



Общество с ограниченной ответственностью «Ребреин»

ИНН 7727409582, ОГРН 1197746106161

Адрес: 123056, город Москва, Большая Грузинская ул, д. 36а стр. 5а, офис 13

Утверждено

Приказом № ПР-1 от 17.06.2025 г.

Генеральный директор

 Фролкина Е.А.
«17» июня 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
– ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«УПРАВЛЕНИЕ ВЕРСИЯМИ И КОМАНДНАЯ РАЗРАБОТКА НА GIT»

Срок реализации: 2 недели (14 дней)

Количество часов: 20 акад. ч.

Форма обучения: заочная форма

Формат обучения: с применением
исключительно дистанционных технологий

Возраст обучающихся: для лиц старше 17
лет, имеющих или получающих среднее
профессиональное и (или) высшее
образование

Москва, 2025 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая дополнительная профессиональная программа – программа повышения квалификации «Управление версиями и командная разработка на Git» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Профессиональным стандартом 06.026 «Системный администратор информационно-коммуникационных систем», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 29 сентября 2020 г. № 680н;
- Профессиональным стандартом 06.001 «Программист», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. № 424н
- ФГОС высшего образования – бакалавриат по направлению 09.03.01 Информатика и вычислительная техника, утв. приказом Минобрнауки России от 19.09.2017 № 929 .
- Локальными нормативными актами ООО «Ребреин».

В данной программе учтены основные идеи формирования универсальных учебных действий обучающихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

Направленность программы: программа имеет техническую направленность.

Адресат:

Программа ориентирована на начинающих специалистов в сфере информационных технологий и инженерной инфраструктуры, осваивающих основы работы с системами контроля версий.

Будет полезна:

- сетевым инженерам,
- системным администраторам,
- разработчикам,
- техническим и релизным менеджерам,
- специалистам по тестированию,
- DevOps- и SRE-инженерам.

Курс подходит слушателям без опыта работы с Git, а также тем, кто хочет систематизировать и углубить знания для командной и проектной деятельности в IT-среде.

Требования к входным знаниям обучающегося: обучающемуся необходимы базовые навыки работы с терминалом / консолью.

Актуальность реализации:

Git (произносится как «гит») — это распределённая система контроля версий, созданная Линусом Торвальдсом для управления исходным кодом. Система контроля версий (от англ. «Version Control System, VCS») — программное обеспечение для облегчения работы с изменяющейся информацией. VCS позволяет хранить несколько

версий одного документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда вносил изменения, и многое другое.

Главное преимущество Git — децентрализованность: каждый разработчик имеет полную копию репозитория и может работать автономно.

DevOps-инженеры ценят Git за:

- ветвление и слияние (branching/merging),
- интеграцию с CI/CD (GitHub Actions, GitLab CI),
- поддержку разных workflows (GitFlow, Trunk-Based Development),
- платформы-надстройки (GitHub, GitLab, Bitbucket).

Git — основа современной разработки и DevOps, без него невозможно представить процессы совместной работы и автоматизированные пайплайны.

Отличительные особенности программы:

- Курс ориентирован на практическое освоение Git с нуля и охватывает как базовые, так и продвинутые приёмы работы с системой контроля версий.
- Материал структурирован с акцентом на реальные задачи, с которыми сталкиваются специалисты в инженерной, DevOps- и разработческой среде.
- Особое внимание уделяется лучшим практикам командной работы, устраниению конфликтов, организации ветвлений и документированию проектов.
- Упор на подготовку к работе в команде с использованием GitHub/GitLab: fork, pull request, code review.
- Разбираются модели рабочих процессов (workflow) и сценарии их применения в реальных проектах.

Объем и срок освоения программы: 20 академ. ч. в течение 2 недель (14 дней).

Доступ к материалам Программы у обучающихся остаётся и после окончания периода обучения. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

Выдаваемый документ о квалификации: удостоверение о повышении квалификации и/или сертификат об успешном освоении программы.

Цели и задачи программы:

Цели программы - сформировать у слушателей целостное представление о принципах работы с системой контроля версий Git, научить эффективно использовать Git при индивидуальной и командной разработке программного обеспечения и управлении инфраструктурными проектами.

Программа направлена на решение следующих основных задач:

Обучающие:

- Освоить основные команды Git и логику работы с локальными и удалёнными репозиториями.
- Научиться управлять историей изменений, ветвлением, слиянием и разрешением конфликтов.
- Понять механизмы stash, rebase, cherry-pick и других команд для продвинутой работы.
- Изучить типовые модели git-workflow и научиться применять их в рабочих проектах.
- Сформировать навыки подготовки репозитория к публикации: README, лицензии, структура проекта.

Развивающие:

- Формирование логического и системного мышления при работе с инструментами контроля версий.
- Развитие навыков анализа, предвидения последствий технических решений и выбора оптимальных сценариев взаимодействия с историей проекта.
- Развитие цифровой грамотности и самостоятельности в освоении новых инструментов.

Воспитательные:

- Формирование ответственного отношения к командной работе и совместному результату.
- Воспитание аккуратности, прозрачности и честности при ведении проектной документации и фиксации изменений.

Планируемые результаты освоения программы:

Обучающийся, успешно освоивший программу повышения квалификации, сможет демонстрировать следующие знания, умения и навыки:

Знания:

- Структура и принципы работы системы контроля версий Git.
- Основные команды и режимы работы Git: commit, merge, branch, checkout и др.
- Модели организации git-процессов: Git Flow, GitHub Flow, trunk-based development.
- Принципы ведения документации в репозиториях (README, CONTRIBUTING и др.).
- Best Practices для командной работы и CI/CD.

Умения:

- Настраивать и использовать локальные и удалённые репозитории.
- Выполнять базовые и продвинутые операции с коммитами и ветками.
- Использовать инструменты для разрешения конфликтов и управления историей.
- Работать в команде через fork, pull request и code review.
- Писать структурированные, понятные README-файлы и вести базовую проектную документацию.

Навыки:

- Уверенная работа с Git в командной и индивидуальной разработке.
- Решение типовых задач по откату, объединению, переносу и анализу изменений.
- Применение git-репозиториев в контексте DevOps, тестирования, системного администрирования и релиз-менеджмента.
- Настройка удобной среды для работы с Git на своём рабочем месте.

Перечень профессиональных компетенций, на получение которых направлено обучение:

На основе профстандарта «Системный администратор информационно-коммуникационных систем»:

- D/01.6 Выполнение работ по выявлению и устранению нетипичных инцидентов, возникающих в серверных операционных системах информационно-коммуникационной системы

На основе профстандарта «Программист»:

- A/04.3 Работа с системой управления версиями программного кода
- C/02.5 Осуществление интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- ОПК-2 – Способность «понимать принципы работы современных информационных технологий и программных средств, в т.ч. отечественного производства, и

использовать их при решении задач проф. деятельности» (форма применения: освоение Git как современного инструмента управления версиями).

- ОПК-3 – «Решать стандартные задачи... с учетом основных требований информационной безопасности» (форма применения: безопасное ведение веток, Umgang с секретными данными через .gitignore).
- ОПК-4 – «Участвовать в разработке стандартов... и технической документации» (форма применения: оформление README, документации к репозиторию).
- ОПК-8 – «Разрабатывать алгоритмы и программы, пригодные для практического применения» (форма применения: применение Git для практических сценариев разработки и поддержки ПО).
- ОПК-9 – «Осваивать методики использования программных средств для решения практических задач» (форма применения: освоение Git как инструментального навыка).

Организационно-педагогические условия реализации программы дополнительного профессионального образования

Язык реализации образовательной программы: обучение проводится на русском языке.
Форма обучения: заочная форма.

Особенности реализации программы: программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

Условия набора: на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

Формы проведения занятий:

- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа с литературой;
- индивидуальные вопросы.

Материально-техническое оснащение

Материальное обеспечение программы

Занятия проводятся в системе дистанционного обучения «Rebrain». Каждый обучающийся и педагог оснащены доступом к системе дистанционного обучения: <https://rebrainme.com/>.

У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

Методическое обеспечение программы

Программа обеспечена:

- учебно-методическими материалами (текстовые занятия, полезными материалами);
- практическими заданиями.

Кадровое обеспечение:

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

- 1) отвечающее одному из требований:
 - а) имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования «Образование и педагогические науки»;
 - б) имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Ребреин», и получение при необходимости дополнительного профессионального образования педагогической направленности;
 - в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности дополнительной общеобразовательной общеразвивающей программе;
 - 2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;
 - 3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

2. УЧЕБНЫЙ ПЛАН

№ п/п	Наименование модуля	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1	Модуль 1 “Онбординг”	1	0,5	0,5	Входное тестирование
2	Модуль 2 “Git”	18	3	15	Тестирование
3	Итоговая аттестация	1		1	Тест с автопроверкой

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 неделя	2 неделя
1	Модуль 1 “Онбординг”	1	
2	Модуль 2 “Git”	9	9
3	Итоговая аттестация		1 А

4. РАБОЧАЯ ПРОГРАММА

Модуль 1. Онбординг

Теория 0,5 академ. ч. Практика 0,5 академ. ч.

Модуль состоит из следующих блоков изучения тем:

Тема 1. Онбординг

В теме обучающемуся предоставляется вводный конспект, содержащий общую информацию о программе, структуре курса, форматах взаимодействия с материалами и ожидаемых результатах обучения.

Предусмотрено прохождение входного тестирования, включающего 6 вопросов, направленных на закрепление информации из онбординга. В рамках блока обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы, указывает желаемые навыки по окончании обучения.

Модуль 2. Git

Теория 3 академ. ч. Практика 15 академ. ч.

Модуль состоит из следующих тем:

Тема 1: Основы работы с Git

Содержание: типы систем контроля версий. Установка и настройка Git. Создание репозитория и добавление файлов.

Тема 2: Игнорирование и удаление файлов в репозитории

Содержание: Файл .gitignore. Файл exclude. Пустые директории в Git. Удаление файлов из репозитория. Переименование файлов.

Тема 3: История изменений

Содержание: просмотр истории. Авторы и Коммитеры. Сравнение изменений. Переключение между коммитами.

Тема 4: Изменение и отмена коммитов

Содержание: Отмена коммитов с помощью revert (безопасная). Отмена нескольких коммитов. Отмена коммитов с помощью reset (опасная). Изменение последнего коммита.

Тема 5: Работа с ветками и тегами

Содержание: Ветки. Создание и удаление веток. Теги.

Тема 6: Слияние веток

Содержание: Истинное Слияние (True Merge). Слияние методом быстрой перемотки (fast-forward merge).

Тема 7: Решение конфликтов при слиянии веток

Содержание: способы решения конфликтов. Графическое решение конфликтов.

Тема 8: Слияние веток через «rebase»

Содержание: Перебазирование коммитов. Конфликты и отмена rebase.

Тема 9: Объединение коммитов

Содержание: Объединение коммитов с помощью «rebase». Объединение коммитов с помощью «reset». Объединение коммитов при слиянии через «merge».

Тема 10: Перенос коммитов из одной ветки в другую

Содержание: Перенос одного коммита. Перенос нескольких коммитов.

Тема 11: Работа с «git stash»

Содержание: Откладываем одно изменение. Работа с несколькими отложенными изменениями. Добавление сообщения в отложенное изменение. Применение отложенных изменений. Очистка отложенных изменений. Поиск потерянных изменений.

Тема 12: Работа с удалёнными репозиториями

Содержание: Синхронизация репозиториев в Git. Краткий обзор популярных git-хостингов. GitLab. Отправка репозитория на git-сервер. Обновление репозитория с git-сервера. Клонирование репозитория с git-сервера.

Тема 13: Работа с командным репозиторием (code review), fork

Содержание: Code review. Fork репозитория. Contribution rules. Issues.

Тема 14: Рабочие процессы: GitHub Flow, GitFlow

Содержание: Git Flow. GitHub Flow. GitLab Flow.

Тема 15: Best practices и README-файл

Содержание: README.md-файл и Markdown-формат. Как написать хороший README. Git Best Practices.

Тема 16: Подмодули (submodules)

Содержание: Краткая справка о submodules. Подключение submodules. Клонирование репозитория с сабмодулем. Обновление сабмодулей.

Итоговая аттестация

Блок посвящён выполнению финального практического задания без предварительного теоретического блока.

5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Программа обеспечена системой дистанционного обучения <https://rebrainme.com/>.

Педагогические технологии:

- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

Дидактический материал:

1. Git: распространённые ошибки и способы их исправления [Электронный ресурс]: [Habr.com](https://habr.com/ru/companies/ruvds/articles/423015/). Доступ по ссылке: <https://habr.com/ru/companies/ruvds/articles/423015/>
2. Git happens! 6 типичных ошибок Git и как их исправить [Электронный ресурс]: [Habr.com](https://habr.com/ru/companies/flant/articles/419733/). Доступ по ссылке: <https://habr.com/ru/companies/flant/articles/419733/>

Электронно-библиотечные ресурсы и системы, информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.RU.
2. Open source maintained by members of the Git community <https://git-scm.com/>
3. Собственные учебные материалы: <https://my.rebrainme.com/course/git>

6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценочные материалы:

Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль, промежуточный контроль и итоговое оценивание.

Текущий контроль

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося в СДО. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходит в автоматическим решиме в СДО.

Модуль направлен на формирование у обучающихся базовых и продвинутых навыков работы с системой контроля версий Git. Каждый блок модуля включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание.

Пример тестовых заданий:

1. Как предпочтительнее организовать структуру разделов в [readme.md](#)?
 - А. Создать один большой раздел под названием “Важная информация”
 - Б. Разделить информацию на блоки и использовать заголовки**
 - С. Использовать только список ключевых правил
2. Как правильно использовать комментарии при git commit?
 - А. Не указывать подробности о сделанных изменениях.
 - Б. Использовать краткие, но информативные описания изменений.**
 - С. Включать в комментарии всю историю проекта.
3. Что необходимо добавить в README.md для вставки изображения из репозитория?
 - А. ![Описание картинки](путь_к_картинке/имя_файла.png)**
 - Б. !(Описание картинки)[путь_к_картинке/имя_файла.png]
 - С. [Описание картинки](путь_к_картинке/имя_файла.png)
 - Д. (Описание картинки)[путь_к_картинке/имя_файла.png]

Итоговое оценивание

Практика является основой для формирования и закрепления навыков, поэтому рекомендуется регулярно возвращаться к Git и самостоятельно тренировать применение инструментов системы контроля версий.

Данный блок включает итоговое тестирование, предназначенное для проверки усвоения ключевых аспектов работы с Git. Результаты теста помогут определить темы, требующие повторного изучения или дополнительного внимания.

Тест состоит из 10 вопросов с выбором одного или нескольких вариантов ответа. Время выполнения не ограничено. Количество попыток прохождения не ограничено.

Примеры теоретических вопросов:

1. Вам нужно сделать настройку, чтобы в репозиторий никогда не попадали файлы с логами. В каком файле нужно описать это правило?

- A. .gitignore
- B. .git/info/gitignore
- C. .gitexclude
- D. .git/info/exclude

2. В своём проекте вы отделились из master-ветки в отдельную ветку, в которой продолжили свою работу, сделали много изменений и очень частые коммиты. В master-ветке при этом появлялись новые коммиты. Какая стратегия слияния с master-веткой позволит оставить историю более чистой?

- A. Rebase
- B. Squash
- C. True Merge
- D. Fas-forward merge