

Telekomünikasyon Yazılımlarının Geliştirilmesinde XP Yaklaşımı

Taylan Şekerci, Aziz Can Yüçetürk, Ensar Gül

Oksijen Teknoloji, İstanbul Dünya Ticaret Merkezi, A1 Blok K:13 34149 Bakırköy, İstanbul
taylan.sekerci, aziz.yuceturk, ensar.gul@oksijen.com

Özet. Bu bildiride, yazılım geliştirmede kullanılan “Extreme Programming” (XP) yaklaşımının telekomünikasyon yazılımlarının geliştirilmesinde ne şekilde kullanılabileceği, yöntemin getirdiği olumlu sonuçlar ile karşılaşılan sorunlar ve zorluklar incelenmiştir.

1 Giriş

“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 verimliliği ön planda tutan çevik bir yöntem olarak kabul edilmektedir. Yazının, bundan sonraki bölümünde XP yaklaşımı tanımlanmış, üçüncü bölümde bu yöntemin telekomünikasyon yazılımlarına uygulanması anlatılmış, takip eden bölümlerde ise uygulama değerlendirilmiş ve sonuçlar tartışılmıştır.

2 Extreme Programming (XP)

XP, küçük veya orta ölçekli yazılım geliştirme grupları tarafından kullanılabilen, çok karmaşık olmayan, etkin, esnek, düşük-riskli ve öngörülebilir bir yazılım geliştirme yaklaşımıdır. Özellikle ürün gereksinimlerinde ve pazar koşullarında sık değişikliklerin meydana geldiği veya belirsizliklerin fazla olduğu yazılım projelerinde etkin olarak kullanılabilmektedir [1].

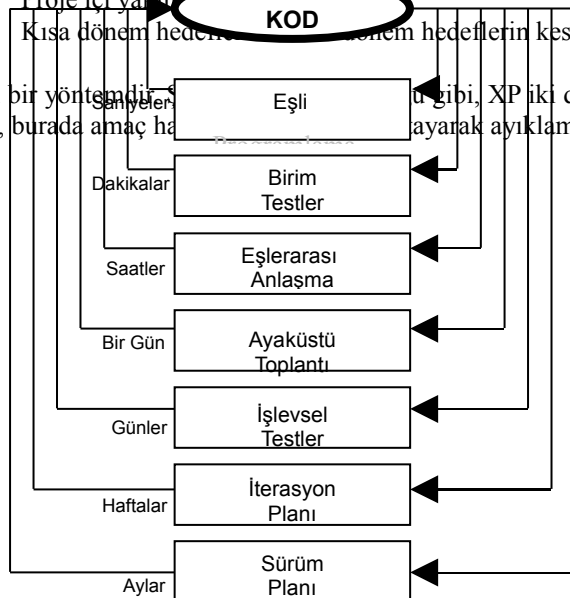
XP’yi diğer yöntemlerden ayırt eden en önemli özellikler, insana dayalı olması, insanın özel kabiliyetlerini değerlendirebilmesi, hafif (lightweight), çevik (agile) ve test odaklı olması şeklinde sıralanabilir.

XP yaklaşımı, genel anlamda dört ana değer ve oniki prensip ile tanımlanmaktadır. Dört ana değer, “iletişim”, “basitlik”, “geri besleme” ve “cesaret” olarak kabul edilmektedir. Yöntemdeki temel uygulamayı tanımlayan oniki prensip ve uygulamada elde edilen sonuçlar bir sonraki bölümde detaylı olarak incelenmiştir. XP’yi şekillendiren bu prensipler, ana değerlerin bir veya birkaçını beslemektedir. Bunlardan bazılarının eksikliği, kurulmaya çalışılan sistemin beklenen yüksek verimi verememesine neden olabilmektedir. Buna karşın, ayrı ayrı her bir XP prensibinin geliştirme sürecine katkısının olacağı da unutulmamalıdır.

XP yaklaşımını, diğer yöntemlerden ayırt eden temel özellikler şunlardır:

- Erken geri besleme
- Artırımlı planlama yaklaşımı
- İşlevlerin esnek olarak planlanabilmesi
- Otomatikleştirilmiş testler ile hataların daha erken keşfedilmesi
- Sözlü iletişim kullanılması
- Devam eden ve sürekli gelişen bir tasarım süreci
- Proje içi yazılım
- Kısa dönem hedefleri ile uzun dönem hedeflerinin kesişmesi

XP test odaklı bir yöntemdir, burada amaç her şeyden önce yazılımın çalışmasını sağlamaktır. XP iki düzeyde test ile geri beslemenin maksimum olmasını sağlar. Bu şekilde, XP iki düzeyde test ile geri beslemenin maksimum olmasını sağlar [2].



Şekil 1. Planlama / Geri Besleme Döngüleri

3 Telekomünikasyon Uygulamalarında XP Yaklaşımı

XP yaklaşımının pratiğe geçirilmesi ve yaklaşıma işlerlik kazandırılması aşamasında on iki prensip temel alınmaktadır. Telekomünikasyon yazılımları geliştirilirken bu temel prensiplerin nasıl uygulandığı ve ne gibi sorunlar yaşandığı aşağıda başlıklar halinde irdelenmiştir.

1. Planlama: Planlama, çeşitli iş öncelikleri ve teknik öngörüler göz önünde tutularak, sonraki sürümün planlamasının yapılması ve elde edilen pratik sonuçlara göre planların güncellenmesi olarak tanımlanmaktadır. Uygulamada, özellikle projenin başlangıcında hem müşterinin hem yazılımcıların konuya yabancı olması ve bazı konular için öğrenme sürecinin uzun zaman alması nedeniyle, planların yapılmasında zorluk çekilmektedir. İterasyonlar ilerledikçe ve “know-how” arttıkça öncelikleri belirlemek ve planları yapmak kolaylaşmaktadır.

2. Küçük Sürümler (Small Releases): Planlamanın tutarlı, geri beslemenin hızlı olabilmesi için sistemin basit ve sürümlerin mümkün olduğunca küçük tutulması prensibi söz konusudur. Telekomünikasyon yazılımı geliştirilirken bazı protokollerin kodlanması ve çalıştırılması bir iterasyondan daha uzun sürebilmektedir. Bu sebeple bazı iterasyonlarda ürün denebilecek sürümlerin çıkarılması zor olmaktadır. Ürünün belli başlı modülleri ortaya çıktıktan sonra, yeni küçük özellikler eklemek daha kolay olmaktadır.

3. Metafor: Bütün geliştirme sürecinin basit bir senaryo ile modellenmesi, böylece çalışanların projenin temel bileşenlerinin ve bunların birbirleri ile ilişkilerinin bilincinde olması olarak tanımlanabilir. Metafor, çalışanların sistemin bütününe kavrama katkısı ile birlikte, kişiler arası iletişimi kolaylaştırarak arttırmaktadır. Telekomünikasyon yazılımları genellikle karmaşık yapılı olmakta, bu sebeple herhangi bir modül üzerinde çalışanlar sadece kendi modülleri üzerinde odaklanmakta, diğer modüller hakkındaki bilgileri yeterli seviyede olmamaktadır.

4. Basitlik: Bu prensibe göre sistem mümkün olduğu kadar basit tasarlanmalıdır. XP yaklaşımı, daha basit bir çözüm için daha fazla zaman harcamayı öngörmektedir, böylece ürünün yeniden yapılandırılması (refactoring), testi ve bakımı çok daha kolay olmaktadır. Basitlik, görev (task), ürün geliştirme senaryoları (user story) ve sistem gibi çeşitli düzeylerde öncelik verilmesi gereken ve projenin geleceğini güvence altına alan bir prensiptir.

5. Test: XP’de yazılım geliştiricilerin birim testler (unit tests) yazmaları kaçınılmaz bir gerekliliktir. Bu prensibin uygulanmasında ise, bazı yazılımcıların ilk defa birim test yazmaları ve bu işin yazılımcılara ekstra yük getirmesi gibi sebeplerden dolayı çeşitli zorluklar yaşanabilmektedir. Değişik sektörlere yönelik yazılımlarda farklı birim testler yazılması bilgi birikimi gerektirdiğinden, telekomünikasyon yazılımlarında da birim testlerin hazırlanması için özel bir bilgi birikimi gerekmektedir. Bu birikimin şirket bünyesinde oluşumu ise zaman almaktadır.

6. Yeniden Yapılandırma (Refactoring): Bu prensip, sistemin davranışını değiştirmeden yeniden yapılandırılması olarak tanımlanmaktadır. Telekomünikasyon yazılımlarında birçok proje, daha özgün tasarımlar kullanılarak yeniden yapılandırılmış, risklere rağmen olumlu sonuçlar alınmıştır. Bu prensip en büyük desteğini birim testlerden almaktadır.

7. Eşli Programlama (Pair Programming): Kodların, her bir makinede iki kişi çalışacak şekilde geliştirilmesidir. Bu

prensibin uygulamaya geçirilmesi aşamasında, iki kişinin aynı program birimi üzerinde çalışması önceleri kaynak israfı olarak görülmüş, fakat bunun koda getireceği kalite ve bundan dolayı hata sayısının azalması gibi sonuçlar bu tür çekinceleri geçersiz hale getirmiştir. Eşli programlama yapılması proje içi bilgi aktarımı açısından da faydalı olmuştur.

8. Ortak Kod Sahipliği: Her yazılımcının sistemin her biriminde kodlama yapabilmesidir. Telekomünikasyon uygulamalarında sistemin karmaşık olmasından ve öğrenme sürelerinin uzun olmasından dolayı, ortak kod sahipliğini tam anlamıyla uygulamak ancak birbirine benzer modüllerde mümkün olmaktadır.

9. Sürekli Entegrasyon: Her iş parçası bittiğinde sistem entegrasyonun yapılmasıdır. Bu amaçla uygulamada, sürüm kontrol araçları kullanılmıştır. Her iş parçası için sürüm kontrol ortamında, ana gövdeden bir dal çıkarılmakta, yeni kod bu dal üzerinde yazılmakta ve kodlama bittikten sonra bu dal ana gövde ile birleştirilmektedir. Birleşmeden önce ve sonra birim testler uygulanmaktadır.

10. Haftada 45 Saat Çalışma: Bu prensip, haftada 45 saatten fazla çalışılmamasını ve üst üste iki haftadan fazla mesaiye kalınmamasını öngörmektedir. Uygulamada iterasyon bazında çalışılması ve proje planlarında sık sık değişiklikler yapılması nedeniyle, proje bitimi için bir tarih belirlemek oldukça zor olmaktadır. Eğer bu tarih, müşteri ve pazar baskısı sonucu belirlenmiş ise fazla mesai yapmak zorunluluk haline gelmektedir.

11. Müşteri ile Beraber Çalışma: Ürünü kullanacak müşterinin bizzat yazılım geliştiren takımda yer alarak ürünün gelişim sürecine katkıda bulunmasıdır. Böylece ürün geliştirme ile müşteri arasındaki geri besleme süresi minimuma inmektedir. Bazı projelerde müşteri temsilcisinin bulunması sağlanabilirken, bazı yazılımcıların daha önce şebeke operatörü olan şirketlerde ürün yöneticisi olarak çalışmış olması, müşteri isteklerinin ürüne yansıtılması konusunda yararlı olmuştur.

12. Kodlama Standartları: Yazılım geliştiricilerin belli kodlama kurallarına uygun kodlama yapmasıdır. Uygulamada, belirlenen kodlama kurallarına genelde uyulmakta, kurallara uymayanlar uyarılmaktadır. Bu hususta önemli olan, kodlama kurallarının çok ayrıntılı olmaması ve uyulabilir olmasıdır.

4 XP Uygulama Değerlendirme

Telekomünikasyon yazılımları geliştiren bir şirkette, XP'nin altı ayrı projede uygulanması ile elde edilen sonuçlar, proje liderleri tarafından on iki temel prensip baz alınarak değerlendirilmiştir. Bu değerlendirme sonuçları Tablo 1'de verilmiştir.

Tablo 1. XP Prensiplerinin Uygulama Sonuçları

	Çok iyi	İyi	Orta	Zayıf	Yok
1. Planlama		√			
2. Küçük Sürümler		√			
3. Metafor					√
4. Basitlik				√	
5. Test			√		
6. Yeniden Yapılandırma				√	
7. Eşli Programlama		√			
8. Ortak Kod Sahipliği			√		
9. Sürekli Entegrasyon				√	
10. Haftada 45 Saat Çalışma		√			
11. Müşteri ile Beraber Çalışma		√			
12. Kodlama Standartları			√		

Değerlendirmeler beş ayrı derecelendirmeye göre yapılmıştır. Sonuçlar incelendiği zaman bu projelerde bazı prensiplerin tam olarak uygulanmadığı görülmektedir. Tablodaki “zayıf” ve “yok” olarak işaretlenmiş olan prensipler aslında bu telekomünikasyon projelerinde XP'den sapmaları temsil etmektedir.

Ayrıca XP'de dokümantasyonun yapılması öngörülmediği halde geliştirilen sistemlerin karmaşıklığı nedeniyle Üst

Seviye Tasarım, Sürüm Planı ve İterasyon Planı gibi bazı dokümanların oluşturulması gerekliliği ortaya çıkmıştır.

XP’de kartlarla yapılması öngörülen planlamanın, proje yönetim araçları ile yerine getirilmesi XP uygulamasında farklılık yaratmaktadır.

5 Sonuç

XP’nin telekomünikasyon yazılım projelerinde uygulanması sonucunda çeşitli gözlemler ve sonuçlar ortaya çıkmıştır. Bu sonuçlar maddeler halinde şöyle değerlendirilebilir.

- Projelerin, XP için çok fazla uygun olmayan açık çalışma alanları bulunmayışı gibi bazı fiziksel engeller bulunmaktadır.
- Geliştirme yapacak kişiler, yaşamı boyunca edindiği alışkanlıklardan farklı davranışların XP tarafından beklenmesi nedeniyle direnç göstermektedirler. Ayrıca birim test kodlarının birimleri yazmadan önce yazılma gereği, yazılımcılara oldukça aykırı gelmektedir. Yazılımcılar birimleri yazdıktan sonra testleri yazma eğiliminde olmaktadır.
- XP’nin en önemli parçalarından olan birim testler konusunda bilgi birikimi olmayışı nedeniyle, birim testlerin yazımı yavaş gelişmektedir.
- Sürekli entegrasyon için gerekli platformun hazırlanma problemleri ortaya çıkmaktadır.
- Sürekli entegrasyonun istenildiği oranda yapılamaması nedeniyle yazılımcılar kodları korkusuzca değiştirememekte, böylece yeniden yapılandırma zayıf kalmaktadır.
- Bütün XP prensiplerini bir anda uygulamaya alma çabası nedeniyle, geliştirme yapanlarca büyük tepki ve dirençle karşılaşılmıştır.
- XP ye uygun planlama ve ölçüm sisteminin bulunmaması zorluk yaratmaktadır.
- XP’nin ofis otomasyon sistemleri ile olan uyuşum problemlerini aşmada zorluklar yaşanmaktadır. Bu sorunun aşılması için şirket içi iş takip sistemi kullanılarak, XP’nin ana değerlerinden “iletişim” önemli ölçüde arttırılmıştır. Bu sistem ile projeler içindeki görev atama, görev süresi ölçümü, görev doküman kayıtları, çeşitli hazır formlarla kayıt tutma, proje izleme, hata raporlama gibi iç iletişim unsurları geliştirilmeye çalışılmıştır.

Kaynakça

1. Beck, K., Extreme Programming Explained: Embrace Change, Addison Wesley, 2000, ISBN: 0201616416.
2. Jeffries, R., Anderson, A., Hendrickson, C., Extreme Programming Installed, Addison Wesley, 2001, ISBN: 0201708426.