

YAZILIM YAŞAM DÖNGÜ MODELLERİ

VE KARŞILAŞTIRILMALARI

200604074 Recep İbrahim Özer

<https://medium.com/@r.ibrahimozer>

<https://github.com/r-ibrahim-ozer/-YAZILIM-YA-AM-D-NG->

MODELLER-VE-KAR-ILA-TIRILMALARI/blob/main/README.md

BAHAR 2022-2023

23 MART 2023

Giriş

Yazılım geliştirme, son yıllarda hızla gelişen teknolojiler ile önemli bir hale gelmiştir. Bu süreçte yazılımın geliştirilmesi, test edilmesi, bakımı ve yönetimi için belirli bir süreç izlenir. Bu süreç, yazılım yaşam döngüsü olarak adlandırılır.

Bu makalede, yazılım yaşam döngüsü modelleri hakkında bilgi verilecektir. SCRUM, V Model, Waterfall ve Spiral modelleri ele alınacak ve her birinin özellikleri, avantajları ve dezavantajları incelenecektir. Ayrıca, bu modellerin karşılaştırıldığı bir sonuç bölümü de yer alacaktır. Bu bilgiler, yazılım geliştirme projelerinde yöneticilerin ve geliştiricilerin doğru kararlar almasına yardımcı olacaktır.

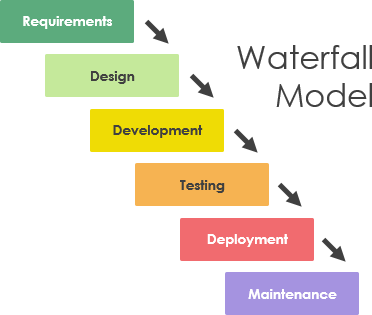
Yazılım Yaşam Döngü Modelleri

1. Şelale Modeli

1.1 Şelale Modeli Nedir?

Şelale metodolojisi, bir projenin başından sonuna kadar doğrusal bir ilerlemeyi vurgulayan bir proje yönetimi yaklaşımıdır. Şelale geliştirme modeli, büyük ölçekte inşaat sektöründe kullanılırken . Yazılım geliştirme için ilk kez benimsendiğinde, hiçbir alternatif olmadığı için benimsenmiştir. Proje faaliyetlerinin lineer ardışık aşamalara bölünmüştür , yani her aşama bir öncekinin çıktılarına bağlıdır. Şelale modeli, yazılım geliştirmede kullanılan en eski yazılım yaşam döngüsü yaklaşımıdır.

Belirlenen her faaliyete ilişkin maliyet ve zaman tahminleri yapılır her aşama belgelenir. Bir Şelale projesinde, parametreler yol boyunca değişirse proje üzerinde değişiklik yapmak oldukça maliyetlidir.



Resim 1.1 Şelale Modeli Süreci

1.2 Şelale Modeli Aşamaları

Bir Şelale sürecinde 5 ortak aşama şunlardır;

* Gereksinimler: Genellikle tek bir belgede yer alan yazılı gereksinimler, maliyetler , varsayımlar, riskler, bağımlılıklar, başarı ölçütleri ve tamamlanma zaman çizelgeleri dahil olmak üzere projenin her aşamasını tanımlamak için kullanılır.
* Analiz: Sistem özellikleri, üretime rehberlik edecek ürün modelleri ve iş mantığı oluşturmak için analiz edilir. Bu aynı zamanda mali ve teknik kaynakların fizibilite açısından denetlendiği zamandır.
* Tasarım: Programlama dili, donanım , veri kaynakları, mimari ve hizmetler gibi teknik tasarım gereksinimlerini ana hatlarıyla belirtmek için bir tasarım belirtimi belgesi oluşturulur .
* Uygulama: Bu aşamada programcılar, uygulamaları proje gereksinimlerine ve özelliklerine göre kodlar, bazı testler ve uygulamalar da yapılır. Bu aşamada önemli değişiklikler gerekiyorsa, bu, tasarım aşamasına geri dönmek anlamına gelebilir.
* Doğrulama veya test etme: Bir ürün müşterilere sunulmadan önce, üründe hata bulunmadığından ve tüm gereksinimlerin tamamlandığından emin olmak için testlerin yapıldığı kısımdır.
* Dağıtım ve bakım: Yazılım piyasaya sürüldüğünde veya müşterilere sunulduktan sonra bakım aşaması başlar. Kusurlar bulunduğunda ve kullanıcılardan değişiklik talepleri geldiğinde, güncellemelerle ilgilenmek ve yazılımın yeni sürümlerini yayınlamak için bir ekip atanır.

1.3 Şelale Modelinin Avantajları

* Kilometre taşlarını ve son teslim tarihlerini açıkça tanımlar.
* Proje durumu, eksiksiz bir program ve kaynak planına dayalı olarak daha kolay ölçülür.
* Proje yöneticileri için iş dağılımı yapmak çok kolaydır.
* Erken sistem tasarımı ve spesifikasyon değişikliklerinin kolayca yapılmasını sağlar.

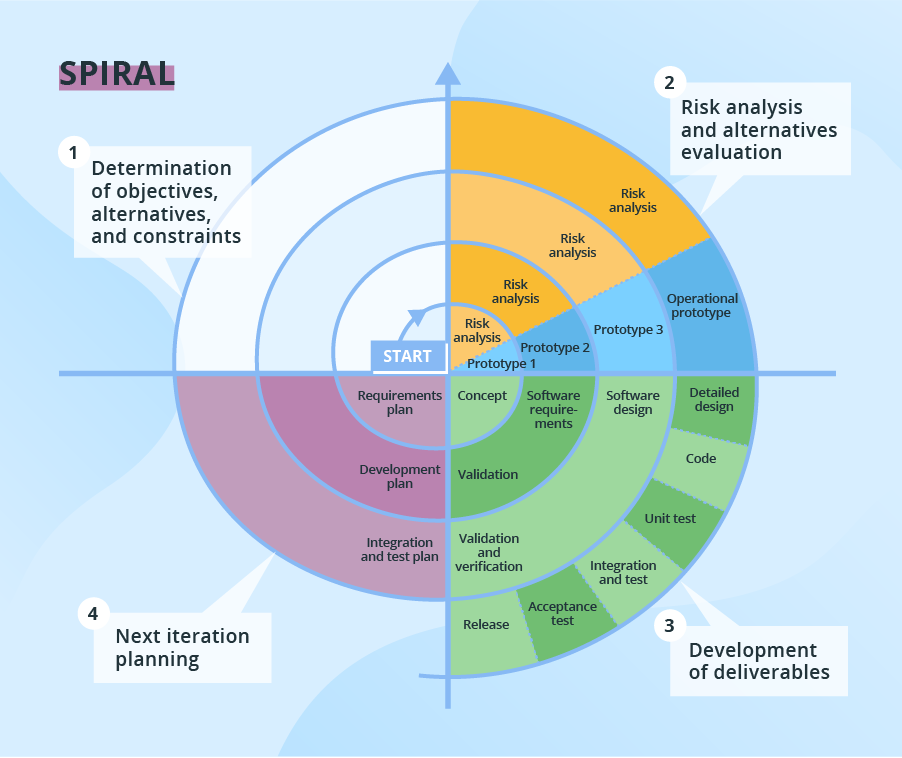
1.4 Şelale Modelinin Dezavantajları

* Tasarım uyarlanabilir değildir; bir kusur bulunduğunda, tüm sürecin genellikle baştan başlaması gerekir.
* Waterfall, farklı aşamalarda eş zamanlı çalışma için süreçlerin çakışmasına izin vermeyerek genel verimliliği düşürür.
* Proje yaşam döngüsünün sonraki aşamalarına kadar hiçbir çalışan ürün mevcut değildir.
* Şelale, karmaşık ve yüksek riskli devam eden projeler için ideal değildir.
* Müşteriler tasarım ve uygulama aşamalarına dahil değildir.

2. Spiral Modeli

2.1 Spiral Modeli Nedir?

Spiral model, risk odaklı bir yazılım geliştirme süreci modelidir. .Prototip oluşturmanın yinelemeli özelliğini doğrusal sıralı modelin kontrollü ve sistematik yönleriyle birleştiren evrimsel bir yazılım süreç modelidir. Proje aşamaları sarmallarla ifade edilir Ürünü geliştirmek için gereken aşamaların tam sayısı, proje risklerine bağlı olarak proje yöneticisi tarafından değiştirilebilir.



Resim 2.1 Spiral Model Süreci

2.2 Spiral Model Aşamaları

Bir Spiral sürecindeki ortak aşamalar şunlardır;

1. Hedef belirleme: Spiraldeki her döngü, o döngünün amacının, hedeflere ulaşmak için mümkün olan çeşitli alternatiflerin ve var olan kısıtlamaların tanımlanmasıyla başlar.
2. Risk Değerlendirmesi ve Azaltma: Döngüdeki bir sonraki aşama, bu çeşitli alternatifleri amaçlara ve kısıtlamalara dayalı olarak hesaplamaktır. Bu aşamadaki değerlendirmenin odak noktası, proje için risk algısıdır.
3. Geliştirme ve doğrulama: Bir sonraki aşama, belirsizlikleri ve riskleri çözen stratejiler geliştirmektir. Bu süreç, kıyaslama, simülasyon ve prototip oluşturma gibi faaliyetleri içerebilir.
4. Planlama: Son olarak, bir sonraki adım planlanır. Proje gözden geçirilir ve sarmalın daha ileri bir dönemine devam edilip edilmeyeceğine dair bir seçim yapılır. Tutmaya karar verilirse, projenin bir sonraki adımı için planlar yapılır.

2.3 Spiral Modelin Avantajları

* Geliştirme sürecinde ortaya çıkan birçok bilinmeyen risk içeren projeler, bu durumda, her aşamada risk analizi ve risk yönetimi nedeniyle Spiral Model izlenecek en iyi geliştirme modelidir.
* Spiral Model, yazılım geliştirmeye yinelemeli ve artımlı bir yaklaşım sunarak, değişen gereksinimlere veya beklenmedik olaylara yanıt olarak esneklik ve uyarlanabilirlik sağlar.
* Bu model kullanılarak, sonraki aşamalardaki gereksinimlerdeki değişiklik talepleri doğru bir şekilde dahil edilebilir.
* Büyük ve görev açısından kritik projeler için kullanışlıdır.

2.4 Spiral Modelin Dezavantajları

* Karmaşık: Spiral Model, diğer SDLC modellerinden çok daha karmaşıktır.
* Pahalı: Spiral Model pahalı olduğu için küçük projeler için uygun değildir.
* Risk Analizine çok fazla güvenilebilirlik: Projenin başarıyla tamamlanması büyük ölçüde Risk Analizine bağlıdır. Çok deneyimli uzmanlar olmadan, bu modeli kullanarak bir proje geliştirmek başarısız olacaktır.
* Zaman yönetimindeki zorluk: Proje başlangıcında faz sayısı bilinmediği için zaman tahmini çok zordur.

3. V-Model

3.1 V-Model Nedir?

V-Model, Verification ve Validation model olarak da anılan doğrusal ve sıralı bir modeldir. Bu modelde, yazılım yaşam döngüsünün her aşaması bir sonraki aşama başlamadan önce tamamlanmalıdır. Şelale modelinde olduğu gibi sıralı bir tasarım süreci izler. Karşılık gelen her geliştirme aşaması için bir test aşamasının ilişkilendirilmesine dayanır. Her geliştirme faaliyeti için buna karşılık gelen bir test faaliyeti vardır.



Resim 3.1 V-Model Süreci

3.2 V-Model Model Aşamaları

1. Kullanıcı modeli: geliştirme sürecinin kullanıcı ile olan ilişkisi tanımlanır. kullanıcı gereksinimlerinin analiz edildiği, olası çözüm yollarının araştırıldığı ve bir sistem tasarımı oluşturulduğu süreçtir.
2. Mimari model, yazılım projesinin tasarım aşamasının bir parçasıdır. Bu aşamada, yazılımın üst düzey tasarımı yapılarak, mimari seçimleri belirlenir. Mimari tasarım, yazılımın daha sonraki aşamalarında kullanılacak olan alt düzey tasarım belgelerinin temelini oluşturur.
3. Gerçekleştirim modeli: yazılım modüllerinin kodlanması ve test edilmesi ile ilişkili fonksiyonların gerçekleştirildiği kısımdır.

3.3 V-Model’in Avantajları

* V-Model'in net yapısı, müşteri ile geliştirme ekibi arasındaki iletişimi geliştirmeye yardımcı olur.
* Gereksinimler ile son ürün arasında net bir bağlantı sağlayarak, yazılımdaki değişiklikleri izlemeyi ve yönetmeyi kolaylaştırır.
* Tasarım ve test planı erken aşamada birlikte yapılabilir
* Doğrulama, potansiyel sorunları erken aşamalarda bulmaya yardımcı olabilir

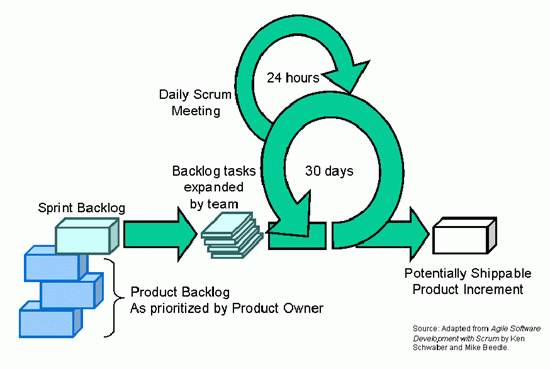
3.4 V-Model’in Dezavantajları

* Yüksek risk ve belirsizlik içerir.
* Çok katı ve az esnektir. Önceki adım doğrulandığında bir sonraki adım başlar. Bir üst aşamadaki herhangi bir değişiklik, alt aşamalarda değişikliklere yol açabilir ve tüm süreç etkilenebilir.
* Çok sayıda belgeleme ve test gerektirdiğinden zaman alıcı olabilir.
* Yazılım, uygulama aşamasında geliştirilir, bu nedenle yazılımın erken prototipleri üretilmez.

4. Scrum Modeli

4.1 Scrum Modeli Nedir?

Scrum metodolojisi, yazılım geliştirme sürecinde dahil edilmesi gereken bir dizi çok tanımlanmış uygulama ve role dayanmaktadır. 12 çevik ilkenin, ürünün tüm ekip üyeleri tarafından kabul edilen bir bağlamda uygulanmasını ödüllendiren esnek bir metodolojidir. Scrum, Sprint adı verilen, genellikle 2 ila 4 hafta arasında değişen, kısa ve periyodik geçici bloklarda yürütülür. on veya daha az üyeden oluşan ekipler için tasarlanmıştır Hesap verebilirliği vurgular ve iyi tanımlanmış bir hedefe doğru tekrarlanan ilerlemedir.



Resim 4.1 Scrum Modeli Süreci

4.2 Scrum Modeli’nin Aşamaları

1. Product Backlog: müşteri ile proje başlangıcında belirlenen yüksek seviyeli gereksinim listesidir.
2. Sprint Backlog: sprint döneminde yapılacak olan işlerin listesidir.
3. Sprint: genellikle 2 ila 4 hafta arasında değişen proje zaman dilimidir. Proje birden fazla sprint’ten oluşur.
4. Scrum Daily Meeting: Düzenli olarak her gün scrum takımı ile yapılan en fazla 30 dakika süren toplantılardır.

4.3 Scrum’daki Roller

* Scrum Master: Metodolojinin kurallarına ve süreçlerine uymaları için onlara rehberlik eden ekibe liderlik eden kişi. Scrum Master, Scrum'ı güncel tutmaktan, ihtiyaç duyması durumunda takımlara mentorluk sağlamaktan sorumludur.

"Scrum Master'ın işi, takımı sürekli gelişime doğru yönlendirmek - düzenli bir şekilde "Yaptığımız şeyi daha iyi nasıl yapabiliriz?" diye sormaktır. – Jeff Sutherland

* Ürün sahibi: Yazılımı kullanan paydaşların ve müşterilerin temsilcisidir., Ürün İş Listesine dahil edilecek hikayelerdeki faydaları doğrularlar ve bunları düzenli olarak önceliklendirirler.
* Scrum Takımı: Her sprintin başında taahhüt ettikleri hikayeleri birlikte yürüterek projeyi geliştiren gerekli teknik bilgiye sahip profesyonellerden oluşan bir grup.

4.4 Scrum Modeli’nin Avantajları

* Müşteri ihtiyaçları veya pazar gelişmeleri tarafından üretilen gereksinimlerdeki değişikliklere hızlı tepki verir.
* Scrum süreçleri yinelemelidir ve belirli çalışma dönemleri içinde ele alınır, bu da ekibin her dönem için belirli işlevlere odaklanmasını kolaylaştırır.
* Çalışma yöntemi ve her yinelemeden sonra işlevsel bir sürüm elde etme ihtiyacı, daha kaliteli bir yazılım elde edilmesine yardımcı olur.
* İlk etapta en değerli işlevleri yerine getirmek ve ekibin projede ilerleme hızını bilmek, riskleri önceden etkin bir şekilde ilerlemeye olanak tanır.

4.5 Scrum Modeli’nin Dezavantajları

* Scrum, belirli bir projede çalışan ekip üyelerinin yüksek düzeyde uzmanlığa sahip olmasını gerektirir. Bu nedenle, bazı projeler için uygun olmayabilir. Scrum, her sprint veya iterasyon için kısa süreli planlama yapılmasını önerir. Bu, bazı projeler için yetersiz planlama anlamına gelebilir.
* Scrum, her ekip üyesinin düşüncesine saygı duyar ve herkesin görüşünün alınmasını gerektirir. Bu, karar verme sürecini yavaşlatabilir.

Sonuç

Yazılım yaşam döngüsü modelleri karşılaştırıldığında, her birinin farklı avantajları ve dezavantajları bulunmaktadır. Bununla birlikte, günümüzde en popüler model olarak SCRUM modeli öne çıkmaktadır. SCRUM modeli, esnek yapısı ve sürekli geri bildirim alınması sayesinde karmaşık ve değişken projelerde tercih edilmektedir.



Şekil 1.1 Modellerin Karşılaştırılması

Projenin, boyutuna ve karmaşıklığına bağlı olarak diğer modellerin de tercih edilebileceği unutulmamalıdır. Örneğin, küçük ve basit projeler için Waterfall modeli uygun olabilirken, büyük ve karmaşık projelerde V Modeli veya Spiral modeli tercih edilebilir. Ayrıca, projenin özellikleri de seçilecek modeli etkileyebilir. Örneğin, bir projede risk analizi önemliyse Spiral modeli tercih edilebilir.

Tüm modellerin avantajları ve dezavantajları dikkate alınarak, hangi modelin seçileceği proje yöneticilerinin kararına bağlıdır. Bu nedenle, projenin ihtiyaçlarına uygun olan model seçilmelidir. Özetle, her modelin kendine özgü özellikleri bulunmaktadır ve projenin boyutu, karmaşıklığı ve özellikleri dikkate alınarak doğru model seçilmelidir

Kaynakça

<https://www.sciencedirect.com/topics/computer-science/waterfall-methodology>

<https://business.adobe.com/blog/basics/waterfall#:~:text=The%20Waterfall%20methodology%20%E2%80%94%20also%20known,before%20the%20next%20phase%20begins>.

<https://en.wikipedia.org/wiki/Waterfall_model>

<https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>

<https://www.geeksforgeeks.org/software-engineering-spiral-model/>

<https://en.wikipedia.org/wiki/Spiral_model>

<https://www.javatpoint.com/software-engineering-spiral-model>

sciencedirect.com/topics/computer-science/spiral-model

<https://artoftesting.com/spiral-model>

<https://www.codeinsideout.com/blog/book/v-model/#application>

<https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/>

<https://www.javatpoint.com/software-engineering-v-model>

<https://en.wikipedia.org/wiki/Scrum_(software_development)>

<https://www.projectmanager.com/blog/scrum-methodology>

<https://www.cprime.com/resources/what-is-agile-what-is-scrum/>

https://existek.com/blog/sdlc-models/