|  |  |  |
| --- | --- | --- |
|  | **Git reflog** | **TeamForge** **Git Integration with ‘History Protect’** |
| **Accessibility** | Requires direct access to file system on server where ‘blessed Git repository’ is hosted which is very unlikely in huge organizations and will keep the server administrators busy | ‘Self –Service’ approach. Users with appropriate permissions in TeamForge can find out/resurrect deleted/rewritten branches by themselves, decreasing work load of server administrators.Gerrit Administrators can also permanently delete selected branches/tags. |
| **Signal-to- Noise Ratio** | reflog records <all> changes in the repository* Any push (also ordinary fast forward)
* Any merge
* Any Branch creation/deletion
* Any Tag creation / deletion

Finding out about history rewrites/deleted branches is like searching for a needle in a haystack | History Protect only reports* Deleted branches/tags
* History rewrites (non fast forward pushes)
 |
| **Notification** |  No notification | * Email to Gerrit Administrators
* Audit log entry whenever branch/tags gets
	+ Deleted
	+ Re-written (non fast forward)
	+ Resurrected
	+ Permanently deleted
 |
| **Ease of use** | * Only manually configurable by administrator having file system access
* To be configured for each and every repository
* Restoring requires running git commands on server
 | * Pre-configured in TeamForge/Gerrit
* Configurable for all repositories by setting site-wide config option or on a per repository basis
* User with appropriate permission can restore history using Gerrit WebUI / Git client
 |
| **Protection against object pruning/reflog expiration** | RefLog expiration and gc pruning settings have to be manually configured by server administrator. Only possibility to not lose commits no longer referenced in a branch is to set both values to <unlimited> which will consume huge amounts of disk space, slows down garbage collection and does not allow to permanently delete specific commits (all or nothing). | Preserved commits will never pruned by garbage collection unless permanently removed using Gerrit Web UI.No need to keep a large ref log. Garbage collection will run faster since all commits are still referenced in the repo. |