YAZILIM YAŞAM DÖNGÜSÜ VE MODELLERİ

Yazılım yaşam döngüsü herhangi bir yazılımın ihtiyaç durumundan gerçekleştirilmesi ve sonrasında bakımına kadar geçen tüm süreçtir. Birçok aşamadan oluşur. Bu aşamalar planlama, analiz, tasarım, gerçekleştirim, test, bakım aşamalarıdır.



1)Planlama: Temel ihtiyaç ve gereksinimler belirlenir. Proje planlaması yapılır.

2)Analiz: Bu aşamada kesin gereksinimleri açıklığa kavuşturmak ve bunları belgeleme aşamasıdır.

3)Tasarım: Bu aşama sistem mimarisinin tanımlanmasına yardımcı olur.

4)Gerçekleştirim (Kodlama): Bu aşamada kodlama yapılır. Ürün programlanmaya başlanır.

5)Test: Programın müşterinin gereksinimlerine uygunluğu test edilir. Bütün program test edilir.

6)Bakım: Müşteriye teslimden sonra kullanımdan sonra 3 farklı etkinlik vardır.

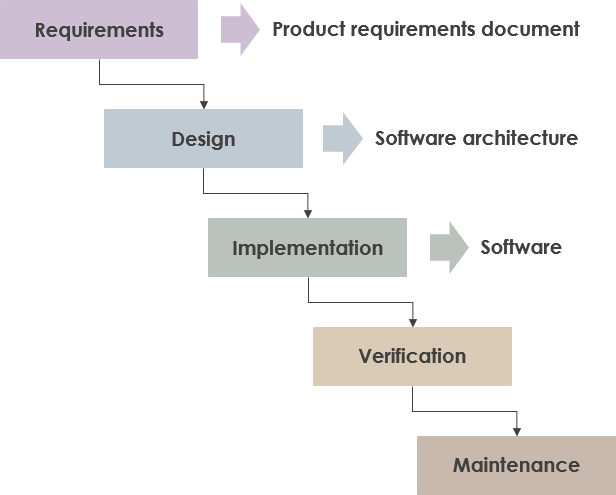
Hata düzeltme (Bug Fixing): Tespit edilmeyen hatalar ayıklanır.

Yükseltme (Upgrade): Daha farklı ve yeni sürümlere yükseltme işlem yapılır.

Geliştirme (Enhancement): Hali hazırda çalışan uygulamaya yeni özelliklerin eklenmesidir.

Yazalım yaşam döngüsüne bağlı olarak sürekli kullanılan ve temel kabul edilebilecek çeşitli modelleri bulunmaktadır. Bunlar;

1)ŞELALE MODELİ (WATERFALL MODEL): En eski, en temel modeldir. Günümüzde kullanımı gittikçe azalsa da iyi analiz edilmiş ve kısa sürede üretimi gerçekleştirilecek projelerde kullanılabilecek en uygun modeldir. Bu modelde işlerin her biri aşama aşama yapılır ve bir önceki aşama tamamlanmadan diğer aşamaya geçilmez. Geniş bir dokümantasyonu vardır. Barok modeli ile burada farklılık gösterir. Barok modelinden farklı olarak dokümantasyonu ayrı bir süreç değil üretimin doğal bir parçası olarak ele alır. Şelale modelinde analiz aşamasında detayın atlanmaması için müşteri ve sistem ihtiyaçlarının en ince ayrıntısına kadar belirlenmesi gerekir. Analiz ve tasarım aşaması en uzun süren aşamalardır. Bu iki aşama eksiksiz yapılsa da uzun süreli projelerde kodlama ve test aşamalarında yapılan değişikliklerin yazılıma yansıtılma maliyeti çok yüksektir. Kullanıcının rolü yoktur buda yazılımın tamamlandıktan sonra isteklerde farklılıklara yol açabilir. Bu değişen isteklerde yazlım geliştirme maliyetini büyük oranda yükseltir.



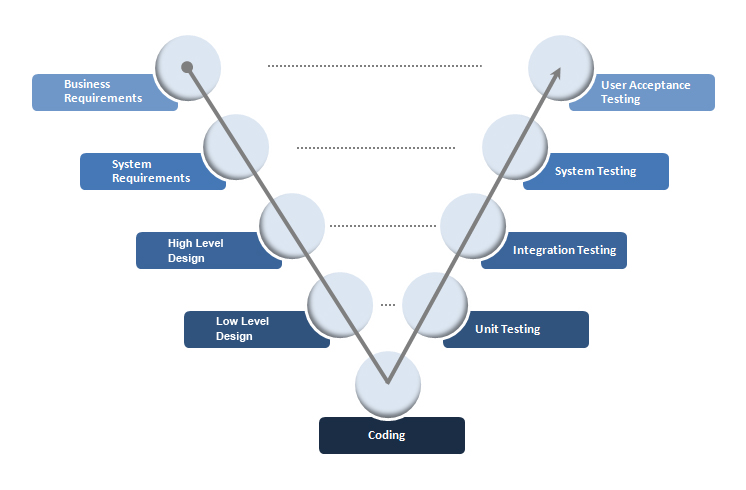
2)V SÜREÇ MODELİ: V süreç modelinin 3 temel çıktısı vardır;

1)Kullanıcı modeli: Kullanıcını istekleri ve bunların proje içinde tanımları kullanılır.

2)Mimari modeli: Sistem tasarımı ve bunları sınama işlemi yapılır.

3)Gerçekleştirim: Kodlama ve bu kodlarının denenmesi işlemi yapılır.

BT( Bilgi Teknolojileri) projeleri için uygun bir modeldir. Bu model kullanıcının projeye olan katkısını arttırmaktadır. Aşamalarda tekrar bulunmaz.



3)BAROK MODELİ: 1970’li yıllarda ortaya çıkmıştır. Aşamalar arası geri dönüş yoktur. Dokümantasyon ayrı bir aşama olarak ele alınır. Günümüzde kullanılmamaktadır.

4)GELİŞİGÜZEL MODEL: Bu yönteme model demek doğru olmaz. Bir model veya yöntem yoktur. Genelde tek bir kişinin basit programlama içeren projelerde ortaya çıkar. Kodların bakımı çok zordur. 1960’lı yıllarda kullanılmıştır.

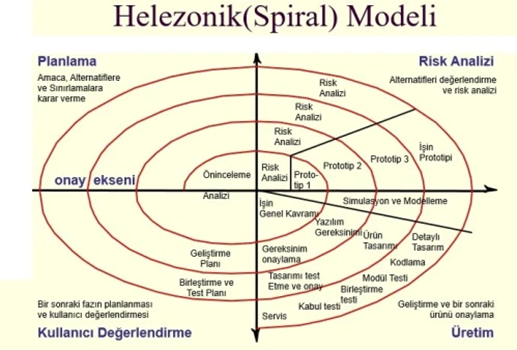
5)HELEZONİK MODEL: Planlama: Ara ürün için planlama, amaç belirleme, bir önceki ara ürün ile birleştirme.

Risk Analizi: Risk oluşturabilecek seçeneklerin araştırılması ve belirlenmesi.

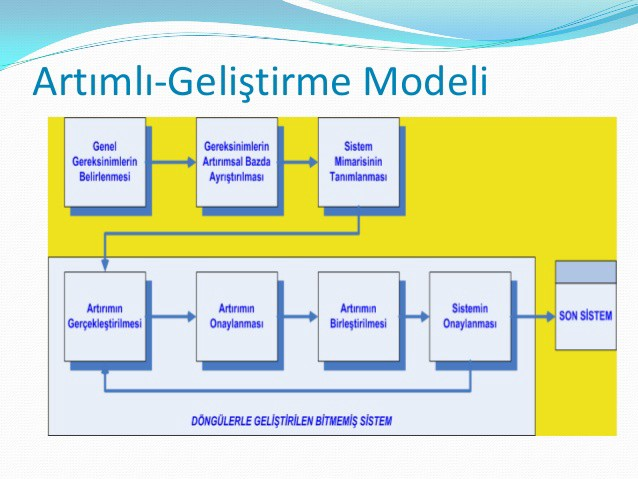
Üretim: Ara ürün üretilmesi

Kullanıcı Değerlendirmesi: Üretilen ara ürüne kullanıcı tarafından yapılan sınamalar.

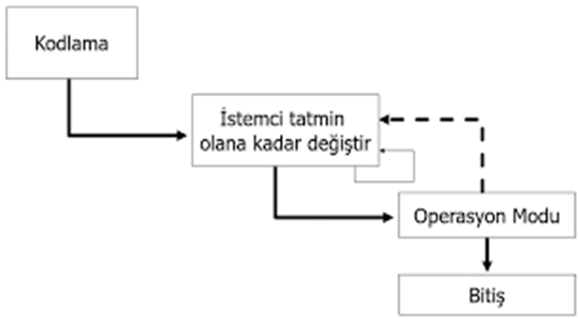
Risk odaklı bir modeldir. Çok sayıda prototip üretilir ve bu prototipler kullanıcı tarafından denenerek değerIendirmelere tabi tutulur.



6)ARTIMSAL GELİŞTİRME SÜREÇ MODELİ: Sistem tek seferde teslim edilmez. Geliştirme ve teslim olmak üzere parçalara ayrılır. Erken teslimler sonraki teslimler için prototip işlevi görür. Kullanıcını ihtiyaçları önceliklendirilir. Gereksinimler önem ve birbirine bağlılıklarına göre sıralanırlar. Bir parçanın geliştirmesi başlandığında, gereksinimleri dondurulur. Ara ürünler sürekli üstlerine bir şeyler katarak çıkarılır. Bu modelde kullanım ve üretim eş zamanlı olarak devam eder. Uzun soluklu projeler için uygundur.



7)KODLA ve DÜZELT YAŞAM-DÖNGÜ MODELİ: Direkt olarak yazılım ürünü gerçekleştirilir. Sistem istenilen şekle gelene kadar sürekli geliştirilerek devam edilir. Büyük projeler için kullanılmaz. Bakım aşaması az ve zordur. Emeklilik safhası vardır. Yazılım geliştirmenin en kolay yolu ama en pahalısıdır. Bu kolaylık küçük ve tecrübesiz yazılım projelerinde model olarak kullanılır.



8)ÇEVİK MODELLER: Projenin daha az maliyetle, daha fazla üretkenlikle ve kalitesini yükseltmek amacıyla kullanılan modellerdir.

A) EXTREME PROGRAMMING: Kent Beck tarafından 1999 yılında ortaya atılmış bir model türüdür. Grup içeresindeki iletişime önem vermektedir. 4 ana değerden oluşur bunlar iletişim, basitlik, geri bildirim, cesarettir. Yazılım ekibi ve müşteri arasındaki iletişime önem verir. Bu sayede sorunların çabuk fark edilmesi amaçlanmıştır. Karmaşık çözümler kullanılmaz o anki en basit çözüm uygulanır. Geri bildirim için müşteri grubun bir parçası olarak görülür, toplantılar yapılıp isteklerin doğru şekilde karşılanması sağlanır. Yazılımcının cesaretli olması ve gerektiğinde en baştan başlayıp yılmadan projeyi yapması gerekir aksi halde bu durumdan korkan bir yazılımcının hızlı bir şekilde projeyi yürütmesi mümkün değildir.

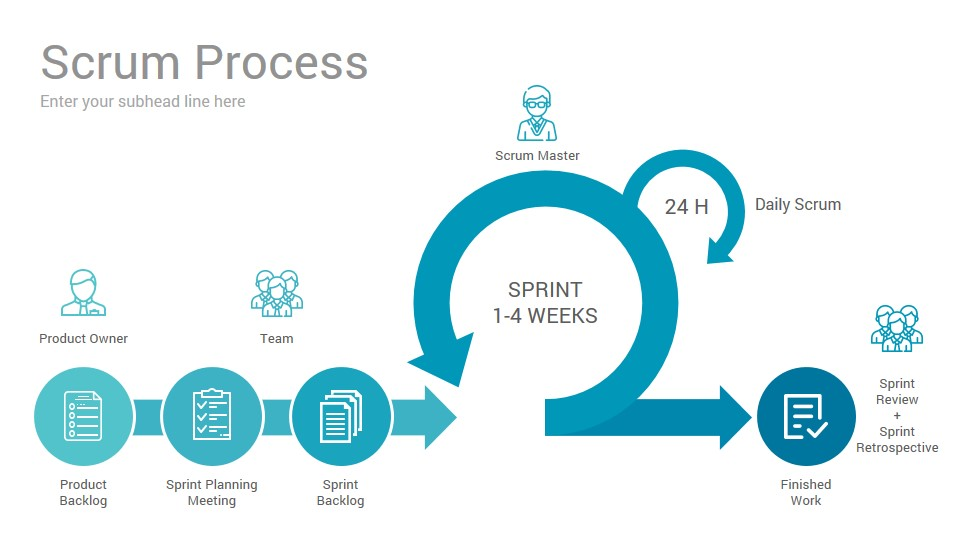
B) SCRUM: Scrum bir projeyi parçalara bölerek (sprint) geliştirme yöntemidir. Scrum sadece yazılımda değil başka birçok sistemde kullanılabilir. 3 temel kavramdan oluşur.

1) Roller: Ürün Sahibi, Scrum Yöneticisi, Scrum Takımı

2) Toplantılar: Sprint Planlama, Sprint gözden geçirme, Günlük Scrum Toplantısı

3) Bileşenler/Araçlar: Ürün Gereksinim Dokümanı, Sprint Dokümanı, Sprint Kalan Zaman Grafiği

SCRUM GÜNÜMÜZDE NEDEN POPÜLER:Scrum günümüzde sıkça kullanılan çok popüler bir modeldir. Bunun belli başlı sebepleri vardır. Scrum projenin hem parasal hem zaman açısından maliyetini büyük ölçüde düşürür. İletişim büyük bir önem arz ettiği için hataların fark edilmesi gecikmez. Teknolojik açıdan güncel süreçlere uyum sağlayabilir. Kullanıcı ile iletişim kuvvetli olduğundan sürekli geri bildirim verir. Karmaşık gözüken ve iyi analiz edilmemiş projeler için elverişli olması onu popüler yapan özellikleridir.



PROJELERİN ÖZELLİKLERİNE GÖRE HANGİ MODELLERİ KULLANMALIYIZ:

Şelale Modeli: Boyutu çok büyük olmayan, iyi analiz edilmiş projelerde kullanılmaya uygundur.

V Süreç Modeli: Küçük ve orta ölçekli, belirsizliklerin az olduğu projelerde kullanılır.

Helezonik Modeli: Büyük ölçekli ve uzun zaman alan projelerde kullanılır.

Artımsal Geliştirme Süreç Modeli: Uzun süreli büyük projelerde kullanılır.

Kodla ve Düzelt Modeli: Tek kişilik, küçük çaplı ve herhangi bir zaman kısıtlaması olmayan projelerde kullanılır.

Çevik Modeller: Büyük ve orta ölçekli, fazla zaman almayan projelerde kullanılır.

Kaynaklar:

<https://fikirjeneratoru.com/yazilim-proje-yonetimi-yontemleri/>

Doç.Dr.Deniz Kılınç, Bakırçay Üniversitesi Yazılım Mühendisliği Giriş Dersi 3. Ve 4. Hafta Sunumları

<https://www.techwell.com/2013/02/why-scrum-so-popular>

Ali Alper Şahin 200601022