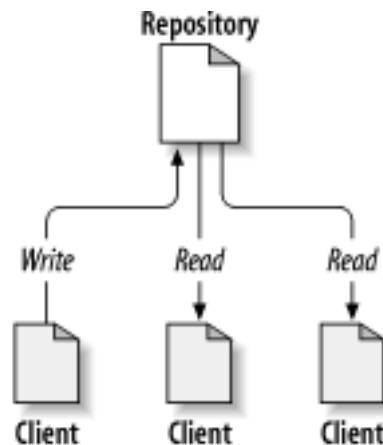


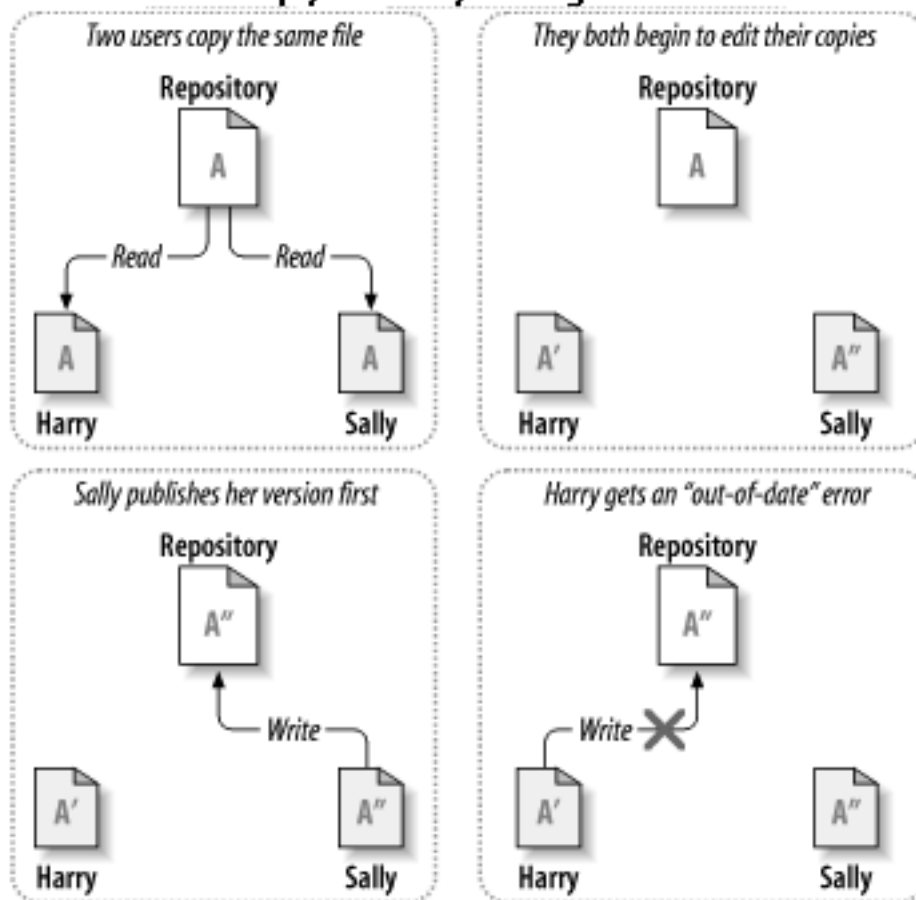
در قسمت قبلی، اهمیت استفاده از سیستم های کنترل نسخه را بیان کردیم و مفاهیم پایه ای گیت را مورد بررسی قرار دادیم. در این قسمت مفاهیم پایه ای SVN را مورد بررسی قرار می دهیم. SVN مخفف عبارت SubVersion هست و یک سیستم کنترل نسخه ای رایگان و متن باز است که توسط شرکت کلاب نت حمایت می شود. به تعدادی از این سیستم ها، سیستم های «مدیریت پیکربندی نرم افزار» (Software Configuration Manager (SCM هم اطلاق می شود.

در این سیستم فایل ها در یک مخزن Repository مرکزی ذخیره می شوند و با هر تغییری که در فایل ها و دایرکتوری ها ایجاد می شود، آن ها را ثبت می کند. این امکان به ما این اجازه را می دهد که نسخه ی قدیمی فایل ها را بازیابی کرده و تاریخچه ی اینکه فایل ها چگونه و چه موقع و توسط چه کسی تغییر کرده اند، به ما نشان دهد. تصویر سلسله مراتبی زیر به خوبی نحوه ارتباط کلاینت ها را به این مخزن نشان می دهد.



SVN برای مدیریت چندین نسخه از فایل ها، از مدل «کپی، ویرایش، ادغام» **Copy-Modify-Merge** استفاده می کند. در این مدل که هر کاربری در مخزن عمل خواندن را انجام می دهد، یک کپی جداگانه و کاملاً شخصی برای او گرفته شده و سپس کپی های شخصی خودش را ویرایش می کند. بعد از اینکه ویرایش تکمیل شد، کپی شخصی خودش را با یک فایل جدید و نهایی ادغام می کند. این روش به شدت از روش «قفل کردن، ویرایش، آزادسازی» «**Lock-Modify-Unlock**» کارآمدتر است و دیگر نیازی نیست که یک کاربر در یک زمان به این ساختار دسترسی داشته باشد و آن را ویرایش کند.

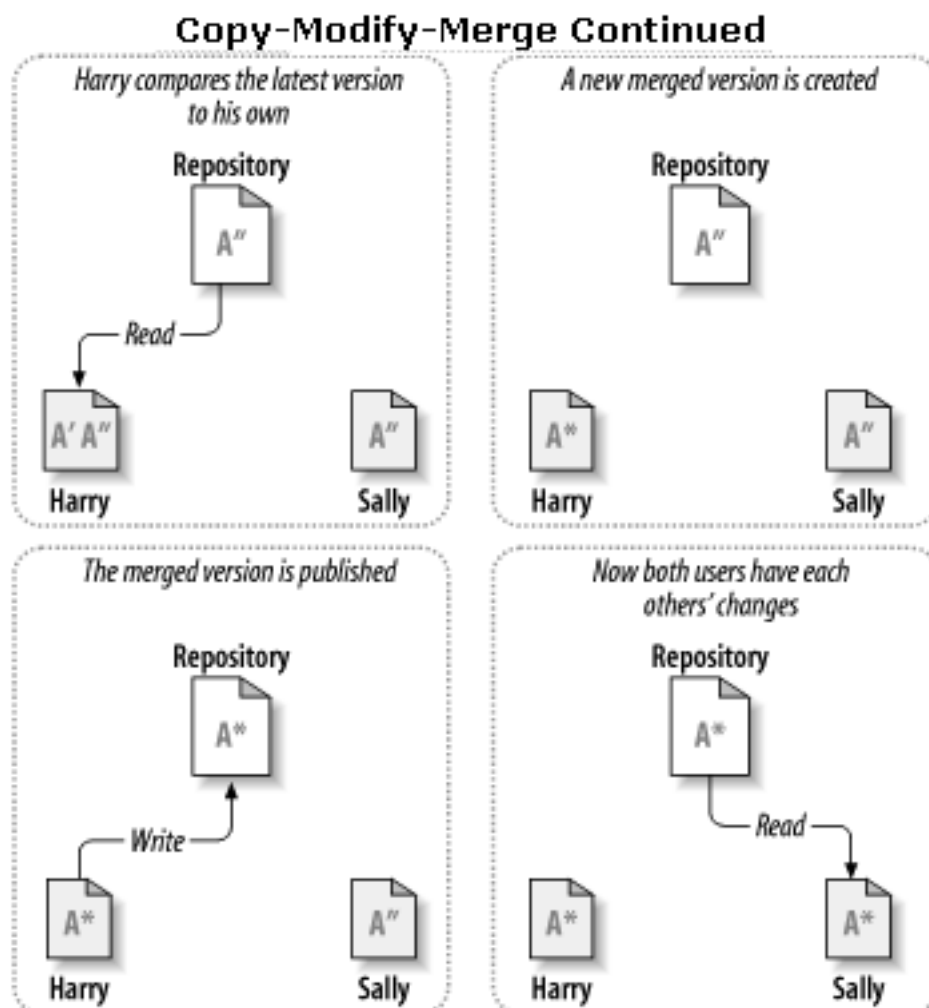
The Copy-Modify-Merge Solution



در تصویر بالا هری و سالی، یک کپی از مخزن موجود را گرفته و سپس هر کدام جداگانه بر روی کپی های خودشان مشغول به کار می شوند. سپس سالی کارش را زودتر به اتمام رسانده و مخزن را به روز می کند. بعد از آن، هری هم کارش به پایان می رسد و قصد به روز کردن مخزن را دارد ولی سیستم به او اجازه این کار را نمی دهد؛ چون این مخزن آن مخزن نیست که هری قبلا از آن کپی گرفته است. آن مخزن بعد از به روزرسانی سالی تغییر یافته است. پس او مجبور است تا تغییرات جدید مخزن را دریافت کرده و کپی خودش را به روز کند. پس از آن می تواند کپی خودش را بر روی مخزن اعمال کند (با فرض اینکه تغییرات جدید هیچ تصادمی با تغییراتی که روی کپی خودش اعمال کرده است ندارند).

سناریو بالا با احتساب وجود تصادم

اگر همین سناریوی بالا را فرض کنیم که تغییراتی که هری روی فایل ها داده است همان تغییراتی است که سالی قبلا روی مخزن اصلی روی همان فایل ها اعمال کرده است، آیا در این حالت دریافت به روزرسانی های جدید باعث ایجاد تصادم می شود؟



هری درخواست ادغام آخرین تغییرات مخزن را با کپی خودش می‌کند. از آنجا که فایل A تصادم دارد یک فلگ *flag* از این وضعیت می‌گیرد. حال هری میتواند تفاوت‌های ایجاد شده را ببیند و بین آنها یکی را انتخاب کند. در این وضعیت هری همپوشانی‌های کدها را برطرف می‌کند و شاید هم بحثی در مورد این تصادم با سالی داشته باشد تا بهترین تغییر کد انتخاب گردد و نهایتاً به روشی کاملاً امن و مطمئن، با مخزن اصلی ادغام می‌شود.

پی نوشت : نرم افزارها نمی‌توانند موضوع تصادم را به طور خودکار اعمال کنند. از آنجا که نیاز به تصمیم گیری و درک هوشمند دارد این کار به صورت انسانی باید بررسی گردد.