

ИСПОЛЬЗОВАНИЕ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

ТЕНИГИН А.А.

*Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича, Российская Федерация*

Аннотация: В статье рассмотрена новая методика внедрения системы контроля версий git в образовательный процесс студентов технических специальностей, изучающих программирование, а также первые результаты использования такой методики. В работе изложена структура и основные этапы внедрения системы, а также достоинства этого подхода. Ключевые слова: система контроля версий, git, обучение программированию, подготовка востребованных кадров.

THE USAGE OF GIT VERSION CONTROL SYSTEM IN TEACHING PROGRAMMING

TENIGIN A.A.

*Federal State Budget-Financed Educational Institution of Higher Education The
Bonch-Bruevich Saint Petersburg State University of Telecommunications,
Russian Federation*

Abstract: The new method of implementing git version control system into the educational process of technical students studying programming is set out in this article, along with the first results of using such method. The structure and main topics of the implementation process, and also the advantages of this method are also stated in this work.

Keywords: version control system, git, teaching programming, teaching of in-demand specialists.

В современной разработке система контроля версий git используется повсеместно – от маленьких проектов до больших, от простых скриптов до крупных коммерческих продуктов. Git позволяет управлять версиями файлов, добавлять комментарии, управлять проблемами, задачами, участниками работы над проектом и многое другое. Существуют исследования, показывающие, что знание системы контроля версий git является общим требованием различных вакансий вне зависимости от конкретной специализации программиста [1]. Выпускающиеся из классических учебных заведений студенты-разработчики сталкиваются с тем, что на работе от них требуют знание git-систем на гораздо более высоком уровне, нежели они даются в процессе учёбы. Более того, выполнение git-команд для работы с репозиторием – это малая часть рабочего процесса программиста, которая должна выполняться как бы сама собой и между делом. Отсутствие большой практики работы с системами контроля версий очень сильно мешает молодым специалистам на первом месте работы.

Эта проблема вынуждает по-новому взглянуть на процесс работы со студентами во время учебных занятий, а также на проверку домашних

заданий, курсовых проектов и других результатов их самостоятельной работы. В данной статье предлагается новая система работы, основанная на той простой идее, что работа с системой контроля версий должна сопровождать студента от первого и до последнего дня его учёбы так же, как будет сопровождать его на работе. Описанная система уже второй год подряд предлагается студентам на занятиях по дисциплинам, напрямую связанных с программированием.

Чтобы обеспечить непрерывную работу с системой git на протяжении всего обучения, на первых же занятиях (сразу после вводной и организационной части) предлагается обучать студентов системе git. К счастью, для использования git-репозитория не требуется знание языков программирования - для начала в свой первый репозиторий студент может отправлять обычные файлы. Предпочтительным вариантом будут являться файлы текстовые – по ним будет легче отслеживать вносимые изменения и наглядным образом видеть результат своей работы.

Таким образом, различные возможности системы контроля версий и сайтов для размещения удаленных репозитория (например, github.com или gitlab.com) внедряются в процесс обучения студентов по мере изучения нового связанного с ними материала. Предлагаемая структура занятий выглядит следующим образом:

- 1) Описание и основные возможности. Локальные и удаленные репозитории. Установка и настройка git на рабочей машине. Удаленный репозиторий, создание аккаунта в удаленной git-системе, пароли и токены. Добавление, удаление изменений. Коммиты. Управление ссылками на удаленный репозиторий. Скачивание и копирование репозитория.
- 2) Работа с ветками. Создание веток. Переключение между ветками. Слияние веток. Работа с конфликтами слияния. Особенности работы с ветками на рабочей машине и на сайте. Pull requests.
- 3) Дополнительные возможности. Логи и графы коммитов. Откат коммитов (reset). Изменение истории коммитов (rebase). Скрытие изменений (stash) [2].
- 4) Командная работа в удаленном репозитории. Совместная работа в github.com. Управление разрешениями. Настройка исключения загружаемых файлов (gitignore). Проекты (projects). Задачи и управление командой проекта. Проблемы (issues). Pull requests в командной работе. Код ревью. Чувствительные данные. Конфигурационные файлы.
- 5) Добавление github workflows к репозиторию. Добавление автоматической проверки линтеров и тестов программного кода [3].

На первый раздел рекомендуется выделять от 4 до 6 академических часов дисциплины. Выполняемые в этом разделе операции предполагают также наличие у обучающихся минимальных знаний в администрировании их операционной системы. Необходимо предоставить подробный полный теоретический материал с подробными объяснениями функционирования системы и управляющих команд. Уже по завершению первого раздела можно

требовать от студентов сдавать задания только через систему git. Также рекомендуется требовать от студентов загружать в git и классные работы тоже, это приводит к увеличению количества самостоятельной работы с системой контроля версий, а значит, и к хорошему запоминанию основных команд. К сожалению, удаленной проверки домашних заданий зачастую бывает недостаточно, потому что это сильно упрощает процесс списывания. Таким образом, эта система не исключает необходимость очной защиты студенческих работ. Этот недостаток системы можно считать возможностью для её доработки в будущем.

Второй и третий раздел можно давать между основными разделами дисциплины. Такой подход добавит разнообразие в учебный процесс, а приобретенных в первом разделе знаний вполне достаточно для освоения следующих двух.

Четвертый раздел (командная работа) удобнее будет давать с наступлением первых командных заданий. Это может быть работа с базами данных или просто крупное задание, которое студенты могли бы разделить на части, делегировав их выполнение членам своей команды. Командная работа хорошо сказывается на заинтересованности студентов учебным процессом. Более того, к четвертому разделу преподаватель уже наверняка будет иметь представление об успеваемости и мотивации каждого из обучающихся. Таким образом, студентов можно разделить на команды - например, по 3 человека. Это похоже на стандартную систему разделения в разработке и программировании – Team Lead, Middle Developer, Junior Developer (руководитель команды, разработчик и младший разработчик). Сильнейших студентов группы можно назначить руководителями команд, а отстающих – младшими разработчиками. Это хорошо скажется на мотивации всех членов команды, потому что теперь дифференцирование заданий по сложности будет происходить само собой – амбициозные студенты получают желанную нагрузку, а “младшие разработчики” в их команде получают посильные задачи от руководителя команды.

Пятый раздел должен даваться студентам после соответствующих разделов дисциплины по линтерам и автоматическому тестированию. Линтер – это программа, которая проверяет (и, возможно, исправляет) оформление программного кода. Автоматическое тестирование кода позволяет при внесении изменений, сохранении или загрузке кода в удаленный репозиторий проверять основной его функционал на наличие ошибок. Система git позволяет настроить этот процесс, используя интуитивный и удобный интерфейс. Проверка коммитов на оформление и наличие ошибок поднимает работу команды на качественно новый уровень, а также значительно уменьшает количество выполняемой руководителем группы работы, при этом обеспечивая ему удобный функционал объективной оценки выполняемой членами команды работы.

В сумме эти разделы позволяют изучить основной функционал системы контроля версий git, который будет им необходим в будущей работе.

Эти знания будут регулярно закрепляться и использоваться в процессе выполнения, сохранения и загрузки классных и домашних заданий, а также курсовых и командных проектов. Благодаря внедрению такой системы, уже после первого курса студенты будут не только обладать знанием языков программирования и некоторых связанных технологий, но и будут уметь работать в команде на профессиональном уровне. Описанная в статье методика применяется в обучении студентов, а результатами её применения можно считать лёгкое и постепенное изучение функционала и возможностей системы контроля версий git и веб-сервисов github.com и gitlab.com, а также желание студентов использовать их в проектах и заданиях даже в условиях, когда это не требуется преподавателем напрямую, что показывает личную заинтересованность и понимание необходимости работы с системой.

Список литературы:

1. Калевко В.В., Лагерев Д.Г., Подвесовский А.Г., “Управление образовательной программой вузов в контексте подготовки конкурентоспособных разработчиков программного обеспечения”, Журнал “Современные информационные технологии и ИТ-образование” Том 14 №4, 2018 г.

2. Scott Chacon, Ben Straub “Pro Git” APress, 2022. - 499 с.

3. Основные функции командной работы в github.com. Электронный ресурс. [<https://github.com/features>]. Дата обращения 10.10.2022.