

Uygulama Yaşam Döngüsü Yönetimi Altyapısı için Gelişlet (DevOps) Deneyimi

Ahsen İkbal Aytekin¹, Yagup Macit¹, Eray Tüzün²

¹HAVELSAN Bilişim Hizmetleri Grup Müdürlüğü

²HAVELSAN Teknoloji ve Akademi Direktörlüğü 06510 Ankara, Türkiye

Özetçe. Günümüz şirketleri, müşterilerinin beklentilerini anlamak ve bu beklentilere hızlı ve doğru bir şekilde tepki vermek durumundadır. Artık, bilişim sistemleri karar destek sistemleri olmaktan çıkmış ve doğrudan yönetimin, üretimin, ticaretin ve sosyal hayatın yaşandığı sistemler haline gelmiştir. Hayatın kendisini beklemeye almak hayattan kopuş ile sonuçlanmaktadır. Uygulama yazılımları için hem ilk fikrin kullanıma verilmesi hem de var olan yazılım üzerinde değişiklik yapılması hızlı davranış yeteneği gerektirmektedir. Bu hız yeteneği, uygulamayı geliştirenler ile aynı uygulamayı işletenler arasındaki süreçsel ve organizasyonel mesafenin kısılması ile olanaklıdır. Gelişlet (DevOps), geliştiriciler ve işleticilerin aynı çatı altında ve işbirliği içerisinde birlikte çalışarak kullanıcı ihtiyaçlarına hızlı tepki vermelerini sağlayan bir yaklaşım olarak tanımlanmaktadır. Bu çalışmada, sistem ve yazılım üreten ve kullanıcının bilişim profesyoneli olduğu büyük ölçekli bir organizasyonda Uygulama Yaşam Döngüsü Yönetimi altyapısının, Gelişlet yaklaşımı ile işletilmesi, bakımı ve kazanımları anlatılmıştır.

Anahtar Kelimeler: Uygulama yaşam döngüsü yönetimi, Hızlı Uygulama Geliştirme, Uygulama Değer Yönetimi, Yazılım İşletimi, DevOps, Gelişlet

1. Giriş

Günümüzde şirketler, ister yazılımları kendileri geliştirsün ister platform olarak tedarik etsin, işletim esnasında kullanıcıların gündelik pratiklerini takip etmek ve onlara daha etkili yazılımlar sağlamak istemektedirler. Yazılım uygulamalarında, fikrin ilk çıkışından başlamak üzere geliştirme, test, dağıtım ve bakım süreçlerinin tamamı Uygulama Yaşam Döngüsü Yönetimi (UYU) olarak tanımlanmaktadır [1].

İlk UYU çözümlenmeleri, yaşam döngüsü aktivitelerinin ayrı yürütüldüğü farklı araçların entegrasyonundan oluşmaktadır. UYU 2.0 yaklaşımında, yazılım geliştirme süreçlerinin bütünlüklü olduğu araçlar çözüm olarak sunulmaktadır. UYU 2.0 ile gelen tümleşik ortam üzerinde yinelemeli geliştirme ve sürekli inşaa gibi yeni deneyimler elde edilmiş ve Çevik Bildiri [2] ile anlayış değişikliği vurgulanmıştır. Çevik pratiklerin ilgi görmesi ve bu alandaki gereksinimlerin artması sonucunda UYU 2.0+ [3] ile planlama, işbirliği ve sürüm dağıtımının da kapsandığı yaklaşıma geçilmiştir. Yazılımı işleten ve kullanan birimlerin de sürece eklenmesi sonucunda, aktiviteleri uçtan uca kapsayan, genişletilmiş UYU denilebilecek Gelişlet [4] yaklaşımı ortaya çıkmıştır.

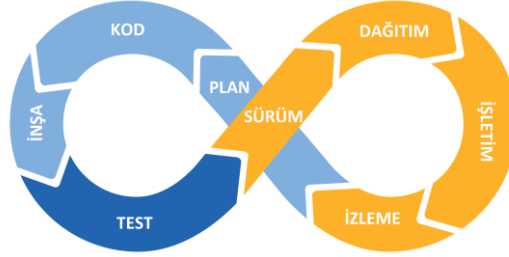
HAVELSAN, yüklenicisi olduğu projelerin yazılım ve sistem çalışmalarını gerçekleştirmek için ihtiyaç duyduğu ortak araçlardan oluşan altyapı hizmetini UYU 2.0+ uyumlu özel bulut şeklinde merkezileştirmiştir [5]. Altyapı hizmet sunumunda süreçler, geliştirme yöntemleri, takım yapıları ve işbirliği konularında proje ve gruplara göre farklılaşan ihtiyaçlar ortaya çıkmıştır. Kısa zamanda değişebilen bu ihtiyaçları,

işbirliği içerisinde hızla karşılayabilmek ve değişim sonuçlarını izleyebilmek için UYY altyapı hizmetinin verilmesinde Gelişlet yaklaşımına geçilmiştir.

Bildirinin ikinci bölümünde, Gelişlet yaklaşımı hakkında bilgi verilmiştir. Üçüncü bölümde, HAVELSAN UYY altyapısının Gelişlet motivasyonu, izlenen yöntem ve elde edilen sonuçlar alt başlıklar halinde anlatılmıştır. Son bölümde ise gerçekleştirilen deneyim genel olarak özetlenerek, yapılan değerlendirmeler aktarılmıştır.

2. Gelişlet

Gelişlet terimi, 2008 yılında “Development” ve “Operations” sözcüklerinin bileşiminden oluşturulan DevOps [4] teriminden yola çıkılarak, “Geliştirme” ve “İşletme” sözcüklerinden türetilmiştir. Gelişlet yaklaşımı, Çevik geliştirme modelinin evrimi ile vizyon, özellik, geliştirme, test, inşa, dağıtım, işletme, izleme, geri bildirim konularını uçtan uca kapsayacak şekilde ürün sahibi, geliştiriciler, işletim birimleri ve kullanıcılar arasındaki işbirliğini arttıran kültür olarak ortaya çıkmıştır[6]. Bu kültürün işletim döngüsü Şekil 1’de görülmektedir.



Şekil 1 - Gelişlet yaklaşımı döngüsü [7]

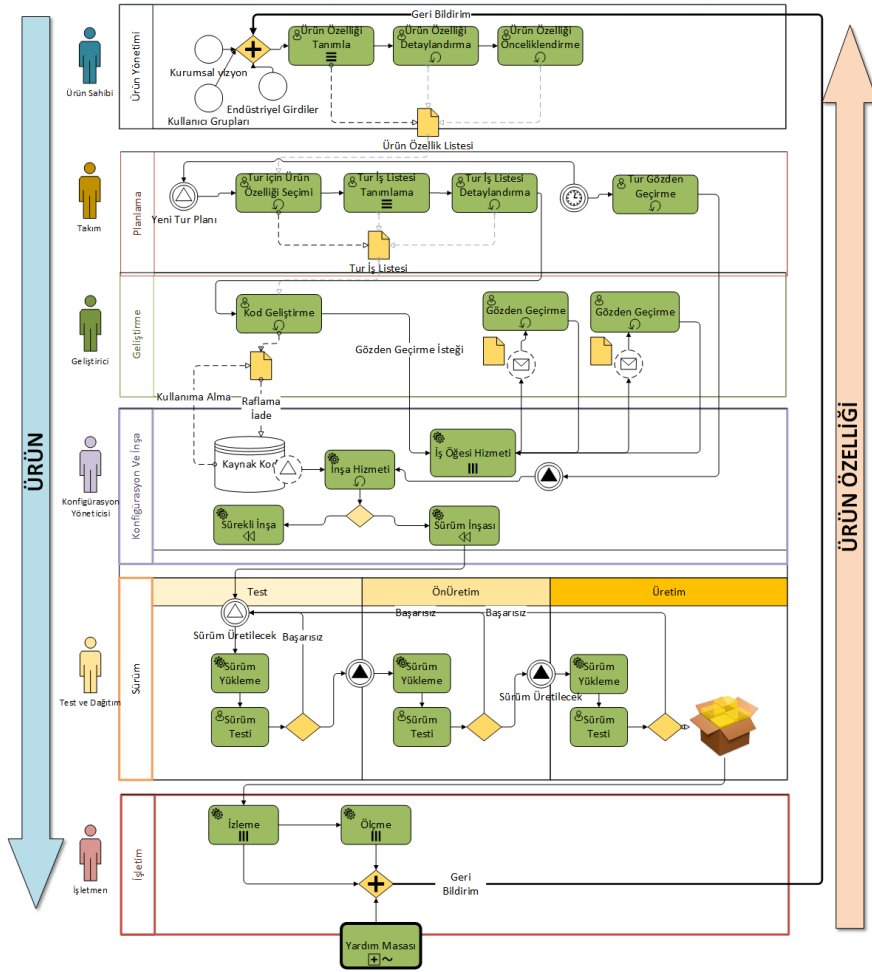
Gelişlet döngüsü, planlama evresi ile başlamaktadır. Planlama evresinde, ürüne ait özellikler ürün sahibi tarafından önceliklendirilerek tur planına dahil edilmekte ve geliştirme faaliyetleri tetiklenmektedir [8]. Geliştirme evresinde, sürekli inşa pratiği ile kod yazımı, inşa ve test zinciri korunmaktadır. Dağıtım evresinde, sürümün inşası alınarak dağıtım gerçekleştirilmektedir. İşletim evresinde ise dağıtılan ürünün işletilmesi ve izlenmesi yapılmaktadır. İzleme sonuçlarına göre tekrar planlama evresine geçilmektedir.

3. HAVELSAN Deneyimi

HAVELSAN, 4 genel müdür yardımcılığı çatısı altında 800+ mühendis olmak üzere 1200 üzerinde çalışanı olan bir savunma sanayi şirkettir. Her yeni gelen proje ile geliştirme ortamının ihtiyaçları sürekli farklılaşabilmektedir. Böyle bir dinamik yapıda, projelerin ihtiyaçlarına hızlı bir şekilde cevap verebilmek ve çoğunluğu mühendislerden oluşan iç müşterilerin memnuniyetlerini sağlayabilmek amacıyla, Kurumsal UYY altyapısı [5] devreye alınmıştır.

Geliştirilen Kurumsal UYY altyapısı ile yaklaşık 60 mühendislik grubuna destek verilmektedir. Bu mühendislik grupları, genel müdür yardımcılıklarına bağlı değişen büyüklüklerde (5 ila 60 kişi) takım ve takımların bağlı olduğu gruplardan oluşmaktadır. Mühendislik grupları, herhangi bir zaman diliminde ortalama 50 aktif projenin

geliştirme etkinliklerinden oluşan bir iş büyüklüğünü yürütmektedir. Bu altyapı sayesinde mühendislik grupları farklı geliştirme metodolojileri uygulayabilmekte ve şelaleden, çok fazlı modele ya da çevik yöntemlere geçebilmektedir. UYY altyapısı, İş Sürekliliği Yönetim Sistemi (ISYS) kapsamında, özel bulut üzerinden, tüm geliştiricilere kesintisiz olarak hizmet vermek gibi ek yükümlülüklerle sahiptir. Bu gereksinimleri karşılayabilmek amacıyla, UYY altyapı hizmetinin işletim ve bakımı için 2014 yılında Gelişlet kültürüne geçmesine karar verilmiştir. Altyapı hizmetinin Gelişlet yaklaşımı ile işletim ve bakımı, 5 kişilik bir ekip tarafından ikişer haftalık planlı sürüm çıkarma şeklinde yapılmaktadır.



Şekil 2 – HAVELSAN UYY altyapısının Gelişlet döngüsü.

UYY altyapı hizmeti, çeşitli kanallardan gelen girdiler ve ihtiyaçlara göre sürekli iyileştirilmektedir. İhtiyaçların planlanması ve gerçekleştirilmesi sonucunda hizmetin sürümü yükseltilecek canlı kullanıma alınmaktadır. Canlı sistem, sürekli izlenerek ve

kullanıcı geri dönüşleri alınarak sisteme tekrar girdi sağlanmaktadır. UYY altyapı hizmetinin Gelişlet döngüsü Şekil 2’de gösterilmektedir. Bu döngüdeki adımlar alt başlıklarda ayrıntılı olarak aktarılmıştır.

3.1. Planla

UYY altyapısı, kullanıcı gruplarından, yardım masasından ve kurumsal vizyon gibi kanallardan gelen girdilerle sürekli iyileştirilmektedir. Birinci kanal, mühendislik grupları ile 15 günlük periyodlarla gerçekleştirilen kullanıcı görüşmeleridir. Bu görüşmede mühendislik gruplarının temsilcileri ile ürünün değerlendirmesi yapılarak istenilen özellikler ve geri bildirimler alınır. İkinci kanal, olarak Yardım Masası talepleri kullanılmaktadır. Kullanıcılar bu platform üzerinden istekte bulunabilir ve hata bildirimini yapabilirler. Üçüncü kanal ise ISYS, kurumsal vizyon ve endüstriyel gelişim ihtiyaçların bu girdilere eklenmesi şeklindedir.

Gelişlet takımı zaman içerisinde değişen ortalama 6 kişiden oluşmakta ve 2 haftalık turlar ile çalışmaktadır. Bu turların planlama evresinde talep edilen özellikler ve bildirilen hatalar, önceliklerine ve büyüklüğüne göre tur içerisinde gerçekleştirilmek üzere seçilir. Seçilen ürün özellikleri, takım üyelerinin yapması gereken görevlere kırılarak koşulacak olan tur hazır hale getirilir.

3.2. Geliştir ve Yapılandır

Bu aşamada, günlük kenetlenme toplantıları ile görevlerin üzerinden geçilerek her takım üyesinin, diğer üyelerin yaptığı işler ve engellenen işler konusunda bilgilendirilmesi sağlanmaktadır. Takım üyeleri, üzerlerine atanan görevleri çalışarak, geliştirdikleri kodların birim testlerden geçtiğine emin olduktan sonra takım üyelerinden birine kod gözden geçirme isteği gönderirler. Kod gözden geçirme işlemi, geliştiricinin kendisi dışındaki bir başka göz tarafından aynı kodun incelenmesi olarak gerçekleştirilir. Gözden geçirme yapan takım üyesi, kod üzerinde gözlemlediği eksiklikler, fazlalıklar ve hatalı yazımlar için ilgili satırlara yorum girerek gözden geçirme isteğini geliştiriciye geri gönderirler. Geliştirici, kod gözden geçirme isteği doğrultusunda kendisine iletilen yorumlara göre düzeltmelerini gerçekleştirerek kodu kaynak kod deposuna iade eder. Bu iade işlemi ile tetiklenen sürekli inşa, hatalı ve inşayı kıran bir kodun, kaynak kod deposuna iadesini reddederek ilgili geliştiriciye düzeltilmek üzere geri gönderir. Bu inşanın başarılı olması durumunda ise inşa sırasında çalışan kod kalite ölçüm aracının (SonarQube) sunmuş olduğu çıktılar dikkate alınarak ek düzeltme faaliyetleri gerçekleştirilir.

3.3. Sına ve Konumlandır

Başarılı olan sürekli inşa çıktıları, geliştirme ortamına otomatik olarak yüklenmekte ve planlama aşamasında oluşturulan test görevleri ile çapraz-fonksiyonlu takım üyeleri tarafından test edilmektedir. Kullanıcı testi olarak adlandırdığımız bu testlerden başarısız olan geliştirmeler için hata açılarak aynı tur içerisinde düzeltme faaliyeti yürütülmektedir. Başarılı olan geliştirmeler ise tur sonu teslimat kümesine aday olarak girerler. 2 haftalık turun sonunda, tur gözden geçirme toplantısı

gerçekleştirilerek aday geliřtirmelerden bitti tanımına (definition of done) uyanlar etiketlenerek tur ıktısı belirlenir.

Konfigürasyon yöneticisi tarafından etiketlenen kaynak kodlar üzerinden, test, ön-üretim ve üretim ortamlarına dağıtım amacıyla sürüm inřası başlatılmaktadır. Sürüm inřasına ait ıktılar Sürüm Yönetim Aracı (Microsoft Release Management) ile dağıtılmak üzere konfigüre edilmektedir. Bu araç üzerinde, tanımlanan her bir ortam için dağıtım onaylamak ve dağıtımın başarısını doğrulamak üzere ilgili takım üyelerine atamalar yapılmıştır. Dağıtım ortamı tek düze bir yapıya sahip olduđu için ayrıca bir dağıtım konteyneri kullanımına gerek görülmemiştir. Sürüm inřası tetiklendiğinde, etiketlenen kaynak kodların ıktısı üretilerek, test ortamı için dağıtım onayına sokulur ve onay sonucunda test ortamına sürüm dağıtımını gerçekleştirilir. Sonrasında yapılan testler ve doğrulama onayı ile dağıtımın ön-üretim ortamına otomatik olarak aktarılması sağlanır. Benzer şekilde ön-üretim ortamında da başarı elde eden sürümün canlı sisteme dağıtımını gerçekleştirilerek planlanmış olan tur ıktısı kullanıma verilir.

3.4. İzle ve Öğren

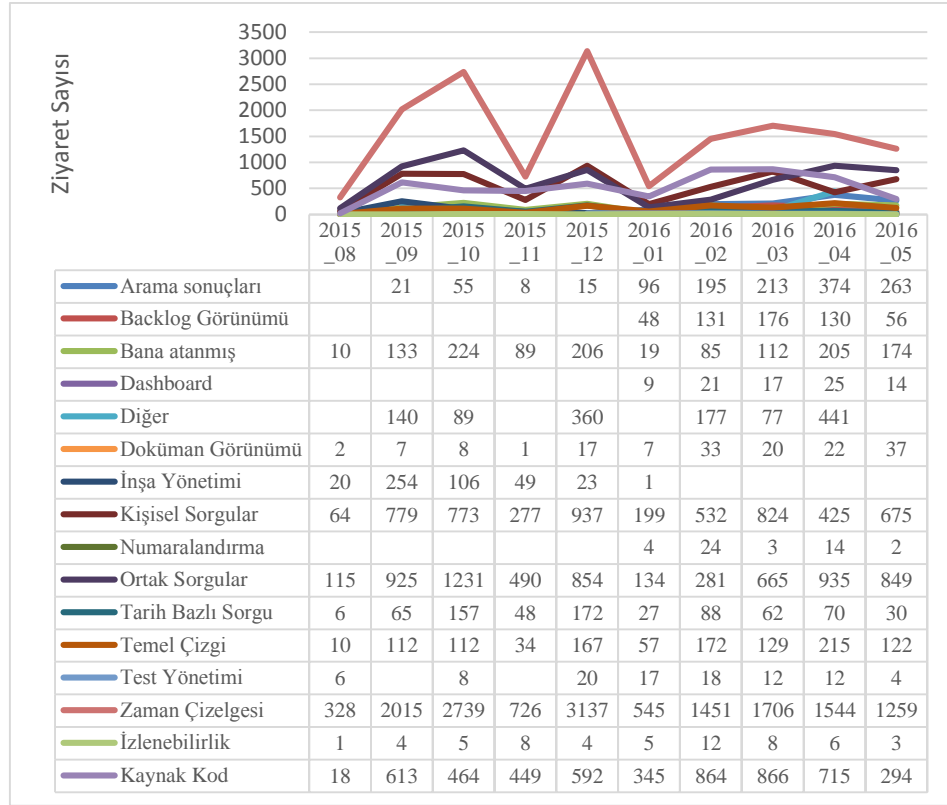
UYY altyapı hizmeti, kurumsal ölçekte yürütölen ISYS kapsamında, süreklilik ve kullanıcı memnuniyeti açısından izlenmektedir. Bu izlemenin parçası olarak, yardım masasından elde edilen, geliştirilen talep sayısı ve memnuniyet anketi sonuçları, 2015 Ağustos ayından 2016 Haziran ayına kadar olan dönem için Tablo 1’de verilmiştir.

	Geliřtirme Sayısı	Müşteri Memnuniyeti
Ağustos 2015	2	5,0
Eylöl 2015	18	4,5
Ekim 2015	8	4,67
Kasım 2015	12	4,67
Aralık 2015	20	4,83
Ocak 2016	19	4,8
Şubat 2016	14	3,67
Mart 2016	13	4,67
Nisan 2016	15	NA
Mayıs 2016	8	5

Tablo 1- UYY hizmeti için aylık bazlı geliřtirme sayısı ve memnuniyeti

Uygulama kullanım izleme aracı (PIWIK) üzerinden, UYY Altyapısı canlı sistemi sürekli olarak izlenmektedir. Bu izleme ile sistem gözlemlenerek, yeni eklenen özelliklerin kullanımı, mevcut ürün özelliklerinin kullanımındaki artış ve düşüşler, uygulamaya bağlanılan tarayıcı türleri ve sürüm bilgileri izlenmekte ve raporlanmaktadır. Yapılan izlemeler sonucunda yeni istenen özelliklerin gerçekleştirimi, mevcut olanların deđiřtirilmesi veya kaldırılması için kararlar alınarak ürün güncelleřtirilmektedir. Bu güncelleřtirme yaklaşımı ile son bir yılda 20 adet dağıtım oluşturulmuş ve dağıtılmıştır.

UYY Altyapısı ürün özelliklerinin 2015 Ağustos ayından 2016 Haziran dönemine kadar geçen süreçte örün (web) istemcisi üzerinden gerçekleştirilen ziyaret sayıları Şekil 3'te verilmiştir.



Şekil 3 - UYY ürün özelliklerinin ziyaret sayısı

Verilen tabloda, Backlog Görünümü ve Dashboard gibi yeni özelliklerin 2016_01 sürümü ile geldiği görülmektedir. Ayrıca, Kişisel Sorgular, Ortak sorgular ve Zaman Çizelgesi gibi özelliklerin yoğun olarak kullanıldığı izlenebilmektedir. Özellikle Zaman Çizelgesinin iniş ve çıkışları, kullanıcının ilgisinin izlenmesi için kritik bir göstergedir. Test Yönetimi özelliklerinde kullanılan erişim aracı, örün tabanlı olmadığı için Test Yönetimi erişiminin sınırlı olduğu görülmektedir. Doküman Görünümü ve İzlenebilirlik özellikleri sadece gereksinim yöneticisi rolü ile ilişkili olduğu için diğer geliştirme özelliklerine göre daha az erişim görülmektedir.

Bu bilgiler bir sonraki tur planına girdi sağlayarak, kullanılmayan özellikler kaldırılmakta, olağan dışı tespit edilen işlemler için analizler yapılmaktadır. Kullanıcıların erişim sağladıkları tarayıcılara göre yapılan geliştirmeler kontrol edilmekte ve gerekli durumlarda kullanıcıların tarayıcı güncellemelerinin merkezi olarak gerçekleştirilmesi talep edilmektedir.

4. Sonuç

Bu çalışmada Kurumsal UYY Altyapısı hizmetinin Gelişlet modeli ile nasıl yürütüldüğü aktarılmıştır. Kurumsal UYY altyapısı hizmeti 2012 yılında pilot projeler ile başlamış, 2013 yılı Mayıs ayında işleme geçilerek mevcut projelerin bu altyapıya geçişleri ile yaygınlaştırılmıştır. 2014 yılında geçilen Gelişlet yaklaşımı ile ikişer haftalık periyodlar halinde yaklaşık 60 tur planlaması gerçekleştirilmiş ve bu turlara ait 60 çıktı ile canlı sistem sürüm yükseltmesi yapılmıştır.

Gelişlet yaklaşımı sonrasında, kullanıcı gruplarının doğrudan ürün özelliklerinin ve önceliklerinin belirlenmesi süreçlerine katılımları ile doğru üretim yapılması sağlanmıştır. Talep edilen özelliklerin kullanıma verilmesi Gelişlet yaklaşımı öncesinde 2 ay sürerken bu yaklaşım sonrasında süre 3 haftaya indirilerek ihtiyacın hızlı karşılanması sağlanmıştır.

3 yıllık işletim süresince, planlı bakım faaliyetleri dışında hizmet yalnızca 1 günlük kesintiye uğramış, bu kesintinin nedeni olarak anti virüs yazılımının davranışı tespit edilmiştir. Elde edilen sonuçlara göre, ürün üzerinde talep edilen özelliklerin kullanıcılara hızlı bir şekilde sunulmasını sağlayan entegre Gelişlet altyapısının başarılı olduğu, memnuniyet anketlerinde görülmüştür.

DevOps terimi için Türkçe karşılık taraması yapılmış ve karşılık bulunamayınca Gelişlet terimi türetilmiştir.

Gelecek dönemde, ölçümlerin ve bu ölçümlere bağlı olarak yapılan değişikliklerin daha etkili olabilmesi için metrik altyapısının geliştirilmesi planlanmaktadır.

Teşekkür. Yazarlar, HAVELSAN yönetimine çalışmaya verdiği destek için teşekkür ederler.

5. Referanslar

- [1] D. Chappell, «What is Application Lifecycle Management?,» Chappell & Associates, 2008.
- [2] «Agile Manifesto,» [Çevrimiçi]. Available: <http://www.agilemanifesto.org/>. [Erişildi: 17 Haziran 2016].
- [3] D. West, «The Time Is Right For ALM 2.0+,» Forrester Research, 2010.
- [4] Debois, Patrick, «Agile 2008 Toronto,» *Just Enough Documented Information*, Toronto, 2015.
- [5] Y.Macit, E.Tüzün, K.Ince, A.I.Aytekin, «Büyük Ölçekli Bir Organizasyonda Uygulama Yaşam Döngüsü Yönetimi Uygulama Deneyimi,» *Proceedings of the 8th Turkish National Software Engineering Symposium*, 2014.
- [6] «Emerging Technology Analysis: DevOps a Culture Shift, Not a Technology,» Gartner, 2013.
- [7] «Collabnet Web Sitesi,» 17 Haziran 2016. [Çevrimiçi]. Available: <http://www.collabnet.cn/products/teamforge/deploy>.
- [8] Ken Schwaber, Jeff Sutherland, «The Scrum Guide,» Temmuz 2013. [Çevrimiçi]. Available: <http://www.scrumguides.org/>. [Erişildi: 17 Haziran 2016].