# Отчет по исследованию возможностей Git в командной строке

### (Terminal/Command Shell)

### Введение

В данном отчете я исследовал возможности работы с системой контроля версий Git через командную строку (Terminal). Этот инструмент предоставляет прямой доступ ко всем функциям Git и является основным способом взаимодействия для многих разработчиков

Выполненные команды и их демонстрация

# 1. Настройка Git

```
Windows PowerShell
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscoreб)
PS C:\Users\maqa2> git config --global user.name "Sadiq"
>> git config --global user.email "sadiqspark@mail.ru"
>> git config --global init.defaultBranch main
PS C:\Users\maqa2> git config --global user.name "Sadiq"
PS C:\Users\maqa2> git config --global user.email "sadiqspark@mail.ru"
PS C:\Users\maqa2> git config --global init.defaultBranch main_
```

2. Создание и инициализация репозитория

```
PS C:\Users\maqa2> mkdir my-git-project
>> cd my-git-project
>> git init
```

3. Проверка статуса репозитория

```
PS C:\Users\maga2\my-git-project> git status
On branch main
No commits yet
```

Особенность: Команда показывает текущее состояние рабочей директории и staging area.

4. Создание файлов и добавление в отслеживание

```
git add README.md
```

```
5. Coздание первого коммита

PS C:\Users\maqa2\my-git-project> git commit -m "!
[main (root-commit) 2d7332b] Добавлен README файл

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md

PS C:\Users\maqa2\my-git-project> _
                                                                                                                                                                                                    Добавлен README
```

6. Просмотр истории коммитов

```
PS C:\Users\maqa2\my-git-project> git log
commit 2d7332bf36c1683babe8082421d16e878acacf93 (HEAD -> main)
Author: Sadiq <sadiqspark@mail.ru>
Date: Sat Sep 27 22:53:15 2025 +0300
     Добавлен README файл
PS C:\Users\maga2\my-git-project> _
```

# 7. Создание и переключение между ветками

```
Добавлен README фаил
PS C:\Users\maqa2\my-git-project> git branch feature/new-feature
>> git checkout feature/new-feature
Switched to branch 'feature/new-feature'
```

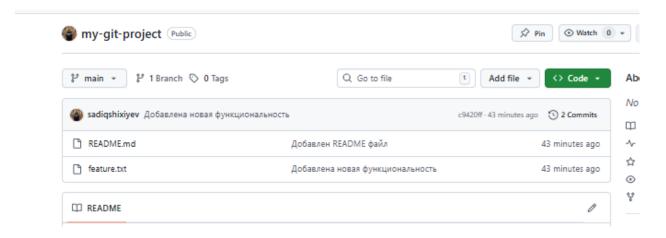
#### 8. Внесение изменений и слияние веток

```
PS C:\Users\maqa2\my-git-project>
>> echo "Новая функциональность"
                               > feature.txt
  git add feature.txt
git commit -m "Добавлена новая функциональность"
>> git switch main
>>
>>
PS C:\Users\maqa2\my-git-project>
```

9. Работа с уда<u>ленным репозиторием</u>

PS C:\Users\maqa2\my-git-project> git remote add origin https://github.com/sadiqshixiyev/my-git-project.git
PS C:\Users\maqa2\my-git-project> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compression using up to 4 threads
Compression objects: 100% (4/4), done.
Writing objects: 100% (6/6), 611 bytes | 305.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sadiqshixiyev/my-git-project.git

\* [new branch] main -> main
branch | main -> main
branch | main -> track 'origin/main'.
PS C:\Users\maqa2\my-git-project>
PS C:\Users\maqa2\my-git-project>



# 10. Получение изменений из удаленного репозитория

```
PS C:\Users\maqa2\my-git-project> git fetch origin
>> git pull origin main
From https://github.com/sadiqshixiyev/my-git-project
* branch main -> FETCH_HEAD
Already up to date.
PS C:\Users\maqa2\my-git-project>
```

### Особенности работы с Git в командной строке

Преимущества:

**Полный контроль**: Доступ ко всем функциям Git без ограничений **Автоматизация**: Легко интегрируется в скрипты и CI/CD процессы

Скорость: Быстрее графических интерфейсов для опытных пользователей

Универсальность: Работает на любом сервере или в контейнере

### Заключение

Работа с Git через командную строку предоставляет максимальную гибкость и контроль над процессом версионирования. Несмотря на более крутую кривую обучения по сравнению с графическими интерфейсами, командная строка остается наиболее мощным и универсальным инструментом для работы с Git.

Основные преимущества включают возможность автоматизации, полный доступ ко всем функциям Git и эффективную работу в серверных средах. Для начинающих рекомендуется сочетать изучение командной строки с визуализацией истории через графические инструменты.