Difference between TFS and GIT (TFS Vs GIT)

Like 1   Tweet   3

TFS and GIT are the tools for revision control. The difference between TFS and GIT are listed in the following table.

TFS	GIT
TFS uses a lone and centralized repository for tracking and versioning files. Any modification made locally in TFS is often updated in the central server, with which other users can obtain the new changes.	GIT supports a repository locally for tracking and versioning files. Modifications made in GIT are pulled in and pushed out via an isolated repository and the modifications are shared with the other users.
TFS is not available for free and requires licensing fees.	GIT is available for free.
TFS works on the basis of file-system and not changesets.	GIT works on the basis of changesets.
Developing branches in TFS is very costly on several resources.	Developing a branch is tremendously fast and inexpensive with a slight overhead.
Developing a branch in TFS means the user is creating a duplicate of the complete directory. For example, developing ten branches in TFS is basically duplicating that file ten times.	Developing a branch in GIT is considered a normal workflow. In GIT, comparison between the binary information i.e. blobs can also be done, so that information is never copied along branches. For example, developing ten branches in GIT means the user is just making a reference of a snapshot. Here the branches receive some bytes of information instead of copying the information again and again.

In TFS, the *'diffs'* are executed by means of file-system characteristics. Hence a file in TFS has to be checked-out before executing it. In case, if that file is over-written without being checked, then TFS cannot identify any changes that are made.	In GIT, *'diffs'* and document comparisons are primarily dependent on 'SHA1' cryptographic hashes. Hence any changes made in the file system will be detected by GIT, on the basis of the difference in hash and the file will be marked for check-in.
TFS depends on the idea of baseless merge. In TFS, while performing the merge operation it does not show a general ancestor among the branches, thus solving a group of merging conflicts. The general concept in TFS is to stock up files and avoid users from working on the similar file.	GIT is based on the idea of base merge. Merging conflicts are resolved in GIT using base merging concept. In GIT if a base is available then the general ancestor is easily identified by GIT (automatically) and the changes alone are merged. This feature offers high flexibility in GIT which is lacking in TFS.
TFS does not support rebasing, as the alterations are merely stored as copy files on a server.	GIT offers support to rebasing, i.e. dragging in the alterations from repository or several other resources is possible.
In TFS, the check-in is made for the complete file and the 'diffs' are executed against previously released files.	In GIT, only an additional '.' is committed instead of checking-in the complete file.
In TFS all the alterations are considered as revisions and the alterations will end in deltas.	In GIT, all the alterations i.e. commit end in a snapshot. These snapshots in GIT will show the way the repository was looking while the alterations were taking place.
TFS supports some features like (i) conditional commit, (ii) save IDE state, (iii) work item tracking, (iv) automated builds and tests, (v) analytics and charting, and (vi) manual test tracking.	GIT does not support any of these features.
TFS does not support offline access and does not support merging of branches that are not related.	GIT supports (i) safe merge between unrelated branches, (ii) offline access to history, and (iii) offline commit.

0 Comments

Sort by | Oldest |

Add a comment...

f Facebook Comments Plugin

0 Comments   **Rapid Programming**   1   **Login** ▾

♥ **Recommend**   ↗ **Share**   Sort by Best ▾

Start the discussion…

Be the first to comment.

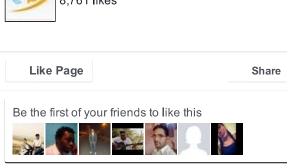✉ Subscribe   Ⓓ Add Disqus to your site Add Disqus Add   🔒 Privacy

Related Questions Answers Articles

- Difference between GIT and VSS (GIT Vs VSS) (http://www.rapidprogramming.com/questions-answers/difference-between-git-and-vss-git-vs-vss-1527)

- Difference between CVS and VSS (CVS Vs VSS) (http://www.rapidprogramming.com/questions-answers/difference-between-cvs-and-vss-cvs-vs-vss-1526)
- Difference between CVS and SVN (CVS Vs SVN) (http://www.rapidprogramming.com/questions-answers/difference-between-cvs-and-svn-cvs-vs-svn-1524)
- Difference between Subversion (SVN) and Visual Source Safe (VSS) (http://www.rapidprogramming.com/questions-answers/difference-between-subversion-svn-and-visual-source-safe-vss-1522)
- Difference between Subversion (SVN) & GNU Interactive Tools (GIT) (http://www.rapidprogramming.com/questions-answers/difference-between-subversion-svn-gnu-interactive-tools-git-1519)
- Team Foundation Server (TFS) Vs Visual Source Safe (VSS) (http://www.rapidprogramming.com/questions-answers/team-foundation-server-tfs-vs-visual-source-safe-vss-1517)
- Difference between SharePoint and Nuxeo (SharePoint Vs Nuxeo) (http://www.rapidprogramming.com/questions-answers/difference-between-sharepoint-and-nuxeo-sharepoint-vs-nuxeo-1515)
- Difference between TFS and SVN (TFS Vs SVN) (http://www.rapidprogramming.com/questions-answers/difference-between-tfs-and-svn-tfs-vs-svn-1514)
- Difference between Selenium RC and Selenium Web Driver (http://www.rapidprogramming.com/questions-answers/difference-between-selenium-rc-and-selenium-web-driver-1513)
- Difference between Selenium and QTP (Selenium Vs QTP) (http://www.rapidprogramming.com/questions-answers/difference-between-selenium-and-qtp-selenium-vs-qtp-1512)