

УДК 004.4'23

С. В. Смирнов, кандидат физико-математических наук

Д. С. Селетков, студент

Ижевский государственный технический университет имени М. Т. Калашникова

УПРАВЛЕНИЕ ВЕРСИЯМИ ПРОЕКТА ИНТЕРНЕТ-МАГАЗИНА НЕСКОЛЬКИМИ РАЗРАБОТЧИКАМИ

В статье рассматривается управление версиями проекта интернет-магазина на основе системы Git. Приводится пример сценария работы с проектом.

Ключевые слова: версионность; разработка; хранение; интернет-магазин.

На сегодняшний день Интернет доступен большому числу человек вне зависимости от их местоположения. Более того, количество людей, постоянно пользующихся Интернетом, непрерывно растет. Создание интернет-магазина позволяет компании значительно расширить клиентскую базу, даже если она изначально была крупной. Полный интернет-магазин, максимально ориентированный на удобство работы клиента с ним, – это сложная система, состоящая из нескольких разноплановых модулей и имеющая свои технические особенности [1]. Такие системы разрабатываются несколькими специалистами одновременно: программистами, дизайнерами, верстальщиками и т. д. Для начала работы специалиста важно иметь самую актуальную версию системы. Для управления версиями распределенных проектов используют системы контроля версий [2].

Система контроля версий – это программное средство для записи изменений в файле или наборе файлов, позволяющее переключиться на их определенную версию [3]. Системы контроля версий сохраняют не только сами версии файлов, но и метаданные о версии: кто сделал изменения, когда они были сделаны и с какими комментариями.

Системы контроля версий могут отслеживать изменения любых типов файлов, что позволяет использовать их для любых целей: написание программного кода, создание макета сайта или написание научной статьи. Любая работа, подразумевающая изменение набора файлов и важность всех их промежуточных версий, может быть существенно облегчена использованием информационной системы этого класса.

Изначально технические специалисты сами управляли версиями своих проектов. Самый примитивный способ – просто сохранять файл как новый и в названии указывать номер версии. Однако если необходимо управлять версиями набора файлов, возникает избыточность, т. к. необходимо хранить полную копию проекта даже при небольшом изменении одного файла. Более того, данный подход подвержен ошибкам человеческого фактора. К тому же при ручном версионировании проекта становится практически невозможно поддерживать актуальность информации при распределенной работе, такой как создание интернет-магазина.

Системы контроля версий используют централизованную модель хранения информации. Такой под-

ход позволяет упростить процесс актуализации информации при распределенной работе.

Для решения проблемы избыточности, системы контроля версий используют принцип хранения информации, называемый дельтакомпрессией. Этот принцип состоит в хранении целиком только первой версии файлов, а все последующие версии хранятся в виде изменений относительно исходного состояния. Так, при небольшом изменении в одном файле крупного проекта новая версия будет хранить лишь эти изменения.

В терминах систем контроля версий весь проект, управляемый системой контроля версий, имеет название *репозиторий* (англ. *repository* – хранилище), а каждая отдельная версия репозитория – **коммит** (англ. *commit* – фиксировать).

Рассмотрим процесс разработки интернет-магазина. Базовыми компонентами интернет-магазина, необходимыми для обеспечения его минимальной функциональности, являются [4]:

- база данных, содержащая информацию о товарах и компании;
- серверная часть, обрабатывающая запросы пользователей и преобразующая данные из базы в вид, требуемый для передачи их в ответ на запрос;
- клиентская часть – дизайн, верстка и бизнес-логика сайта, позволяющего просматривать и заказывать товары.

Для соответствующего уровня качества интернет-магазина описанные выше компоненты должны быть разработаны специалистами по каждому профилю. Из этого следует, что для разработки системы требуется как минимум четыре специалиста:

- 1) инженер баз данных разрабатывает структуру базы данных, поддерживает ее стабильную работу и оптимизирует внутренние и внешние механизмы обработки запросов к ней;
- 2) *backend*-разработчик проектирует и разрабатывает серверную часть магазина, отвечающую за обработку запросов заказчика, генерацию и отправку данных из базы в необходимом формате;
- 3) веб-дизайнер проектирует дизайн сайта и представляет его в виде макета;
- 4) *frontend*-разработчик производит верстку сайта в соответствии с макетом дизайнера и программирует бизнес-логику сайта.

В связи с тем, что над проектом одновременно работают несколько специалистов и разработка интернет-магазина представляет из себя работу с изме-

няющимися файлами, в данном случае целесообразно использовать одну из систем контроля версий.

Существует множество бесплатных систем контроля версий, самыми распространенными являются *Git*, *Mercurial* и *Subversion*. Выбор системы *Git* в данном проекте обусловлен преимуществами:

1. *Git* хранит локально весь проект, что ускоряет его работу (по сравнению с *Subversion*, который локально хранит только последние версии файлов).

2. Для хранения версий в *Git* вместо дельтакомпресии, используемой в *Mercurial*, используются сжатые копии файлов. Такой подход позволяет облегчить изменение и удаление отдельных версий, работу с ветками.

Веткой в *Git* называют независимый набор коммитов. Обычно ветки создают для работы над обособленным от основного функционалом системы. Основная ветка разработки в *Git* по умолчанию называется мастер-веткой (англ. *master*).

Для управления *Git* используется одноименная консольная команда *git*. Так, для создания *git*-репозитория достаточно выполнить команду *git init* в требуемой директории. Добавление файлов в список отслеживаемых системой выполняется командой *git add "имя_файла"* или *git add -A*, чтобы рекурсивно добавить все файлы из корневой директории репозитория. Фиксация новых версий отслеживаемых файлов (создание коммита) производится командой *git commit -m "комментарий"*. Для отправки локальных изменений на сервер используется команда *git push*, а для получения проекта с сервера – *git pull*.

Рассмотрим применение *Git* на примере добавления функции фильтрации товаров в интернет-магазине. Эта функция комплексная в плане реализации, поэтому для ее внедрения следует создать отдельную ветку. Вариант реализации этого функционала простым добавлением коммитов в мастер-ветку не является подходящим, т. к. в таком случае мастер-ветка будет содержать промежуточный, неполноценный функционал, чего следует избегать.

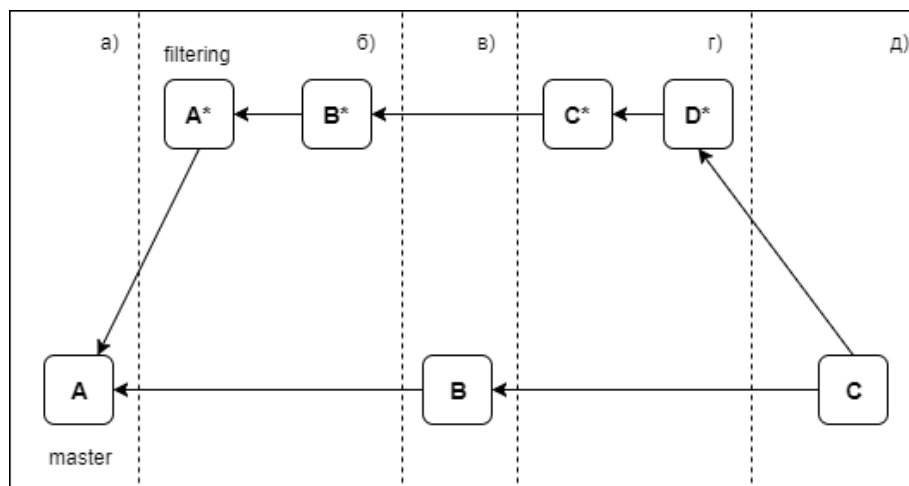
После окончания разработки необходимо произвести слияние новой и мастер-веток, чтобы новые изменения оказались в основной версии системы.

Визуализация процесса добавления этого функционала приведена на рисунке, где коммиты представлены в виде сокращенных названий в прямоугольниках, а стрелки указывают на «родительский» коммит. Состояние системы до внесения описываемых изменений здесь представлено одним коммитом *A* в мастер-ветке (рисунок, а).

Для добавления фильтрации *backend*-разработчику необходимо добавить данный функционал в серверную часть интернет-магазина. Перед началом работы ему необходимо получить актуальную версию проекта с сервера (*git pull*). После этого он может приступить к разработке. Для этого ему необходимо создать ветку с названием, соответствующим цели создания ветки, например, *git branch filtering* и перейти к изменениям в новой ветке командой *git checkout filtering*. Пусть коммит *A** содержит изменения, необходимые для фильтрации товаров по производителю, а коммит *B** – для фильтрации по цене (рисунок, б).

Преимущество использования веток состоит в их обособленности, что позволяет разработчикам одновременно делать различные изменения в системе независимо друг от друга. Так, например, инженер баз данных мог внести небольшие корректировки в *SQL*-запросы проекта коммитом *B* прямо в мастер-ветку, никак не мешая *backend*-разработчику (рисунок, в).

Для того чтобы пользователи могли использовать фильтрацию предметов, веб-дизайнеру необходимо спроектировать элементы управления фильтрацией, а *frontend*-разработчику – реализовать логику на сайте. Поскольку ветка *filtering* была создана для добавления функции фильтрации, специалисты могут добавить свои коммиты *C** и *D** соответственно в эту же ветку (рисунок, г).



Процесс добавления изменений в проект с помощью веток: а – репозиторий до внесения изменений; б – создание новой ветки и работа в ней; в – параллельная разработка в обеих ветках; г – окончание добавления нового функционала; д – слияние веток

После полного окончания разработки функции фильтрации товаров в интернет-магазине остается лишь слить две ветки, чтобы изменения присутствовали в основной версии проекта (рисунок, д). Для этого необходимо перейти в мастер-ветку командой *git checkout master* и выполнить слияние командой *git merge filtering*.

Применение системы контроля версий *Git* в процессе разработки интернет-магазина позволяет облегчить процесс актуализации данных у разных специалистов. Использование «веток» также ускоряет процесс разработки, позволяя проводить его параллельно нескольким специалистам без риска помешать друг другу.

Библиографические ссылки

1. *Смирнов С. В.* Формирование коммерческих интернет-предложений с учетом региональных особенностей // *Современные проблемы экономики, бизнеса и менеджмента: теория и практика* : сб. ст. Междунар. онлайн-видеоконф. – Ижевск : Изд-во ИжГТУ, 2013.
2. *Jon Loeliger, Matthew McCullough.* Version Control with Git: Powerful tools and techniques for collaborative software development. – 2nd Edition. – Boston : O'Reilly Media, 2012. – 456 p.
3. *Scott Chacon, Ben Straub.* Pro Git. – 2nd Edition. – New York : Apress, 2014. – 456 p.
4. *Смирнов С. В.* Указ. соч.

S. V. Smirnov, PhD (Physics and Mathematics)
D. S. Seletkov, Student
Kalashnikov Izhevsk State Technical University

VERSIONS MANAGING OF ONLINE STORE PROJECT BY SEVERAL DEVELOPERS

In the article, version control of the Internet store project based on the Git system is considered. An example of the scenario of working with the project is given.

Keywords: versioning; development; storage; online store.