Актуализация систем контроля версии

Системы контроля версий (СКВ) стали неотъемлемой частью разработки проектов . С их помощью удается решить рад задач таких как:

- Работа над проектом одновременно нескольких разработчиков

- Возврат при необходимости к предыдущим версиям

- Создавать ветвление проекта

Исторически первая СКВ была выпущена в 1982г под названием RCS (Revision Control System). Со временем СКВ видоизменялись, обрастая новыми возможностями и избавляясь от старых недостатков.

Сегодня можно выделить три наиболее популярные СКВ. Это Subversion , Git и Mercurial. Основным отличием этих СКВ является подход к решению поставленной задачи. Существует два основных подхода – это централизованная модель и распределенная модель. Каждая из моделей обладает своими преимущеествами и недостатками . Рассмотрим каждую из них:

Централизованная модель - основана на использовании единого хранилища файлов , управляемое специальным сервером , который и выполняет большую часть функций по управлению версиями.

Пользователь, работающий с документами, должен сначала получить нужную ему версию документа из хранилища. Может быть получена последняя версия или любая из предыдущих, которая может быть выбрана по номеру версии или дате создания, иногда и по другим признакам. После того, как в файл внесены нужные изменения, новая версия помещается в хранилище. В отличие от простого сохранения файла, предыдущая версия не стирается, а тоже остаётся в хранилище и может быть оттуда получена в любое время.

Преимуществом такой модели является четкий контроль администратором проекта всей работы проводящимися над проектом.

Однако в такой модели есть и серьезные недостатки . Наиболее существенным является выход из строя центрального сервера в результате чего разработчики не могут взаимодействовать, и никто не может сохранить новой версии своей работы.

Распределенная модель - не нуждаются в централизованном хранилище , вся история изменения файлов хранится на каждом компьютере . При внесении изменения одним разработчиком все остальные копии синхронизируются .Таким образом отсутствует возможность потери данных.

К недостаткам распределенной модели можно отнести увеличение требуемого объёма дисковой памяти в на каждом компьютере так как приходится хранить полную историю версий, тогда как в централизованной системе на компьютере разработчика обычно хранится лишь рабочая копия, то есть срез репозитория на какой-то момент времени и внесённые изменения.

Проведем актуализацию самых популярных СКВ , а также опишем их достоинства и недостатки .

Subversion (SVN) - централизованная система управления версиями официальная дата первого релиза 2004год. Последняя версия 1.9.7 на 10 августа 2017. Возможности SVN :

- Отслеживает версии не только файлов, но и директорий.

- Поддерживает атомарные многофайловые фиксации

- Поддерживает копирование, перемещение и переименование файлов и директорий без потери истории изменений.

- С каждым файлом и директорией может быть связан произвольный набор свойств, состоящих из названия и значения. Свойства тоже находятся под управлением версиями.

- Поддерживается необязательная блокировка файлов

- Ветви реализованы в пространстве путей.

- Вместо системы меток используется иерархия директорий — для метки создаётся отдельная директория (как и для ветви).

Основываясь на данных возможностях SVN эффективна в следующих областях.:

Существенно уменьшает трафик так как между клиентом и сервером передаются только различия между файлами.

Одинаково эффективно работает как с [текстовыми](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), так и с [двоичными файлами](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB).

Требуется небольшое фиксированное количество времени и дискового пространства.

* Git  (**Global** **Information** Tracker) -  распределённая [система управления версиями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8). Первая версия выпущена [7 апреля](https://ru.wikipedia.org/wiki/7_%D0%B0%D0%BF%D1%80%D0%B5%D0%BB%D1%8F) [2005 года](https://ru.wikipedia.org/wiki/2005_%D0%B3%D0%BE%D0%B4). на сегодняшний день версия 2.17.0  ([2 апреля](https://ru.wikipedia.org/wiki/2_%D0%B0%D0%BF%D1%80%D0%B5%D0%BB%D1%8F) [2018](https://ru.wikipedia.org/wiki/2018_%D0%B3%D0%BE%D0%B4)).

Возможности Git :

- поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки.

- предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой

система сравнения ревизий и проверки корректности данных, основанные на алгоритме хеширования SHA1 (Secure Hash Algorithm 1).

Наличие локального репозитория, содержащего полную информацию обо всех изменениях, позволяет вести полноценный локальный контроль версий и записывать в главный репозиторий только полностью прошедшие проверку изменения.

Управление с помощью удобных и интуитивно понятный набор команд.

Систем основана на использовании скриптов , что позволяет легко создавать графические оболочки , позволяющих быстро и качественно вести работы с Git’ом.

Возможность делать контрольные точки, в которых данные сохраняются без дельта компрессии, а полностью. Это позволяет уменьшить скорость восстановления данных, так как за основу берется ближайшая контрольная точка, и восстановление идет от нее.

Mercurial - распределённая [система управления версиями](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8). Первая версия выпущена [19 апреля](https://ru.wikipedia.org/wiki/7_%D0%B0%D0%BF%D1%80%D0%B5%D0%BB%D1%8F) [2005 года](https://ru.wikipedia.org/wiki/2005_%D0%B3%D0%BE%D0%B4). на сегодняшний день версия 4.5.3  ([1](https://ru.wikipedia.org/wiki/2_%D0%B0%D0%BF%D1%80%D0%B5%D0%BB%D1%8F) апреля  [2018](https://ru.wikipedia.org/wiki/2018_%D0%B3%D0%BE%D0%B4)).

* Mercurial сочетает в себе основные качества распределенной СКВ , обладая при этом своими особенностями к поставленной задачи. При сравнении с Git можно выделить следующие отличия .

#### **Преимущества:**

По сравнению с Git легче в освоении

Подробная документация

#### **Недостатки:**

Нет возможности слияния двух родительских веток

Использование плагинов, а не скриптов

Меньше возможностей для нестандартных решений

Анализ актуализации СКВ и рекомендации для применения для различных типов проектов.

Анализируя самые популярные СКВ очевидно ,что все системы развиваются и выбор конкретной СКВ сводится к узконаправленным требованиям и субъективным мнениям. Если необходимо четко контролировать процесс, разработки , блокировать изменение отдельных файлов то предпочтение отдают SVN, если проект разрабатывается удаленно и часто центральный репозиторий оказывается недоступным то предпочтение отдают Git и Mercurial. Большое значение для выбора используемой СКВ является быстрота обучения пользователей считается что порог вхождения в SVN самый низкий .

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |