

ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 378.147
ББК 4448.026

ГСНТИ 14.25.07; 20.01.45

Код ВАК 13.00.02; 13.00.08

П. И. Алексеевский

Екатеринбург

ПРИМЕНЕНИЕ СРЕДСТВ УПРАВЛЕНИЯ ВЕРСИЯМИ ДЛЯ КОЛЛЕКТИВНОЙ РАБОТЫ СТУДЕНТОВ НАД ПРОЕКТОМ КОМПЬЮТЕРНОЙ ИГРЫ

КЛЮЧЕВЫЕ СЛОВА: обучение программированию; системы управления версиями; коллективная разработка.

АННОТАЦИЯ. Рассматривается подход к хранению студенческих программных проектов в информационной среде образовательного учреждения. Обосновывается необходимость обучения студентов работе с системами управления версиями и использования их в учебной деятельности.

P. I. Alexeevsky

Ekaterinburg

USING VERSION CONTROL SYSTEMS FOR STUDENTS' COLLABORATIVE WORK ON THE COMPUTER GAME DEVELOPMENT

KEY WORDS: teaching programming; version control systems; team development.

ABSTRACT. This paper examines a storage technique for students' programming developments in the information environment of educational institutions. The necessity of training to use version control systems in educational and professional activities is substantiated.

Неотъемлемой составляющей курса программирования в вузах является деятельность по закреплению учебного материала и получению навыков использования изучаемых инструментальных средств. Для этого используются специальным образом составленные практические задания, предусматривающие не только реализацию изученных алгоритмов обработки данных, но и творческую и коллективную работу [1], в связи с чем возникает необходимость долговременного хранения исходного кода программ для последующего использования его студентами. Такая задача присутствует и при работе студентов над учебными и исследовательскими проектами.

Типичным решением данной проблемы является предоставление файлового хранилища в локальной сети образовательного учреждения, но оно не решает задач, обусловленных специфическими особенностями как самой хранимой информации, так и приемов работы с ней.

При работе с хранилищем исходного кода программы необходимо также обеспечить безопасный и эффективный удаленный доступ, что в случае использования простого файлового хранилища в сети образовательного учреждения затруднено или невозможно. Это создает для студентов необходимость использования различных носителей информации (flash-накопители и

пр.), что сопряжено с различными проблемами (утрача носителя, угроза заражения вредоносным программным обеспечением, использование ресурсов файлового хранилища для хранения информации, не имеющей отношения к учебному процессу). Выполнение резервного копирования своих работ в этом случае также обычно возлагается на студентов.

При совместной работе над проектом возникает проблема отслеживания и синхронизации изменений в разрабатываемой программе, что требует применения специальных средств для обеспечения безопасного совместного доступа к файлам проекта.

Поскольку студенческие работы в ряде случаев имеют определенную практическую значимость, следует обеспечить возможность представлять работы для независимого тестирования и обсуждения, что также обуславливает необходимость организации безопасного удаленного доступа к хранилищу.

При разработке сложных проектов возможна ситуация, когда допущенную в программе ошибку становится сложно найти из-за большого количества изменений в соответствующем модуле. В этом случае необходимо иметь возможность вернуться к предыдущей версии исходного кода программы. При использовании обычного файлового хранилища автоматический откат версии файлов часто невозможен (тре-

бует использования специализированных файловых систем).

Эффективным решением перечисленных задач является использование систем управления версиями — класса программного обеспечения, предназначенного для работы с часто изменяющейся информацией. Основные возможности систем управления версиями включают в себя:

- Хранение файлов — независимо от типа, что позволяет хранить не только исходный код программы, но и используемые программой ресурсы. В отличие от файлового хранилища, система управления версиями может рассматривать хранимый проект (набор файлов) как один объект либо как совокупность отдельных файлов (в зависимости от используемого программного обеспечения и его настроек).
- Сохранение информации о различиях между версиями файлов — это позволяет при необходимости вернуться в одну из предыдущих версий отдельного файла или всего проекта.
- Возможность создания ответвлений проекта — вариантов исходного кода программы с различным функционалом. Это позволяет тестировать новые функции в отдельном проекте с возможностью дальнейшего объединения ветвей и добавления новых функций в основную ветку.
- Возможность автоматического создания патчей — наборов данных, содержащих информацию о различиях между определенными версиями исходного кода, что позволяет обновить рабочую копию программного проекта без возможности непосредственного доступа к репозиторию.
- Возможность безопасного удаленного доступа к хранилищу — в ряде систем используется несколько различных способов: с помощью специализированного клиентского ПО, непосредственно среды разработки, веб-браузера.
- Интеграция с различным программным обеспечением (среды разработки программ, системы автоматизированного проектирования и т. п.).

Системы управления версиями успешно применяются в различных образовательных учреждениях (в России — преимущественно в виде спецкурсов и факультативов, например, в МГУ) и компаниях-разработчиках программного обеспечения по всему миру (ввиду практической необходимости и целесообразности). Такие хранилища часто используются также для распространения разработанных программ в виде исходного кода (обычная практика для

свободного программного обеспечения).

Существующие системы управления версиями различаются по своей архитектуре и функциональности, некоторые из них специфичны для определенных сред программирования или проектирования (такие, как Microsoft Team Foundation), но основная часть универсальна. Наиболее распространены такие системы, как CVS, Subversion, Git, Mercurial, Bazaar.

Цикл работы над проектом с использованием системы управления версиями включает в себя три основных этапа:

- создание или обновление рабочей копии — набора последних версий файлов проекта, полученных с сервера;
- изменение файлов проекта;
- сохранение изменений — при этом на сервер загружаются только измененные файлы (или информация о различиях между рабочей и сохраненной копиями) с отметками об авторе, времени изменения и, опционально, с комментарием.

При сохранении изменений могут возникать конфликты, в случае если один и тот же файл был одновременно изменен несколькими разработчиками. Обычно в таких случаях система не в состоянии самостоятельно обновить проект и требует вмешательства со стороны разработчика. Тем не менее, существует возможность предотвратить конфликты путем блокировки отдельных файлов — в этом случае файл может быть изменен только тем пользователем, который запросил такую блокировку.

Наиболее распространенные в настоящее время среды разработки, как правило, поддерживают, по крайней мере, одну систему управления версиями, часто поставляемую в составе пакета программ, включающего саму среду. В ряде случаев за счет использования модулей расширения (для сред, которые их поддерживают) можно обеспечить возможность работы с любой из распространенных систем. В качестве примеров сред разработки, имеющих поддержку всех используемых при совместной работе методик работы с исходным кодом программы в системе управления версиями, можно привести:

- Eclipse — полная интеграция с системой управления версиями Subversion с использованием модуля Subclipse или Subversive, поддержка других систем управления версиями;
- Embarcadero RAD Studio — поддержка Subversion, имеется также своя встроенная система управления версиями;
- Microsoft Visual Studio — интеграция с Team Foundation Server, возможность использования других систем управле-

ния версиями с помощью сторонних модулей.

Использование систем управления версиями в учебном процессе при обучении программированию позволяет решить следующие задачи:

- организация централизованного и безопасного хранилища исходных кодов программ студентов;
- представление разработок студентов на интернет-портале образовательного учреждения;
- возможность удаленного доступа для студентов к их разработкам и лабораторным работам [2];
- подготовка студентов к использованию систем управления версиями в профессиональной деятельности для обеспечения совместной работы над проектами;
- подготовка студентов к использованию публичных репозиториях программного обеспечения (SourceForge, GitHub, Google Code и др.).

Таким образом, представляется актуальным и целесообразным обучение работе с системами управления версиями и использование их в рамках курса программирования в учебных заведениях.

Обучение использованию систем управления версиями в Уральском государственном педагогическом университете входит в курс изучения языков программирования С и С++ в рамках специализации «Компьютерные игровые технологии в образовании». Данные языки наиболее часто используются при разработке крупных программных проектов, в связи с чем применение какой-либо системы управления версиями как для непосредственно отслеживания изменений в файлах проекта, так и для взаимодействия разработчиков является необходимым. Это актуально и для игрового программного обеспечения, на принципах проектирования и реализации которого и основывается специализация.

Изучение систем управления версиями осуществляется на примере системы Subversion. В качестве серверной платформы используется веб-сервер Apache с модулем `mod_dav_svn`, работающий наряду с обычным для данной системы сервером `svnserve`. Это позволяет рассмотреть особенности работы с репозиторием как с использованием стандартных средств (специализированный Subversion-клиент, а также модули интеграции для сред разработки), так и через веб-интерфейс WebSVN.

При изучении системы управления версиями студентам предлагается создать многомодульный проект (точные условия учебного задания могут варьироваться) и

поместить его в репозиторий Subversion. По мере выполнения учебного задания требуется периодически сохранять изменения в репозиторий, делая их доступными для студентов, работающих над другими модулями. Каждое обновление проекта должно сопровождаться кратким описанием изменений с момента предыдущего обновления.

Применение системы управления версиями в учебном процессе позволило обеспечить эффективность обучения, что было выявлено в процессе разработки игровой программы студентами, обучающимися по специализации «Компьютерные игровые технологии в образовании». Дисциплины специализации вводились в учебный процесс со второго года обучения и рассматривали все основные этапы жизненного цикла программы, начиная от проектирования и реализации, (дисциплины «Игровые проекты» и «Программирование на С / С++») и заканчивая сопровождением.

На этапе проектирования игровой программы система управления версиями использовалась для обмена информацией о проекте (UML-диаграммы, текстовая информация, графические изображения, задающие концепцию будущей компьютерной игры) между студентами. Сохранение имени автора того или иного изменения в репозитории, наряду с временными метками, позволило не только обеспечить контроль за своевременным выполнением заданий, но и проследить его динамику. В некоторых случаях возникала необходимость вернуться к предыдущей версии отдельных файлов проекта, поскольку по мере развития проекта некоторые из решений оказывались нецелесообразными и возникала необходимость в их изменении.

На этапе реализации проекта на языке программирования С++ система управления версиями использовалась для хранения файлов исходного кода проекта и используемых ресурсов. Разрабатываемая программа имела модульную архитектуру, что позволило распределить обязанности по разработке модулей между различными студентами. Таким образом, система управления версиями служила в качестве инструмента для обеспечения совместной работы над проектом.

При реализации программного кода становилась актуальной задача синхронизации изменений, когда отдельные модули программы использовали различные версии межмодульного интерфейса. Возможность проследить историю изменений файлов проекта позволила внести соответствующие корректировки в модули, использующие утратившие актуальность интерфейсы.

Задача синхронизации изменений оставалась актуальной на протяжении всего времени разработки программы, поскольку значительная часть программного проекта была реализована в рамках внеаудиторной работы.

Использование системы управления версиями, таким образом, позволило студентам не только проследить все этапы разработки программы, но и получить навыки использования средств коллективной разработки. Проблемы, являющиеся актуальными при коллективной разработке сложной многомодульной программы, удалось

эффективно решить, используя соответствующие функции системы управления версиями. Решение данных проблем при использовании файлового хранилища было бы неэффективным ввиду отсутствия требуемого функционала у файловых систем общего назначения.

Рассмотренный подход позволяет эффективно освоить основные приемы работы с системами управления версиями и обеспечить качественное усвоение учебного материала по программированию на различных языках.

ЛИТЕРАТУРА

1. ГАЗЕЙКИНА А. И. Обучение программированию будущего учителя информатики // Педагогическое образование в России. 2012. №5.
2. ЛАПЁНОК М. В. Информационная среда дистанционного обучения как средство реализации индивидуализированного обучения в общей школе // Вестник Московского государственного гуманитарного университета им. М. Шолохова. Сер. «Педагогика и психология». 2011. Вып. 4.
3. СЕРДЮКОВ В. И. О количественном оценивании достоверности результатов автоматизированного контроля знаний // Информатика и образование. 2010. №3.

Статью рекомендует канд. техн. наук, доц. М. В. Лапёнок