Servis Tabanlı Kurumsal Yazılım Geliştirilmesinde XP Kullanımı ve Eleştirisi

Ensar GÜL[1], Ünal YILDIRIM[2]

[1]Avrupa Yazılım, Tübitak Mam Tekseb A Blok No: 201 Gebze, Kocaeli
[1]eposta: ensar.gul@avrupayazilim.com

Özet

Bu bildiride, yazılım geliştirmede kullanılan Uç Programlama "Extreme Programming" (XP) yaklaşımının servis tabanlı kurumsal yazılımlarının geliştirilmesinde kullanılmasının doğurduğu pratik sonuçlar anlatılmış ve XP'ye eleştirel bir bakış açısı getirilmiştir.

Abstract

In this paper the practical results of applying Extreme Programming(XP) in the development of an enterprise application using service oriented architecture are presented and XP methodology has been criticized.

1. Giriş

Uç Programlama "Extreme Programming" (XP), kaliteli bir yazılımı, mümkün olan en kısa sürede gerçekleştirmeyi hedefleyen, bir dizi, değer, kural ve uygulamadan oluşan bir yazılım geliştirme yöntemidir. XP, yazılım projelerinde riskleri azaltan, projenin ortamdaki değişimlere en iyi şekilde uyum sağlamasına izin veren ve sistem yaşam döngüsü kapsamında verimliliği ön planda tutan çevik bir yöntem olarak kabul edilmektedir[1]. XP hakkında ayrıntılı bilgi için [2,3,4,5,6] kaynaklarına bakılabilir. Fakat uygulamada XP nin bu iddialarının her zaman ve ortamda gerçekleşmediği gözlenmiştir. Bu bildiride geniş ölçekli bir kurumsal yazılımın geliştirilmesinde uygulanan XP yaklaşımından kazanılan deneyimler anlatılacaktır. Bildirinin ikinci bölümünde XP`nin kullanıldığı yazılım hakkında bilgi verilmiş, üçüncü bölümde bu yöntemin servis tabanlı kurumsal yazılım geliştirilmesinde uygulanması ve karşılaşılan güçlükler anlatılmış, dördüncü bölümde ise sonuçlar tartışılmıştır.

2. Tedarikçi Yönetim Sistemi (TYS)

Proje Kapsamı

Yazılımın geliştirildiği kurum, üretime yönelik hammadde ve yarı mamul alımları ile satış destek (promosyon) malzemelerinin yurt içi alımlarını Satın Alma Koordinatörlüğü ve bu koordinatörlüğe bağlı 18 Satın Alma Müdürlüğü ile gerçekleştirmektedir.

Satın Alma Müdürlükleri stok takibi, talepleri izleme, sipariş açma gibi operasyonel işlemlerini gerçekleştirmek için AS-400 sistemini kullanmaktadırlar. Satın almanın yanı sıra malzeme yönetimi, kalite kontrol, üretim planlaması, satış ve muhasebe gibi temel işletme ihtiyaçlarını karşılamaya yönelik olarak geliştirilen AS-400 Sistemleri her şirkette birbirinden bağımsız olarak çalışmaktadır.

TYS öncelikle tedarikçi ve malzeme fiyatı onaylama süreçlerini elektronik ortama taşımayı ve böylece bu süreçleri kayıt altına alıp, daha verimli izlenebilir bir hale getirmeyi hedeflemektedir.

İkinci olarak TYS intranet aracılığı ile tüm şirketlerin ortak olarak kullanacağı ve katkıda bulunacağı bir bilgi tabanı olarak karar alma sürecinde Satın Alma Müdürlüklerine destek niteliğindedir.

TYS malzeme ihtiyaçlarını belirleme, talep yaratma, sipariş açma ve malzeme kabul etme gibi operasyonel ihtiyaçlara bu versiyonda cevap vermemektedir. Sistemin kapsamı, tedarikçi ve fiyat onaylama süreçleri çevresinde satın alma karar destek ortamına gerekli bilgi dağarcığını oluşturmak ve bunun tüm grup şirketleri arasında paylaşılmasını sağlamaktır.

Bu amaçla sistemin şu fonksiyonları gerçekleştirilmiştir:

a. Genel Kullanıcı İşlemleri
b. Tedarikçi Bilgileri Gir/Güncelle
c. Tedarikçi Onayla
d. Malzeme Bilgileri Gir/Güncelle
e. Malzeme ve Fiyat Belirle
f. Satın Almayı Onayla
g. Sistemi Yönet
h. Rapor Çek

Tedarikçi İlişkileri Yönetim Sistemi ile; satın alma sürecinde farklı birimlerden/şirketlerden toplanan bilgilerin paylaşıma açılması, onay süreçlerinin elektronik olarak kayıt/gözlem altına alınması ve bu yolla satın alma kararlarının verilme sürecinde önemli bir verimlilik ve kontrol artışı sağlanması hedeflenmektedir.

2.1. Proje Teknik Yapısı

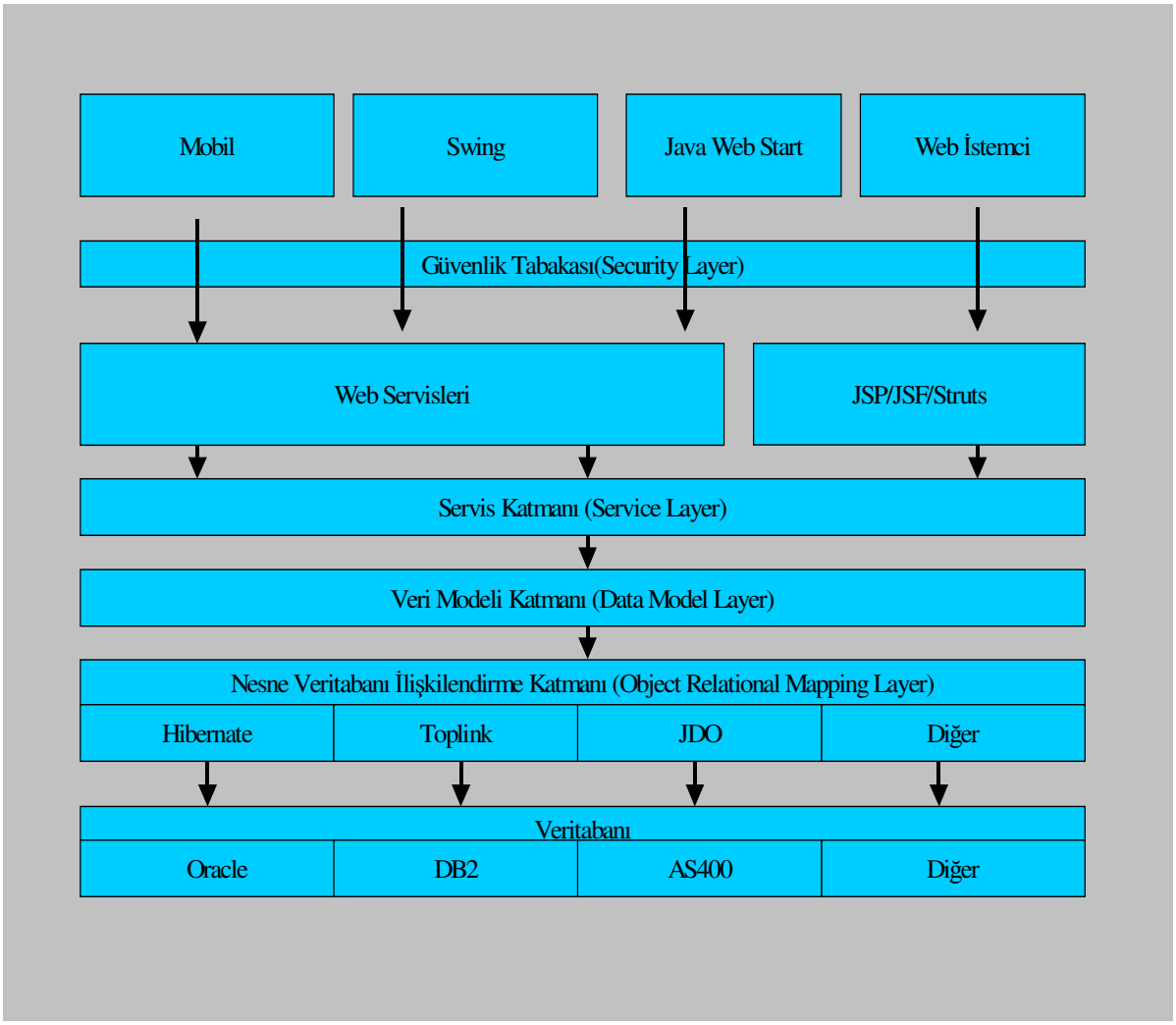
Ürün bir Java uygulaması olarak, J2EE teknolojileri kullanılarak, *çok-katmanlı (multi-tier)* mimari yaklaşım ile tasarlanmıştır. Platform bağımsızlık (Unix, Windows, Linux), veritabanı bağımsızlık (Oracle, DB2, mysql, postgresql, DB2400, Sybase, vb.) ve uygulama sunucusu bağımsızlık ürünün ana özellikleri olarak planlanmıştır. Uygulamanın genel katmanları Şekil 1'de görülmektedir. Uygulama web, mobil telefon ve masa üstü kullanıcılarını destekleyecek şekilde tasarlanmış ancak şu ana kadar sadece web arayüzü gerçekleştirilmiştir**.**

TYS sistem mimarisi, MVC (Model View Controller) modeline uygun bir şekilde, modüler yapıya sahip olacak şekilde ve esnek bir veri modeli üzerine kurulmuştur. Kullanıcı ön yüzü değişik ara birimlerden olabileceği gibi, ilk tasarım web uygulaması olarak yapılmıştır. Sunucu kısmında iş mantığı modüler bir yapıya bölünmüş, dışarıdan ve içerden çağrılabilecek servisler üzerine kurulmuştur. Kalıcı Nesneler (Persistent Object) tabakasında `hibernate`[7] kullanılarak JDBC API

karmaşası uygulamadan soyutlanmış; veri modeli, kavramsal ve fiziksel olmak üzere iki seviyede normalize bir şekilde tasarlanmıştır.

Kullanıcı ara yüzü alt yapısı olarak JSF[8] kullanılmıştır. Uygulamanın alt yapısında Spring[9] yapısından faydalanılarak, 'singleton' sınıflar uygulama başlangıcında oluşturularak tek bir yerden yönetimi sağlanmıştır. Uygulamanın çoklu dili desteklemesi amacıyla *özellikler (properties)* dosyaları kullanılmıştır. Uygulamadaki veri tabanı işlemlerin bütünlüğü için bir işlem yöneticisi (Transaction Manager) ve servislere erişim için servis yöneticisi kullanılmıştır. Uygulamada kullanıcıya hata mesajlarının gösterilmesinde çoklu dil desteklenmiş ve hata mesajları sınıflandırılmıştır. Genel amaçlı kullanılacak sınıflar ve arayüzler *common*, *share* ve *util* gibi paketlere ayrılmıştır. Bütün uygulamada *geliştirme, test* ve *üretim aşamasında*ki log mesajlarının ayrılması için log4j kullanılmıştır.

Geliştirilen yazılım yaklaşık 180K satırdan oluşmaktadır.



Şekil 1. Yazılım Mimari Katmanları

3. XP Uygulaması ve Karşılaşılan Zorluklar

Yazılım geliştirme sürecince sürümler iki hafta olarak planlandı. Proje ekibinde 14 kişi görev aldı. Bu kişiler 7 çift olarak programlama yaptılar. 5. sürümün sonunda yazılım bitirilmesi hedeflenmişti. Bazı eksiklere rağmen bu sürenin sonunda çalışan bir yazılım üretildi. Aşağıda XP prensiplerinin uygulması konusunda karşılaşılan zorluklar ve bu bağlamda XP'ye getirdiğimiz eleştiriler sıralanmıştır.

Planlama ve küçük sürümler

- XP çerçevesinde ilk sürümlerde çalışır bir programın iskeletinin çıkarılması kolay değildir. Özellikle karmaşık sistemlerde sistemin tümününe hakim olunması, iş parçalarının tanımlanması, bunların gerçekleştirilmesi XP nin iddia ettği gibi kolay olmamaktadadır. Geliştirdiğimiz uygulamada veri erişim katmanı, servis katmanı, GUI`de kullanılan teknolojiler başlı başına uzmanlık gerektiren, öğrenilmesi zaman alan teknolojilerdir. Bunların hepsinin yazılımcılar tarafından iyi derecede öğrenilmesi ve verimli bir şekilde kod üretilmesi zaman kaybına yol açmaktadır.

- İşler yapıldıkça sürekli yeni işler ortaya çıkmaktadır. Bu da proje ile ilgili sağlıklı bir tahmin yapılmasını zorlaştırmaktadır. İlk başta yaklaşık 300 adet iş birimi (task) planlanmış iken yazılım süreci boyunca bu iş birimi sayısı artarak yaklaşık 700 adet olmuştur. Ayrıca buna ek olarak yaklaşık 600 adet değişiklik ya da hata eklenmiştir.

Eşli Programlama (Pair Programming)

Yazılım sırasında başlangıçta % 80 oranında eşli programlama (pair programming) kullanılmış, sona doğru rotasyonun getirdiği uygulama zorlukları nedeni ile bu oran % 20 'ye kadar düşmüştür. Eşli programlamada karşılaşılan zorluklar:

- Sürekli eş değişimi nedeni ile yazılım güçlükleri ortaya çıkmıştır. Kişilerin yaptıkları işler yarım kalmıştır. Yarım kalan işleri eşler arasında paylaştırmak sorun olmuştur.

- Yazılımcıların çalışma alanlarının belli olmayışı herhangi bir alanda uzmanlaşmayı engellemiştir. (Bu durum sık sık eş değiştirilmesinden kaynaklanmıştır).

- Bir işin iki kişiye aynı anda verilmesi, kişilerde sorumluluk duygusu oluşmamasına neden olmaktadır.

- Bir işten iki kişinin sorumlu olması, kişilerin iş hakkında karar verme ve uygulamaya geçme hakkını kendilerinde görememesine neden olmaktadır.

- İş yapıldıktan sonra hangi işi kimin yaptığının tam olarak belli olmayışı hataların sorumlusunun tam olarak anlaşılamamasına neden olmaktadır.

- Sonradan hangi işi kimin yaptığının tam olarak bilinmemesi başarı değerlendirilmesi konusunda sağlıklı ve somut veri bulunmamasına neden olmaktadır.

- İşlerin değişik parçalarının değişik eşler tarafından yapılması, işler arasında kopukluğa neden olmaktadır.

- İşlerin değişik kişiler tarafından yapılması ve bir işin iki kişiye paylaştırılması işlerin hızlı ve verimli yapılması ile çelişmektedir.

3.1. Ortak Kod Sahipliği

Yazılım büyük ölçekli olduğu için bir sürüm boyunca işlerin tanımlanması hayli zor olmuştur. Özellikle kısa süreli işler bulmak ve tanımlamak kolay olmamaktadır. Bazı iş parçaları sürüm süresi boyunca devam etti. Bu süre sonunda eğer verilen iş eksik kaldıysa, bundan sonraki sürümde farklı bir çift programcıya verildi. Yeni işi alan programcılar yazılan kısımları öğrenmek ve bunun üzerine kod yazabilmekte zorlandılar. Ortak kod sahipliği yazılımcıların çıkan sorunları ve eksikleri sahiplenmemesine yol açmaktadır.

3.2. Ayaküstü toplantılar

Bu toplantılar çok faydalı olmaktadır. Bu toplantılarda alınan, yazılımda uygulanacak yöntemlere ilişkin önemli teknik kararlar, yazılı hale getirilerek yayımlanmalıdır. Aksi halde bu yöntemlerin uygulanmasında zorluklar ortaya çıkmaktadır.

3.3. Birim Testler (Unit Testing)

Geliştirilen yazılıma her yazılımcı müdahale edebilmektedir. Bir çift yazılımcının yaptığı hatanın bütün sistemi etkilememesi ve sistem entegrasyonun bozulmaması içim birim testler yapılmalıdır. Veri tabanı içeren sistemlerde yazılım veri tabanını değiştirdiği için, birim testlerinin yazılması zor olmuştur.Veri tabanı bağımsız olan modüller için birim testler yazılmış olup, bunun için junit kullanılmıştır.

3.4. Yeniden Yapılandırma (Refactoring)

XP, yazılım mimarisinin önceden belirlenmediği, sürekli yapılan yeniden yapılandırmalarla nihai mimarinin ve yazılım yapısının ortaya çıkmasını hedefliyor. Biz bu konuda üst seviye bir mimari tasarımın olmasının daha faydalı olduğunu düşünüyoruz. Bu yaklaşımımızla belirsizliklerin biraz daha azaltılmasını hedefledik. Hemen kodlamaya başlamak ve korkmadan yazılan bölümleri çöpe atmak pratikte söylenildiği kadar kolay olmamaktadır. Yazılım ortaya çıkmaya başlayınca çeşitli nedenler, örneğin bazı modüllerin yavaş çalışması, yeniden yapılandırılacak kısımları belirlemektedir. Bu kısımlar tamamen atılarak veya yeniden yazılarak müşterinin istediği ürün ortaya çıkmaktadır.

Yeniden yapılandırma sırasında, sürümün tamamen bitirilmesi ya da sürüm tam olarak bitmiyor ise bile dondurularak genel bir yeniden yapılandırma için zaman ayrılmalıdır. Sürüm sonunda yazılımın tam olarak dondurulamadığı durumlarda sorun yaşanmıştır.

3.5. Müşteri ile birlikte çalışma

XP bir müşteri temsilcisinin sürekli programcılarla birlikte çalışmasını öngörüyor. Bu erken geri besleme açısından çok faydalıdır. Hatta analiz aşamasında sadece analizcilerin degil bütün programcıların bir ölçüde analiz çalışmalarına katılması ve işin bütününü görmesi program geliştirirken işlerini kolaylaştıracaktır.

4. Sonuçlar

Geliştirdiğmiz yazılımda sürüm (iteration) süresi 2 hafta olarak planlanmıştı. Her sürüm sonunda olgunlaşmış işlerin ortaya çıkması için bu sürenin yeterli olmadığı gözlendi. Proje boyunca gerekirse sürüm süresi değiştirilmeli, fakat önceden belirlenmiş en fazla süreyi, örneğin 3 haftayı, aşmamalıdır.

Veri tabanı üzerinde birim testlerin zorluğu nedeni ile bu testlerin verimli kullanılması mümkün olmamaktadır.

Yazılımcıların çalışma alanlarının sürekli değişmesi, nedeni ile herhangi bir alanda uzmanlaşmayı engellemektedir.

XP de rotasyon yapılması yazılımı güçleştirmektedir. Eşli programla, herhangi iki kişiden birisinin ayrılması durumunda, bilgi birikiminin korunmasını yeterince sağlamaktadır. Fakat rotasyon yapılmasının iş gücü kaybına neden olacağı göz önüne alınmalıdır. Mutlaka eş değişimi yapılacak ise bu küçük gruplar içinde olmalıdır.

Proje süresince bir işin sorumlusu mümkün olduğu kadar sabit kalmalıdır. Eğer işin sorumlusu sık sık değişir ise, iş ile ilgili yapılmayan bölümün sorumluluğunu alamamaktadır.

XP ile projenin ne kadar süreceğini tahmin etmek zordur. Bu, metodolojinin kendisinden kaynaklanmakadır. Sürekli yeniden yapılandırma ile sonuca varılmaya çalışılmakdadır. Yapılan yeniden yapılandırmanın, projeye nasıl etki edecegi ve projenin ne kadar süreceği tahmin edilememektedir.

Karmaşık sistemleri iyi tanımlı alt bölümlere ayırarak XP nin uygulanması faydalı olabilir. Sistemin tümüne birden XP uygulanması beklenen faydaları sağlamamıştır.

Diğer yazılım metodolojileri gibi XP de her kapıyı açan bir anahtar değildir. Kullanılmadan önce getireceği olumlu ve olumsuz sonuçlar düşünülmeli, gerekirse projeye has yeni bir metodoloji kullanılmalıdır.

5. Kaynakça

1. Telekomünikasyon Yazılımlarının Geliştirilmesinde XP Yaklaşımı, Taylan Şekerci, Aziz Can Yücetürk, Ensar Gül, UYMS 2003, İzmir.
2. Beck, K., Extreme Programming Explained: Embrace Change, Addison Wesley, 2000, ISBN: 0201616416.
3. Jeffries, R., Anderson, A., Hendrickson, C., Extreme Programming Installed, Addison Wesley, 2001, ISBN: 0201708426.
4. http://www.xprogramming.com/
5. Exreme Programming FAQ, http://www.jera.com/techinfo/xpfaq.htm
6. The new Methodology, Martin Fowler, http://www.martinfowler.com/articles/newMethodology.html
7. www.hibernate.org
8. Java Server Faces, http://java.sun.com/j2ee/javaserverfaces/reference/api/index.html
9. www.**spring**framework.org