

УДК 378.147

*А. В. Аксенов\**

старший преподаватель

\*Санкт-Петербургский государственный университет аэрокосмического приборостроения

## ИСПОЛЬЗОВАНИЕ НОВЫХ ТЕХНОЛОГИЙ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРАКТИЧЕСКОГО КУРСА РЯДА ТЕХНИЧЕСКИХ ДИСЦИПЛИН

Предложен сценарий использования в образовательном процессе практического курса технической дисциплины системы контроля версий Git и сервиса GitHub, проанализированы достоинства и недостатки данного подхода.

**Ключевые слова:** системы управления версиями, новые технологии образовательного процесса.

*A. V. Aksenov\**

Senior Lecturer

\*St. Petersburg State University of Aerospace Instrumentation

## NEW TECHNOLOGIES FOR INCREASING EFFECTIVENESS OF PRACTICAL WORK IN A NUMBER OF TECHNICAL SUBJECTS

Proposed is a method of using Git version control system in educational process during practical work of some technical subject, analyzed are advantages and disadvantages of such an approach.

**Keywords:** version control systems, new educational technologies.

Образовательный процесс дисциплин технических специальностей во многих учебных заведениях лишен некой технологической составляющей, которая выгодно бы отличала технические дисциплины от нетехнических в глазах обучающихся, а также использовала бы особенности практических заданий для повышения эффективности обучения и, возможно, облегчения работы всех вовлеченных сторон.

### **Публикация и распространение учебных материалов: классический подход**

Методические указания, учебные пособия и другие виды инструкций и материалов теоретического характера публикуются в виде doc- или pdf-файла и распространяются преподавателем в лучшем случае посредством электронной почты или облачных сервисов, в худшем – путем сохранения на компьютере лаборатории с последующим скачиванием. Другой способ предполагает публикацию бумажной версии материалов средствами учебного заведения, которая впоследствии распространяется через библиотеку. Ни тот, ни другой способ не могут быть признаны отвечающими современным тенденциям, не являются достаточно надежными и не лишены ряда существенных недостатков.

Учебные материалы (в особенности технических дисциплин) подвержены довольно быстрому устареванию и, как следствие, должны периодически обновляться. Практика показывает, что использование традиционных средств публикации и распространения материалов ведет к «конфликтам версий», то есть явлениям, когда различные группы студентов располагают различными версиями материалов, взятыми в библиотеке или позаимствованными у студентов предыдущих лет обучения или других потоков. Эти конфликты, в свою очередь, порождают неоднородность в выполнении студентами заданий и, следовательно, сложности при их оценивании и аттестации.

Еще одной важной проблемой является невозможность оперативного внесения изменений в опубликованные таким образом материалы. В самом деле часто ошибки или недочеты обнаруживаются в материале уже после того, как он опубликован и распространен. В этом случае преподаватель вынужден предпринимать ряд действий, направленных на оповещение всех студентов о выходе новой версии материалов (в случае использования электронных публикаций) или исправлять ошибки во всех экземплярах бумажной версии материалов, а возможно, и задуматься о публикации их новой версии. Но поскольку этот процесс занимает месяцы, такой подход нельзя назвать приемлемым. Как нетрудно догадаться, прямым следствием таких ситуаций является тот же самый «конфликт версий».

В ряде случаев первыми людьми, обнаруживающими ошибки в материалах, полученных от преподавателя, являются студенты. Как им поступить в такой ситуации? Они могут оповестить преподавателя об ошибке или недочете (его дальнейшие действия рассмотрены выше), а могут самостоятельно внести исправления и распространить их. К сожалению, такой подход также не функционален, поскольку в этих ситуациях об изменениях чаще всего имеет представление только одна группа на потоке, в которой учится студент, обнаруживший ошибку, или часть этой группы, которая входит в круг общения этого студента. Такие ошибки могут пережить не одно переиздание материала. Таким образом, спектр действий студентов не так уж широк и не предполагает варианта активных действий, которые могли бы привести к быстрому и эффективному исправлению ошибки и распространению новой версии материала среди других студентов с избежанием «конфликта версий».

### **Выполнение и сдача лабораторных и курсовых работ: классический подход**

Практические работы, связанные с написанием кода, обычно сдаются двумя основными способами: код демонстрируется на экране компьютера в лаборатории, будучи принесен на портативном накопителе, либо на экране личного компьютера студента. Ни тот, ни другой способ нельзя назвать отвечающим требованиям надежности. Нередки ситуации, когда поводом не сдавать лабораторную или курсовую работу является поломка накопителя или компьютера студента либо потеря накопителя.

Также удобно было бы иметь средство дистанционной сдачи практических работ, предоставляющее возможность получения студентом обратной связи от преподавателя и поддержкой версионирования выполняемых работ.

С другой стороны, хранение сданных работ также может потенциально представлять интерес.

### **Применимость и актуальность предлагаемого метода**

Предлагаемый подход будет наиболее применим и целесообразен для тех дисциплин, у которых конечным и главным результатом хотя бы части лабораторных работ является код, иными словами – текст программы. У ряда направлений количество таких дисциплин в учебном плане может быть значительным относительно их общего числа, в связи с чем поднятая проблема и предлагаемые методы ее решения могут быть признаны актуальными.

### **Технологические принципы системы управления версиями Git**

Технологической платформой для реализации предлагаемого подхода является использование системы управления версиями Git и сервиса GitHub.

Git является системой управления версиями, на данный момент представляющей собой стандарт де-факто инструмента, используемого при разработке программных продуктов любой степени сложности как с открытым исходным кодом, так и проприетарных.

GitHub – наиболее популярный хостинг проектов, использующих Git для версионирования.

Необходимо остановиться на терминологии, используемой системой Git и сервисом GitHub, поскольку описание предлагаемых методик будет происходить в контексте практик использования этой технологической платформы.

Система управления версиями (version control system, VCS) – программный продукт, который позволяет ассоциировать с некоторым проектом не только его текущее состояние, но и всю историю изменений в нем, а также возможность документировать эти изменения и переключаться между состояниями проекта в разные моменты времени. На рынке представлено большое количество таких систем. В дальнейшем будем придерживаться терминологии и идеологии системы управления версиями Git.

Каталог, содержащий в себе все файлы проекта, а также историю его версий, называется «репозиторием» (repository) проекта.

Один фиксированный элемент истории проекта (одно из его исторических состояний) называется «коммит» (commit). Пользователь системы управления версиями может создавать коммиты для любых изменений в любых файлах репозитория и документировать эти коммиты (и соответствующие им изменения).

Существует возможность создавать серверную копию публично доступного репозитория в другом аккаунте. Этот процесс называется «форк» и широко распространен в сообществе разработчиков программного обеспечения с открытым исходным кодом. Это может преследовать разные цели:

- внесение собственного вклада в развитие чужого проекта (для этого можно вносить соответствующие улучшения в свою копию проекта, а впоследствии путем использования запроса на включение предложить принять их в исходный проект его хозяину);
- создание собственного проекта на базе существующего.

После создания форка его автор вправе изменять проект на собственное усмотрение, создавая в истории версий проекта коммиты своего авторства. Если эти изменения, по его мнению, достойны быть скопированными обратно в исходный проект, он имеет возможность создать запрос на включение – уведомление из форка, формируемое в адрес владельца исходного репозитория, о намерении владельца форка предложить свои изменения. Запросы на включение являются основным способом вклада в проекты с открытым исходным кодом лицами, не являющимися членами проекта, но имеющими желание помочь [1].

### **Суть предлагаемого подхода**

Преподаватель и студент регистрируются на GitHub, устанавливают на свои компьютеры клиентское программное обеспечение Git и конфигурируют его.

На GitHub в качестве репозитория преподавателем публикуются:

- методические указания и прочие учебные материалы;
- шаблонный код для лабораторных работ (по одному репозиторию на каждую работу).

Сдача лабораторных работ происходит по следующему сценарию.

1. Студент делает форк репозитория с лабораторной работой и клонирует его себе на компьютер. При этом он получает шаблонный код, который может быть использован для более быстрого ориентирования в проекте, если работа имеет сложную структуру.

2. При необходимости преподаватель выдает индивидуальное задание, записывая его в форк студента.

3. Студент инкрементально выполняет задание по лабораторной работе и фиксирует промежуточные изменения в системе управления версиями (локально и синхронизируя с сервисом GitHub).

4. Когда выполнение работы закончено, студент фиксирует последние изменения и делает запрос на включение в исходный репозиторий, созданный преподавателем. Этот запрос трактуется как сообщение о намерении сдать лабораторную работу.

5. Сервис непрерывной интеграции (если он настроен) запускает задания (сборку, тестирование, статический анализ кода).

6. Преподаватель получает уведомление о запросе студента, оценивает проделанную работу, видит отчет сервиса непрерывной интеграции. После этого он может ее принять или дать обратную связь. При этом поддерживается как просто переписка со студентом, так и комментирование конкретных участков кода.

7. Студент, получив обратную связь от преподавателя, в случае необходимости модифицирует работу, фиксирует изменения и уведомляет преподавателя о необходимости повторного рассмотрения.

8. В случае если для изучаемой дисциплины результатом лабораторной работы является HTML-код, CSS-код или код на JavaScript, существует возможность просмотра преподавателем результатов выполнения работы прямо на GitHub с использованием встроенного хостинга сайтов GitHub Pages без необходимости скачивания кода на свой компьютер [2].

Размещение учебных материалов в репозитории GitHub под контролем системы управления версиями Git дает ряд преимуществ:

- Преподаватель может в любой момент изменять материалы, дополняя их или исправляя ошибки. При этом студенты моментально видят результат.
- Студенты также видят историю изменений учебных материалов, что позволяет им лучше ориентироваться в эволюции курса (если она есть).
- Если студент видит ошибку в учебных материалах, он может сделать форк материалов, исправить ее и создать запрос на включение, который увидит преподаватель. Последний вправе принять или отклонить изменение либо исправить его и применить к учебному материалу.

### **Недостатки предложенного подхода**

Помимо очевидных достоинств, использование сервиса системы контроля версий в практическом курсе дисциплины сопряжено с рядом недостатков.

И преподавателю, и студенту необходимо изучить дополнительную технологию, не связанную (или связанную опосредованно) с сутью преподаваемой или изучаемой дисциплины. Если с одной стороны это может рассматриваться как расширение кругозора и повышение технической грамотности, то с другой – чревато временными затратами.

Преподавателю дополнительно необходимо тратить значительные временные ресурсы на публикацию материалов, настройку сервисов контроля версий и непрерывной интеграции, проверку работ и изменений в учебные материалы, выдачу индивидуальных заданий.

#### **Библиографический список**

1. Chacon S., Straub B. Pro Git. URL: <https://git-scm.com/book/ru/v1> (дата обращения: 14.04.2019).
2. How CS50 uses GitHub to teach computer science. URL: <https://github.blog/2017-02-15-how-cs50-uses-github-to-teach-computer-science> (дата обращения: 14.04.2019).