Yazılım Geliştirme Sürecinin Verimliliğini Arttırmak: Bir Bilgi Sistemi Önerisi

Serkan Nalbant

MilSOFT Yazılım Teknolojileri A.Ş., ODTÜ İkizleri B Blok, 06531, ODTÜ, Ankara
snalbant@milsoft.com.tr

Özet. Bu makale ile, yazılım üreten organizasyonlarda kullanımı hedeflenen ve yazılım geliştirme sürecinin otomasyonunu sağlayan bir bilgi sistemi önerilmektedir. Bu kapsamda, önerilen sistemin özellikleri ve bu sistem ile ulaşılması planlanan hedefler tanımlanmaktadır. Buna ek olarak, bilgi sisteminin kullanımı Yazılım Geliştirme Süreci Tanımlama ve Kontrol Sistemi (PACE) yazılımı yardımıyla örneklenerek ortaya konmaktadır. PACE yazılımı, yazılım geliştirme sürecinin ve bu süreç çerçevesinde yürütülen projelerin planlanmasını, yönetilmesini, kontrolünü, ölçülmesini ve iyileştirilmesini amaçlayan bir bilgi sistemidir. Ayrıca, makalede, önerilen bilgi sisteminin dünya çapında yaygın olarak kullanılan Yazılım Yetenek Olgunluk Modeli (Software Capability Maturity Model - CMM) tarafından karşılanması beklenen gerekleri ne ölçüde yerine getirdiği, PACE yazılımı kapsamındaki yeteneklerin açıklanması sonucu yapılan karşılaştırma ile irdelenmektedir.

1 Giriş

Günümüzde yazılım sistemleri bir çok iş alanının temel bileşenleri arasında yer almaktadır. Hatta, bankacılık, telekomünikasyon gibi bazı sektörlerde gerekli yazılımlar olmadan organizasyonun yaşamını devam ettirmesi mümkün gözükmemektedir. Bu yüzden, artan rekabet, gelişen teknoloji ve yazılım kuruluşlarının artan kabiliyetlerinin de etkisiyle gelişmiş yazılım sistemlerine olan ihtiyaç her geçen gün artmaktadır. Bu seviyedeki yazılım sistemlerinin gerçekleştirilmesi karmaşık projelerin başarıyla tamamlanmasına bağlıdır. Yazılım geliştirme işlemlerini destekleyen süreçlerin yokluğunda ise, bu projelerin başarıya ulaşma ihtimali çok düşüktür.

Carnegie Mellon Üniversitesi bünyesindeki Yazılım Mühendisliği Enstitüsü (Software Engineering Institute - SEI) tarafından oluşturulan Yazılım Yetenek Olgunluk Modeli (Software Capability Maturity Model - CMM) yazılım sürecini, "yazılımın ve yazılımla bütünleşik diğer ürünlerin (örnek olarak; proje planları, tasarım dokümanları, kod, test durumları, kullanıcı kılavuzları) geliştirilmesi ve bakımı için kullanılan aktivitelerin, metodların, uygulamaların, ve dönüşümlerin oluşturduğu bütün" şeklinde tanımlamaktadır[1]. Son yirmi yıl içerisinde bir çok yazılım süreci modeli geliştirilmiştir. Bunların başlıcaları arasında TickIT, ISO 9001, BOOTSTRAP, CMM, ISO/IEC 12207, ISO/IEC TR 15504 (SPICE) sayılabilir [1,2,4,5,6,7,8]. SEI'nin ortaya koyduğu CMM bunlar arasında önemli bir yere sahiptir. Bu, CMM'in en geniş şekilde uygulanmış ve kabul görmüş modellerden biri olmasından kaynaklanmaktadır.

CMM, yazılım geliştiren organizasyonlara süreçlerini iyileştirmeleri için, yazılım alımında bulunan organizasyonlara ise yazılımı üreten müteahhit firmaların kalite düzeyini değerlendirmeleri için yardımcı olmayı amaçlamaktadır. Bu amaç doğrultusunda, CMM, yazılım firmalarına süreçlerinin mevcut olgunluk düzeyini belirleyip süreçleri iyileştirme stratejilerini seçmelerine ve bu iyileştirme çalışmaları sırasında kendilerini başarıya götürecek anahtar faktörleri tespit etmelerini sağlar. Buradaki, temel varsayım, her süreç modelinde olduğu gibi, süreçler iyileştikçe, bu süreçlere göre üretilen ürünün kalitesinin de artacağıdır[2].

Yazılım geliştiren organizasyonların yürüttükleri projeleri düzgün bir şekilde, yani planlanan bütçe içerisinde kalarak, zamanında ve beklenen kalite seviyesini tutturarak, tamamlayabilmeleri için çeşitli yazılım araçlarına ihtiyaçları vardır. Yazılım projeleri yürütülürken karşılaşılan yönetsel ve teknik zorluklar bu ihtiyacı daha da arttırmaktadır. Aslında, bir yazılımın ortaya çıkarılması için yapılması gereken bir çok işin ilgili yazılım araçları olmadan tamamlanması pek mümkün görülmemektedir. Özellikle, yazılım geliştirme projelerinin yürütülmesine ve yönetilmesine yönelik araçların projede varlığı gittikçe daha önemli bir faktör haline gelmektedir. Bu yüzden, bu makalede "Yazılım Geliştirme Bilgi Sistemleri" olarak adlandırılabilecek bu tür araçlar açıklanacak ve bu araçların kullanımı bir bilgi sistemi modeli önerisi ile gösterilecektir.

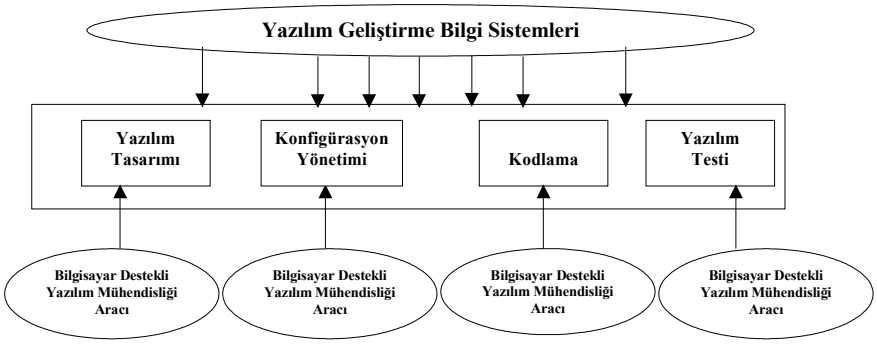
Bölüm 2'de yazılım geliştirmede kullanılan araçlar tanımlanmaktadır. Bunu izleyen bölüm 3 ise yazılım geliştiren organizasyonların önerilen bilgi sisteminin ortaya çıkmasına neden olan ihtiyaçları ve bilgi sisteminin gerçekleştirilmesi socunu ulaşılması planlanan amaçları listelemektedir. Bölüm 4'te önerilen yazılım geliştirme bilgi sisteminin temel işlevleri sunulmaktadır. Bölüm 5 önerilen bilgi sistemini Yazılım Geliştirme Süreci Tanımlama ve Kontrol Sistemi (PACE) yazılımı aracılığı ile örnekleyerek açıklamaktadır. Son olarak, bölüm 6'da kısa bir değerlendirme verilmektedir.

2 Yazılım Geliştirme Araçları

Müşterisinin ihtiyacını tam olarak karşılayan, planlanan süre ve bütçe içerisinde kalınarak gerçekleştirilen ve mümkün olduğu kadar az hata içeren bir yazılımı ortaya koyabilmek için yazılım geliştiren organizasyonların da (aynı diğerleri

gibi) bir çok araçtan (yazılımdan) faydalanması gerekir. Bu makalenin kapsamı çerçevesinde bu yazılımlar, "Bilgisayar Destekli Yazılım Mühendisliği Araçları" ve "Yazılım Geliştirme Bilgi Sistemleri" olmak üzere iki kategoriye ayrılabilir. Ghezzi, Jazayeri, Mandrioli Bilgisayar Destekli Yazılım Mühendisliği Araçları'nı 'Bir Yazılım Mühendisliği işleminin otomasyonunu amaçlayan araç veya ortam' şeklinde tanımlamaktadır[2]. Yazılım Geliştirme Bilgi Sistemleri ise bir yazılım üretilirken yerine getirilen birçok aktivitenin yönetilmesini ve bütünleştirilmesini hedefler. Yazılım Geliştirme Bilgi Sistemleri bu hedefin gerçekleştirilmesini şu özellikleri sunarak sağlar; 1)Bilginin paylaşımı ve birleşimi, 2) Yazılım organizasyondaki işlemlerin otomasyonu, 3) Gerekli planlama, kontrol ve ölçüm mekanizmalarının içerilmesi.

Yazılım Geliştirme Bilgi Sistemleri'nin kapsamı yazılım geliştiren bir kuruluştaki tüm işlemler iken, Bilgisayar Destekli Yazılım Mühendisliği Araçları belirli bir işlemin optimizasyonunu ve/veya otomasyonunu yerine getirmeye çalışır (Şekil 1'e başvurulmalıdır). Bu makalede, Yazılım Geliştirme Bilgi Sistemleri'ne yönelik bir model sunulacak ve açıklanacaktır. Bu açıklamanın daha sağlıklı olabilmesi ve önerilen modelin okuyucu tarafından daha iyi özümsenebilmesi amacıyla MilSOFT tarafından geliştirilen Yazılım Geliştirme Süreci Tanımlama ve Kontrol Sistemi (PACE)[1] yazılımından faydalanılacaktır. PACE, yazılım geliştiren kuruluşlarda kullanılmak üzere meydana getirilen web-tabanlı bir yazılım geliştirme bilgi sistemidir. PACE, bir yazılım kuruluşundaki farklı birimlerin entegrasyonunu içerdiği proje yönetimi, insan kaynakları, kalite güvence, konfigürasyon yönetimi ve bütçeleme fonksiyonları ile gerçekleştirmeyi amaçlamaktadır.



Şekil 1. Yazılım kategorilerinin yazılım geliştirme aktiviteleri ile etkileşimi

3 Önerilen Yazılım Geliştirme Bilgi Sisteminin Ortaya Çıkışı

Önerilen yazılım geliştirme bilgi sistemi yazılım geliştiren kuruluşlara yönelik olarak aşağıdaki hedeflere ulaşılmasını sağlamaya çalışmaktadır;

- Yazılım projelerinin geliştirme ve yönetim aktivitelerini standartlaştırmak,
- Kuruluşun CMM (Seviye 3), IEEE/EIA 12207 (Bilgi Teknolojisi Standartı), IEEE 1490 (Proje Yönetimi Standartı) gibi yazılım geliştirme standartlarına uygun olarak çalışmasını kolaylaştırmak,
- Kuruluşun birimleri arasındaki bütünleşmeyi en üst düzeye çıkarmak ve, kişiler arasındaki bilgi alış-verişini ve paylaşımını arttırmak,
- İşlemlerin daha etkin ve verimli olarak yapılmasına imkan vererek maliyet açısından kazanımlar sağlamak,
- Organizasyonun kaynaklarının ve süreçlerinin performansının değerlendirilebilmesi için gerekli bilgilerin oluşturulması sonucu bunların iyileştirilmesi için gerekli işlemlerin başlatılmasına imkan tanımak,
- Yukarıda listelenenlerin sonucunda, ortaya çıkarılan yazılımın kalitesini arttırmak,

Aşağıda verilen tipik iş akışı, önerilen sistemin karşılamayı amaçladığı ihtiyaçların nereden kaynaklandığını ortaya koymaktadır. Genel olarak, yazılım projeleri başlangıcından bitişine kadar şu aşamalardan geçmektedir:

- Yazılımın boyut, iş gücü ve takvim tahminlemesinin yapılması,
- Üst yönetimin onayının alınması sonrasında projenin başlatılması,
- Proje yöneticisinin atanması,
- Proje kapsamında tamamlanması gereken işleri belirten iş paketlerinin oluşturulması,
- Proje ekibinde yer almaya aday çalışanların, personel eğitim, beceri ve iş yükü bilgileri incelenerek belirlenmesi,

[1] PACE yazılımı Türkiye Teknoloji Geliştirme Vakfı (TTGV) ve Türkiye Bilimsel ve Teknik Araştırma Kurumu (TÜBİTAK) desteğiyle geliştirilmiştir.

- Proje ekibinin, seçilen personelin ilgili rollerle proje iş grubuna atanması sonucu son haline getirilmesi,
- Proje risklerinin tanımlanması ve izlenmesi,
- Projenin gelir ve gider bütçelerinin hazırlanıp onaylanması,
- Daha önce belirlenen iş paketleri için zaman ve kaynak bilgilerinin girilmesi sonucu proje takviminin oluşturulması,
- Proje takviminin üst yönetim tarafından onaylanması,
- Projenin takvim, maliyet ve kalite açılarından performansını izlemek üzere yapılacak ölçümlerin seçilmesi,
- Seçilen ölçümler için hedeflenen değerlerin belirlenmesi,
- Proje süresince üretilecek konfigürasyon maddelerinin (doküman, yazılım birimleri ve modülleri) tanımlanması ve izlenmesi,
- Onaylı proje takvimindeki iş paketlerinin proje ekibindeki çalışanlara atanması,
- Atanan işlerin tamamlanmasına yönelik izleme işlemlerinin yapılması,
- Personelin proje kapsamında yapılan işler için harcadığı zamanı bildirmesi,
- Geliştirme faaliyetleri sırasında konfigürasyon maddelerinde farkedilen yazılım ve doküman problemlerinin tanımlanması,
- Tespit edilen problemlerin, ilgili konfigürasyon kontrol merciinin onayı doğrultusunda düzeltilmesi ve bu düzeltmenin doğrulaması,
- Proje personelinin bilgi ve becerisinin yapılacak iş için eksik kaldığı noktalarda eğitim talebinde bulunulması,
- Eğitimin tamamlanması sonucu personelin beceri bilgilerinin güncellenmesi,
- Tamamlanan yazılım birimleri, yapılan testler ve gözden geçirmeler ile ilgili kayıtların tutulması,
- Projenin ihtiyaç duyduğu yazılım ve donanımın satınalınması, ve satınalma sonucunda projenin yazılım ve donanım envanterlerinin güncellenmesi,
- İş gücü, maliyet, takvim raporlarının incelenip gereken önlemleri almak için proje performansının değerlendirilmesi,
- Projenin, ilgili süreçlere uygun olarak yürütüldüğünün kontrol edilmesi amacıyla denetlenmesi,
- Proje takviminin, bütçesinin, konfigürasyon maddelerinin ve buna benzer diğer proje bileşenlerinin gerektiğinde güncellenmesi,
- Projelerde yürütülen aktivitelerin verimliliklerinin ve etkinliklerinin ölçülmesi, ve buna bağlı olarak iyileştirme işlemlerinin yerine getirilmesi.

4 Önerilen Bilgi Sisteminin İşlevleri

Kullanıcının, önerilen sistemi kullanarak gerçekleştirebileceği işlem gruplarından, yani fonksiyonlardan (işlev) başlıcaları aşağıda açıklanmıştır.

Proje Risk Yönetimi: Proje çerçevesindeki risklerin tanımlanması, değerlendirilmesi ve izlenmesi bu fonksiyon ile gerçekleştirilir. Bunun için sistemde her proje için bir risk matrisi tutulur. Risk matrisleri farklı versiyonlar halinde saklanabilir.

Yazılım Tahmini: Bu fonksiyonun amacı kullanıcının iş gücü, boyut, maliyet ve takvim tahminlerini mümkün olduğu kadar gerçeğe yakın şekilde yapabilmesi için gereken yetenekleri sunmaktır. Yazılım projesinin tahminini yapmak üzere modelde üç metodolojinin içerilmesi düşünülmüştür, bunlar; fonksiyon noktası tahminlemesi (function point estimation), nesne noktası tahminlemesi (object point estimation) ve COCOMO II (Constructive Cost Modeling) metodolojileridir. Ayrıca, bilgi sistemi projenin farklı fazları ve bileşenleri için tahmin yapılmasına imkan vermektedir.

Ölçümlerin Toplanması ve Değerlendirilmesi: Kullanıcının, projelerin zamanında tamamlanmasına ve ayrılan bütçe içerisinde kalmasına yönelik düzeltici işlemleri belirlemek üzere proje ölçümlerini izlemesi bu fonksiyon tarafından sağlanır. Sistemde bir projenin her bir ölçümü için veri listeleri saklanır. Bu listelere, yapılan ölçüm değerleri kaydedilir. Ölçüm değerleri genellikle sistemdeki verilerden, nadiren ise kullanıcıdan elde edilir. Veri listelerindeki değerler kullanılarak ortaya çıkarılan grafikler yardımıyla, kullanıcı görsel bir analiz yapma imkanı bularak projedeki potansiyel problemleri daha kolay bir şekilde tespit eder. Sistemde yer alması planlanan ölçümlerden bazıları şunlardır: Kazanılan Değer (Earned Value), Yeniden Yapılan İşler için Harcanan İş Gücü (Rework Effort), Hata Yoğunluğu (Defect Density), Yazılım Üretkenliği (Software Productivity), Gereksinim Durumu (Requirement Status), Problem Durumu (Problem Status).

Proje İş Paketlerinin ve Takviminin Oluşturulması: Bu fonksiyon kullanılarak projenin iş paketleri tanımlanır. Bu işlem sırasında sistemde tanımlı faaliyetler kullanılır. İş atamaları yapılırken, ilgili iş, daha önce sisteme girilen iş paketleri arasından seçilerek personele verilir. Sistemdeki proje iş paketi yapısı ürün tabanlı olarak, yani yazılım modülü, doküman gibi konfigürasyon maddeleri içerilecek şekilde yaratılır. Buna ek olarak, merkezi olarak tanımlanan iş paketi şablonları, belirli bir projenin iş paketi yapısı oluşturulurken kullanılarak, iş gücünden önemli bir tasarruf sağlanabilir. İş paketleri son halini aldıktan sonra, bunların her biri için başlangıç tarihi, süre, iş gücü, görevli personel,

diğer işlerle bağımlılıklar, nakit akışları bilgileri girilerek proje takvimine ulaşılır.

İş grubu (Ekip) Yönetimi: Bir grup personelin belirli bir hedefe ulaşmak için yer aldığı takımlara iş grubu adı verilir. Dolayısıyla, sistemde tanımlanan her bir bölüm ve proje için de otomatik olarak bir iş grubu yaratılır. Ayrıca, sistemde tanımlı iş grupları için hiyerarşik bir yapı oluşturmak mümkündür, yani bir iş grubunun altında bir başkası yer alabilir.

İş Atama: İş Atama fonksiyonu yardımıyla iş grubu yöneticileri altlarında çalışan personele yaptırmak istedikleri işleri belirtirler. Bir proje iş grubu için iş ataması yapılırken projenin iş paketleri kullanılır. İş ataması yapılırken sisteme girilmesi gereken temel veriler şunlardır: Personel, Faaliyet (Aktivite), Başlangıç Tarihi, Bitiş Tarihi, Süre, İşlem Tipi (İlk kez veya yeniden). Süre bilgisi, personelin ilgili iş grubundaki iş yükü ve personele yapılan diğer iş atamaları dikkate alınarak belirli iş atamasının olabilirliği değerlendirilir. Bu sayede, kuruluşun personel kaynaklarının etkin yönetimi sağlanır.

Zaman Çizelgesi: Her bir personel kendisine atanan işler için harcadığı zamanı bu fonksiyon yardımıyla sisteme girer. İş grubu (proje, bölüm) yöneticileri bir atama yaptığında bununla ilgili zaman çizelgesi kaydı otomatik olarak oluşturulur. Böylece, personelin daha verimli bir şekilde zaman girmesi sağlanır.

Eğitim Yönetimi: Bu fonksiyonda şirket personeline sunulan dahili ve harici eğitimlerin yürütülmesine yönelik yetenekler içerilmektedir. Bu kapsamda, eğitimlerin tanımlanması, eğitimin personel tarafından başarıyla tamamlanması sonucu edinilecek personel becerileri belirtilerek gerçekleştirilir. Bunun yanında, planlanan ve gerçekleşen eğitimlerle ilgili kayıtların tutulması da bu fonksiyon çerçevesinde ele alınır. Ayrıca, eğitim taleplerinin değerlendirilmesi ve sonuçlandırılması yapılır. Aynı zamanda, personelin aldığı eğitimler de burada saklanır.

Problem Tanımlama ve Çözümleme: Problem çözümleme süreci, farklı tipteki problemlerin kaydedilmesini ve çözümlenmesini sağlayan formların yardımıyla yerine getirilir. Bir yazılım projesi çerçevesinde ortaya çıkan problemler şunlar olabilir; dokümanlarda tespit edilen uygunsuzluklar, yazılım kodunda bulunan hatalar, projenin uymayı taahhüt ettiği standartlardan sapması, yapılan gözden geçirmeler, denetlemeler, testler vs. sonucu gündeme gelen projenin sürece göre tutarsız ve uyumsuz hareket ettiği noktalar.

Denetleme İşlemleri: Bu işlev proje denetlemelerinin planlanmasına olanak tanır. Bunun yanında, yapılan denetlemelerin sonuçlarının kaydedilmesi ve bunlarda tespit edilen uygunsuzlukların belirtilmesi yine burada gerçekleşir.

Konfigürasyon Tanımlaması ve Kontrolü: Bu fonksiyon aracılığıyla projenin konfigürasyon maddelerinin (yazılım ürünlerinin, yani modül, kod, doküman vs.) tanımlanması ve bunların durumlarının (versiyonlarının) izlenmesi sağlanır. Bu çerçevede, bu fonksiyon, konfigürasyon maddelerindeki değişikliklerin gerekli kontrol mekanizmalarının tamamlanması sonucu yerine getirilmesine de imkan verir.

Yukarıda detayları verilenler dışında, burada yer verilmemesine karşın modelde şu fonksiyonlar da yer almaktadır: Proje Tanımlama ve Başlatma, Faaliyet Tanımlama, Bütçeleme, Personel Performans Değerlendirmesi, Personel Bilgileri Yönetimi, Proje Maliyetlendirme.

5 Önerilen Modelin Örneklenmesi ve Örnek Bilgi Sisteminin Yazılım Yetenek Olgunluk Modeli ile İlişkisi

Belirtildiği üzere bu makalede, önerilen bilgi sistemi modelinin örneklenmesi amacıyla PACE yazılımı kısaca tanıtılacaktır. PACE yazılımı Şekil 2'de (Ek A) alt modülleriyle beraber gösterildiği gibi 5 modülden oluşmaktadır. PACE yazılımı Java 2 Enterprise Edition (J2EE) teknolojisinin N-katmanlı olarak uygulanması ile ortaya çıkarılmıştır. Yazılım platform, uygulama sunucusu ve veritabanı bağımsız olarak geliştirilmiştir. PACE yazılımının örnek bir ekran görüntüsü Şekil 3'te (Ek A) görülebilir.

CMM (Seviye 3) PACE'in desteklemeyi amaçladığı temel standarttır. Bunun nedeni, CMM modelinin dünya çapında yoğun olarak benimsenmesidir. Buna ek olarak, Yazılım Mühendisliği Enstitüsü (SEI), CMM seviye 3'ü organizasyondaki yazılım mühendisliği ve yönetim süreçlerinin tüm projeler bazında etkin bir şekilde kurumsallaşmasını sağlayan altyapıyı sunan seviye olarak tanımlamaktadır [3]. CMM'in her seviyesi Anahtar Süreç Alanlarına (Key Process Areas - KPA) ayrılmıştır. SEI bu alanları, yerine getirildiğinde organizasyonun süreç yeteneğini tesis etmesi açısından önem teşkil eden belirli bir grup hedefin gerçekleştirilmesini sağlayan faaliyetler kümesi olarak tanımlamaktadır [1]. PACE yazılımının CMM seviye 2 ve 3'ün ilgili Anahtar Süreç Alanlarından herbirini ne şekilde ele aldığı aşağıda verilmiştir.

Gereksinim Yönetimi (Requirements Management): Farklı gereksinim tipleri tanımlanır ve bunlar, *Gereksinim Durumu* ve *Gereksinim İstikrarı* ölçümleri ile izlenir. Ayrıca, gereksinimlerdeki değişiklikler resmi değişiklik yönetimi mekanizmaları ile gerçekleştirilir.

Yazılım Projesi Planlama (Software Project Planning): Yazılımın boyut, iş gücü ve takvim tahminleri fonksiyon noktası tahminlemesi (function point estimation), nesne noktası tahminlemesi (object point estimation) ve COCOMO II (Constructive Cost Modeling) metodolojileri ile yapılır. Buna ek olarak, yazılım projesinin iş paketi yapısı ve takvimi oluşturulur, proje riskleri tanımlanır, projenin yazılım ve donanım envanterleri yönetilir.

Yazılım Projesi Takibi (Software Project Tracking and Oversight): Yazılım projesinin iş gücüne, kalitesine, takvimine, üretkenliğine yönelik birçok ölçüm izlenir. Gerektiğinde, proje takvimi güncellenir. Aynı zamanda, projenin riskleri risk matrisi yardımıyla izlenir. Son olarak, proje çerçevesinde çıkarılan dersler kazanılan tecrübenin kuruluşun geneline yayılımını sağlamak üzere kaydedilir.

Yazılım Kalite Güvence (Software Quality Assurance): Kalite faaliyetleri planlanır ve izlenir, denetleme sonuçları kaydedilir, uygunsuzlukların çözümlenmesi gerçekleştirilir.

Yazılım Konfigürasyon Yönetimi (Software Configuration Management): Konfigürasyon maddeleri (yazılım ürünleri) belirlenir ve tanımlanır. Ayrıca, konfigürasyon maddelerinin durumları ilgili gruplara ve kişilere yazılımda hazırlanan durum raporlarıyla bildirilir. Buna ek olarak, konfigürasyon kontrol kurullarının onayına tabi olan ilgili değişiklik formlarının doldurulması sonucu konfigürasyon maddelerinde değişiklikler yapılır.

Organizasyon Süreç Odaklılığı (Organization Process Focus) VE Organizasyon Süreçlerini Tanımlama (Organization Process Definition): Yazılım geliştirme sürecinin yürütülmesi ve iyileştirilmesi gerekli talep ve onay mekanizmaları yardımıyla gerçekleştirilir. Bunun yanında, yazılım projeleri kapsamında yerine getirilen süreçler ve faaliyetler tanımlanır, bunlar daha sonra proje iş paketleri oluşturulurken ve iş atamaları yapılırken kullanılır. Tüm projelerde tamamlanan işler sonucu toplanan iş gücü ve maliyet ortalamaları raporlanır. Bu raporlar süreçlerin ve aktivitelerin performanslarının ölçülmesi amacıyla kullanılır. Bir faaliyetin prosedürünün değiştirilmesinin etkileri de faaliyet versiyonlama mekanizması sayesinde izlenebilir. Projenin seçtiği yazılım geliştirme sürecine tekabül eden organizasyonel standartlar kullanıcılara sunulurak günlük işleyişte kolaylık sağlanır.

Bütünleşik Yazılım Yönetimi (Integrated Software Management): Proje takvimi, projenin yazılım geliştirme planına göre organizasyon çapındaki süreç veritabınından gerekli süreçler ve faaliyetler seçilerek oluşturulur. Geçmiş ölçüm ve tahmin değerleri yeni yapılacak tahminlerde kullanılmak üzere kullanıcıların kullanımına sunulur.

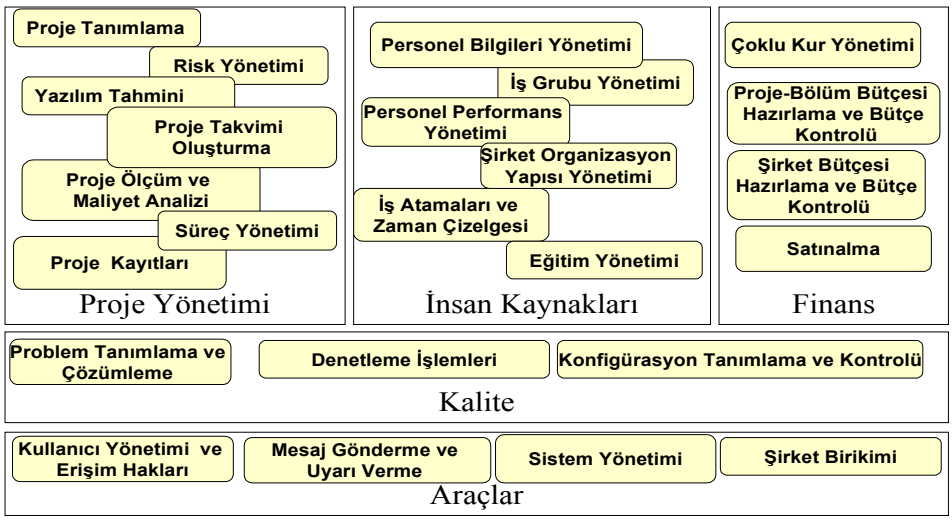
Gruplararası Koordinasyon (Intergroup Coordination): Geliştirme ekipleri arasındaki bağımlılıklar iş paketleri ve takvim oluşturulurken belirtilir. Belirli olaylar gerçekleştiğinde ilgili kişileri haberdar etmek üzere otomatik mesajlar gönderilir.

Eş Gözden Geçirmeleri (Peer Reviews): Eş gözden geçirmeleri proje takviminde planlanır. Yapılan eş gözden geçirmelerinin sonuçları kaydedilir ve bunlarda tespit edilen problemlerin veya işlemlerin durumu eş gözden geçirme raporlarından referans verilen formlar sayesinde izlenir. Buna ek olarak, *Gözden Geçirme Durumu* ölçümü yardımıyla, eş gözden geçirmeleri analiz edilir.

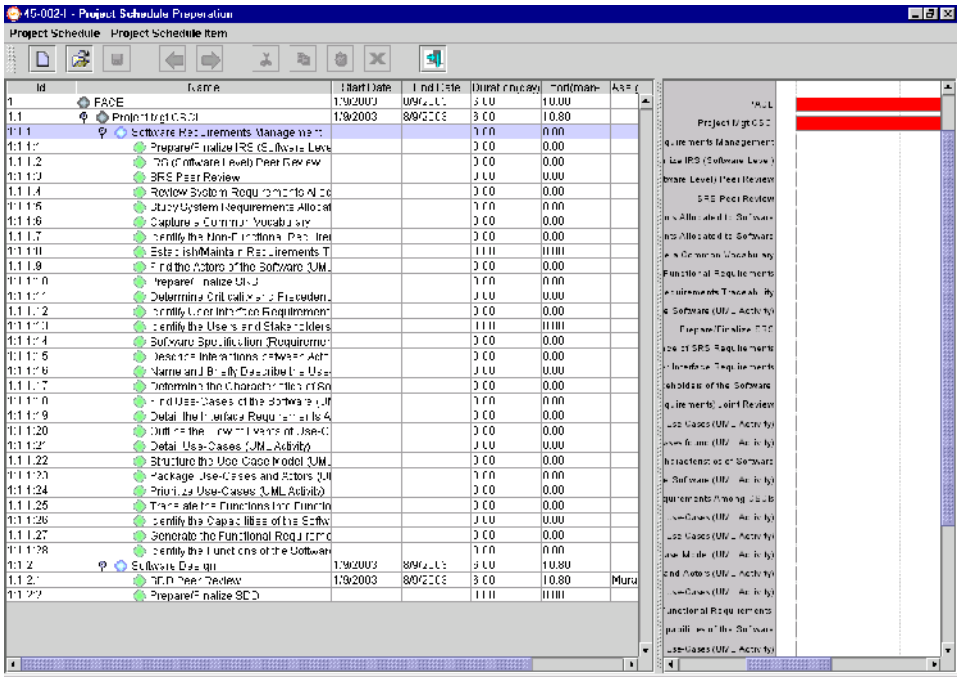
Eğitim Programı (Training Program): Bu anahtar süreç alanı ile ilgili olarak şu özellikler sağlanır: eğitimlerin tanımlanması, eğitim planlarının yapılması, tamamlanan eğitimlerin kayıtlarının tutulması, personelin eğitime katılması sonucunda becerilerinin otomatik olarak güncellenmesi.

6 Sonuç

Yazılım geliştiren organizasyonlar, rekabetçi pazarlarda sürekli var olabilmek için aynı anda yürütülen bir çok karmaşık proje ile baş etmek zorundadırlar. Projelerin karmaşıklığının yanında, eş zamanlı olarak yürütülmeleri de yazılım geliştirme ortamını karmaşıklaştırmakta ve yazılım projelerinin başarı ile tamamlanmasını daha zor hale getirmektedir. Dolayısıyla, yazılım geliştiren organizasyonlar, hedeflerine ulaşabilmek için, yazılım geliştirme bilgi sistemlerini bünyelerine katarak yazılım yönetim ve geliştirme süreçlerini daha etkin ve verimli hale getirmelidirler. Bu makalede, bu tür bilgi sistemleri için yazılım kuruluşlarına yönelik yüksek faydalar sağlamayı amaçlayan bir model önerilmiştir.



Şekil 2. PACE Modülleri



Şekil 3. Örnek Bir PACE ekranı : Proje Takvimi Hazırlama

Kaynakça

1. T.G. Olson, N.R. Reizer, J.W. Over, Handbook CMU/SEI-94-HB-01: A Software Process Framework for the SEI Capability Maturity Model, SEI, Eylül 1994.
2. C. Ghezzi, M. Jazayeri, D. Mandrioli, Fundamentals of Software Engineering, Prentice Hall, Pearson Education Inc., New Jersey, 2003.
3. M.C. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber, Technical Report CMU/SEI93-TR-024: Capability Maturity Model for Software, Versiyon 1.1, SEI, Şubat 1993.
4. "BOOTSTRAP: A Software Process Assessment And Improvement Methodology", Similä, J., Kuvaja. P., Krzanik L., International Journal of Software Engineering and Knowledge Engineering, 1995, vol. 5, no 4.
5. IEEE/EIA Standard: Industry Implementation of International Standard ISO/IEC 12207:1995 Standard for Information Technology--Software Life Cycle Processes.

6. ISO/IEC (1998a) 15504-2 Information technology - Software process assessment – Part 2: A reference model for processes and process capability. ISO/IEC TR 15504-2: 1998(E).

7. TickIT Guide, Guide to Software Quality Management System Construction and Certification Using EN29001, Issue 2.0, DISC TickIT Office, 1992.

8. ISO 9001: 2000, Quality management systems - Requirements