

Yazılım Süreç Modelleri

Süreç modelleri, yazılım yaşam döngüsünde belirtilen süreçlerin geliştirme aşamasında, hangi düzen ya da sırada, nasıl uygulanılacağını tanımlar. Yazılım üretim yaşam döngüsü sekiz sınıfta incelenebilir:

1. Gelişigüzel Model
2. Barok Model
3. Çağlayan Model
4. V Modeli
5. Spiral (Helezonik) Model
6. Prototipleme Modeli
7. Evrimsel Geliştirme Modeli
8. Artımlı Geliştirme Modeli
9. Scrum Modeli

Gelişigüzel Model-1960lar:

Yazılım geliştirme ortamında herhangi bir model veya yöntemin kullanılmadığı yalnızca geliştiren kişiye bağımlı hatta geliştiren kişinin bile aradan belirli bir zaman geçtikten sonra anlayamayacağı ve değiştirme gücü olduğu yaşanan ortamlardaki üretim tarzı gelişigüzel model olarak adlandırılmaktadır. Bu modelle üretilen ürünlerin izlenebilirliği, bakılabilirliği oldukça zor bazı durumlarda olanaksızdır.

2.Barok Model-1970ler:

Yazılım mühendisliği 14 barok modeli planlama, çözümleme, tasarım ve gerçekleştirim işlevleri içermektedir. Bu modelin ayırıcı özelliği belgeleme olgusuna ayrı bir önem atfetmesi ve belgelemeyi bir süreç olarak ele almasıdır. Belgelemenin yazılımın geliştirilmesi ve sınanmasından hemen sonra yapılması öngörülmektedir. Barok modelin zayıf tarafı ise aşamalar arasındaki geri dönüşlerin nasıl yapılacağını tanımlı olmamasıdır.

3. Çağlayan Model:

Çağlayan modelinde yazılım aşamalarının en az birer kez tekrarlanması ile geliştirilir. Üretimi az zaman gerektiren projeler için uygun bir model olmakla birlikte günümüzde kullanımı gittikçe azalmaktadır.

Çağlayan modelinin avantajları:

- Müşteriler için anlaşılır adımlardan oluşur.
- Tekrarlamalar sonraki ve önceki adımlarla gerçekleşir.
- Gereksinim adımı tamamlandıktan sonra sağlam bir temel oluşur.
- Gereksinimler ne kadar iyi anlaşılıyorsa projelerde o kadar iyi çalışır.
- Yöneticiler için proje dağıtımını yapmak kolaydır.

Çağlayan modelinin dezavantajları:

- Gerçek yaşamdaki projelerin çok azı yineleme gerektirmez.
- Yazılımın kullanıcıya ulaşma zamanı oldukça uzundur.
- Gereksinim tanımları çoğu kez net olarak yapılamaz bu konudaki eksiklik ya da yanlışın ortaya çıkma zamanı gerçekleştirim sonrasına rastlar. Bu durumda yanlışların düzeltilme ya da eksikliklerin giderilme maliyetleri yükselir.
- Üst düzey yönetimlerin hedeflenen ürünü görme süresinin uzun oluşu projelerde sorunlara yol açmaktadır.

4. V Modeli:

V modeli, çağlayan modelinin uygulanmasını üretim ve sınaıa işlevlerinin ne zaman yapılacağını vurgulayarak daha anlamlı hale getirmektedir. V modelinin sol tarafı üretim sağ tarafı ise sınaıa işlevleri ile ilgilidir. Model aynı zamanda sınaıa işlemlerinde hata bulma durumunda nereye dönüleceğini de belirtmektedir. Sağ tarafta yapılan sınaıa işlemlerinde bulunan bir hata durumunda yatay olarak karşısına gelen sol taraf işlevlerine dönülmektedir. Örneğin sistem sınaıa işlemlerinde bulunan yanlışların düzeltilmesi amacıyla mimari modelin sol tarafındaki sistem tasarımına dönülmekte alt sistem tasarımı, modül geliştirme, modül sınaıa, alt sistem sınaıa ve sistem sınaıa işlemleri yinelenmektedir.

V modeli çıktıları:

1.Kullanıcı modeli:

Geliştirme sürecinin kullanıcı ile olan ilişkisini tanımlamaktadır.Söz konusu ilişkiler gerek üretim gerekse kabul açısından kullanıcıdan beklenenleri ortaya koymaktadır.Üretim amacıyla kullanıcı gereksinimlerini tanımlamakta ve sistemin nasıl sınaıanacağı ya da kabul edileceğine ilişkin sınaıa belirtileri ve planları ortaya çıkarılmaktadır.

2.Mimari modeli:

Sistem tasarımı ve oluşacak alt sistem ve tüm sistemin sınaıa işlemlerine ilişkin işlevleri içermektedir.

3.Gerçekleştirim modeli:

Yazılım modüllerinin kodlanması ve sınaıanmasına ilişkin işlevleri içermektedir.

5.Spiral (Helezonik) Model:

- *Risk analizi olgusu ön plana çıkmıştır.
- *Her döngü bir fazı ifade eder.
- *Yinelemeli artımsal bir yaklaşım vardır.
- *Prototip yaklaşımı vardır.

Avantajları:

- Kullanıcılar sistemi erken görebilirler.
- Geliştirmeyi parçalara böler.En riskli kısımlar önce gerçekleştirilir.
- Pek çok yazılım modelini içinde bulundurur.
- Riske duyarlı yaklaşımı potansiel zorlukları engeller.
- Hataları erken gidermeye odaklanır.
- Yazılımın kodlanmasını ve sınanmasını daha erken başlar.

Dezavantajları:

- Karmaşıktır,spiral sonsuza kadar gidebilir.
- Ara adımlar çok olduğu için çok fazla belgeleme gerekir.
- Kontrat tabanlı yazılıma uymaz.
- Öznel risk değerlendirme deneyimine dayanır.

6.Prototipleme Modeli:

Gereksinimler hızlıca toplanarak işe başlanılır. Geliştiriciler ve kullanıcılar aynı masa etrafında buluşarak yazılımdan elde edilecek bütün çıktılarına, bu çıktılar için gerekli girdilerin nasıl sağlanacağına, nasıl korunacağına, hangi işlemlere uğrayacağına karar verirler. Daha sonra hızlıca yapılan bir tasarım ile yazılımın kullanıcıya yansıyacak yönünü aktaran bir ilk örnek üretilir. Prototip kullanıcının kullanımına ve değerlendirilmesine sunulur. Bu değerlendirmelere bakılarak ilk örnek üzerinde gerekli değişiklikler yapılır. Prototipin yeni hali kullanıcı tarafından yeniden değerlendirilir. Böylece kullanıcının istediği yazılıma iyice yaklaşmış bir ilk örnek üzerinde yazılımın neler yapacağı konusunda kullanıcı ile anlaşmaya varılır. Doğrusal modelin döngüsel versiyonudur. Bu modelde, gereksinim analizi ve prototipleme için tasarım yapıldıktan sonra, geliştirme süreci başlatılır. Prototipleme yaratıldıktan sonra, müşteriye değerlendirme için verilir. Müşteri paketi test eder ve düşüncelerini, ürünü müşterinin tam beklentilerine göre düzenleyen geliştiriciye iletir. Sınırlı sayıdaki yinelemelerden sonra, son yazılım paketi müşteriye verilir. Bu metodolojide, yazılım müşteri ve geliştirici arasında periyodik bilgi gidip gelmeleri sonucunda gelişir.

Avantajları:

- Kullanıcı sistem gereksinimlerini görebilir.
- Karmaşa ve yanlış anlaşılımları engeller.
- Yeni ve beklenmeyen gereksinimler netleştirilebilir.
- Risk kontrolü sağlanır.

Dezavantajları:

- Belgelendirmesi olmayan hızlı ve kirli prototipler.
- Prototip hedefleri net değilse kod hackleme ya da jenga başlar
- Düzeltme aşaması atlanırsa düşük performansa yol açar.

7.Evrimsel Geliştirme Modeli:

Evrimsel geliştirme süreç modeli daha çok coğrafik olarak geniş alana yayılmış çok birimli organizasyonlar için önerilmektedir.

Avantajları:

- Kullanıcıların kendi gereksinimlerini daha iyi anlamalarını sağlar.
- Sürekli değerlendirme erken aşamalardaki geliştirme risklerini azaltır.

Dezavantajları:

- Sistemler sıklıkla iyi yapılandırılmaz.
- Bakımı zordur.
- Yazılım gereksinimini yenilemek gerekebilir.

8.Artımlı Geliştirme Modeli:

Modelde üretilen ve uygulamaya alınan her ürün sürümü birbirini içerecek şekilde giderek artan sayıda işlev içerecek biçimde geliştirilmektedir. Öncelikle ürüne ilişkin çekirdek bir kısım geliştirilerek uygulamaya alınmakta ardından yeni işlevsellikler eklenerek yeni sürümler elde edilmektedir.

Avantajları:

- Sistem için gerekli olan gereksinimler müşterilerle belirlenir.
- Gereksinimlerin önemine göre teslim edilecek artımlar belirlenir.
- Öncelikle en önemli gereksinimleri karşılayan çekirdek bir sistem geliştirilir.
- Erken artımlar prototip gibi davranarak, gereksinimlerin daha iyi anlaşılmasını sağlar.
- Tüm projenin başarısız olma riskini azaltır.
- Böl ve yönet yaklaşımıdır.

Dezavantajları:

- Artımları tanımlamak için tüm sistemin tanımlanmasına ihtiyaç vardır.
- Gereksinimleri doğru boyuttaki artımlara atamak bazen zor olabilir.
- Artımların kendi içlerinde tekrarlamalara izin vermez.
- Deneyimli personel gerektirir.

9.Scrum modeli:

Scrum; Agile proje yönetim metotlarından biridir. Scrum Metodu yazılım süreçlerinin yönetimi için kullanılmaktadır. Ayrıca düzenli geri bildirim ve planlamalar ile hedefe ulaşmayı sağlar. Bu açıdan ihtiyaca yönelik ve esnek bir yapıya sahiptir. Müşteri ihtiyacına yönelik olduğu için geri bildirimlere göre yapılanmayı sağlar. Scrum Metodu sürecinde iletişim ve takım çalışması oldukça önemlidir. Bu metot 3 temel prensip üzerine kurulmuştur. Bunlar aşağıdaki gibidir:

Şeffaflık: Projenin ilerlemesi ve problemler, gelişmeler herkes tarafından görülebilir olmalıdır.

Gözlemler: Projenin ilerlemesi düzenli olarak kontrol edilmelidir.

Uyarılma: Proje, uygulanacak değişiklikler ile uyumlu olmalıdır.

SCRUM günümüzde en çok kullanılan yazılım geliştirme yöntemidir. Hatta sadece yazılım geliştirmede değil birçok sistemin geliştirilmesinde de kullanılır. Bunun nedenleri ise:

- Zamandan ve paradan büyük ölçekte tasarruf edilmesi
- Yüksek teknolojiler ve son gelişmelere kolaylıkla uyum sağlayabilmesi
- Karmaşık görülen ve gereksinimleri tam belirlenmemiş projeler için ideal olması
- Ekip içi iletişimin yüksek tutulması ve bununla beraber hataların erken fark edilip düzeltilmesi
- Kullanıcıdan sürekli geri bildirim gerektirmesi ve bununla beraber sorunların azalması

Hangi model günümüzde daha kullanışlıdır?

Gelişigüzel ve Barok modelleri günümüzde çok kullanışlı değildir.Çünkü bu modeller yinelemeli değildir ve ayrıca bu modellerde belgeleme yoktur.

Çağlayan modelindeki dezavantajlar avantajlarını gölgede bırakmaktadır.Evet,müşteriler için anlaşılması kolay model olmasının yanında sağlam bir temel oluşturulabilir ancak günümüzdeki projelerin çok azında yineleme gerekmez.Ve ayrıca müşteriye ulaşma süresi çok uzun oluyor.

V modelinin kullanımı ve takibi çok kolay olmasının yanı sıra tekrarlamalara imkan tanınmaması ve risk çözmek gibi bir çözüm üretimi olmamasını gibi sorunları vardır.

Spiral modelde risk analizi çok önemlidir,hatalar erkenden çözüme kavuşturulur.Çok karmaşık bir model olduğundan dolayı büyük projelerde tercih edilir.

Prototipleme modelinde spiral modelde olduğu gibi risk analizi önemlidir bu da karmaşa ve yanlış anlaşılmayı engeller.Müşterinin sistem gereksinimlerini görmesi de güzel bir özelliktir.Tek sıkıntısı hedefler net değilse ve düzeltme aşaması atlanırsa hacklenme ve performans sorunlarına sebebiyet verebilir.

Evrimsel geliştirme modeli biraz daha global oluyor yani farklı coğrafyadan farklı organizasyonların olması gerekiyor.Bu sebepten bakımı çok zordur.

Artımlı geliştirme modelinde sürüm sürüm ilerleme kat edildiği için hata yapılması çok zordur.Gereksinimlerin müşteri ile belirlenmesi de işin içine girdiği zaman gerçekten başarılı projeler çıkabiliyor bu modelle birlikte.Ancak Scrum geliştirme modeli günümüzde daha ön plandadır.Çünkü scrum modeli geliştirme sürecini takım odaklı yapmakla beraber aynı zamanda müşteriyle daha çok iç içelik söz konusu.Artımlı geliştirme modeli bir parçada dönüt alırken scrum modeli bir döngüde dönüt almaya bakıyor.Ve scrum modelinin teknolojiyle daha kolay uyum sağlaması da onu son dönemlerin vazgeçilmez modeli yapıyor.

Mehmet Demir

220601032