Çağlayan (Waterfall) Modeli:

Yazılım geliştirme süreci, Çağlayan modeline göre analiz, tasarım, kodlama, test, sürüm ve bakım aşamalarından oluşur. Bu aşamalar şelale modelinde olduğu gibi doğrusal olarak ilerler. Ancak, her aşama, bir önceki aşamanın ürettiği dokümanlarla başlar.

Analiz adımıyla başlayan süreçte, yazılım gereksinimleri net bir şekilde belirlenir ve analiz dokümanı üretilir. Sonrasında tasarım adımında, yazılımın arayüz, veritabanı ve sınıf gibi tasarımları yapılır ve tasarım dokümanı oluşturulur.

Kodlama aşamasında, analiz ve tasarım dokümanlarında belirtilen şekilde yazılım kodlanır. Daha sonra test adımında, tüm gereksinimler ve tasarımlar için test senaryoları oluşturulur ve bu testler yapılır. Hata bulunması durumunda, hatanın kaynağına göre ilgili adıma geri dönülerek hata giderilir.

Eğer yazılımda herhangi bir hata bulunmazsa, entegrasyon adımına geçilir ve yazılım canlı ortama entegre edilir. Bu sayede müşterilerin kullanımına açılır. En son olarak da bakım aşaması başlar ve yazılımın düzenli olarak güncellenmesi ve bakımı yapılır.

Şelale modeli, yazılım geliştirme sürecinde aktivitelerin tamamının eksiksiz bir şekilde gerçekleştirilmesini şart koşar. Bu nedenle, her safhanın sonunda bir doküman oluşturulur ve bu model doküman odaklıdır. Yazılım süreci doğrusal olarak işler, yani bir sonraki safhaya geçmek için önceki safhanın aktivitelerinin tamamlanması gereklidir. Müşteri ve kullanıcıların katılımı sadece başlangıç safhasında mümkündür ve bu safhada gereksinimler tespit edilir ve ayrıntılı bir şekilde ele alınır. Ancak, tasarım ve kodlama aşamalarında müşteri ve kullanıcılarla diyaloğa girilmez. Analiz ve tasarım aşamaları oldukça detaylı ve uzun sürerken, kodlama ve test aşamaları daha kısa bir süreçtir.

### Avantajları

Şelale modelinin avantajları arasında, basit planlama ve kontrol olanakları yer alır. Bu model, adım adım ilerlediği için, her aşama için net bir plan yapmak mümkündür ve bu plana göre kontrol yapılabilir. Ayrıca, şelale modeli oldukça basit ve anlaşılabilir bir modeldir. Yazılım geliştirme sürecinin her safhası net bir şekilde belirlenmiştir ve her aşama ardışık bir şekilde ilerler. Bu nedenle, bu modeli kullanmak, karmaşık modelleri kullanmaktan daha kolaydır. Son olarak, şelale modeli düşük maliyetlidir. Bu modelde her aşama için ayrı bir doküman oluşturulduğu için, geliştirme sürecinde yapılan hatalar erken aşamalarda tespit edilebilir. Bu da maliyetleri düşürür.

### Dezavantajları

Kullanıcılar sadece tanım aşamasında katılım gösterebilirler. Şelale modelinde üst aşamalarda yapılan hataların düzeltilmesi ciddi zaman ve para kaybına sebep olabilir. Ürünün tamamlanması için tüm aşamaların eksiksiz tamamlanması gereklidir ve bu süre oldukça uzun olabilir. Bazı müşteriler için bu süre sabırsızlık yaratabilir, bu durumda daha çevik yöntemler tercih edilebilir.

## Neden Kullanılmamalı ?

Şelale yöntemi iletişim açısından tek yönlüdür ve safhalar arasında geri dönüş yoktur. Bu da hataların tamir edilmesini zorlaştırır. Safhaların kesin sınırlarının olması gerçekçi değildir ve projelerde safhalar arasındaki bu sınırlar belirsiz olabilir. Önceki safhalara geri dönülmek gerektiğinde projelerde kesintiler yaşanabilir. Model değişikliğe açık değildir ve başlangıçta yapılan hataların tespiti uzun zaman alabilir. Sistemin kullanılabilir hale gelmesi zaman alabilir ve modül implementasyonları için yapılan zaman tahminleri genellikle yanlıştır. Zamanında yapılmayan hataların düzeltilmesi maliyeti yükseltir ve planlama sürecini negatif etkiler.

**Model nerelerde kullanılır ?**

Şelale modeli, yüksek risk taşıyan, uzun vadeli, kritik projelerde sıklıkla tercih edilen bir yöntemdir. Modelin en önemli avantajı, analiz ve tasarım aşamalarına verilen ayrıntılı önem sayesinde sonraki safhalarda ortaya çıkabilecek hataların minimize edilmesidir. Bu yöntem, daha çok askeri gibi sektörlerde kullanılır ve bu sektörlerdeki bazı projelerde analiz ve tasarım süreci birkaç yıl sürebilir.

V modeli:

V-model, yazılım geliştirme sürecinde şelale modelinin ileri bir versiyonudur. Bu model, doğrusal bir yönde ilerleme yerine, süreç adımlarının kodlama evresinden sonra yukarı doğru eğim alarak V şekli oluşturmasını sağlar. V-model, geliştirme yaşam çevriminin her bir evresi arasındaki ilişkileri gösterir ve yatay ve dikey açılar zaman veya projenin tamamlanabilirliğini ve soyut seviyeyi gösterir.

Gereksinim analizi evresinde, sistemin gereksinimleri toplanır ve kullanıcıların ihtiyaçları analiz edilerek belirlenir. Kullanıcı gereksinimleri dokümanı, sistemin fonksiyonelliği, arayüzü, performansı, verisi ve güvenliği gibi kullanıcının beklediği gereksinimleri tanımlar. Bu dokümanlar, sistem tasarımcıları için bir kılavuz gibi hizmet eder ve kullanıcı kabul testleri bu evrede tasarlanır. Hem yazılımsal hem de donanımsal metodolojileri içeren gereksinimleri toplamak için farklı metotlar kullanılabilir.

Sistem tasarımı evresinde, sistem mühendisleri kullanıcı gereksinim dokümanlarını çalışarak sistemin işleyişini anlamaya çalışırlar. Eğer herhangi bir gereksinim uyuşmazlığı varsa, kullanıcı bilgilendirilir ve çözüm bulunur. Yazılım talimatname dokümanı, geliştirme evresinde üretilmesi için bir tasarı gibi hizmet eder ve genel sistem organizasyonu, menü yapıları ve veri yapılarını içerir. Aynı zamanda iş senaryoları, örnek pencereleri ve raporları da içerebilir.

Mimari tasarım evresi, bilgisayar mimarisinin ve yazılım mimarisinin tasarımını içerir. Bu evrede, seçilen mimariye uygun modüllerin listesi, her bir modülün özet fonksiyonelliği, arayüz ilişkisi, bağımlılıkları, veritabanı tabloları, mimari diyagramları ve teknoloji detayları sunulur. Entegrasyon test etme tasarımı özel bir evrede gerçekleştirilir.

Modül tasarımı evresi, düşük seviyeli tasarım olarak adlandırılır ve tasarım sistemi daha küçük birimlere veya modüllere ayrılır. Her bir modül programcıya doğrudan kodlamaya başlayacak şekilde açıklanır ve düşük seviyeli tasarım dokümanı, modülün ayrıntılı bir fonksiyonel mantığını içeren sahte kod içerir.

# Nerede kullanılır ?

Ürün tanımı, bir proje veya geliştirme sürecinde önemli bir adımdır ve birçok fayda sağlar. Bu adım, proje kapsamının ve hedeflerinin belirlenmesine yardımcı olur ve ekip üyelerinin proje hakkında daha net bir anlayışa sahip olmasını sağlar. Ayrıca, ürün tanımı, projenin tüm paydaşları arasında bir anlayış birliği sağlar ve projenin başarılı bir şekilde tamamlanması için gerekli kaynakların belirlenmesine yardımcı olur.

Eğer proje küçük ve sabit ise, ürün tanımı daha basit bir şekilde oluşturulabilir ve projenin amacı ve hedefleri daha açık bir şekilde belirlenebilir. Ancak, eğer proje büyük, karmaşık veya belirsiz gerekliliklere sahipse, ürün tanımı daha detaylı ve kapsamlı bir şekilde hazırlanmalıdır. Bu durumda, ekip üyelerinin daha fazla araştırma yapması ve gereksinimleri daha iyi anlaması gerekebilir.

Sonuç olarak, ürün tanımı, herhangi bir proje veya geliştirme sürecinde önemli bir adımdır ve proje kapsamının ve hedeflerinin belirlenmesine yardımcı olur. Proje küçük ve sabit ise daha basit bir şekilde hazırlanabilirken, büyük veya belirsiz gerekliliklere sahip projelerde daha kapsamlı ve detaylı bir şekilde hazırlanması gerekmektedir.

Avantajları

Doğrulama planları erken aşamalarda vurgulanır.

Proje yönetimi tarafında takibi kolaydır.

Kullanımı kolaydır​​.

Dezavantajları

Fazlar arasında tekrarlamaları kullanmaz.

Risk çözümleme ile ilgili aktiviteleri içermez.

Yazılım da diğer sistemler gibi zamanla evrimleşir.

Geliştirme devam ettikçe iş ve ürün gereksinimleri de değişkenlik gösterebilir

Son ürüne ulaşma düz bir çizgi ile ifade edilemez​​

Evrimsel Model:

Bu model, öncelikle geniş coğrafyalara yayılmış çoklu birimli organizasyonlar için tasarlanmıştır ve ilk tam ölçekli modeldir. Özellikle bankacılık uygulamalarında kullanıma uygun olan bu model, her bir aşamada üretilen ürünlerin tam işlevselliği içermesine olanak tanır. Pilot uygulama yaparak önce test edilir ve ardından güncellenerek diğer birimlere taşınır. Modelin başarısı, ilk aşamanın başarısına bağlıdır ve başarılı bir şekilde uygulandığında organizasyonun verimliliğini arttırabilir.

Avantajları

Kullanıcıların kendi gereksinimlerini daha iyi anlamalarını sağlar

Sürekli değerlendirme erken aşamalardaki​​ geliştirme risklerini azaltır

Hatalar azalır​​

Dezavantajları

Sürecin görünürlüğü azdır (düzenli teslim edilebilir ürün yoktur)

Sistemler sıklıkla iyi yapılandırılmaz (sürekli değişiklik yazılımın yapısına zarar verir)

Bakımı zordur

Yazılım gereksinimini yenilemek gerekebilir​​

Scrum, Agile proje yönetim metodlarından biri olup, yazılım süreçlerinin yönetimi için kullanılır. Esnek yapısı sayesinde ihtiyaca yönelik çözümler sunar ve düzenli geri bildirimler ile hedefe ulaşmayı hedefler. Müşteri odaklı bir yapıya sahip olduğu için geri bildirimler doğrultusunda şekillenir. Scrum Metodu iletişim ve takım çalışmasını ön planda tutar ve geliştirici, gözlemci ve tekrara dayalı prensipler üzerine kuruludur. Projedeki ilerlemeler ve sorunlar günlük olarak takip edilerek şeffaflık sağlanır. Ürünün parçaları düzenli aralıklarla teslim edilerek gözlem yapılır ve gereksinimler tekrar değerlendirilerek uyum sağlanır. Scrum, hafif ağırlıkta bir yönetim sürecidir ve projenin boyutuna göre uygulanabilir. Küçük takımların iletişim ve bilgi değişimine teşvik eder ve teknolojik değişimlere uygunluk sağlar. Ayrıca, dokümante edilmiş ve teftiş edilmiş sık yaratılan ürün sürümlerini kullanır.

Scrum avantajları

Projelerde esneklik ve uyarlanabilirlik oldukça önemlidir çünkü bu durum hem çalışanların hem de müşterilerin memnuniyetini arttırırken organizasyonel sinerjiyi de arttırır. Projeye yaratıcılık, yenilik ve kalite bakımından katkı sağlar ve aynı zamanda maliyetleri düşürür.

Projelerin açık ve net olması zaman kazandırır ve başarılı sonuçlanmasını sağlar. Projelerin anlaşılabilir ve yönetilebilir parçalara ayrılması olası sorunları hızlı bir şekilde tespit etmeyi ve düzeltmeyi mümkün kılar. Tüm ekip projenin tüm akışı hakkında bilgi sahibi olduğu için takım içi iletişim artar. Müşteriler ve geliştiriciler arasında güven oluşur ve bu da projenin başarılı bir şekilde sonuçlanmasına yardımcı olur.

Roller:

Scrum Master: Scrum geliştirme sürecinde kolaylaştırıcı rolü üstlenen kişi Scrum Master'dır. Scrum ekibiyle günlük toplantılar düzenler ve Scrum kurallarının doğru şekilde uygulanmasını sağlar. Scrum Master, liderlik özellikleriyle ekip üyelerini koçluk eder ve motive eder. Ayrıca, Sprintler için engelleri kaldırmaya ve ekibin en iyi koşullarda çalışmasını sağlamaya çalışır.

Ürün Sahibi: Ürün Sahibi (Product Owner), ürünün değerini maksimize etmeye çalışan kişidir. Tüm paydaşların beklentilerini karşıladığından emin olmak için ürünü kullanan veya kullanacak olan herkesin beklentilerini karşılamalıdır. Ürün beklentilerini belirleyip, Scrum projesi için ayrıntılı bir yapılacaklar listesi olan Scrum biriktirme listesini yönetir. Sprint hedeflerine ulaşılıp ulaşılmadığını kontrol eder ve sprint toplantılarında fikirlerini Scrum Master veya Scrum ekibiyle paylaşır.

Scrum Ekibi: Scrum Ekibi, kendini organize eden, işi tasarım, analitik ve geliştirme becerilerine sahip üç ila dokuz kişiden oluşan bir grup. Teslim edilebilir ürünler üretmek için sorunları çözmekle sorumludur ve her sprint hedefine ulaşmak için çalışır.

Karşılaştırma

Gereksinimin belirlenmesi çağlayan, V modeli ve evrimsel geliştirme için başlangıçtır. Çağlayan ve V modelinin maliyeti fazla iken Evrimsel modelinki uygundur. Çağlayan modeli basit, V modeli Orta düzeyde iken evrimsel model ise karmaşıktır. Esneklik bakımından bakacak olursak Çağlayan modeli bu konuda katı iken V modeli ve evrimsel model bu konuda kötüdür. 3 modelde uzun zaman almaktadır fakat uygulanmaları kolaydır.