

TC
BAKIRÇAY ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM MÜHENDİSLİĞİ TEMELLERİ
BAHAR DÖNEMİ

VERSİYON KONTROL SİSTEMLERİ

HAZIRLAYAN
210601017-SUEDA AHSEN GÖKALP

Versiyon Kontrol Sistemleri

Versiyon Kontrol Sistemleri, bir dokümanda zamanla yapılan değişiklikleri yönetmemize, adım adım kaydetmemize, internet üzerinde depoda saklamamızı ve yönetmemizi sağlayan sistemlerdir. Açık kaynaklı projeler ele alınarak geliştirilen yeni projeler için kolaylık sağlamış olur. Versiyon kontrol sistemi görünürlüğü arttırması, her bir ekip üyesinin farklı ülkelerde bulunmasına rağmen iş birliği yapabilmesine yardımcı olması, ürün geliştirme sürecini hızlandırması açılarından çok önemlidir. CVS Version Control, SVN, GIT, GitHub, GitLab, Bitbucket gibi sistemler versiyon kontrol sistemlerine örnek verilebilir. Versiyon Kontrol Sistemleri üç tiptir bunlar: Yerel Versiyon Kontrol Sistemleri, Merkezi Versiyon Kontrol Sistem Tipleri ve Dağıtık Versiyon Kontrol Sistemleridir.

1. Dağıtık Versiyon Model Sistemleri: En çok kullanılan ve en yeni versiyon sistemi modelidir. Yerel ve merkezi sistemlerin özelliklerini taşıyan karma bir sistemdir. Geliştiriciler uzak depoya değişiklik yapmak istediklerinde veya proje geçişine bakmak istediklerinde bağlanmak zorunda değillerdir. Verileri kaybetme riski düşük bir sistemdir. Git, Mercurial, BitKeeper gibi versiyon kontrol sistemleri buna örnek verilebilir.
2. Merkezi Kontrol Model Sistemleri: Takım çalışmaları için üretilmiştir. Bu modelde Versiyon Kontrol Sunucusu ve versiyon veri tabanı vardır. Bilgisayarlar bu sunucu sayesinde iletişimde olurlar ve versiyon kontrol sunucusuna doğrudan erişebilirler. Değişiklik yapılacak dosya sunucudan indirilip güncellemesi yapılır. Ancak bu kontrol sisteminde sunucunun diskinde bir bozulma meydana gelirse ve dosyalar yedeklenme konusunda eksikse tüm dosyaların kaybedilme riski vardır. Bu modele örnek olarak Subversion, Vesta veya CVS verilebilir.
3. Yerel Versiyon Kontrol Sistemi: Eski bir versiyon kontrol sistemi yaklaşımıdır. Projelerdeki değişiklikler kullanıcı bilgisayarında, yerel bir veri tabanında saklanır. Bu kontrol sisteminde sadece bir kullanıcıyı etkin bir şekilde çalıştırmaktadır yani sadece bir kişi dosya üzerinde değişiklik yapılabilir. Dosyada yapılan güncellemeler başka bir ekip üyesine atıldığında ekip üyesi dosyanın en eski halini verisine erişememektedir.

Günümüzde merkezileştirilmiş ve dağıtık kontrol sistemleri kullanılmaktadır.

1. Git Versiyon Kontrol Sistemi:

Optimizasyon prensipli bir versiyon kontrol sistemidir. İşlemci haberleşmesindeki mimari yapı gelen veriyi referanslayıp dosyayı kaydetmek yerine referansladığı yerin bilgisini depolar. Git öncelikle gönderilecek dosyayı hazırlar daha sonra bu dosyanın bellekte yer edecek kopyasını geçici alana dahil eder, bu geçici alandaki bellek kopyalarını kalıcı olarak git sunucusuna depolar. Git gibi versiyon kontrol sistemi kullanan yazılım geliştirme yaşam döngüsünde ekiplerin tercih edebileceği web depolama sistemlerinden iki tanesi de GitHub ve GitLab'dır. GitHub açık kaynaklı projeler için tercih edilen en popüler depolama servisi. GitHub'ın içindekilerinin tümünü sunan açık kaynak projeleri ücretsiz bir şekilde oluşturulabilecek ve yönetebilecek Git servisine GitLab denir. Amaç açık ortama sahip olmaksa GitLab, daha fazla tesis ve web hizmetinin önemli olduğu bir projeyse GitHub kullanılabilir.

Git Versiyon Sistemi Avantajları:

- Dağıtık yapıda olması sebebiyle kullanıcının kendi yerel kopyasını tutabilmesi, bağımsız, ağ bağlantısı olmadan çalışabilmesi.
- Kolayca branch oluşturulabilmesi ve farklı branchlarda çalışabilme imkanı, farklı branch birleştirme işleminin kolaylığı.
- Her değişiklik için geçmiş kaydı tutmasıyla hataların giderilmesini kolaylaştırması.
- Hızlı ve verimli bir yapıya sahip olması.
- GitHub ve GitLab sayesinde projelerin diğer geliştiricilerle paylaşılmasının kolay olması.

Git Versiyon Sistemi Zorlukları:

- Dosya kilitleme işlevselliğinin olmaması.
- Kullanıcının repository'e tam erişimi olmasından kaynaklı güvenlik sorunlarının çıkma ihtimalinin bulunması.

Git Windows Kullanımı:

Git, Windows işletim sistemine yüklenir ve kurulumu yapılır. Git deposu oluşturmak için proje klasörü sağ tık ile seçilerek "Git Bash Here" seçeneği seçilir böylelikle git komut istemi açılır. Komut istemi açıldıktan sonra proje klasöründe "git init" komutu çalıştırılarak bir Git deposu başlatılır. Git'e hangi dosyaların izlenmesi gerektiği söylenmesi gerekmektedir bunun için "git add dosya_adi" komutu kullanılabilir. Değişiklikleri onaylamak için "git commit-m "değişiklik açıklaması"" komutu kullanılabilir. Uzak bir depo eklemek için "git remote add origin uzak_depo_ursi" komutu kullanılır. Uzak depoya dosyaları ve onaylanmış değişiklikleri göndermek için "git push origin ana_dal" komutu kullanılır. Böylelikle proje yedeklenmesi veya diğer geliştiricilerle iş birliği imkanı sağlanmış olur.

2. Mercurial Versiyon Kontrol Sistemi:

Python ile yazılmıştır ve yazılım geliştiricilere yöneliktir. Merkezi olmayan bir yapıda olan bu sistem bütünleşmiş bir web arayüzüne sahiptir. Gelişmiş dallanma ve birleştirme olanakları vardır. Yüksek performanslıdır ve ölçeklenebilirlik mevcuttur.

Mercurial Versiyon Kontrol Sistemi Avantajları:

- Hızlı ve güçlü bir yapısının olması.
- Öğrenmesinin kolay olması.
- Esnek bir yapısının olması.
- Ücretsiz olması.

Mercurial Versiyon Kontrol Sistemi Zorlukları:

- Tüm eklentilerin pythonla yazılmış olması.
- Kısmi checkoutlara izin vermemesi.
- Ek uzantılarla kullanımının sorunlu olması.

Mercurial Versiyon Kontrol Sistemi Windows Kullanımı:

İlk olarak, Mercurial istemcisi indirilir ve bilgisayara kurulumu yapılır. Mercurial iki ana yöntem ile Windows üzerinde kullanılabilir. İlki, komut satırı arayüzünü kullanma; Windows'un komut sistemi veya PowerShell gibi bir komut satırı uygulamasını açarak Mercurial komutları doğrudan kullanılabilir. İkincisi ise Mercurial'ın grafiksel kullanıcı arayüzü olan TortoiseHg kullanmaktır. Bu işlem Mercurial işlemlerinin daha kolay gerçekleştirilmesini sağlar. Mercurial Deposu oluşturmak için projenin bulunduğu bir klasörde komut isteminde "hg init" komutu kullanılabilir. Bu komut, mevcut klasörü bir Mercurial deposu olarak işaretleyecektir. Projenin bulunduğu klasörde komut istemini açarak çeşitli mercurial komutları kullanılabilir. Örnek verilecek olunursa "hg add" komutu dosyaları depoya ekler. "hg commit" komutu ile değişiklikler onaylanır, "hg push" komutu ile değişiklikler uzak bir sunucuya gönderilir. Mercurial, sürüm geçmişi görüntüleme dal oluşturma birleştirme gibi işlemleri de destekleyen bir sistemdir ve bu işlemler komut istemi üzerinden gerçekleştirilebilir.

3.CVS Versiyon Kontrol Sistemi:

Popüler kontrol sistemlerinden biridir. Eş zamanlı sürüm oluşturma sistemi olarak da bilinmektedir. Dağıtık olmayan ortamlarda kullanılır. Müşteri-sunucu mimarisiyle çalışmakla beraber birden çok geliştirici aynı proje üzerinde paralel olarak çalışabilme olanağı sağlar. Verimli depolama için delta sıkıştırma tekniğini kullanır ve projelerin farklı dallarını destekler. Modern versiyon kontrol sistemlerinin öncüsü niteliğindedir.

CVS Versiyon Kontrol Sistemi Avantajları:

- CVS topluluğu geniş olmasından kaynaklı destek almak oldukça kolaydır.
- Çapraz platform uygulaması vardır.
- Kaynak kod deposunda web taramasına izin vermektedir.
- Eski olması nedeniyle iyi bilinen ve anlaşılan bir sistemdir.
- Güçlü bir komut dosyası oluşturulabilir.
- Açık kaynaktır.
- Ücretsizdir

CVS Versiyon Kontrol Sistemi Zorlukları:

- Kaynak kod deposu için bütünlük denetiminin olmaması.
- 2008'den bu yana güncelleme yapılmaması.
- Projelerde sınırlı ekip üyesi ve kullanımı sunması.
- Atomik kontrolleri ve taahhütleri desteklememesi.
- Dağıtılmış kaynak denetimi için zayıf destek sunması.
- İmzalı versiyonları ve birleştirme izlemeyi desteklememesi.

CVS Versiyon Kontrol Sistemi Windows Kullanımı:

Cvs istemcisi indirilir ve bilgisayara kurulumu yapılır. İşlemcinin arayüzünde connect seçeneğine tıklanarak CVS deposunu bağlama işlemi yapılır. Depo bilgileri girilir. Bu depo bilgileri kullanıcı adı ve paroladan oluşur. Daha sonra "get" seçeneği seçilir böylelikle projenin son sürümü bilgisayara indirilmiş olur. Yapılan değişiklikleri geri CVS deposuna aktarmak için

“submit” seçeneği kullanılır. Güncellemeleri almak veya kopyayı güncellemek için içinde “update” seçeneği kullanılır.

4.SVN Versiyon Kontrol Sistemi:

Apache Foundation tarafından geliştirilen, istemci-sunucu veri havuzu modeli sunan, dizinlerde versiyonlanabilir bir versiyon kontrol sistemidir. Kopyalama, silme, taşıma ve yeniden adlandırma işlemleri de versiyonlanmıştır. Atomik comitleri destekleyen bir yapısı vardır. Alan verimli iki farklı depolama alanı sunan ve birleştirme izleme, tam MIME desteği, yola dalı yetkilendirme dosya kilitleme, bağımsız sunucu çalışması gibi bir ok özelliği destekleyen bir yapısı vardır.

SVN Avantajları:

- Boş dizinleri destekler.
- Kurulumu ve yönetimi oldukça kolaydır.
- İyi bir Windows desteği vardır.
- Açık kaynaktır.
- Stabil sürüm oluşturulabilir.

SVN Zorlukları:

- Dosyaların değişiklik zamanını saklamaması.
- İmzalı revizyonları desteklememesi.
- Çevrimdışı çalışma desteğinin olmaması.
- Branch birleştirme konusunda Git’e kıyasla daha yavaş olması.

SVN Windows Kullanımı:

İlk olarak bir SVN istemcisi internet üzerinden indirilir ve Windows üzerine kurulur. SVN istemcisi olarak popüler seçeneklerden biri TortoiseSVN tercih edilebilir. Daha sonra proje yönetimi için bir depo oluşturulması gerektir. Depoyu oluşturmak için sunucu tarafında SVN sunucusu kurulmalı veya mevcut bir SVN sunucusuna erişimin olması gerektir. Proje kontrolü için projenin bulunduğu klasöre sağ tıklanıp TortoiseSVN menüsü açılır. Bu menüde klasördeki dosyaları ve alt klasörleri SVN deposuna eklenmesini sağlayan bir seçenekler vardır. Bu seçenekler kullanılarak SVN deposuna projeler eklenebilir. Var olan bir projeyi güncellemek için projenin klasöründe sağ tıklayarak “SVN update” seçeneği seçilir. Belirli bir versiyona geri dönmek için ise SVN update seçeneğini seçtikten sonra “Update to Revision” seçeneği kullanılabilir.

5.TFS Versiyon Kontrol Sistemi:

Team Foundation Server’ın kısaltmasıdır. Microsoft’ a ait bir versiyon kontrol sistemidir. Sunucu-müşteri, dağıtılmış depo modeline dayanmakla beraber özel bir lisansı vardır. Kaynak kod yönetimi, proje yönetimi, raporlama, otomatikleştirilmiş derlemeler, test etme, sürüm yönetimi ve gereksinim yönetimi gibi yaşam döngüsünü destekler.

TFS Versiyon Kontrol Sistemi Avantajları:

- Kolay yönetilebilen bir sistemdir.
- Dallanma ve birleştirme işlemlerinde kolaylık sağlar.

TFS Versiyon Kontrol Sistemi Zorlukları:

- Birleştirme çakışmaları yaşanma durumu mevcuttur.
- Merkezi depoya sürekli bağlantı gereklidir.
- Dallara ayırma gibi çeşitli işlemlerde yavaş kalır.

TFS Versiyon Kontrol Sistemi Windows Kullanımı:

TFS sunucusu kurulumu yapılır. TFS sunucusu kurulumu için Microsoft'un resmi belgelerine başvurulmalıdır. TFS sunucusu kurulduktan sonra TFS sunucusunun web arayüzü kullanılarak yeni bir proje oluşturulabilir. Daha sonra TFS deposuyla iletişim için TFS istemcisinin kurulması gerekir. İstemci kurulduktan sonra, TFS sunucusuyla bağlantı kurulması gerekir. TFS istemcisini yapılandırırken TFS sunucusunun adresinin belirtilmesi gerekir. Visual Studio IDE'sinde veya Team Explorer aracılığıyla TFS istemcisini yapılandırabilir ve TFS sunucusuyla bağlantı kurulabilir. TFS istemcisi aracılığıyla TFS deposunda yer alan bir proje üzerinde çalışılır. TFS istemcisini kullanarak projeyi indirebilir, değişiklikleri onaylayabilir, branch oluşturabilir, merge işlemleri gerçekleştirilebilir ve diğer sürüm kontrol işlemleri yapılır.

Sonuç olarak versiyon kontrol sistemlerini kullanmak birden fazla kişi çalıştırılan projelerde geliştirmenin hızlanmasını, ekip üyelerinin en son sürümle çalışmasını, kimin neyi ne zaman değiştirdiğinin bilinmesini, proje geliştirilirken başlanılan yeri anlamak için eski ve yeni kod arasında karşılaştırmayı, projelerde hatayla karşılaşılması durumunda eski koda dönülebilmesini sağlar ve projede ona uygun olan versiyon kontrol sistemini seçmek oldukça önemlidir. Versiyon kontrol sistemlerini kullanmadan da bir proje oluşturmak mümkündür fakat bu işi yavaşlatmakla beraber zorlaştırmış olur.

Kaynakça:

- <http://tanertemelce.blogspot.com/2013/02/cvsnt-kurulumu-ve-konfigurasyonu-cvs.html>
- <https://www.nongnu.org/cvs/>
- <https://continium.io/en/blog/devops-donusumu-sifir-noktasi-neden-git-tercih-etmelisiniz/>
- <https://bidb.itu.edu.tr/sevir-defteri/blog/2019/02/13/git>
- <https://www.argenova.com.tr/en-iyi-versiyon-kontrol-sistemleri>
- <https://halilozel1903.medium.com/git-github-ve-gitlab-nedir-e51a87e8e4e7>
- <https://basaransuleyman.medium.com/versiyon-kontrol-sistem-nedir-vcs-faydalar%C4%B1-vcs-t%C3%BCrleri-git-nve-git-komutlar%C4%B1-4facd044fed7>