

Анализ систем управления версиями

Система управления версиями (также используется определение «система контроля версий», от англ. *Version Control System, VCS* или *Revision Control System*) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Рассмотрим самые распространенные системы контроля версий: CVS, SVN, Git, Mercurial.

- 1. Система одновременных версий (CVS)** - позволяет получать с сервера нужную версию проекта — «check-out» (извлечение), а затем пересылать обратно на сервер, «check-in» (возврат), с внесенными изменениями. Это была первая система контроля версий. Пользователю нужно было быстро фиксировать изменения в репозитории, пока другие не опередили его.

Преимущества:

- Испытанная временем технология, которая удерживается на рынке десятки лет.

Недостатки:

- Переименование или перемещение файлов не отражается в истории
- Риски безопасности, связанные с символическими ссылками на файлы
- Нет поддержки атомарных операций, что может привести к повреждению кода
- Операции с ветками программного кода дорогостоящие, так как эта система контроля не предназначена для долгосрочных проектов с ветками кода

- 2. Apache Subversion (SVN)** - создавалась как альтернатива CVS с целью исправить недостатки CVS и в то же время обеспечить высокую совместимость с ней. SVN это бесплатная система контроля версий с открытым исходным кодом.

Преимущества:

- Система на основе CVS
- Допускает атомарные операции
- Операции с ветвлением кода менее затратны
- Широкий выбор плагинов IDE

- Не использует пиринговую модель

Недостатки:

- Все еще сохраняются ошибки, связанные с переименованием файлов и директорий
- Неудовлетворительный набор команд для работы с репозиторием
- Сравнительно небольшая скорость

- 3. Git** - эта система была создана для управления разработкой ядра Linux и использует подход, который в корне отличается от CVS и SVN. В Git есть множество инструментов для навигации по истории изменений. Каждая рабочая копия исходного кода содержит всю историю разработки, что крайне полезно, когда программируешь без Интернет-соединения.

Преимущества:

- Прекрасно подходит для тех, кто ненавидит CVS/SVN
- Значительное увеличение быстродействия
- Дешевые операции с ветками кода
- Полная история разработки доступная оффлайн
- Распределенная, пиринговая модель

Недостатки:

- Высокий порог вхождения (обучения) для тех, кто ранее использовал SVN
- Ограниченная поддержка Windows (по сравнению с Linux)

- 4. Mercurial** - была выпущена одновременно с Git. Это также распределенная система контроля версий. создавалась в качестве альтернативы Git для разработки модулей ядра Linux. Но так как выбрали все-таки Git, то Mercurial используется меньше. Система контроля версий Mercurial отличается от других систем контроля версий тем, что главным образом она написана на Python (а не C).

Преимущества:

- По сравнению с Git легче в освоении
- Подробная документация
- Распределенная модель системы контроля версий

Недостатки:

- Нет возможности слияния двух родительских веток
- Использование плагинов, а не скриптов
- Меньше возможностей для нестандартных решений