YAZILIM MÜHENDİSLİĞİ TEMELLERİ

2. DÖNEM 1. ÖDEV

HAZIRLAYAN:

AD-SOYAD: ESMA NUR AKDEMIR

OGRNO: 220601012

YAŞAM DÖNGÜ MODELLERİ

Yazılım yaşam döngüsü; herhangi bir yazılımın, üretim ve kullanım aşaması birlikte olmak üzere geçirdiği tüm aşamalar biçiminde tanımlanır. Yazılım işlevleri ile ilgili gereksinimler sürekli olarak değiştiği ve genişlediği için söz konusu aşamalar bir döngü biçiminde ele alınır. Bu döngü modellerinden bazıları aşağıda açıklanmıştır.

ÇAĞLAYAN / ŞELALE /WATERFALL YAŞAM DÖNGÜ MODELİ

Şelale yönteminde yazılım geliştirme süreci; analiz, tasarım, kodlama, test, sürüm ve bakım gibi safhalardan oluşur. Geleneksel yazılım metotlarında bu safhalar şelale modelinde olduğu gibi lineer olarak işler. Her safha, başlangıç noktasında bir önceki safhanın ürettiklerini bulur. Kendi bünyesindeki değişikler doğrultusunda teslim aldıklarını bir sonraki safhanın kullanabileceği şekilde değiştirir. Projenin safhaları ayrı olduğundan iş bölümü ve iş planı projenin en başında net bir şekilde bellidir. Bu durum proje yönetimini de oldukça kolay hale getirir. Kullanımı ve anlaması basit bir modeldir. Temel adımları baştan sona en az bir kez izlenerek gerçekleştirilir. Belgeleme işlevini ayrı bir aşama olarak ele almaz ve belgelemeyi üretimin doğal bir parçası olarak görür. İyi tanımlı ve üretimi az zaman gerektiren yazılım projeleri için uygun bir modeldir. Eski askeri yazılımlar ve projelerde bu model oldukça sık kullanılmıştır. Eski programların yazılma, işletilme maliyetleri fazladır. Yani çalıştığı zaman sorunsuz çalışacak sistemler üretmek çok kritiktir. Bunun için baştan analizlerin çok iyi yapılması gerekir. Uzay/Uydu aracında kullanılacak program, Askeri Cihaz/Füze/Teçhizatta kullanılacak program vb. için tekrar safhalı yazılım geliştirme mümkün değildir. Gerçek yaşamdaki projeler yineleme gerektirir. Ancak bu model oluşabilecek her türlü değişime elverişsiz, katı bir modeldir. Karmaşık ve nesne yönelimli projeler için uygun değildir. Devam eden ve uzun projeler için zayıftır. Yanlışların düzeltilme ve eksiklerin giderilme maliyetleri yüksektir. Safhalardan oluştuğu için ürün son safhada tamamlanır. Genelde yazılımın kullanıcıya ulaşma zamanı uzundur. Yazılım üretim ekipleri bir an önce program yazma, çalıştırma ve sonucu görme eğiliminde olduklarından, bu model ile yapılan üretimlerde ekip mutsuzlaşmakta ve kod yazma dışında kalan (ve iş yükünün %80'ini içeren) kesime önem vermemektedir. Üst düzey yönetimlerin ürünü görme süresinin uzun oluşu, projenin bitmeyeceği ve sürekli gider merkezi haline geldiği düşüncesini yaygınlaştırmaktadır. Gereksinimlerin iyi tanımlanmadığı, müşterinin ne istediğinin anlaşılmadığı bir projede bu durum projenin bittikten sonra iptal edilmesine ve başka gerginliklere sebep olmaktadır. Bu ve bunun gibi onlarca sebepten ötürü geleneksel model olarak da bilinen şelale modelinin kullanımı günümüzde giderek azalmaktadır.

V SÜREÇ MODELİ

V süreç modeli çağlayan(şelale) modelinin gelişmiş hali olarak düşünülebilecek bir yazılım geliştirme süreci sunar. Başka bir deyişle şelale modeline Verification(Doğrulama) ve Validation(Onaylama) mekanizmasının eklenmiş halidir. Doğrusal bir yönde ilerlemek yerine, süreç adımları kodlama evresinden sonra yukarıya doğru eğim alır ve tipik V şeklini oluşturur. Belirsizliklerin az, iş tanımlarının belirgin olduğu BT projeleri için uygun bir modeldir. Model, kullanıcının projeye katkısını arttırmaktadır. BT projesinin iki aşamalı olarak ihale edilmesi için oldukça uygundur: İlk ihalede kullanıcı modeli hedeflenerek iş analizi ve kabul sınamalarının tanımları yapılmakta ikinci ihalede ise ilkinde elde edilmiş olan kullanıcı modeli tasarlanıp gerçekleşmektedir.

Sol taraf üretim, sağ taraf sınama işlemleridir. V süreç modelinin temel çıktıları;

Kullanıcı Modeli

Geliştirme sürecinin kullanıcı ile olan ilişkileri tanımlanmakta ve sistemin nasıl kabul edileceğine ilişkin sınama belirtimleri, planları ortaya çıkarılmaktadır.

Mimari Model

Sistem tasarımı ve oluşacak alt sistem ile tüm sistemin sınama işlemlerine ilişkin işlevleri kapsar.

Gerçekleştirim Modeli

Yazılım modüllerinin kodlanması ve sınanmasına ilişkin fonksiyonları içerir.

V süreç modeli, sistem analizinin sonucunda yukarıdan aşağıya parçalandığı, tasarlandığı ve gerçekleştirme sırasında da sistem yapılarının toplandığı bir yapıdır. Bu toplama işleminin her aşamasında, ilgili doğrulama ve onaylama mekanizması V süreç modeli tarafından karşılanır.

SPİRAL (HELEZOİK) MODEL

Tasarımı doğrusal bir süreç olarak gören diğer modellerin aksine, bu model spiral bir süreç olarak görür. Bu, yineleyici tasarım döngülerini genişleyen bir spiral olarak temsil ederek yapılır. Genellikle iç spiraller, gereksinim tanımının rafine edilmesi için prototipleme ile birlikte ihtiyaç analizinin erken evresini ve dış spiraller yazılım tasarımını aşamalı olarak temsil eder. Her helezonda, tasarım çabalarını ve bu yineleme için ilgili riski değerlendirmek için bir risk değerlendirme aşaması vardır. Ve ayrıca her spiralin sonunda, mevcut spiralin gözden geçirilebilmesi ve bir sonraki aşamanın planlanabilmesi için gözden geçirme aşaması vardır. Her tasarım spiralinin dört ana faaliyeti dört temel görevle temsil edilmektedir: Planlama, risk analizi, üretim ve kullanıcı değerlendirme.

Planlama :

Üretilecek ara ürün için planlama, amaç belirleme, bir önceki adımda üretilen ara ürün ile bütünleştirme,

Risk Analizi :

Risk seçeneklerinin araştırılması ve risklerin belirlenmesi,

Üretim :

Ara ürünün üretilmesi,

Kullanıcı Değerlendirmesi :

Ara ürün ile ilgili olarak kullanıcı tarafından yapılan sınama ve değerlendirmeler şeklinde açıklanabilir.

Spiral modeli aynı safhalara geri dönülmesinin bir zorunluluk olduğunu vurgular. Spiral modeli şelale modelinde yok sayılan riskleri göz önünde bulundurur. Proje çevrimlere ayrılır ve her bir çevrimin riskleri ayrı ayrı ele alınır. Çağdaş modellere son derece yakındır. Pek çok yazılım modelini içinde bulundurur. Kullanıcılar sistemi erken görebilirler. Geliştirmeyi küçük parçalara böler. En riskli kısımlar önce gerçekleşir. Riske duyarlı yaklaşımı potansiyel zorlukları engeller. Seçeneklere erken dikkate ve hataları erken gidermeye odaklanır. Yazılım-donanım sistemi geliştirme için bir çerçeve sağlar. Büyük ölçekte projeler için uygundur. Küçük ve düşük riskli projeler için pahalı bir yöntemdir. Komplekstir (karmaşık). Spiral sonsuza gidebilir. Ara adımların fazlalılığı nedeniyle çok fazla dokümantasyon gerektirir. Kontrat tabanlı yazılıma uymaz. Kontrat tabanlı yazılımlar adım adım anlaşma esnekliğini sağlamaz. Yazılımın içten geliştirileceğini varsayar. Öznel risk değerlendirme

deneyimine dayanır. Yüksek riskli öğelere yoğunlaşmak, yüksek riskli öğelerin doğru belirlenmesini gerektirir. Bu da zaman alan bir işlemdir.

SCRUM

Çevik(Agile) Proje Yönetimi projeyi aşamalı olarak geliştirmeyi öngörerek bu aşamalar sırasında mümkün olduğunca esnek davranmaya imkan tanıyan ve kullanıcılarla iletişim halinde olunmasını öneren bir proje yönetim metodudur. Scrum, Agile proje yönetimi uygulamalarının en popüler olanlarından biridir.
"Scrum bir şeyin kısaltması mı?" diye düşünülebilir. Cevap hayır. İsim konusunda rugby sporundaki bir saldırıdan esinlenilmiştir. Rugby'de takım, topu ileriye taşımak için bir araya gelir. Bu bağlamda Scrum, ekibin ürünü ileriye taşımak için bir araya geldiği yerdir.
Scrum metodolojisi; bir yazılım çerçevesidir. İçerisinde tanımladığı belirli roller, aktiviteler, çıktılar vardır. Bunlar arasındaki kuralları tanımlar. Herhangi bir mühendislik pratiği önermez. İçerisini istediğiniz gibi doldurabilirsiniz.
Değerli iş artımları, Sprint adı verilen bir aylık veya daha kısa döngülerde teslim edilir. Sprint sırasında devam eden geri bildirim, sürecin ve neyin teslim edileceğinin incelenmesine ve uyarlanmasına izin verir. Scrum takımında bir Sprint sırasında iş seçimini bir değer parçasına dönüştürmekten sorumlu olan bir Scrum master , ürün sahibi ve geliştiriciler bulunur.
Scrum ekibi ve paydaşlar olarak bilinen kuruluşlarının, işlerinin, kullanıcılarının veya müşteri tabanlarının diğer üyeleri , Sprint'in sonuçlarını inceler ve bir sonraki için ayarlar.
Scrum, üç temel ilkeye dayanır: şeffaflık, denetleme ve adaptasyon. Proje boyunca tüm ekip üyelerinin çalışmaları açık bir şekilde görünür olmalıdır. Bu, proje ilerlemesinin izlenmesini ve takımın hedeflerine ulaşmak için nasıl çalıştığının anlaşılmasını kolaylaştırır. Denetleme; her Sprint sonunda takımın geri bildirimlerini değerlendirmesini, takımın performansını ve ürünün durumunu değerlendirmesini ve bunlara dayanarak gelecek Sprint'i planlamasını içerir. Bu süreç, takımın sürekli olarak kendini geliştirmesine yardımcı olur ve ürünün kalitesini artırır. Scrum, her döngü sonunda takımın geri bildirimlerini almasını ve bu geri bildirimlere dayanarak gelecek çalışmalarını planlamasını sağlar. Bu, takımın esnek olmasını ve değişen şartlara hızlı bir şekilde adapte olmasına olanak sağlar.
Tüm unsurları birbirine bağlayan kritik bir Scrum takımı özelliği "güven"dir . Bir Scrum takımında güven yoksa, işlerin yapılması yolunda büyük olasılıkla gerilim ve darboğazlar olacaktır. Scrum değerleri, çalışma şeklinize rehberlik etmeye ve güveni artırmaya yardımcı olduklarından, Scrum takımlarının bağlı kalması için de büyük öneme sahiptir. Scrum değerleri olan cesaret, odaklanma, bağlılık, saygı ve açıklık; Scrum takımı üyelerinin birlikte çalışırken göz önünde bulundurması gereken önemli unsurlardır. Scrum değerleri, özellikle denemenin ilerleme kaydetmenin temeli olduğu ortamlarda önemlidir.

SCRUM VS YAZILIM YAŞAM DÖNGÜ MODELLERİ

Scrum ve diğer yazılım yaşam döngüsü modelleri arasındaki en önemli fark, yaklaşım ve odak noktalarıdır. Scrum , Agile Manifesto'nun temel prensiplerine dayanan bir çerçeve olarak, iteratif(etkileşimli) ve inkremental(artımlı) bir yaklaşım benimser. Diğer yazılım yaşam döngüsü modelleri ise genellikle daha belirli ve önceden belirlenmiş bir süreç izlerler.

Çağlayan Modeli, geleneksel bir yazılım yaşam döngüsü modelidir ve genellikle lineer bir yaklaşım benimser. Bu model, proje için gerekli tüm gereksinimleri ve tasarımları belirler, ardından kodlama ve test sürecine geçer. Scrum'ın aksine, Çağlayan Modeli belirli bir süreç izler ve geri dönüşümlü değildir.

Spiral Modeli, Su Çarkı Modeli ile Çevik Model arasında bir yerde bulunur. Bu model, iteratif bir yaklaşım benimser ve yazılımın farklı aşamaları arasında geri dönüşler sağlar. Spiral Model, risk yönetimine ve kalite kontrolüne odaklanırken, Scrum daha çok ekip işbirliğine ve sürekli gelişime odaklanır.

V Süreç Modeli ve Scrum arasındaki en büyük fark, yaklaşımlarıdır. Scrum, müşteri odaklı ve iteratif bir yaklaşım benimserken, V süreç modeli, sürecin her adımının test edildiği ve onaylandığı sıkı bir şekilde planlanmış bir yaklaşım benimser. Scrum, değişen gereksinimlere hızlı bir şekilde yanıt vermek için tasarlanmıştır, ancak V süreç modeli, doğru ve öngörülebilir bir sonuç elde etmek için tasarlanmıştır.

Sonuç olarak, Scrum, özellikle müşteri odaklı, değişen gereksinimlere yanıt verme ve hızlı geliştirme ihtiyacı olan projeler için daha uygunken, V süreç modeli, doğru ve öngörülebilir sonuçlar için planlı bir yaklaşım gerektiren projeler için daha uygundur.

SCRUM GÜNÜMÜZDE NEDEN POPÜLER?

Scrum, ekiplerin daha verimli ve etkili çalışmasına yardımcı olmadaki etkinliği nedeniyle son yıllarda giderek daha popüler hale geldi. Scrum; işbirliğini, esnekliği ve sürekli iyileştirmeyi vurgulayan çevik yazılım geliştirmeye yönelik bir yapıdır. Günümüzün hızlı tempolu iş ortamında değişim süreklidir ve Scrum, ekiplerin gereksinimler, teknoloji ve diğer faktörlerdeki değişikliklere hızlı ve etkili bir şekilde yanıt vermesine yardımcı olur. Şeffaftır. Hesap verebilirliği, sorunları erkenden belirlemeye ve ele almaya yardımcı olabilecek sık iletişimi ve geri bildirimi teşvik eder. Scrum, ekiplerin yüksek kaliteli ürünleri hızlı bir şekilde teslim etmesini sağladığı için de popülerdir. Scrum süreci, ekiplerin erken ve sık sık geri bildirim almalarını ve gerekli ayarlamaları hızla yapmalarını sağlayan, çalışan yazılımın kısa iterasyonlarda düzenli olarak teslim edilmesini vurgular. Genel olarak Scrum, ekiplerin daha verimli çalışmasına, değişime uyum sağlamasına ve hızlı bir şekilde yüksek kaliteli ürünler sunmasına yardımcı olan çevik yazılım geliştirme için kanıtlanmış, etkili bir yapı sağladığı için popüler hale geldi.