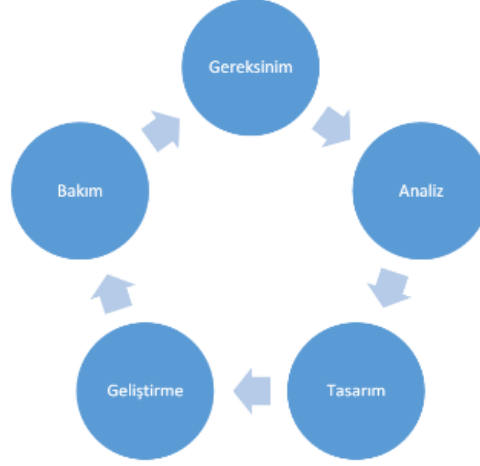


YAZILIM GELİŞTİRME YAŞAM DÖNGÜSÜ (SDLC)



SDLC, yazılım geliştirme süreçlerinin planlanmasından uygulanmasına, test edilmesine ve bakımına kadar olan aşamaları tanımlar. Bu döngü, yazılım projelerinde sistematik bir yapı sağlayarak, kaliteli, güvenli ve hedefe uygun bir yazılım üretmeyi amaçlar. SDLC'nin temel amacı, yazılımın zamanında, bütçesine uygun ve gereksinimleri karşılayacak şekilde geliştirilmesini sağlamaktır. SDLC genel olarak 5 aşamaya ayrılmaktadır;

SDLC'nin Beş Aşaması

1. Gereksinim (Requirement)
2. Analiz (Analysis)
3. Tasarım (Design)
4. Geliştirme (Development)
5. Bakım (Maintenance)

SDLC'nin Avantajları

1. SDLC, yazılım geliştirme sürecinde adım adım bir yapı sağlar. Her aşama belirli hedeflere odaklanarak işlerin daha sistematik yapılmasını sağlar.
2. Gereksinim analizi ile yazılımın gereksinimleri baştan net bir şekilde belirlenir. Bu, son üründe kullanıcı beklentilerinin tam olarak karşılanmasını sağlar.
3. Projelerin planlanmasını, kaynakların verimli kullanılmasını ve süreç boyunca potansiyel risklerin belirlenip azaltılmasını sağlar.
4. SDLC sayesinde projelerin tamamlanma süresi, maliyeti ve kullanılan kaynaklar önceden planlanabilir. Bu sayede beklenmedik gecikmelerin ve maliyet artışlarının önüne geçilebilir.

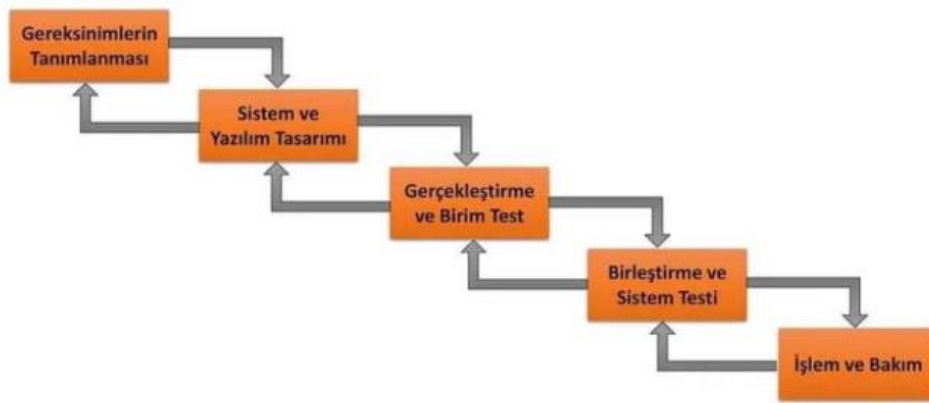
SDLC'nin Dezavantajları

1. SDLC, özellikle Waterfall Modeli gibi yaklaşımlarda, bir aşama tamamlanmadan diğer aşamaya geçilemediği için değişen müşteri gereksinimlerine hızlı uyum sağlamakta zorlanabilir. Bu, çevik (agile) projelere göre daha az esneklik sağlar.
2. SDLC süreci detaylı bir yapı izlediğinden, zaman ve maliyet açısından diğer yöntemlere göre daha yüksek olabilir. Bu da küçük ve hızlı projeler için SDLC'nin kullanılışlığını azaltabilir.
3. SDLC, çok fazla dokümantasyon gerektirir. Bu belgeler projeyi kontrol etmede yararlı olsa da, belge oluşturma ve düzenleme süreci zaman alıcı olabilir.
4. Eğer basit bir proje yapılacaksa SDLC'ye gerek yoktur. Ancak büyük ve karmaşık projelerde daha sık kullanıldığı için projede bir uzman gerekir. Uzman olmadığı takdirde kolaylaştıracağını düşündüğümüz SDLC modelleri projeyi içinden çıkılmaz hale getirir.

Yazılım Geliştirme Yaşam Döngüsü Modelleri (SDLC)

Waterfall (Şelale/Çağlayan) Modeli : 1970'lerde Winston W. Royce tarafından tanıtılan ve yazılım geliştirme ile proje yönetimde yaygın olarak kullanılan bir yaklaşımdır. Waterfall modeli, belirli ilkeler üzerine kuruludur:

- **Sıralı İlerleme:** Proje adım adım ilerler ve bir aşama tamamlanmadan diğerine geçilmez.
- **Dökümantasyon:** Her aşamada dokümantasyon yapılır, bu da projenin ilerleyişini ve gereksinimlerin net olarak anlaşılmasını sağlar.
- **Kontrol ve İzleme:** Proje yönetimi, her aşamanın sonunda ilerlemeyi değerlendirir.
- **Değişiklik Yönetimi:** Değişiklik yapmak zordur ve maliyetlidir, bu nedenle gereksinimlerin baştan net bir şekilde belirlenmesi önemlidir.



Modelin aşamaları'nın sayısı değişkendir ancak yapısı aynıdır. Ben genellikle kullanılan 5 aşamayı söylemek istiyorum. Bu **bütün aşamalarda döküman çıkarmak zorunludur**. Aksi takdirde bir aşamadan diğer aşamaya geçilmesi mümkün değildir.

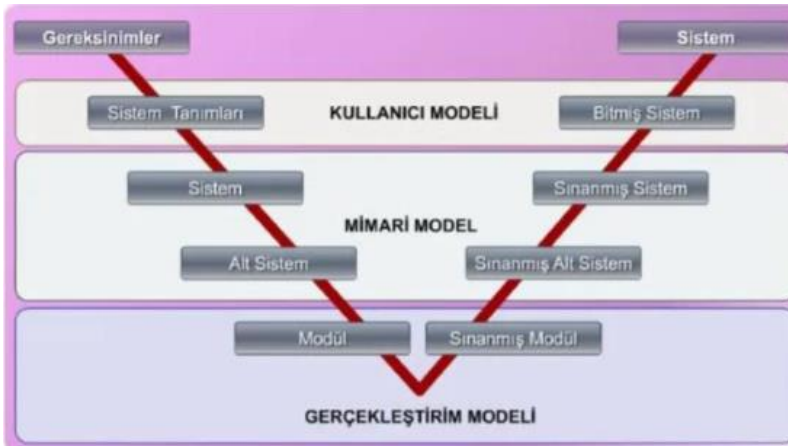
- 1) Gereksinimlerin Tanımlanması
- 2) Sistem ve Yazılım Tasarımı
- 3) Gerçekleştirme ve Birim Test
- 4) Birleştirme ve Sistem Testi
- 5) İşlem ve Bakım

Waterfall Modeli'nin Avantajları

- Her aşama belirgin ve sıralıdır, bu da özellikle yeni başlayan ekipler için kolay anlaşılır bir yöntem sunar.
- Önceden belirlenmiş aşamalar sayesinde proje yöneticileri için zaman çizelgesi ve bütçe planlaması yapmak kolaydır.
- Her aşamada yapılan ayrıntılı dökümantasyon, projenin anlaşılmasını ve sürdürülebilirliğini artırır.

Waterfall Modeli'nin Dezavantajları

- Süreç sırasında yeni gereksinimler ortaya çıkarsa, bu gereksinimleri karşılamak zor olabilir.
- Bir aşamadayken önceki aşamalardan kaynaklı bir sorun ortaya çıktığında sorunlu aşamaya dönmek hem zaman kaybettirir hem de maliyetli olur.
- Her aşamada dökümantasyon hazırlamak ciddi zaman maliyeti içerir.



V Model:

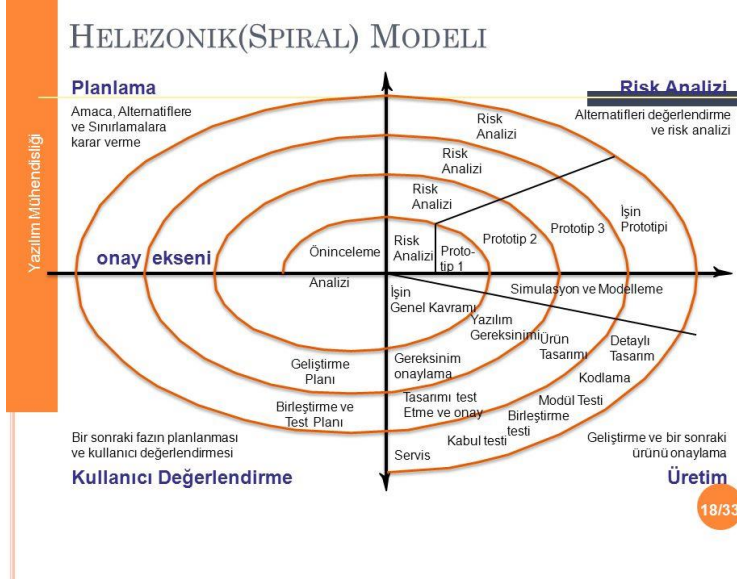
V Model, yazılım doğrulaması ya da daha geniş bir ifadeyle, doğrulama ve geçerlilik (V&V) süreçlerini içerir. Bu model, bir sistemin hem teknik spesifikasyonlarına uygunluğunu hem de müşteri beklentilerini karşıladığını göstermeyi amaçlar. İlk bakışta şelale modeline benzese de V Model'in en belirgin farkı, her geliştirme aşamasının kendi test süreci ile birlikte yürütülmesidir.

V Model'in Yapısı:

V Model, geliştirme ve test süreçlerini birbirine paralel olarak 'V' harfi şeklinde ilerleten bir yapıya sahiptir. Geliştirme aşamaları 'V'nin sol tarafında, test aşamaları ise sağ tarafında konumlanır. Bu yapı sayesinde testler, proje geliştirilirken yapılır ve sürecin iç içe ilerlemesini sağlar. Aşamalar aşağıdaki gibidir:

1. **Kullanıcı Modeli:** Geliştirme sürecinin kullanıcı ile olan ilişkilerini tanımlamakta ve sistemin nasıl kabul edileceğine ilişkin sına belirimleri ve planları ortaya çıkarılmaktadır.

2. **Mimari Model:** Sistem tasarımı ve alt sistem ile bütün sistemin test işlemleri yapılmaktadır.
3. **Gerçekleştirim Modeli:** Yazılım modüllerinin kodlanması ve testine ilişkin fonksiyonlar ele alınmaktadır.



Helezonik sarmal model (Spiral Model) : Spiral model ilk olarak 1988 yılında Boehm tarafından, waterfall modeli'nin dezavantajlarını gidermek için geliştirilmiştir. Bu model, 4 aşamadan oluşmaktadır :

planlama, risk analizi, üretim ve kullanıcı değerlendirme.

Spiral model, waterfall, artırımlı ve evrimsel geliştirme modellerinin gelişmiş halidir. Spiral modelin temel farkı, risk odaklı olması ve sürecin sarmal bir biçimde ilerlemesidir. Bu modelin en önemli özelliği, her aşamada risk analizi yapılmasıdır.

Spiral model, diğerlerinden yine farklı olarak, süreci oluşturan aşamalardan tekrar tekrar geçilmesini ve her geçişte projenin ilerlemesini amaçlar. Örneğin, projenin her döngüsünde yeni bir prototip geliştirilmesi isteniyorsa, toplamda üç prototip ve üç döngü boyunca, proje aşamalarında içeriden dışarıya doğru spiral şekilde ilerlenir.

Spiral'de 4 aşama vardır.

- 1) Planlama
- 2) Risk Analizi
- 3) Üretim
- 4) Kullanıcı Değerlendirme

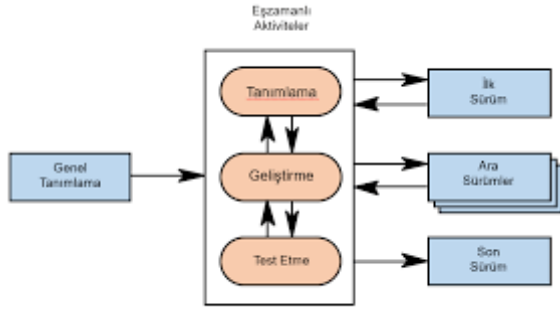
Spiral Modeli'nin Avantajları

- 1) Her aşamada risk analizi yapılması ile olası hata ve problemlerin önüne geçilir.
- 2) Yazılım kademeli olarak geliştirildiği için, kullanıcı gereksinimleri veya teknolojiye ilişkin değişiklikler her döngüde göz önünde bulundurulabilir. Bu da modeli biraz dinamik hale getirir.
- 3) Proje kaynakları, iterasyonlar sırasında daha etkili şekilde dağıtılabilir.

Spiral Modeli'nin Dezavantajları

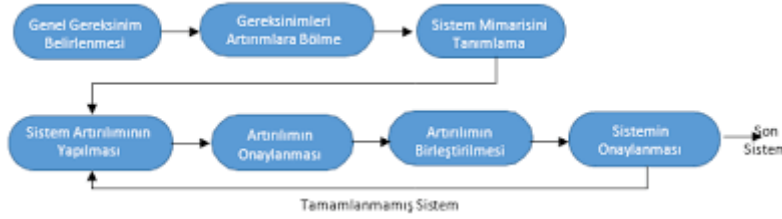
- 1) Spiral Modeli'nin risk analizi ve prototip geliştirme süreçleri, diğer modellere göre daha fazla zaman ve maliyet gerektirebilir. Küçük projelerde bu model maliyet açısından uygun olmayabilir.

- 2) Spiral Model, yapılandırılması ve yönetilmesi açısından diğer modellere kıyasla daha karmaşıktır. Bu nedenle uzman ekipler gerektirir.



Evrimsel Geliştirme Süreç Modeli : Evrimsel geliştirme süreç modeli ilk tam ölçekli modeldir. Coğrafi olarak geniş Alana yayılmış, çok birimli organizasyonlarda kullanılmak üzere geliştirilecek projeler için önemlidir.

Evrimsel model,artırımlı geliştirme modelinin biraz daha gelişmiş halidir.. Artırımlı geliştirme modelinde ihtiyaçların başlangıçta belirli olma gereksinimi bu modeller yoktur. Zaman içerisinde sürümler ile fark edilerek eklenmektedir.



Artırımlı Model

(Incremental Model) :

Artırımlı model, yazılım geliştirme sürecini parçalara bölerek her bir parçanın bağımsız olarak geliştirilip ayrı teslim edilmesine

dayanan yazılım yaşam döngüsü modelidir. Bu modelde proje, işlevsel bölümlere ayrılarak küçük ve yönetilebilir dilimlere bölünür (Pastayı dilimlere bölmek gibi). Her dilim veya artırım, belirli bir fonksiyonellik içerir ve tamamlandığında müşteriye sunulabilir.

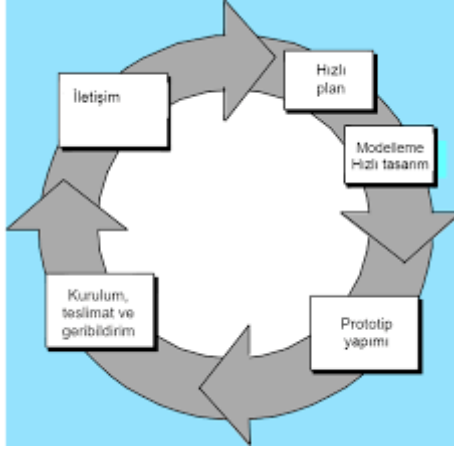
Bu sayede;

- Yapılan geliştirme çalışmaları hakkında müşteri geri bildirimi almak daha kolaydır.
- Yazılım tam bitmese bile işlevsel olan yararlı yazılımın müşteriye erken teslimi mümkündür.
- Erken artırımlar prototip gibi davranarak, gereksinimlerin daha iyi anlaşılmasını sağlar.



Kodla ve düzelt modeli : İhtiyaç analizi tam olarak bitmeden ya da kısmen tamamlanmış olarak, herhangi bir tasarım yapmadan doğrudan kodlama işlemine geçilmektedir.

SDLC Nedir,SDLC Modelleri (Waterfall, V model, Helezonik, Evrimsel, Artırmalı, Kodla ve Düzelt, Hızlı Prototipleme)



Hızlı Prototipleme Yaşam Döngü Modeli : Bu modelde, projeyi geliştirmek için öncelikle bir prototip oluşturulur ve bu prototip üzerinden çalışmalar sürdürülür. Gereksinim tanımlama aşamasında hızlıca yapılan kısmi gerçekleştirmelerle ihtiyaçlar belirlenir ve prototip, gereksinimler netleştikçe iyileştirilir. Müşteri memnuniyetine ulaşana kadar prototip üzerinde düzeltmeler yapılabilir. Bu süreçte kullanıcı, sistem gereksinimlerini doğrudan görme imkanı bulur, böylece karmaşa ve yanlış anlaşılımların önüne geçilir.

KAYNAKÇA

Ç.YAŞAR, “Yazılım-Muhendisligine-Giris” Github, [Online]. Erişim Linki: <https://github.com/cyasar34/Yazilim-Muhendisligine-Giris>. Son Erişim Tarihi: 17.11.2024

<https://www.linkedin.com/pulse/yaz%C4%B1%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-ve-modelleri-ekinsu-o%C4%9Fuz/>

<https://www.linkedin.com/pulse/yaz%C4%B1%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-beyza-nur-karako%C3%A7/>

Yılmaz, Y. (2007). Bilişim Sistemlerinin Geliştirilmesinde Doğru Yaşam Döngüsü Modelinin Seçimi. Öneri Dergisi, 7(27), 171-176.

Keskinkılıç, M., & Özmen, E. YAZILIM PROJELERİNDE YAZILIM GELİŞTİRİCİLERİN YAZILIM SÜREÇ MODELLERİNİ KULLANIM FARKINDALIKLARI. Akademi Sosyal Bilimler Dergisi, 5(15), 61-78.

<https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf>

<https://www.linkedin.com/pulse/yaz%C4%B1%C4%B1m-ya%C5%9Fam-d%C3%B6ng%C3%BCs%C3%BC-ve-modelleri-ramazan-arda-a%C5%9Fc%C4%B1/>

<https://www.youtube.com/watch?v=u2rU8Wss4bw>

<https://www.youtube.com/watch?v=OnRZSXSo7os>

<https://medium.com/@kabusteyyimhakemokeha/art%C4%B1ml%C4%B1-incremental-yaz%C4%B1%C4%B1m-geli%C5%9Firme-modeli-4ca54315054a#:~:text=Art%C4%B1ml%C4%B1%20model%2C%20yaz%C4%B1%C4%B1m%20geli%C5%9Firme%20d%C3%B6ng%C3%BCs%C3%BCn%C3%BCn,ne%20geli%C5%9Ftirilece%C4%9Fi%20hakk%C4%B1nda%20fikir%20mevcuttur.>