



М.В. Денисова, С.Ю. Ржеуцкая
Вологодский государственный университет

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ В ДИСТАНЦИОННОМ ЛАБОРАТОРНОМ ПРАКТИКУМЕ ПО ИТ-ДИСЦИПЛИНАМ

В статье предлагается способ перевода в дистанционный формат лабораторного практикума по профильным ИТ-дисциплинам. Обоснован выбор базовой технологии непрерывной интеграции (CI) и поддерживающей ее платформы GitHub в качестве основы для организации лабораторного практикума, представлена архитектура рабочей среды для выполнения работ, описаны пошаговые алгоритмы действий преподавателя и студента.

Дистанционный лабораторный практикум, технология непрерывной интеграции, автоматизированное тестирование, платформа GitHub.

В процессе перевода обучения в дистанционный формат для многих дисциплин технических направлений самые серьезные проблемы возникают при организации дистанционного лабораторного практикума. Особую актуальность эта проблема имеет для профильных дисциплин ИТ-направлений, например курсов «Программирование», «Базы данных», «Распределенные системы обработки информации», развивающих базовые профессиональные компетенции ИТ-специалистов. В статье предлагается подход к организации дистанционного лабораторного практикума по программированию и технологиям баз данных, который основан на применении современных практик разработки программного обеспечения, адаптированных к их применению в учебном процессе.

На начальном этапе исследования было проанализировано современное состояние проблемы. Несмотря на большое количество публикаций, посвященных электронным и дистанционным образовательным технологиям, среди которых стоит выделить фундаментальную монографию [4], технологии перевода в дистанционный формат лабораторного практикума по ИТ-дисциплинам развиваются в основном в практической плоскости. Имеется немало количество программных продуктов, поддерживающих дистанционное обучение ИТ-специалистов – можно выделить такие очень популярные ресурсы, как Coursera, Codeforces, Stepik, SQLEx и некоторые другие. Студенты многих российских вузов используют для тренировки навыков дистанционный практикум по программированию и технологиям баз данных, разработанный для использования в учебном процессе и подготовки к олимпиадам по программированию на кафедре автоматики и вычислительной техники ВоГУ [2].

Все перечисленные ресурсы поддерживают возможность автоматической проверки программного кода, разработанного студентами в процессе обучения, поэтому с их помощью можно хорошо отработать практические навыки разработки программного обеспечения, необходимые квалифицированному ИТ-специалисту [3]. Однако ни один из них, включая и дистанционный практикум кафедры АВТ, не поддер-

живает в полной мере процессы подготовки лабораторных работ преподавателем и пошагового выполнения работы студентами. По этим причинам все указанные ресурсы рекомендуются студентам как прекрасное дополнительное средство обучения, но не как полноценную замену лабораторному практикуму.

В поисках технологий и программных средств, позволяющих максимально сохранить все преимущества традиционного лабораторного практикума и дополнить их возможностями современных цифровых технологий, внимание привлекла технология разработки программного обеспечения, известная под названием «непрерывная интеграция» (Continuous Integration, имеет устойчивую аббревиатуру CI). Суть CI состоит в регулярных сборках и автоматическом тестировании разрабатываемого программного продукта (могут выполняться модульные, интеграционные и системные тесты) [5]. Таким образом, использование данной технологии позволяет обеспечить более универсальный и гибкий механизм контроля результатов выполнения лабораторных работ студентами, чем использование перечисленных выше обучающих ресурсов с системой автоматической проверки программного кода, которые фактически поддерживают только модульное тестирование.

Тем не менее, CI – это технология и практика разработки программного обеспечения, но не обучения информационным технологиям и программированию. В связи с этим потребовалась ее адаптация к учебному процессу, которая была выполнена для лабораторных практикумов по двум профильным для ИТ-направлений дисциплинам – «Базы данных» (бакалавриат) и «Распределенные системы обработки информации» (магистратура).

При адаптации технологии непрерывной интеграции к учебному процессу решающее значение имеет выбор подходящего программного обеспечения, которое должно быть доступным дистанционно для преподавателей и студентов, а также бесплатным. В ходе исследования в качестве возможных вариантов были проанализированы платформы GitHub, Bitbucket и GitLab. Каждый из этих известных веб-ресурсов

представляет собой хостинг для разработки, сборки и автоматического тестирования программного обеспечения и контроля версий, т.е. полностью поддерживает технологию CI.

Выбор был сделан в пользу платформы GitHub [1] благодаря ее обширным функциональным возможностям, доступности и наличию бесплатных аккаунтов (с разумными ограничениями, подходящими для учебного лабораторного практикума). GitHub предоставляет пользователям различные инструменты разработки программного обеспечения, среди которых можно выделить распределенную систему управления версиями программного продукта Git, а также систему непрерывной интеграции GitHub Actions, которая позволяет создавать конвейеры непрерывной интеграции и непрерывного развертывания для тестирования, выпуска и развертывания программного обеспечения без использования сторонних платформ. По мнению авторов, указанные программные средства могут служить хорошей основой при подготовке и проведении дистанционных лабораторных работ по профильным ИТ-дисциплинам.

Общая архитектура рабочей среды, в которой происходит дистанционное выполнение лабораторных работ, представлена на рисунке 1.

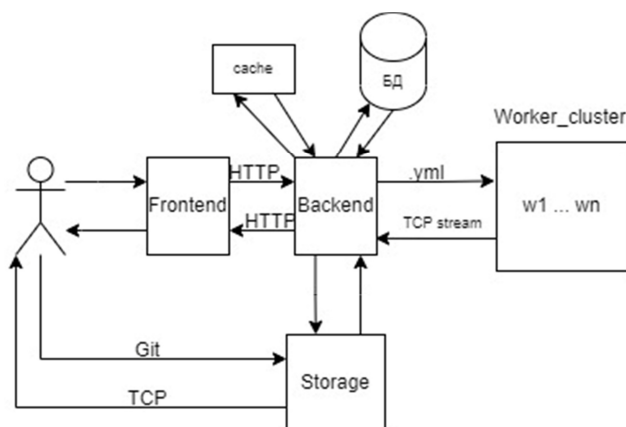


Рис. 1. Общая архитектура среды для дистанционного выполнения лабораторных работ с использованием платформы GitHub

С помощью frontend пользователь (преподаватель или студент) взаимодействует с сайтом GitHub, с помощью backend происходит взаимодействие с разными системами, такими как базы данных, кеш (отвечает за распределение задач), storage (хранит историю изменений программного продукта, который студент загружает при помощи Git в процессе выполнения лабораторной работы) и worker_cluster (рабочий кластер), на котором развернуты основные программные средства и методические материалы, необходимые для выполнения лабораторных работ. Состав и структура рабочего кластера определяется исходя из цели и задач каждой лабораторной работы или темы, объединяющей несколько лабораторных работ. Этот компонент рабочей среды формируется преподавателем в процессе подготовки работ.

В качестве конкретного примера организации рабочего кластера рассмотрим одну из тем лабораторного

практикума по дисциплине «Распределенные системы обработки информации» (PCOI) – «Разработка распределенных приложений». Тема охватывает несколько лабораторных работ, на которых осваивается процесс разработки web-приложения с использованием современных средств разработки – платформы Node.js, библиотек React и Express и «облачной» СУБД MongoDB. В процессе выполнения лабораторных работ студент должен научиться создавать серверную часть web-приложения (backend) с использованием платформы Node.js и фреймворка Express, изучить удаленную работу с СУБД MongoDB, научиться разрабатывать клиентскую часть приложения (frontend) с использованием библиотеки React. В традиционном очном варианте практикума все компоненты web-приложения, разработанные студентом в ходе лабораторных работ, проверяются (тестируются) преподавателем вручную. Дистанционный практикум кафедры АВТ не поддерживает автоматическое тестирование распределенных приложений.

На рисунке 2 представлена структурная схема рабочего кластера для дистанционного выполнения лабораторных работ с автоматическим выполнением приемочного тестирования.

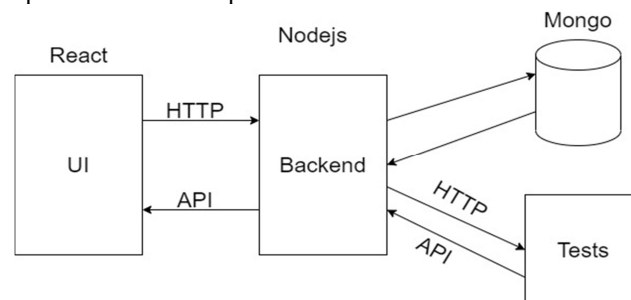


Рис. 2. Структурная схема рабочего кластера для выполнения лабораторных работ по теме «Разработка распределенных приложений»

Кластер содержит серверную часть приложения с использованием платформы Node.js (язык программирования JavaScript) и фреймворка Express, СУБД MongoDB, клиентскую часть приложения с использованием библиотеки React и тесты, написанные преподавателем.

Разберем последовательно процессы подготовки лабораторной работы преподавателем и выполнения работы студентом. Пошаговый алгоритм действий преподавателя при подготовке работы представлен на рисунке 3.

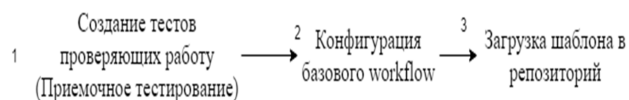


Рис. 3. Пошаговый алгоритм действий преподавателя при подготовке работ по теме «Разработка распределенных приложений»

Первый шаг преподавателя – создание приемочных тестов, которые будут покрывать основной функционал приложения. В нашем случае автотесты были написаны на языке Python. Безусловно, подготовка тестов – очень трудоемкий процесс, но при выполнении лабораторных работ студентами он избавит

преподавателя от ручной проверки результатов выполнения работы.

Второй шаг при подготовке лабораторной работы – конфигурация базового workflow (автоматизированный запуск этапов приема работы) с обязательными проверками. На GitHub создается репозиторий (хранилище всех файлов лабораторной работы) и выполняется настройка GitHub Actions на выполнение необходимых действий по приемке результатов выполнения лабораторной работы.

Наконец, заключительный шаг – загрузка шаблона в репозиторий, который имеет подготовленные тесты и настроенные автоматизированные проверки, т.е. результаты первых двух шагов.

Подготовленная таким образом лабораторная работа уже готова к дистанционному выполнению студентами. Пошаговый алгоритм выполнения работы студентами представлен на рисунке 4.

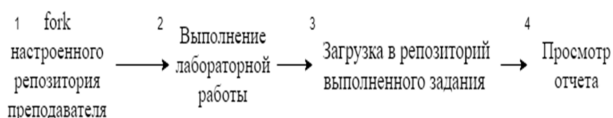


Рис. 4. Пошаговый алгоритм действий студента при дистанционном выполнении лабораторной работы

Студент делает fork (копирование) настроенного репозитория преподавателя, после чего работа выполняется в настроенной среде. Студент выполняет лабораторную работу согласно методическим указаниям, а затем загружает результаты с помощью git'a в GitHub.

В ответ на это событие GitHub Actions запускает автоматическое тестирование, после чего студент просматривает отчет о выполненных тестах. Если все тесты пройдены успешно, то лабораторная работа считается выполненной, если есть ошибки – в отчете будут они указаны, при этом будет выведен http-код. После исправления ошибок студент имеет возможность снова загрузить свои файлы в репозиторий и получить новый отчет о результатах проверки. Количество попыток сдачи результатов выполнения лабораторной работы не ограничено, основной критерий

приема работы – прохождение всех приемочных тестов без ошибок.

В настоящее время репозиторий на GitHub полностью подготовлен и настроен. Проведены эксперименты по дистанционному выполнению лабораторных работ магистрантами, которые показали хорошие результаты. Для того, чтобы не потерять возможности обратной связи, в течение всей лабораторной работы открыта видеоконференция в Google Meet.

Представленная в статье технология позволяет полностью сохранить весь функционал имеющегося лабораторного практикума, проводимого в очной форме, и дополнить его новыми возможностями. Благодаря автоматическому тестированию проверка результатов выполнения работы стала более качественной. Преподаватель избавился от рутинной и трудоемкой работы по проверке программного кода. Студенты в процессе выполнения работы получают опыт удаленной разработки программного обеспечения с использованием современной цифровой платформы.

Литература

1. GitHub: [Электронный ресурс]. – URL: <https://github.com/> Дата обращения 11.07.2022
2. Андрианов, И. А., Ржеуцкая, С.Ю., Харина, М. В. Междисциплинарный дистанционный практикум для студентов ИТ-направлений // Открытое образование. – 2021. – Т. 25. – № 2. – С. 41–50
3. Ржеуцкая, С. Ю., Харина, М. В. Способ автоматического подбора учебно-тренировочных заданий в информационной среде обучения студентов ИТ-направлений // Открытое образование. – 2020. – Т. 24. – № 2. – С. 17–28.
4. Роберт, И. В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты) – Москва : БИНОМ. Лаборатория знаний, 2014. – 398 с.
5. Сергушичева, А. П. Технологии разработки программного обеспечения : учебное пособие. – Вологда : ВоГУ, 2019. – 92 с.

M.V. Denisova, S.Yu. Rzhetskaya
Vologda State University

USE OF CONTINUOUS INTEGRATION TECHNOLOGY IN DISTANT LABORATORY PRACTICE ON IT DISCIPLINES

The article suggests a way to transfer a laboratory practice in specialized IT disciplines to a remote format. The choice of the basic continuous integration (CI) technology and the GitHub platform supporting it as the basis for organizing a laboratory practice is substantiated, the architecture of the work environment for performing work is presented, step-by-step algorithms of teacher and student actions are considered.

Distant laboratory practice, continuous integration technology, automated testing, GitHub platform.