

بسمه تعالی



دانشگاه جامع علمی کاربردی

سیستم های کنترل نسخه

Version Control Systems



مدرس : مهندس نوید همراهی

n.hamrahi@gmail.com

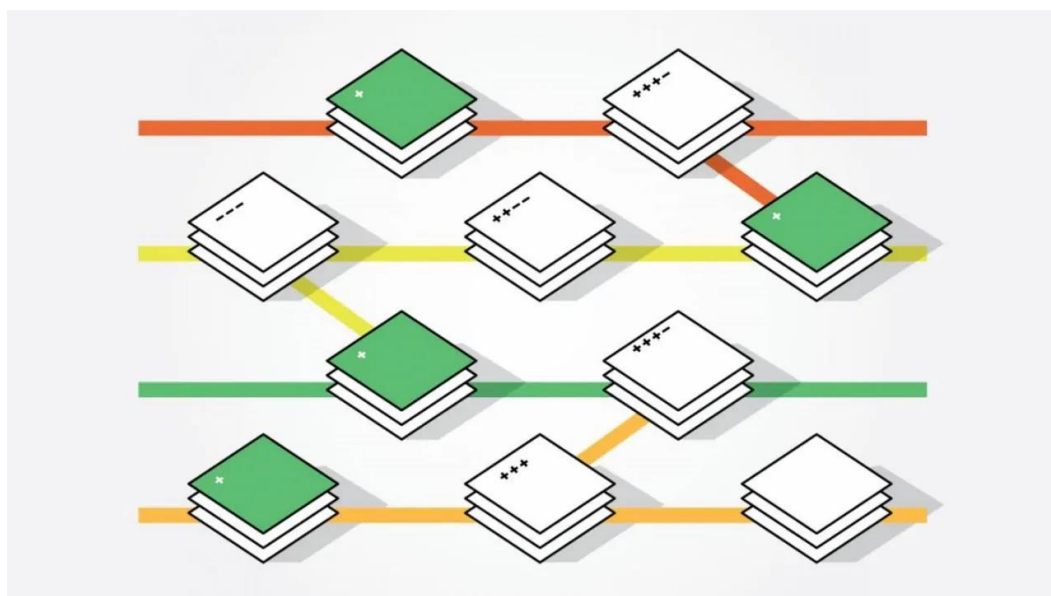
www.navidhamrahi.ir

جهت دریافت جزوه و اطلاع از روند کلاس به وب سایت بالا

بخش « دانشگاه مجازی » مراجعه فرمایید.

سیستم کنترل نسخه چیست؟

می‌توان به جرات گفت که کد منبع تقریباً برای همه پروژه‌های نرم‌افزاری، حکم تاج پادشاهی را دارد و در واقع، یک دارایی بسیار گران‌بها است که باید از ارزش آن محافظت کرد. برای اغلب تیم‌های نرم‌افزاری، کد منبع یک مخزن از دانش و اطلاعات فوق‌العاده گران‌بها پیرامون دامنه مسئله (Problem Domain) است که توسعه‌دهندگان نرم‌افزار، آن را با تلاش‌های خود گردآوری و پالایش کرده‌اند. کنترل نسخه، از کد منبع در مقابل فجایع و تخریب‌های گاه به گاهی که خطای انسانی منجر به ایجاد آن شده است و عواقب ناخواسته آن جلوگیری می‌کند.



«سیستم‌های کنترل نسخه» (Version Control Systems) دسته‌ای از ابزارهای نرم‌افزاری هستند که به یک تیم نرم‌افزاری کمک می‌کنند تا تغییراتی که در طول زمان روی «کد منبع» (Source Code) انجام می‌شود را مدیریت کنند. نرم‌افزار کنترل نسخه، هر ویرایش و تغییری در کد را روی نوع خاصی از «پایگاه داده» (Database) ردیابی می‌کند. بنابراین اگر اشتباهی در تغییر کدها اتفاق بیفتد، توسعه‌دهندگان می‌توانند زمان را به عقب بازگردانند و نسخه‌های اولیه کد را با نسخه دارای مشکل مقایسه کنند. این کار به رفع اشتباهات تیم با کمترین سطح از سختی، کمک می‌کند.

توسعه‌دهندگان نرم‌افزاری که در تیم‌های کاری روی پروژه‌ها کار می‌کنند، به طور مداوم کد منبع‌های جدیدی می‌نویسند و کد منبع موجود را تغییر می‌دهند. کد منبع یک پروژه، برنامه کاربردی یا مولفه نرم‌افزاری در یک ساختار پوشه‌ای یا «درخت فایل» (File Tree) سازمان‌دهی شده است. در یک پروژه نرم‌افزاری، یک توسعه‌دهنده در تیم ممکن است روی ویژگی‌های جدید کار کند، در حالی که توسعه‌دهنده دیگری ممکن است روی اصلاح یک «اشکال» (Bug) با انجام تغییراتی در کد، کار کند. هر یک از توسعه‌دهنده‌ها تغییرات خود را در بخش‌های مختلفی از درخت فایل ایجاد می‌کنند.

کنترل نسخه به تیم‌ها برای حل این نوع از مسائل، پیگیری هرگونه تغییر انجام شده توسط مشارکت‌کنندگان و پیشگیری از وقوع ناسازگاری و تناقض در کارهایی که به صورت هم‌زمان انجام می‌شوند کمک می‌کند. تغییراتی که در یک بخش از نرم‌افزار انجام می‌شوند، ممکن است با تغییرات دیگری ناسازگار باشد که به وسیله توسعه‌دهنده دیگری در همان زمان انجام شده است.

این مسئله را باید به شیوه‌ای منظم و بدون مسدود و متوقف کردن کار بقیه اعضای تیم حل کرد. علاوه بر این، در همه توسعه‌های نرم‌افزاری، هر تغییری ممکن است منجر به ایجاد باگ‌های جدید شود و نرم‌افزار جدید تا هنگامی که تست نشده، قابل اعتماد نیست. بنابراین، تست و توسعه تا هنگامی که یک نسخه جدید آماده شود، با هم به پیش می‌روند.

یک نرم‌افزار کنترل نسخه خوب، از جریان‌های کاری مورد نظر کاربر، بدون تحمیل یک راه خاص کار کردن، پشتیبانی می‌کند. چنین نرم‌افزاری در حالت ایده‌آل، روی هر پلتفرمی کار می‌کند و به توسعه‌دهنده دیکته نمی‌کند که از چه سیستم‌عامل یا ابزارهایی استفاده کند. همچنین، یک سیستم کنترل نسخه خوب، جریان روان و پیوسته تغییرات روی کد را تسهیل می‌کند و فاقد مکانیزم‌های ناامیدکننده و دست و پاگیر قفل کردن فایل است که طی آن‌ها، امکان ویرایش به یک توسعه‌دهنده داده می‌شود و بقیه توسعه‌دهندگان مسدود می‌شوند و امکان کار روی آن بخش از کد را ندارند.

تیم‌های نرم‌افزاری که از هیچ نوعی از کنترل نسخه استفاده نمی‌کنند، دچار مشکلاتی مانند عدم آگاهی از اینکه کدام بخش از تغییرات انجام شده در دسترس کاربران هستند و یا ایجاد تغییراتی

می‌شوند که منجر به ناسازگاری بین دو بخش نامرتبط از کار می‌شود و بنابراین، دادرهای آتی و نیاز به اصلاحات مجدد کد را در پی دارد. اگر فرد یک توسعه‌دهنده وب است که تاکنون هرگز از کنترل نسخه استفاده نکرده، احتمالاً فایل‌های خود را با دادن اسامی و پسوندهایی مانند «Final» و «Last» نام‌گذاری کرده و همین موجب شده تا طی انجام پروژه و پیشرفت آن، فایل‌های «NewFinal.LastFinal» و چنین مواردی را داشته باشد.

برنامه‌نویس ممکن است بلوک‌های کد خود را با این هدف که می‌خواهد کارکردهای خاصی را بدون حذف کد (و شاید به صورت موقت) غیر فعال کند، به صورت «توضیحات» (Comment) درآورده باشد. دلیل این امر گاه این موضوع است که فرد نگران آن است که بعداً به این بخش از کد نیاز پیدا کند و بنابراین آن را حذف نمی‌کند. کنترل نسخه، راهکاری برای حل چنین مشکلات و مسائلی است. نرم‌افزار کنترل نسخه، یک بخش مهم و حیاتی از فعالیت‌های روزانه تیم‌های توسعه نرم‌افزاری است. توسعه‌دهندگان نرم‌افزاری نیز که به تنهایی در کارهای تیمی خود عادت به کار کردن با یک سیستم کنترل نسخه توانمند دارند، معمولاً از ارزش بسیاری زیادی که کنترل نسخه در پروژه‌های کوچک به آن‌ها می‌دهد آگاه هستند. هنگامی که توسعه‌دهنده به مزایای قدرتمند کار با سیستم‌های کنترل نسخه عادت کرد، حتی برای پروژه‌های غیر نرم‌افزاری (مانند نوشتن کتاب، پایان‌نامه و مقاله) نیز از آن استفاده می‌کند.

۵ موردی که باید برای version control systems بدانید!

هر گزینه version control systems دارای ویژگی‌های مختلفی است. مهم‌ترین چیزی که مد نظر شماست به نیازهای تیم شما بستگی دارد. اما به هر حال در این قسمت پنج مورد از بهترین مواردی که تیم‌های توسعه امروز به دنبال آن هستند را برای شما لیست می‌کنیم:

۱. توسعه همزمان

همانطور که پروژه‌ها پیچیده‌تر می‌شوند، یک نیاز فزاینده برای مدیریت نسخه‌های مختلف کد و پرونده و کل محصولات وجود دارد. منظر ما این است که چندین توسعه دهنده و طراح می‌توانند بدون نگرانی از این که آنها تلاش تکراری دارند یا کارهای اعضای دیگر تیم را نادیده می‌گیرند، روی پروژه‌های یکسانی کار کنند. بگذارید بگوییم شما در حال مدیریت استقرار IOT برای دوربین‌های امنیتی پیشرفته متصل به اینترنت هستید. در طول چرخه عمر محصول، شما ممکن است از ده نوع مختلف دوربین استفاده کنید که هر کدام تراشه متفاوتی دارند. در نتیجه، هر یک از آنها نرم افزارهای مختلفی خواهند داشت. استفاده از version control systems مناسب به این معنی است که می‌توانید چندین نسخه از کد خود را حفظ کنید تا عملکرد خاص تراشه و سیستم عامل هر دوربین را مدیریت کنید. بنابراین، هنگامی که برای جلوگیری از آدم ربایی در هواپیماهای ربایی نیاز به ایجاد یک وصله امنیتی مهم دارید، این کار آسان است. فوراً خواهید دید که کد مورد نظر تحت تأثیر قرار گرفته، تغییرات را ایجاد کرده و فقط کافیست تا شما راه حل درست را پیاده سازی کنید.

۲. اتوماسیون

دومین مورد برای در نظر داشتن یک version control systems مناسب، اتوماسیون است! کیفیت بالاتر و افزایش بهره وری از اولویت‌های مهم تیم‌های توسعه امروز است و تیم شما می‌تواند با انجام خودکار وظایف، مانند آزمایش و استقرار به اهداف شما برسد. برای توسعه سخت افزار، مانند نیمه هادی‌ها، اتوماسیون می‌تواند شامل موارد زیر باشد:

- آزمایش با آرایه‌های گیت قابل برنامه ریزی میدانی (FPGA).
- ادغام با سیستم‌های تأیید شبیه سازی و سنتز.

چنین FPGA ها یا سیستم‌های تأیید و آزمایش باید به همراه پرونده‌های داده‌ای بسیار بزرگ، کنترل و نسخه بندی شوند. بسیار مهم است که حتی تغییرات کوچک نیز پیگیری و مدیریت شوند. این یعنی

سیستم کنترل نسخه شما مرکز جهان IP است. پس یک مورد مناسب می‌تواند میلیون‌ها معامله خودکار در روز را با میلیون‌ها پرونده اداره کند.

۳. همکاری تیمی

شرکت‌ها محل استعداد و خلاقیت هستند و شما نیز ممکن است مراکز طراحی و توسعه در شهرهای بزرگ ایران داشته باشید و هنگامی که ضرب الاجل‌ها پیش می‌آید، ممکن است لازم باشد مهندسانی را از شرکت‌های خارجی به شرکت خود اضافه کنید. شما به روشی برای دسترسی جهانی اعضای تیم خود در همه امکانات نیاز دارید. داشتن یک منبع واحد با هویت مناسب و مدیریت دسترسی، برای موفقیت شما حیاتی است. با VCS مناسب، هر یک از اعضای تیم در حال کار بر روی آخرین نسخه هستند و این باعث سهولت در همکاری می‌شود.

۴. تغییرات پیگیری شده

هر تیم توسعه به تغییر و تحول نیاز دارد. پیگیری این که چه کسی، چه چیزی، چه زمانی و چرا تغییرات ایجاد می‌شود و اصلاً چرا مهم است. نرم افزار کنترل نسخه این اطلاعات دقیق را ضبط و این تاریخچه را برای همیشه حفظ می‌کند. پس هر کسی به این که چه کسی روی فلان پروژه در حال کار کردن است، دسترسی دارند، (مانند version control for power bi).

این امر به ویژه اگر حاکمیت، خطر و انطباق (GRC) یا نیازهای نظارتی داشته باشید بسیار مهم است. سابقه گزارش حسابرسی به ویژه در صنعت خودرو، هوا فضا، تجهیزات پزشکی و نیمه هادی بسیار مهم است.

۵. دسترسی بالا / بازیابی فاجعه

گران‌ترین دارایی شما تیم توسعه محصول شماست. نمی‌توانید آنها را بخاطر از دست دادن دسترسی به کد بیکار کنید. با استفاده از version control systems مناسب، می‌توانید ماکت مخزن

سازمانی خود، یعنی منبع واحد حقیقت، داشته باید و آن را در مکان دیگری به کار بگیرید. اگر هم اتفاقی افتاد، می‌توانید بلافاصله به نسخه اصلی دسترسی پیدا کنید.

مزایای سیستم‌های کنترل نسخه

توسعه نرم‌افزار بدون استفاده از کنترل نسخه، مانند عدم تهیه نسخه پشتیبان، کار پرمخاطره‌ای است. کنترل نسخه می‌تواند توسعه‌دهندگان را قادر به حرکت سریع‌تر کند و به تیم‌های نرم‌افزاری امکان حفظ کارایی و چابکی را هم‌گام با بزرگ‌ترین شدن تیم و اضافه شدن توسعه‌دهندگان جدید به آن، می‌دهد.

«سیستم‌های کنترل نسخه» (Version Control Systems | VCS) طی چند دهه اخیر شاهد بهبودهای قابل توجهی بوده‌اند. برخی از VCSها، بهتر از برخی دیگر هستند. سیستم کنترل نسخه، گاهی نیز با عنوان ابزارهای «مدیریت کد منبع» (Source Code Management | SCM) یا «سیستم کنترل بازنگری» (Revision Control System | RCS) شناخته می‌شود. یکی از محبوب‌ترین ابزارهای سیستم کنترل نسخه، نرم‌افزار گیت است. گیت، یک «سیستم کنترل نسخه توزیع شده» (Distributed VCS) (دسته‌ای که با عنوان DVCS شناخته شده است) محسوب می‌شود. در بخش‌های بعدی این مطلب، به طور کامل به گیت پرداخته خواهد شد.

همچون دیگر انواع سیستم‌های کنترل نسخه محبوبی که این روزها در دسترس توسعه‌دهندگان هستند، گیت نیز رایگان و اوپن سورس است. صرف‌نظر از اینکه نام این سیستم‌ها چیست و کدام سیستم توسط کاربر استفاده می‌شود، مزایای اولیه‌ای که از سیستم کنترل نسخه انتظار می‌رود، به شرح زیر است.

- تاریخچه کامل و بلند مدت از هر فایل: این یعنی همه تغییراتی که توسط کلیه افراد و طی سال‌ها روی هر فایل انجام شده، موجود خواهد بود. این تغییرات شامل ساختن و حذف فایل‌ها و همچنین، ویرایش محتوای آن‌ها می‌شود. ابزارهای VCS گوناگون، در چگونگی

مدیریت تغییر نام و جابجایی فایل‌ها با یکدیگر متفاوت هستند. این تاریخچه ممکن است شامل نام پدیدآورنده، تاریخ و یادداشت‌های نوشته شده پیرامون هر تغییر شود. داشتن تاریخچه کامل، امکان بازگشت به نسخه‌های پیشین را برای کمک به تحلیل‌های ریشه‌ای اشکالات (باگ‌ها) فراهم می‌کند و هنگامی که نیاز به رفع اشکال در نسخه‌های قدیمی‌تر نرم‌افزار است، قابلیت‌های حیاتی محسوب می‌شود. اگر نرم‌افزار به طور فعالانه‌ای کار کند، تقریباً هر چیزی را می‌توان به عنوان نسخه قدیمی‌تر نرم‌افزار در نظر گرفت.

- امکان انشعاب و ادغام: به کار گرفتن اعضای یک تیم پروژه نرم‌افزاری به گونه‌ای که به صورت هم‌زمان روی یک پروژه کار کنند، کار عاقلانه‌ای نیست؛ مگر آنکه تلاش‌های هر فرد روی جریان‌های کاملاً مستقل تغییرات، مزایای خودش را داشته باشد. ساخت یک انشعاب در سیستم کنترل نسخه، چندین جریان کاری را مستقل از یکدیگر نگه می‌دارد و در عین حال، امکان ادغام این تلاش‌ها با یکدیگر را فراهم می‌کند. این امر، توسعه‌دهندگان را قادر می‌سازد تا تایید کنند که تغییرات در هر شاخه، با دیگر شاخه‌ها ناسازگار نیستند. بسیاری از تیم‌های نرم‌افزاری، امکان انشعاب برای هر ویژگی یا حتی انشعاب برای هر «انتشار» یا هر دو این موارد را فراهم می‌کنند. جریان‌های کاری متفاوتی وجود دارد که تیم‌ها می‌توانند هنگام تصمیم‌گیری پیرامون چگونگی استفاده از تسهیلات انشعاب و ادغام در سیستم‌های کنترل نسخه، از میان آن‌ها انتخاب کنند.

- قابلیت ردیابی: وجود قابلیت ردیابی برای پیگیری هر تغییری که در نرم‌افزار انجام شده و مرتبط کردن آن به مدیریت پروژه و نرم‌افزارهای ردیابی باگ مانند «جیرا» (Jira) و وجود توانایی حاشیه‌نویسی برای هر تغییر با متنی که هدف و نیت تغییر را مشخص می‌کند، به تحلیل ریشه‌ای دلایل تغییر کمک می‌کند. داشتن تاریخچه حاشیه‌نویسی شده برای ردیابی توسعه‌دهنده، در هنگام خواندن کد برای فهمیدن کاری که انجام می‌دهد و دلیل طراحی آن به این صورت، به توسعه‌دهنده کمک می‌کند که تغییرات درست و هماهنگی را انجام دهد که با طراحی بلند مدت مورد نظر سیستم سازگار است. این ویژگی، به طور ویژه برای کار

کردن موثر با کدهای موروثی نیست و در توانمندسازی توسعه‌دهندگان برای تخمین کارهای بعدی با هر میزانی از صحت، نقش حیاتی دارد.

در حالی این امکان برای توسعه‌دهندگان وجود دارد که بدون استفاده از هر سیستم کنترل نسخه‌ای کار کنند، انجام چنین کاری پروژه را در خطر بزرگی قرار می‌دهد که هیچ تیم تخصصی نمی‌تواند پذیرای آن باشد. بنابراین، اکنون پرسشی که مطرح می‌شود این است که از کدام سیستم کنترل نسخه باید استفاده کرد. انواع مختلفی از سیستم‌های کنترل نسخه وجود دارد و برای هر یک از این انواع نیز نمونه‌های گوناگونی موجود است؛ اما توصیه این مطلب، استفاده از سیستم کنترل نسخه گیت (Git) است.

