

Построение образовательной платформы вокруг системы контроля версий Git

Рыданов Н.С.

find.art.in.living@gmail.com

Саратовский государственный университет имени Н. Г. Чернышевского

Аннотация. Предметом обсуждения данной статьи является разрабатываемая образовательная платформа с учётом специфики образовательного процесса IT-специальностей.

Введение

Сегодня система контроля версий Git стала промышленным стандартом в IT-компаниях.

Отметим, что между промышленными и образовательными процессами достаточно много схожего. Например, процесс сдачи выполненной работы специалисту звена выше очень похож на сдачу задания преподавателю — что для преподавателя, что для сотрудника значение имеют прежде всего внесенные изменения. На эти изменения выполняющий работу получает обратную связь, после чего вносит правки и процесс повторяется.

Всё это даёт повод задуматься об образовательной платформе, которая включает в себя современные практики промышленного программирования.

О гипотетических преимуществах такой платформы, её внутреннем устройстве и ведущейся на текущий момент работе и пойдёт речь далее.

1. Польза внедрения Git в образовательную деятельность

Процесс образования, как правило, двусторонний, и в нём принимают участие студент и преподаватель. В этой связи очень важно внедрять инструменты, повышающие производительность труда участников образовательного процесса.

Кроме того, в образовательном опыте студента критически важно формирование полезных навыков, направленных на профориентацию.

1.1. Положительные эффекты для обучающегося

Процесс обучения тесно связан с экспериментами. В связи с этим очень важно обеспечить надёжный способ сохранять промежуточные результаты, различные версии работы.

Система контроля версий Git позволяет добиться этого множеством путей. Например, можно фиксировать стабильные версии (`git commit`), переключаться между ними (`git checkout`), временно или полностью отменять внесённые изменения относительно текущей стабильной версии (`git stash`, `git restore`), создавать параллельно существующие версии с различным набором изменений, с помощью веток (`git branch`) и даже интегрировать изменения одной параллельной версии в другую (`git merge`, `git rebase`).

Эти и некоторые другие возможности оказываются крайне полезными в групповой проектной деятельности в рамках образовательного процесса, либо внеучебной деятельности. Кроме вышеупомянутого функционала, Git предоставляет возможность синхронизации этих процессов (`git pull`, `git push`).

Наконец, ещё раз отметим, что процесс приобретения таких навыков носит профориентационный характер, поскольку именно Git лежит в основе взаимодействия команд разработки во многих компаниях.

1.2. Положительные эффекты для преподавателя

Преподавательский состав регулярно проверяет работы студентов. Согласно официальным данным от 2023 года всего 125176 студентов IT специальностей, в том время как число преподавателей составляет лишь 16430 человек. Таким образом, по официальной статике на одного преподавателя приходится около 8 студентов, а на практике это число может достигать 15-20 студентов.

В зависимости от специфики дисциплины, это число может быть как приемлемой нагрузкой, так и критически высокой. В последнем случае преподаватель либо снижает учебную нагрузку, чтобы справиться с контролем за успеваемостью, либо значительно уменьшает время, уделяемое одному студенту.

В связи с этим, кроме особое внимание должны привлекать инструменты, которые позволяют повысить производительность труда преподавателя, выражающуюся в том числе во времени проверки одного задания, сданного студентом.

В инструментарий Git включена возможность просмотра построчной разницы между фиксированными версиями. В ситуации, когда число изменений много меньше общего объема выполненной работы, этот инструмент позволяет быть уверенным, что только выделенные строки получили изменение, а всё остальное содержимое осталось без изменений.

Иллюстрация такого подхода приведена на рис. 1.

```
- let db_url = env::var("DATABASE_URL"?);  
+ let db_url = env::var("DATABASE_URL"?);  
  let db = PgConnection::new(&db_url).await?;  
  
  let ctx = Arc::new(Context {  
    config: config.clone(),  
    db,  
  });  
  
  log::info!("Connected to database");
```

Рис. 1. Демонстрация принципа работы команды diff

Кроме этого, Git предоставляет надежный контроль за временем и авторством сдачи заданий, что, безусловно, в интересах преподавателя.

1.3. Долгосрочный эффект от популяризации Git

Сегодня Git является основной технологией для распространения свободного программного обеспечения. Свободное программное обеспечение,

в свою очередь, является одной из опорных точек для создания независимой технологической базы. Как правило, свободно распространяемое ПО распространяется без ограничений, в том числе санкционных, а также проще в лицензировании.

На рис. 2 приведено изображение веб-ресурса, на котором размещены открытые разработки компании BaseALT.

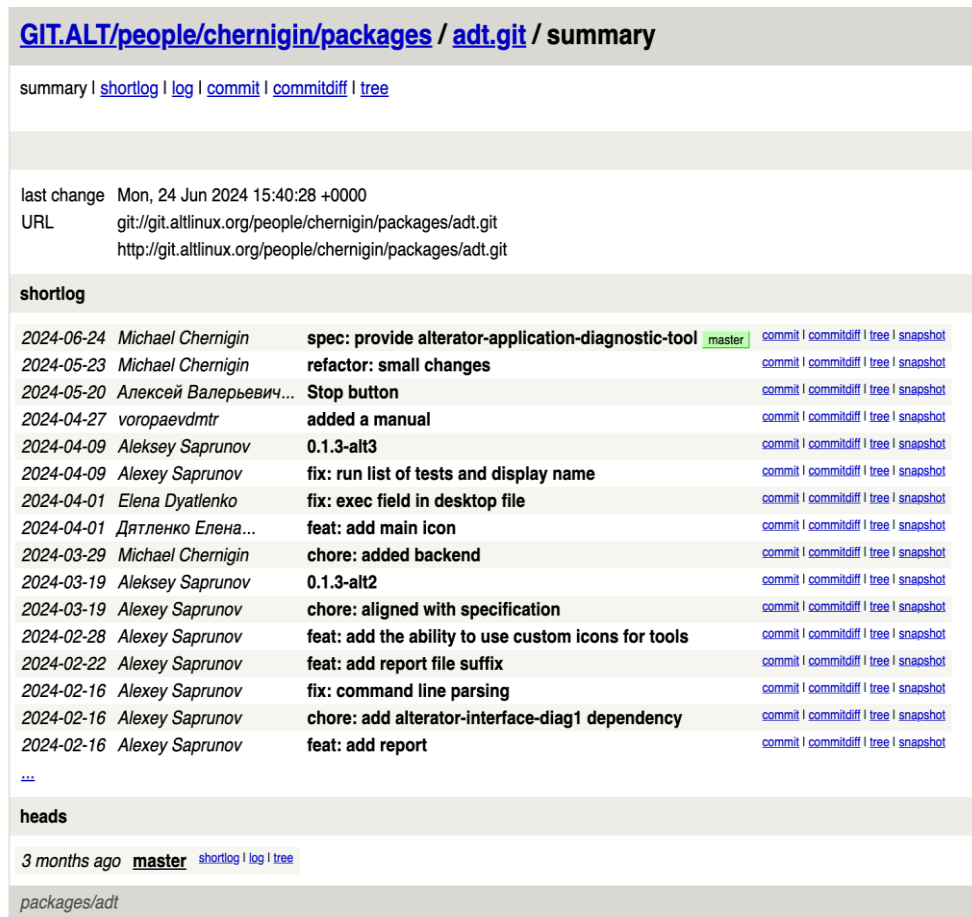


Рис. 2. Один из репозиториев, поддерживаемый компанией BaseAlt

2. Возможные сценарии внедрения системы контроля Git в образовательную деятельность

Ранее в рамках дисциплины “Языки программирования” были рассмотрены варианты интеграции Git в LMS Moodle, применяемую в СГУ им. Чернышевского.

Выбранный подход показал ряд недостатков, в частности:

1. требовал регистрации на стороннем сервисе, помимо регистрации на самой образовательной платформе
2. работал с внешним API, который не устойчив к блокировкам проблемам с международными сетями
3. требовал значительное число действий со стороны преподавателя, как на этапе внедрения плагина, так и на этапе зачисления студентов на курс

Всё это говорит о необходимости дополнительных вложений в функционал Moodle, полезность чего под вопросом, ведь:

1. платформа использует устаревшие, уязвимые технологии
2. имеет большую кодовую базу, затрудняющую изменения
3. использует ныне менее популярный язык PHP

Альтернативным вариантом было бы применение системы контроля версий параллельно с системой оценивания, однако в таком случае значительно растут временные издержки на перенос оценок.

Ещё одним вариантом было бы создание новой платформы, изначально спроектированной с учётом новых требований. Это, безусловно, трудоемко, однако позволит пересмотреть современные подходы к образованию студентов ИТ-специальностей.

3. Модель образовательной платформы

3.1. Пользовательский опыт

Образовательная платформа предоставляет два интерфейса взаимодействия:

1. привычный графический интерфейс, внешне схожий с сегодняшним стандартом в образовании — Moodle;
2. интерфейс командной строки, позволяющий взаимодействовать с Git, а значит, загружать изменения без авторизации.

В дополнение к готовому набору инструкций через SSH доступ предполагается доступ к утилите, позволяющей студенту зафиксировать попытку на текущей зафиксированной версии работы.

Предполагается, что при оценке заданий преподаватель может смотреть не только полный исходный код отправленного файла, но и внесенные изменения, игнорируя строки, в которых код программы не был изменён.

3.2. Внутреннее устройство платформы

При создании курса преподавателей для каждого студента создаёт Git репозиторий, в котором отслеживаются все изменения. При создании задания преподаватель с помощью маски файла выделяет из репозитория подмножество файлов, которое будет учитываться в рамках этого задания.

Затем, при отправке попытки сохраняется уникальный идентификатор (хэш) текущего состояния репозитория, и при проверке преподаватель получает файлы в зафиксированном на момент отправки попытки виде.

Если какое-то из заданий требует отдельного учёта (например, форма задания — проект), то возможно создание отдельного репозитория.

При необходимости, преподаватель может загрузить себе копию Git репозитория, чтобы получить все возможности, предоставляемые системой контроля версий.

3.2.1. Архитектура платформы

Разработка веб-интерфейса предполагает наличие front-end сервиса приложения. В данном случае в качестве основополагающей технологии был выбран веб-фреймворк Svelte.

Интерфейс командной строки предполагается реализовать через применение нестандартной оболочки операционной системы на стороне

сервера. Это позволит предоставить необходимый функционал, но при этом ограничить студентов от запуска иных нежелательных команд.

Оба интерфейса взаимодействуют с внутренней частью приложения, архитектура которой прорабатывается.

4. Образовательный эффект от разработки платформы

Сегодня работа над проектом ведется в группе студентов и выпускников факультета компьютерных наук и информационных технологий, в том числе в виде курсовых работ.

Такой подход весьма хорош тем, что студенты с младших курсов вовлекаются в полноценную команду разработки, что в свою очередь способствует развитию у обучающихся компетенций в профессиональной сфере, коммуникативных навыков, навыков командной работы и проектной деятельности.