YAZILIM MODELLERİ

1-Yazılım modeli nedir?

Yazılım geliştirme modellerini tıpkı bir yaşam döngüsüne benzetebiliriz.Fazlası ise bu yaşam döngüsünün farklı biçimlerde uygulanmasıdır. Şelale, V-model, artımlı model, iteratif model ve çevik modeller gibi birçok çeşit yazılım geliştirme modelleri vardır.Yazılım geliştirici olan bizler modellere uygun şekilde test faaliyetlerini şekillendirmeliyiz.Bu yüzden yazılım geliştirme modelleri hakkında bilgi sahibi olmalıyız.Yazılım geliştirme modeli,bir yazılım projesinin belirli bir süreç içinde naıl geliştirileceğini ve uygulanacağını tanımlayan bir yöntemdir.Yazılım geliştirme modelleri,yazılım mühendislerine,bir proje boyuncatakip edilmesi gereken belirli bir yol haritası sunar.Bu model,yazılımın işlevel,kaliteli ve müşteri gereksinimlerine uygun olmasını sağlar.Bir yazılım geliştirme modeli,belirli bir dizi adımın ve sürecin takip edildiği bir planlamayı ifade eder.Bu süreçler yazılımın her aşamasında doğru şekilde ilerlemesini sağlar.Genellikle bir yazılım geliştirme modeli,gereksinimlerin belirlenmesi,tasarım,kodlama,test ve bakım gibi aşamaları içerir.Ancak,bu aşamaların sırası ve detayları,kullanılan modelin türüne ve gereksinimlerine göre değişebilir.Yazılım geliştirme modelleri, farklı disiplinlerdeki yazılım geliştiricileri tarafından kullanılır. Örneğin, büyük yazılım projelerinde, işletmeler genellikle belirli bir yazılım geliştirme modeli kullanır ve bu modeldeki adımların uygun şekilde takip edildiğinden emin olmak için bir proje yöneticisi görevlendirirler.Sonuç olarak, yazılım geliştirme modelleri, yazılım geliştirme sürecinin planlanması ve yönetimi için bir rehber sunar. Bu modeller, yazılım geliştiricilerin, müşterilerin ve paydaşların ihtiyaçlarını karşılamak için uygun yöntemleri kullanarak yazılım projelerini yönetmelerine yardımcı olur.

2-Hangi projede hangisini kullanmalıyız?

Hangi yazılım geliştirme modelinin kullanılacağı, projenin özelliklerine ve ihtiyaçlarına bağlıdır. Her projenin kendine özgü gereksinimleri vardır ve bir model, her projeye uymayabilir. Aşağıda bazı yazılım geliştirme modelleri verilmiştir.

1-WaterFall Modeli

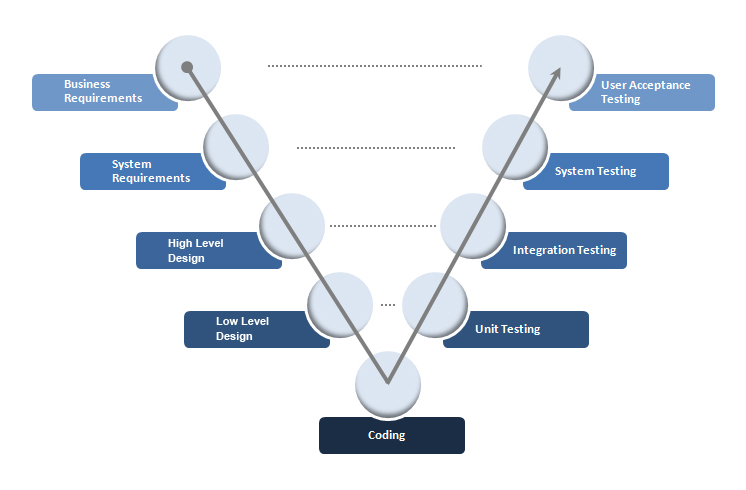
Analiz, Tasarım, Gerçekleştirim, Kodlama, Test, Entegrasyon … adımların sıra ile işletilerek bitirilmesi sürecidir. Geri dönülmemesi için her fazın hakkını vererek doğru bir şekilde yapılması gerekir. Eski askeri yazılımlar ve projelerde bu model oldukça sık kullanılmıştır.

Ama sonradan bu modeli işletmenin o kadar da kolay olmadığı,anazilin süreç içinde değişebileceği ,kodlama sırasında tasarımın değiştiği farkedilmiş ve bu modelin yerine daha uygulanabilir modeller aranmaya başlanmıştır. Tabi bundaki bir diğer etkende eskiden programların yazılma, işletilme maliyetleri fazlaydı. Yani çalıştığı zaman sorunsuz çalışacak sistemler üretmek çok kritikti. Bunun için baştan analizlerin çok çok iyi yapılması gerekiyorki Uzay/Uydu aracında kullanılacak program, Askeri Cihaz/Füze/Teçhizat kullanılacak program vb için böyle tekrar safhalı yazılım geliştirme mümkün değildi. İleri süreçte,teknolojiyle yazılımın hayatımızın her alanında kullanılmasıyla, değişen ihtiyaçlar ve koşullar sayesinde farklı farklı yazılım geliştirme modelleride ortaya çıkmıştır.

2-V modeli

Waterfall Modeline Verification(Doğrulama) ve Validation(Onaylama) mekanizmasının eklenmiş halidir.gerçekleştirme fazı içerisinde kodlama ve test aşamalarını içerir. Amaç her aşamanın karşısında ve doğrulama ve onaylama mekanizmalarını koyarak sistemin düzgün ve kalite çalıştığından emin olarak ilerlediğiniz geliştirme modelidir.

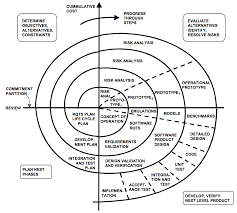
V modeli, yazılım geliştirme sürecinin bir çeşit su yolu modelidir. Bu modelde, her aşama, bir önceki aşamanın doğrulanması ve test edilmesiyle eşleştirilir. Bu nedenle, V modeli bazen doğrulama ve doğrulama modeli olarak da adlandırılır. V modelinde, her bir geliştirme aşaması karşılık gelen bir doğrulama veya test aşamasına sahiptir. Bu aşamalar, aşağıdaki şekilde gösterilen V şekline benzeyen bir diyagramda gösterilir:



Yukarıdaki diyagramda, gereksinimler aşamasından başlanır ve her aşama, aşağıdaki test aşamasıyla birlikte uygulanır. Örneğin, gereksinimler aşamasından sonra tasarım aşamasına geçilir ve tasarım doğrulanır ve test edilir. Daha sonra, uygulama veya kodlama aşamasına geçilir ve kodlama, tasarım doğrulaması ve test aşamasından geçirilir. V modeli, yazılım geliştirme projelerinde sıklıkla kullanılan bir modeldir ve çeşitli sektörlerde uygulanabilir. Ancak, modelin uygulanması için projenin gereksinimlerinin tam ve doğru bir şekilde belirlenmiş olması gerekir. Ayrıca, her aşamanın doğrulama ve test aşamalarının doğru bir şekilde planlanması ve uygulanması mühimdir.

3-Spiral model

Barry Boehm tarafından 1986 yılında geliştirilen spiral (sarmal) model; klasik çevrim ve prototip oluşturma yöntemlerinin en iyi yönlerinin bir araya getirilmesiyle oluşturulmuşturBu model yazılım gelişiminin tekrarlanarak artmasına dayanmaktadır. Şelale modeli ile karşılaştırıldığında en büyük farklarından biri bu metodolojinin yazılımın gelişimi ve kullanımı boyunca sürdürülebilir olmasıdır. Spiral model şelale modelinde olduğu gibi sistematik basamak yaklaşımı ile tekrarlanan yapıyı kullanmaktadır. Ancak bu modelde tekrar sayısı için herhangi bir sınırlama bulunmamaktadır. Spiral modelin diğer modellere kıyasla en önemli özelliği; her sarmalda ürünle ilişkisi olanların (kullanıcı veya geliştirici ekibinden) bakış açılarıyla çevrimin tamamlanmasını sağlamalarıdır. Bu bakış açısı ilk döngüden bir sonraki döngünün planlanmasına ve bunların gerçekleştirilmesi için gerekli kaynaklara kadar bütün ürün geliştirme sürecini kapsamaktadır. Spiral modelin sunulmasında dört temel soru kendini göstermektedir; **sarmal desen** uygulama geliştirme sürecinin bir modelidir. Yazılım geliştirmenin, belirlenen hedeflere ulaşılıncaya kadar tekrarlanan yinelemeli bir döngü olduğu hipotezine dayanmaktadır. Herhangi bir yazılım geliştirirken oluşabilecek çok sayıda riski yönetme yeteneğine sahiptir.Risk yönetimini destekleyen en önemli modellerden biridir. Adından da anlaşılacağı gibi bu model, modelin farklı aşamalarının farklı döngülerde dağıtıldığı spiral şeklinde gösterilmektedir. Modeldeki döngü sayısı sabit değildir ve projeden projeye değişebilir. Boehm, yayınında, spiral modelini daha önce oluşturulmuş şelale modeline olası bir alternatif olarak tanımladı ve bu da onun uygulamasının temelini oluşturdu.Spiral model döngüsel gelişimi tartışan ilk model değildi, ancak yinelemenin neden önemli olduğunu açıklayan ilk modeldi. Başlangıçta planlandığı gibi, yinelemeleri tipik olarak 6 ay ile 2 yıl arasında değişen büyük, karmaşık projeleri hedef almıştır.Bu model, şelale modelinden farklı olarak yazılım geliştirme görevlerinin doğrusal olarak tasarlandığını varsaymaz, aksine bunları yinelemeli görevler olarak görür.Bu döngüsel model, Model Tabanlı Yazılım Mühendisliği Mimarisini (MBASE) ve aşırı programlamayı etkiledi.Önerilen tüm alternatifler değerlendirilir. Hedefler ve kısıtlamalar, en iyi çözümü seçmek için referansları belirleme görevi görür.Ayrıca deneyim eksikliği, yeni teknolojiler, sıkışık programlar, eksik süreçler gibi projenin başarısını engelleyebilecek riskler tespit edilerek en düşük riskle en karlı stratejiler uygulanmaktadır.Son olarak, prototipleme, simülasyonlar, analitik modeller ve kullanıcı anketleri gibi yöntemler kullanılır.Aşşağıda şeması gösterilmiştir.



4-Scrum model

Scrum modeli, yazılım geliştirme sürecinde kullanılan bir çerçevelemedir. Bu model, özellikle esnek ve hızlı bir şekilde değişen projelerde kullanılan bir çevrimiçi yazılım geliştirme sürecidir. Scrum, çevik yazılım geliştirme yöntemleri arasında en popüler olanıdır.Scrum modeli, geliştirme sürecindeki işlerin bölünmesi ve yönetilmesi için bir takım tabanlı bir yaklaşım kullanır. Scrum ekibi, geliştirme sürecinin her aşamasında birlikte çalışarak hedeflere ulaşmak için birlikte çalışırlar. Bu ekip, genellikle 5-9 kişiden oluşur ve bir Scrum Master liderliğinde çalışır.Scrum modelinde, işler küçük parçalara bölünür ve her parça kısa bir süre içinde tamamlanması planlanır. Bu parçalar, Sprint olarak adlandırılan bir dizi zaman dilimi boyunca tamamlanır. Her Sprint, genellikle 2-4 hafta sürer ve sonunda ürünün bir sonraki aşamasına geçmek için bir inceleme yapılır. Scrum, Agile yazılım geliştirme metodolojilerinden biridir. Bu model, takım çalışması ve sürekli geri bildirimler ile esnek, hızlı ve verimli bir şekilde yazılım geliştirmeye odaklanır.

Scrum modeli, üç temel rol, üç ana süreç ve üç ana çıktıdan oluşur.

Üç Temel Rol:

1. Ürün Sahibi (Product Owner): Ürün Sahibi, müşterilerin ihtiyaçlarını anlayan ve bu ihtiyaçları takımın anlaması ve bu ihtiyaçları karşılaması için gerekli olan ürün özelliklerini belirleyen kişidir.
2. Scrum Takımı (Scrum Team): Scrum Takımı, geliştirme sürecinde birlikte çalışan yazılım geliştiricileri, test uzmanları, tasarımcılar gibi kişilerden oluşan bir ekiptir.
3. Scrum Master: Scrum Master, takımın gelişimini ve performansını takip eden, takımın sorunlarını çözmeye yardımcı olan ve Scrum sürecinin doğru bir şekilde uygulanmasını sağlayan kişidir.

Üç Ana Süreç:

1. Planlama Toplantıları (Sprint Planning Meetings): Bu toplantıda, Scrum Takımı, Ürün Sahibi tarafından sağlanan gereksinimleri inceler ve her Sprint döngüsü için bir plan yapar.
2. Günlük Toplantılar (Daily Scrum Meetings): Günlük toplantılarda, Scrum Takımı, birbirleriyle çalışmaları hakkında bilgi paylaşır ve olası sorunları tartışır.
3. İnceleme Toplantıları (Sprint Review Meetings): Bu toplantıda, Scrum Takımı, son Sprint döngüsü boyunca yapılan işleri ve ürün özelliklerini Ürün Sahibi ile birlikte inceler.

Üç Ana Çıktı:

1. Artımlı Ürün Geliştirme: Scrum Takımı, her Sprint döngüsü sonunda, bir artımlı ürün geliştirir.
2. Sürekli Geri Bildirim: Scrum Takımı, sürekli olarak geri bildirim alır ve bu geri bildirimleri ürünün geliştirilmesine yönelik kullanır.
3. Esneklik: Scrum, esnek bir yapıya sahiptir ve ihtiyaçlar değiştikçe adapte olabilir.

Scrum, yazılım geliştirme sürecindeki işbirliği ve hızlı prototip oluşturma ile birlikte esneklik ve sürekli geri bildirim sağlar. Bu model, özellikle karmaşık ve değişken projelerde kullanılmak üzere tasarlanmıştır.

İsim Soyisim:Ahmet Akdemir

Öğrenci No:220601011