

Çoklu Bileşenlerden Oluşan Sistemlerde Çevik Yazılım Geliştirme Deneyimi

Hilal Coşkun¹, İbrahim Doğru¹

¹ Netaş Telekomünikasyon A.Ş., İstanbul, Türkiye
{hkaptan, idogru}@netas.com.tr

Özet. Büyük ve karmaşık ürünler geliştiren organizasyonlarda proje ve süreç yönetimi normalden daha zorlu ve vakit harcayan bir iştir. Farklı iş konuları üzerinde çalışan birimlerin bir araya gelerek, tek bir amaca hizmet etmesi ve başarıya ulaşması, ortak bir çalışma platformunun sağlanması, bu platformun iş birliği içerisinde yürütülen sürecin en iyi şekilde yönetilmesi ile mümkün olmaktadır. Deneyim sonuçlarını paylaşacağımız NETAŞ Akıllı Ofis Uygulamaları Projesi, bahsettiğimiz karmaşık sistem tanımına uygun olup, bu süreçte çevik yazılım süreç yönetiminden faydalanılmıştır. Süreç yönetiminin eksik kaldığı noktalarda (proje yönetimine kıyasla çoklu bileşenli çözüm yönetimi) ihtiyaçlar doğrultusunda uyarlamalar yapılmış ve bu uyarlamaların projeye etkisi/faydası paylaşılmıştır.

Anahtar Kelimeler: Çevik Yazılım, Çoklu Bileşenlerden Oluşan Sistemler, Yazılım Proje Yönetimi, Akıllı Ofis Uygulamaları

1 Giriş

Uluslararası endüstride ortak işbirlikçileri ve müşterileri bulunan NETAŞ 'ın rekabet gücünün arttığı günümüz dünyasında bu işbirliklerini sürdürebilmesi, yazılım ihracat şampiyonlukları kazanması, zamanında ve kaliteli işler ortaya koymasıyla mümkün olmaktadır. Yakın geçmişe kadar Waterfall[1] süreçleri ile projelerini başarılı bir şekilde yürüten NETAŞ, günümüzün hızlı ve dinamik teknolojik gelişmelerine ayak uydurabilmek, müşteri gereksinimlerine daha hızlı ve esnek bir şekilde cevap verebilmeyi amaçlamaktadır. Bu amaç doğrultusunda çevik yazılım süreçlerini uygulamaya başlamış ve bu proje ile birlikte bir adım daha ileri giderek çoklu bileşenli sistemler için de adapte etmiştir. Bu sayede müşteri ile işbirliği yaparak daha az işgücü ve maliyetle zamanında ve beklentileri karşılayan kalite seviyesinde işler üretmektedir.

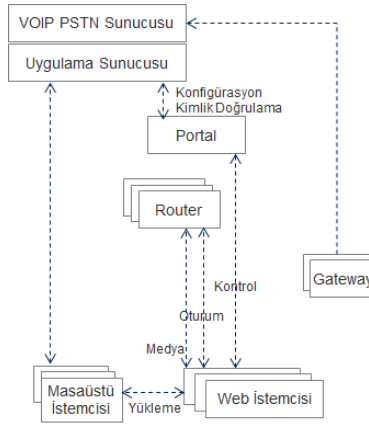
Bu çalışma kapsamında, Çevik Yazılım Geliştirmenin[1][2][3] bu karmaşık çözüm üzerinde nasıl uygulandığı açıklanacak ve uygulamanın avantajları /dezavantajlarından bahsedilecektir. Bu çalışma için CMMI-DEV v1.3 Seviye 3 sertifikasyonuna sahip olan NETAŞ 'ın Çoklu Ortam İletişim Sistemleri kapsamında yürüttüğü Akıllı Ofis Çözümü, örnek uygulama olarak ele alınarak yorumlanacak ve değerlendirilecektir.

2 Çoklu Bileşenlerden Oluşan Sistemler

Çoklu Bileşenlerden Oluşan Sistemler birden fazla ürünün birleşiminden oluşan, tüm veya bazı ürünlerin üzerinde diğer ürünlerdeki yeteneklere bağımlı veya o yetenekler ile tümleşik çalışan özelliklerin olduğu sistemlerdir. Akıllı Ofis çözümü de bu tür bir sistemdir; hem birden fazla ürün üzerinde bağımlı ve bağımsız yazılım geliştirmeleri hem de bağımlı yazılım geliştirmelerinin bütünleştirilerek teslim edilebilir bir çözüm üretilmesini kapsamaktadır.

2.1 Akıllı Ofis Çözümü

Akıllı Ofis çözümü, günümüz teknolojisi ile birlikte bant genişliklerinin artması sonucu daha da önem kazanan görüntülü konuşmanın, görüntülü konferans olarak kullanılmasına olanak sağlayıp, aynı zamanda ekran paylaşımı ve sohbet odası uygulamalarını da aynı platformda sağlamayı hedefleyen çok bileşenden oluşan yazılımların geliştirilmesi projesidir.



Şekil 1. Akıllı Ofis Çözümü Mimarisi

Bu çözüm kapsamında Şekil 1'de gösterilen birçok bileşen üzerinde geliştirmeler yapılması gerekmektedir;

- Multimedya uygulama sunucusu (EXPERiUS) geliştirmeleri
 - Sesli ve görüntülü konferans servislerinin konfigüre edilebilmeleri için gerekli olan arayüz ve arka plan geliştirmelerini kapsamaktadır.
- Alt yüklenici firma tarafından yapılan kullanıcı masaüstü istemcisi(GENCom Windows) geliştirmeleri
 - Masaüstü istemcisinden web istemcisini yükleyebilmek için gerekli geliştirmeleri ve yeni arayüz geliştirmesini kapsamaktadır.

- Kullanıcı web istemcisi (Collab Client)
 - Kullanıcıya tarayıcı üzerinden sesli ve görüntülü konferansa katılma imkanı sunacak olan istemci geliştirmesini kapsamaktadır.
- İş birlikçi firmadan sağlanan çoklu ortam video konferans sunucusunun (Portal, Router, Gateway) diğer bileşenler(VoIP PSTN[4] Sunucu vb.) ile birleşimi
- Uçtan uca çözüm fonksiyon ve kalite testleri

2.2 Çevik Yazılım Scrum Yöntemi

Çevik Yazılım kapsamında tanımlanan Scrum[2] yöntemi bir yazılım geliştirme işinin en küçük iş parçalarına bölünerek (Backlog[2]), sabit bir ekip tarafından (Scrum Ekibi[2]), süreli yazılım geliştirme ve test döngüleri(Sprint[2]) takip edilerek geliştirilmesini öğütleyen, her döngüde çalışan yazılıma ve geri besleme almaya odaklanan bir yöntemdir.

2.3 Çoklu Bileşenlerden Oluşan Sistemlerin Geliştirme Zorlukları

Ürün İş Listesinin büyüklüğüne ve ürün teslim süresine bağlı olarak bazı yazılım geliştirme projelerinde birden fazla Scrum Takımı oluşturulması ve bu takımların birbirine paralel, bağımsız veya bütünleşmiş işleri geliştirmeleri gerekebilmektedir. Çevik Yöntem uygulamalarında bu tür ihtiyaçlar için çeşitli yaklaşımlar tanımlanmıştır, bunlardan bir tanesi Scrum of Scrum[5] yaklaşımıdır. Çoklu Bileşenlerden Oluşan Sistemlerin geliştirilmesinde Scrum of Scrum yaklaşımının uygulanması faydalı görünmekte, ancak mevcut pratikleri yetersiz kalabilmektedir. Çünkü mevcut Scrum of Scrum pratikleri birden fazla Scrum takımının ortak Çevik Yazılım pratiklerini(sprint uzunluğu, sizing için kullanılan değerler) uygulamasını öngörmekte ve Scrum of Scrum rolü bu öngörüden beslenmektedir. Ancak bu öngörünün farklı ürünler üzerinde, özellikle farklı şirketlerin ürünleri üzerinde uzak ofislerde, çalışan Scrum takımları arasında gerçekleştirilmesi ciddi bir zorluktur. Çoklu Bileşenlerden Oluşan Sistemlerin geliştirilmesinde bağımsız sürüm planı olan (belki farklı şirketlere ait) birden fazla ürün üzerinde farklı Scrum takımları tarafından gerçekleştirilen yazılım geliştirmelerinin bir araya getirilerek ortak bir kullanılabilir, teslim edilebilir çıktı oluşturulması gerekmekte ve bu durum açıklanan zorlukları doğrudan karşımıza çıkarmaktadır.

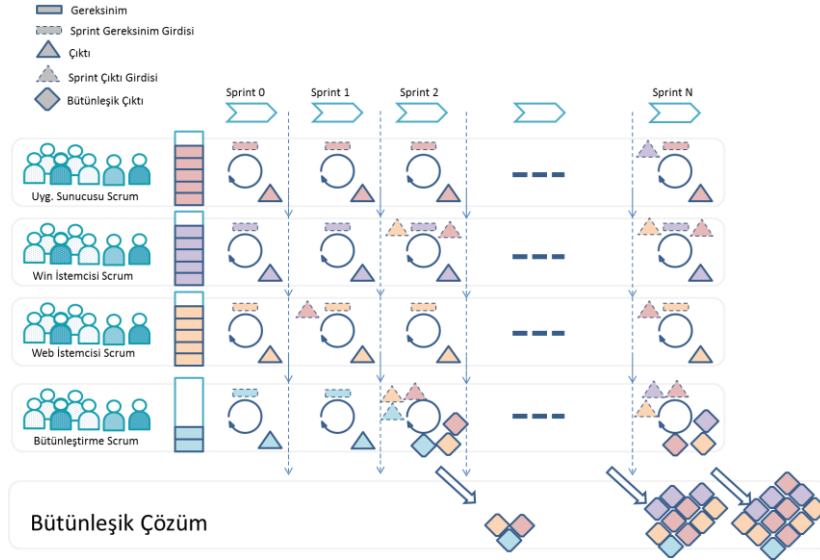
3 Çok Bileşenlerden Oluşan Sistemlerin Geliştirilmesinde Çevik Yazılım Deneyimi

Proje başından itibaren karar alınarak 2 haftalık planlanan sprintlerle birlikte bu bütünleşme noktaları yakından takip edilmiş, özellikle kullanıcı ara yüzü gibi geri besleme alınan geliştirmelerde esnek bir şekilde değişiklik isteklerine cevap verilebilmiştir.

3.1 Planlama

Bu çözüm kapsamında istemci, sunucu geliřtirmelerinin yanı sıra sisteme yeni ürünlerle bütünleřtirme geliřtirmeleri de planlanmıřtır. Sürüm bařlangıcı yapılmadan önce tüm iř birimleri için bir Hazırlık Döngüsü(Spring 0) süreci planlanmıř, bu döngü sürecinde çözüm ve yazılım mimarları, iř gereksinim listelerinin üzerinde çalıřarak Sistem Gereksinimlerini belirlemiřtir. Çözüm ve yazılım mimarları tarafından yapılan analizler sonucu bu gereksinimler dođrultusunda $\pm\%50$ yanılma oranlı bir kaynak/zaman ihtiyacı belirlenip bu kaynak/zaman bilgisi dođrultusunda gerekli kaynak profili belirlenerek kaynak, zaman ve bütçe kısıtları göz önünde bulundurularak proje planı çıkartılmıřtır.

Proje planı oluřturulurken, bu çözümün çoklu bileřenlerden oluřan bütünleřik yapısı göz önünde bulundurularak etkileřim ve bütünleřme noktaları belirlenmiř, bu noktaların olabildiđince erken dođrulanabilmeleri için de çözüm dođrulama döngüleri sürekli ve geliřtirme döngülerine paralel olacak řekilde tasarlanmıřtır.



Şekil 2. Çoklu Bileřenlerden Oluřan Sistem ve Çevik Yazılım

Proje planındaki önemli noktalar Şekil 2. Çoklu Bileřenlerden Oluřan Sistem ve Çevik Yazılım ile gösterilmiřtir. Proje planına göre;

- Toplam 4 adet Scrum Takımı oluřturulmuřtur.
- Her bir takımın görevi sorumlu olduđu(genellikle farklı bir ürün) iř listesinin tamamlanması ve çıktıların teslim olarak belirlenmiřtir.
- Bütün Scrum Takımlarının Çevik Yazılım çerçevesine uymaları zorunlu hale getirilmiřtir.
- Sistemin toplam kalitesinden bütün Scrum Takımları sorumlu hale getirilmiřtir.

- Bütün Scrum Takımları kendi 2 haftalık döngü(Sprint) süresinde çalışmayı planlamışlardır.
- Bir Scrum Takımı (örneğin Uygulama Scrum Takımı)'nın teslim edeceği çıktı bir başka Scrum Takımı (örneğin Web İstemcisi Scrum Takımı) için girdi (kendi iş listesindeki özel bir işi yapabilmesi için ihtiyaç duyacağı sonucu işi) olduğu durumlar iş listelerinde öncelikli planlanarak proje için toplam zaman ve kaynak faydasının artırılması planlanmıştır.

Projenin içeriğine özgü olarak (yeni web kullanıcı ara yüzünün geliştirilmesi, var olan masaüstü istemcisinde yeni ara yüz formatına geçiliyor olması) Genel Kullanılabilirlik ilan edilmeden önce alfa ve beta kullanıma açılması ve deneyimlenmesi fazlarının gerekliliği düşünülmüş, zaman, kaynak ve bulgu yönetimlerine etkisi göz önüne alınarak planlama safhasına bu süreçler eklenmiştir.

Projenin işleyişi ile ilgili bilgi veren anahtar metriklerin operasyon ve kurumsal seviyelere raporlanmalarının JIRA[6] izleme aracı yardımı ile tüm alt bileşenler tek bir başlık altında toplanarak yapılmasına karar verilmiştir.

GENEL KULLANILABİLİRLİK: Tasarım, geliştirme ve doğrulama fazları tamamlanan sürümlerin müşterilere sunulması için hazır hale geldiği nokta sürüm için genel kullanılabilirlik noktasıdır.

3.2 Yürütme ve Zorluklar

Yürütme Proje planına uyularak yapılmış, kalan iş ve teslim edilen iş miktarları yürütme boyunca haftalık olarak çeşitli raporlama yöntemleri kullanılarak takip edilmiştir. Proje planında iş, kaynak ve zaman değişimlerine adaptasyon gerektiği noktalarda gerekli aksiyonlar alınmıştır.

Çözümün geliştirilmesi sırasında hem yönetim hem yürütüme hem de teknik bazı zorluklarla karşılaşmıştır;

- Planlama safhasında öngörülen zaman, kaynak ve bulgu yönetiminin belirlenen kalite hedeflerini tutturmasının, birden fazla ürün üzerinde farklı Scrum pratikleri uygulayan ekipler tarafından geliştirme yapıldığında daha da zorlaştığı tecrübe edilmiştir. Çoklu Bileşenlerden Oluşan Sistemlerin geliştirilmesinde kesin hedefler yerine hedef aralığı belirlenmesinin daha isabetli olacağı tecrübe edilmiştir.
- Bir Scrum Takımının sorumlu olduğu iş listesindeki bir işin önceliğinin değişmesi (örneğin önceliğinin yükseltilmesi) başka bir Scrum Takımının sorumlu olduğu iş listesindeki bir başka işi de önceliğinin değişmesine zorlamıştır. Bu işlerin Scrum Takımı kapasitesine göre büyüklüklerinin veya Scrum Takımlarının benzer boyutlardaki işleri ölçeklendirmelerinin farklı olması değişimi planlamayı zorlaştırmıştır. Bu zorlukların ekipler arasında ortak ölçü değerleri belirlenerek bir miktar aşılabileceği gözlenmiştir.
- Projede durum takip, bulgu yönetimi, raporlama ve kalite konularının seçilen araçlar vasıtasıyla sağlanması sırasında alt yüklenici firmalarla (bütün ekipler arasında) ortak araç ve platformlara sahip olunmamasından dolayı iş faaliyetlerinin proje içerisinde işleyişi için işbirliği ve koordinasyon zorlukları yaşanmıştır.

- Kalite performans ölçümleri anlamında anahtar bir gösterge olan bulgu yönetiminin raporlanmasında üstteki sebepten dolayı manuel hazırlık yapılması gerekliliği yönetimi zorlaştıran bir durum olmuştur. Bulgu yönetimini kolaylaştırmak için bütün araçların otomatik raporlar üretebilen ve farklı ölçekli bulgu listesine sahip çoklu bileşenlerin bulgu yönetimini ortak ölçekli bir raporda birleştirebilen araçlar hale getirilmesinin gerekliliği tecrübe edilmiştir.
- Scrum Takımlarındaki Geliştirme Ekibinin sürekli ihtiyaç duyduğu bazı işlemlerin(compile, build, lab upgrade) yeterli otomasyon altyapısına sahip olmayan ürünlerde uzun sürmesi kaynak ve zaman tahminlerinin aşılmasına bağlı zorluklar getirmiştir. Bu zorlukların aşılabilmesi için yenilikçi geliştirme ortamlarının, sürekli doğrulama ortamlarının ve otomasyon altyapılarının yazılım geliştirmede kritik öneme sahip olduğu tecrübe edilmiştir.

Proje yürütmesi sırasında her ne kadar birçok zorluk ile karşılaşılsa da bütün Scrum Takımlarının değişime hızlı adaptasyon sağlayabilmesi projenin planlama aşamasında öngörülen kaynak ve zaman çizelgesi ile başarıya ulaşmasını sağlamıştır.

4 Sonuçlar

Proje sürecinde elde edilen ve benzer projelerde uygulanması tavsiye olunan değerlendirmeler şunlardır:

- Bütün Scrum Takımları tarafından ortak araçların kullanılması
- Bütün Scrum Takımları için iş boyutu belirlemede(sizing) ortak ölçü değerlerinin belirlenmesi
- Otomatik ortak raporlama ve ortak raporun ortak ölçü değerine sahip olması
- Alfa ve Beta kullanıcı test süreçleri planlanması

Teşekkür

Bu bildiriye ön inceleme yaparak ve tavsiyelerde bulunarak katkı sağlayan Oğuzhan Yavuz, Ali Yener Mutlu ve Yunus Dönmez'e katkılarından dolayı teşekkür ederiz.

Kaynaklar

1. Özvural, Ö. G., Gün, Ö., AK, E., “Etkin Bir Yazılım Süreç Yönetimi İçin Süreç Yönetim Aracı Seçimi”, 8. Ulusal Yazılım Mühendisliği Uygulamaları Sempozyumu (UYMS 2014), (2014).
2. Schwaber, K., Sutherland, J., “The Scrum Guide”, (2013).
3. Çetin, E., Durdu, P. O., “Türkiye’de Çevik Yazılım Geliştirme Üzerine Bir İnceleme”, 8. Ulusal Yazılım Mühendisliği Uygulamaları Sempozyumu (UYMS 2014), (2014).
4. Goode, B., “Voice over Internet Protocol (VoIP)”, Proceedings of the IEEE, vol. 90 (6), (2002).

5. <http://www.scaledagileframework.com>.
6. <https://www.atlassian.com/software/jira>.