

УДК 621.398
СОВРЕМЕННЫЙ ПОДХОД К УПРАВЛЕНИЮ ИТ-ПРОЕКТАМИ

Н. В. ВЫГОВСКАЯ, В. М. ПРУДНИКОВ, С. А. СИДОРЕНКО
Белорусско-Российский университет
Могилев, Беларусь

Разработка современных ИТ-проектов, как правило, требует привлечения большого количества участников. Каждый из участников решает свою задачу в рамках общего единого проекта. При этом, возникает необходимость просматривать и учитывать результаты работы других участников и объединения их для создания целостного конечного проекта.

В процессе подготовки проекта разработчикам приходится неоднократно возвращаться к предыдущим версиям создаваемых файлов. С этой целью используются системы контроля версий, или vcs.

В этом случае перед разработчиками возникает проблема выбора – какая из систем контроля версий наилучшим образом может подойти для вашего проекта?

Сравним несколько известных решений: система одновременных версий (CVS), Apache Subversion (SVN), Mercurial и Git. У каждой из систем есть свои особенности, преимущества и недостатки. Выбор разработчика определяется операционной системой, классом решаемых задач, доступностью системы, количеством участников проекта, хостингом для размещения, экономическими критериями.

CVS является одной из наиболее ранних систем и создавалась с целью избегания конфликта версий. В настоящее время CVS поддерживает разработку проектов с разными ветками кода. В итоге имеем несколько вариантов продукта с разными характеристиками, которые объединяются позже в финальном проекте.

CVS обычно работают под управлением Unix, но разработана также версия CVS для Windows.

Существенными недостатками CVS являются: нет контроля за переименованием и перемещением файлов; дорогостоящие операции с ветками программного кода; нет долгосрочной поддержки проектов с ветками кода.

SVN разрабатывалась, как совместимая с CVS, и для исправления ее недостатков. Код в SVN защищен от беспорядочных правок, которые конфликтуют между собой и вызывают ошибки. SVN используется для более крупных проектов, отличающихся ветвлением кода и многими направлениями разработки.

К недостаткам SVN относятся: невысокая скорость; как и в CVS, сохраняются ошибки при переименовании файлов и директорий; недостаточный набор команд для операций с репозиторием.



Mercurial создавалась под Linux как альтернатива Git. Система контроля версий Mercurial написана на Python, ориентирована, главным образом, на Python-программистов, относительно легче в освоении, имеет распределенную модель системы контроля версий.

Недостатки Mercurial: отсутствует возможность для слияния двух родительских веток; ограничены возможности неординарных решений.

Git исходно была разработана под Linux и имеет существенные отличия от CVS и SVN. Одно из основных требований к разработке Git – создание более быстродействующей распределенной системы контроля версий, чем CVS. Несмотря на то, что наиболее быстро Git работает под Linux, она также может работать на Windows и на Unix-подобных системах (например, MacOS). Программный код доступен даже на компьютере без репозитория. В Git предусмотрен большой набор инструментов для навигации по журналу изменений, в котором содержится вся история разработки. Преимуществами также являются: распределенная, пиринговая модель; недорогие операции с ветками кода.

Благодаря своим превосходным характеристикам и неоспоримым преимуществам именно vcs Git используется в большинстве ведущих IT-фирм по разработке программного обеспечения. Возможность бесплатного использования большим числом пользователей делает ее привлекательной для студентов. Таким образом, целесообразно внедрение Git в учебный процесс.

В настоящее время система контроля версий Git уже используется студентами специальности АСОИ в дисциплине «Объектно-ориентированное программирование», которая проводится на филиале ЕРАМ-systems. Также планируется применение системы Git в дисциплине «Технологии интернет-программирования» для анонсирования курсовых проектов и в других целях.

Установка Git производится с бесплатного интернет-ресурса <https://git-scm.com/downloads>. Для удобства работы с Git можно использовать не только командную строку, но и графический интерфейс.

Другим полезным фактором использования данной системы контроля версий Git является возможность бесплатного хостинга на GitHub. GitHub Pages предоставляет возможность разместить один сайт для аккаунта и подключить свой домен бесплатно, для чего нужно создать репозиторий с именем пользователя.

Хостинг сайта основан на ветке master git-репозитория. Каждый коммит в master приводит к его обновлению. Поэтому разработку и тестирование лучше проводить в отдельной ветке и потом через pull-request сливать в master.