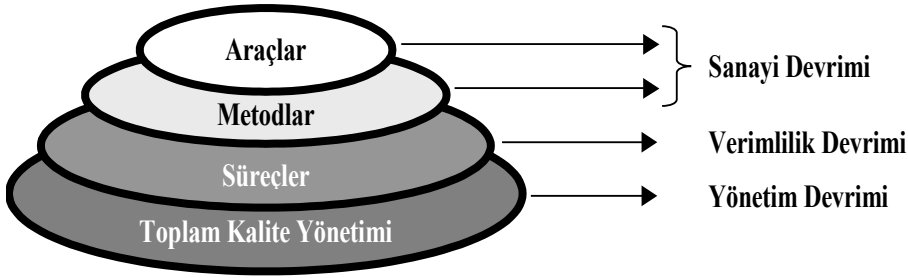


CMM ve XP (Extreme Programming)

Orhan Kalaycı
Nitelik Yazılım Danışmanlık Ltd.
www.nitelik.net

Özet. Bir yanda 1990 ların başından itibaren başarısını özellikle büyük askeri yazılım projelerinde kanıtlamış, yazılım olgunluk modeli: CMM. Diğer yanda, son yıllarda adından sıkça söz edilen özellikle küçük yazılım projeleri için geliştirilmiş çevik yazılım geliştirme metodolojisi: XP (Extreme Programming) Bu yazıda literatürde çeşitli defalar incelenen CMM ve XP ilişkisi referanslarına dayanarak irdelenecektir.

1 CMM



Şekil 1. Sanayi, verimlilik ve yönetim devrimlerinin Sistem Mühendisliğine getirdiği yeni bakış açıları

Tarihsel bir bakış açısı ile bakıldığında, bilgi çağına şirketlerinde olması gereken temel bileşenler: **Araçlar**, **metodlar**, ve bunların insanlar tarafından nasıl (hangi sıra ile) kullanılacağını tarif eden **süreçler** ve son olarak süreçlerin tanımlanması ve devamlı iyileştirilmesi işi olan **Toplam Kalite Yönetimi** bileşenlerinin [12] Türk yazılım şirketlerine yerleştirilmesi Türkiye için kaçınılmaz bir gerekliliktir.

Başlangıçta, Amerikan Savunma Bakanlığının (DoD – Department of Defence) bir ihtiyacına cevap olarak 1987’de ilk sürümü yayınlanan CMM, zaman içinde askeri olmayan şirketler arasında da geniş kabul görmüştür.

CMM’nin sahibi SEI – Software Engineering Enstitute tarafından en son, Ağustos 2002 de yayınlanan raporda CMM değerlendirmesine giren şirketlerin %70’e yakın bir kısmının askeri olmayan ticari şirketler olduğu görülüyor. [14]

CMM’in iki kullanılışı: [2]

1. Bir yazılım ihalesi açan kuruluşların ihaleye katılan firmaların yeterliliklerini değerlendirebilmeleri.
2. Yazılım firmaları kendi verimliliklerini ve performanslarını uluslararası rekabet karşısında arttırabilmeleri.

Ağustos 2002, SEI istatistiklerine göre 1992’den günümüze SEI CMM değerlendirmesine katılan 1094 firmanın %48’i 100 ve daha az yazılım mühendisi çalıştıran kuruluşlardan oluşmaktadır. [14]

CMM modeli kullanılarak bir şirkete 5 üzerinden bir not verilir. 5 en iyi, 1 de en kötü nottur. Türkiye’de CMM 3. seviye olduğunu uluslararası geçerli olarak belgeleyen iki şirket vardır: Milsoft ve Aydın Yazılım. Bu, güzel bir gelişme olmakla birlikte, Türkiye’nin önünde alması gereken çok yol olduğunu gösteren bir veridir. Çünkü, dünyada resmi olarak CMM 5. seviyesinde olduğu belgelenmiş 74 firmanın yarısından fazlası Hindistan’dadır. [15]

Tablo 1. CMM 4. ve 5. Seviyedeki şirketlerin ülkelere göre dağılımı [15]

Ülkeler	Level 4	Level 5
Singapur	1	
Kanada		1
Fransa	1	

İsrail	1	
İrlanda	1	
Avustralya	2	
Hindistan	27	50
Çin		2
Rusya		1
Amerika	39	20
Dünya	72	74

1.1 CMM L3'e ne kadar sürede ulaşılır?

Dünya ortalamalarında, CMM L1 den L2 ye ulaşmak 24 ay, L2'den L3'e 22 ay sürmektedir.

Araştırmaya katılan bütün şirketleri düşünürsek bu sürelerin ortama 300 eleman çalıştıran bir yazılım şirketi için doğru olduğu varsayılabilir. [14]

Süreler eleman sayısı ile doğrudan ilintilidir. Süreçler ancak onları bilen ve uygulayan kişiler olduğu müddetce var kabul edilir. Her bir eleman için teker teker vakit harcamak gereklidir. Koçluk ve liderlik çok önemlidir.

1.2 CMM Süreçleri

CMM, her bir seviyeyi kurumsallaştırılması gereken süreçler olarak tarif eder. 2. Seviye proje yönetimi süreçleri, 3. seviye kurumsal süreç tanımları, 4. seviye sayısal ölçümler ve süreç yönetiminde sayısal yaklaşımlar, ve 5. seviye sürekli iyileştirme amacıyla tanımlanmıştır. Bu süreçler aşağıda görüldüğü gibidir. [2]

Tablo 2. CMM Modelini oluşturan süreçlerin seviyelere göre dağılımı [2]

<p>1. Seviye – Başlangıç - Bireysel Kahramanlar</p> <p>2. Seviye – Tekrarlanabilir – Proje Yönetimi Süreçleri</p> <p>I. Müşteri İsterler Yönetimi</p> <p>II. Yazılım Proje Planlanma</p> <p>III. Yazılım Proje Kontrol</p> <p>IV. Yazılım Altyüklenici Yönetimi</p> <p>V. Yazılım Kalite Güvence</p> <p>VI. Yazılım Konfigurasyon Yönetimi</p> <p>3. Seviye – Tanımlı – Mühendislik Süreçleri ve Kurumsal Destek</p> <p>I. Kurumsal Süreç Odaklanması</p> <p>II. Kurumsal Süreç Tanımları</p> <p>III. Kurumsal Eğitim Programı</p> <p>IV. Bütünleşik Yazılım Yönetimi</p> <p>V. Yazılım Ürün Mühendisliği</p> <p>VI. Gruplararası Koordinasyon</p> <p>VII. Karşılıklı Gözden Geçirme</p>	<p>4. Seviye – Sayılarla Yönetilebilen – Ürün ve Süreç Kalitesi</p> <p>I. Kurumsal Süreç Performansı</p> <p>II. İstatistiksel Süreç Yönetimi</p> <p>5. Seviye – En İyilenebilen (Optimum) – Sürekli Süreç İyileştirme</p> <p>I. Hata Önleme</p> <p>II. Kurumsal Süreç Geliştirme (Innovation)</p> <p>III. Kurumsal İyileştirmenin Yerleştirilmesi</p>
---	---

Süreç kontrol, ölçülebilirlik (quantitative) ve süreç iyileştirme arasındaki ilişki şeklinde görüldüğü gibi analiz edilebilir. Şirkette öncelikle sayısal olmasa da süreçler kontrol altına alınıyor (2. ve 3. seviye), kontrol altına alınan süreçler için sayısal veriler toplanıyor (4. seviye) ve böylece nihai hedef olan sürekli süreç iyileştirme (5. seviye) başlıyor:

Aşağıda görüldüğü gibi CMM'de üst seviyelere çıktıkça proje maliyetleri azalmakta, projenin parametrelerinin tahmin edilebilirliği artmaktadır.

2 XP – Extreme Programming

XP – Extreme Programming, son yıllarda hızla yayılan çevik (agile) yöntemlerden bir tanesidir. Kişisel görüşüm, yazılım sektörüne özel bir üretim biçimi olan XP önümüzdeki yıllarda yazılım geliştiriciler arasında tamamen yaygınlaşacak ve Nesne Tabanlı sistemler gibi temel bir yaklaşım olarak kabul görecektir.

Yüksek teknolojinin bir ürünü olan XP, geçmiş yıllarda müşteri ile üretim takımları arasındaki mesefanın azalması ve hızlı ve düşük maliyetli geliştirme ve test ortamlarının yaygınlaşması ile mümkün olabilmiştir.

Benzer bir durumu, bir ormana kamp yapmaya giden bir grup izci için düşünebiliriz. Eski teknoloji ile kamp programını en baştan çok ayrıntılı şekilde planlamaları gerekirken artık temel hedeflerde anlaştıktan sonra ayrıntılı zaman içinde konuşmak üzere erteleyebiliyorlar, çünkü kamp ortamında, ormanda örneğin, her an iletişimlerini sağlayabilecek cep telefonları ve ceplerinde her an nerede olduklarını görebilecekleri GPS sistemleri var. Öyleyse, kamp programını en baştan aşırı detaylı tanımlamaya çalışmak hem gereksiz enerji kaybı olacaktır, çünkü ormanın durumuna göre başlangıçtaki hedefler her an değişebilecektir. Yeni yeni görmek istenecek yerler keşfedilebilir, vb.

Yazılım geliştirmede de, yazılım macerasının en baştan çok detaylı tanımlanmaya çalışılması hem çok enerji gerektirecek hem de en başta düşünülen hedefler zaman içinde büyük ihtimalle değişecektir. O yüzden mümkün olduğunca az planlama ile en temel ihtiyaçlardan başlayarak devamlı çalışır bir sistem üzerinde yeni ihtiyaçların eklenmesi son derece sağlıklı ve ekonomik bir yaklaşım olacaktır ki, XP'nin ana fikri de budur.

Müşterinin isteklerini küçük parçalara, *hikayelere*, ayırarak en önemli isteğini, *hikayesini*, çalışır duruma getirip sistemin çalışırlığını bozmadan sürekli test ve müşteri kontrolünde ilerlemek sağ duyuya çok uygun basit ama çok daha başarılı bir yaklaşım ortaya çıkartmıştır.

XP en baştan değişiklikleri, sürecin doğal bir bileşeni kabul ettiğinden müşteriden ya da geliştirme ekibinden gelecek olan değişiklik istekleri süreci sekteye uğratmaz, normal kabul edilir. Klasik hayat döngülerinde değişiklik istekleri en başa dönüp herşeyi yeni baştan ele almayı gerektirebilirken XP de böyle bir geriye dönüş yaşanmaz çünkü her zaman değişiklik olacağı en baştan var sayılmış ve bütün süreç ona göre tasarlanmıştır. [16, 17]

2.1 XP'nin dört temel taşı

İletişim.

Yüz yüze iletişim taviz verilemez bir gerekliliktir. Yüz yüze iletişimi engelleyecek her türlü kısıt kesinlikle ortadan kaldırılmalıdır.

Basitlik.

Basitlik, yapılmaması gereken işlerin maximum düzeye çıkarılması sanatıdır. Yapılmaması gerekli olmayan bir şey kesinlikle yapılmamalıdır.

Geri Besleme.

Sık sık projelerde yazılım ekibi ile müşteri önemli bir sorunları olduğunu ürün teslim tarihinden çok kısa bir süre önce keşfederler. XP de her hafta müşteriye çalışan bir ürün teslim edilmesi hedeflendiğinde ürün üzerinden müşteri ile yazılım ekibinin ortak bir zeminde buluşmaları en baştan kolaylaştırılır.

Cesaret.

Yazılım projeleri genellikle başarısız olur. Başarısız olmak kaçınılması gereken bir durum değildir. Doğaldır. Başarısızlıktan korkan ekipler yavaş hareket ederler, genellikle başarısızlığı ortadan kaldırmaz sadece geciktirirler. Başarısızlığa mümkün olduğunca erken uğramak daha sonra telafi etme şansını arttırır. XP, başarısızlıktan korkmayı değil en kısa zamanda başarısız olmayı ve onu en kısa zamanda telafi etmeyi önerir.

XP'nin 12 Uygulaması.

Her bir XP uygulaması basit kuralları tarif ederler, ancak bir arada kullanıldıklarında birbirlerinin eksiklerini telafi ederler ve başarılı sonuçlara ulaşırlar. Eski VISA Uluslararası Genel Müdürü, Dee Hock'un dediği gibi, "Basit, açık ve net hedefler karmaşık davranışları gerektirir. Karmaşık kural ve prensipler ise basit ve aptal davranışları doğurur." XP'nin kuralları ve uygulamaları basittir ama bu kuralları başarı ile uygulamak ileri yetenekli insanları ve akıllı davranışları gerektirmektedir. [17]

Kodlama Standartları.

Kodlama bir takım çalışmasıdır. Zaman içinde değişik kod parçaları üzerinde değişik kişiler çalışacaktır. Takım çalışmasını verimli ve etkin kılmak için bütün kodlar sanki bir kişi tarafından yazılmış gibi okunabilmesini sağlamak için "kodlama standartlarına" ihtiyaç vardır.

Basit Tasarım.

Değişimin maliyetini en azda tutmanın yolu sistemin tasarımını olabildiğince basit yapmakla doğrudan ilişkilidir. Ayrıca karmaşık tasarımlar, ileride değişme ihtimali yüksek olan özellikler için boşuna emek harcama riskini taşımaktadır. XP ileride değişiklik maliyetini en az seviyede tutmak adına en basit tasarımı yapmayı amaçlar.

Test.

Bütün istekler kabul testlerinde tarif edilmelidir. Kabul testleri müşteri tarafından yapılmalıdır. Programcılar, kodlarını yazmadan önce test senaryolarını yazarlar. Neyi kodlayacaklarını test senaryolarında tarif ederler (Önce test et, sonra tasarla) Bütün Test senaryoları otomatik olarak koşabilmelidir. Böylece, programcılar istedikleri kadar çılgınca kod yazabilirler, çünkü testler hataları yakalayacaktır.

Çift Programcı.

Programcılar, aynı makineyi paylaşan ikerlik takımlar olarak organize olurlar. Sağ duyuya aykırı olmasına rağmen bu iki programcının paralel olarak kod yazmasından daha verimlidir. Bilgi paylaşımı ve takım ruhu artar. Kod gözden geçirme, böylece, devamlı olarak icra edilir.

Ortak Kod Mülkiyeti.

Kodlama standartlarına uyması, ortağının (çiftli programlama) itiraz etmemesi, ve değişiklik sonrası testlerin başarı ile sistem üzerinde koşması kaydı ile, Takımdaki herkesin, her hangi bir kod parçası üzerinde istediği değişikliği yapmaya yetkisi vardır.

Yeniden Tasarım.

Yeniden tasarım, programa yeni bir özellik eklenmemesine rağmen tasarımın ve kodun değiştirilmesi işlemidir. Yeniden tasarım, yapılabilir bir şeydir çünkü eğer yanlış bir şey yapılırsa otomatik testler bunu yakalayacaktır. Yeniden tasarım, üzerinde uğraşılan sorun hakkında daha iyi bir anlayış geliştirildiğinin göstergesidir.

Devamlı Bütünleştirme.

XP takımları küçük adımlar üzerinde çalışır ve gün içinde bir çok defalar bütünleştirme yaparlar. Böylece bütünleştirme sorunları çok çabuk fark edilir ve çözümler üretilir. Devamlı bütünleştirme, takımların uzun süreler birbirinden bağımsız ve uyumsuz kodlar üzerinde çalışmasını engeller ve herkesin en son sürüm üzerinde çalışmasını garantiler.

Küçük Sürümler.

Sürümler, en kısa zamanda geliştirilebilecek kadar küçük, müşteriye anlamlı bir fonksiyon sağlayacak kadar anlamlı olmalıdır. Devamlı bütünleştirme ve sürekli test ortamında çalışan XP takımları, bazen birkaç haftada bir sürüm çıkardığı gibi her gün bir sürüm çıkaran takımlar da olabilmektedir.

Takım Üyesi Müşteri.

Yazılı istekler genellikle eksik ya da muğlak olabilmektedir. İlk istek metinleri ne kadar detaylı yazılmış olsada programcılar genellikle müşteriye danışmak ve konuşmak zorunda kalmaktadır. XP takımlarında devamlı bir müşteri bulundurulur ve soruna kalıcı bir çözüm yaratılmaktadır. Programcılar her an müşteriye ulaşabilmekte ve birlikte çalışabilmektedir.

Oyunun Planlanması.

Hikaye havuzundan, müşteri, iş ihtiyaçları, programcılar da teknik kısıtlar açısından bakarak her bir sürümde hangi fonksiyonların, hikaye, yer alacağına karar verirler. Hem müşteri hem programcılar, sürümde hangi fonksiyonların, hikayelerin, olacağını çok iyi bilirler ve olası sürprizler en baştan engellenmiş olur.

Hafta 40 saat.

Yazılım üretmek yaratıcı bir faaliyet olduğundan, programcılar haftada 40 saatten fazla çalışması verimliliği ve etkinliği olumsuz yönde etkilemekte, işten ayrılmaları arttırmakta ve morali azaltmaktadır. Haftada 40 saatten fazla çalışmamak uzun vadede daha etkin ve verimli sonuçlar doğurmaktadır.

Metafor.

Üzerinde çalışılan sistem hakkında bir benzetim (analoji) hikayesi geliştirilmesi takım üyeleri arasında ortak bir dil oluşturacak ve iletişimi geliştirecektir.

3 CMM ve XP (Extreme Programming)

CMM, 1980'lerden bugüne, en iyi örneklerden yola çıkarak, bir yazılım organizasyonunda hangi süreç özelliklerinin bulunması gerektiğini tarif eder. Kişisel olarak CMM konusunda bilgi verdiğim bir çok yazılım firması bana "CMM

bize eksikliklerimizi (sorunlarımızı) çok güzel gösteriyor ama çözüm önerilerini nereden bulacağız?” diye soruyorlar. Gerçekten de, CMM esasında yazılım firmalarına işinizi şöyle yapmanız gerekir diye bir yöntem önermez. Kurulmuşta var olan, yöntemi inceler ve yöntemin eksiklikleri hakkında bir rapor hazırlanmasına yardımcı olur. Bu anlamda XP ile CMM ayrı kategorilerde birer modeldir. CMM yazılım firmalarının eksikliklerini ya da başarılarını keşfetmek üzere kullanılırken, XP yazılım firmalarına yeni bir çalışma (yazılım üretme) yöntemi önermektedir. [13]

CMM'nin sahibi SEI (Software Engineering Institute) baş uzmanlarından Mark C. Paulk tarafından yapılan CMM ve XP incelemesinden çıkan sonuca göre: CMM konusunda hiç bir çalışma yapmayan ama sadece XP pratiklerini tam olarak uygulayan bir firma, bazı ufak tefek eksikliklerini tamamlayarak rahatlıkla CMM 3. Seviye düzeyinde değerlendirilebilir. [13]

3.1 XP ve CMM 2. Seviye Süreçleri

XP'de kullanılan *hikayeler*, *takım üyesi müşteri* ve *devamlı bütünleştirme* faaliyetleri Müşteri İsterleri Süreci kapsamındadır. Her ne kadar, müşteri isterleri zaman içinde büyük oranda değişebilse de, *küçük sürümler*, *takım üyesi müşteri*, *sürekli geri besleme* sayesinde *hikaye* üretme ve bir sonraki sürüm için en önemli hikayenin seçilmesi işlemi isterler konusunda ortak bir anlayışın oluşturulmasını sağlamaktadır.

XP'nin *oyunun planlanması* ve *küçük sürümler* Proje Planlama Süreci kapsamındadır. Proje Planlama Sürecinde istenen, yazılım ekibinin planlama sürecine katılması, oyunun planlanmasında tamamen karşılanır. Bir sonraki sürümde hangi hikayenin hayata geçirileceğine müşteri ve yazılım ekibi birlikte karar verirler. *Küçük Sürümler* sayesinde projenin kontrol altında tutulması, *Metafor* sayesinde ise büyük resmin devamlı canlı tutulması ve proje genelinde risklerin takip edilmesi sağlanabilmektedir.

Proje Takip ve İzleme Süreci kapsamında XP *Küçük Sürümler* çerçevesinde *Büyük Görsel Grafik* (Big Visual Chart), *Proje Hızı* (Project Velocity), ve taahhütler (*hikayeler*) ele alınabilir. Sürümlerin küçük olması ve hayata geçirilecek hikayelerin müşteri ile birlikte seçilmesi projenin kontrollü ilerlemesini sağlıyor. *Haftada 40 saatlik* çalışma saatlerinin projenin başarısına olumlu katkıda bulunacağı sanayiden alınan istatistiklerle doğrulanmaktadır.

Yazılım Altyüklenici Yönetimi Süreci, XP'de ele alınmamaktadır.

Her ne kadar ayrı bir Yazılım Kalite Güvence ekibi XP kültüründe yer almasa da, *Çift Programcı* uygulaması yazılımcıların birbirlerinin işlerini devamlı kontrol etmelerine neden olmakta ve dolaylı da olsa Yazılım Kalite Güvence Süreci isterleri yerine getirilmektedir. Özellikle büyük takımlarda sorunların yönetime görünebilir olması için özel önlemler alınmalıdır.

Konfigurasyon Yönetimi Süreci, açıkca adreslenmese de XP'deki *Ortak Kod Mülkiyeti*, *Küçük Sürümler*, ve *Devamlı Bütünleştirme* uygulamaları dolaylı olarak konfigürasyon yönetimini pratiklerini ima etmektedir. Büyük takımlarda Ortak Kod Mülkiyeti sorun yaratabilir. [13]

3.2 XP ve CMM 3. Seviye Süreçleri

CMM 3. Seviye süreçlerinde XP'nin genel eksikliği, organizasyonun tamamına değil, takımlara odaklanmasıdır. Bu genel eksiklik dışında genel olarak CMM 3. Seviye süreçleri desteklemektedir.

Kurumsal Süreç Odaklanması Sürecinde, XP bir çalışma yöntemi tanımlayarak bu sürece hizmet etmektedir. Ancak, XP tanımlanan sürecin organizasyonun genelinde tutarlı bir şekilde kullanıldığının takip edilmesi için bir kurumsal alt yapıya ihtiyaç duyulacağı kaçınılmaz bir gerekliliktir.

Benzer şekilde, çeşitli XP kitaplarında ve diğer kaynaklarda Kurumsal Süreç Tanımla, ve Kurumsal Eğitim Süreçlerine, katkı sağlayacak yaklaşımlar ve yöntemler tarif edilmekte ancak bunlar takım seviyesinde kalmaktadır, organizasyonun tamamına hitap edecek şekilde ayrıca ele alınmaları gereklidir.

Organizasyonun tamamına odaklanmanın doğal sonucu olarak Bütünleşik Yazılım Yönetimi Sürecinde istenen organizasyonun tamamında uygulanan bir süreç olmadığından bunun projelere göre adaptasyonu da söz konusu olmamaktadır.

Yazılım Ürün Mühendisliği Süreci, bir çok XP uygulaması ile etkin bir şekilde adreslenmektedir: *Metafor*, *Basit Tasarım*, *Yeniden Tasarım*, *Kodlama Standartları*, *Testler*.

XP, sadece yazılıma odaklanmış yaklaşımı ile çok disiplinli çalışmaları ihmal etmesine rağmen, dört temel prensibi içinde özellikle önem verdiği, *İletişim* ile Gruplararası Koordinasyon Sürecini fazlasıyla yerine getirmektedir.

Tam tanımlı bir yapısının olmamasının etkinliğini azaltması ihtimaline rağmen, *Çift Programcı* uygulaması, tartışmaya açık da olsa, *Karşılıklı Gözden Geçirme Sürecini* tam olarak yerine getirebilmektedir. [13]

3.3 XP ve CMM 4. ve 5. Seviye Süreçleri

Hızlı sürümler sayesinde erken hata bulmak ve hata engellemek mümkün olsa da XP, CMM 4. ve 5. seviye süreçlerinin, aşağıdaki tabloda görüldüğü gibi, çok azını karşılamaktadır.

XP tarafından doğrudan adreslenmeyen bir çok CMM süreci, hiç kuşkusuz ki,, gerçek projelerde doğal olarak ele alınacaktır. XP açıkça ifade etmese de uygulamalarının başarı ile yürütülebilmesi için CMM de istenen bazı yönetsel ve alt yapı ihtiyaçları vardır.

Tablo 3. XP uygulamalarının CMM Süreçlerini karşılamada yeterlilikleri [13]

Seviye	Süreç	Tatmin Düzeyi
2	Yazılım İsterler Yönetimi	++
2	Yazılım Proje Planlama	++
2	Yazılım Proje Takip ve İzleme	++
2	Yazılım Altyüklenici Yönetimi	--
2	Yazılım Kalite Güvence	+
2	Yazılım Konfigürasyon Yönetimi	+
3	Kurumsal Süreç Odaklanması	+
3	Kurumsal Süreç Tanımlama	+
3	Eğitim Programı	--
3	Bütünleşik Yazılım Yönetimi	--
3	Karşılıklı Gözden Geçirme	++
4	Sayısal Süreç Yönetimi	--
4	Yazılım Kalite Yönetimi	--
5	Hata Önleme	+
5	Teknoloji Değişimi Yönetimi	--
5	Süreç Değişimi Yönetimi	--

+ : Kısmen sağlanmakta

++: Büyük ölçüde sağlanmakta (yoruma ihtiyaç olabilir)

--: XP uygulamalarında yer almamaktadır.

4 Sonuç

XP, yazılım üretim tekniğine yönelik bir çözüm önerirken, CMM şirketin kurumsal olgunluğunun ölçülebilmesi için bir yöntem ortaya koymaktadır. XP, çift programcı gibi bazı yöntemlerle kurumsal bazı sorunlara dolaylı çözümler önerse de, aşağıdaki tabloda görüldüğü gibi, birçok önemli yönetsel ve kurumsallaşmaya yönelik pratikleri ihmal etmektedir.

XP nin yönetsel ve kurumsal eksikliklerine rağmen, yazılım geliştirme disiplinine yeni bir yaklaşım getirdiği göz ardı edilemez. Süreç odaklı CMM yaklaşımıyla XP 'nin uyuşmadığı şeklindeki yorumlar tamamen asılsız ve gerçekten çok uzaktır. Öncelikle, XP son derece iyi tanımlanmış süreçlerden oluşmaktadır. CMM ile XP nin uyumsuz olduğu konusunda iddialar, bu iki yaklaşımın tamamen birbirinden bağımsız olduğunu hatta birbirlerini tamamladıklarını gözden kaçırmaktadır. CMM, bir kuruluşun organizasyonel ve yönetsel yapısındaki eksiklikleri göstermek üzere tasarlanmışken, XP tamamen yazılım geliştirme süreçlerine odaklanmış durumdadır. CMM ile XP çelişmek ya da ters düşmek bir yana birbirlerini birçok açıdan tamamlar durumdadırlar. Son olarak, XP 'nin şu hali ile ancak 10 ve daha az sayıda elemandan oluşan takımlar tarafından başarı ile uygulanabileceği, daha büyük takımlar için sorunlar çıkabileceği yaygın bir görüştür. Büyük takımların da XP den öğrenecekleri çok şeyler olduğu gibi XP nin büyük takımlar için uyarlanmış sürümlerinin de yakın gelecekte ortaya çıkacağını tahmin etmek çok büyük bir hayalperestlik olmayacaktır.

Tablo 4. XP ve CMM Kurumsal Pratikleri [13]

XP ve CMM Kurumsal Pratikler		
Ortak Özellik (bütün Süreçler için geçerli)	Pratik uygulama	Tatmin Düzeyi
Uygulama kararlılığı	Politikalar Liderlik ve sahiplenme	-- --
Uygulama becerisi	Organizasyonel yapı	+

	Kaynaklar ve fonlar	+
	Eđitim	+
Ölçme ve Analiz	Ölçme	+
Gerçekleřtirmenin Doğrulanması	Üst Yönetim Takibi	--
	Proje Yönetiminin Takibi	++
	Yazılım Kalite Güvence	+

+ : Kısmen sağlanmakta

++: Büyük ölçüde sağlanmakta (yoruma ihtiyaç olabilir)

--: XP uygulamalarında yer almamaktadır.

Kaynakça

1. Pressman, Roger S., "Software Engineering: a practioner's approach 5th ed.", McGraw-Hill, 2001
2. Paulk M.C., Curtis B., Chrisis M.B., Weber C.V., Capability Maturity Model for Software Version 1.1, CMU/SEI-93-TR-024, 1993
3. Arthur, L.J., "Rapid Evolutionary Development," Wiley, 1992
4. Boehm, B.W., Brown, J.R., Kaspar J.R., Lipow, M., MacCleod, G.L., ve Merrit, M.J., "Characteristics of Software Quality", North-Holland, Amsterdam, 1978
5. Brooks, F., "The Mythical Man Month," Addison-Wesley, 1975
6. Crosby, P.B., "Quality is Still Free Making Quality Certain in Uncertain Times," McGraw-Hill, 1996
7. Hausen, H., "Yet Another Modelling of Software Quality and Productivity", GMD, Germany, 1987
8. Humphrey, W.S., "Managing the Software Process," Addison-Wesley, 1989
9. McCall J.A., Richards P.K., ve Walters G.F., "Factors in Software Quality", Technical Report RADC-TR-77-369, US Department of Commerce, 1977
10. Rae, A.; Robert P.; Hausen, H. L.: "Software Evaluation for Certification: Principles, Practice and Legal Liability". McGraw Hill, International Software Quality Assurance Series, London 1995.
11. Ross, N., "Version 2.0 High Level Data Model Design", ESPRIT REUEST Project, 1989
12. Vliet H.V., "Software Enginnering", WILEY, 2000
13. Paulk M.C., "Extreme Programming from a CMM perspective", IEEE Sofwtare, Nov./Dec. 2001
14. CMU-SEI, Process Maturity Profile of the Software Community – 2002 Mid-Year Update, Agust 2002.
15. CMU-SEI, High Maturity Organizations, May 2002. (<http://www.sei.cmu.edu/cmm/high-maturity/HighMatOrgs.pdf>)
16. Beck, Kent. "Extreme programming explained," Addison-Wesley, 1999.
17. Steve Hayes, "An Introduction to Extreme Programming", Khatovar Technology, 2001