220601035 İsmail Düzalan

YAZILIM YAŞAM DÖNGÜ MODELLERİ ve SCRUM

A. YAZILIM YAŞAM DÖNGÜ MODELLERİ

Yazılım yaşam döngüsü (SDLC), bir yazılım ürünü oluştururken, geliştirirken veya değiştirmek için tasarlanan sürece verilen isimdir. Kısaca yazılım geliştirmek için oluşturulmuş bir yoldur. Belli başlı aşamaları vardır. Bu aşamalar modelden modele farklılık gösterebilir. Bunun sebebi her modelin farklı bir işleyiş ve planlama tarzının olmasıdır. SDLC doğrusal bir yol izlemek yerine döngüsel bir yol izler. Bunun sebebi geri bildirimlerin olması, oluşan sorunu çözmek için önceki aşamalara gidilmesidir. SDLC temel olarak 5 aşamadan oluşur.

- **Gereksinim**

Müşterinin gereksinimleri elde edilir ve fizibilite çalışması yapılır.

- **Analiz**

Sistemin gereksinimleri ve işlevleri ayrıntılı olarak incelenir.

Ana sorunlar öğrenilmiş olur.

- **Tasarım**

Yazılımın veya sistemin tasarımı yapılır. Projeleri çizilir.

Karar verilir, seçimler yapılır.

- **Geliştirme**

Sistemin hayata geçirildiği ilk aşama sayılabilir.

Yazılım için kodlama aşaması olarak düşünülebilir.

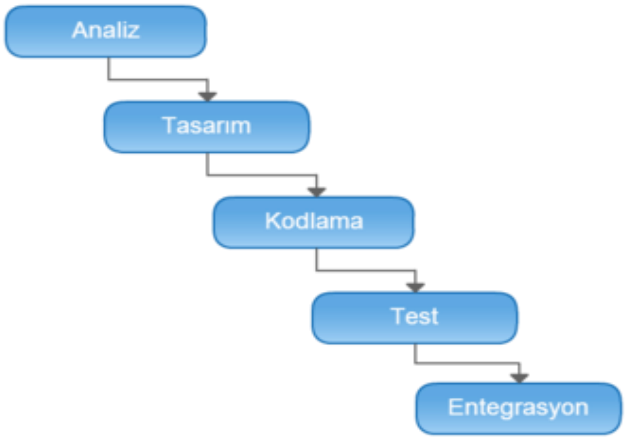
Tasarım aşamasında verilen kararlar doğrultusunda projeye başlanır.

- **Bakım**

Bakım yapılırken farklı aşamalardan geçirilir. Yeni tasarımlar yapılabilir ve bu şekilde döngü devam eder. Yazılımın uzun süre hayatta kalması istenir bunun için de zamanla yazılım geliştirilir ve düzeltmeler yapılmalıdır. Yazılım Geliştirme Yaşam Döngü Modeli tam olarak buradadır. SDLC yapılacak düzenlemeler için yol göstericidir (Seker, 2015).

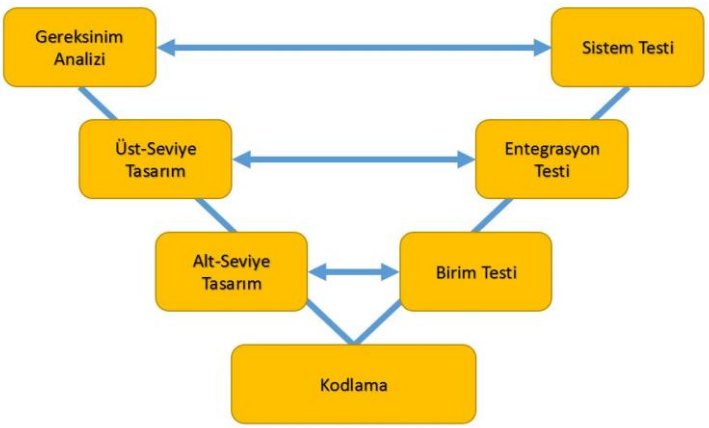
1. **Model: Şelale Modeli**

Test için veya yazılım ürünü oluşturmak için bakılmaksızın "yukarıdan aşağıya " düşüncesini benimser. Bu modelde sadece mevcut adımı tamamladıktan sonra diğer adıma geçebilirsiniz. Aynı anda iki aşama birlikte yapılamaz. Bu model tekrarlamalı olmayan bir yol izler. Avantajı basit ve sistematik olarak yaklaşmasıdır. Dezavantajı ise test aşamasına kadar yapılan hatalar bulunamadığı için pek çok kusuru vardır. Zaman ve para israfına sebep olabilir.



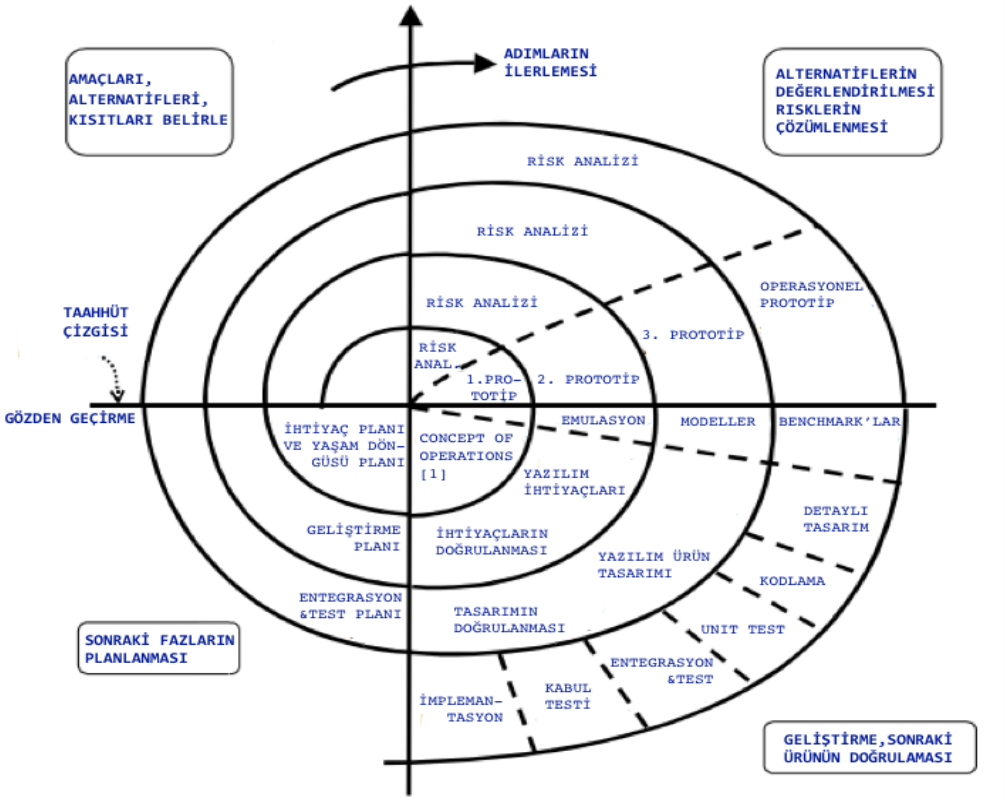
2. **Model: V Model**

Adını grafik gösteriminin 'V' harfine benzemesinden alır. Bu metodolojide yer alan temel adımlar Waterfall modeliyle yaklaşık olarak aynıdır. Farkı ise Çağlayan model sadece " yukarıdan aşağıya" yaklaşımını benimserken bu model hem "Yukarıdan aşağıya" hem de "Aşağıdan yukarı" yaklaşımını benimser. Bu metodolojinin faydası, hem geliştirme hem de test faaliyetlerinin birlikte yürütülebilmesidir. Örnek olarak, geliştirme ekibi gereksinim analizi etkinliklerine devam ettikçe, test ekibi aynı anda kabul test etkinliklerine devam eder. Bundan dolayı zaman gecikmesi en aza indirilir ve kaynakların en uygun kullanımı sağlanır (Nalbant, 2020).



3. Model: Spiral Model

Bir diğer adı sarmal modeldir. Kısa yaşamlara sahip döngülerle yazılım geliştirme tamamlanır. Her döngü bitiminde yazılımın yeni bir versiyonu ortaya çıkar. Amaç riskleri en aza indirmektir. Birinci döngü bittikten sonra, elde edilen ürünün ayrıntılı bir analizi yapılır ve gözden geçirilir. Belirli gereksinimlere uygun değilse ikinci döngüyü takip eder ve böyle devam eder (Nalbant, 2020).



4. Model: Artırımlı Model

İlk aşaması planlamadır. Planlamalardan geçip bu planlamanın ardından bir döngüye girilir. Analizler, tasarımlar yapıldıktan sonra uygulamaya geçilir. Uygulamadan sonra iki durum vardır. Birincisi canlıya geçiş durumudur. Yazılım sistemini kullanmaya hazır hale getiren tüm işlemler yapılır ve sorunlar tüm aşamalarda olduğu gibi bu aşamada da yaşanır. Bu sorunlar testlerin parçası olarak görülebilir. Diğer yandan testler devam etmektedir. Bu sorunlar sistemin gelişmesini sağlar. Sonra tekrar döngü devam eder. Bu sırada planlamada değişiklikler olabilir. (Basili, 2003). Örneğin; tekstil firmasına yazılım yazılıyor olsun. Tekstil firması iç süreçler değişebilir. Web sayfası üzerinden satış yapılıyor olsun. Web sayfası ile ilgili yeni ihtiyaçlar gelmeye başlar. Bir yandan yazılımın çalışan hali ile ilgili bir problemler ortaya çıkar. Bir

yandan da “Orada şu modüle şu ekranı da ekleyelim” gibi fikirler, yeni istekler çıkmaya başlar. İşte bütün bunlar artırımlı modelde her seferinde artırarak iterasyon şeklinde devam eder (Seker, 2015).

5. **Model: Evrimsel Geliştirme**

Genellikle geniş alana yayılmış, birikimi çok organizasyonlar kullanır (banka uygulamaları). İki çeşit evrimsel geliştirme vardır:

- Keşifçi geliştirme

Hedef: Müşterinin gereksinimlerini incelemek için müşteriyle çalışıp son sistemi teslim etmek.

İyi anlaşılan gereksinimlerle başlanmalıdır.

- Atılacak prototipleme

Hedef: Sistemin gereksinimlerini anlamak.

Tam anlaşılmamış gereksinimlerle başlar.

Evrimsel geliştirme modelinde kullanıcı kendi gereksinimlerini daha iyi anlar, sürekli değerlendirme olduğu için ilk aşamalarda hata riski azalır. Dezavantajları ise düzenli teslim edilebilen ürün yoktur, sürekli değişim yazılıma zarar verir ve bakımı zordur.

Hangi Projede Hangi Modeli Kullanmalıyız?

Çağlayan model hükümetler ve büyük şirketler tarafından birçok projenin işleyişinde tercih pedilir. Artırımsal model düşük risk istenen projelerde kullanılır. V model teknik uzmanlığa sahip çok miktarda teknik kaynak mevcut olduğunda kullanılır. Spiral model genellikle uzun ve kompleks projelerin oluşturulmasında kullanılır (Nalbant, 2020).

B. SCRUM NEDİR?

Scrum, 1993 yılında Jeff Sutherland tarafından geliştirildi. Scrum sadece yazılım geliştirmek için değil, her projeye uygulanabilir bir özelliktedir. Scrum kompleks yapıdaki proje içerisinden üretken ve yaratıcı projelerini sunabilecekleri bir programdır. Kesinlikle süreç, tekbik veya metot değildir. Süreç geliştirmeye ve teknik oluşturmaya yarar. Scrum’ın kullanılması kolay ama uzmanlaşılması zordur Scrum takımlardan oluşur ve belirli görevler verilir. Tüm bileşenler belirli bir hedefe hizmet etmeli ki Scrum süreci başarılı şekilde ulaştırabilmelidir. Scrum’da

oluşan roller birbiriyle bağımlıdır. Bir iterasyonun tamamlanması 30 günden fazla zaman almaz. Gündelik kısa süren toplantılarla işin takibi sağlanır (Sönmez, 2019).

Scrum ve Çalışma Mantığı

Karışık yazılımları daha küçük birimlere (sprint) böler. Müşterinin istediği işlevler iki veya dört haftalık dönemler (sprint) arasında geliştirilir ve bir daha gözden geçirilir. Her bir sprint bitiminde müşteriye teslim edilebilir durumda olmalıdır.

Scrum'ın Temel Mantığı

Scrum ın üç temel mantığı vardır:

- Roller

Proje yöneticisi diye bir rol yoktur. Takımın üç farklı yeteneği vardır: Yazılımcı, tasarımcı ve testçi. Proje yöneticisi olmadığı için acemi ekiplerde sorun ortaya çıkar.

- Toplantılar

Kapsamı geniş gereksinim listeleri oluşturulur. Takımlar belirlenir. Risk değerlendirilmesi yapılır. Maliyet hesaplanır.

- Bileşenler
- Ürün Gereksinim Dokümanı
- Sprint Dokümanı
- Sprint Kalan Zaman Grafiği

Scrum Kullanan Bazı Firmalar: Google, Microsoft, Yahoo, İKEA, Siemens, Nokia.

Kaynakça

Nalbant, E. (2020). *Yazılım yaşam döngüsünde testin önemi ve bir test otomasyonunun gerçekleştirilmesi* (Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

Seker, S. (2015). Yazılım geliştirme modelleri ve sistem/yazılım yaşam döngüsü. *YBS Ansiklopedi 2*(3). 18-29.

Sönmez, O. (2019). *Scrum uygulaması için bir yönetim aracı geliştirilmesi* (Yüksek Lisans Tezi). Bahçeşehir Üniversitesi, İstanbul.