BİL 102 Yazılım Mühendisliği Temelleri Ödev#2

VKS (Versiyon kontrol sistemleri)

Damla Nur Alper

220601017

İçindekiler:

* VCS TANITIMI
* VCS BÖLÜMLERİ
* BÖLÜMLERİN KARŞILAŞTIRILMASI
* ÖNEMLİ KONTROL YAZILIMLARI
* İŞLETİM SİSTEMLERİYLE UYUMLULUK

***VCS***

Versiyon kontrol sistemleri belirli versiyonların daha sonra çağırılabilmesi için zaman içinde yazılım geliştirmedeki anlamından bağımsız olarak bir doküman, bir dosya üzerindeki değişiklikleri(düzenleme, silme, ekleme, isim değiştirme, kopyalama vb.) kaydeden yöneten sistemlerdir. Bu dökümanlar, herhangi bir ofisin belgeleri, web siteleri veya uygulama programları bile olabilir.

Örneğin bu belgelerde versiyon kontrol için yazılım kodları kullanılacaktır ama versiyon kontrol bilgisayardaki herhangi bir dosya türü için de kullanılabilmektedir. Genel olarak yazılım projelerinde tercih edilse de bir kitap yazma gibi bir uygulama için de tercih edilebilir. Yazan kişi kitabı veya başka bir şey için dallar(branches) oluşturabilir ve bu dallar altında sürümler olarak yürütebilir. Tasarımcılar da oluşturduğu varyasyonları ve bu varyasyondaki değişikleri versiyonlar şeklinde yönetebilir. Böylece versiyonlama sayesinde büyük ekip çalışmalarında ya da uzak çalışmalarda tasarımcılar, geliştiriciler vb. farklı çözümlemelere odaklanabilirler. Çünkü bu değişiklikler kolayca karşılaştırılabilir, aşamalandırılabilir ve yanlış giden durumlarda (dosya içeriğinin yok olması, sürümler arası geçiş gereksinimi vb.) ise istenen kısma dosya veya proje aracılığıyla geri dönebilirler.

Versiyon kontrol sisteminin ne işe yaradığına baktıktan sonra şimdi bu yapının çeşitlerini açıklayabiliriz. Bu sistem üç bölüm olarak incelenir:

* Merkezi kontrol sistemleri
* Dağıtık kontrol sistemleri
* Yerel kontrol sistemleri

Merkezi kontrol sistemleri

Merkezi modelde bir tane versiyon kontrol sunucusu bulunur. Bu sunucu içinde versiyon veri tabanını bulundurmaktadır. Bu sunucu projenin çalıştığı diğer bilgisayarlar ile iletişim içinde olmaktadır. Bu sistem daha çok beraber çalışma sürecindeki dosya kontrolleri için daha geniş kapsamlı bir çözüm olarak CVCS geliştirilmiştir. Merkezi kontrol sistemleri yıllardan beri sürüm kontrol sisteminde standart yöntem olarak kabul görmüştür.

Bu sisteme örnek olaraktan CVS, subversion, perforce gibi sistemler örnek gösterilebilir. Sunucular sürüm kontrolüne alınan bütün dosyaları kaydeder. Dosyalar bu sunuculardan istemciler aracılığıyla çekilerek alınırlar (check out). Kısaca merkezi yönetim sistemlerinde asıl amaç elimizdeki versiyonun uzak sunucu ile aynı olmasıdır. Proje bittiğinde iki taraftada da aynı versiyon bulunacaktır. Bu sistemlerde tek depo(repository) ve birçok kullanıcı vardır. Merkezi olmayan sitemlerde de versiyonlama iki kademe şekilde olur. Bunlarda ise birçok yerel depo(local repository) ve birçok kullanıcı ile birlikte bir tane de uzak depo(remote repository) bulunur. Bu sistemlerde commit işlemi bir tek yerel depoyu etkiler, bu commit işlemi de “push” eylemi ile bütün revizyonları sırasıyla uzak depoya uygular. Merkezi olmayan sistemlerde Uzak depodaki versiyonun yerel depoya indirilmesi işlemine de “pull” denir.

Önemli kontrol yazılımları:

-CVS

-SVN

-Git

Merkezi kontrol sisteminin önemli versiyon kontrol yazılımlarından biri olan CVS en eski sürüm kontrol sistemlerinden biridir. Hem ticari hem de açık kaynaklı geliştiriciler arasında iyi bilinen bir araçtır. Diğer versiyon kontrol sistemleri SVN, Git vb.) CVS özellik ve eksiklikleri temel alınarak geliştirilmiştir.

SVN, Git de önemli versiyon kontrol yazılımlarındandır. SVN (subversion) açık kaynaklı versiyon kontrol sistemidir. Bu sistem CVS örnek alınarak yapılmıştır, amacı da cvs’de daha iyi bir VCS oluşturmak olduğundan birçok CVS özelliğine sahiptir. Kopyalama, silme, değişiklik işlemlerinde SVN tarafından yeni versiyonlar oluşturulur. Bu işlemler ya hep ya hiç prensibiyle gerçekleştirilir. Subversion versiyon kontrol serveri olarak da görev yapabilir.

Git ise merkezi olmayan versiyon kontrol sistemlerinin en iyi örneklerinden biridir. Git yerel depolar ile çalışan, fonksiyonel açıdan SVN e benzeyen bir sistemdir. Ayrıca internete bağlı olmadığınız sürede kendi değişikliklerinizi aşama aşama versiyonlamak mümkündür. SVN’i anlamak Git’i anlamaktan biraz daha kolaydır çünkü SVN dosyaları daha düzenli ve organizedir. Git de birden fazla özellik ekleyerek dosyaları değiştirebilme, çalışma saklayabilme, ucuz lokal dallanma sağlamak mevcuttur. SVN e göre küçük ve hızlıdır. Git, disk alanını SVN e göre daha verimli kullanır. Sosyal bir ağa benzeyen “github” servisini sunar, insanlar burada benzeri çalışma ve projeleri yazan yayımlayan insanları, projeleri bulabilirler.

Dağıtık kontrol sistemleri

Dağıtık sürüm kontrol sistemleri ( Distributed Version Control Systems / DVCS) ‘nde projeye dahil olan bilgisayarlar içinde versiyon veri tabanı bulunur ama projenin tüm versiyonlarını barındırmak için merkezi bir sunucu üzerinden işlem yapmaz. Bir DVKS’de kullanıcılar sadece dosyaların son görünümlerini denetlemeyip daha çok depoyu deponun tam tarihiyle birlikte yansıtırlar. Eğer herhangi bir sunucu devre dışı kalırsa birbiriyle o kullanıcı aracılığıyla iş birliği yapan sistemlerde herhangi bir istemci deposu sunucuyu yenilemek için geri yüklenebilir. Bu sistemlerin çoğu uzak depoyla çalışabilir o yüzden bu sistemi kullanırken farklı ekiplerden insanlarla farklı yollarla eş zamanlı iş birliği yapma olanağı sağlar.

DVCS’ nin CVCS e kıyasla avantajları ve dezavantajları şu şekildedir:

- internet bağlantısı olmadan kullanıcılarla ilgili çalışabilirler

- DVCS işlemleri(commit, işlem geçmişi görüntülenmesi, değişikliklerin geri alınması) daha hızlı işler, merkezi sunucuyla iletişime gerek kalmaz çünkü.

- private çalışma alanları oluşturmak mümkündür.

- herhangi bir donanım hatasından etkilenmezler çünkü kopyalar aynı zamanda remote backup görevi üstlenirler.

-bir reponun checkout edilmesi merkezi sistemlere göre yavaştır çünkü tüm dallar(branch) ve revizyon geçmişi varsayılan olarak yerel makineye kopyalanır.

Yerel kontrol sistemleri

Bir uygulama veya arayüz tasarımı vb geliştirdiğimizde dosyaları ve tabi bu süreç içinde ortaya çıkabilecek varyasyonları tutmanın en mantıklı yollarından biri dosyaları türlerine göre tarih ve sıra bilgisi içerecek şekilde düzenlemektir. Tüm sorunlardan ötürü uzun bir zaman önce geliştiriciler yapılan değişiklikleri gözden geçirilebilir kısımlar halinde basit veritabanı üzerinde tutan yerel versiyon kontrol sistemini geliştirdiler. En popüler VKS aracı RCS sistemiydi ki bu sistem hala birçok bilgisayarda kullanılmaya devam etmektedir.

RCS dosyalar üzerindeki farklılıkları disk üzerinde özel bir formatta tutarak çalışır, daha sonra bunları bir araya getirerek istediğimiz bir zamanda istediğimiz dosyanın görünüşüne bakarak onu tekrar oluşturabilir.

Son olarak bu popüler kontrol sistemlerinden hangilerinin Windows, macos işletim sistemlerini desteklediğine bakabiliriz:

Git; kökeni Linux tabanlı bir topluluk olsa da Windows tarafındaki gönüllüler sayesinde Windows dünyasına ve aynı zamanda da Macos’a da kazandırılmıştır.

CVS; açık kaynak olarak geliştirilmiştir. SVN ise CVS en büyük rakiptir ama Git e kıyasla daha iyi Windows desteğine sahiptir. Önde gelen IDE ve çevik araçlar olan Windows ile iyi bir şekilde bütünleşir.

Mercurial; Python ile yazılmış bir revizyon kontrol aracıdır, desteklediği işletim sistemleri Unix benzeri, Windows ve macOS tur.

Monotone; c++ ile yazılmıştır ve desteklediği işletim sistemleri Unix, Linux, BSD, macOS, ve Windows’u içerir.

Bunun gibi benzeri bir sürü kontrol ürünü işletim sistemlerine destek sağlamaktadır.

Medium link :

<https://medium.com/@damlanuralper19/vcs-version-control-systems-d55a15f5cf9a>

gitHub link: