

# ОПЫТ ИСПОЛЬЗОВАНИЯ GIT И GITHUB ПРИ ПРОВЕДЕНИИ ЗАНЯТИЙ СО СТУДЕНТАМИ

**Барвенков С. А., Станкевич А. А.**

*Белорусский Государственный университет, Минск, Беларусь,  
e-mail: bars@bsu.by, harald\_zealot@tut.by*

В рамках изучения курсов, связанных с бурно развивающимися информационными технологиями, стоит задача не только обучить студентов соответствующим знаниям, но и привить им навыки работы с такими технологическими инструментами, которые активно используются в профессиональной деятельности.

Поскольку реальная промышленная разработка ПО давно перестала быть делом одиночек, а требует умения работы в команде, то с нашей точки зрения надо, начиная с младших курсов, прививать студентам опыт совместной работы над проектами (например, в рамках изучения курса “XML технологии” или “Методы программирования”). Но, даже при выполнении индивидуальных заданий (например, лабораторных работ в рамках изучения курса “Программирование”) студентам стоит почувствовать, что разработка программного обеспечения давно стала чем-то большим, чем просто набором исходного кода программы. В частности, используется различный дополнительный инструментарий (багтрекеры, системы для управления проектами, систем контроля версий (СКВ) и т.д.). Как указано в [1], использование GitHub помогает “не только отслеживать изменения в версиях ..., но и практико-ориентированными методами показать студентам способы организации и ведения совместной работы над проектами”.

С другой стороны, при наличии в группе даже 15 учащихся преподавателю не хватает отведенного аудиторного времени для качественной проверки всех сданных работ. Поэтому нам хотелось так организовать учебный процесс, чтобы максимально использовать онлайн-сервисы. В частности, желательно иметь следующие возможности для преподавателя и для учащихся:

- публиковать в сети задания и оценки;
- просматривать студенческие листинги, контролируя время их размещения и авторство;
- устанавливать dead-line т.е. запрещать с определенного момента запись работ;
- создавать комментарии не только в целом к проекту, но и к определенным строкам программы. Указывать на недочеты в проделанной работе и размещать указания к выполнению заданий;
- в любой момент студенту пользоваться своим и общим репозиториями;
- автоматически получать студентом уведомления о проверке работы и комментариях преподавателя.

Всеми этим требованиям удовлетворяет СКВ git в связке с облачным сервисом GitHub (web-репозиторий для проектов с открытым исходным кодом, использующих git для контроля версий). Важной особенностью git является то, что можно работать на локальном компьютере с периодическим обновлением репозитория (синхронизация) на GitHub. Базовым элементом git репозитория является коммит (commit) – зафиксированное пользователем состояние репозитория. К любому из коммитов можно вернуться и

посмотреть изменения и дополнения по сравнению с другими коммитами. Именно это позволяет преподавателю прямо в тексте сданных программ указывать на недочеты и ошибки. Используя GitHub легко находить изменения в коде, внесенные для исправления предыдущих ошибок. Это позволяет не просматривать заново листинг у каждого студента, а видеть только те части, в которые были внесены изменения.

GitHub предоставляет также широкий набор дополнительных инструментов в частности просматривать “историю” любого файла, выяснять авторство любой строки кода проекта, визуализировать топологию ветвей и коммитов, ставить студентами задачи в багтрекере, создавать электронную документацию и отображать оценки за работы используя простую викиразметку, собирать и визуализировать многочисленные статистики. Также на этом сервисе имеется возможность администрирования командной работы, что даёт преподавателю широкие возможности по управлению учебным процессом.

Ещё одним опробованным применением является сопровождение руководителем написания курсовой работы, если использовать указанные технологии совместно с технологией TeX, так как всё форматирование TeX хранит в текстовом виде.

Опишем также некоторые проблемы, возникшие у авторов при использовании GitHub (см. также [2]). Во-первых, не все студенты осознают всю широту возможностей СКВ, и пытаются использовать git+GitHub, как простой облачный файлообменник (аналогично iCloud, Dropbox, Google Drive и т.п.).

Во-вторых, поскольку многие используют ОС Windows, то часто у них возникает проблема с конвертацией кодировок файлов из Win1251 в UTF.

Во-третьих, любая СКВ рассчитана на работу в основном с текстовыми файлами. Поэтому преподавателю удобно работать с репозиториями студентов только, когда там хранятся лишь файлы, необходимые для создания и дальнейшей компиляции проекта. Однако студенты могут работать в разных IDE. И потому нам приходится создавать конфигурационные и прочие файлы самостоятельно (отдельно, например, для QtCreator, CodeBlock или Visual Studio), Эту проблему можно решить, используя ещё одну промышленную технологию: автоматизированные IDE-независимые сборщики (например, CMake) Внедрение этого подхода только запланировано. В то же время, если нет жесткого требования к языку программирования (например, в рамках курса “Компьютерная графика”), то надеяться, что на компьютере преподавателя будет развернута именно та система, которой пользовался студент – нереально. Поэтому студенты должны хранить в репозитории помимо исходных кодов также бинарники с готовым приложением. Но это не дает возможности проверить соответствие предоставленной программы с размещенным текстом кода.

В целом внедрение указанной связки технологий в учебный процесс существенно улучшило качество учебного процесса, облегчило работу преподавателя, и дало студентам чувство сопричастности современным технологиям промышленной разработки программного обеспечения.

### **Литература**

1. Шалтунович, А.В. Организация совместной разработки веб-приложений в рамках социальной сети github / А.В. Шалтунович // Вестник Нижневартковского гос. университета. – 2011. – № 3.