

Büyük Ölçekli Bir Organizasyonda Uygulama Yaşam Döngüsü Yönetimi Uygulama Deneyimi

Yagup Macit¹, Eray Tüzün¹, Kürşat İnce¹, Ahsen İkbal AYTEKİN¹,

¹ HAVELSAN, Bilgi Sistemleri ve Yönetimi Müdürlüğü
06510 Ankara, Türkiye
{ymacit, etuzun, kince, ayergok}@havelsan.com.tr

Özetçe. Günümüz şirketleri, faaliyet alanlarından bağımsız olarak, bilişim sistemleri konusunda üçüncü kişilerin sağladığı yazılımlarla yetinmemekte, yenilikçiliği ve hızı arttırmak için kendi bilgi sistemlerini yaratmayı hedeflemektedirler. Uygulama yazılımlarının fikir aşamasından itibaren geliştirme, dağıtım ve bakım süreçlerinin tamamı Uygulama Yaşam Döngüsü Yönetimi (UYU) olarak tanımlanmaktadır. Bu çalışmada tümleşik uygulama yaşam döngüsü yönetiminin tanımı, tarihçesi ve uygulama kazanımları anlatılmıştır. Ayrıca UYU'ye geçiş sürecindeki yöntemler ve tecrübeler paylaşılmıştır.

Anahtar Kelimeler: Uygulama yaşam döngüsü yönetimi, uygulama yaşam döngüsü kazanımları, yazılım geliştirme araçları, yazılım süreçleri

1 Giriş

Günümüzde uygulama yazılım geliştirme işlevi yalnız yazılım şirketlerinin değil bütün şirketlerin temel işlevlerinden biri olduğunu gözlemlenmektedir. Uygulama yazılımlarının fikir aşamasından itibaren geliştirme, dağıtım ve bakım süreçlerinin tamamı Uygulama Yaşam Döngüsü Yönetimi (UYU) olarak tanımlanmaktadır[1]. Tümleşik çalışan bir UYU sürecinin getirdiği kazanımlar literatürde belirtilmiştir [2][3].

Bu çalışmada özellikle tümleşik UYU altyapısının yazılım organizasyonlarında kazanımları ve verimliliğe etkileri, savunma sanayinde 1000+ çalışanı ile faaliyet gösteren, HAVELSAN örneğinde ayrıntılandırılmıştır. Projeleri ortak bir UYU altyapısı üzerinde geliştirmek üzere HAVELSAN'da Yazılım Geliştirme Ortamı (YGO) projesi 2012 yılında başlatılmıştır. YGO projesi ile geliştirme ortamı ve araçları için donanım, lisans, bakım ve eğitim maliyetlerinin azaltılması, geliştirme personeli arasındaki bilgi paylaşımının artırılması hedeflenmiştir. 2014 yılı itibariyle projenin kurum çapında kullanımı ve yaygınlaştırılması devam etmektedir.

Bildirinin ikinci bölümünde Uygulama Yaşam Döngüsü yönetiminin tanımı ve faydaları özetlenmiştir. Üçüncü bölümünde UYU'nin HAVELSAN'da uygulanması, ve bu geçiş sırasında elde edilen deneyimler ve kazanımlar sunulmaktadır. Son bölümde ise çalışma genel olarak özetlenip değerlendirilerek bundan sonraki dönemde yapılacak çalışmalar kısaca gözden geçirilmektedir.

2 Uygulama Yaşam Döngüsü Yönetimi

Uygulama Yaşam Döngüsü Yönetimi (UYY), yazılım uygulamalarına ait bütünsel yaşam döngüsü bakışı ve geliştirme süreçlerinin koordinasyonunu sağlamak ve süreç çıktılarını yönetmek amacıyla ortaya çıkmıştır. UYY, yazılım uygulamalarının yüksek kalitede ve hızlı bir şekilde teslim edilebilmesi için, yaşam döngüsü süreçlerinin otomasyonunu sağlayarak, yazılım geliştirme sürecine izlenebilirlik ve güçlü raporlama yeteneği kazandırmaktadır. İlk sürümü UYY 1.0 olarak adlandırılan yaklaşım, gereksinim yönetimi, tasarım, geliştirme, inşa ve test gibi geleneksel yaşam döngüsü aktivitelerinin yönetimi için bütünleşmiş araç kümeleri olarak sunulmuştur [4]. Bu yaklaşımda her bir araç, disiplin temelli olarak uygulama yazılımlarının ilgili sürecini ele alırken, diğer süreçleri yöneten araçlarla iletişim kurabilmektedir.

Birinci nesil UYY çözümü, özellikle farklı üreticilere ait araçların birbirleri ile yaşadığı entegrasyon eksiklikleri nedeniyle, tüm süreçlerin senkronizasyonunu sağlamak ve korumak oldukça maliyet gerektirmiştir. Bu eksikliklerin getirdiği kısıtlamalardan dolayı UYY 2.0 yaklaşımına geçilmiştir [4][5]. Yeni nesil UYY yaklaşımında, farklı araç kümelerinin entegrasyonu yerine rol tabanlı olarak yaşam döngüsü süreç yönetiminin koordinasyonunu esas alınmıştır. Bu doğrultuda ortaya çıkan UYY çözümü, geliştirme süreçlerinin tümüyle entegre olduğu araçlardan oluşmaktadır. Entegre olan bu araçlar, süreçler arasında iletişim kurabilme yeteneğine sahip oldukları için sağlıklı bir izlenebilirliğin oluşturulması açısından büyük bir avantaj sağlamıştır. Öte yandan, bütün disiplinlerde sunulan ortak özellikler ve ortak hizmetler sayesinde şeffaf bir şekilde çıktılarının üretilmesi ve yönetilmesi yapılabilmektedir. Tümüyle entegre olan bu çözümlerin yazılım geliştiricilere, anlık ve doğru şekilde raporlama, doğru ve kaliteli yazılım üretme, verimliliği artırma ve hızlı teslimat gibi faydalar kazandırdığı görülmüştür[2][6][7][8].

3 HAVELSAN Deneyimi

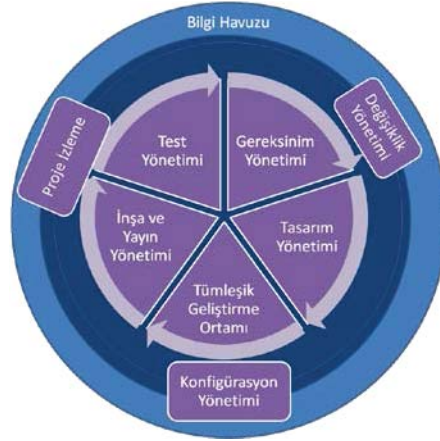
3.1 Motivasyon

HAVELSAN'da Matris organizasyon yapısının uygulanmaya alınması ile birlikte faaliyetler, Program Grup Müdürlükleri ve Mühendislik Grup Müdürlükleri olarak iki farklı müdürlüğün yönetiminde gerçekleştirilmiştir. Bu yapıda çalışanlar birden fazla projede görev almak durumundaydılar. Proje yapısından matris yapısına geçişteki süreçte, her proje için ayrı ayrı konumlandırılan geliştirme ortamlarının iyileştirilmesi gerektiği tespit edilmiştir. Geliştirme ortamlarındaki temel sorunlar;

- Proje geliştirme ortamında kullanılan araçların farklılığı ve birbirleri ile tümleşik çalışmaması,
- Belirli bir araç setinin olmadığı durumlarda süreçlerin elle işletilmesi,
- Her proje ortamı için personele yeni eğitimler verme gerekliliği,
- Müşteriye yapılan teslimatlar da dokümantasyonun çoğunlukta olması ve bu dokümantasyonların hazırlanma sürecinin fazla zaman alması,
- Proje geliştirme ortamları için sunucu ve yazılım tedarikinde karşılaşılan problemler sonucunda geliştirme faaliyetlerine başlanamaması, tedarik

- edildiğinde ise personelin bu donanım ve yazılımın bakım ve idamesi ile görevlendirilmesi,
- Projede üretilen bilgi varlıklarının birbirleri ile olan izlenebilirliğinin sağlanamaması,
- şeklinde özetlenebilir.

Yukarıdaki problemlerin çözümüne odaklanıldığında geliştirme ortamını ortaklaştırmak, standartlaştırmak ve her seviyedeki izlenebilirliği, erişilebilirliği ve verimliliği arttırmak için 2012 yılı başında YGO projesi tanımlanmış ve çalışmalara başlanmıştır.



Şekil 1. Uygulama Yaşam Döngüsü Yönetimi

HAVELSAN YGO Projesi Şekil 1 de görüldüğü üzere: Gereksinim Yönetimi, Tasarım Yönetimi, Konfigürasyon Yönetimi, Değişiklik Yönetimi, İnşa ve Yayın Yönetimi, Test Yönetimi, Proje İzleme ve Bilgi paylaşımı fonksiyonlarını kapsayan tümleşik bir çözüm sunmaktadır.

3.2 İzlenen Yöntem

YGO'da kapsanan disiplinlerin çeşitliliği, desteklenen projelerin alan ve teknoloji açısından farklılıklar içermesi, kullanıcıya yakın olma, yeni teknoloji kullanımı gibi faktörlerden dolayı, proje geliştirme sürecinde çevik (agile) metodoloji tercih edilmiştir. Çevik yöntemlerin en yaygın hali olan Kenetlenme (Scrum) yöntemi kullanılarak yapılan uyarlamaların dörder haftalık turlarla (iterations) kullanıcıya sunulması ve geri bildirim alınması sağlanmıştır.

Kenetlenme yöntemi, tur planlaması, günlük olarak yapılan kenetlenme toplantıları ve her tur sonunda yapılan değerlendirme çalışmaları ile geliştirme süreci verimliliğini arttırmayı hedefleyen deneysel bir yöntemdir [9][10][11]. Bu yöntem ile her tur sonunda kullanıma sunulmaya hazır bir sürüm geliştirilmesi hedeflenmektedir.

Kenetlenme yönteminin uygulanması için doğrudan YGO çözümünün kendisi kullanılmıştır. Bu amaçla ALM aracı olarak Microsoft Team Foundation Server, tasarım aracı olarak Sparx Systems Enterprise Architect ve bilgi ve doküman

paylaşım aracı olarak da Microsoft Sharepoint Server kullanılmış ve tüm çalışmalar YGO'nun üzerinden yürütülmüştür.

Deneyimsel yöntemin avantajlarından faydalanmak için tüm hedef kitle ile başlamak yerine pilot projeler ile işleme başlanmıştır. Pilot projelerden gelen istek ve bildirimler iş kaydı (Backlog) biçiminde alınarak bir sonraki Kenetlenme planlamalarına dahil edilmiş ve en geç bir ay içerisinde de uyarlamanın/geliştiriminin kullanıcı ile buluşması sağlanmıştır.

Pilot projelerle başlanan kullanım ve uyarlama döngüsü, sisteme yeni projelerin katılımını getirmiş, yeni projelerden gelen istek ve bildirimlerin üretilmesi ile ürünün yetkinliği artmıştır. Bu yöntem ile kullanım ve yetenek gelişimi, yükselen ve genişleyen spiral şeklinde ilerlemiştir.

Proje başlangıcında, teknik istekler ve ürün özelliklerinden oluşan bir iş kaydı kümesi ile yola çıkılmıştır. Bu iş kayıtları, pilot projelerin çalışması desteklemek için önceliklendirilerek eritilmiş, pilot projelerden gelen bildirimler ile artmıştır. Bu yaklaşımla yakalanan "Yap-Bildirim Al-Düzel" döngüsü ile çıktının kendisi hayatın içerisinde elde edilmiştir.

3.3 Elde edilen Sonuçlar

YGO Projesi kapsamında geliştirilen çözüm ile

- Mühendislik süreçlerinin ortaklaştırılması, tümleştirilmesi ve otomasyonu,
- Ürün geliştiriciler için standart bir geliştirme ortamı,
- Süreç şeffaflığı ve ürün/ara ürün çıktıları izlenebilirliği,
- Geliştirme ortamı için ihtiyaç duyulan donanım, lisans ve eğitim maliyetlerinin azalması
- Bilgi paylaşımı artışı sağlanmıştır.

YGO Projesi ile varılmak istenilen hedeflere dönük olarak, aşağıda belirtilen ana başlıklarda kazanımlar sağlanmıştır.

Edinim

HAVELSAN, sistem/donanım/yazılım geliştiren bir kurum olarak, diğer sistem/donanım ve yazılımları da üretim aracı olarak kullanmaktadır. Kurum faaliyet alanında yer alan projelerin, her birinin kendi sistem/donanım/yazılım alt yapılarını tedarik edip çalışır hale getirmeleri söz konusu edinimlerin yüksek maliyetlerle kazanılmasına neden olmaktadır. Ayrıca edinilen araçların başka projeler tarafından tekrar kullanımı da çoğu zaman mümkün olamamaktadır. YGO sonucunda bir kez edinim ve sürekli kullanım gerçekleştirilerek kazanımlar sağlanmıştır.

İşletim

Tedarik edilerek kurulan sistemlerin, projeler için çalışma ortamı hizmeti verebilmesi için devamlılık içeren işletme faaliyetleri gerekir. Projelerin hizmet aldığı sisteme dönük olarak, fonksiyonel kullanıcı desteğinin sağlanması, sistemin ihtiyaçlara göre uyarlanmış şekilde güncel tutulması ve yedeklemelerinin yapılması gereklidir. YGO sonucunda işletimin merkezileşmesi ile işletim maliyetleri azaltılmıştır.

Organizasyon

Yeni bir proje başlatıldığına, bu projeye ilişkin insan kaynağının da sağlanması gerekir. Söz konusu proje, kutudan çıkmış bir sistem/donanım/yazılım ile kurulmuş bir ortamda çalışmak istediğinde hem kendisine özgü uyarlamalar yapılmasına, hem de personelinin ilgili sistemleri kullanabilmesi için eğitime, ihtiyaç duyar. Eğer projelerin diğerlerinden farklı sistemleri seçerek çalıştığı düşünülürse söz konusu uyarlama ve eğitim faaliyetleri tekrarlı hale gelmektedir. Ayrıca, eğitim alınan dönemde yaşanan iş gücü kayıpları da verimliliği etkileyen ek alanları oluşturmaktadır. YGO sonucunda uyarlama ve öğrenme konusunda tekrarlar ortadan kaldırılarak kazanım sağlanmıştır.

Üretim

Projeler, sözleşme ile bağlı kaldıkları takvimlerine göre, gerçekleştirmekle yükümlü oldukları teslimatlara sahiptir. Bu teslimatların üretilmesi için; projenin çalışma ortamına kavuşma süresi, kullanılan araçların iç/dış veri aktarım çabaları, erişim kesintilerinin ortadan kaldırılması ve karar anında doğru veriyi elde etme önem taşımaktadır. YGO sonucunda, servis olarak sağlanan üretim altyapısı ile süreklilik sağlanarak verimlilik artışı elde edilmiştir.

Tablo 1 YGO Kazanımları Tablosu

Ana Başlık	Aktivite	YGO Öncesi	YGO Sonrası
Edinim	Kurulum Faaliyetleri	30 AA ¹	1 AA
	Tedarik Faaliyetleri	45 AA	9 AA
İşletim	Bakım/Yedekleme Faaliyetleri	248 AA	60 AA
	Destek Faaliyetleri	495 AA	180 AA
Organizasyon	Uyarlama	15 AA	4 AA
	Eğitim Tekrarı	150 adet	50 adet
	Eğitimde İşgücü Kaybı	204 AA	15 AA
Üretim	Proje Çalışma Ortamının Hazırlanması (Proje başına)	3 ay	3 saat
	İç Veri Aktarımı/Değişimi	48 AA	0 AA
	Dış Veri Aktarımı/Değişimi	9 AA	0 AA
	Erişim Gecikmesi	60 AA	0 AA
	Karar Destek/Telafi Etme	75 AA	8 AA
	Denetim Kolaylığı	18 AA	4 AA
	Raporlama	18 AA	0 AA

Genel Değerlendirme

Geliştirme ortamının merkezileşmesi ve bulut üzerinden hizmet verilmesi sonucunda; edinim, işletim, eğitim ve üretim altyapısı konularında verimlilik artışı sağlanmıştır. Verimlilik değerleri, süreli projeksiyona dayalı bir işletme modellemesi yapılarak hesaplanmıştır ve Tablo 1'de verilmiştir². Model tabanlı hesaba göre,

¹ Adam.Ay

² İşletme modeli, 5 yıllık zaman diliminde 50 projenin gerçekleştirileceği öngörülerek yapılmıştır

tasarruf edilen iş gücü 984 AA'dır. Ayrıca, geliştirme ortamının ortaklaştırılması ile sunucu donanımları ve lisans alanında önemli tasarruflar sağlanmıştır.

4 Sonuç

Bu çalışmada savunma sanayinde yer alan HAVELSAN'ın tümleşik UYY sürecini uygulamaya geçişinde yaşadığı tecrübeler aktarılmıştır. YGO'nun bir sistemi olarak ele alınması ve diğer bilgi sistemleri ile entegrasyonunun sağlanması kurumsal otomasyonun sağlanması açısından önemlidir. Tümleşik ve merkezi çalışan UYY altyapısı ile şirketin geliştirme ortamı kurulumu için harcadığı toplam donanım ve yazılım maliyetleri ile kurulum ve uyarlama (bakım) emeği azalmıştır. Ürün geliştiricileri için, standart bir geliştirme ortamı sağlanarak birden fazla projede çalışırken ortaya çıkan eğitim ihtiyacının ve araç yabancılaşmasının azaltılması, bilgi paylaşımının ve personel verimliliğinin artması sağlanmıştır. Sürekli iyileştirme ve raporlama için güvenilir veri toplanması, süreç faaliyetlerinin standartlaştırılması sağlanmıştır.

Bundan sonraki dönemde UYY altyapısı kullanılarak; kurumsal entegrasyonların artırılması, süreçlerin etkinlik ve verimliliğin ölçülmesi, analiz edilmesi ve iyileştirmelerin bu analizlere dayanarak yapılması hedeflenmektedir.

Teşekkür. Yazarlar HAVELSAN yönetimine çalışmaya verdiği destek için teşekkürler ederler.

Referanslar

- [1] D. Chappell, "What is Application Lifecycle Management?", *Chappell & Associates* 2008.
- [2] P. J. W. Lipsitz, "The Total Economic Impact Of Microsoft Application Lifecycle Management," March, 2013.
- [3] J. Kääriäinen and A. Välimäki, "Impact of Application Lifecycle Management – A Case Study."
- [4] C. Schwaber, "The Changing Face of Application Life-Cycle Management," 2006.
- [5] K. A. Shaw, "Application lifecycle management for the enterprise," no. April, 2007.
- [6] C. Ebert, "Improving engineering efficiency with PLM / ALM," pp. 443–449, 2013.
- [7] J. K. and A. Välimäki, "Applying Application Lifecycle Management for the Development of Complex Systems : Experiences from the Automation Industry," in *EuroSPI*, 2009, pp. 149–160.
- [8] H. Lacheiner and R. Ramler, "Application Lifecycle Management as Infrastructure for Software Process Improvement and Evolution: Experience and Insights from Industry," *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 286–293, 2011.
- [9] K. Schwaber and J. Sutherland, "The scrum guide," *Scrum.org*, October, vol. 2, no. July, p. 17, 2011.
- [10] M. Fowler and J. Highsmith, "The agile manifesto," *Software Development*, vol. 9, no. August, pp. 28–35, 2001.
- [11] H. Takeuchi and I. Nonaka, "The new new product development game," *Harvard business review*, no. January-February, pp. 137–147, 1986.