|  |  |  |
| --- | --- | --- |
| **Capability** | **TFS** | **Git** |
| Alerts | Team members can receive email alerts when check-ins occur. | Team members can receive email alerts when commits are pushed to the server. |
| Auditability | Because your team checks in all their work into a centralized system, you can identify which user checked in a changeset and use compare to see what they changed. Looking at a file, you can annotate it to identify who changed a block of code, and when they did it. | You can identify which user pushed a commit. (Anyone can claim any identity as the author or committer.) You can identify when changes were made what was changed using history, compare, and annotate. |
| Branching | Path-based branches are used mostly as long-standing constructs to isolate risk of change among feature teams and releases. Team members typically set up an additional workspace for each branch they work on.  Changes in each branch are independent from each other, so you don’t have to check them in before switching from one branch to another. Merging between sibling branches requires a baseless merging.  You can get visualizations of your branch structures and where your changesets have been merged. | Branching is lightweight and path independent. Many developers create a branch for each new feature they are coding, sometimes on a daily basis. You can quickly switch from one branch to another to pivot among different variations of your codebase. You can create branches that exist only on your dev machine and share them if and when you’re ready.  You must commit, branch, stash, or undo changes before switching branches. Merging is simple and independent of the commit that the branch is based on.  You can compare branches to see which commits exist on which branches. |
| Builds (automated by TFBuild) | You can use all TFBuild capabilities to build any combination of content you want within the team project collection. | You can use most TFBuild capabilities to build one team project at a time, and one or more repositories at a time. |
| Changes | Team members can concurrently change files on their dev machines. You upload (check-in) changesets to the server when you create them. You can upload your changes at any time. However, you might be interrupted by conflicts.  You can change the comment of a changeset after you check it in. You can link changesets to work items and associate them with completed builds. | Team members can concurrently change files on their dev machines. You create commits on your dev machine independently of contributing them to the team. When you’re ready you must pull the latest commits before you upload (push) yours to the server. When you pull, you might be interrupted by conflicts.  You can link commits to work items and associate them with completed builds.  You can modify and combine commits from the command prompt. |
| Code reviews | Yes | Not available yet, but you can comment on and send email about a commit from the web portal. |
| Conflict resolution | You might have to resolve conflicts when you get, check in, merge, or unshelve. You can resolve all types of conflicts in Visual Studio. | You might have to resolve conflicts when you pull or merge. You can resolve content conflicts in Visual Studio. Other types of conflicts can be resolved from the command prompt. |
| File storage | You can check in large binary files. You might also want to use NuGet in combination or as an alternative. | You can check in small binary files. You might want to use [NuGet](http://go.microsoft.com/fwlink/?LinkId=246165" \t "_blank) as an alternative. |
| Files on the client dev machine | You can browse your files using Source Control Explorer in Visual Studio, or using Windows File Explorer or the [command prompt](http://msdn.microsoft.com/en-us/library/cc31bk2e.aspx). | You can browse your files using Windows File Explorer or the [command prompt](http://msdn.microsoft.com/en-us/library/dd286572.aspx). You cannot yet browse files in Visual Studio. |
| Files on the server | Each team project contains all files under a single root path (for example: $/FabrikamTFVC). You can apply permissions at the file level. You can lock files.  You can browse your files on the web portal and using Source Control Explorer in Visual Studio. | Each team project can contain one or more Git repositories and each Git repository can contain one or more branches. The most granular permissions you can apply are to a repository or a branch. Files cannot be locked.  You can browse your files on the web portal. |
| History | File history is not replicated on the client dev machine and so can be viewed only when you’re connected to the server.  You can view history in Visual Studio and on the web portal. You can annotate files to see who changed a line, and when they changed it. | File history is replicated on the client dev machine and can be viewed even when not connected to the server.  You can view history in Visual Studio and on the web portal. On the web portal you can annotate files to see who changed a line, and when they changed it. |
| Manage work on your dev machine | Pending changes and my work pages | Changes, commits, and branches pages. |
| Scale | You can work on small or very large scale projects using local workspaces. Supports massive scale (millions of files per branch and large binary files) projects using server workspaces. | You can quickly begin small projects. You can scale up to very large projects, but you have to plan ahead to modularize your codebase. You can create multiple repositories in a team project. |
| Suspend your work | You can suspend from my work page or shelve your changes. | You can create a branch from (from Visual Studio or the command prompt) from or stash (from the command prompt) |
| Tag your files | You can apply labels to a version of one or more files from either Visual Studio or the command prompt. Each file can have label applied to a different version. | You can apply tags from the command prompt to individual commits. View tags in the Visual Studio history window. |
| Visual Studio compatibility | You can use all supported previous versions of Visual Studio. | You can use Visual Studio 2013 (Git is included) or Visual Studio 2012 Update 3 (you must also install Visual Studio Tools for Git). |
| Web portal | You can browse your codebase (including branches), view history, annotate and comment on changesets and shelvesets, and perform other tasks such as ad hoc downloading of selected parts of your codebase as a .zip file. | You can browse your codebase, view history, compare branches, annotate and comment on commits, and perform other tasks such as ad hoc downloading of selected parts of your codebase as a .zip file. |