VKS Nedir?

Versiyon kontrol sistemleri (VKS), yazılım geliştirme sürecinde, yazılım dosyalarının, kaynak kodunun veya dokümantasyonun farklı sürümlerini yönetmek için kullanılan bir yazılım aracıdır. VKS, yazılım geliştiricilerine, bir projenin geçmişinde yapılan değişikliklerin tarihçesini takip etme, eski bir sürüme geri dönme gibi imkanlar sunarken aynı zamanda paralel geliştirme, çatallanma ve birleştirme gibi işlemleri kolaylaştıran bir mekanizmalar da sunar. Bu mekanizmalar ve imkanlar bize şu şekilde fırsatlar sağlar. İlk olarak gelişme sürecini çok daha hızlı tamamlamamızı sağlar. Ayrıca Proje geliştirme aşamasında, bulunduğumuz konuma nereden geldiğimizi anlamak için eski ve yeni kodumuz arasında karşılaştırma yapmamızı sağlar. Bunlarla birlikte; Açık kaynaklı projeler baz alınarak geliştirilecek yeni projelerde, süreci kolaylaştırmayı sağlar. Ve en önemlisi projede hatayla karşılaştığımız durumlarda eski kod kaydına dönmemizi sağlar. Bu bizi daha hatasız ve sağlam bir ilerleme şansı sunar. VKS, genellikle bir merkezi depolama yerine bir dağıtık depolama kullanır. Merkezi VKS'de; tüm kaynak kodu bir merkezi sunucuda saklanır ve geliştiriciler, bu sunucuya bağlanarak kaynak kodu güncelleyebilirler. Ama dağıtık VKS’de bu durum farklı ilerler. Dağıtık VCS, her geliştiricinin kendi bilgisayarında bir klon oluşturduğu ve her bir klonun geçmiş kaydı içerdiği bir versiyon kontrol sistemidir. Bu kopyalar, geliştiricilerin yerel olarak değişiklik yapmasına izin verir. Daha sonra bu değişiklikler, diğer geliştiricilerle paylaşılmak üzere merkezi bir depoya gönderilebilir. Bu özelliği ile geliştiricilerin birbirinden bağımsız olarak çalışmalarına imkân sağlar. Ayrıca ortak çalışmaya da oldukça fayda sağlar. Merkezi sunucunun olmadığı durumlarda bile çalışma süreci aksamaz. Uzak veya offline durumlarda çalışma devam eder. Ayrıca Dağıtık VKS, Merkezi VKS’lere kıyasla daha güvenli bir ortam kurar. Her geliştirici, kendi klonunda çalıştığı için, merkezi sunucu çökse bile herkes hala kendi çalışmalarına devam edebilir. VKS’nin avantaj ve dezavantajlarından bahsedecek olursak. Geçmiş kaydını barındırması ilk ve büyük avantajlarından biridir. Geçmiş kaydı sayesinde geçmişte yapılan değişikliklerin nedeni, kim tarafından yapıldığı ve ne zaman yapıldığı gibi bilgilere kolayca erişebilirsiniz. Bir diğer avantajı ise Geri Dönüşüm şansıdır. Geri Dönüşüm sayesinde bir önceki sürüme geri dönme şansınız bulunmaktadır. Bu, yanlışlıkla yapılan bir değişikliği geri almak, hataları düzeltmek veya daha önceki bir sürümde çalışan kodu kullanmak gibi birçok durumda faydalıdır. Kolay iş birliği imkânı da VKS’nin avantajlarından biridir. Birden fazla geliştirici aynı proje üzerinde beraber çalışabilir. Ayrıca farklı klonlar oluşturarak yerel değişiklikler yapabilir ve bu değişiklikleri ekiple paylaşabilir. Bir diğer avantajı ise paralel geliştirme imkanıdır. Birden fazla geliştirici bir önceki avantajda bahsedilen gibi farklı parçaları aynı anda geliştirebilir. Bu sayede geliştirme süreci hızlanır ve projeyi tamamlamaya yakın parçaları birleştirmek çok daha kolay olur. Son ve önemli avantajlarından biri Takım yönetiminin olmasıdır. Paralel geliştirme ile birlikte proje geliştirme aşamasında görev atamaları yapmaya imkân sağlar. VKS’nin birtakım dezavantajları da bulunmaktadır. Bunlardan ilki ; VKS öğrenme eğrisinin çok yüksek olmasıdır. Yeni başlayan geliştiriciler için karışık terimler ve kavramlar barındırır. Bu da yeni başlayanlar için öğrenmeyi zorlaştıran bir durumdur. Bir diğer dezavantajı; projelerin kopyalarını saklamanın büyük miktarda veri yedeklemesi gerektirmesidir. Bu problem, disk alanı ve bant genişliği problemlerine neden olabilir. Dezavantajları arasında merkezi sunucu hizmetinin gerekliliği de bulunmaktadır. Merkezi sunucun çökmesi veya çalışmayı durdurması, geliştirilen projede ciddi aksaklıklara sebep olabilir. Ayrıca merkezi sunucu ile yaşanabilecek bağlantı sorunları da proje gelişimine zarar verebilir. Paralel geliştirme bize birçok fayda sağlarken bir de yanında bir dezavantaj getirir. Bu da yaratacağı karmaşıklıktır. Projenin büyümesi ile birlikte; geçmiş yönetimi ve proje birleştirme işlemleri çok daha karmaşık hale gelecektir. Son bahsedeceğim dezavantaj ise iletişim sorunlarıdır. VKS’de proje geliştirmek için düzgün bir iletişim ve iş birliği kültürü oluşturmak önemlidir. Geliştiriciler arasındaki iletişimi teşvik etmek, düzenli toplantılar düzenlemek; proje gelişimi açısından çok önemlidir. Son olarak VKS bazı zorluklar barındırır. Dezavantajlarda bahsedilen öğrenme eğrisinin yüksekliği, yeni başlayanlar için zorluk doğurabilir. Temel işlevler ve komutları anlamak oldukça zaman alır. Bir diğer zorluğu ise Çakışan değişiklikler olacaktır. Aynı projeyi birden farklı geliştiricinin geliştirmesi durumunda bazı değişikliklerde çakışmalar meydana gelebilir. Böyle durumlarda değişikliklerin birleştirilip çözülmesi gerekir. Ancak bu aşamada değişikliklerin birleştirilip çakışmaların çözülmesi zorluklara yol açanbilir. Zorluklarda üçüncü sırada Büyük projeleri yönetmek geliyor. Dezavantajlarda bahsedilen karmaşıklık burada büyük bir zorluk doğuracaktır. Proje büyüdükçe daha da karmaşıklaşacak ve ileriki aşamalarda gelen birleştirme işlemini oldukça zor hale getirebilir. Bir diğer zorluğu ise senkronizasyondur. Eş zamanlı ve senkron bir ekip çalışması VKS’nin önemli aşamalarından biridir. Ama aynı dosyalar üzerinde aynı anda çalışmak, bazı sorunlar doğurabilir. Ekip üyelerinin senkronize şekilde hareket etmesi ve koordinasyonlu geliştirmeye devam etmesi çok önemlidir ancak bir o kadar da zordur. Son olarak bahsedeceğimiz zorluk Veri Yedeklemenin güçlüğüdür. Tekrardan dezavantajlar içinde bahsedilen veri yedekleme ve depolama; ciddi miktarda depolama alanı istemektedir. Ayrıca verileri geri yükleme süreçlerini yönetirken de birtakım zorluklar çıkabilir. Bu zorluklar VKS'nin karmaşıklığından ve geliştirme sürecindeki dinamiklerden kaynaklanır. Ancak, iyi bir VKS kullanımı için eğitim, iletişim, koordinasyon ve disiplinli bir çalışma yaklaşımı, bu zorlukların üstesinden kolaylıkla gelebilir. VKS’lere 6 adet örnek vermek gerekirse. İlki Preforce. Preforce; büyük dosyalar için daha uygun olup ölçeklenebilirlik özellikleriyle öne çıkar. Büyük ölçekli ve ticari bir VKS olarak bulunur. İkinci olarak Git’den bahsedebiliriz. Git günümüz dünyasında en popüler VKS’lerin başında gelir. Hızlı, verimli ve esnek yapısıyla öne çıkar. İlk oluşturulma amacı Linux çekirdeği geliştirme sürecidir. Birçok projenin kullandığı açık kaynaklı bir dağıtık VKS türüdür. Üçüncü olarak SVN gelir. SVN birçok büyük projede kullanılmaktadır. Önceki sürümlere dönme, değişikleri takip etme ve merkezi bir sunucuda kodları saklama gibi avantajları bünyesinde barındırır. Dördüncü sırada Mercurial gelir. Python ile yazılmış dağıtık VKS’dir. Kolay ve basit bir ara yüzle öne çıkar. Birçok özelliği bakımından Git’e benzer. Beşinci olarak TFS gelir. TFS, Microsoft tarafından geliştirilen bir VKS’dir. En sık .NET platformunda çalışan projelerde görülür. Bu türde birtakım avantajlar bulunur. Proje takibi, test otomasyonu ve iş süreç yönetimleri; bunlardan bazılarıdır. Son olarak Bitbucket gelir. Çeşitli iş birliği özellikleri barındıran bulut tabanlı bir hizmettir. Git ve Mercurial desteği sağlar. Proje geliştiriciler farklı ihtiyaç ve isteklere göre; projenin büyüklüğüne, ekibin tecrübesine ve isteğe bağlı olarak bir VKS türünden geliştirmeye başlarlar. VKS’ler kod kalitesini artırmak için biçilmiş kaftandır. Peki bu sistemler Windows işletim sisteminde nasıl çalışır? İlk örnek Git olacaktır. Windows işletim sisteminde kullanılabilecek güçlü bir versiyon kontrol sistemidir. İlk olarak Git’i yükleyerek başlayabiliriz. Git resmî web sitesi üzerinden kolaylıkla indirilebilir. Bu siteden indireceğimiz kurulum dosyasını çalıştırıp karşımıza gelen yükleme adımlarını takip ederiz. Kurulum aşamasında; “Git Bash” , “Git GUI” gibi seçenekler de mevcut olacaktır. Bu seçenekler kullanıcı grafiksel bir arayüzü ile kullanma veya Git’i terminal üzerinden çalıştırma gibi imkanlar sağlar. Yükleme işlemi bittikten sonra yapılandırma işlemine geçeceğiz. Yapılandırma aşaması birkaç adımdan oluşmaktadır. Bu adımlar şu şekildedir. İlk olarak Git Bash veya komut istemcisini açmamız gerekmektedir. Açtıktan sonra Adı, Soyadı ve e-posta adreslerinizi ayarlamak için belirli komutlar girmeniz gerekmektedir. Ad ve Soyadı için (git config --global user.name "Adınız Soyadınız") girilecektir. E-posta için (git config --global user.email [eposta@xxxxxx.com](mailto:eposta@xxxxxx.com)) girilecektir. İleride yapacağınız Git işlemlerinde kimliğinizin belirtilmesi için bu ayarlamayı yapmanız gerekmektedir.