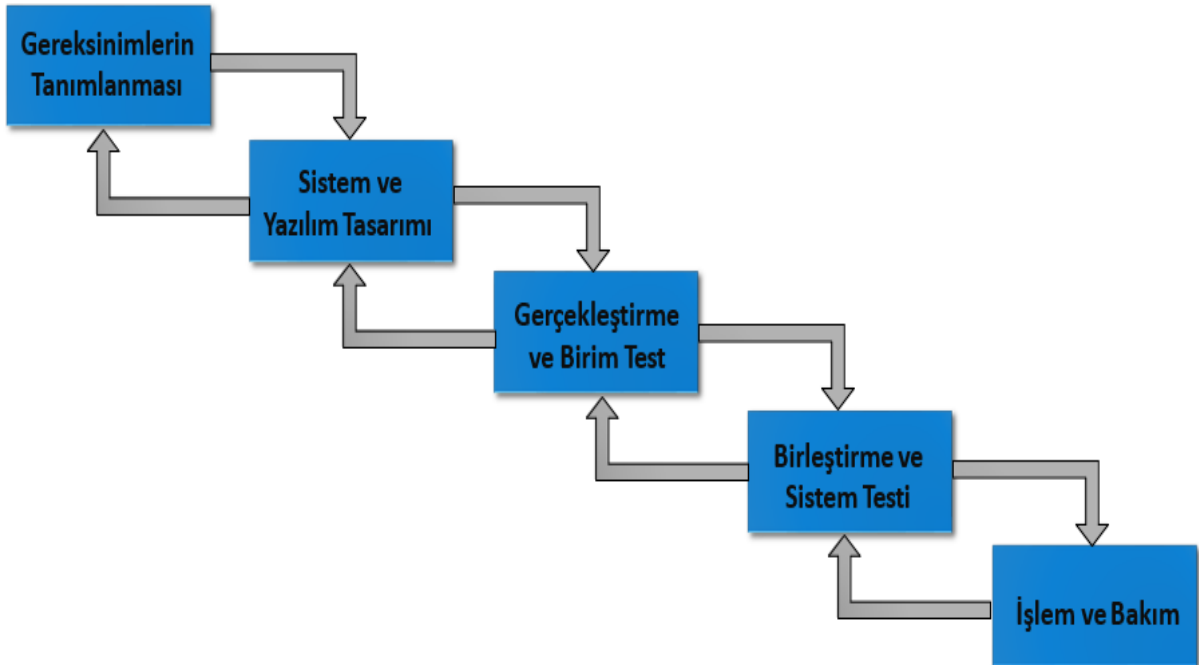
Model olarak tanımlamak doğru değildir çünkü kuraldan bağımsızdır. Belirli bir yöntemi ve modeli yoktur. Kişiye özel yazılım geliştirilir. Denetlenmesi ve bakımı oldukça zordur. .Bu model, yazılıma yeni başlayanların kullandığı adımlara verilen model adıdır. 1960'larda kullanılmıştır. Günümüzde kullanımı yok denecek kadar azdır. Güncel değildir.

### Barok Modeli

Barok modelde yazılım yaşam-döngüsü adımları doğrusal ilerler ve bu adımlar geliştirilir. Döngü yoktur. Aşamalar arasında geri dönüşlerin nasıl yapılacağı bilinmez. Dokümantasyon günümüz modellerinden farklı bir şekilde ayrı bir süreçte işler. 1970'lerde kullanılmıştır. Günümüzde kullanımı yok denecek kadar azdır. Güncel değildir.

### Çağlayan/Şelale(waterfall) Model

Geleneksel bir model olarak bilinir ve geçmişte en popüler döngü modeliydi. Bu nedenle diğer güncel modellerin temelidir diyebiliriz. Bu modelde aşamalar en az birer kez tekrarlanır. Üretimi çok iyi tanımlanmış, az zaman gerektiren, gereksinimleri belli projeler için idealdir. Günümüzde kullanımı azdır.



Yazılım Mühendisliğinde yaygın olarak kullanılan ilk SDLC Modelidir. Aşama aşama yapılır. Bir aşama bittirmeden diğer aşamaya geçilmez. Her aşamada dokümantasyon bulunmalıdır. Eğer aşamada dokümantasyon ve test yoksa aşama bitmiştir denemez.

Barok modelin aksine belgeleme işlevini ayrı bir aşama olarak ele almaz ve üretimin doğal bir parçası olarak ele alır. Barok modele göre geri dönüşler daha açık ve işlevseldir. Modelin tasarım aşaması için detaylar incelenmeli ve sistemin gereksinimleri iyi anlaşılmalıdır. Proje üretimi zaman aldıkça gereksinimlerde değişecektir (Değişen hedef problemi). Pek çok avantaj sağlar bunlar:

Kullanıcı ve müşteri için anlaşılır adımlardan oluşur.

Gereksinimler sağlam bir temel oluşturur.

Proje yöneticileri için iş bölümü yapmak kolaylaştırır.

İterasyonlar (tekrarlamalar) sonraki ve önceki aşamalarda gerçekleşir.

Her model gibi bunda da dezavantajlar mevcuttur:

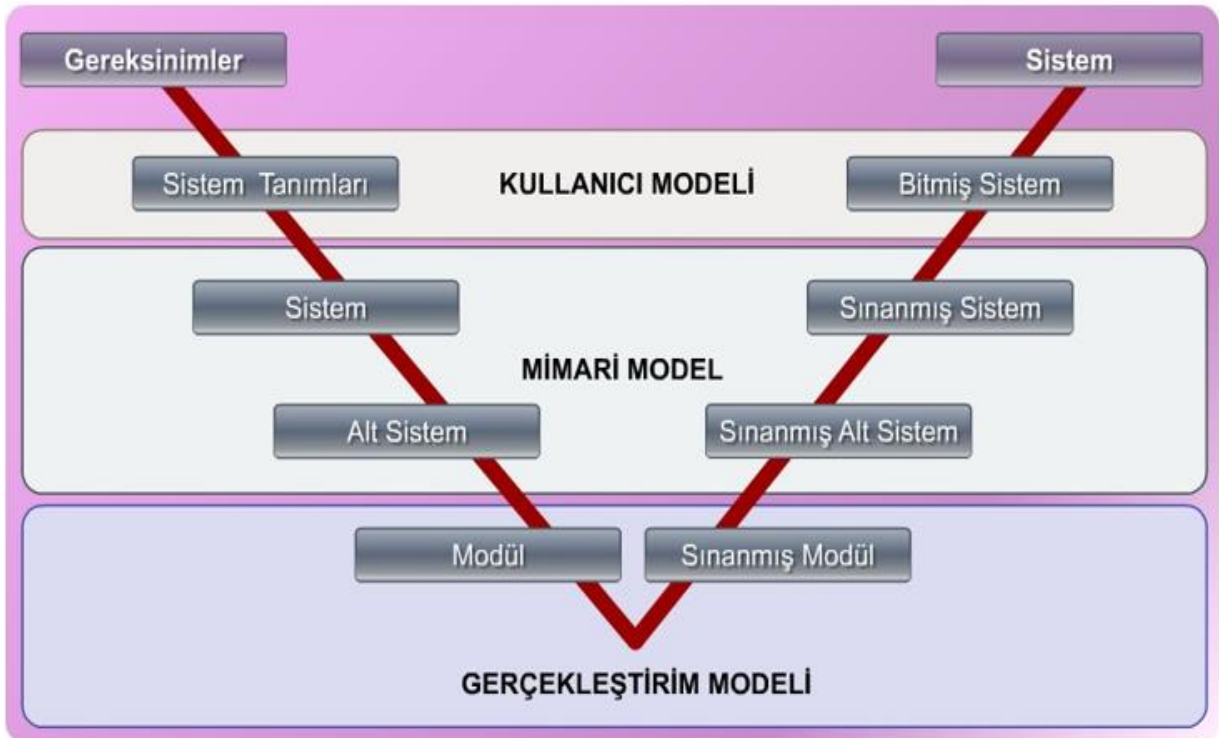
Uzun bir zamana yayılmıştır.

Kullanıcı sürecin içinde yoktur fakat yazılım tamamlandıktan sonra geri dönüşleri fazla olabilir.

Gereksinimler iyi tanımlanmadığı takdirde yanlışları düzeltilme ve eksiklerin giderilme maliyeti yüksektir.

### V Süreç Modeli

Bu model Çağlayan modelden daha gelişmiştir. Sol taraf Üretim ve sağ taraf Sınama (Test) kısmından oluşur. Bunlarda içinde bulunan 3 modele birlikte daller. Kullanıcı Modeli, Mimari Model, Gerçekleştirim Modeli. Bu model açık ve anlaşılır olan iş tanımlarının ise belirgin olduğu projelerde kullanılır. BT projeleri için uygundur.



İş analizi ve kabul sınamalarının tanımları yapılmaktadır. Bu modelin avantajları:

Model, kullanıcıyı aşamalara katar böylece kullanıcının projeye katkısı artırılmış olur.

Proje yönetimini ve denetiminde kolaylık sağlar.

Verification ve validation planları erken aşamalarda uygulanır.

Kullanımı kolaydır.

Dezavantajları :

Fazlar arasında tekrarlama yoktur.

İş ve ürün gereksinimleri süreç içerisinde değişkenlik gösterebilir.

Aynı anda gerçekleşebilecek olaylara zorluk çıkarır.

## Çevik Modeller

Çevik modeller yazılım projelerindeki başarıyı artırmak ve ekip içi iletişimi güçlendirmek amacıyla ortaya çıkmıştır. Bazı çevik yazılım geliştirme modelleri şunlardır:

### 1-Uç Değer Programlama-Extreme Programming (XP)

Aşırı programlama yani XP, programlama yapmadan önce testler yazarak kaliteyi ön plana çıkarır. Tüm yazılım bir araya getirilir ve birkaç kez test edilir. Bu, yazılımın çökertmemesini sağlamak için yapılır. Aşırı programlamanın bilinen yönlerinden biri, iki programcının bir bilgisayarda birlikte çalışır ve çift programlama oluşur. Bu yaklaşım sayesinde yazılımın yaratılmasına giren düşünce süreci, tasarlama ve kodlama anında kontrol edilir. 4 temel maddeden oluşur: Basitlik, Cesaret, Geri Dönüş, İletişim.

**Basitlik:** Yazılan kodun ve yapılan işin sade ve anlaşılır olmasıdır. Uzun dokümantasyondan uzak durur .

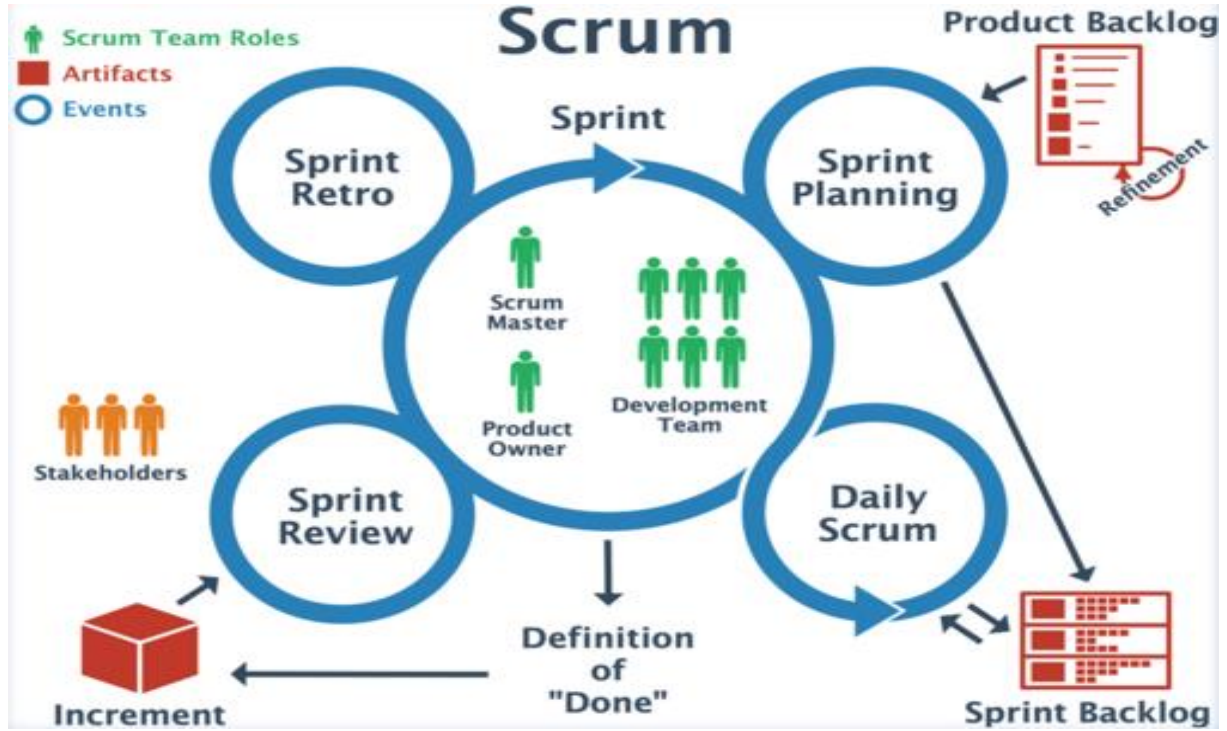
**Cesaret:** Proje aşamasında cesur olunmalı . Bir kodun gerekirse tamamen silinip yeniden yazılmalıdır.

**Geri Dönüş:** Geri dönüşler ile oluşabilecek hatalar azaltılır. Müşteri ile yazılım ekibi birbirleriyle sürekli iletişim halinde olurlar.

**İletişim:** İletişim, projelerde önemli sorunlardan birisidir. XP ise bunu aşmaya çalışmaktadır. Ekip içi iletişime önem verir ve artırılması için çalışır.

2-2-2-

### 2 -SCRUM



SCRUM büyük projeleri parçalara bölerek her birine "sprint" adını verir. Projeyi böler ve her bir sprinti teker teker geliştirir. SCRUM'da ekip içi iletişim çok önemlidir bu yüzden her gün "SCRUM meetings" denen toplantılar yapılır. SCRUM'da üç temel kavram

bulunmaktadır, bunlar: Roller, Toplantılar ve Bileşenler ve Araçlardır.

**Roller:** Ürün sahibi, Scrum yöneticisi ve Scrum takımından oluşur. Ürün sahibi projenin beynidir. Scrum yöneticisi, takımı Scrum etiklerine göre kontrol eder. Scrum takımı ise 5–9 kişiden oluşan birbirleriyle sürekli iletişim halinde olan bir yazılım geliştirme ekibidir.

**Toplantılar:** SCRUM'ın olmazsa olmazı toplantılardır. Her gün SCRUM toplantıları yapılır bu günlük toplantılarda her gün yazılım geliştiricilerinin önceki gün neler yaptıkları karşılaştıkları sorunlar ve bugün neler yapacakları hakkında bir konuşma olur. Hafta değerlendirmeleri yapılır. Her sprint için de bir gözden geçirme toplantısı yapılır.

**Bileşenler ve Araçlar:** Ürün Gereksinim Dokümanı oluşturulur. Bu dokümanda proje boyunca yapılması gerekenler basitçe yazılır. Sprint Dokümanı oluşturulur. Sprint dokümanı ürün gereksinim dokümanına takiben oluşturulur ve amacı her sprintin ona uygun ayarlanmasıdır ayrıca bu dokümanı sadece takımdakiler değiştirebilir. Sprint Kalan Zaman Grafiği ise yapılan işin ne seviyede olduğu ve aslında planlanan zaman göre nerede olduğunu belirlemek için hazırlanır.

#### **SCRUM Yapısı**

1-Product Backlog; Proje için gerekli olan gereksinimler listesidir. Proje sonunda “Ne üretilmek isteniyor?” sorusuna cevap aranır. Product owner tarafından müşteriden gereksinimler alınır, öncelik sırasına göre sıralanır.

2-Product Backlog Item; Product backlog içindeki her bir gereksinime verilen isimdir

3-Sprint; Scrum içerisindeki tüm aktiviteler sprint içerisinde gerçekleşir.

4-Sprint Backlog; Bir sprint boyunca yapılacak itemlerin listesini oluşturur. İşlerin detaylı olarak zaman çizelgesi çıkarılır.

5-Scrum board; Yapılacak olan tasklar “TO DO” bölümüne alınır. Takım üyesi bu işe başladığında “IN PROGRESS” bölümüne getirilir. Bir iş, test için hazırsa “TO VERIFY” durumuna getirilir. İş, kontrol edildikten sonra “DONE” bölümüne getirilir .

6-Burndown Chart; Yatay ekseninde sprintin günlerini, dikey ekseninde sprintte kalan işi gösteren grafikdir.

Rollere gelirse ilk olarak bir ürün Sahibi (Product Owner) olmalıdır

Scrum yöneticisi takımlar ve ürün sahibiyle olan iletişime yardımcı olan, takımın ve organizasyonun Scrum'a adapte olmasını sağlayarak üretkenliğin artmasına katkıda bulunan kişidir.

Scrum Takımı (Scrum Team) , birbirleriyle iletişim halinde olan ve programlama işini yapan gruptur.

Toplantılar: Sprint Planning de sprintlerde neler yapılacağı, gereksinimlerin listesi, takımların belirlenmesi, risk analizleri ve maliyet hesaplamaları yapılır.

Tüm sprint boyunca her gün takım üyeleri bir araya gelerek 15 dakika boyunca ayaküstü toplantılar yaparlar. Bu toplantılara “Daily Scrum Meeting” denir. Bu toplantılarda önceki gün neler yapıldığı, ne gibi sorunlarla karşılaşıldığı, o gün neler yapılacağı ve nasıl yapılacağı hakkında özet şeklinde konuşmalar yapılır.

Sprint Listesi (Sprint Backlog), bir sprint turunda takımın yapması gereken işlerin listesidir.

Sprint Kalan Zaman Grafiği (Burndown Chart), bir sprint turunda görevler yerine getirildikçe yapılan iş ile kalan iş arasındaki bağıntıyı ortaya çıkaran grafikdir.

Scrum günümüzde neden popüler?

SCRUM günümüzde en çok kullanılan yazılım geliştirme yöntemidir. Hatta sadece yazılım geliştirmede değil birçok sistemin geliştirilmesinde de kullanılır. Bunun nedenleri ise Yüksek teknolojiler ve son gelişmelere kolaylıkla uyum sağlayabilmesi

Ekip içi iletişimin yüksek olması ve bununla beraber hataların erken fark edilip düzeltilmesi

Zamandan ve paradan büyük ölçekte tasarruf edilmesi

Kullanıcıdan sürekli geri bildirim gerektirmesi ve bununla beraber sorunların azalması

Diğer yazılım geliştirme metodolojileri gibi yinelemeli olması

Değişen gereksinimlere hızlı bir şekilde tepki vermesi

Scrum, uzmanlık gerektiren ve maliyeti yüksek bir modeldir. Kısa sürmesi, kolay uygulanması ve başarı potansiyelinin yüksek olması nedeniyle günümüzde kullanımı oldukça fazladır. Google, Microsoft, IBM ve Yahoo gibi büyük şirketler tarafından tercih edilmektedir.

#### KAYNAKLAR

1-) <https://www.productplan.com/glossary/general-availability/>

2-) <http://bilgisayarkavramlari.sadievrenseker.com/2013/01/10/sdlc-yazilim-gelistirme-yasam-dairesi/>

3-) Bil 102 Yazılım mühendisliği temeleri-İzmir Bakırçay Üniversitesi 2.ve 3. Hafta sunumları

4-) <https://medium.com/architectural-patterns/yazilim-gelistirme-modelleri-62915545c51e>

5-) [https://www.tutorialspoint.com/sdlc/sdlc\\_v\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm)

İzmir Bakırçay Üniversitesi-Yıldırım ALADAĞ-220601015