

AGILE MANİFESTO(CEVİK MANİFESTO)

Çevik Süreç, Metodoloji ve Manifesto Süreçleri

12126035 Özge Dilhan Atilla --- 12126039 Meltem Gökgöz----12126029 Onur Gündüz

07.05.2015

Agile Manifesto (Çevik Manifesto)

Öncelikle manifesto kelimesini açıklamak istiyorum. Günlük hayatımızda kullandığımız bir kelime olmadığından ilk okuduğunuzda çoğunuzun da benim gibi anlamadığı bir kelime.

Manifesto vikipedi'nin tanımına göre ; “**Toplumsal bir hareketin duyurulması ve savların belirtilmesi üzerine kurulan, bir akımın, bir hareketin oluşunu bildiren yazılara manifesto ya da bildiri denmektedir.**”

Çevik Sürecin Geçmişi

2000 senesinde Kent Beck ve arkadaşlarının yer aldığı bir toplantı düzenlendi. Kent Beck ve arkadaşları belli bir süredir çevik yazılım metotlarını projelerde uyguluyorlardı ve fikir alış verişi için böyle bir toplantının faydalı olacağını düşündüler.

Smaltalk gibi nesneye yönelik modern programlama dilleri yanı sıra iteratif (tekrarlamalı) yazılım metotları geliştirilmiş ve değişik ekipler tarafından uygulanmaktaydı. Smalktalk ile başlayan bu yeni akım doksanlı yılların ortalarına doğru iyice kuvvetlenmişti. Çevik (Agile) kelimesi ilk kez bu toplantıda değişik iteratif yazılım metotlarını bir çatı altında toplamak için kullanıldı.

Toplantıya katılanlar tarafından bir manifesto (Agile Manifesto) hazırlandı. Yeni bir yazılım filozofisi doğmuştu. Çevik sürecin mimarları olan katılımcılar kısa bir zaman sonra Agile Alliance4 organizasyonunu kurdular. Bu organizasyon ile çevik süreç ve çevik yazılımın desteklenmesi, geliştirilmesi ve uygulanması hedef alındı. Konuyla ilgilenen kurum, kuruluş ve şahıslar için bu organizasyon merkezi bir buluşma noktasıdır. Agile Alliance organizasyonu ilgilenenlere çevik süreç ve çevik yazılım hakkında geniş bir literatür

sunmakta ve her yıl bu konuda konferans düzenleyerek, katılımcıları bilgiledirmektedir.

² Bakınız: http://en.wikipedia.org/wiki/Waterfall_model

³ Bakınız: http://en.wikipedia.org/wiki/V-Model_%28software_development%29

⁴ Bakınız: <http://www.agilealliance.org>

Yani;

2001 yılında yazılan ve o günden itibaren sürekli geliştirilen , bir çok programcının altına imzasını koyduğu , çevik programlama yönteminin belirli kuralla ve ne amaca hizmet ettiğini anlatır. Agile Manifesto'nun kendi sitesinden bu belgeye ulaşabilir ve oradan da inceleyebilirsiniz. Ama ben sizin için çevirilmiş halini paylaşayım.

Çeviklik Manifestosu

Daha iyi yazılım geliştirmenin yöntemlerini, uygulayarak ve başkalarının uygulamasına yardım ederek, su yüzüne çıkarıyoruz.

Bu çalışmayla değerlerimiz :

Bireyler ve etkileşimler, süreçler ve araçlardan
Çalışan yazılım, kapsamlı dökümantasyondan
Müşteri ile işbirliği, kontrat görüşmesinden
Değişikliklere yanıt vermek, bir planı takip etmekten önce gelir.

Her ne kadar sağda yazılı olanlar da değerli olsa da Biz soldakilere daha fazla değer veriyoruz.

Manifesto yayımlandıktan sonra tabi ki bir çok programcı ve programlama üstüne kurulan şirketler çevik süreç , çevik metodoloji , çevik manifesto gibi kavramları çalışma hayatlarına soktular. Hatta bir çok blog sahibi “Yazılım sektörü yıllardan beri kan kaybediyor. Ama artık taze kan bulundu ve hastalığın tedavisi kolaylaştı.”(Özcan Acar-KurumsalJava) tarz yorumlarla çevik manifestoyu övdü.

Buraya kadar özetlersek çevik metodoloji ve manifesto 2001 yılında devrim niteliği taşımaktadır. Metodoloji ve manifesto yayınlanmadan önce yapılan projelerdeki başarısızlık oranı yayınlandıktan sonra düşmüştür.

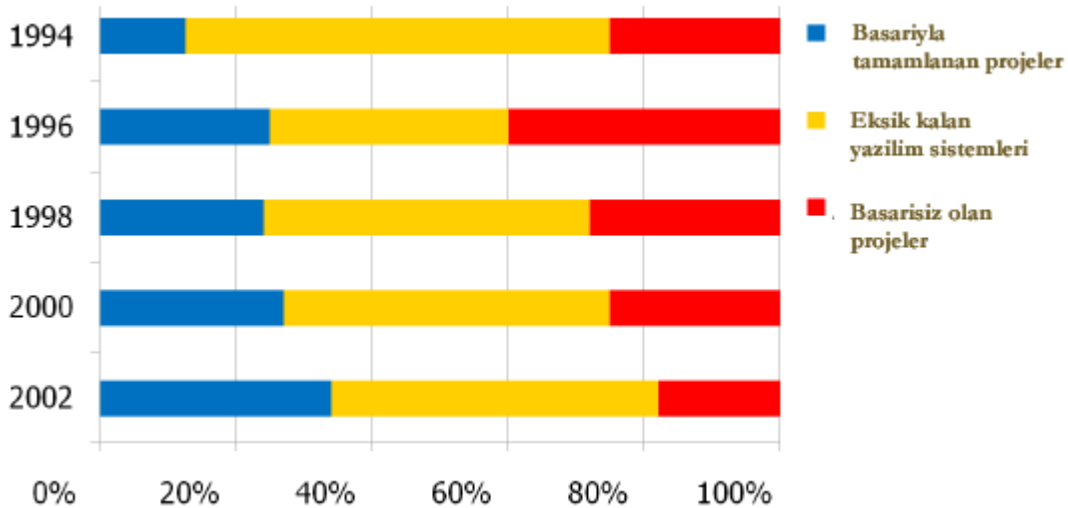
Başarısızlığın sebebi ise;

Hedefin net bir şekilde tarifinin yapılabilmesine rağmen, günümüzde uygulanan birçok projenin başarıyla tamamlanamadığını ya da oluşturulan yazılım sistemlerinin müşteri gereksinimlerini tatmin etmediğini görmekteyiz.

Peki ya bunun sebebi?

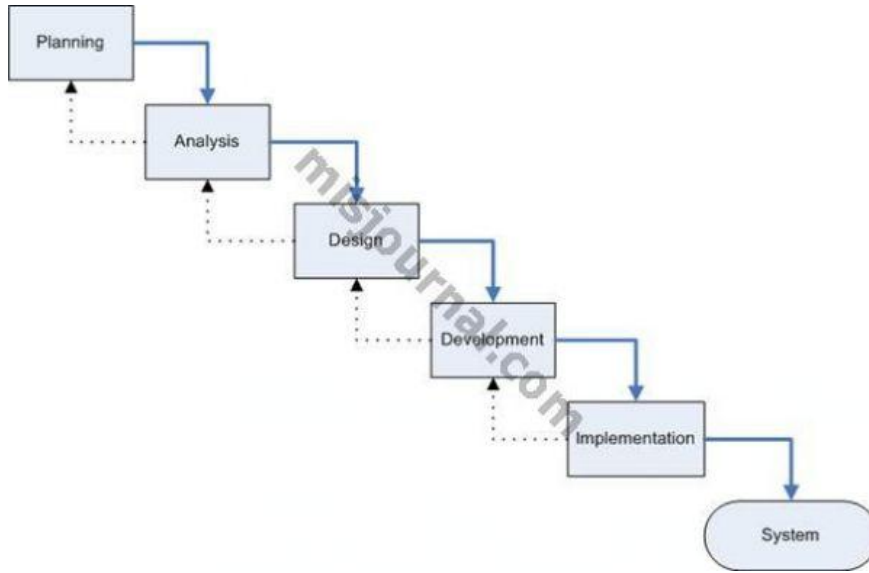
Çoğunlukla müşterinin ne istediğini bilmeden size gelmesidir. Ya da müşterinin kafasında çizdiği resmi size aktaramaması –ki kimse bu konuda başarılı değildir- , sonradan yaptığı ani karar değişiklikleri projeye hem zaman hem de para kaybettirir. Yapılan projeler müşteri odaklı değil proje bitirme odaklı olduğu içinde başarısızlık oranları gittikçe artmıştır.

İşte manifestodan önce ve sonra ki oranlar;

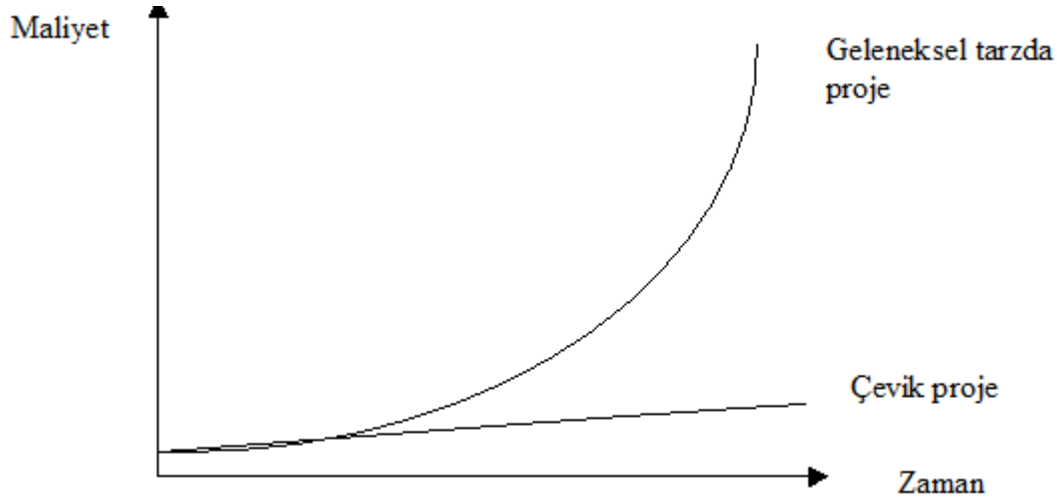


Çevik metodolojiden önce şirketler ve yazılımcılar genel olarak şelale modelini kullanırlardı. Şelale modelini özetlersek müşteri ile sadece ilk proje alımı toplantısında diyaloga girer , istediklerini alır ve diğer aşamalarda müşteri ile diyaloga girmez. Bir sonraki safhaya geçebilmek için bulunduğunuz safhanın tamamen hatasız olması gerekir ve her basamağın sonunda doküman hazırlanır. Doküman tabanlıdır.

Bahsettiğim Şelale Modeli;



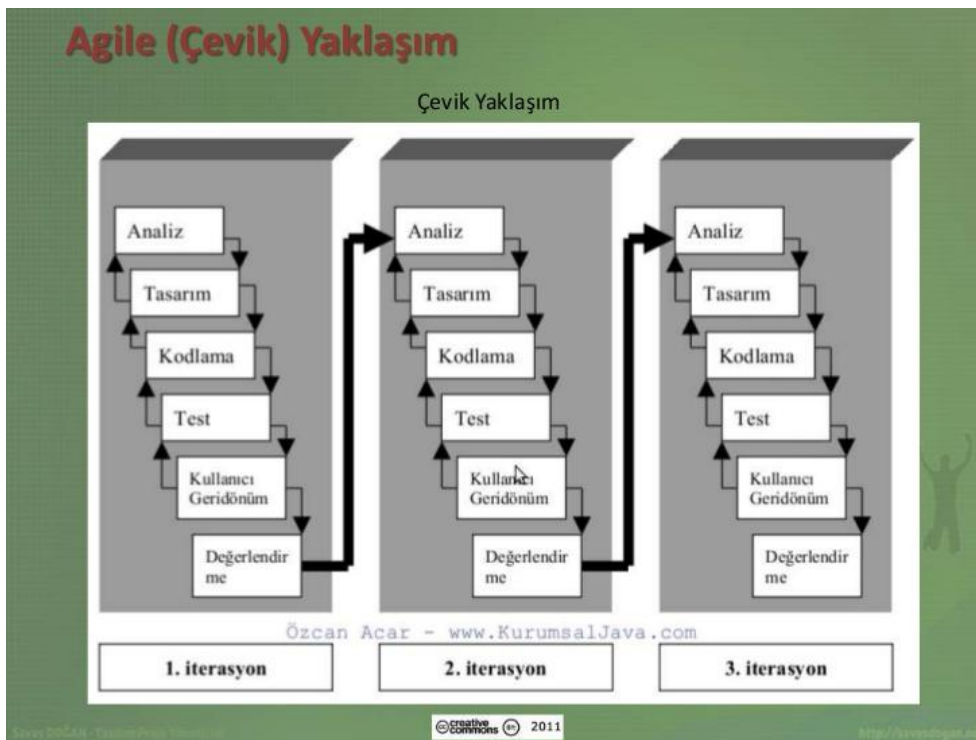
Sırasıyla plan,analiz,tasarım,geliştirme,test,sürüm ve bakım basamakları vardır.



Resim 1.3 Geleneksel tarzda yapılan projelerde gerekli değişikliklerin yapılması maliyete zamanla yükseltir.

Çevik süreçlerde durum farklıdır. Çevik süreç değişimi kabul eder ve onunla yaşamayı kolaylaştırmak için yeni yazılım metotları sunar.

Çevik süreçlerde iterasyon bazında çalışmalar sürdürülür. Her iterasyon bir ile dört haftalık zaman dilimlerinden oluşur ve şelale modelinde yer alan safhaları ihtiva eder. Aslında her iterasyona bir mini şelale modeli ihtiva ediyor diyebiliriz.



Agile Proje Yönetiminin geçmişinden daha önce bahsetmiştik. Bu süreç içerisinde Agile geliştirmenin kolları oluşmaya başladı, 1995 yılında SCRUM ortaya çıktı, bununla beraber Adaptive Software Development, Feature Driven Development, ve Dynamic Systems Development Method (DSDM) oluştu. Hemen akabinde ise Crystal Clear, Extreme Programming (XP) (1996) ortaya çıktı. Tipik Agile Metodolojileri olan bu yöntemler sonrasında 2001 yılında dünyanın en iyi yazılım geliştiricileri olarak bilinen 17 bağımsız yazılım geliştirici bir araya gelerek Çevik Yazılım Geliştirme Manifestosu yayınladı. Bununla beraber bu oluşumu desteklemek ve geliştirmek için kar amacı güdülmeleyen bir organizasyon kurdular; Agile Alliance...

Peki bu Agile Metodolojiyi hangi firmalar kullanıyor dersek size bir çırpıda Microsoft, Google, Yahoo, oyun devi Electronic Arts, Philips, Siemens, BBC gibi kuruluşların isimlerini verebilirim...

Avantajları ve Dezavantajları üzerinde durursak;

Avantajları:

- Bir kere ekip ruhu kazandırır
- Planlama ve yürütme bir arada olduğu için sürdürülebilir kalite sağlar
- Ayrıntılı plan süreçleri yerine iteratif planlar (tekrar eden süreçler) uygulanır
- Müşteri ihtiyaçlarına uygunluk önemlidir, sık sık ürün çıktısı verilir ve müşteriden gelen geri bildirimlerle ihtiyaca uygun, doğru ürün çıkarılır.

Dezavantajları:

- Kurumsal bir yapıda uygulamak için ciddi anlamda zorlanırsınız
- Hedefler kısa vadeli olduğu için ekip üzerinde sonuç baskısı oluşabilir

- Kısa süreli ürün teslimi sonucunda gelen geri bildirimlerle çalışma sürelerinin artması muhtemeldir
- Dokümantasyon konusundaki bilinen birçok şeyin, tabiri caizse, “gereksiz” kılınması durumu vardır.

Önemli çevik süreç türlerini şunlardır:

Scrum: Scrum seksenli yıllarda Kent Schwaber ve Jeff Sutherland tarafından geliştirilmiş bir çevik süreçtir. Scrum Rugby oyununda kullanılan bir terimdir. Oyuncular kısa bir süre için bir araya gelerek, bir sonraki oyun hamlesi hakkında fikir alış verişinde bulunurlar, yani kısa bir toplantı yaparlar. Scrum daha çok proje yönetim metotlarına konsantre olmaktadır. Yazılımın nasıl yapılması gerektiği hakkında detay ihtiva etmez. Birçok projede Scrum Extreme Programming (XP) ile kombine edilir.

XP: Extreme Programming (XP) doksanlı yılların sonunda Kent Beck, Ron Jeffries ve Ward Cunningham tarafından Chrysler için yapılan bir proje sonrasında oluşmuş bir çevik süreçtir. XP Scrum’dan esinlenilerek geliştirilmiş bir çevik süreçtir. XP Scrum’ın aksine daha çok yazılım metotlarına konsantre olmaktadır. Bu sebepten dolayı Scrum ve XP bir projede kombine edilebilir.

IXP: XP’den doğan IXP’nin (Industrial XP⁵) amacı XP yi geliştirmek ve XP’de yeralan metot ve tekniklerin daha büyük organizasyonlar için adapte etmektir.

FDD: Jeff DeLuca tarafından doksanlı yılların sonunda geliştirilmiş bir çevik süreç türüdür (FDD = Feature Driven Development⁶). FDD, yazılım özelliği (feature,function) güdümlü çalışır. Sisteme yeni bir özellik kazandırılmadan önce, detaylı bir tasarım çalışması yapılarak bu özelliği kapsayan mimarî yapı oluşturulur. Bu yüzden FDD daha çok tasarım odaklı işleyen bir çevik süreçtir.

⁵ Bakınız: <http://industrialxp.org>

⁶ Bakınız: <http://www.featuredrivendevelopment.com>

Çevik manifestosu araştırmamız bu kadardı. Okuduğunuz için teşekkür ederiz. ☺

Kaynaklar;

- <http://e-bergi.com/y/Cevik-Modelleme-ve-Cevik-Yazilim-Gelistirme>
- <https://eksisozluk.com/agile-manifesto--1377041>
- <http://www.canbu.info/blog/?p=188>
- <http://agilemanifesto.org/iso/tr/>
- <http://tr.wikipedia.org/wiki/Manifesto>
- <http://barbarosgurcan.com/post/Agile-Software-Development-Cevik-Yaz%C4%B1%C4%B1m-Gelistirme.aspx>
- Özcan Acar - www.KurumsalJava.com

