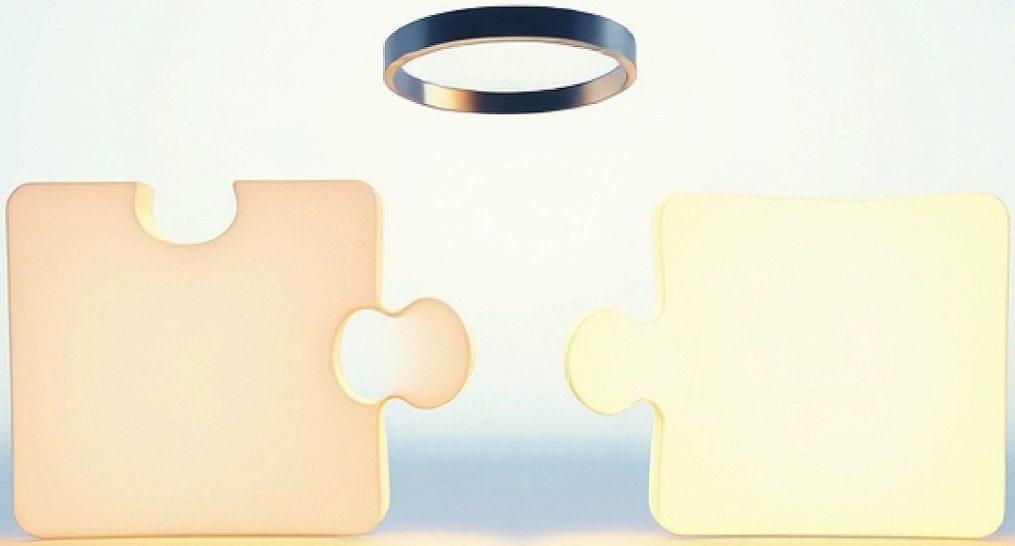
Kod birleştirme çakışması nedir?



www.yzokul.com.tr

Kod birleştirme çakışması nedir?



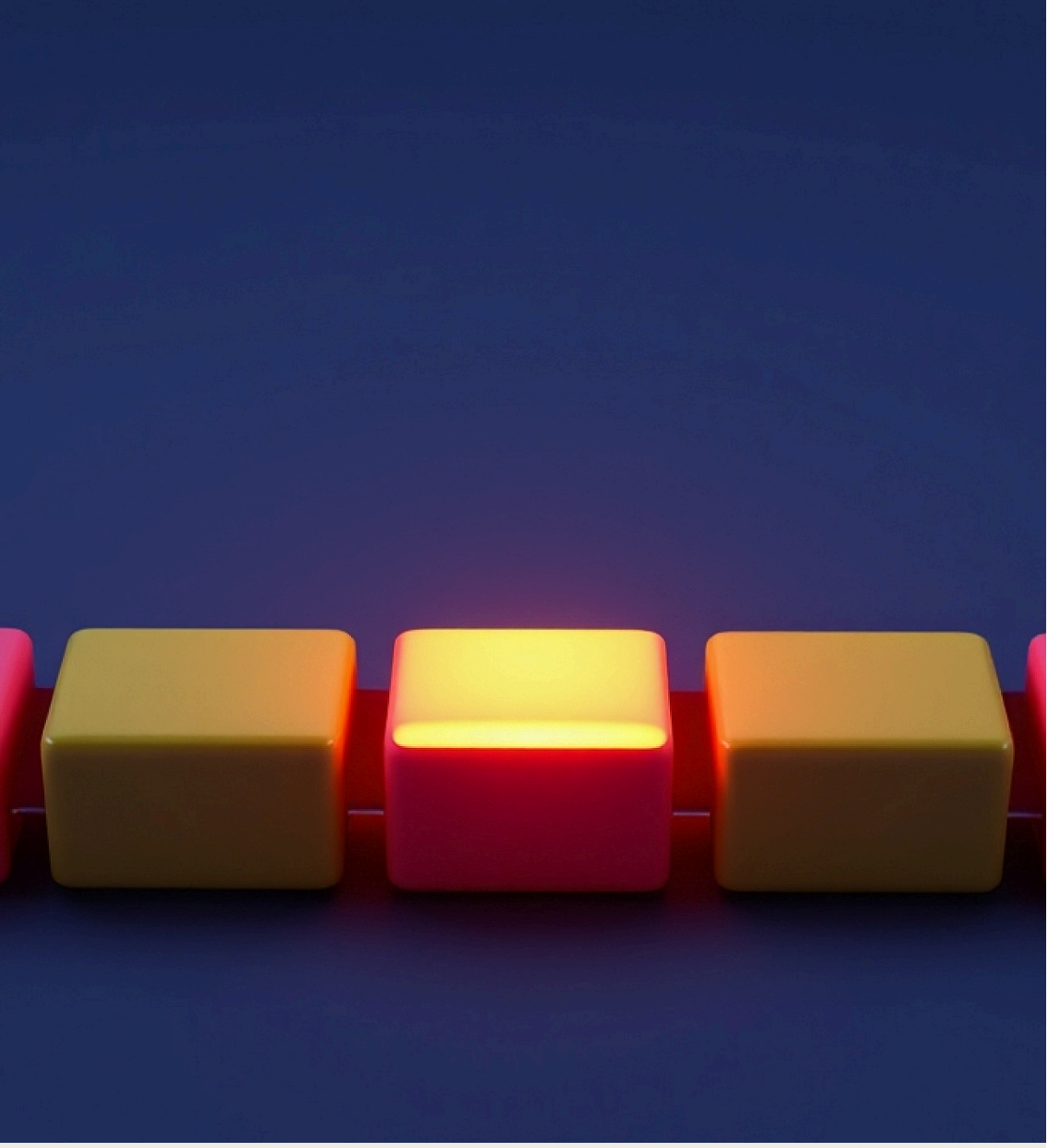
Bir kod birleştirme çakışması, aynı koda yapılan iki değişiklik kümesi otomatik olarak birleştirilemediğinde ortaya çıkar; bu durumda neyin tutulacağına bir kişinin karar vermesi gerekir. Bu yanıt kitabı, konuyu en başından, sade sözcüklerle açıklıyor. Sürüm kontrolünün ve dallandırmanın nasıl çalıştığını, çakışmaların neden ortaya çıktığını, bir çakışmanın elle nasıl çözüleceğini ve sonrasında çalışmanın nasıl kontrol edileceğini göreceksin. Kısa yanıt: çakışma, bir felaket değil, insan kararına duyulan bir istektir.



Kod, bilgisayara ne yapacağını söyleyen, dosyalarda saklanan metinden ibarettir. Bir yazılım projesi, çoğu zaman alt klasörlere ayrılmış, bu dosyalarla dolu bir klasördür. Her dosyanın bir adı ve açıp okuyabileceğin bir içeriği vardır. İnsanlar bir proje üzerinde birlikte çalışırken bu dosyaları zamanla değiştirir. Bu değişiklikleri takip etmek bir sürüm kontrol aracının işidir.



İnsanlar çoğu zaman dosyalarının eski sürümlerini, geri dönebilmek ya da karşılaştırabilmek için saklamak ister. Sürüm kontrolü, kaydedilen her sürümü senin için hatırlayan, böylece hiçbir şeyin kaybolmadığı bir araçtır. Klasörleri elle kopyalamak yerine her değişikliği aracın kaydetmesine izin verirsin. Bu, neyin, ne zaman ve kim tarafından değiştiğini görmeyi kolaylaştırır. Ayrıca birkaç kişinin birbirinin çalışmasının üzerine yazmadan aynı proje üzerinde çalışmasını sağlar.



Bir commit, dosyalarının kaydedilmiş bir anlık görüntüsü ve neyin değiştiğine dair kısa bir mesajdır. Her commit, projenin geçmişinde bir yer imi gibi, sonradan geri dönebileceğin bir noktadır. Örneğin, giriş düğmesinin rengini düzelt mesajıyla commit yapabilirsin. Sürüm kontrol aracı bu anlık görüntüyü saklar ve onu bir öncekine bağlar. Zamanla commit'ler, projenin nasıl büyüdüğünü kaydeden bir zincir oluşturur.



Bir dal, bir commit'ten başlayan ayrı bir çalışma hattıdır. İnsanlar, projenin ana sürümünü rahatsız etmeden değişiklikleri deneyebilmek için dal oluşturur. Örneğin, yeni bir tasarımı denemek için yeni-menu adında bir dal oluşturursun. Ana dal olduğu gibi kalır ve senin dalın kendi başına büyür. Çalışma hazır olduğunda onu ana hatta geri getirebilirsin.



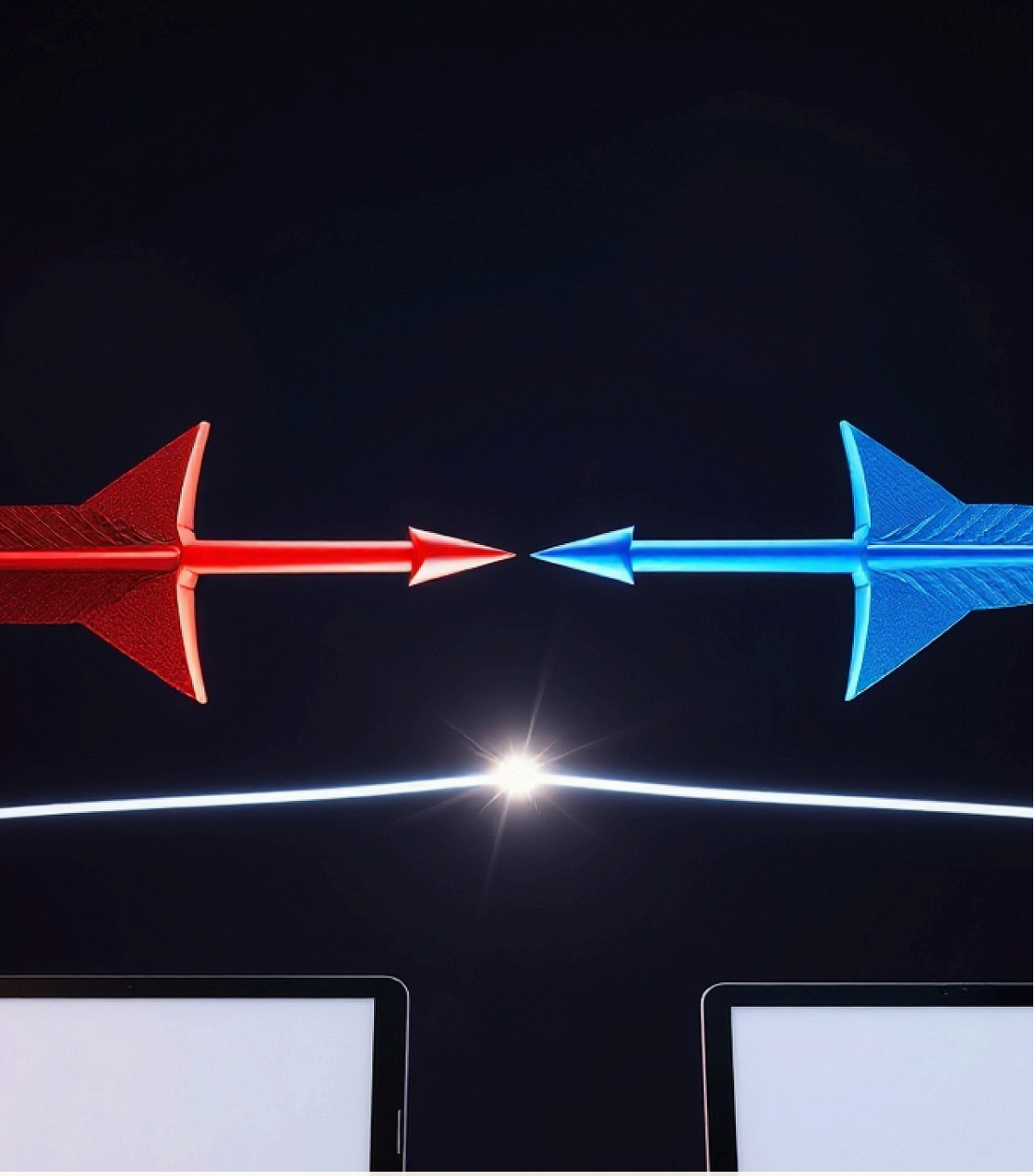
Aynı başlangıç commit'inden büyüyen, her birinin kendi yeni commit'leri olan iki dal düşün. İki geliştirme hattına sahip olmak tam olarak bunu ifade eder. Bir kişi bir dalda bir özellik eklerken bir başkası farklı bir dalda bir hatayı düzeltebilir. İki dal, ayrıldıkları noktaya kadar aynı geçmişi paylaşır. Ondan sonra her dal kendi ayrı değişikliklerini kaydeder.



Bir birleştirme, bir daldaki değişiklikleri alıp bir başkasına ekler. Sürüm kontrol aracı değişiklikleri senin için otomatik olarak birleştirmeye çalışır. Örneğin, herkesin yeni tasarımı alması için yeni-menu dalını ana dalla birleştirirsin. Araç, ayrıldıklarından beri her iki tarafta neyin değiştiğine bakar. Her şeyi temiz bir şekilde birleştirebiliyorsa birleştirme sana hiçbir şey sormadan tamamlanır.



İki dal dosyaların farklı bölümlerini değiştirdiyse araç bunları sorunsuz bir şekilde otomatik olarak birleştirir. Örneğin, bir dal bir dosyanın üst kısmını düzenlerken diğeri alt kısmını düzenler. Araç her iki düzenlemeyi de olduğu gibi tutar, çünkü bunlar çakışmaz. Çoğu birleştirme bu şekilde, sessizce ve hiçbir çakışma olmadan gerçekleşir. Çoğu zaman bir birleştirmenin gerçekleştiğini fark bile etmezsin.



Kilit durum, iki dalın da tam aynı satırı farklı şekillerde değiştirmesidir. Örneğin, bir dal başlığı Hoş geldiniz olarak ayarlarken diğeri Merhaba olarak ayarlar. Araç hangi sürümü istediğini bilemez, çünkü ikisi de geçerli değişikliklerdir. Bu yüzden durur ve senden karar vermeni ister. Bir birleştirme çakışmasına yol açan durum tam olarak budur.



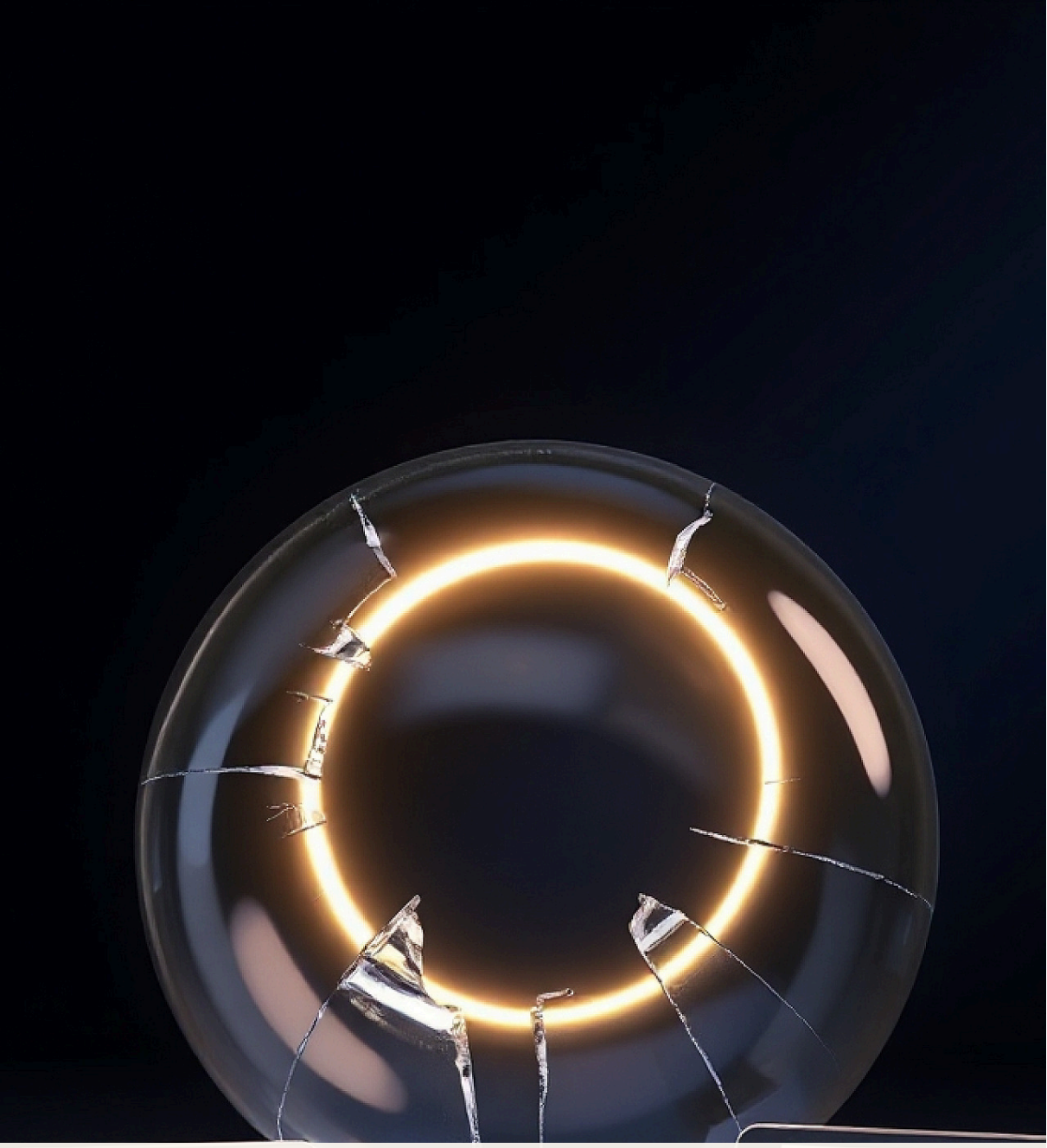
Bu durmaya birleştirme çakışması denir. Bu, aracın bir şeyin bozulduğunu değil, bir insan kararına ihtiyaç duyduğunu söylemesidir. Çakışma sözcüğü dramatik kulağa gelir, ama yalnızca otomatik birleştirmenin tamamlanamadığı anlamına gelir. Hiçbir şey atılmamıştır ve hiçbir çalışma kaybolmamıştır. Araç duraklar ve son dosyanın ne içermesi gerektiğini seçmeni bekler.



Araç, çakışmayı dosyanın içinde özel işaretler kullanarak gösterir. Bir işaret senin sürümünü diğer sürümden ayırır, bir başka işaret diğer sürümü bitirir. Örneğin, bir satır kümesinin üstünde dalının adıyla, bir başka satır kümesinin üstünde diğer dalın adıyla etiketlenmiş bir blok görebilirsin. İki seçeneği de açıkça okuyabilir ve yan yana karşılaştırabilirsin.



Şimdi neyi tutacağına sen karar verirsin. Bir sürümü tutabilir, diğerini tutabilir ya da ikisinin parçalarını yeni bir şeyde birleştirebilirsin. Bu, aracın senin yerine veremeyeceği bir insan kararıdır. Örneğin, diğer dalın başlığını tutup altına kendi sürümünden bir satır ekleyebilirsin. Dosyayı tam olarak istediğini içerene kadar düzenler, sonra kaydedersin.



Bir çakışma, projenin kaybolduğu ya da mahvolduğu anlamına gelmez. Hiçbir şey kalıcı olarak silinmez ve commit geçmişindeki eski sürümlere her zaman geri bakabilirsin. Çözüm sırasında bir hata yaparsan geri alabilir ya da birleştirmeyi yeniden başlatabilirsin. Çakışma yalnızca bir karar için verilen bir moladır. Ona bir başarısızlık olarak değil, dallarla çalışmanın olağan bir parçası olarak bak.



Neyi tutacağını seçtikten sonra programın bir bütün olarak hâlâ doğru çalıştığını kontrol etmelisin. Parçalar tek tek iyi olabilir ama birbirine uymayabilir. Örneğin, bir dal bir işlevi yeniden adlandırırken diğeri hâlâ eski adı çağırıyor olabilir. Her şeyin beklendiği gibi davrandığını doğrulamak için programı ve testlerini çalıştır. Ancak ondan sonra birleştirmeyi tamamla ve sonucu kaydet.



Yeni başlayanların sık yaptığı hatalar arasında işaretleri görünce paniğe kapılıp yanlışlıkla her iki sürümü de silmek vardır. Bir başka hata da sonrasındaki kontrolü atlamak, böylece bozuk bir programın gözden kaçmasıdır. Bazıları da çakışma işaretlerini kendilerinin kaldırmayı unutur ve bunlar dosyada kalır. Bu tuzakları bilmek onlardan kaçınmana yardımcı olur. Sakin kal, iki tarafı da dikkatle oku ve sonucu her zaman test et.



Bir birleştirme çakışması, iki çalışma hattı bir dosyanın aynı bölümünü farklı şekillerde değiştirdiğinde ortaya çıkar. Bir kişi neyin tutulacağına ya da bunun nasıl birleştirileceğine karar verir, sonra tüm programın hâlâ çalıştığını kontrol eder. Proje asla kaybolmaz, çünkü eski sürümler geçmişte kalır. Pratikle, çakışmaları çözmek yazılım geliştirmenin rutin ve yönetilebilir bir parçası hâline gelir.