|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**  **РОССИЙСКОЙ ФЕДЕРАЦИИ**  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«ЮЖНО-РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ  УНИВЕРСИТЕТ (НПИ) имени М.И. ПЛАТОВА»**  Факультет Информационных технологий и управления  Кафедра «Информационные и измерительные системы и технологии»  Направление подготовки 09.03.02 Информационные системы и технологии  **ОТЧЁТ**  **об**  учебной **практике**  наименование, вид практики  **на** кафедре ИИСТ ЮРГПУ (НПИ)  название организации, профильной организации  **студента** 2 курса, группы 090302-ИCТа-018  курса, группы    фамилия, имя, отчество  **Руководитель практики Оценка**   **М.П.**  **от профильной организации**      должность, звание, степень подпись, дата    (Ф. И. О.)  **Руководитель практики Оценка**  **от кафедры** ИИСТ    к.т.н. доцент кафедры ИИСТ  должность, звание, степень подпись, дата  Хорошко М.Б.  (Ф. И. О.)  **г. Новочеркасск, 2020 г.** |

**Оглавление:**

[Введение 3](#_Toc47012763)

[Изучение функционала GitHub 4](#_Toc47012764)

[Справочная информация. 4](#_Toc47012765)

[Ключ к пониманию концепции git — знание о «трех деревьях»: 4](#_Toc47012766)

[Указатели: 4](#_Toc47012767)

[Основные команды Git: 5](#_Toc47012768)

[Ход работы: 6](#_Toc47012769)

[Изучение поисковых систем 20](#_Toc47012770)

[Типы поисковиков: 20](#_Toc47012771)

[Примеры поисковых систем: 21](#_Toc47012772)

[Google 21](#_Toc47012773)

[Яндекс 22](#_Toc47012774)

[Техническое задание 24](#_Toc47012775)

[Общие сведения 24](#_Toc47012776)

[Назначение и цель создания сайта 25](#_Toc47012777)

[Краткая информация 25](#_Toc47012778)

[Цель создания сайта 25](#_Toc47012779)

[Назначение сайта 25](#_Toc47012780)

[Целевая аудитория 26](#_Toc47012781)

[Мотивация использования сайта 26](#_Toc47012782)

[Сценарий пользования сайтом 27](#_Toc47012783)

[Разработка прототипа проекта 28](#_Toc47012784)

[Прототипирование 28](#_Toc47012785)

[Заключение 32](#_Toc47012786)

# Введение

*Цель учебной практики* – ознакомление с крупнейшим веб-сервисом для хостинга IT-проектов и их совместной разработки «GitHub», а также определиться с тематикой работы в IT-сфере.

После выбора направления изучить аналоги у конкурентов, составить таблицу преимуществ и недостатков их продукта. На следующем шаге следует разработать правильное техническое задание для нашей системы, которое сэкономит время и средства.

После занимаемся изучением среды разработки и разработкой самого прототипа. Описываем его преимущества и недостатки и для чего служит протипирование.

# ИЗУЧЕНИЕ ФУНКЦИОНАЛА GITHUB

**Справочная информация:** Git — это набор консольных утилит, которые отслеживают и фиксируют изменения в файлах (чаще всего речь идет об исходном коде программ, но вы можете использовать его для любых файлов на ваш вкус). С его помощью вы можете откатиться на более старую версию вашего проекта, сравнивать, анализировать, сливать изменения и многое другое. Этот процесс называется контролем версий. Существуют различные системы для контроля версий. Вы, возможно, о них слышали: SVN, Mercurial, Perforce, CVS, Bitkeeper и другие.

Git является распределенным, то есть не зависит от одного центрального сервера, на котором хранятся файлы. Вместо этого он работает полностью локально, сохраняя данные в папках на жестком диске, которые называются репозиторием. Тем не менее, вы можете хранить копию репозитория онлайн, это сильно облегчает работу над одним проектом для нескольких людей. Для этого используются сайты вроде github и bitbucket.

**Ключ к пониманию концепции git — знание о «трех деревьях»:**

* **Рабочая директория** — файловая система проекта (те файлы, с которыми вы работаете).
* **Индекс** — список отслеживаемых git-ом файлов и директорий, промежуточное хранилище изменений (редактирование, удаление отслеживаемых файлов).
* **Директория .git/** — все данные контроля версий этого проекта (вся история разработки: коммиты, ветки, теги и пр.).
* **Коммит** — «сохранение» (хранит набор изменений, сделанный в рабочей директории с момента предыдущего коммита). Коммит неизменен, его нельзя отредактировать.У всех коммитов (кроме самого первого) есть один или более родительских коммитов, поскольку коммиты хранят изменения от предыдущих состояний.

**Указатели:**

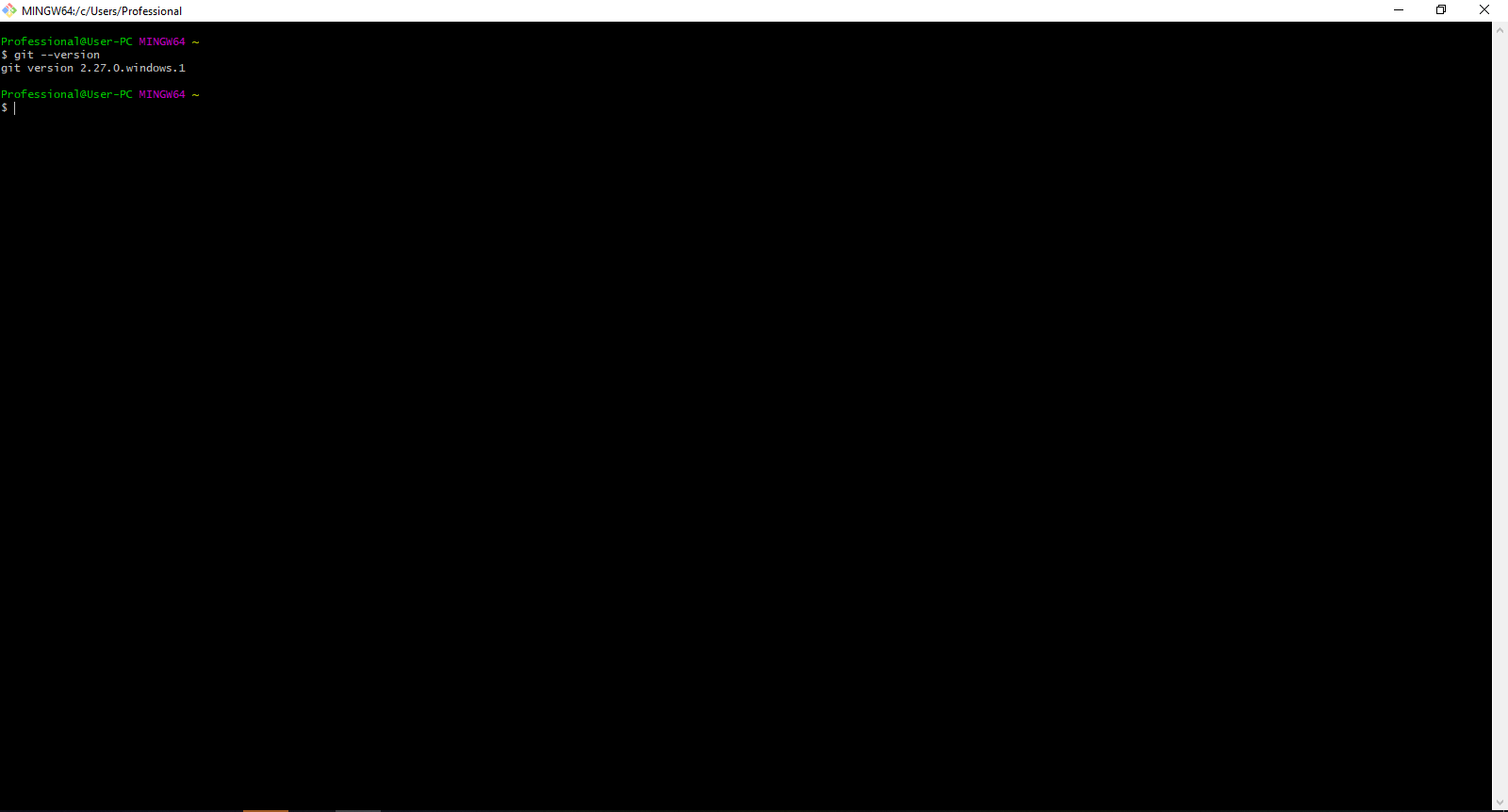
* **HEAD** — указатель на текущий коммит или на текущую ветку (то есть, в любом случае, на коммит). Указывает на родителя коммита, который будет создан следующим.
* **ORIG\_HEAD** — указатель на коммит, с которого вы только что переместили HEAD (командой git reset ..., например).
* **Ветка** (master, develop etc.) — указатель на коммит. При добавлении коммита, указатель ветки перемещается с родительского коммита на новый.
* **Теги** — простые указатели на коммиты. Не перемещаются.

**Основные команды Git:**

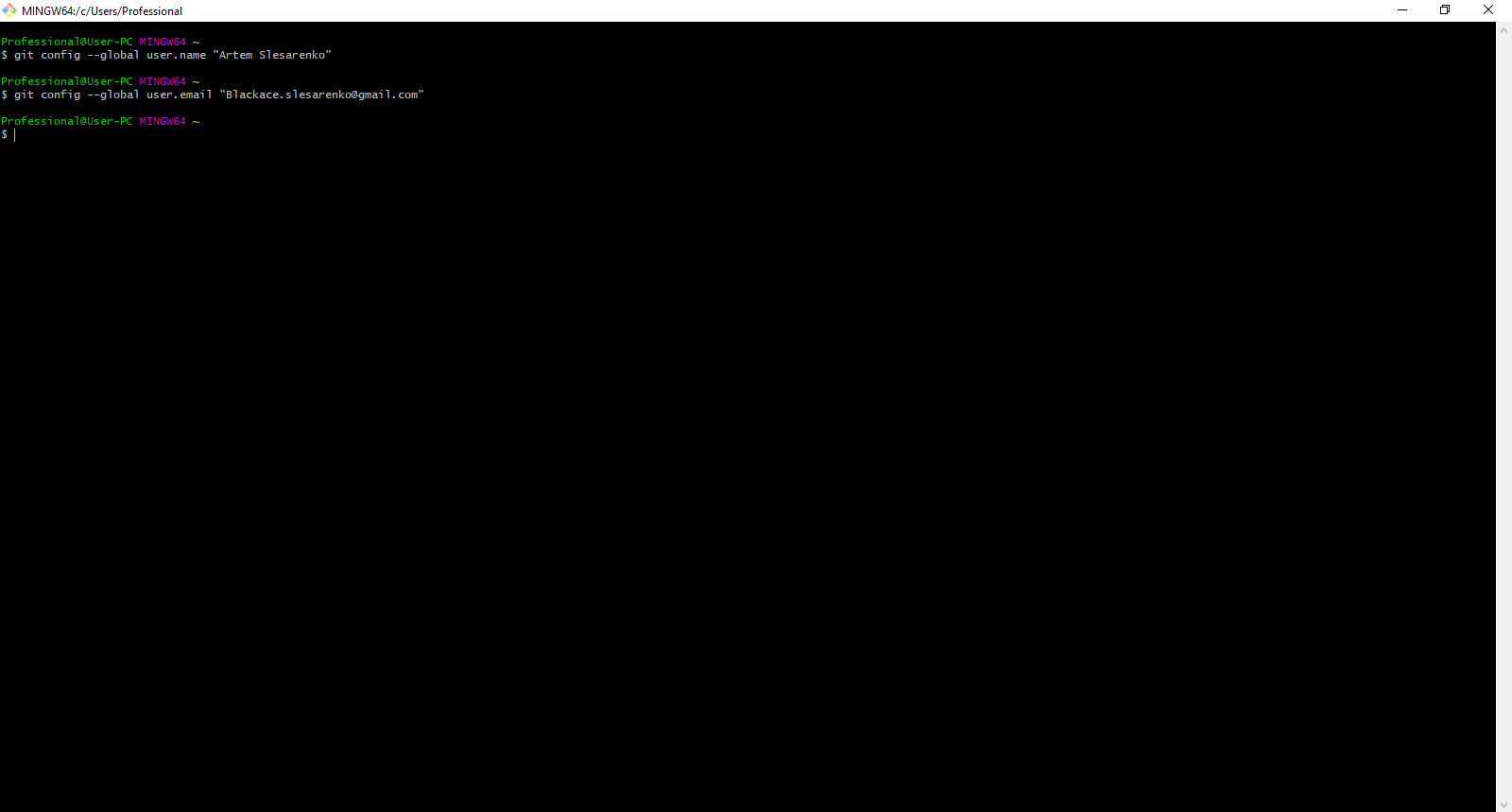
* VIM-команды:
  1. Нажатия кнопок
     1. ESC — переход в командный режим
     2. i — переход в режим редактирования текста
     3. ZQ (зажат Shift, поочередное нажатие) — выход без сохранения
     4. ZZ (зажат Shift, поочередное нажатие) — сохранить и выйти
  2. Ввод в командном режиме
     1. :q! — выйти без сохранения
     2. :wq — сохранить файл и выйти
     3. :w filename.txt — сохранить файл как filename.txt
* Консольные команды

1. git init # создать новый проект в текущей директории
2. git init folder-name # создать новый проект в указанной директории
3. git status # показать состояние репозитория (отслеживаемые, изменённые, новые файлы и пр.)
4. git add . # добавить в индекс все новые, изменённые, удалённые файлы из текущей директории и её поддиректорий
5. git reset # убрать из индекса все добавленные в него изменения (в рабочей директории все изменения сохранятся), антипод git add
6. git commit -m "Name of commit" # зафиксировать в коммите проиндексированные изменения (закоммитить), добавить сообщение
7. git log master # показать коммиты в указанной ветке
8. git branch # показать список веток
9. git branch new\_branch # создать новую ветку с указанным именем на текущем коммите
10. git stash # временно сохранить незакоммиченные изменения и убрать их из рабочей директории

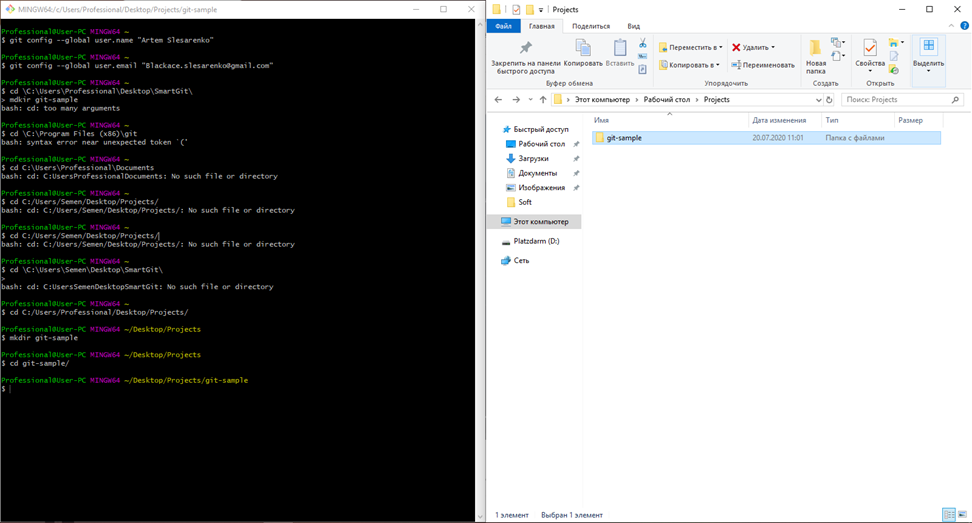
**Ход работы:**

****

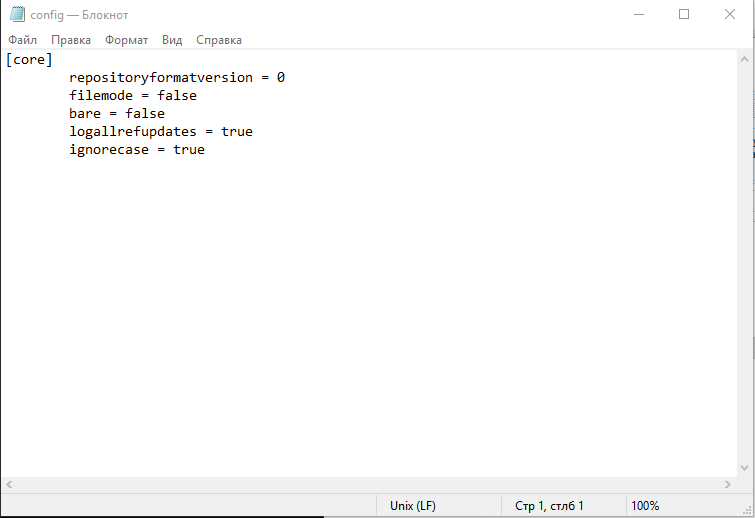
*Рис.1 - Окно программы GIT*

**

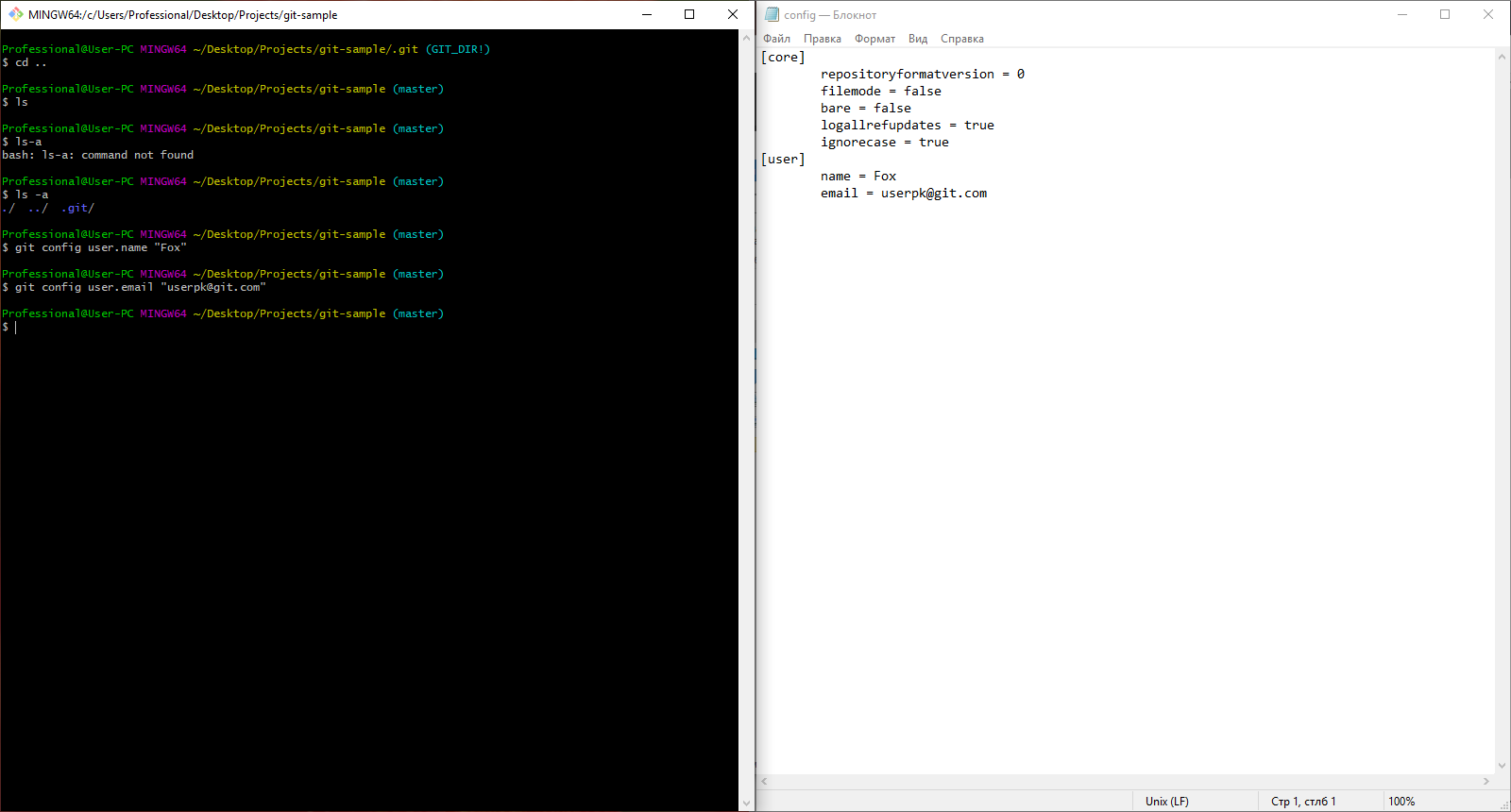
*Рис.2 – Ввод глобальных имени пользователя и электронного адреса*

**

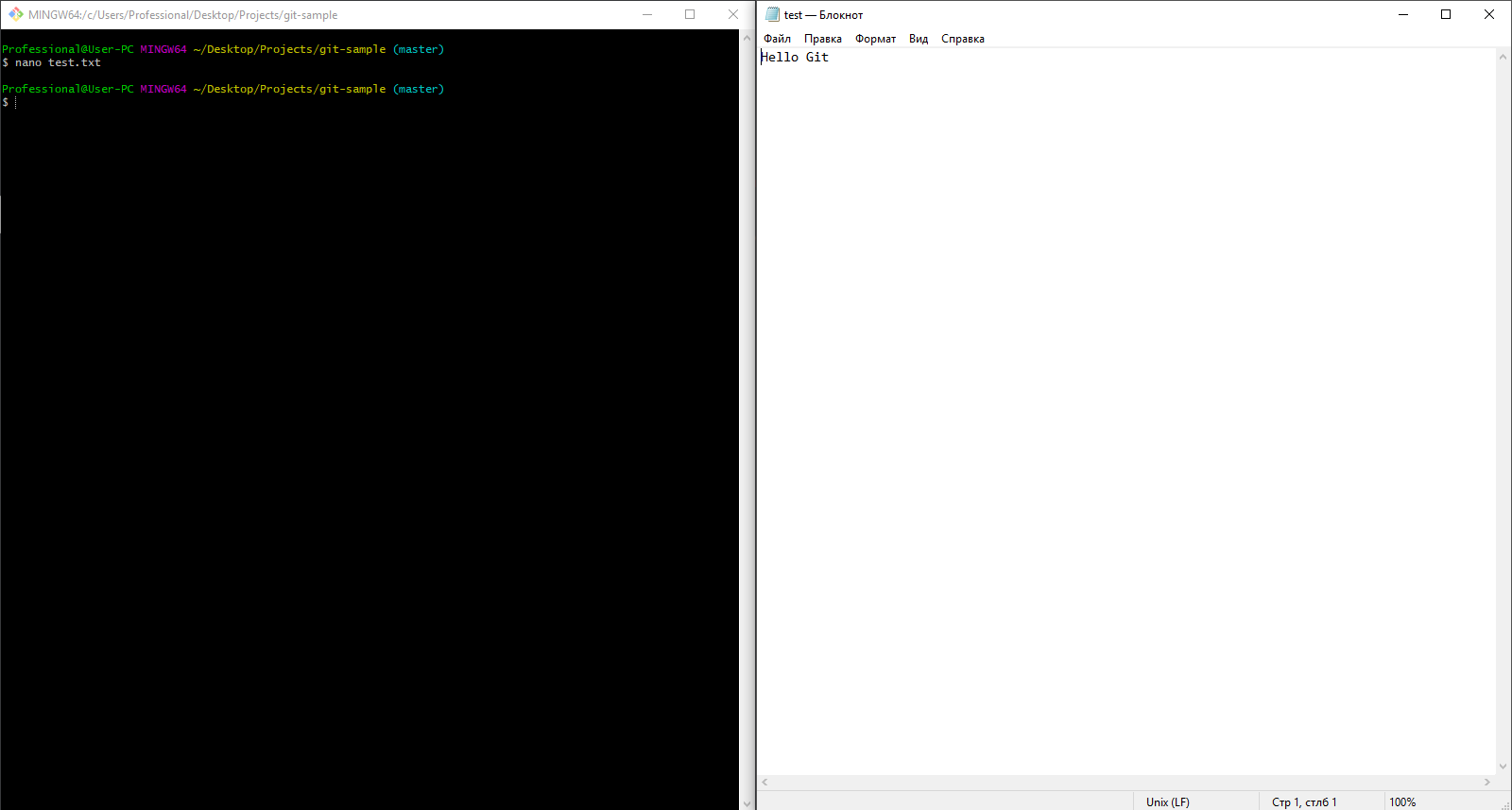
*Рис.3 -Создание рабочей директории*

**

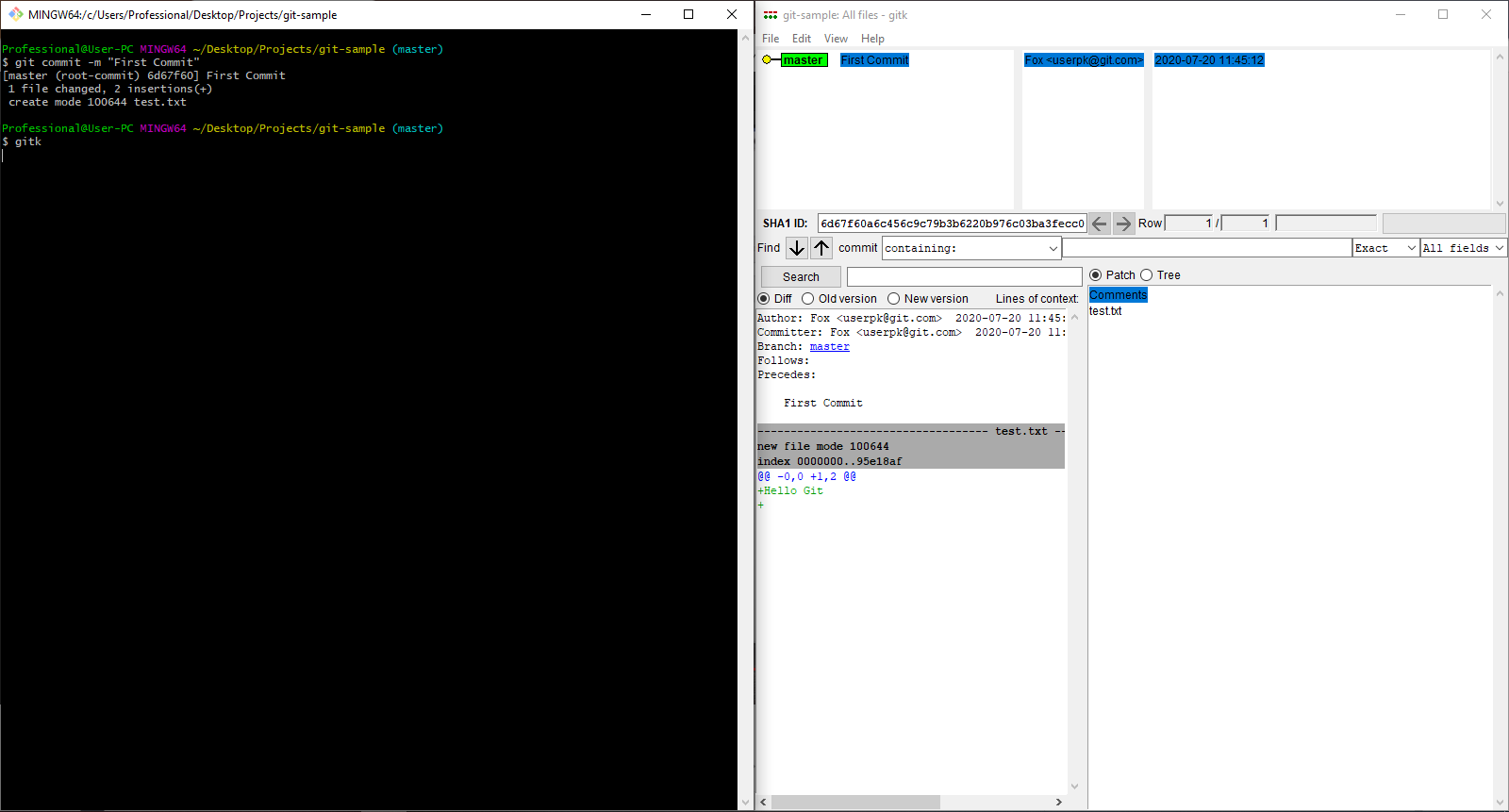
*Рис.4 – Файл Config*



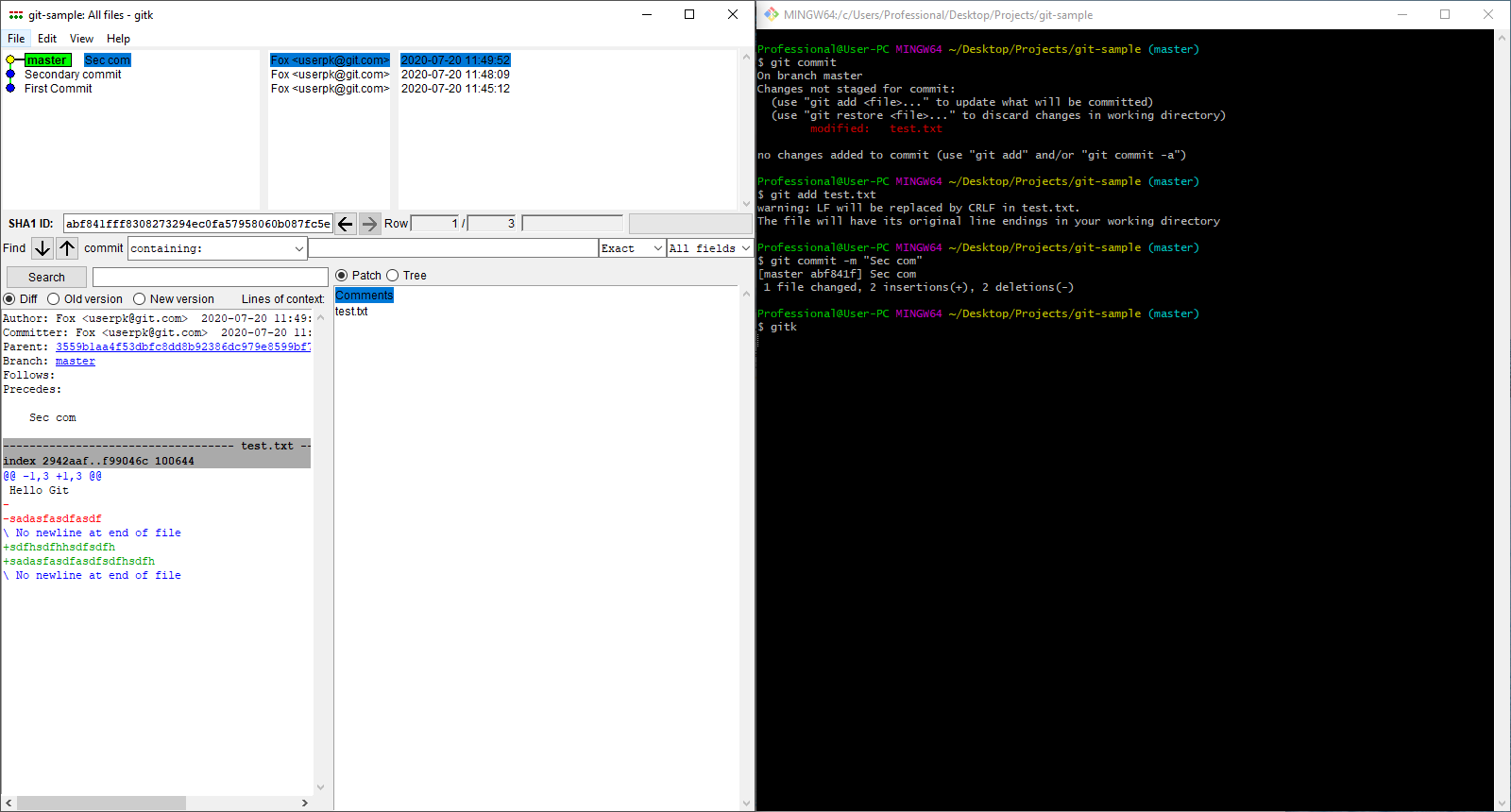
*Рис.5 – Ввод локальных имени пользователя и электронного адреса*

**

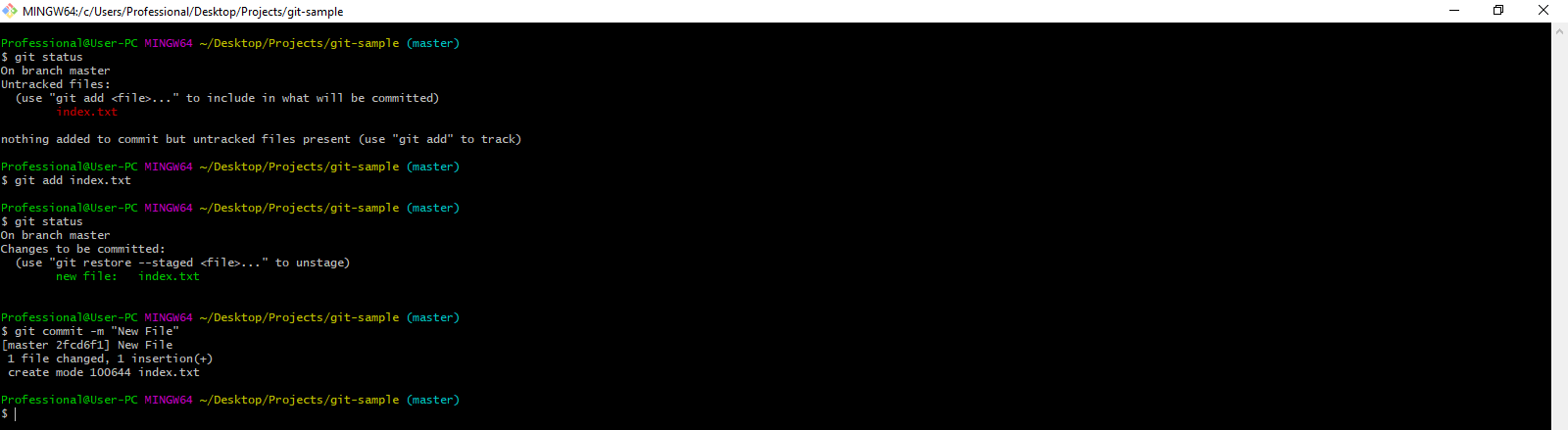
*Рис.6 – Создание текстового файла test.txt*

**

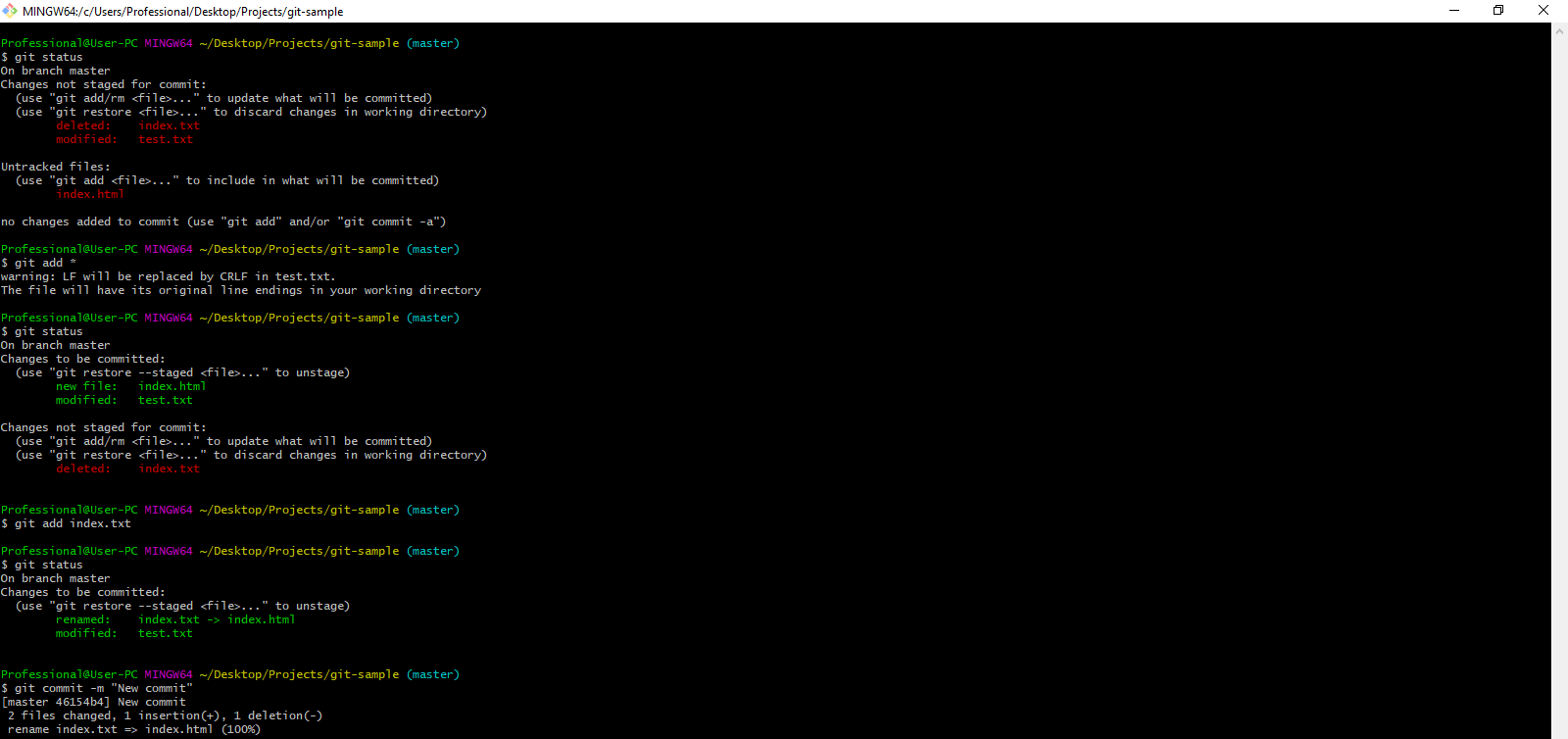
*Рис.7 – Демонстрация команды gitk*

**

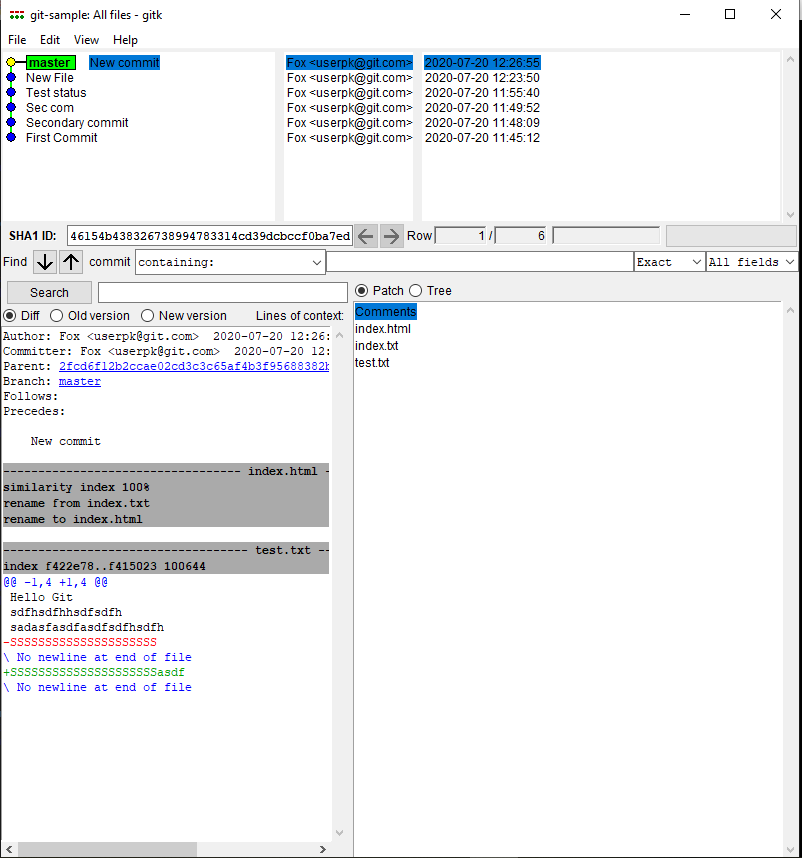
*Рис.8 – Редактирование текстового файла test.txt*

**

*Рис.9 – Создание текстового файла index.txt*

**

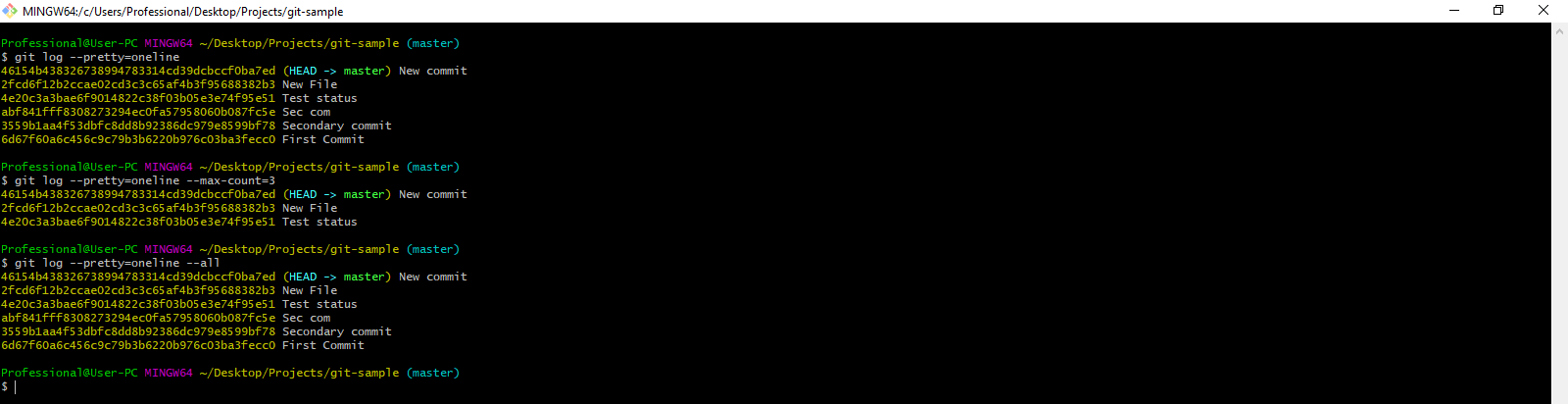
*Рис.10 – Изменение формата файла index*

**

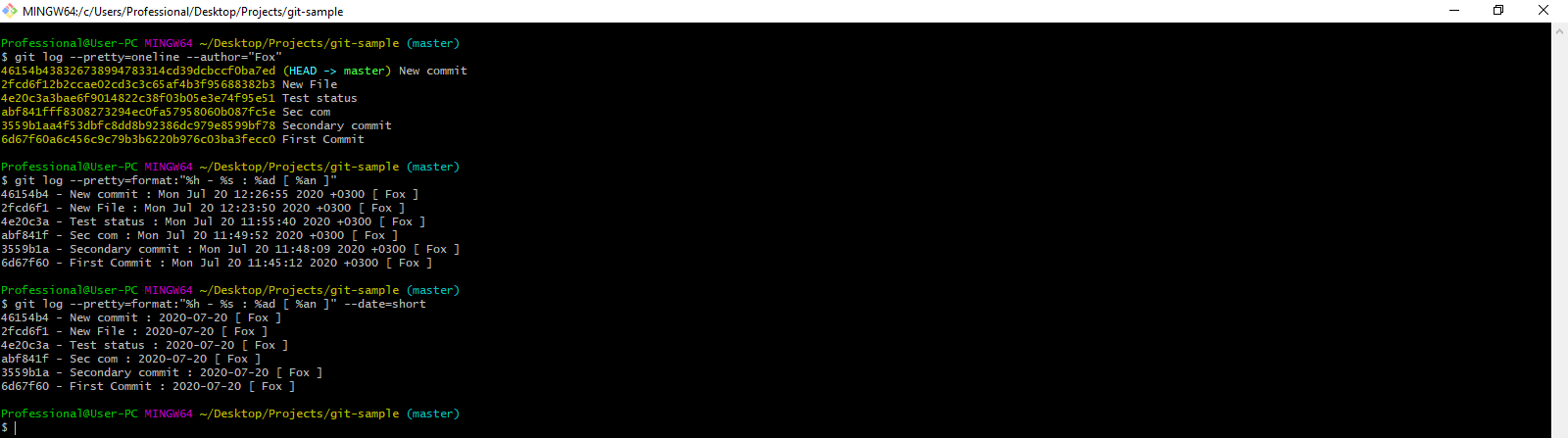
*Рис.11 – gitk*

**

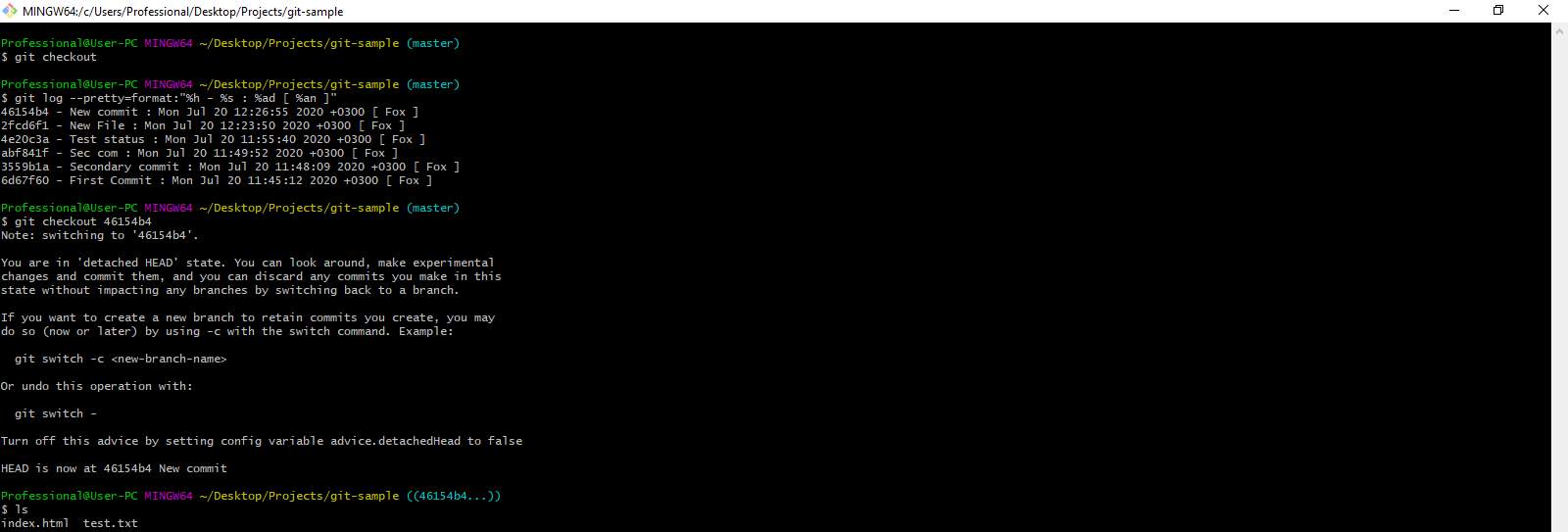
*Рис.12 – Демонстрация команды git log*

**

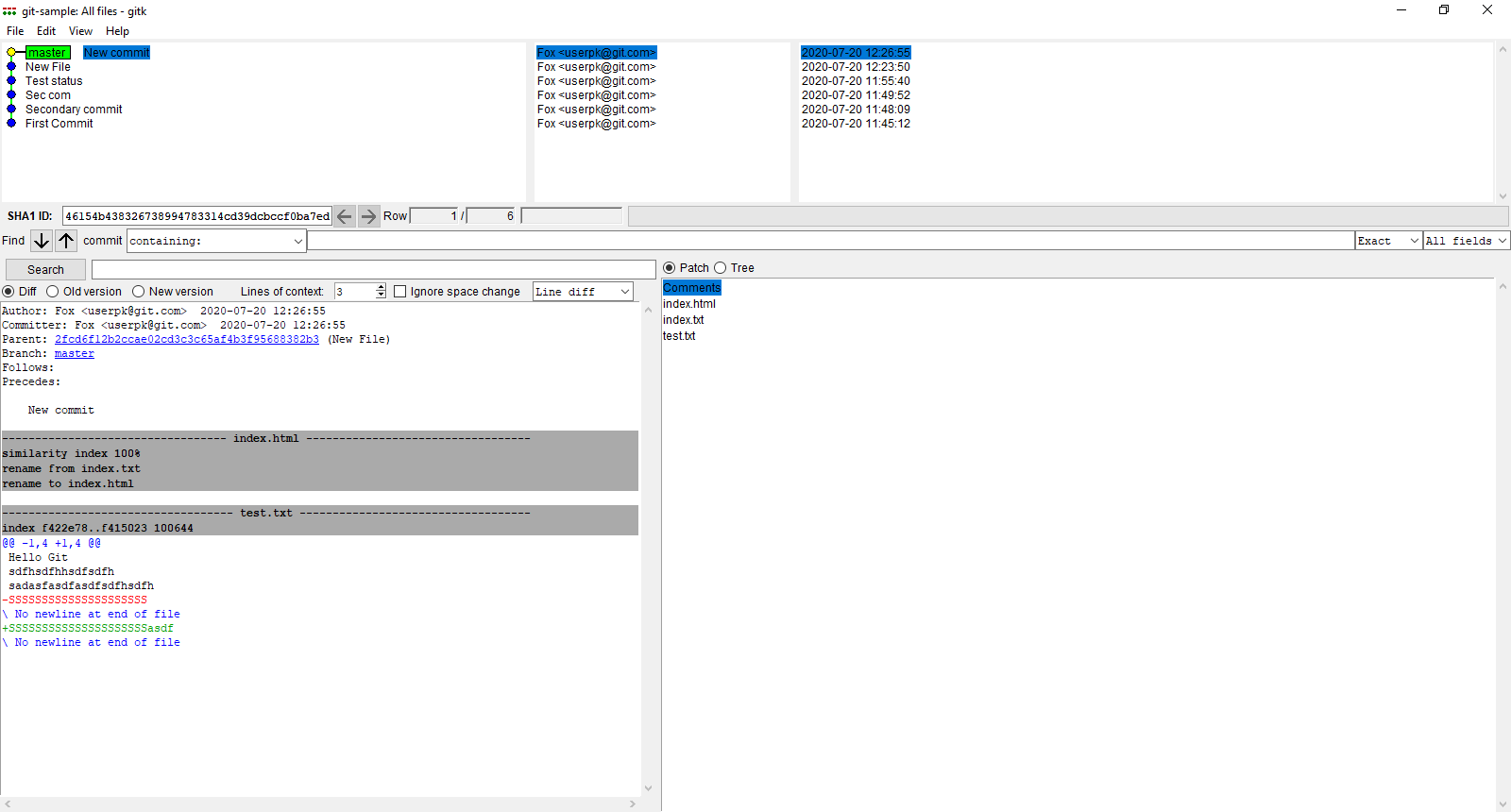
*Рис.13 – Разновидности команды git log(1)*

**

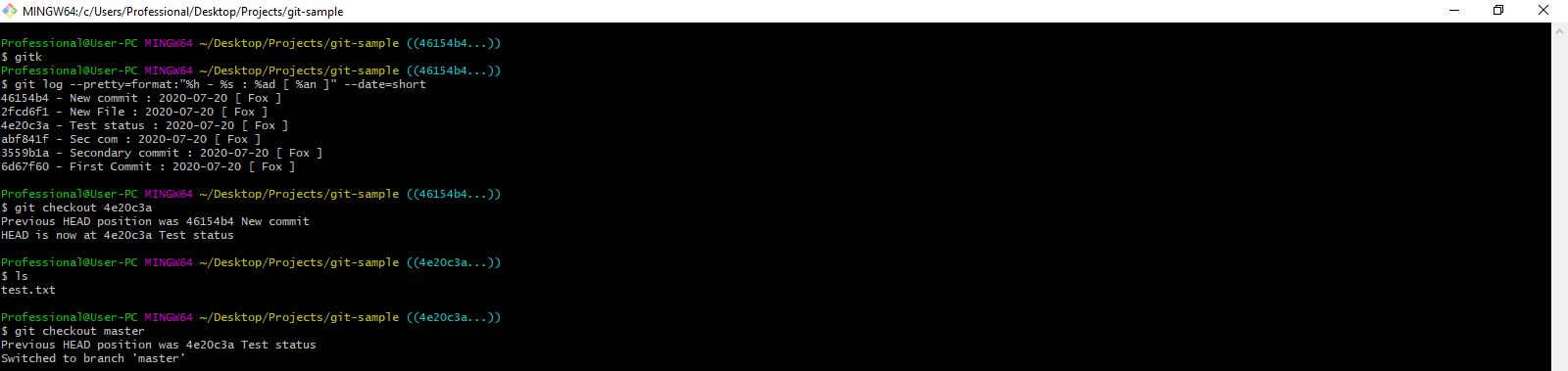
*Рис.13 – Разновидности команды git log(2)*

**

*Рис.14 – Смена ветви*

**

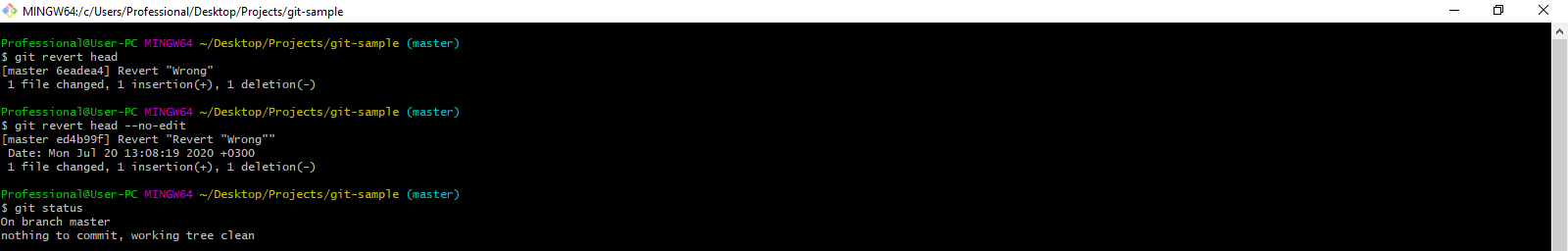
*Рис.15 – gitk*

**

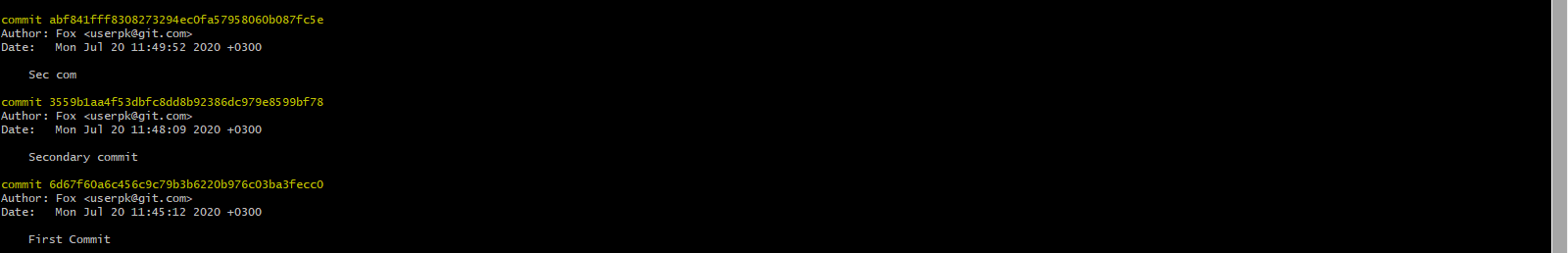
*Рис.16 – Просмотр информации о ветви*

**

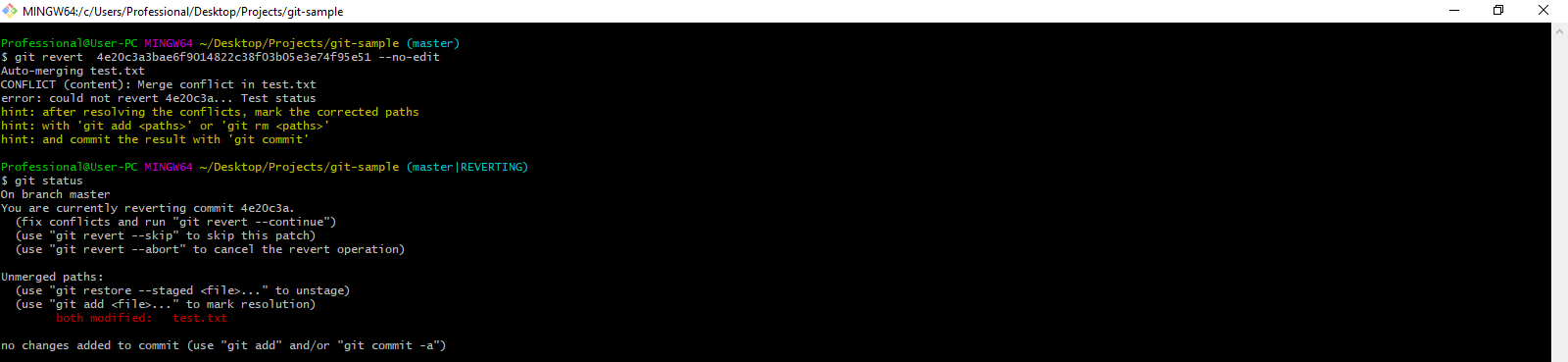
*Рис.17 – Работа с указателем HEAD*

**

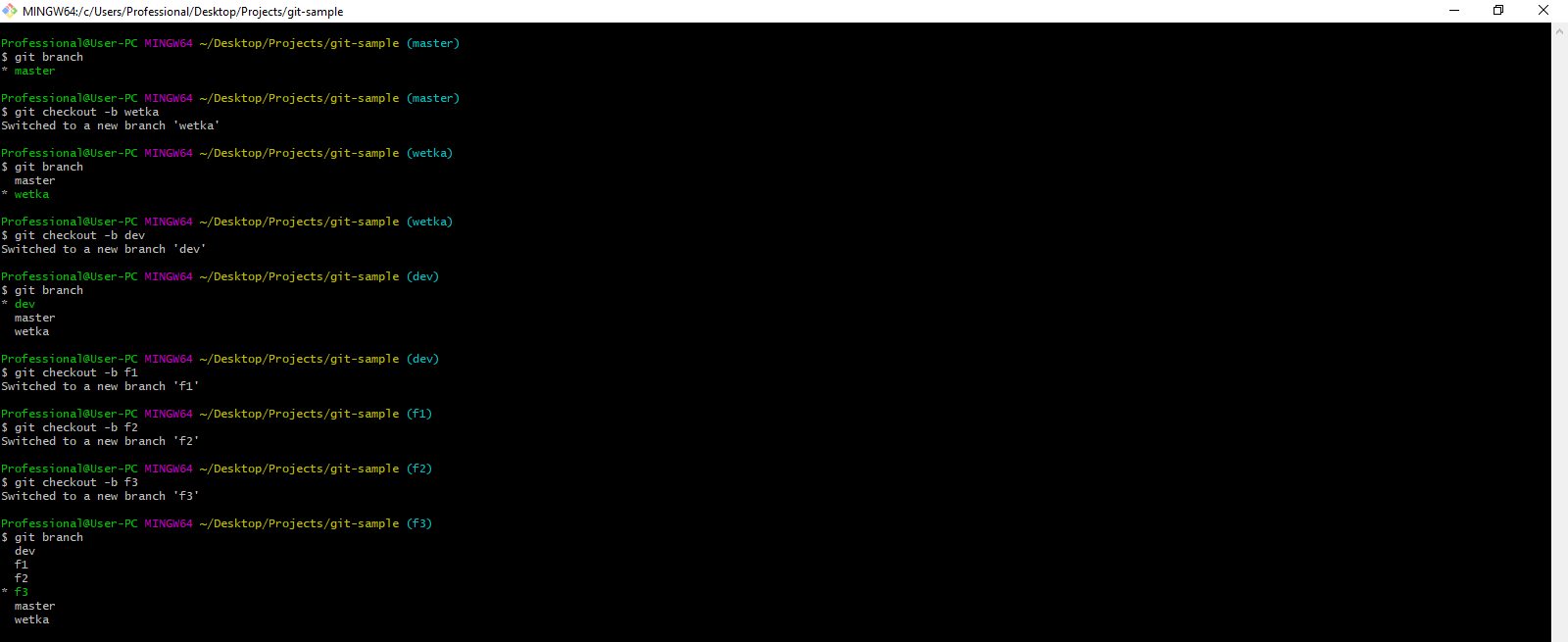
*Рис.18 – Изменение указателя HEAD*

**

*Рис.19 – git log*

**

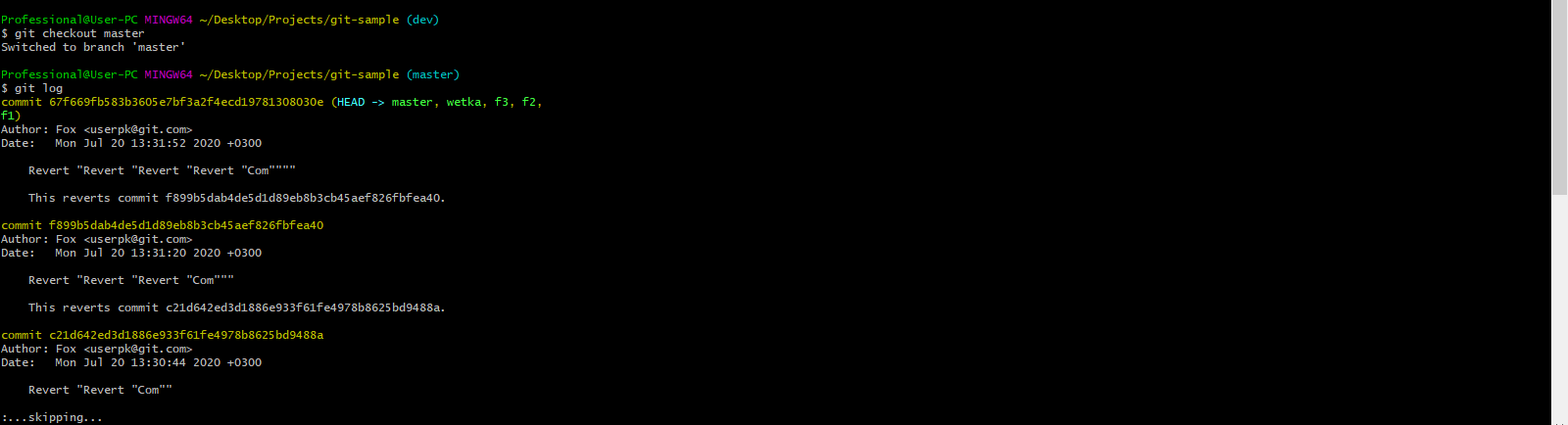
*Рис.20 – Ошибка при выполнении команды revert*

**

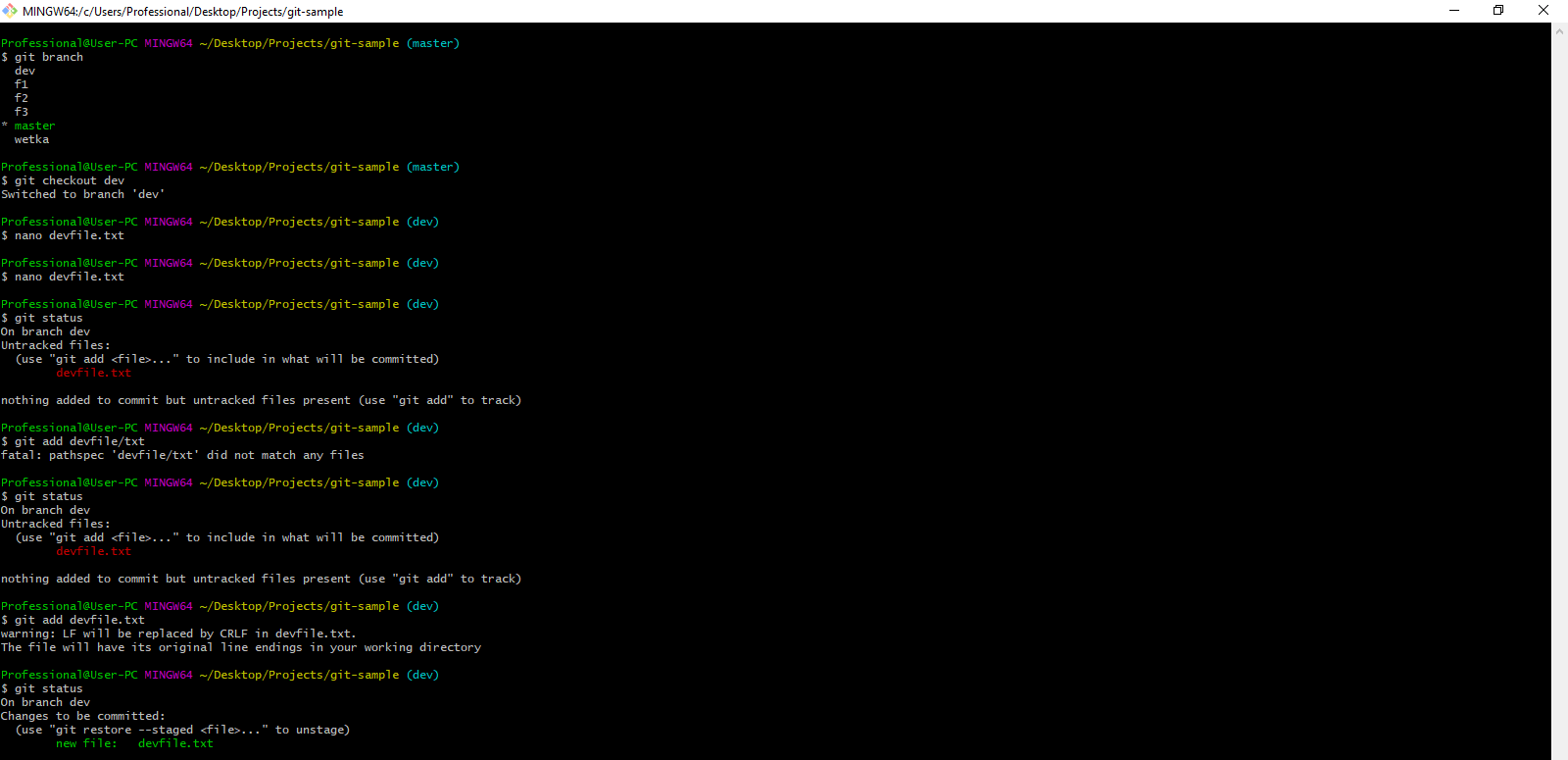
*Рис.21 – Создание ветвей*

**

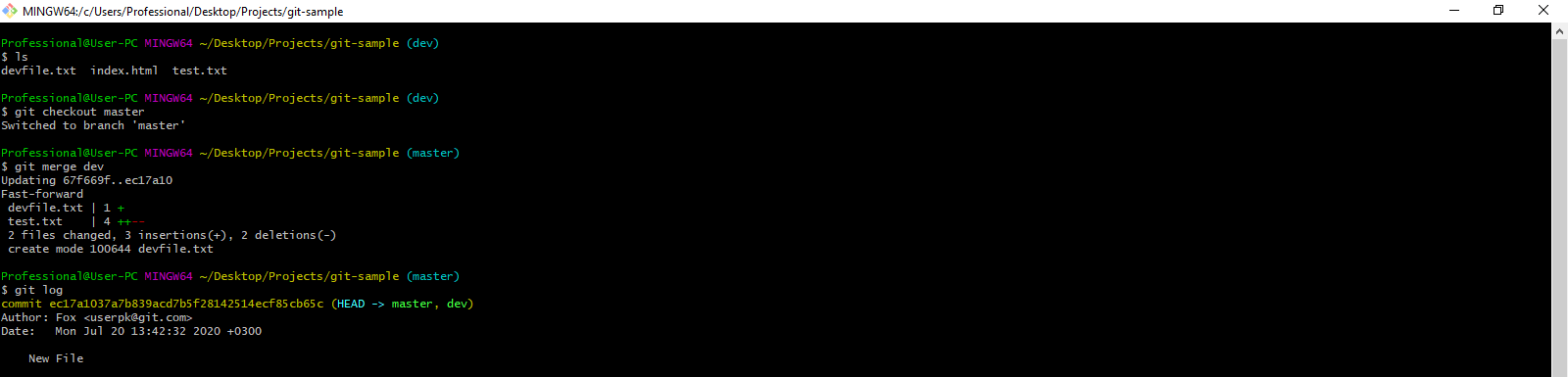
*Рис.22 – Работа с ветвями*

**

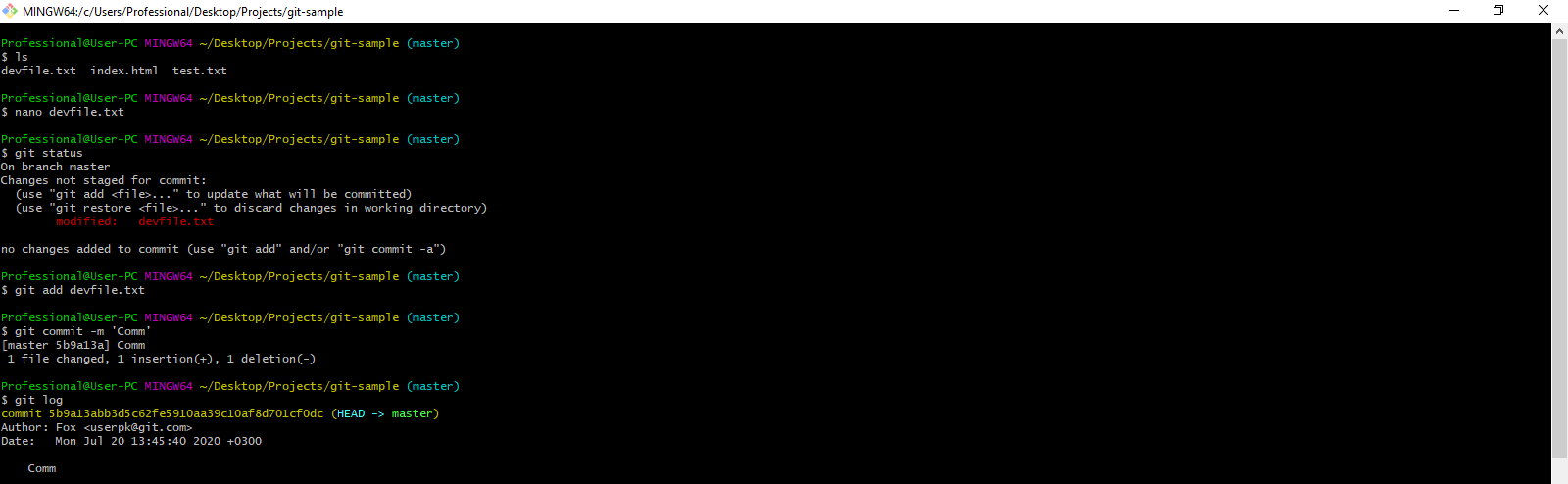
*Рис.23 – git log*

**

*Рис.24 – Создание нового текстового файла*

**

*Рис.25 – слияние ветвей*

**

*Рис.25 – Изменение файлов в соединённых ветвях*

**

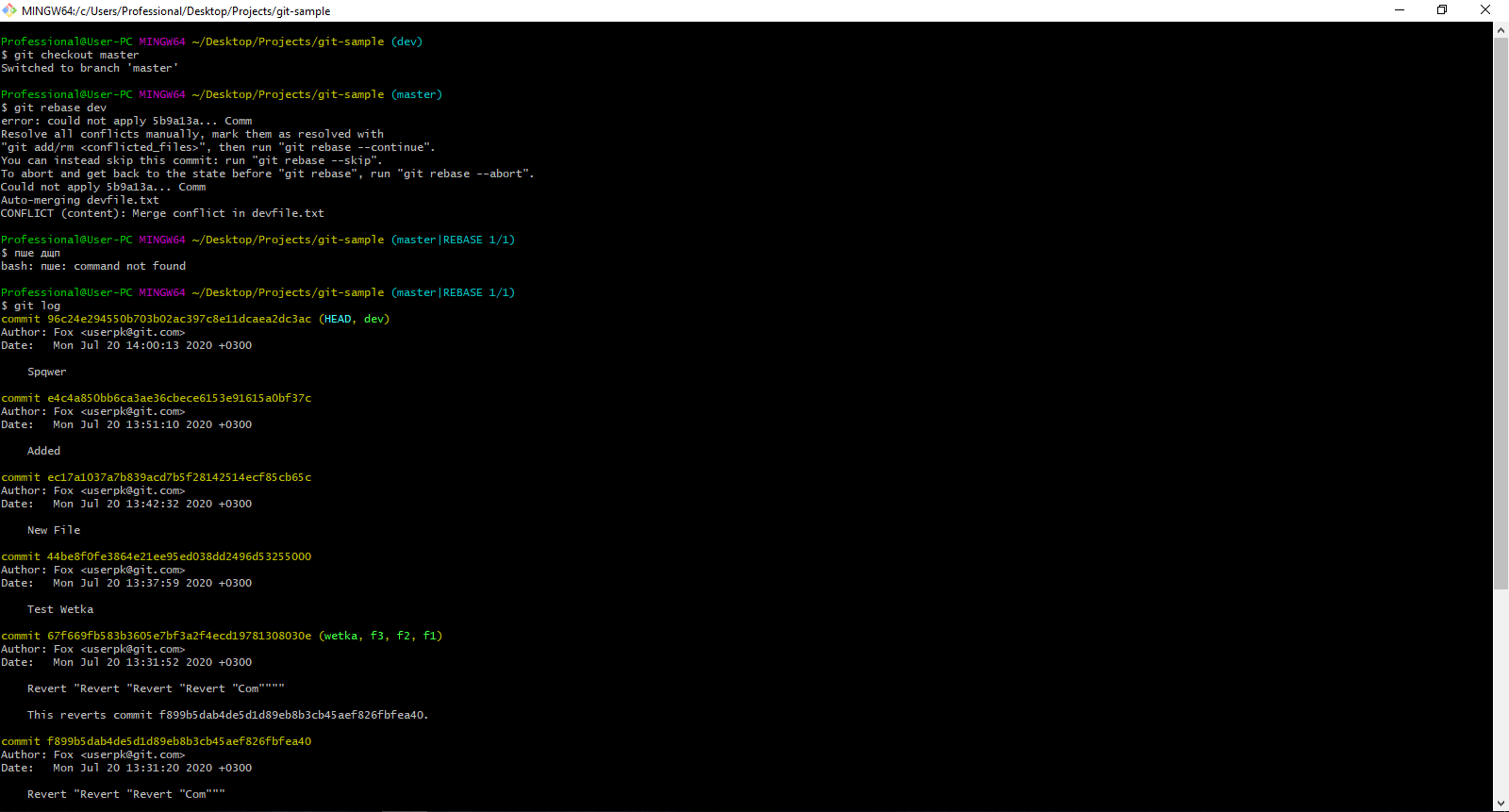
*Рис.26 – Файл devfile.txt*

**

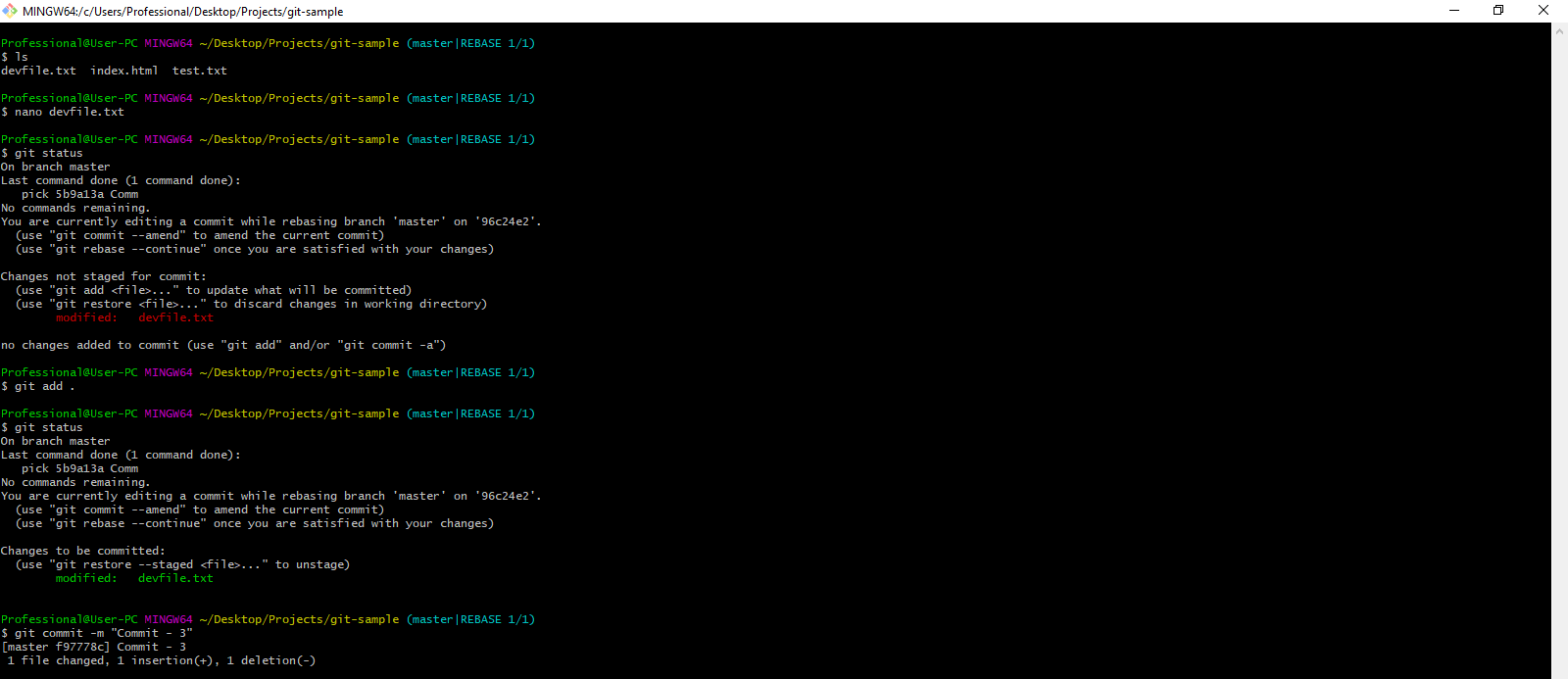
*Рис.27 – Выполнение команды merge*

**

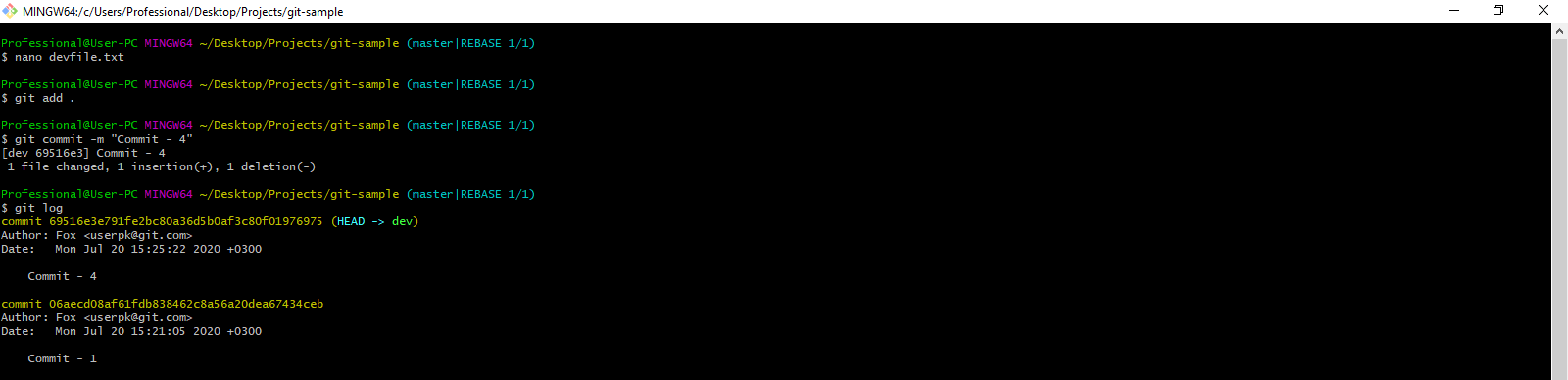
*Рис.28 – Команда git log после выполнения merge*

**

*Рис.29 – Команда rebase*

**

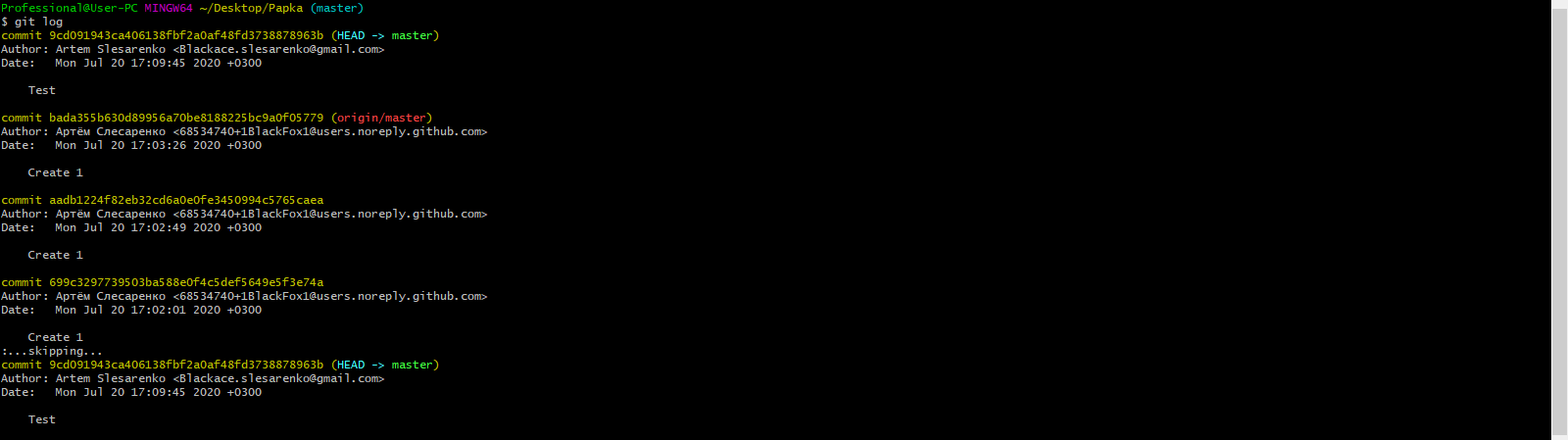
*Рис.30 – Изменение файлов в ветвях, соединённых при помощи merge(1)*

**

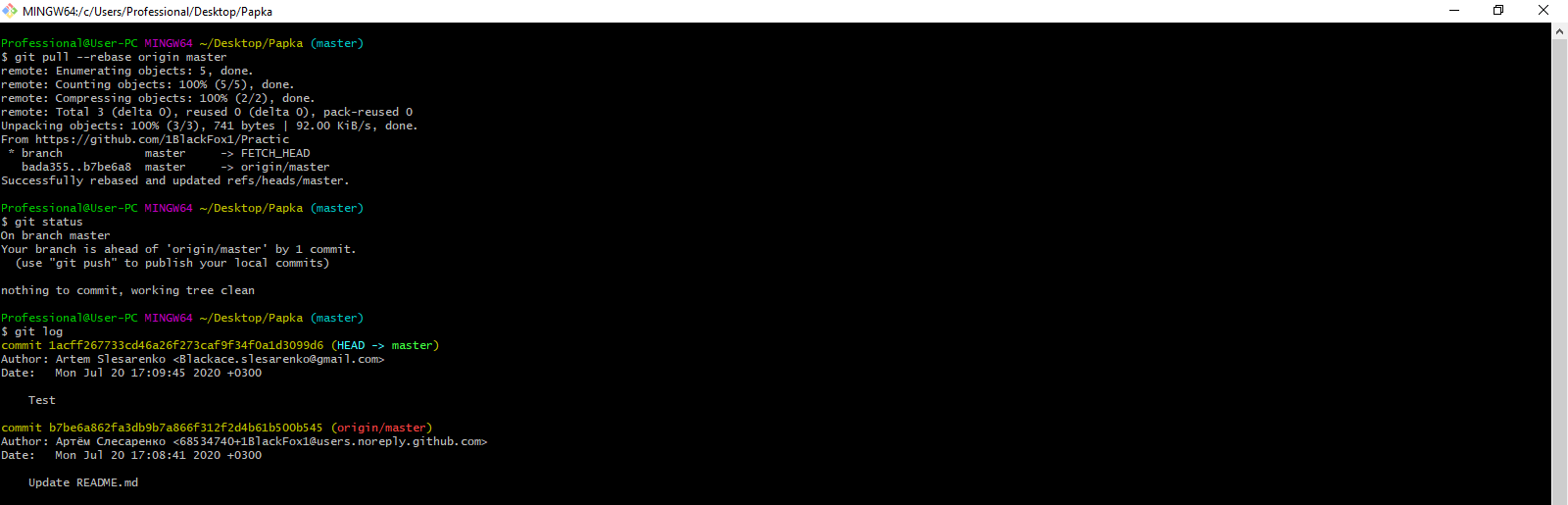
*Рис.30 – Изменение файлов в ветвях, соединённых при помощи merge(2)*

**

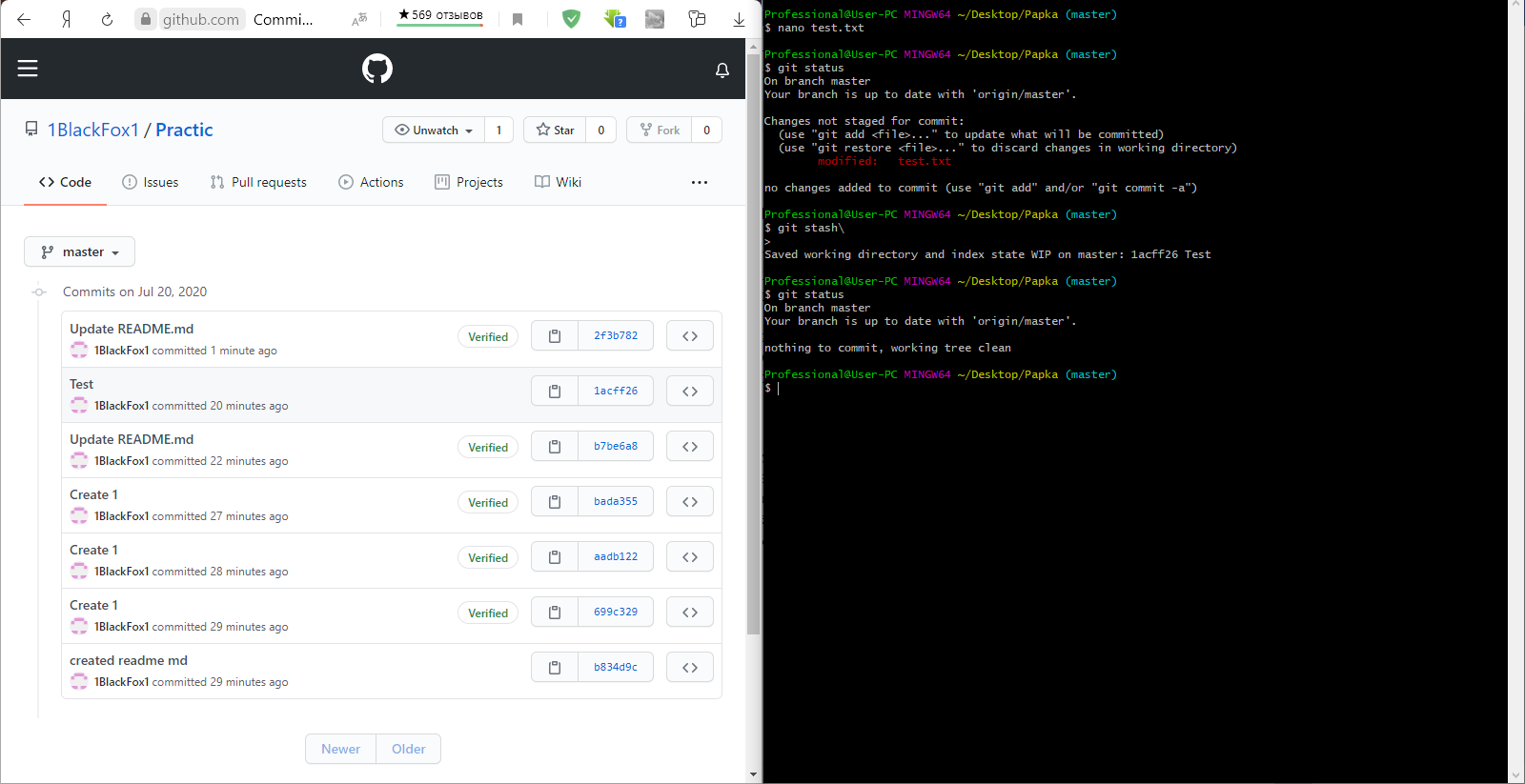
*Рис.31 – Связь с депозиторием сайта GitHub*

**

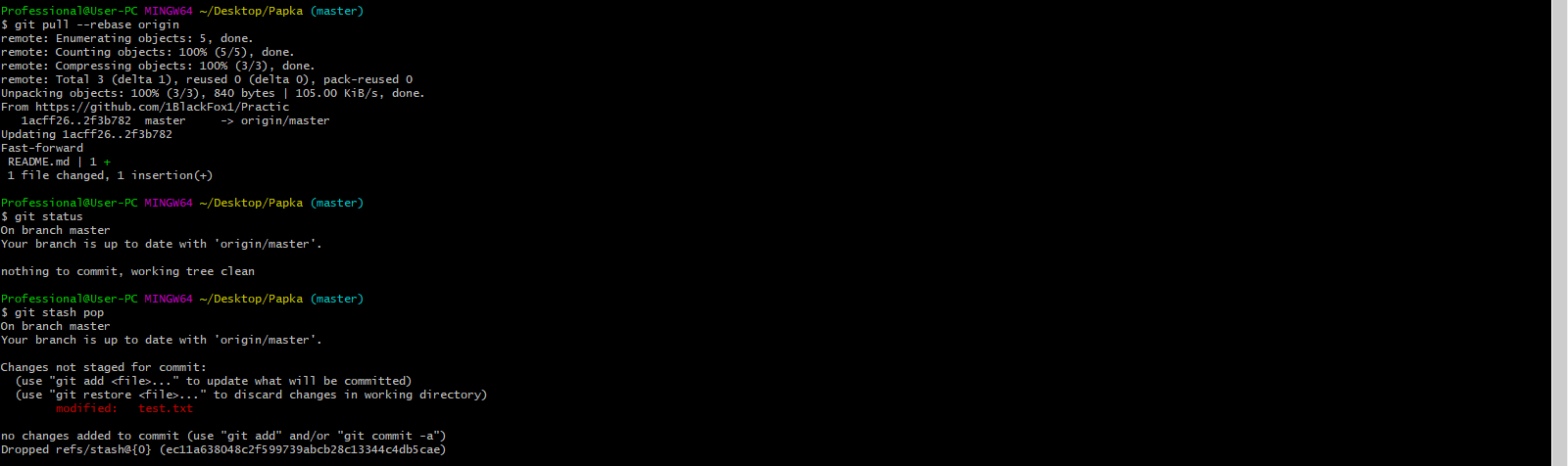
*Рис.32 – Git log*

**

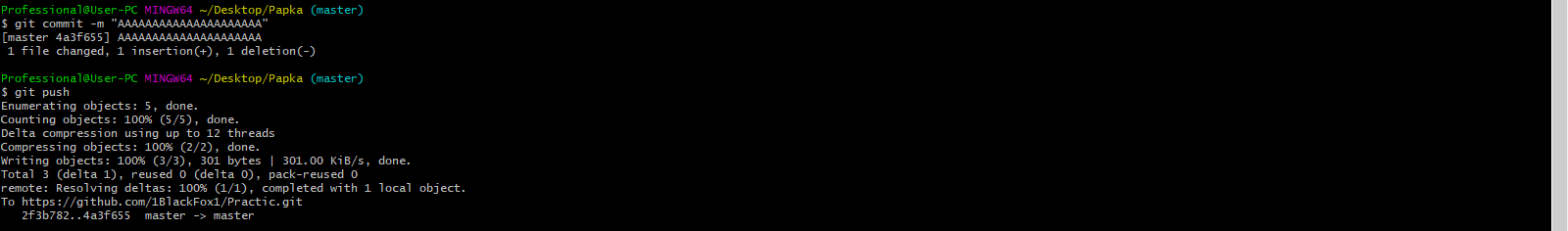
*Рис.33 – Обновление онлайн депозитория*

**

*Рис.34 – Локальное и онлайн изменение разных текстовых файлов*

**

*Рис.35 – Обновление депозитория после изменения локальных файлов(1)*

**

*Рис.35 – Обновление депозитория после изменения локальных файлов(2)*

## **Изучение поисковых систем**

**Поисковая система** (или так называемый «поисковик») – это компьютерная система, созданная специально для поиска необходимой информации в интернете. Самое распространенное применение – сервисы для поиска текстового и графического контента, а также файлов на FTP-серверах, товаров в интернет-магазинах и т.д.

Чтобы найти что-то конкретное, нужно ввести поисковый запрос. Дальше система ищет документы по ключевым фразам, либо слова, как-то связанные с ключевиками. Во время этого поисковик генерирует страницу с результатами поиска. Выдача содержит не только текстовый контент, но и изображения, а также аудиофайлы.

Полезность поисковиков напрямую зависит от релевантности интернет-страниц, которые они возвращают. Яндекс и Google – самые популярные поисковики в нашей стране.

**Типы поисковиков:**

* Поисковые машины. Краулер (робот) обходит сеть, индекс содержит огромный архив копий интернет-страниц, а программное обеспечение сканирует результаты поиска. Многие современные поисковые системы относятся именно к этому типу.
* Каталоги интернет-ресурсов. Данные типы поисковиков получают списки интернет-страниц. Каталоги включают в себя адрес, заголовок и краткие данные об интернет-ресурсе. Каталоги ищут результаты только из описаний страниц, предоставленных веб-мастерами. Преимущество каталогов в ручной проверке сайтов, что непосредственно влияет на качество контента. В этом случае оно будет выше, чем результаты, автоматически обработанные первым типом поисковиков.
* Гибридные системы. Поисковики типа Google и Yahoo сочетают 2 типа поисковиков (поисковые машины, и системы, которыми управляет человек).
* Метасистемы (или мультипоточные системы). Не имеют своей базы данных и поискового индекса. Формируют выдачу за счет переранжирования результатов других поисковиков. В те времена, когда поисковики были не такими умными и имели свой уникальный адрес, этот вид поисковых систем считался полезным.

**Примеры поисковых систем:**

* **Google**



*Рис.36 – Внешний вид поисковой системы Google*

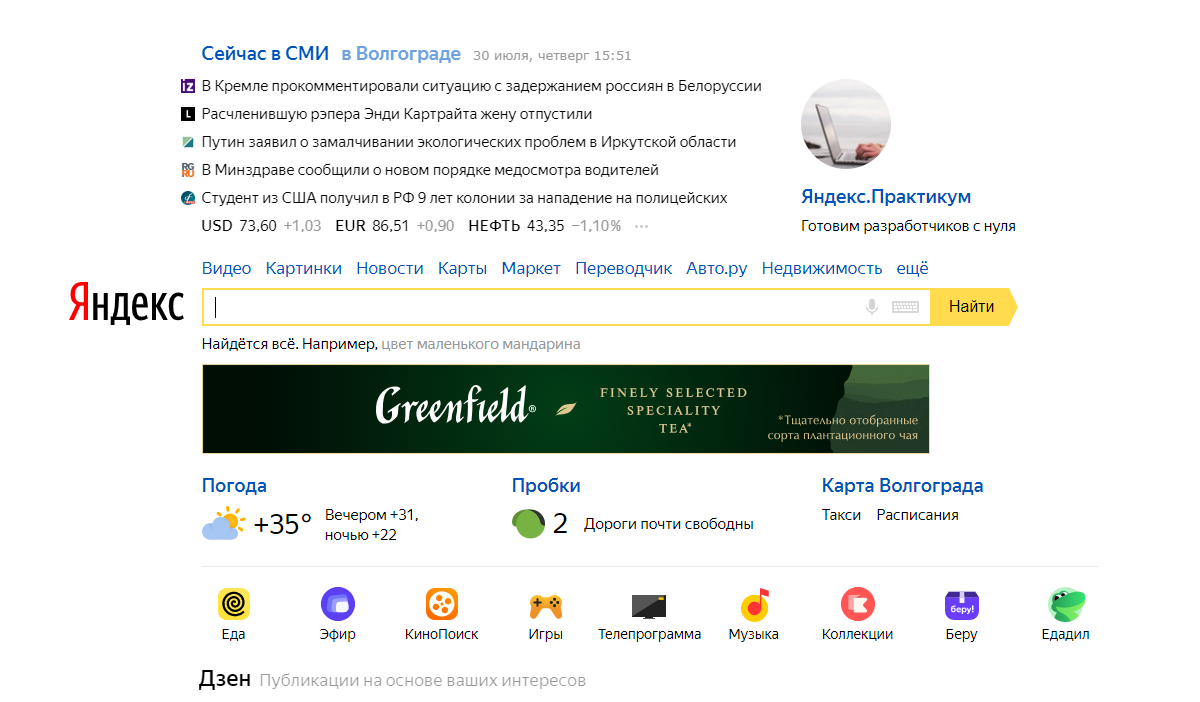
Поисковая система Google по-прежнему остается самой популярной поисковой системой сети Интернет.

За все время нахождения Google на рынке пока ещё ни одной из конкурирующих поисковых систем не удалось сместить американскую поисковую систему с лидирующих позиций.

*Таблица 1:*

|  |
| --- |
|  |
| **Плюсы:** | **Минусы:** |
| Простота использования | Встречаются ссылки с устаревшей информации |
| Обширная база данных | Некоторые ссылки ведут на сайт, который находится на стадии разработки |
| Высокая скорость работы | При совершении запросов необходима внимательность |
| Достаточно точная информация | Отсутствие в некоторых слов ударения усложняет поиск |
| Функция «Мне повезет» |  |
| Живой поиск |  |
|  |

* **Яндекс**



*Рис.37 – Внешний вид поисковой системы Яндекс*

Поисковая система «Яндекс» ориентирована на русскоязычный сегмент интернета, она способна обрабатывать миллионы запросов пользователей ежедневно, предлагая релевантную информацию. В основе системы – искусственный интеллект на машинном обучении, который ежедневно совершенствуется и оптимизирует поисковую выдачу. Сложные алгоритмы ранжирования сайтов нацелены на то, чтобы оградить пользователя сети от низкосортного контента и в полной мере дать ответы на его вопросы.

*Таблица 2:*

|  |
| --- |
|  |
| **Плюсы** | **Минусы** |
| Альтернатива «живого поиска» от Google. | Сильно надоедающая CAPTCHA |
| Можно найти результат из нужного города. | Существует только русская версия |
| Поиск можно переключить с поиска по интернету на поиск по картинкам, видео, блогам, словарям. |  |
| Система периодически подкидывает популярные запросы. |  |

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПО СОЗДАНИЮ ПОИСКОВОГО САЙТА “FOX SEARCH”**

**1.Общие сведения**

Наименование сайта: FoxSearch

Плановые сроки начала и окончания работы: 23.07.2020 – 15.09.2020.

**2.Назначение и цель создания сайта**

**КРАТКАЯ ИНФОРМАЦИЯ**

Поисковая система (англ. search engine) -- это компьютерная система, предназначенная для поиска информации. Она помогает пользователям быстро найти необходимые сведения, реагируя на запрос пользователя выдачей списка ссылок на источники информации. Достаточно набрать в строке поиска интересующий вопрос или фразу, нажать на кнопку «Поиск» или «Search», и через несколько секунд поисковая система выдаст необходимую информацию.

Основным назначением поисковых систем, является предоставления наиболее релевантной (соответствующей вашему запросу) информации за минимально возможное время.

Именно благодаря этому впечатлению, вы возвращаетесь в одну и ту же поисковую систему раз за разом, часто даже не подозревая об этом. Один раз создав впечатления наиболее релевантной и быстрой поисковой системой, вы практически постоянно будете пользоваться этой системой.

Именно для повышения релевантности своей выдачи и ее скорости предоставления, поисковые движки вкладывают огромные суммы денег, времени и энергии. Производят огромное количество аналитической информации, изучают ваши действия и реакцию на ту или иную выдачу, изучают выдачу других систем, исследуют перемещения ваших глаз и многое другое.

**ЦЕЛЬ СОЗДАНИЯ САЙТА**

Предоставить простой сервис поиска информации

**НАЗНАЧЕНИЕ САЙТА**

Сайт предназначен для безопасного и точного поиска информации в сети интернет

**ЦЕЛЕВАЯ АУДИТОРИЯ**

Сайт рассчитан на русскоязычных людей. Возраст 10-60 лет. Пол не важен.

Сайт предназначен в основном для тех людей, которые предпочитают поиск информации без назойливой рекламы и слежки за их историей поиска.

**МОТИВАЦИЯ ИСПОЛЬЗОВАНИЯ САЙТА:**

* Возможность поиска по определённому критерию (например, только по сайтам, содержащим в себе информацию о кинофильмах)
* Возможность фильтрации шокирующего контента и контента «для взрослых»
* Отсутствие сбора данных о пользователях, осуществляющих поиск информации

**3.Сценарий пользования сайтом**

1. Пользователь заходит на сайт и видит строку поиска
2. Он может, при желании, настроить безопасный поиск, т.е. поиск сайтов, не содержащих шокирующего контента или контента «для взрослых»
3. Пользователь вводит нужную информацию и интересующую его тематику (опционально)
4. Поисковая система считывает введённые данные и начинает поиск.
5. На экран выводятся найденные сайты

# Разработка прототипа проекта

**Прототипирование** - это разработка прототипа сайта. Прототип - это эскиз будущего ресурса, где отрисована вся его структура до мельчайших подробностей. Именно использование статистических и динамических прототипов позволяет разработчику сделать продуманный, хорошо работающий сайт.

Хорошо проработанный прототип — это своеобразный чертеж, который сокращает время на дальнейших этапах разработки. Он позволяет представить проект в конечном виде и внести корректировки еще до старта работ. После остается только технически реализовать задуманное, не изобретая ничего на ходу. Время вложенное в прототип хорошо окупится в будущем, так как сократит количество доработок, которые появятся в конце разработки. Они в любом случае будут, но так их будет минимум. Совсем без прототипа, может оказаться, что после запуска нужно переделывать добрую половину сайта. И снова в работу вступят программист, верстальщик, дизайнер… И каждому нужно будет опять платить деньги. То что на этапе исправление прототипа займет 30 минут, на этапе разработки превратится в 300 минут рабочего времени. Создание прототипа сайта — это часть работы, за которую отвечает интернет-маркетолог и от того, как она будет выполнена зависит дальнейший успех веб-сайта (какая будет конверсия, как будут вести себя пользователи, удобно ли им будет взаимодействовать с сайтом и т. д.)

Прототип страницы сайта понятие достаточно широкое, так как это может быть просто эскиз от руки на бумаге, схема в формате картинки, либо детальный, интерактивный прототип. Классифицируем их по основным характеристиками.

**По глубине проработки деталей:**

* с низкой детализацией;
* с высокой детализацией.
* По возможности взаимодействовать с прототипом:
* графический (статичный) — в виде графического изображения;
* интерактивный — с элементами взаимодействия, например, с кликабельными ссылками, рабочим слайдером, всплывающими формами или другими активными элементами.

По месту хранения:

* аналоговый — в виде рисунка;
* цифровой — созданный в программе (доступ к редактированию только у разработчика);
* облачный — находящийся в облаке (доступ к редактированию возможен для нескольких участников).

Прототипирование модели сайта выполним с помощью программы Axure RP. Axure существует уже около десяти лет, чтобы позволить инженерам UX с легкостью добраться до деталей создания прототипа программного обеспечения. Axure, помимо того, что является инструментом для создания прототипов, имеет мощное сообщество, вносящее свой вклад в мир UX с множеством примеров и умелых взаимодействий.

На более высоком уровне ниже перечислены функции, предлагаемые Axure:

* Диаграммы и документация
* Эффективное прототипирование с динамическим контентом
* Условные потоки
* Красивые анимации для улучшения взаимодействия
* Адаптивные виды
* Поддержка на Windows и Mac

Для быстрого создания прототипов Axure RP предоставляет разнообразный набор инструментов и методов, которые всегда помогают инженерам-аналитикам / пользовательскому интерфейсу визуализировать конечную цель.

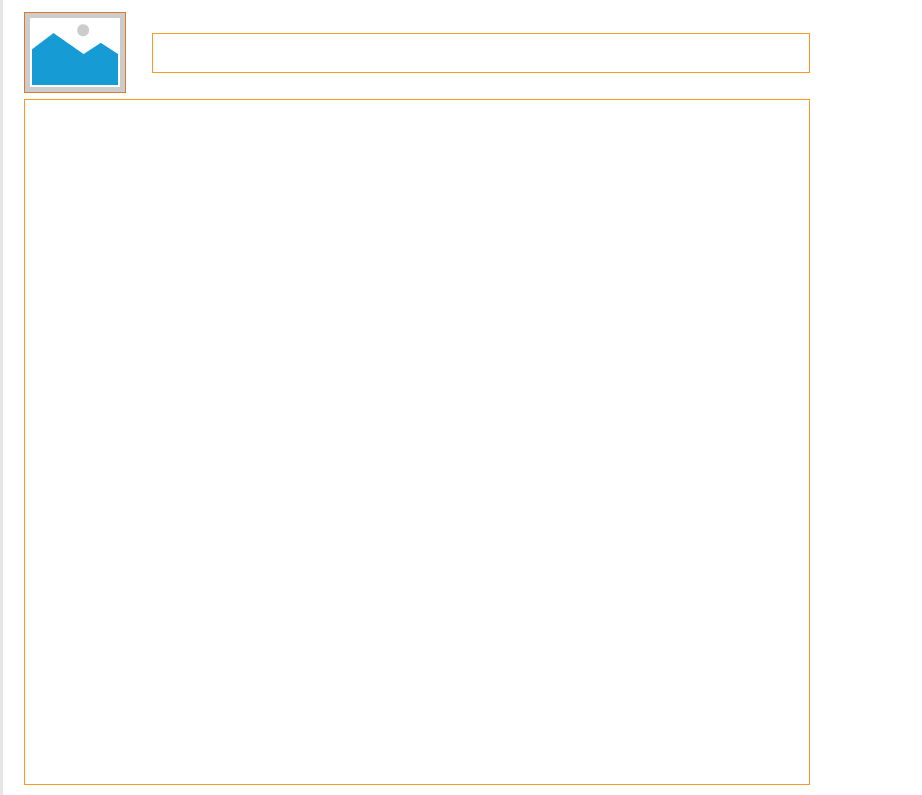
Благодаря сильному сообществу, которое всегда готово оказать помощь, Axure RP становится предпочтительным инструментом для энтузиастов и практиков UX.

**Ход работы:**

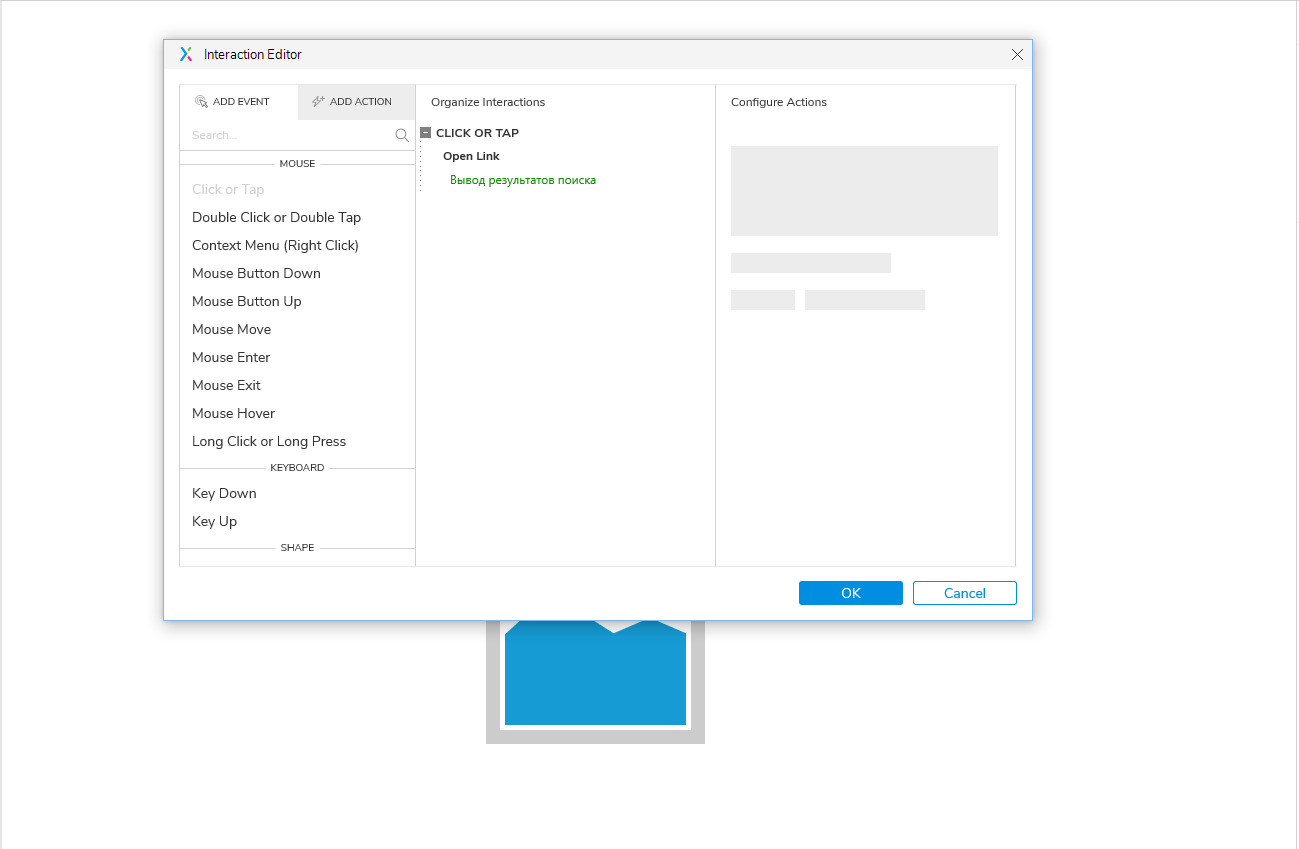
1. Загрузим и установим программу Axure RP 9
2. Ознакомимся с функционалом программы
3. Создадим прототип главной страницы сайта
4. Создадим прототип страницы результатов поиска



*Рис.1 – Внешний вид главной страницы*



*Рис.2 – Внешний вид страницы вывода результатов поиска*

 *Рис.3 – Добавление простых интерактивов*

# Заключение

В процессе прохождения учебной практики овладели навыками использования «Git», научились анализировать проекты конкурентов, составлять техническое задание для системы, а также реализовывать прототипа с помощью среды разработки.