Dağıtık ve Eşzamanlı Yazılım Geliştirme Yöntemi

Ökkeş Emin Balçiçek, Hüseyin Çağrı Çıraklı
Kuveyt Türk Katılım Bankası, Ar-Ge Merkezi, Konya
emin.balcicek@kuveytturk.com.tr , huseyin.cirakli@kuveytturk.com.tr

Özet

Büyük ölçekli sistemlere sahip olan şirketler, var olan iş gereksinimlerinin karmaşıklığı, iş ihtiyaçlarının ve teknolojinin hızlı değişimi ve kullanıcı sayısının çokluğu nedeni ile oluşan performans ihtiyacı sebebi ile daha sistematik, daha hızlı ve optimize yazılım geliştime yöntemlerine ihtiyaç duymaktadırlar. Kurumlar söz konusu ihtiyacı "yazılım üretim bandı" vasıtası ile karşılamaktadırlar. Bu bandın oluşturulmasındaki en temel gereksinimlerden birisi de kurumsal mimarinin kurulmasıdır. Bu bildiride söz konusu ihtiyaca daha iyi bir çözüm olabilecek "çoklu yazılım üretim bandı" içeren eş zamanlı ve dağıtık yazılım geliştirme yapılabilen bir yöntem önerilecektir. Bu yöntem için kurumların, kurumsal yazılım mimarilerini eş zamanlı yazılım geliştirmeye uygun olarak kurmaları gerekmektedir. Bu sayede söz konusu sistemler için farklı lokasyonlarda yazılım geliştirebilmek mümkün olabilecektir. Her ne kadar mimarisi bu şekilde olan sistemler için farklı lokasyonlar dahilinde üretim yapılmak mümkün olsa da, söz konusu lokasyondan bağımsız üretim ülkemizde yeni uygulanmaya başlanmıştır. Yapılan çalışmada; yazılım geliştirmenin en önemli kalemi olan insan kaynağını ele alarak lokasyondan bağımsız, dağıtık, eş zamanlı üretim yapan yazılım geliştirme yöntemi üzerinde durulmuştur.

Anahtar Sözcükler: Uzaktan Yazılım Geliştirme, Eşzamanlı Yazılım Geliştirme, Dağıtık Yazılım Geliştirme

Distributed and Synchronized Software Development Methodology

Abstract

Companies with large-scale systems need more systematic, quick and optimized software development methods by reason of the complexity and rapid change of business requirements, improvement on technology and performance needs depending on the number of users. Institutions meet the requirements via "Software Production Line". One of the most fundamental requirement at forming of this line is the establishment of the enterprise architecture. In this paper, a methodology will be proposed to find a better solution for the aforementioned need and carrying out the applicable, synchronized and distributed software development which is containing "multi software product lines". For this methodology, institutions must establish their enterprise architectures according to synchronized software development. In this way, software development may be possible at different locations for the systems. Although the systems has an architecture that are able to make production at different locations, software development with independent of location has been initiated in our country recently. In this study, we focus on a software methodology that is distributed, synchronized and independent from location by considering the human resource which is the most important item of software development.

Keywords: Remote Software Development, Synchronized Software Development,Distributed Software Development

1.Giriş

Hızlı değişen ihtiyaçlar ve teknoloji devrimi bütün firmaları gün be gün daha fazla, daha kaliteli, daha teknolojik ve daha kullanılabilir yazılım geliştirme mecburiyetine itmektedirler. Bu nedenle yazılım geliştirme işi zamanla artmaktadır. Yazılım geliştirme süreci konu hakkında eğitimli ve nitelikli iş gücüne ihtiyaç duymaktadır. Piyasa şartlarının iyi olduğu zamanlarda ekonominin iyi olduğu ülkelerde özellikle büyük şehirlerde bu nitelikli iş gücünü kontrol etmek, işgücü devrini kontrol altında tutmak kişi maliyetlerini belli standartlarda tutmak oldukça zordur. Bu nedenlerden dolayı yazılım geliştirme

işleminin lokasyon bağımsız yapılabilmesi kurumlar için oldukça güçlü bir stratejik avantajdır. Ofis ve insan kaynağı maliyetini azaltabilmek, insan kaynağı adına cazibe merkezi olabilmek ve işgücü devrini yönetebilmek için bir dağıtık uygulanabilir yazılım geliştirme yöntemine ihtiyaç vardır. Bu yöntem sayesinde ulusal ve uluslarası lokasyonlarda farklı şehirlerde nitelikli ve uygun fiyata çalışabilecek iş gücüne erişerek yazılım geliştirme yapılabilmektedir.

Uzaktan yazılım geliştirme yönteminin uygulanması durumunda şirketlerin elde edeceklerinin yazılım sektörü için ciddi kazanımlar olduğu aşikardır. Fakat bunun yanında uygulanabilirliği konusunda bazı sorunlarla karşılaşılmakta ve bu sorunların yönetimi hususunda farklı yaklaşımlar uygulanmaktadır. Söz konusu yaklaşımlar ulusal ve uluslararası olarak iki farklı durum için oluşturulabilmektedir. Uluslarası düzeyde ve farklı zaman dilimlerinde bulunan lokasyonlar için uzaktan yazılım geliştirme yöntemlerinden domain tabanlı bir yaklaşım daha uygun gözükmektedir. Bunun sebebi ise yazılımcı veya analistlerin birbirleri ile olan iletişimlerinden kaynaklı problemleri minimize etmesidir.[1]

Ulusal ve aynı zaman bulunan farklı lokasyonlarda ise belirli bir domainden bağımsız olarak yazılım geliştirilebilmektedir. Uygun yazılım geliştirme yöntemleri ve insan kaynaklarının iletişimden kaynaklı kayıplarının önlenmesi için kullanılan iletişim araçlarının uygunluğu bunu desteklemektedir.

2. Dağıtık Yazılım Geliştirme Yönteminin İhtiyaçları

Kurumsal firmalarda yazılım geliştirme işi bir üretim bandı yaklaşımı gerektirmektedir. Üretim bandının olmazsa olmazı, en büyük yapı taşı kuruma ait bir kurumsal yazılım mimarisinin oluşturulması gerekliliğidir. Bu mimari sayesinde büyük resim ortada olacak, yazılım geliştirme standartları, kuralları ve kütüphaneleri sayesinde bir üretim bandının temelleri atılmış olacaktır. Bu nedenle bu yöntemin ilk temel yapı taşı kurumsal yazılım mimarisi olarak adlandırılmalıdır.

Kurumsal Yazılım Mimarisi; Kurumda kullanılan bütün teknolojilerin tanımını ve nasıl kullanılması gerektiğini içermelidir. Bunun yanında IT üzerinden yapılan bütün business process design bu kurumsal mimari içerisinde resmedilmelidir. Bu mimaride nasıl yazılım geliştirilmesi gerektiği, yazılım geliştirme rolleri, araçları yani yazılım geliştirme yaşam döngüsü (YGYD) açıkça beyan edilmelidir. Bu mimarinin içerisinde sistemde bulunan ve ortak kullanılan bütün yazılım bileşenleri hazırlanmış olmalı ve yazılım geliştirme standartları açıkça ortaya konmalıdır.

Dağıtık yazılım geliştirmenin temel yapı taşlarından birisi ise lokasyonları birbirine bağlayan ağ teknolojisidir. Burada dikkat edilmesi gereken husus, bir lokasyonun üst seçilmesi ve YGYD süreci için gerekli araçların ve otomasyon altyapısının buradan host edilmesi ve yönetilmesidir. Lokasyonlar arasındaki bağlantı hızı kod paylaşım sistemlerinin, otomasyon yazılımlarının ve aradaki video konferans sisteminin ihtiyaç duyduğu hız kadardır. Günümüz teknolojilerinde yüz yüze görüşmek video konferans teknolojileri sayesinde dağıtık ekipler arasında da devam edebilmektedir. Bu da yan yana olmak ihtiyacını ortadan kaldırmaktadır.

Uzaktan iletişimin maliyetlerinin azaltılması noktasında iletişim araçlarının doğru ve yerinde kullanımı büyük önem arz etmektedir. Bilindiği üzere gerek uzaktan analiz ve gerek farklı lokasyonlarda yazılımcılar için bilgi alış verişi yazılım geliştirme projelerinin en önemli aktivatörüdür. Doğru kullanılan iletişim araçları bilgi alış verişi yapılması safhasında zamanın efektif kullanımını sağlamakta ve bu durum insan kaynağı maliyetini önemli ölçüde azaltmaktadır. Bu hususta İletişim araçlarını iki ana başlık altında toplayabiliriz bunlar; asenkron ve senkron araçlardır. Asenkron araçlar için başlıca örnekler Ağ paylaşımları ve mailler'dir. Senkron iletişim araçları olarak ise video konferans ve ekran paylaşımı gerçekleştiren toollar bilgi alış verişi açısından en verim alınan araçlardır. Buna ek olarak kurumsal firmalarda Bilgi Teknoloji Departmanının yazılım geliştirme yaşam döngüsü (YGYD) süreçleri içerisinde genelde kullanılan araçlardan: yazılım geliştirme uygulaması(Integrated Development Platform), kaynak kodu ve konfigürasyon yönetimi uygulaması , analiz girdilerinin oluşturulması ve kalite testleri yönetimi için kalite kontrol uygulaması ve proje yönetimi için ise güçlü bir proje yönetimi otomasyonu kullanılmalıdır.

Diğer bir önemli sorun ise iş ihtiyaçlarına nasıl erişileceğidir. İş ihtiyaçlarını iç ve dış müşteriden toplayan kişilere Sistem Analisti denir. Varolan bir iş ihtiyacını uygulama olarak karşılamak, hali

hazırda işleyen iş sürecini daha iyi işletebilmek adına yapılacak uygulama geliştirmeleri için gerekli ihtiyaçların tespitinde birebir görüşmenin önemi çok fazladır. Talep edilen yazılıma göre bu süre değişebilir. Uzaktan yazılım geliştirme için birebir görüşme ihtiyacı, senkron ve asenkron iletişim araçları ile sağlanabilmektedir. Fakat daha önce yapılan tecrübeler ve araştırmalar sonucu projelerin başlangıcında proje ekibinin ve iş birimlerinin yüz yüze bir etkileşimde bulunması kişiler arası güven sorununu ortadan kaldırmaktadır.[2]

3.Dağıtık Yazılım Geliştirme sürecinde oluşabilecek sorunlar ve çözümleri

Dağıtık yazılım geliştirme süreçlerinde karşılaşılan sorunları yapılan araştırmalar neticesinde 7 ana başlık altında toplamak mümkündür. Bunlar iletişim, kordinasyon, kontrol, bilgi birikimi yönetimi, teknik, işlem ve alt yapı problemlerdir[1].

Bu sorunlar, farklı zaman dilimde bulunan lokasyonlar için karşılaşılaşılacak problemlerdir. Fakat aynı lokasyon içerisinde olan ve mimarisi güçlü sistemlerde teknik ve altyapı problemleri kolaylıkla aşılabilmektedir. Bunun dışında kalan problemler değerlendirilecek olunursa;

i) İletişim Problemleri ve Çözümleri

Bir yazılım geliştirme sürecinde oluşabilecek risklerin azaltılması ve sürecin daha hızlı ilerlemesi için bilgi alış-verişinin net olması çok önemli bir husustur. Projedeki paydaşların iletişimi projenin başarılı olmasının temel unsurudur. Dağıtık yazılım geliştirmede paydaşların aşması gereken bu büyük sorun için en elverişili çözüm senkron iletişim araçlarıdır. Bu hususta oluşan genel görüşe göre, senkron iletişim araçları farklı lokasyonlarda bulunan insanların birbiri ile olan etkileşimini arttırır [3],[4]. Fakat genelleme dışında kalan bazı görüşlere göre gerek toplantı vakitlerinin kordine edilmesinin zorluğu ve gerekse kaliteli senkron iletişim araçlarının maliyetinin fazlalığı sebebi ile senkron iletişimin az tercih edildiği söylenmektedir [5]. Halbuki senkron iletişim teknolojilerinin giderek gelişmesi, gerek maliyet gerekse kullanım kolaylığı açısından uzaktan bilgi aktarımında kullanıcılar tarafından benimsenmesine sebep olmuştur.[6]

ii) Kordinasyon Problemleri ve Çözümleri

Yazılım geliştirme süreçlerinde proje takımındaki motivasyon eksikliği kordinasyonun yönetimindeki en büyük engel olarak gözükmektedir. Bu noktada proje yöneticilerinin proje çalışanları ile sürekli iletişim halinde bulunabilmesi bu sorunun aşılmasını kolaylaştırıcı niteliktedir. Senkron iletişim araçlarının kullanımı ile, aynı saat diliminde bulunan paydaşlar için proje yöneticisinin anlık müdahalelerde bulunması kolaylaşmaktadır. Bu sayede proje ekibinin dinamik ve motive olarak çalışması sağlanabilmektedir.

iii) Kontrol Problemleri ve Çözümleri

Yazılım geliştirme sürecinde kaliteli bir yazılım oluşturmak kadar geliştirmenin zamanında tamamlanması da önemlidir. Bu noktada geliştirmenin tamamlanmasına engel olabilecek tıkanmaların yönetilmesi ve anlık olarak müdahale edilmesi sürecin zamanında bitirilmesini sağlayacak yegane unsurdur. Senkron iletişim araçları söz konusu sorunları gidermeyi kolaylaştırmaktadır. Büyük ve yerinde müdahale gerektiren sorunlarda ise mesafenin uzun olmaması sebebi ile ulaşım kolaylığı eşzamanlı yazılım geliştirmenin bir diğer artısı olmaktadır.

iv) Bilgi Yönetim Problemleri ve Çözümleri

Genel bir görüşe göre senkron iletişimin farklı lokasyonlardaki kaynakların öğrenmelerine duygusal etkileşim katmaktadır.[7]Bu durum projenin ilerlemesine engel olabilecek kurumsal yavaşlığı ortadan kaldırmaktadır. Araştırmalarda proje içerisindeki iletişimin öğrenmeyi tetiklediği ve bilgi paylaşımını kolaylaştırdığı net olarak ifade edilmiştir [8]. Her ne kadar iletişim için yüz yüze görüşmek daha etkin bir yol olarak gözükmekte ise de fazla etkileşim görev paylaşımındaki belirsizliği ve kararsızlığı beraberinde getirebilmektedir. Gerçekleşen projelerdeki gözlemlere dayanarak fazla iletişim projenin performansının düşmesine neden olmaktadır [9]. Bu açıdan bakıldığında ise her ne kadar senkron iletişim araçları kaynaklar arasındaki etkileşimi arttıracak bir yapıda olsa da karar alınmasını zorlaştırıcı düzeyde etkileşime sebep olmamaktadır.

4. Dağıtık Yazılım Geliştirmeni Avantaj ve Dezavantajları

Yazılım geliştirmenin en önemli maliyet kalemi adam/saat birimi yani insan kaynağıdır. Bu sebeple süreçlerin insan kaynağını temel alarak optimize edilmesi gerekmektedir. İnsan kaynağını verimli kullanmak, oluşabilecek zaman kayıplarının engellenmesini sağlayacaktır. Günümüzde tüm paydaşların yüz yüze iletişim kurabileceği bir platformda yazılım geliştirmenin insan kaynağı bakımından avantajlı olduğu düşünülmektedir. Fakat geleneksel bir yazılım geliştirme sürecinde yüz yüze görüşmeleri oluşturabilmek adına oluşan lokasyon değişiklikleri ilgi dağınıklığına ve dolayısı ile insan kaynağı kaybına sebep olmaktadır. Bu noktada uzaktan yazılım geliştirme süreçleri içerisinde lokasyon değişiklikleri sayısı geleneksel yazılım geliştirme süreçlerine nazaran daha az olduğu için insan kaynağı kullanımının verimini artmaktadır. Buna ek olarak geleneksel bir yazılım geliştirme ekibi içerisindeki kaynakların, karar alma adına yapmış olduğu toplantılarda oluşan etkileşimlerin fazlalığı zaman kayıplarının oluşmasının bir diğer nedenidir.

Bu minvalde dağıtık yazılım geliştirmenin avantajı, bilgi paylaşımı yapılması istenen paydaşlar ile direkt iletişime geçebilmek ve konu dışına çıkılmadan etkileşimde bulunabilmektir. Buna ek olarak anlık yazılı iletişimlerde toplantıların kayıt altında tutulması sağlanacak ve bu durum işleyişin kurumsallığını arttıracaktır. Günümüzde bir çok büyük ve kurumsal firmada Bilgi Teknoloji çalışanlarının ve iş birimlerinin olduğu yerleşkeler farklı olmaktadır. Bu sebeple toplantı yapacak olan kişilerin bir araya gelmesi ve toplantıya başlaması vakit almaktadır. Ayrıca toplantıya konu ile tam anlamıyla alakası olmayan kişiler katılabilmekte ve söz konusu katılımcılar için de zaman kaybına sebep olmaktadır. Dağıtık yazılım geliştirme bahsi geçen bu problemleri aşabilecek yapıda bir yöntemdir. Fakat bu noktada en çok dikkat edilmesi gereken husus, projenin büyüklüğü ve kapsamı göz önünde bulundurularak, projeye en uygun yöntemin tespit edilmesidir. Yöntem doğru tespit edildiği takdirde, insan kaynağının kullanımı adına geleneksel bir yazılım geliştirmeye kıyasla daha etkili sonuçlar doğurabilecektir.

Uzaktan yazılım geliştirme süreçleri göz önünde bulundurulduğunda, olumlu ve olumsuz yönleri aşağıdaki konu başlıkları altına toplanabilmektedir.

Avantajları:

- Lokasyon değişikliğinden kaynaklı zaman kayıplarının olmaması.
- Alınan kararların kayıt altına tutulmasına binaen daha kurumsal bir yaklaşım.
- Fazla iletişim kaynaklı zaman kaybının minimize edilmesi.
  Lokasyon maliyetlerinin minimize edilmesi.
  Kalifiye iş gücü bulma kolaylığı ve maliyetlerinin minimize edilmesi.
  Çalışan devinimin düşük olacağı yerlere konuşlanabilme kabiliyeti
- Devlet teşviklerinin olduğu yerlere ofis açma kabiliyeti
- Ülke yararına farklı lokasyonlarda istihdam sağlanması ve teknoloji transferi

Dezavantajları:

- Paydaşların çok olması durumunda karar alma zorluğu.
- Paydaşlar arası güven problemlerinin oluşması.
- İletişim araçlarının maliyetinin yüksek olması hissiyatının hakim olması.

Uzaktan yazılım geliştirme süreçlerinde karşılaşılabilecek zorluklar, geleneksel yazılım geliştirmeye göre çok fazla olmamakla birlikte çözümü noktasında uygulanacak bir kaç yöntem ile kolaylıkla aşılabilmektedir. Bu doğrultuda oluşabilecek olan dezavantajlar değerlendirildiğinde;

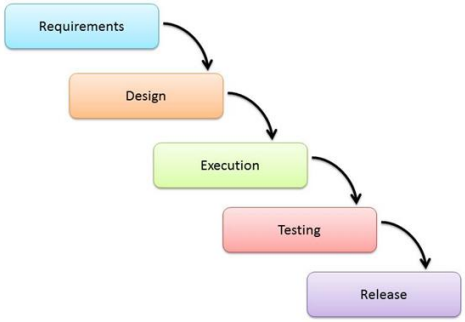
Paydaşların çok olması durumunda karar alma süreci sistem analisti tarafından yönetilerek oluşturulan bir video konferans toplantısı ile sağlanabilmektedir. Paydaşlar arası oluşabilecek olan güven probleminin aşılması adına ilk analiz toplantısının yerinde yapılması doğru bir karar olacaktır. Söz konusu toplantıda analistin özellikle iş birimi ile teması çok önemlidir. İlk toplantıda analist, iş birimi ile iletişiminde, iş biriminin kendisinin müşterisi olduğunu unutmamalıdır. İletişim araçlarının maliyetinin yüksek olması noktasında internet ağı üzerinden yapılan telefon görüşmeleri ve video konfernsların kullanılması söz konusu maliyetleri düşürmektedir. Buna ek olarak söz konusu iletişim araçlarının kullanımı yer değiştirilmeden kaynaklı zaman kaybını engellemektedir.

5. Dağıtık Yazılım Geliştirmeye Uygun Yöntemler

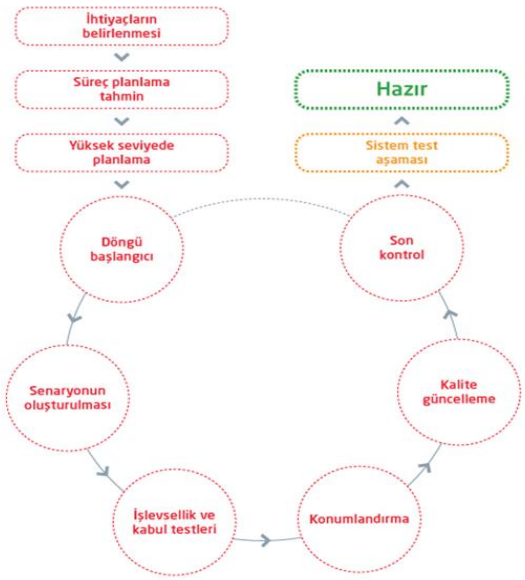
Yazılım geliştime içerisinde taleplerin toplanması ve sistem analizinin yapılması süreçleri kişiler arasındaki iletişimin en yoğun olduğu safhalardır. Bu noktada dağıtık yazılım geliştirme süreçleri içerisinde analiz safhasının yerinden mi yoksa uzaktan mı yapılması gerekliliği hep tartışılmaktadır. Analiz safhası dağıtık yazılım geliştime içerisinde yönetilebilmesi adına iki farklı durum altında değerlendirilebilir. Birincisi, yazılım geliştirme talebinin etki ettiği iş birimin veya müşterinin çok sayıda olması durumudur. Bu durumda daha kesin ve hızlı kararlar almak adına yüz yüze görüşmeleri destekleyecek toplantıların düzenlenmesi analiz süreçlerine dahil edilerek çözümlenmektedir. Diğeri ise, birinci önermenin tersine etki ettiği iş birimin veya müşterinin az sayıda olduğu dolayısı ile daha kolay kontrol edilebildiği durumdur. Bu durumda ise lokasyondan bağımsız olarak video konferans sistemlerinin kullanılması yeterli olacaktır. Bu noktada uzaktan çalışmanın analizin hazırlanması safhasındaki en önemli avantajı olan yazılı iletişimin aktif ve etkin kullanılması, oldukça kurumsal bir sürecin çalışmasına sağlamakta ve alınan kararların değişikliğinden kaynaklı proje risklerini azaltmaktadır.

Dağıtık yazılım geliştirme için uygulanacak olan sistemin geleneksel yazılım geliştirme metodolojilerinin tümünü kapsar nitelikte olması gerekmektedir. Örneğin, Şelale (WaterFall) tipi yazılım geliştime metodolojisi dağıtık yazılım geliştime için uygulanabilmektir. Çünkü her adım bir basamak halinde olduğu için analizler yerinde yapılıp sonrasında lokasyonlara dağıtılabilir. Prototip, Spiral veya Agile geliştirme metodolojileri de bu sistemde kullanılabilir çünkü senkron ve asenkron iletişim araçları sürekli geliştirmelere uygun zemin hazırlamaktadır. Bu noktada en uç örnek olarak hep sorulan çevik (agile) yöntemlerden olan Scrum içerisinde bulunan ikili programlama(pair programming) nasıl yapılabilir sorusunun cevabı basit bir anlık message sistemi üzerinde dahi bulunan masaüstü paylaşımı, video konferans veya sesli arama fonksiyonları ile yapılabilmektedir.

Dağıtık yazılım geliştirmenin, tüm modelleri kapsayarak, en iyi şekilde çalışabilmesi için ihtiyaçların oldukça formal bir halde dokümante edilmesi ve otomasyona dahil edilmesi gerekmektedir. yazılım geliştirme sürecinin analiz safasında,bilgi paylaşımı çok fazla olduğu için geliştirme, değişiklik ve hata düzeltme işlemine (bugfixe) nazaran daha fazla zorlanılabilir.



Şekil 1. Waterfall Uygulama Geliştirme Metodu



Şekil 2. Çevik (Agile) Uygulama Geliştirme Metodu

6 .Uygulama Örnekleri

Gelişen teknolojinin ürünleri ile birlikte uzaktan yazılım geliştirme yapmanın kolaylaşmakta olduğu ve yüzyüze iletişim ile elde edilen kazanımların uzaktan iletişim ile de sağlanabilir durumda olduğu yukarıda belirtilmişti. Fakat bazı durumlarda taleplerin anlaşılabilirliğinin artması ve sosyal etkileşime bağlı olarak güven ortamının sağlanması önem arz etmektedir. Bu nedenle büyük projeler için YGYD'nin en temel safhalarından taleplerin tespiti sürecinde müşteri ile yüz yüze görüşülerek taleplerin alınması daha etkili olabilmektedir. Bu

doğrultuda analistin uzaktan yazılım geliştirmedeki yerini iki başlık altında değerlendirebiliriz. İlki analist ile müşterinin aynı lokasyon içerisinde bulunarak sürekli yüzyüze etkileşim içerisinde olması ve yazılım ekibinin ise toplu olarak farklı bir lokasyon içerisinde olması durumudur. Bu durum içerisinde yazılım geliştirme metodolojisi için en uygun olan yöntem Şelale (WaterFall) metodudur. Çünkü Şelale metodunda yapılması gereken uzun süreli yüzyüze görüşmeler ile taleplerin netleşmesi ve buna binaen yazılım ekibine nihai bir analiz sunulması gerçekleşmiş olacaktır. Böylece netleştirilmiş bir analiz, sistemin anlaşılamamasından dolayı oluşacak riskleri azaltmakta ve yazılımcıya aktarılmasında dolayı oluşacak verim kayıplarını ortadan kaldıracaktır. İkincisi ise, analistin ve yazılım ekibinin toplu olarak aynı lokasyon içerisinde bulunması, müşterinin ise bunlardan farklı bir lokasyon içerisinde bulunması durumudur. Bu durum için yazılım geliştirme metodolojisi olarak en uygun yöntem Çevik (Agile) yöntemidir. Çünkü Çevik metodunda oluşturulan prototip üzerinden ihtiyaçların tespit edilmesi taleplerin yazılım sürecinde verimsiz insan kaynağı kullanımı riskini azaltmaktadır. Fakat bu durumda daha önce de bahsedildiği üzere özellikle büyük çapta bir yazılım geliştirme projesinde kısa süreli bile olsa yüzyüze iletişim sağlanması işin geniş perspektifte algılanmasını ciddi oranda kolaylaştıracak, aynı zamanda güven ortamının oluşmasını sağlayacaktır. Bundan sonraki safhalarda uzaktan iletişim sağlayan araçların kullanımı büyük önem arz etmektedir. Asenkron ve senkron iletişim araçlarının efektif kullanımı sayesinde analiz içerisindeki eksikliklikler kademe kademe giderilmekte ve yazılı iletişim sayesinde talepler kesin hatları ile belirlenmektedir böylelikle müşteri memnuniyeti artmaktadır.

Analiz safhasının iş birimlerinden farklı lokasyonda yapılması durumunu daha detaylı incelemek adın üç farklı senaryo değerlendirmesi bulunmaktadır. Bunlar;

1) *Bir iş sürecinin ilk defa yeni bir uygulama üzerinden çalıştırılması*

İlk defa oluşturulacak bir sistemde iş birimin taleplerini netleştirmesi ve bu talepleri yazılı olarak kayıt altına alması önemli bir hususdur. Talepler detaylı olarak dökümante edildikten sonra analist ile paylaşılması analiz toplantıları öncesi analistin hazırlıklı olmasını sağlayacaktır. Soruların önceden belirlenmiş olması toplantıları verimini ciddi oranda arttıran bir durumdur.

İlk defa kurulacak olan sistemlerde İş biriminin taleplerini netleşmesi süreci vakit almaktadır. Bu noktada analistin iş birimin süreçlerini tam anlamı ile anlaması ve yönlendirmesi gerekmektedir. Süreçlerin tam olarak anlaşılması noktasında sürekli olarak senkron görüşmelerin yapılması çok doğru olacaktır. İş biriminin yönlendirilmesi ise taleplerin prototipinin daimi olarak paylaşılması ile sağlanabilecektir.

2) *Mevcutta bulunan bir sistemin yeni bir sisteme dönüşmesi*

Mevcutta bulunan bir sistemin dönüştürülmesi sürecinde eski ekranların ve sistemin irdelenmesi en kritik nokta olup, iş birimine iletilecek soruların tespitini kolaylaştırılacaktır. Söz konusu tipte yazılım geliştirmelerde eski sistemin tam anlamı ile irdelenmesi, doğru ve yerinde soruların oluşturuması halinde diğer yazılım geliştirme tiplerine göre daha az iletişim ile sağlanabilmektedir. Buna ek olarak dönüşüm projelerinde senkron iletişim araçları kadar asenkron iletişim araçları da kullanılabilmektedir.

.

3) *Hali hazırda bir uygulama üzerinden çalışan bir sürecin yeni gereksinimlerle dönüştürülmesi ve kısmen dönüşmüş daha önceki sistemlere entegrasyonu*

Entegrasyon projelerinin analizinin oluşturulma safhasındaki dikkat edilmesi gereken en önemli nokta halihazırda çalışan sistemin analisti veya söz konusu servisin yöneticisi ile iş biriminin talepler ile çalışan sistem ortak sürecinin oluşturulmasıdır. Bu noktadan sonra ise çalışan sistemin iyi analiz edilmesi ve ekranların işleyişinin tam olarak anlaşılması önem arz etmektedir. Analiz süreci boyunca analistin gerek çalışan sistem sahipleri gerek ise iş birimi ile sürekli temas halinde olması gerekmektedir. Özellikle çalışan sistemin iyi analiz edilmesi ve işbirimine entegrasyon prototipinin aktarılması adına, ekran paylaşımlarının etkili olarak kullanılması önemlidir.

7.Sonuç

Teknolojinin gelişmesi ile gelinen nokta gösteriyor ki, yeni iletişim altyapısı ve maliyetleri eşzamanlı ve dağıtık yazılım geliştirme için oldukça uygun imkan ve olanak sağlamaktadır. Ayrıca farklı lokasyonlarda olmak hem ofis maliyetlerini azaltmakta hem de daha kalifiye ve daha düşük maliyetli insan kaynağından yararlanılabilmektedir. Bunun sebebi de tek bir lokasyona bağlı kalmayarak ilgili lokasyonun ekonomik ve insan kaynağı dezavantajlarını minimize etmektir. Bütün bunların yanında bu yöntem, bazı bölgelerdeki devlet teşviklerinden faydalanmak imkanı sunmakla birlikte istihamın ve bilgi birikiminin yurdun dört bir yanına yayılmasında öncü olma fırsatı sağlamaktadır.

Bugüne kadar uygulanan tüm yazılım geliştirme metodları, farkı zaman diliminde olmayan ulusal veya uluslararası konumlarda çalışmak için uygundur. Bu durumu sağlayan aynı zaman diliminde yaşandığı için farklı lokasyonların etkileşim ihtiyacı olduğunda eş zamanlı olarak iletişim kurulabilmekte ve adeta aynı lokasyonda çalışıyormuşcasına kolayca yol alınabilmektedir.

Tüm bu kapsamda, uzaktan yazılım geliştirme sürecinde, projenin kapsamının belirlenmesinin ardından, yazılım ve analiz süreçlerinde uygulanacak olan metodun, iletişime geçilecek kişilerin ve kullanılacak olan iletişim araçlarının doğru tespiti en kritik başlıklardır.

8. Kaynaklar

[1] Tufekci O.,Cetin S.,Arifoglu A.,"Proposing a Federated Approach to Global Software Development",**Fourth International Conference on Digital Society (2010).**

[2] Sepulveda C., "Agile Development and Remote Teams:Learning to Love the Phone",**Proceedings of the Agile Development Conference (2003).**

[3] Lobel M.,Neubauer M.,Swedburg R.,"Elements of group interaction in a real-time synchronous online learning-by-doing classroom without F2F participation",**USDLA Journal,16(2002).**

[4] Locatis C.,Fontelo P.,Sneiderman C.,Ackerman M.,Uijtdehaage S.,Candler C., "Webcasting Videoconferences over IP:A synchronous communucation experiment." **Journal of the American Medical Informatics Association(2003).**

[5] Burnet C., "Learning to chat: Tutor participation in synchronous online chat." **Teaching in Higher Education(2003).**

[6] Serce F.,Kathleen S.,Alparslan F.,Brazile R.,Dafoulas G.,Lopez V., "Online Collaboration: Collaborative behavior patterns and factors affecting globally distributed team performance.",**Computers in Human Behavior(2011)**

[7] Chou C., "A comparative content analysis of student interaction in synchronous and asynchronous learning networks.",**In Proceedings of the 35th annual Hawaii international conference on system sciences (2002)**

[8] Dillenbourg P., "Virtual learning environments." **In Workshop on virtual learning environmentsof the EUN conference: Learning in the new millenium: Building new education strategies for school(2000).**

[9] Swigger K.,Hoyt M.,Serçe F.,Lopez V.,Alpaslan F., "The temporal communication behaviors of global software development." **Computers in Human Behavior(2012).**