

YAZILIM GELİŞTİRME YAŞAM DÖNGÜSÜ MODELLERİ

Yusuf Karahan-220601048

İzmir Bakırçay Üniversitesi

Bilgisayar Mühendisliği 1.Sınıf

Özet

Bu yazının amacı yazılım geliştirme yaşam döngüsü nedir, bu döngünün basamakları nelerdir, yazılım yaşam döngülerinden olan çağlayan modelini, spiral modelini, kodla ve düzelt modelini açıklayıp çevik geliştirme metotlarından biri olan Scrum hakkında bilgiler vermektir.

YAZILIM GELİŞTİRME YAŞAM DÖNGÜSÜ NEDİR?

Geliştirdiğimiz bir yazılım projesinin planlamasından başlayarak teslimatına kadar geçirmiş olduğu bütün aşamalara ve bu aşamalardan oluşan döngüye yazılım geliştirme yaşam döngüsü denir aynı zamanda SDLC(Software Development Lifecycle) olarak da bilinir.SDLC mümkün olan en kısa sürede, en yüksek kalitede ve en düşük maliyetle yazılım üreten süreçtir.Yazılım geliştirme yaşam döngüsü yazılım işlevleri ve ihtiyaçları sürekli değiştiği ve geliştiği için mutlaka bir döngü biçiminde düşünülmelidir.Tek yönlü yani doğrusal olarak kesinlikle düşünülmemelidir.

YAZILIM GELİŞTİRME YAŞAM DÖNGÜSÜ AŞAMALARI:

1-Planlama: ”Ne istiyoruz?” sorusuna cevap arar.Yazılım geliştirme yaşam döngüsünün başlangıç aşamasıdır.Yazılım geliştirme yaşam döngüsünün bu aşamasında ekip, analiz edilen gereksinimlerin uygulanması için gereken maliyeti ve kaynakları belirler.Ayrıca ilgili riskleri detaylandırır ve bu riskleri azaltmak için planlar sunar.Başka bir deyişle, ekip projenin fizibilitesini ve projeyi en düşük riski göz önünde bulundurarak nasıl başarılı bir şekilde uygulayabileceklerini belirlemelidir.

2-Analiz: Bu aşamanın amacı sistemin işlevlerini ve kesin gereksinimleri açıklığa kavuşturmak ve sonucunda bunları belirli bir formatta dokümanete etmektir.Bu çalışma müşteri, yazılım mühendisi, sistem analisti, iş analisti, ürün yöneticisi vb. rollerin bir araya geldiği gruplar tarafından yapılabilir.İhtiyaçların net olmadığı durumlarda yazılım mühendisi ve müşteri arasında iletişim ve birlikte çalışmanın çok daha fazla olması gerekir.Çeşitli yazılım geliştirme metodolojilerinde bu aşamada kullanım dokümanları ve test plan dokümanları da oluşturulabilir.Kısaca analiz sürecinde projenin tüm detayları ortaya çıkartılır.

3-Tasarım: Gereksinimlerin tamamlanmasıyla beraber sistem tasarım aşamasına başlanır.Yazılım ürün tasarımı, müşterinin gereksinim ve isteklerini karşılamak üzere yazılım ürününün özelliklerinin, yeteneklerinin ve arayüzlerinin belirlenmesi etkinliğidir.İki tür tasarımdan bahsetmek mümkündür(Yüksek düzeyde tasarım-Mimarî tasarım ve Detaylı tasarım).Mimarî tasarım, yazılım modüllerinin genel yapıları ve organizasyon içerisindeki etkileşimleri ile ilgilenir.Detaylı tasarım aşamasında Mimarî tasarım dokümanları

oluşturulur. Detaylı tasarım aşamasında Mimarı tasarım dokümanları genelde revize edilirler. Tasarım dokümanında proje bilgileri, sistem tasarım bilgileri, tasarım detayları, veri modeli, kullanıcı arayüz tasarımları bulunur. Tasarım dokümanının amacı yazılım geliştiricinin yazılımı geliştirirken referans alacağı ve proje sürecinde/sonrasında projeye dahil olacak yeni yazılımcıların projeyi daha kolay anlayabilmesini sağlayacak teknik bir dokümantasyona sahip olması gerekliliğidir.

4-Gerçekleştirme: Planlama, analiz ve tasarımı tamamlanıp yapılacak işlemleri detaylı olarak belirlenen projenin geliştirme aşamasıdır. Bu noktadan sonra herhangi bir analiz işlemi yapılmamalıdır. Bu aşamada yazılımcı beta testlerini de gerçekleştirmektedir. Tasarım aşamasında belirlenen detaylar, günler, başlangıç ve bitiş tarihleri kapsamında projenin başlangıç tarihinde başlaması, bitiş tarihinde tamamlanacak şekilde kodlanması ve beta testleri gerçekleştirilmesi gerekmektedir. Bu aşama tamamlandıktan ve varsa hatalar giderildikten sonra proje yayına alınmaktadır.

5-Bakım: Yazılımın artık canlı olması ve müşteriler tarafından kullanılmasıyla birlikte, geliştirme ekibinin odak noktası onu korumaya kayacaktır. Geliştiricilerin iyileştirmeler, hata düzeltmeleri ve yeni özellikler için istekleri karşılamaya hazır olmaları gerekecektir. Bu süreç zarfında müşterilerden gelen bilgiler doğrultusunda bu istekler gerçekleştirilmektedir.

BAZI YAZILIM GELİŞTİRME YAŞAM DÖNGÜSÜ MODELLERİ İNCELENMESİ

1-ÇAĞLAYAN(WATERFALL) MODELİ:

Yazılım geliştirme yaşam döngüsündeki her aşama eksiksiz olarak yerine getirilir. Her aşamanın sonunda bir doküman oluşturulur. Yazılım süreci doğrusal bir şekildedir yani bir adımı atlayıp öbürüne geçemezsiniz ve basamak basamak ilerlemeniz gerekmektedir. Müşteri katılımı sadece başlangıçta mümkündür. Müşteri gereksinimleri bu safhada tespit edilir. Daha sonraki aşamalarda müşteri ile diyaloga girilmez. Günümüzde kullanımı gittikçe azalmaktadır.

ÇAĞLAYAN MODELİ AVANTAJLARI:

- 1-Fazlar net bir şekilde sınırlandırılmıştır.
- 2-Planlaması basittir.
- 3-Anlaşılması kolay bir modeldir.
- 4-Analiz ve tasarım aşamaları oldukça detaylıdır ve gereksinimler açık bir şekilde ortaya konulmuştur bu sayede kodlama ve test aşamaları kısa sürmektedir.
- 5-Analiz ve tasarım aşamaları iyi bir şekilde yapıldığı için test aşamasında çıkan hata sayısı azdır.
- 6-Küçük ve gereksinimleri iyi anlaşılmış projelerde kullanımı tercih edilir.

ÇAĞLAYAN MODELİ DEZAVANTAJLARI:

- 1-Müşteri katılımı sadece tanım aşamasındadır.
- 2-Her aşama sonrası doküman oluşturulması zaman kaybı oluşturabilir.
- 3-Statiktir yani değişimler için uygun bir model değildir.
- 4-Üst adımlarda oluşan hataların yarattığı zaman kaybı fazladır. Örneğin test aşamasında karşılaşılan bir hatanın sebebi tasarımdan kaynaklıysa tasarım ve test dokümanlarının güncellenmesi buna bağlı olarak da kodun revize edilmesi gerekir bu olaylar da bize zaman ve para kaybı olarak geri döner.
- 5-Bir başka dezavantaj ise ürünün ortaya çıkması için tüm aşamalardan geçmesi gerekmektedir bu da sabırsız müşteriler için soruna yol açabilir.

HANGİ PROJELERDE ÇAĞLAYAN MODELİNİ KULLANMALIYIZ?

Çağlayan modeli, daha çok askeri vb. sektörlerde, uzun süreli, analiz ve tasarımın çok önemli olduğu, hata yaptıktan sonra düzeltme maliyetinin yüksek olduğu kritik projelerde kullanılmalıdır.

2-SPİRAL(HELEZONİK) MODELİ:

Spiral model, sıralı ve prototip modellerin birleşimidir. Bu model en iyi, sürekli iyileştirmeler içeren projeler için kullanılır. Her bir aşama bir tasarım hedefi ile başlar ve müşterinin gelişimi gözlemlemesi ile sonuçlanır. Proje çevrimlere ayrılır ve her bir çevrimin riskleri ayrı ayrı ele alınır. Çağdaş modellere yakın bir modeldir. 4 aşaması vardır:

1-Planlama aşaması:İhtiyaçlar incelenir ve toplanır. Fizibilite çalışması yapılır. Gereksinimleri anlama belgesi ve gereksinim listesi çıkartılır.

2-Risk analizi:Olası riskleri belirlemek için analizler yapılır. Riskler belirlendikten sonra risk azaltması planlanır ve uygulanır. Çıktı olarak riskleri ve azaltma planlarını gösteren belge kullanılır.

3-Mühendislik:Risk analizi yapıldıktan sonra kodlama ve test yapılır. Çıktı olarak test raporu kullanılır.

4-Değerlendirme:Müşteriler yazılımı değerlendirip geri bildirimini yapar.

SPİRAL MODELİ AVANTAJLARI:

- 1-Geliştirme hızlıdır.
- 2-Risk değerlendirmesi uygundur.
- 3-Tüm aşamalarda kontrol edilebilir.
- 4-Yazılım erken üretilir.
- 5-Geliştirme hızlıdır ve özellikler sistematik bir şekilde eklenir.

SPİRAL MODELİ DEZAVANTAJLARI:

- 1-Küçük projeler için işlevli değildir.
- 2-Dokümantasyon, ara aşamalar olduğu için daha fazladır.
- 3-Düzenli bir şekilde çalışması için spiral model aşamalarına kesinlikle uyulmalıdır.
- 4-Risk analizi önemli bir basamak olduğu için bu konuda uzman kişiler gerekir.
- 5-Son ürüne ulaşabilmek için çok sayıda yenileme gerektirebileceğinden maliyet yüksek olabilir.

HANGİ PROJELERDE SPİRAL MODELİNİ KULLANMALIYIZ?

Proje büyük ise spiral model bu proje için uygundur. Risk ve maliyet değerlendirmesi önemli olduğunda ve gereksinimler belirsiz ve karışık olduğunda spiral modeli kullanılmalıdır.

3-KODLA VE DÜZELT(CODE&FIX) MODELİ:

En basit ürün geliştirme olarak adlandırılan bu modelde en kısa sürede prototipleyerek sonuca gitme hedeflenir. Genellikle resmi olmayan bir ürün fikriyle başlar ve süreç ürün hazır olana dek kodlama yapılarak devam edilir. Planlama, analiz yapma gibi aşamalar ile vakit kaybedilmez.

KODLA VE DÜZELT MODELİ AVANTAJLARI:

- 1-Herhangi bir planlamaya ihtiyaç duyulmaz.
- 2-Program aşamaları çabuk geçilir.
- 3-Uzman görüşüne ihtiyaç düşüktür.
- 4-Kullanım açısından kolaydır.

KODLA VE DÜZELT MODELİ DEZAVANTAJLARI:

- 1-Kontrollü değildir.

2-Bitiş süresi belirsizdir.

3-Hataların bulunması zordur.

4-Kodları düzeltmek maliyet açısından zor olabilir.

HANGİ PROJELERDE KODLA VE DÜZELT MODELİNİ KULLANMALIYIZ?

Genelde bir konuda araştırma yaparken, bir konuyu yeni öğrenirken veya bir problemi çözmeye çalışırken en basit yöntem kodla ve düzelt olduğu için kullanılmalıdır.

4-SCRUM MODELİ:

Çevik yazılım geliştirme metodolojileri uygulanabilecek bir proje yönetim yaklaşımıdır.Sadece yazılıma değil her projeye uygulanabilir.Ayrıca düzenli geri bildirim ve planlamalar ile hedefe ulaşmayı sağlar.Bu açıdan ihtiyaca yönelik ve esnek bir yapıya sahiptir.Karmaşık yazılım işlerini küçük birimlere(Sprint) bölerek geliştirmeyi öngörür.Bu süreçte iletişim ve takım çalışması oldukça önemlidir.Bu metodoloji, karmaşık ortamlarda adım adım yazılım geliştiren takımlar için uygundur.Bu metot 3 temel prensip üzerine kurulmuştur:

1-Şeffaflık: Projenin ilerlemesi, problemler ve gelişmeler herkes tarafından görünür olmalıdır.

2-Gözlemleme: Projenin ilerlemesi düzenli olarak kontrol edilmelidir.

3-Uyarılama: Proje, uygulanacak değişiklikler ile uyumlu olmalıdır.

SCRUM'IN ÇALIŞMA MANTIĞI:

Müşteri tarafından istenen işlevler, iki ya da dört haftalık “Sprint” adı verilen dönemler içerisinde geliştirilir ve yeniden gözden geçirilir.Her Sprint sonunda yazılımın fonksiyonel bir parçası bitmiş ve yazılım müşteriye teslim edilebilir bir durumda olur.Günlük kısa toplantılarla devamlı iş takibi yapılmaktadır.

SCRUM'IN YAPISI:

1-Product Backlog: Proje için gerekli olan gereksinimler listesidir.Proje sonunda “Ne üretilmek isteniyor?” sorusuna yanıt aranır.Product Owner tarafından müşteriden gereksinimler alınır, öncelik sırasına göre sıralanır.Product Owner, değişen ihtiyaçlara göre Product Backlog’a ekleme veya çıkarma yapabilir.Bu sayede değişim projenin her safhasında kolayca entegre edilebilir.

2-Product Backlog Item: Product Backlog içindeki her bir gereksinime verilen isimdir.

3-Sprint: Proje “Sprint” adı verilen küçük kısımlara ayrılır.Scrum içerisindeki tüm aktiviteler Sprint içerisinde gerçekleşir.

4-Sprint Backlog: Geliştirme takımı tarafından Product Backlog Item’lar öncelik sırasına göre Sprint içerisine alınırlar.İşlerin detaylı olacak şekilde zaman çizelgesi çıkarılır.

5-Scrum Board: Bir Sprint içerisinde yapılacak olan maddeler burada yönetilir.Yapılacaklar “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 Sprint’in günlerini, dikey ekseninde Sprintte kalan işi gösteren grafikdir.Scrum’ın temel ilkelerinden olan şeffaflığı sağlar.

SCRUM'DA ROLLER:

1-Product Owner: Geliştirme takımı ve müşteri arasındaki iletişimi sağlar.Projenin özelliklerini tanımlar.Sprint’i iptal yetkisine sahiptir.Projenin önceliklerine göre Product Backlog’u oluşturur.

2-Scrum Master: Scrum kurallarını, teoriklerini ve pratiklerini iyi bilir ve ekibinin bu kurallara uymasından sorumlu olan kişidir.Ekibin yöneticisi değildir.Ekibe rahatsızlık veren, verimli çalışmalarını engelleyen durumları ortadan kaldırır.

3-Geliştirme Takımı: Bir Sprint'e alınan, bütün işleri tamamlayacak özelliklere sahip kişilerdir.Sprint Backlog'u oluştururlar.Kendi kendini yönetir.İşin verilmesini beklemez, işi kendileri alır ve geliştirirler.Kişilerin tek bir görevi yoktur, çapraz görev dağılımı yaparlar.Takımdaki kişi sayısı 5-9 kişi arasında değişir.Projenin geliştirilmesi ile ilgili sorumluluk kendilerine aittir.

SCRUM'DA TOPLANTILAR:

1-Sprint Planning: Geniş kapsamlı gereksinim listesinin çıkarılması.Başarılı geliştirme için uygun dağıtım gereksinimlerinin belirlenmesi.Takımların belirlenmesi.Risk değerlendirmesi.

2-Sprint Review: Takım üyeleri, ürün gereksinim listesinin en başından başlayarak Sprint sonucunda gerçekleştirilecek bir Sprint gereksinim listesini oluştururlar.Bu Scrum'ın olmazlarından.Takım, gereksinimlerden yapılabilecekleri belirler.

3-Daily Scrum Meeting: 15 dakikalık bu kısa toplantı her iş gününde önceden ayarlanmış bir saatte gerçekleştirilir ve tüm takım katılır.Takım için önlerine çıkabilecek engelleri görmek için iyi bir fırsattır.

SCRUM NEDEN BU KADAR POPÜLER:

Scrum'ın günümüzde bu kadar popüler olmasının birkaç sebebi vardır bunları şu şekilde sıralayabiliriz:

- 1-Scrum yaklaşımında geliştirme ekibi tarafından yapılan işler ardışık değil,aynı anda gerçekleştirilir.
- 2-Projenin ömrü boyunca ve sonrasında bile her şey esnek ve değiştirilebilirdir.
- 3-Scrum, takımın kendi kendini yönetmesine olanak sağlayan bir yapıdır.
- 4-Takım üyelerinin işine daha bağlı olmasını sağlar ve üretkenliklerini en üst düzeye çıkarır.

KAYNAKÇA

- <https://caglartelefon.com/yazilim-vasam-dongusu/>
- <https://www.vkcgrup.com.tr/selale-modeli-nedir/>
- <https://medium.com/@secilcor/scrum-nedi%CC%87r-6a4326951dd8>
- <https://tr.myservername.com/spiral-model-what-is-sdlc-spiral-model>
- <https://medium.com/@esra /spiral-model-nedir-avantajlar%C4%B1-dezavantajlar%C4%B1-89c7ddffaae0>
- <https://medium.com/@brfn.kcr26/yazilim-geli%CC%87%C5%9Fti%CC%87rme-ve-s%C3%BCre%C3%A7-modelleri%CC%87-2131ea5f09b2>
- <https://talentgrid.io/tr/yazilim-gelistirme-modelleri/>
- <https://medium.com/@denizkilinc/yaz%C4%B1%C4%B1m-va%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-temel-a%C5%9Famalar%C4%B1-software-development-life-cycle-core-processes-197a4b503696>
- <https://www.linkedin.com/pulse/yaz%C4%B1%C4%B1m%C4%B1n-va%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-bevza-nur-karako%C3%A7/?trk=pulse-article&originalSubdomain=tr>

İLETİŞİM

<https://www.linkedin.com/in/yusuf-karahan-6a335725b/>

<https://github.com/YusufKarahen20>

<https://medium.com/@Yusuf.Karahan>