

Yazılım Geliştirme Süreçlerinde Sürekli Entegrasyonun Sağlanması

Doç.Dr.Erhan Akyazı¹, Şafak Kayıkcı²

¹ Marmara Üniversitesi, İletişim Fakültesi, İstanbul

² Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul

eakyazi@marmaraedu.tr, safakkayicki@yahoo.com

Özet: Günümüz dünyasında, büyük yazılım projeleri kalabalık yazılım ekipleri tarafından hazırlanmaktadır. Çok sayıda yazılımcının bulunduğu bu projelerin geliştirme süreçlerinde oldukça sık rastlanan kod değişiklikleri ve yeni kod eklendiği durumlarında sistemin devamlılığını sağlamak oldukça önemlidir. Sürekli entegrasyon (Continuous Integration = CI), yapılan bu değişiklikler ve geliştirmelerin sisteme entegrasyonunu sağlar. Ayrıca sistemin bazı bölümlerinde bozulmalara yol açıp açmadığını tespit etmek için kullanılan yöntemdir. Sürekli entegrasyon sisteminin uygulanabilmesi için sistemde, CI sunucusu, versiyon yönetim sistemi, geri bildirimler için email sunucusu, otomatik yapılandırma ve birim test araçlarının bulunması gerekir.

Bu çalışmada, sürekli entegrasyon sistemlerin çalışma prensipleri ve sistem içerisindeki elemanların kullanım amaçları anlatılmıştır. Ayrıca günümüzde en çok kullanılan sürekli entegrasyon araçları olan Hudson ve Cruise Control karşılaştırılmalı olarak incelenmiştir.

Anahtar Sözcükler: Yazılım, yazılım geliştirme, entegrasyon.

Providing Continuity of Integration in Software Development Processes

Abstract: In today's world, big software projects are developed by large software teams. It is very important to keep the project stable and continuous when new builds and code changes occur in such a crowded environment. Continuous integration(CI) is a software development practice to provide integration of these changes and developments to the system. Also CI checks if there has been any corruption in any part of system after integration. There should be a CI server, version control system, email server for feedbacks, automatic build and unit test tools in the system.

In this study, the principles of continuous integration system and purposes of the system elements are described. Besides, mostly used integration tools which are Hudson and Cruise Control are comparatively researched.

Keywords: Software, software development, integration.

1. Giriş

Sürekli entegrasyon, birçok organizasyon tarafından yapılandırma işlemlerini otomatikleştirmek için kullanılan bir süreçtir. Yazılım projelerinde görev alan kişilerin geliştirdikleri yazılımın, mevcut geliştirilmiş olan sisteme sürekli ve otomatik bir şekilde entegrasyonun yapılabiliyor olması demektir.

Entegrasyon işleminin sürekli ve otomatik yapılıyor olması, hem projenin esnek olmasına katkı sağlar hemde çıkabilecek hatalara karşı daha kolay çözüm seçenekleri sunar.

Martin Fowler'a göre sürekli entegrasyon :

“Programcı ekibi tarafından yapılan değişikliklerin sık aralıklarla sisteme entegre

edilmesi için kullanılan yazılım geliştirme yöntemidir. Her programcı günde en az bir kere yaptığı değişiklikleri entegre eder. Bu sayede gün boyunca birden fazla entegrasyon gerçekleşir. Otomatik sürüm oluşturulması ve mevcut testler yardımı ile entegrasyon kontrol edilir. Birçok ekip tarafından sürekli entegrasyon metodu ile entegrasyon sorunlarının azaldığı tespit edilmiştir. Sürekli entegrasyon ile bir ekip hızlı bir şekilde kendi içinde bütün ve çalışır programlar yazabilmektedirler.” [1]

Sürekli entegrasyon işlemlerinde gerek duyulan elemanları şu şekilde sıralayabiliriz:

- Sürekli Entegrasyonu otomatik olarak gerçekleştiren bir sunucu
- Versiyon yönetim sistemi
- Geri bildirim için kullanılmak üzere bir email serveri
- Otomatik yapılandırma aracı

Sürekli entegrasyon ortamlarında, kaynak kod, uygulamanın sürekli gözlemlendiği ve bir değişiklik(commit) gördüğü zaman müdahale ettiği merkezi bir lokasyonda tutulur.

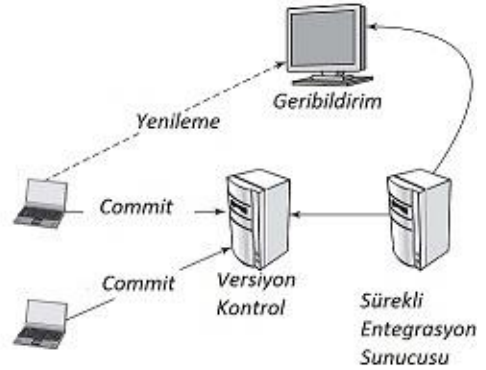
2. Sürekli Entegrasyon Avantajları

- Her an başarılı yapılandırılmış bir sürümün çıkarılmasını garanti eder.
- Takip edilebilir bir ilerleme raporu ve problem izsürümü sağlar.
- Varolan geliştirme ortamına kolayca entegre olur.
- Kaliteli kod bulundurulmasını sağlayarak geliştirme standartlarını artırır.
- Hızlıca hatalar saptanarak çok kısa sürede düzeltilir.

3. Sürekli Entegrasyon Dezavantajları

- Birçok geliştirme ortamına yayılmış çok sayıda projenin entegrasyon ortamına aktarılması oldukça sıkı bir planlama ve koordinasyon gerektirir.
- Yapılandırma ve versiyon kontrol makineleri donanım maliyetleri çıkarır.

4. Sistemin Çalışması



Her geliştirici, versiyon kontrol sisteminden projenin en son halini çekerek kendi lokal ortamına alır. Kendi güncellemelerini ve geliştirmelerini yaptıktan sonra, yapılandırma scripti ile derlemelerini yapar. Eğer hatalar varsa düzeltilir ve daha sonra versiyon kontrol sistemi içerisine atılır (commit). Sunucu versiyon kontrol sisteminde bir değişiklik gördüğü zaman otomatik yapılandırma ile tüm projeyi yapılandırır ve eğer hatalar varsa geliştiricilere mail göndererek hatayı bildirir (feedback). Hata olmadığı zamanda sunucu sistemi devreye alır ve geliştiricilere olumlu geribildirimde bulunur.

5. Hudson ve Cruise Control Karşılaştırılması

Şu an günümüz dünyasında en çok kullanılan entegrasyon yazılımları Hudson ve Cruise Control'dür. Bu bölümde iki sistemin özellikleri karşılaştırılmıştır.[2]

Özellik	Cruise Control	Hudson
Kaynak Kod	Java, .NET	Java
Lisans	BSD	MIT
Sponsorlar		Sun Oracle
Container Teknolojisi	Jetty	Winstone Ayrıca Jetty ve Tomcat ile çalışabilir
Güncelleme	Sürümler seyrek ancak elle güncelleme gerekli	Küçük güncellemeler çok sık. Web arayüzü ile güncellenebilir
Kararlılık	İyi Ancak bazı derleme hatalarına karşı dayanıksız (örn:deadlocks)	Oldukça iyi
Kontrollü Kapama	Yok	Var
Konfigürasyon	XML değişikliği ile	Web arayüzü ile
Plugin Konfigürasyon	Kaynak dosyaların elle değiştirilmesi	Web arayüzü ile
Gösterge Tablosu Güvenliği	Yok. Bütün kullanıcılar sadece birkaç parametre değiştirebilir ama çoğu konfigürasyon XML ile	Kullanıcı doğrulaması HTTP ile yapıp lokal bir tablo yada LDAP yada servlet container ile
Desteklenen Diğer SCM'ler	Accurev, Alien Brain, ClearCase, CVS, Darcs, Git, Harvest, Mercurial, MKS, Perforce, PVCS, SNApshotCM, StarTeam, Synergy, Surround, MS Team Foundation, VisualWorks Store, MS VSS	Accurev, Bazaar, Bitkeeper, ClearCase, CVS, Git, Mercurial, Perforce, PVCS, StarTeam, Synergy, MS Team Foundation, MS VSS

Ant Desteği	Var	Var
Maven2 Desteği	Var	Var
Email / RSS Bilgilendirme	Var	Var
Jabber Bilgilendirme	Var (plugin ile)	Var (plugin ile)
Bugzilla Bilgilendirme	Var (plugin ile)	Var (plugin ile)

Tablo 1: Cruise Control – Hudson Karşılaştırması

6. Sonuç

Sürekli entegrasyon düzgün olarak uygulandığı takdirde, projede bulunan takım elemanlarına vaktinde geribildirimde bulunarak, hatalara karşı en kısa zamanda düzenlemeler yapılmasını sağlar ve böylece yazılımcı üretkenliğini önemli ölçüde artırır. Ayrıca sürekli entegrasyon araçları gecelik otomatik yapılandırma gibi işlerde de kullanılabilir. Özellikle çevik süreçlerle adı sıkça birlikte anılan sürekli entegrasyon, yazılım geliştirme kalitesini arttıran önemli bir yöntem haline gelmiştir.

7. Kaynaklar

- [1] <http://www.martinfowler.com/articles/continuousIntegration.html> (24/12/2011)
- [2] <http://bigbeeconsultants.co.uk/blog/cruise-control-vs-hudson/> (24/12/2011)
- [3] <http://www.kurumsaljava.com> (25/11/11)
- [4] Duvall Paul M., Steve Matyas, Andrew Glover, **Continuous Integration: Improving Software Quality and Reducing Risk**, Addison Wesley, Boston, 12-20, 209-211, (2007)