۱- ابزارهای کنترل ورژن محل خوبی برای نگه داری سورس کد های ما هستند و همچین باعث میشوند تا به راحتی تغییرات را ردگیری کنیم و همچنین همه اعضا روی آخرین نسخه از کد کار میکنند و سرعت و چابکی بیشتری به تیم میبخشد.

۲- repository در واقع مخزن کد ماست و به ما امکان نگهداری کد و همه وابسته های آن ها را یکجا نگهداریم و از برنامه نسخه های متفاوتی داشته باشیم و هر موقع خواستیم به آنها دسترسی داشته باشیم (همان git. است). یک repository خوب دارای ۳فایل مهم readme.md و license و gitignore میباشد.

۳- هر داکیومنت خوب از دو بخش اجباری و یک بخش اختیاری تشکیل شده است. اولین قسمت اجباری introduction است که اگر کسی این repository را باز کرد، بداند برای چی اینجاست و دقیقا این repository چه کاری انجام میدهد. قسمت دوم اجباری اصطلاحا how to یا getting started میباشد که نحوه اجرا را شرح میدهد. قسمت سوم که اختیاری است، قسمت مشارکت یا contribute است.

۴- ایجاد یک نسخه مشابه از یک پروژه یا repository در سیستم local خودمان.

۵- git pull

۶- دستور checkout به طور کلی برای جا به جایی بین commit ها و یا branch هاست و با دستور git checkout <commit id> میتوانیم به commit مورد نظر برگردیم و تغییرات و فایل آن لحظه را بررسی کنیم. دستور git reset HEAD~<No>، مشخص میکند HEAD به چند commit عقب تبر برگردد و هیچ history ای هم به جا نمیگذارد. دستور revert جدا از اینکه به عقب بر میگردد، یک پیغام و commit به جا میگذارد تا بقیه اعضا هم از این اتفاق و دلیل اینکار اطلاع داشته باشند.

۷- هم merge و هم rebase برای به هم چسباندن شاخه ها استفاده میشود با این تفاوت که در rebase تغییرات مثلا شاخه feature بر اساس زمان در شاخه master قرار میگیرد ولی merge می آید و شاخه feature را به انتهای شاخه master میچسباند.

۸- با استفاده از دستور git log

۹- میتوانیم از دستور git show <commit id> استفاده کرده و همه تغییرات اعمال شده در آن commit را ببینیم. میتوانیم از دستور git diff هم استفاده کنیم و با انتخاب ۲ commit id یا ۲ branch name تغییرات مربوطه را مشاهده کنیم.

۱۰- tag ها به یک نقطه خاصی در history اشاره میکنند که معمولا بسیار مهم است. برای مثال وقتی شما میخواهید ورژن جدیدی از برنامه را منتشر کنید، با دستور git tag میتواند tag مورد نظر را اعمال کنید.

۱۱- ابتدا پروژه را fork کرده و به اکانت خود اضافه میکنید، سپس با clone روی سیستم local کپی میکنیم و تغییرات مورد نظر را انجام میدهیم. سپس وقتی از انجام تغییرات خیالمان راحت شد، باید یک pull request ایجاد کنیم، اکانت های مبدا و مقصد را مشخص کرده، فایل ها را مورد نظر برای pull را بررسی کرده و سپس با نوشتن یک پیغام شرح درخواست را مینویسیم و درخواست pull را ایجاد میکنیم. سپس این درخواست به دست مدیر پروژه میرسد و پس از بررسی نهایی تایید یا رد میشود.

۱۲- شاخه ها در واقع یک line جدا و مستقل توسعه هستند که کاری با شاخه اصلی ندارد و وقتی به سر انجام رسیدند، میتوانیم آنها را با شاخه اصلی یا یکدیگر merge کنیم.