

SDLC

software development life cycle
(yazılım geliştirme yaşam döngüsü)

1. Ders

Ayşe Sezen

Full-Stack Automation Engineer

SDLC NEDİR?

Yazılım Geliştirme Yaşam Döngüsü'nü (**Software Development Life Cycle ,SDLC**), yazılımları tasarlamak, geliştirmek ve test etmek amaçlı kullanılan bir süreç olarak ifade edebiliriz. Yazılımın nasıl geliştirileceği, sürdürüleceği ve daha iyi hale nasıl getirileceğini açıklayan bir plandan oluşmaktadır. Buradan da yazılımın aslında bir ürün olduğu ve o ürünün de bir yaşam süreci olduğunu gözlemlemiş oluyoruz.

SDLC, *kullanıcı taleplerini (user expectation)* karşılayacak şekilde zaman ve maliyet tahminleri dahilinde tamamlanması beklenen, *yüksek kaliteli (high quality)* yazılım üretmeyi hedefler.

hardware (donanım)

Donanımlar bilgisayarların elle tutulabilir,gözle görülebilir somut fiziki parçalarıdır.

Örneğin; Ekran Kartı, Klavye, Monitör, Anakart, Fare, Yazıcı, Harddisk gibi bileşenler.

Donanımlar kasanın içerisinde yer alabileceği gibi kasanın dışında da yer alabilir. Donanım iki kısma ayrılır. **İç Donanım** (Dahili Donanım) veya **Dış Donanım** (Harici Donanım).



software (yazılım)



Bilgisayar donanımlarının çalışmasını ve kendi aralarında haberleşmesini olanaklı hale getiren programların tamamına yazılım adı verilmektedir. Yazılım olmadan donanım olmaz. Yazılım çalıştırır, donanım ise yazılım sayesinde çalışır.

Yazılımlar **sistem yazılımları** ve **uygulama yazılımları** olarak ikiye ayrılır.

NEDEN SDLC ?



Twitter'ın güvenlik kalkanlarını aşmayı başaran siber korsanlar, platformdaki birçok tanınmış hesabı (Elon Musk, Bill Gates, Jeff Bezos, Uber, Apple ve Kanye West gibi) ele geçirerek kullanıcılardan bitcoin istedi.

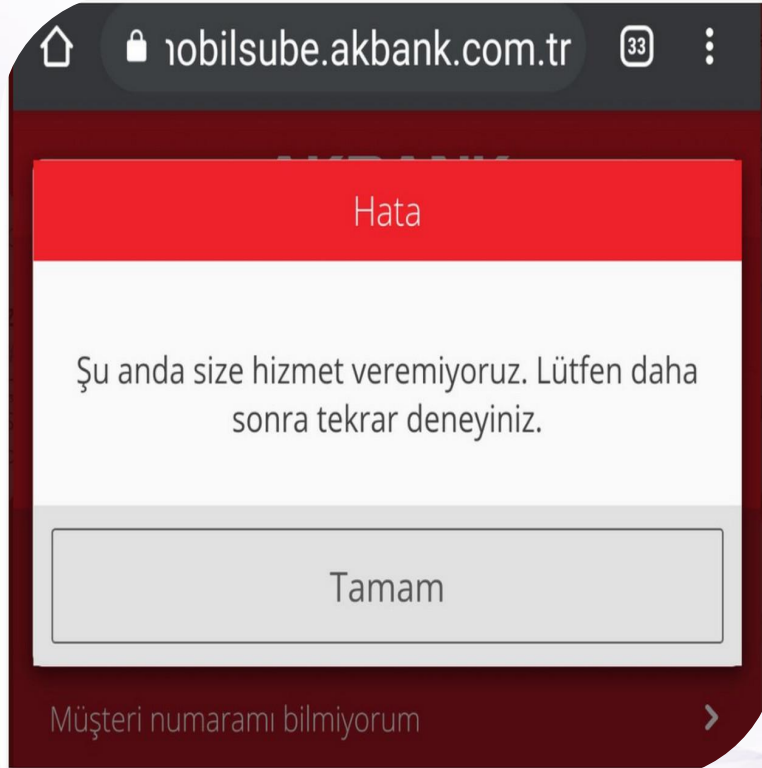
NEDEN SDLC ?

Endonezya ve Etiyopya'daki uçak kazalarının ardından Boeing model 737-8 Max ve 737-9 Max uçaklarının, tüm uçuşlarının birçok ülke ve hava yolu şirketi tarafından durdurulması, şirketin hisselerinin adeta çakılmasına sebep oldu. Gelen son verilere göre Boeing'in piyasa değeri yaklaşık 22 milyar dolar eridi.



22 Milyar Dolar Kaybetti
422 dolardan el değiştiren Boeing hisseleri 377 dolara indi. Kazadan önce Boeing'in piyasa değeri 238 milyar doların üzerindeyken bugün itibarıyla 213 milyar dolara geriledi ve 22 milyar doların üzerinde değer kaybetti.

NEDEN SDLC ?



Akbank müşterileri, bankanın ana sisteminin çökmesi sebebiyle günlerce işlem gerçekleştiremedi. Banka, web sitesinde yaptığı açıklamada sıkıntının merkez binalarındaki ana bilgisayarda meydana gelen teknik arızadan kaynaklandığını bildirdi.

NEDEN SDLC ?

Amazon 70 bin TL'lik ürün 3 kuruşa satıldı

16/12/2014 09:52 | A⁺ A⁻

ABD'li online alışveriş devi Amazon.com'un İngiltere operasyonunda yaşanan bir hacker skandalı, 20 bine yakın ürünün 1 penny'lik (yaklaşık 3.6 kuruş) fiyattan satılmasına yol açtı.



Amazon.co.uk sitesi üzerinden yapılan satışlarla ilgili bir açıklama yapan şirket, olaya bir yazılım hatasının yol açtığını bildirdi.

Software Hatalarının Sonuçları

NASA, 4 Haziran 1996'da fırlatılması planlanan Ariane 5 uzay aracını kodlarken, Ariane 4 roketinin kodlarını kopyalayarak, bir hata yaptığının farkında değildi. O gün fırlatma için geri sayım yapıldı ve roketin motorları ateşlenerek kalkış başladı.

Hızlanarak yoluna 37 saniye boyunca devam eden Ariane 5 roketi; o saniyeden sonra yanlış yöne doğru 90 derece dönmeye başladı. Bu durum, roketin kendini imha etme mekanizmasını tetikledi. Bu kaza, NASA'ya 370 milyon dolara mal oldu.

Software Hatalarının Sonuçları

Therac-25 makinesi, kanser hastalarının tedavisinde kullanılmak için tasarlanmıştı. Yapılan değişiklik Therac-20'de güvenlik önlemi olarak bulunan elektromekanik güvenlik kilitlerinin yazılımsal güvenlik önlemleriyle değiştirilmesiydi. Ne yazık ki gelişim olarak görülen bu güncelleme, Therac-20'nin kodlarında bulunan ancak fark edilemeyen hatanın, Therac-25'te ortaya çıkmasına sebep oldu. Bu bug yüzünden yaklaşık 5 hastanın ağır dozda radyasyon sebebiyle hayatını kaybettiği raporlandı.

Software Hatalarının Sonuçları

Mars Climate Orbiter Hatası (23 Eylül 1999): Gezegenler arası ilk iklim uydusu olarak 1997'de fırlatıldı. Mars Orbiter, 1999'da Mars'ın yörüngesinde kayboldu. Kazanın yazılımda kullanılan İngiliz ölçü birimlerinin metrik sisteme yanlış çevrilmesinden kaynaklandığı belirtildi. NASA'da bir ekip hesaplarında İngiliz ölçü birimini (inç) kullanırken, projeye katılan diğer ekip ise metrik (cm) sistemi kullanmıştı. 125 milyon dolarlık uydunun yörüngeye sabitlenmeye çalışırken Mars'a olması gerekenden daha fazla yaklaşarak imha olduğu düşünülüyor.

Software Hatalarının Sonuçları

Dakikalar içinde kaybedilen 460 milyon dolar. New York borsasında piyasaları yönlendiren en büyük şirketlerden biri olan Knight Capital, 1 Ağustos 2012'de yeni bir yazılım güncellemesi yapma kararı aldı. Saat 09.00 sularında New York Borsası işlemler için açıldı ve Knight Capital'ın yatırımcıları, varlıklarını satmak veya almak için talimat verdi. Yalnızca 45 dakika sonra Knight Capital'ın sunucuları 4 milyon işlem gerçekleştirdi ve şirkete 460 milyon dolar kaybettirerek iflasın eşiğine getirdi.

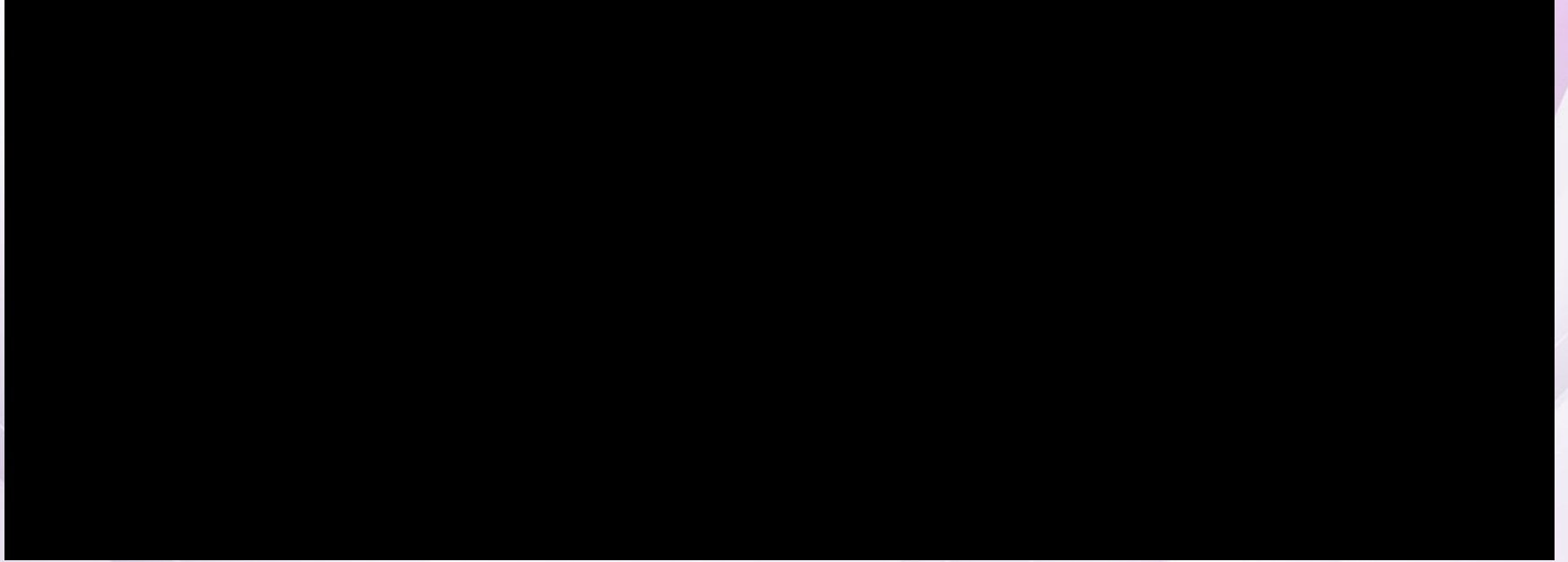
Olay, bir teknisyenin yeni Perakende Likidite Programı (RLP) kodunu, Knight'ın hisse senedi siparişleri için otomatik yönlendirme sistemi olan sekiz adet SMARS bilgisayar sunucusundan birine kopyalamayı unutması sonrasında meydana geldi.

Software Hatalarının Sonuçları

1991 yılının şubat aylarında gerçekleşen Körfez Savaşı sırasında ABD'nin Suudi Arabistan'ın Zahran şehrindeki üssünde bir patlama yaşandı. Patlamanın sebebi ise üste bulunan anti balistik füze sisteminin doğru çalışmamasıydı. Yapılan sorgulamaların ve araştırmaların sonucunda patlama sebebinin üste bulunan anti balistik füze sisteminin bir yazılım hatası yüzünden ateşlenmemesi olduğu anlaşıldı.

Bir insan için inanılmaz küçük olan 0,33 saniye, Al Hussein füzelerini takip etmek için yapılan bir sistem için inanılmaz büyük bir hataydı. MIM-104 Patriot, havada bir cisim olduğunu algılamayı başardı ancak bug yüzünden cismi takip edemedi ve bunun bir füze olduğunu anlayamadı. Engellenemeyen füze yüzünden üste bulunan 28 asker hayatını kaybetti.

NEDEN SDLC ?



1996 - 2022 yılları arası dünya genelindeki en popüler web tarayıcısı değişimleri

NEDEN SDLC ?

The 100 Billion Dollar Club

Billionaires with more than \$100 billion in net worth
(as of Oct 15, 2021)

	Net worth (in \$b)	Year-to-date change (in \$b)
Elon Musk  Tesla	230	+60
Jeff Bezos  Amazon	191	+1
Bernard Arnault  LVMH	166	+51
Bill Gates  Microsoft	130	-2
Larry Page  Alphabet	126	+43
Mark Zuckerberg  Facebook	122	+19
Sergey Brin  Alphabet	121	+41
Larry Ellison  Oracle	110	+30
Steve Ballmer  Microsoft	108	+28
Warren Buffett  Berkshire Hathaway	102	+15
Mukesh Ambani  Reliance Industries	102	+25

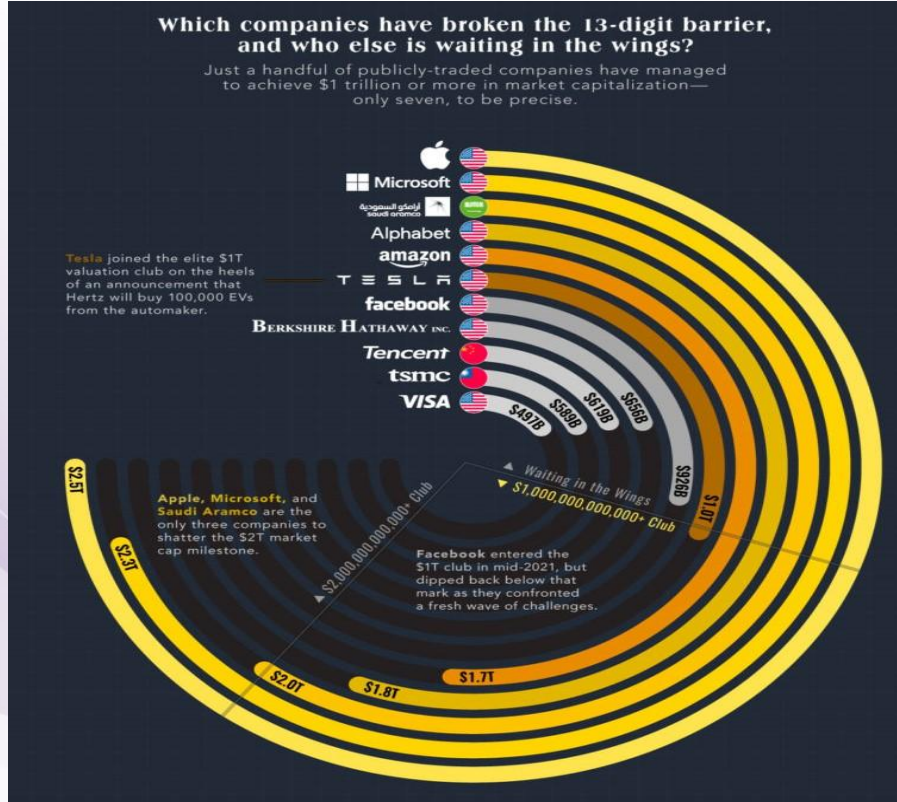
Source: Bloomberg Billionaires Index



statista

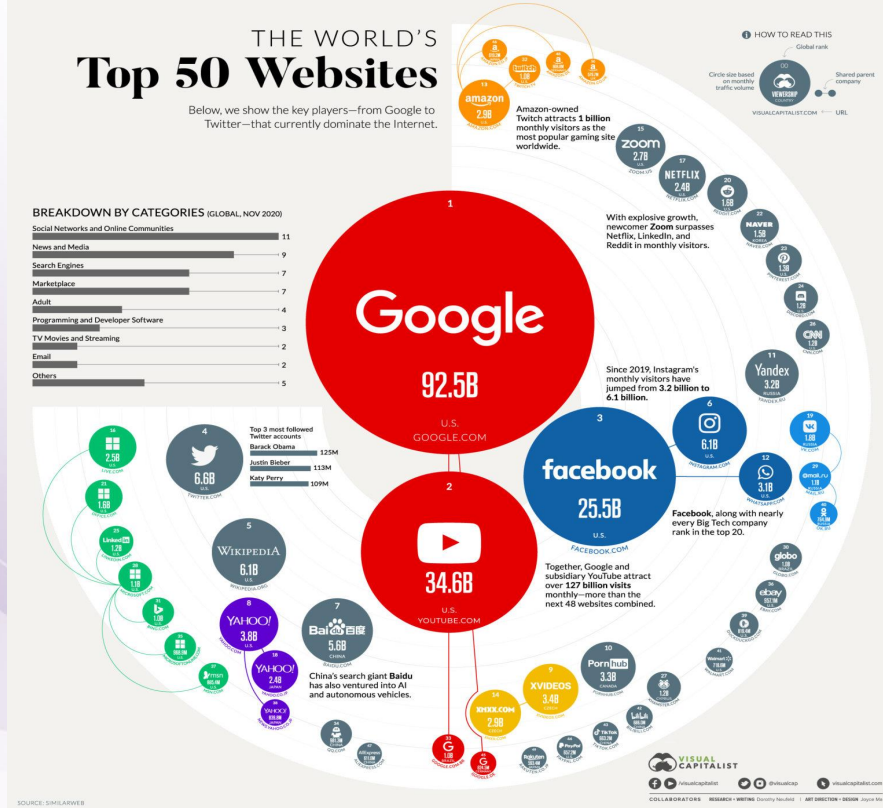
Serveti 100 Milyar
Dolari Aşan Zenginler
Kulübü'nde yer alan
isimlerden 7 tanesinin
yatırımları teknoloji
sektöründedir.

NEDEN SDLC ?



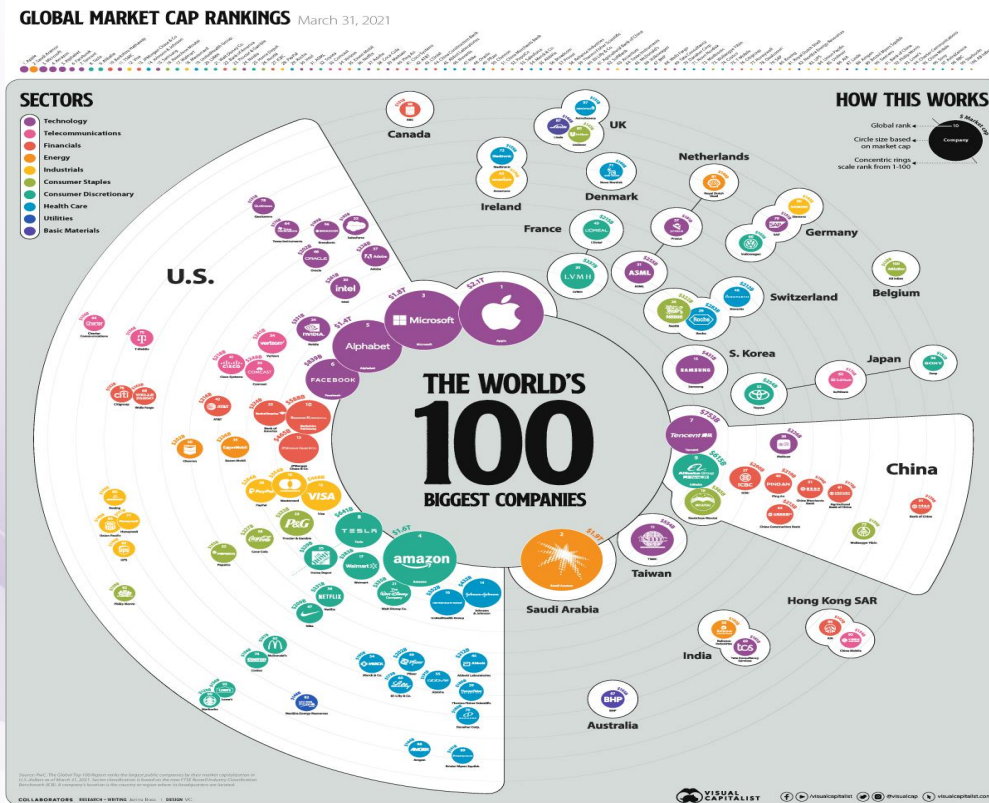
Bir otomobil markası olan **TESLA** teknoloji ve yazılıma yaptığı yatırımlarla sektöründeki rakiplerinden ayrılmış ve **Piyasa Değeri Trilyon Doları Geçen Şirketler Kulübü'ne** girmeyi başaran tek otomobil şirketi olmuştur.

NEDEN SDLC ?



Dünyanın En Çok Ziyaretçi Alan Web Siteleri

NEDEN SDLC ?



Dünyanın En
Büyük
Şirketleri

SOFTWARE DEVELOPMENT PHASES

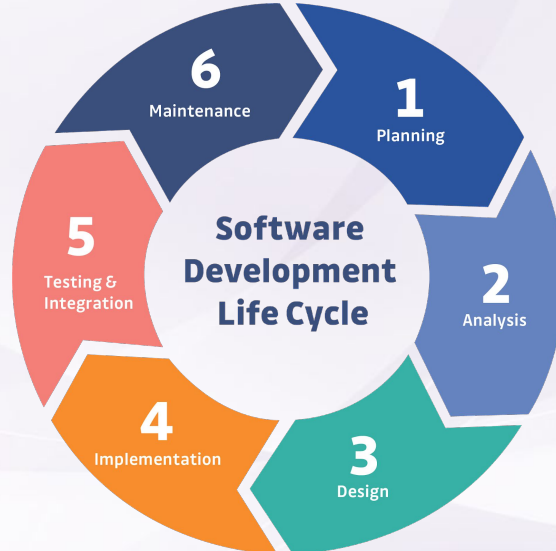
(yazılım geliştirme aşamaları)

- PLANLAMA
- ANALİZ
- TASARIM
- GELİSTİRME - KODLAMA
- TEST
- BAKIM - GELİSTİRME

SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)

SDLC mümkün olan en kısa sürede, en yüksek kalitede ve en düşük maliyetle yazılım üretmeyi hedefleyen bir süreçtir. SDLC, bir organizasyonun iyi test edilmiş ve üretim aşamasında, kullanıma hazır yüksek kaliteli yazılımı hızla üretmesine yardımcı olan iyi yapılandırılmış bir aşama akışı sağlar.



SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)

1	Planning	Planlama
2	Analysis	Analiz
3	Design	Tasarım
4	Implementation	Kodlama
5	Testing	Test
6	Maintenance	Teslim ve Bakım

SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)



Planlama (Planning)

“Ne istiyoruz?” SDLC’ nin bu aşamasında ekip, analiz edilen gereksinimlerin uygulanması için gereken maliyeti ve kaynakları belirler. Ayrıca ilgili riskleri detaylandırır ve bu riskleri azaltmak için alt planlar sunar.

Başka bir deyişle, ekip projenin fizibilitesini ve projeyi en düşük riski göz önünde bulundurarak nasıl başarılı bir şekilde uygulayabileceklerini belirlemelidir.

SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)

Analiz (Analysis)

Buradaki temel amaç yazılım mühendisi açısından mevcut plandaki çözümün incelenip müşteri taleplerinin doğru bir şekilde anlaşılıp anlaşılmadığının ortaya çıkarılmasıdır. Bu aşamada gereksinimler detaylı bir şekilde incelenir ve projede nelerin istenildiği ile ilgili analiz çalışmaları yapılır. Daha ayrıntılı bir şekilde yapılan problem tanımlama aşamasıdır da diyebiliriz.

Her şey iki tarafça açıkça anlaşılıp kesinleştikten sonra dokümantasyon yapılır.



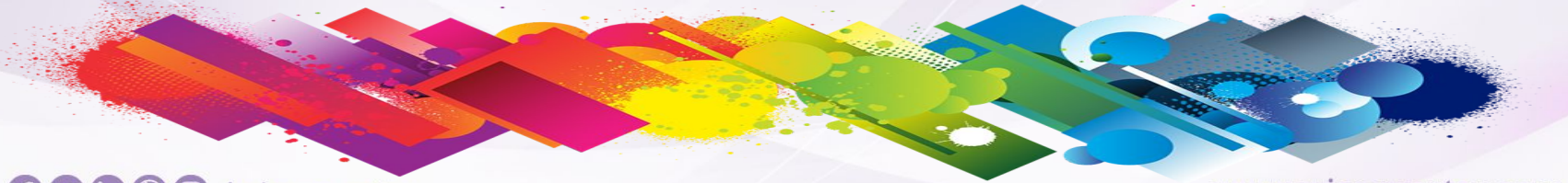
SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)

Tasarım (Design)

Bu aşamada gereksinimlerin analiz edilmesiyle yazılım sistemi tasarlanır. Tasarım aşamasında kodlama yapmak söz konusu değildir. Analiz aşamasında problemin ne olduğu belirlenirken tasarım aşamasında problemin nasıl çözüleceği belirlenir. Yazılım ürününün gereksinimleri karşılayan özellikleri, faydaları ve yetenekleri belirlenir.

Mimari tasarımda yazılım ürününün genel bir planı yapılır ve modüller belirlenir. Ayrıntılı tasarımda yazılımda kullanılacak algoritmalar, programlama dilleri ,veritabanları ve bunun gibi detaylar belirlenir.



SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)

Kodlama (Implementation)



Kodlama sürecinin başladığı aşamadır. Bu aşamada doğru kodlama yapılması önemlidir. Doğru kodlama şekli bir başkasının da rahatça okuyabileceği ve bakım yapabileceği kodlamadır. Bu kodlama biçimi temiz kodlama (clean code) olarak adlandırılır.

SOFTWARE DEVELOPMENT PHASES

(yazılım geliştirme aşamaları)

Test (Testing)

Kodlama yapılırken ve kodlama sonrasında birçok test yapılır. Bu testlerden bazıları ; birim testleri, zorlanım-performans testi , yanlış değer testleri, tümleyim testi, kullanım senaryo testleri, yük testleri, kullanıcı kabul testi, test otomasyonu gibi testlerdir. Ayrıca analiz aşamasından itibaren testlerin yapılması yazılım ürününde hata oranını azaltacağı için kaliteyi artırır, maliyetleri(para, zaman vb) azaltır. Bu yaklaşım erken test yaklaşımı (early testing) olarak adlandırılır.

