

## Знакомство с системами Контроля версий

Вопросы:

- «1. Какой тип систем контроля версий наиболее совершенный?*
- 2. В какой системе контроля версий впервые появилось сжатие файлов?*
- 3. Какая возможность есть в системе контроля версий SVN, которой нет в GIT?*
- 4. Какая команда поможет посмотреть историю коммитов, для активной ветви в GIT?*
- 5. В какой последовательности нужно использовать команды `git commit`, `git push` и `git pull`, чтобы залить свой код в общую ветку?*
- 6. Какая команда поможет создать репозиторий в GIT для вашего проекта?»*

Ответы:

1. Наиболее совершенными являются наиболее современные – распределённые системы контроля версий. Такие как Git и Mercuriy. В них решена проблема единой точки отказа, поскольку все клиенты полностью копируют репозиторий. Также они обладают положительными качествами предыдущего поколения – Централизованных систем, поскольку предоставляют одновременный доступ клиентам через сеть, файлы подвергаются сжатию и хешированию.
2. Сжатие впервые появилось в Subversion (SVN) в 2000 году. Ему подвергаются файлы внесённых изменений диффы или дельты с помощью алгоритмов Lz4 или zlib.
3. GIT сохраняет метаданные изменений, а SVN целые файлы. Это экономит место и время. Файлы сжимаются, сохраняются дельты, которые экономят место. По достижению 1023 дельт – файл перезаписывается целиком и таким образом достигается баланс скорости и объёма хранения.
4. Поможет посмотреть историю коммитов и соответствующие сообщения для активной ветви команда **git log**.
5. Насколько я понимаю, правильная последовательность такова: **git add - git commit - git push**. Первая команда позволяет добавить файл, вторая фиксирует набор изменённых файлов, а третьей командой осуществляется перенос упакованных файлов локальных коммитов в удалённый репозиторий.
6. Команда **git init** создаёт репозиторий на основе текущего каталога.