

Довольно удобным инструментом для организации работы команды является GitHub. Изначально эта платформа предназначена для отслеживания и фиксации изменений в исходном коде программ, но в данном пособии мы рассмотрим работу с разными типами файлов.

## Начало работы

---

Каждому члену команды нужно бесплатно зарегистрировать аккаунт на GitHub (<https://github.com>)

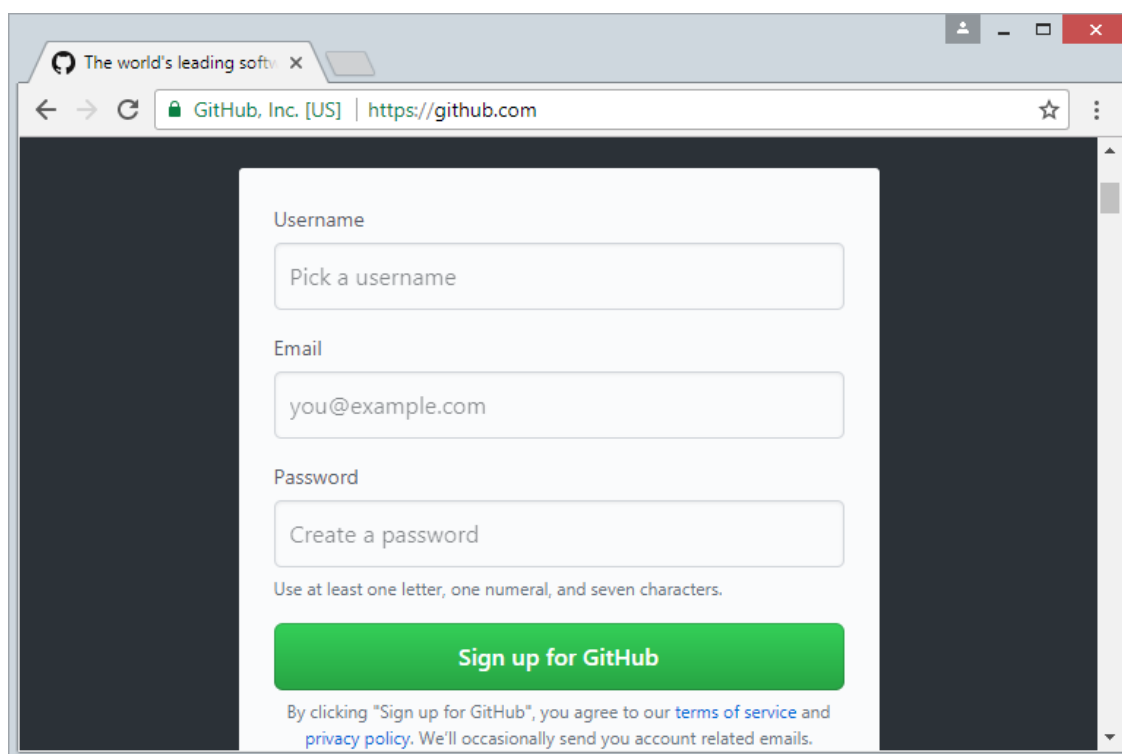


Рисунок 1. Общий вид окна регистрации на платформе GitHub

Регистрация аккаунта очень проста. Нужно ввести username/password и email. После регистрации нужно войти в свой почтовый ящик и активировать аккаунт.

**Совет:** лучше заранее попросить участников команды указывать в качестве username'ов реальные имена и фамилии, либо указывать их в графе name в профиле, чтобы потом было проще отследить, кто именно внёс какие правки в общий проект.

## Создание Организации

---

Так как в ходе работы над проектом используется довольно большой набор файлов разного типа (чертежи, 3D-модели, коды программ, текстовые файлы, etc.), поэтому руководитель или капитан команды должны создать для команды Организацию.

**Примечание:** Организация позволяет создавать репозитории принадлежащие группе пользователей. Организация позволяет нескольким участникам вносить свободно изменения в проект, без дополнительных операций по слиянию.

Приступим к созданию Организации. В верхней панели находим «Добавить», в выпадающем меню выбираем вариант «New Organization».

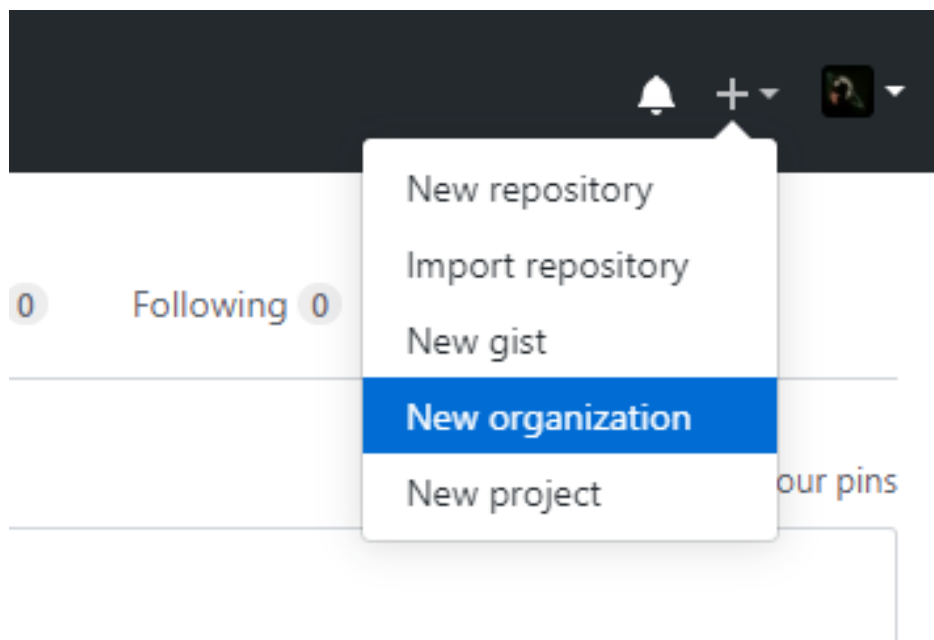


Рисунок 2. Выпадающее меню функции «Добавить»

Для наших целей достаточно выбрать вариант «Team for Open Source».

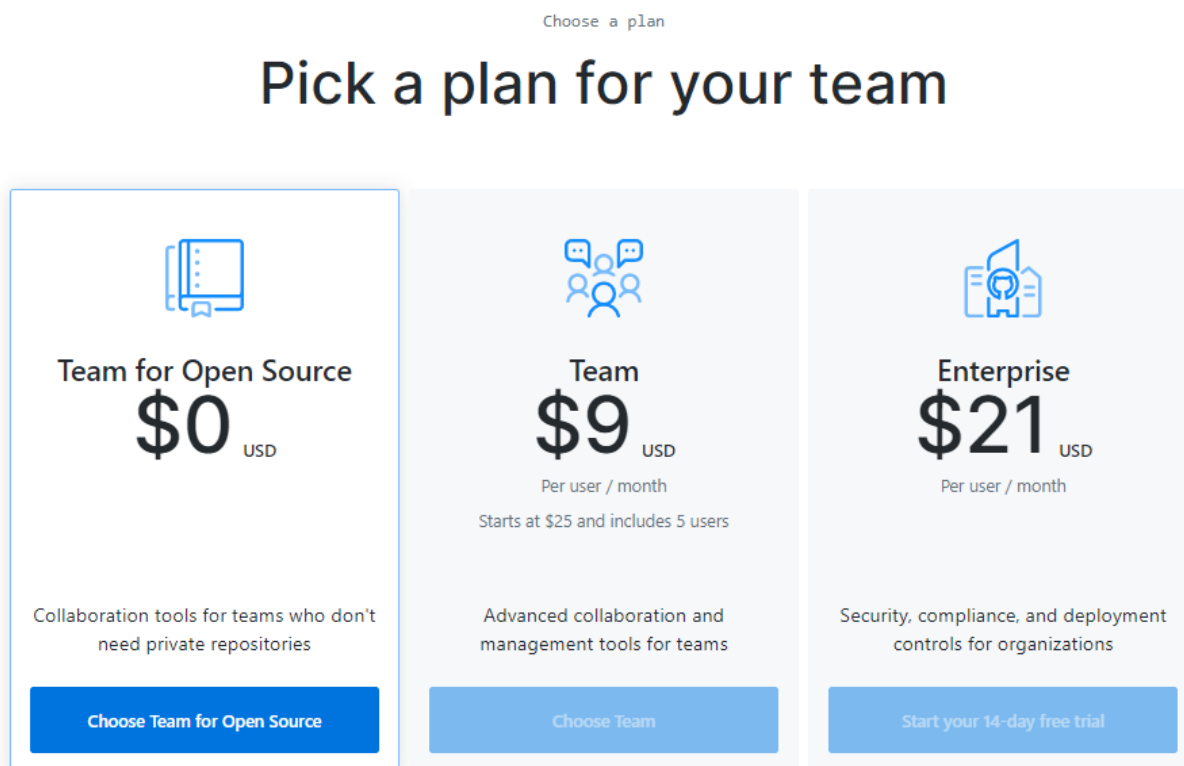


Рисунок 3. Варианты «тарифов» для Организации

**Примечание:** все репозитории в «Team for Open Source» публичные, т.е. любой даже сторонний пользователь может просматривать их содержимое. Но не стоит беспокоиться! Вряд ли кто-то станет тратить время на поиск репозитория другой команды, чтобы «подсмотреть» чужие решения. Игра не стоит свеч!

Заполняем данные об Организации. В графе «Organization account name» стоит указать название команды, в графе «Contact email» можно указать либо электронную почту руководителя, либо капитана команды. В графе «This organization belongs to» выбираем вариант «My personal account» для простоты работы.

Tell us about your organization

## Set up your Team

**Organization account name \***

CanSat-Test-Team✓

This will be the name of your account on GitHub.  
Your URL will be: <https://github.com/CanSat-Test-Team>.

**Contact email \***

example@mail.ru✓

**This organization belongs to: \***

- My personal account**  
I.e., radical95 (Arseny Radushin)
- A business or institution**  
For example: GitHub, Inc., Example Institute, American Red Cross

**Next**

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

Рисунок 4. Пример заполнения анкеты Организации

Организация создана. Теперь можно добавить уже успевших зарегистрироваться членов команды. Для этого в графе «Search by username, full name or email address» вводим данные о нужном пользователе, кликаем по профилю.

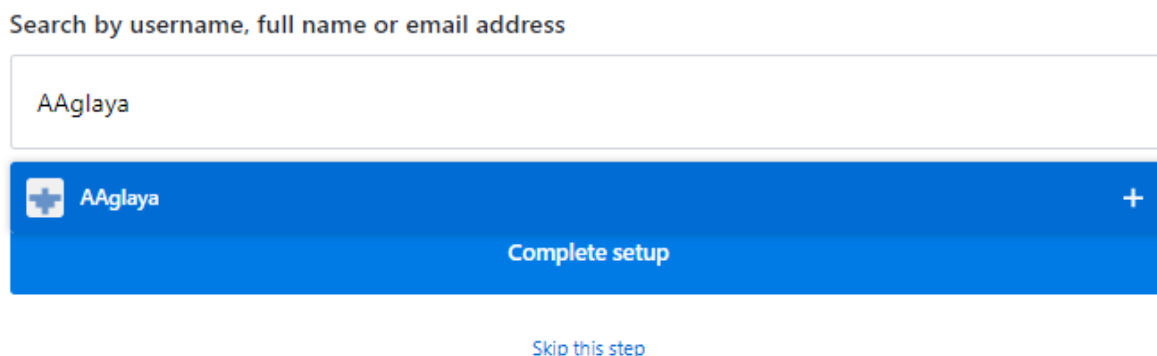


Рисунок 5. Пример поиска члена команды и добавление его из списка

Если к моменту создания организации никто не успел зарегистрироваться, то пропускаем этот шаг нажатием на опцию «Skip this Step».

## Добавление участников

В главном меню Организации есть вкладка «People». Здесь отображаются списки действующих участников Организации и отправленные приглашения.

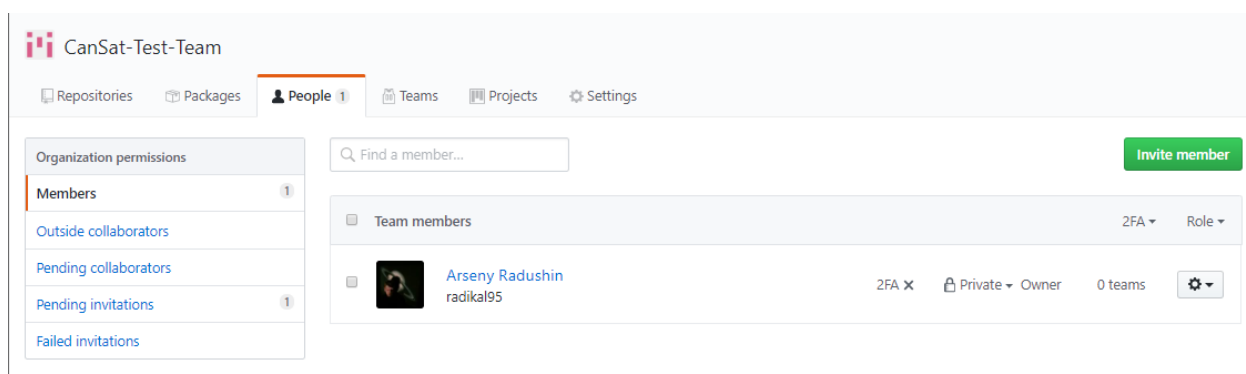


Рисунок 6. Вид вкладки «People»

Изначально единственным участником Организации является только её создатель. Остальных требуется приглашать отдельно. Разберём два случая:

1. Если участник был приглашён во время создания Организации, то ему на почту придёт письмо-приглашение. Участник должен его принять в течение семи дней с даты отправки.

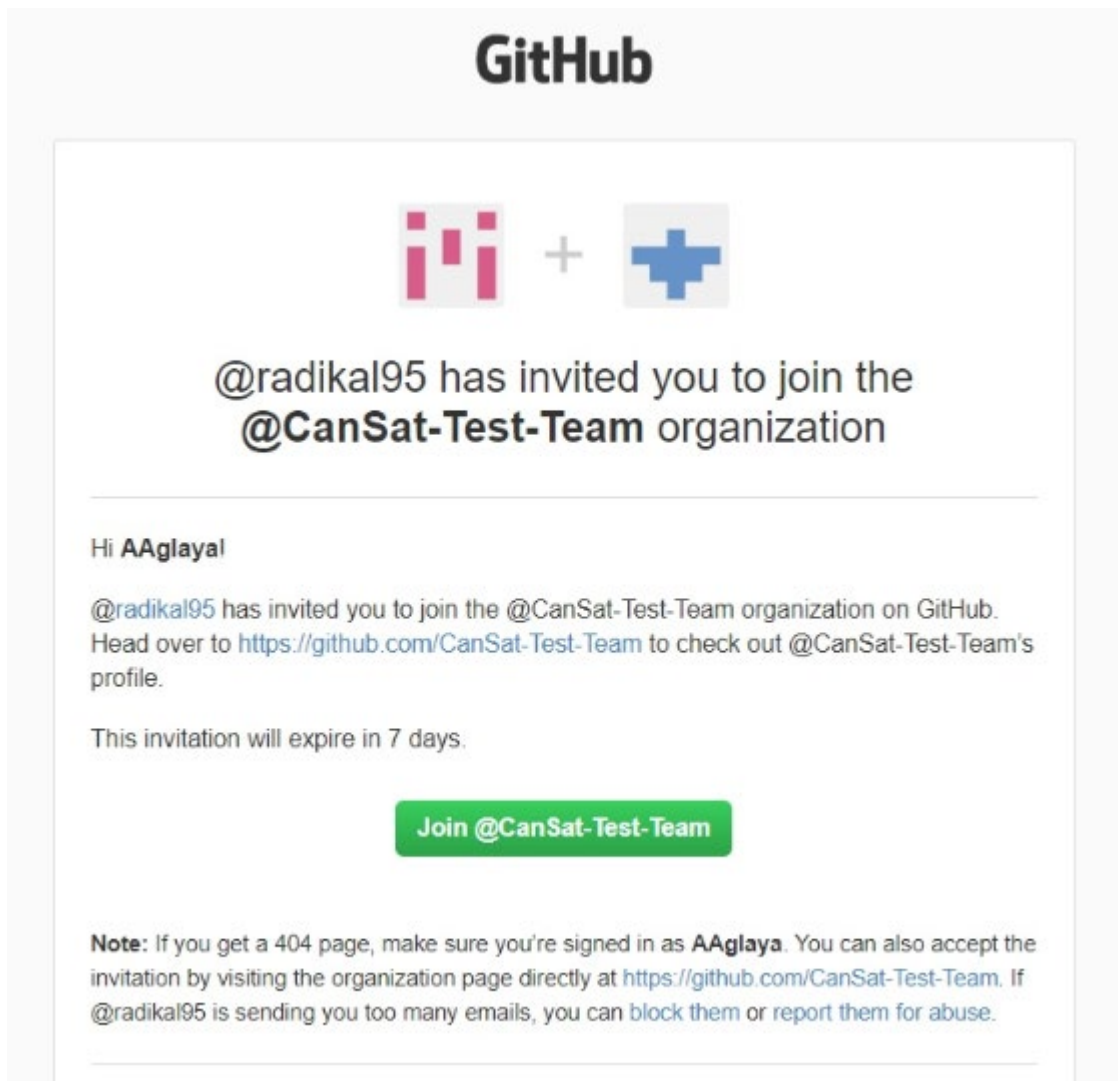


Рисунок 7. Вид письма-приглашения, полученного участником

Когда участник примет приглашение, мы увидим его в списке «Members» вкладки «People».

По умолчанию такие участники добавляются с ролью «Member». Рекомендуется для простоты работы изменить роль на «Owner». Для этого:

- в списке «Members» вкладки «People» в строке нужного участника нажать «Настроить» (значок шестерёнки);
- выбрать опцию «Change role...» в выпадающем списке;

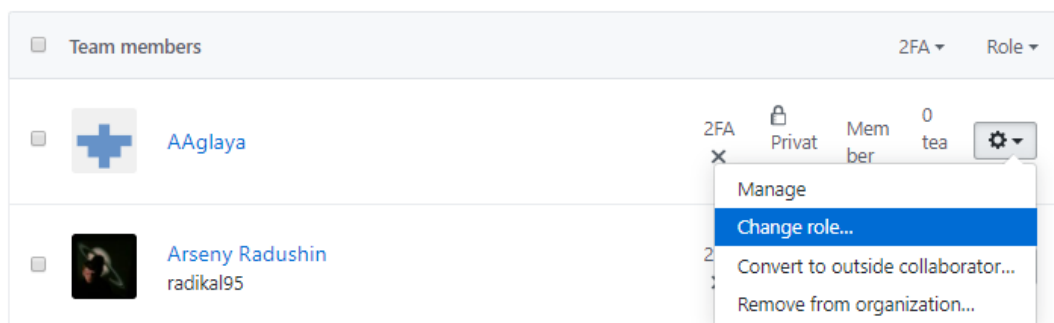


Рисунок 8. Вид выпадающего списка для смены роли

- в предложенном списке выбрать вариант «Owner»;
2. Если участник был **не был** приглашён во время создания Организации, то его надо пригласить через специальную форму. Для этого:
- в вкладке «People» нажать зелёную клавишу «Invite member»;
  - в строке ввести username участника, найти его в выпадающем списке;
  - нажать клавишу «Invite»;
  - в предложенном списке выбрать роль «Owner»;
  - нажать клавишу «Send invitation».

В этом случае участник тоже получит письмо приглашение (см. рисунок 7), которое надо будет также принять в течение семи дней. В этом случае менять отдельно его роль на «Owner» уже не нужно.

## Создание репозитория

Приступим к созданию репозитория. В общем-то, для простоты можно условно принять, что репозиторий — это папка, в которой мы компонуем одинаковые по тематике файлы: коды программ, чертежи, модели, etc. Для создания репозитория нужно:

- перейти в вкладку «Repositories»;
- нажать клавишу «Create a new repository»;
- далее в графе «Repository name» нужно ввести название будущей папки;
- нажать клавишу «Create repository».

**Совет:** папки следует называть односложно, согласно их наполнению. Для исходных кодов лучше назвать папку «Codes», для чертежей — «Blueprints». Но это всегда остаётся на усмотрение капитана или руководителя.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

---

Owner: CanSat-Test-Team / Repository name \*: Codes ✓

Great repository names are short and memorable. Need inspiration? How about [redesigned-sniffle?](#)

Description (optional)

Рисунок 9. Пример заполнения анкеты при создании репозитория

Репозиторий создан! Теперь установим графический клиент и свяжем его с созданным репозиторием.

## Работа с GitHub Desktop

Каждому члену команды нужно загрузить (<https://desktop.github.com/>). После скачивания нужно установить GitHub на компьютер. После успешной установки и запуска на экране появится приветственное окно. В нём надо нажать «Sign in to GitHub.com» и ввести данные своего профиля.

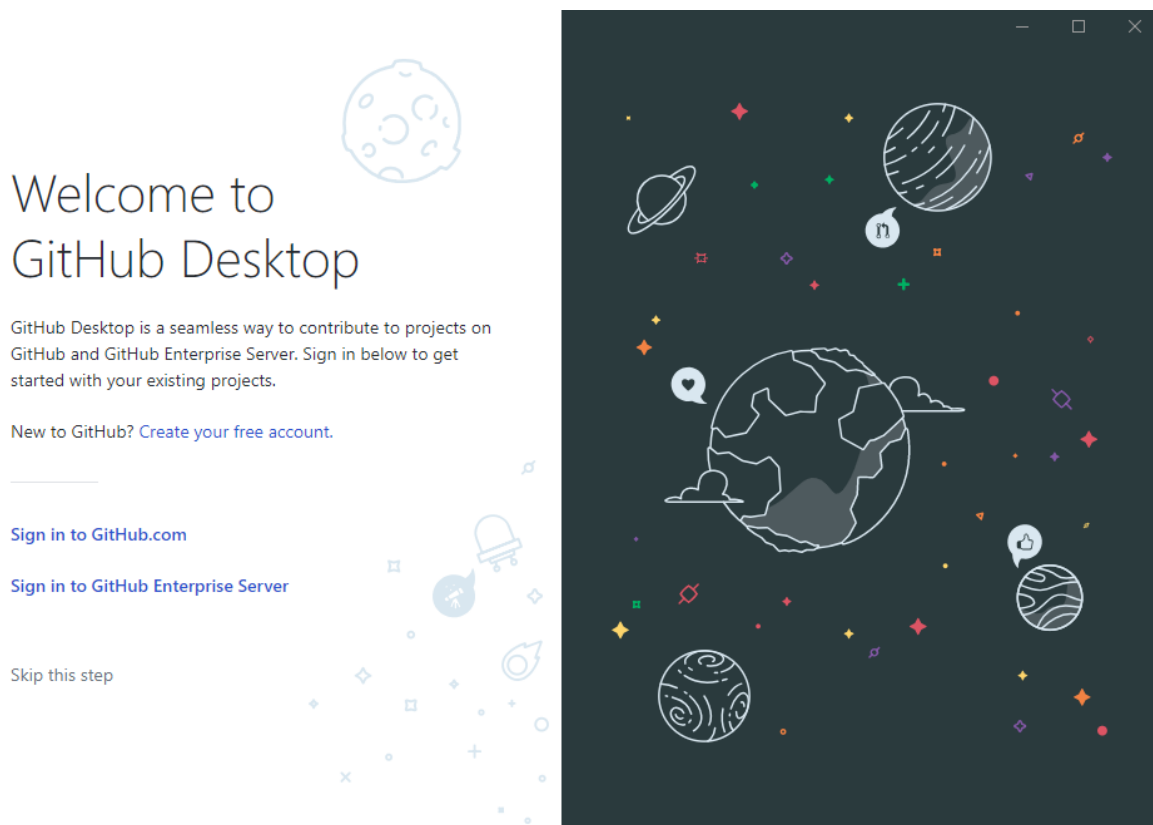


Рисунок 10. Приветственное окно GitHub Desktop

Теперь добавим уже созданный ранее репозиторий. Для этого нужно:

- в верхнем меню выбрать вкладку «File»;
- в выпадающем списке выбрать опцию «Clone a repository»;
- в всплывающем окне в списке найти созданный до этого репозиторий;

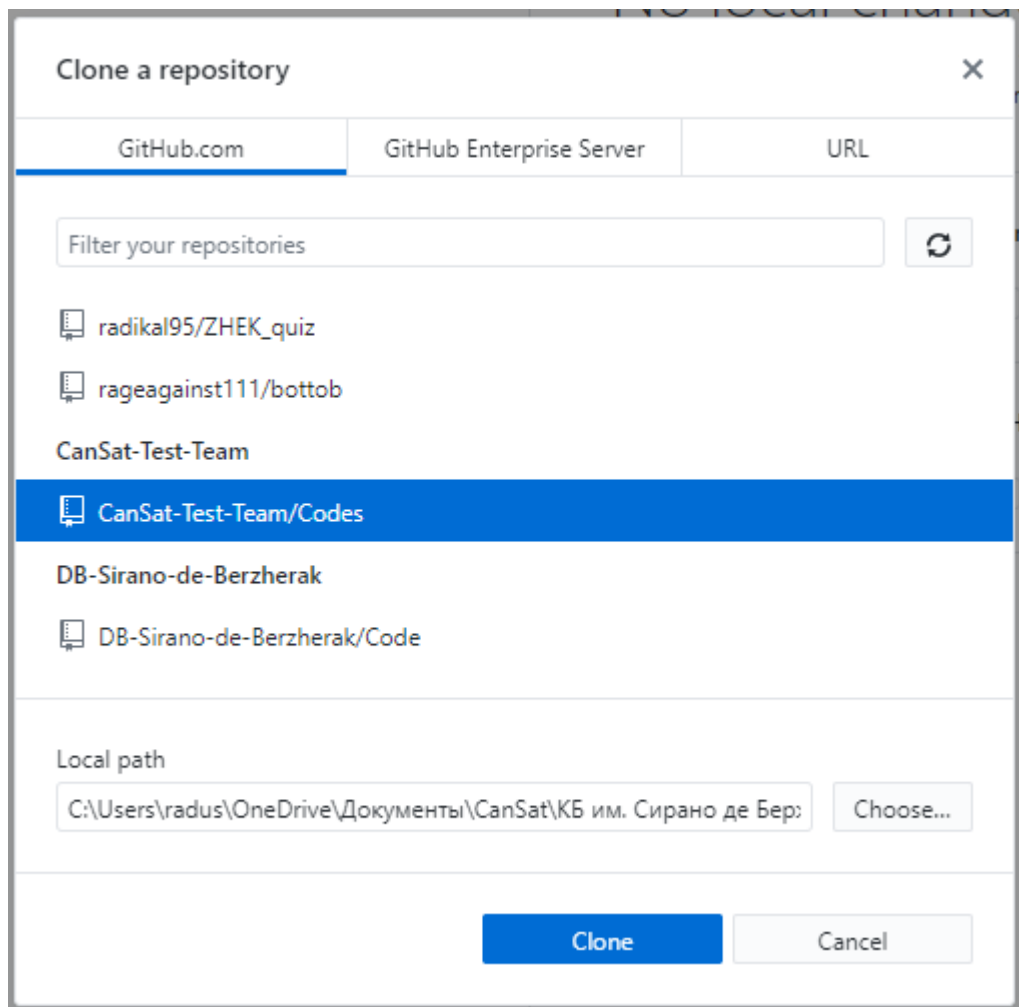


Рисунок 11. Окно «клонирования» репозитория

- в графе «Local path» нажать «Choose...» и выбрать ту папку, куда клонировать репозиторий. В будущем эта папка будет «рабочей» локальной папкой для этого репозитория;
- нажать клавишу «Clone».

Теперь, если мы внесём какие-то изменения в файлах в этой папке, они будут отслежены и зафиксированы. Рассмотрим пример:

- добавим в локальную папку текстовый файл;

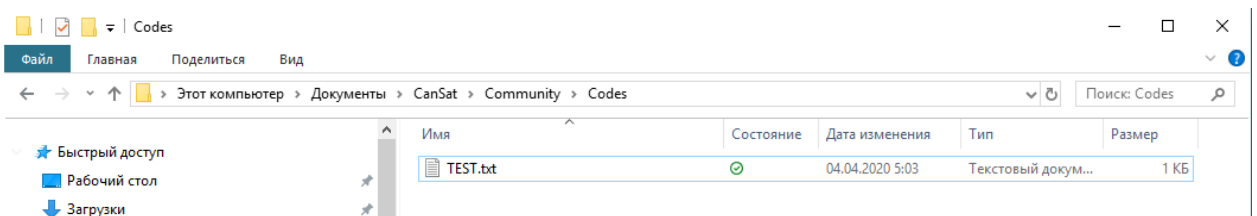


Рисунок 12. Окно локальной папки с тестовым текстовым файлом.

- откроем репозиторий в GitHub Desktop. В нём отображаются файлы, в которых произошли изменения;



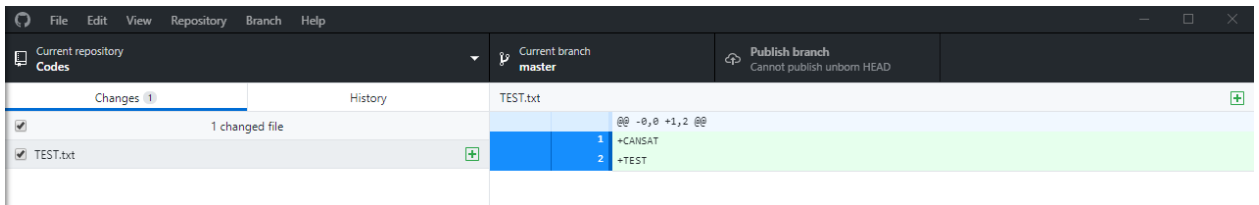


Рисунок 13. Окно с отслеживанием изменений в локальной папке репозитория

- теперь нужно зафиксировать изменения. Для этого нужно нажать кнопку Commit;

**Совет:** можно несколько раз подряд фиксировать несколько изменений, особенно если они относятся к разным файлам и требуют разных пояснений.

- затем нужно отправить зафиксированные изменения на сервер. Для этого нажимаем клавишу «Publish branch».

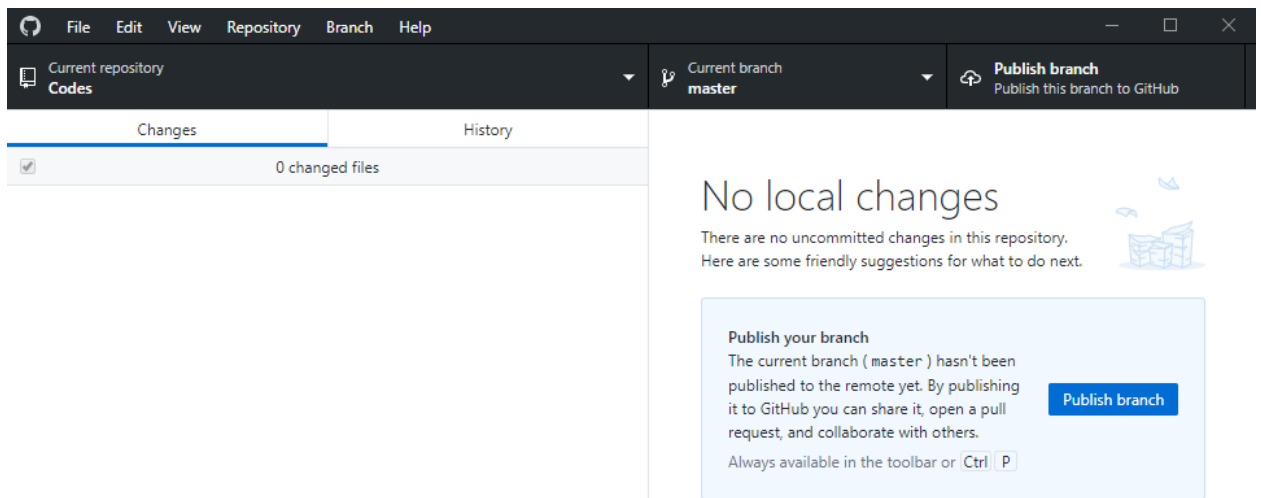


Рисунок 14. Вид окна Github Desktop перед отправкой данных на сервер

- в конце стоит убедиться в наличии файлов на сервере через браузер.

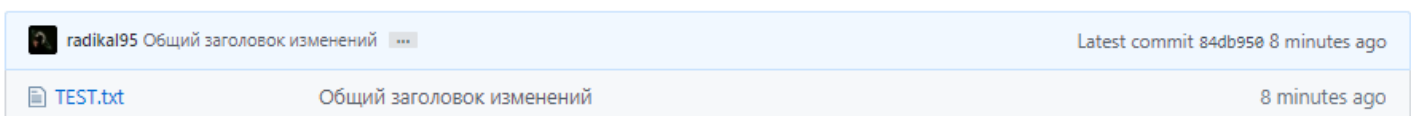


Рисунок 15. Вид зафиксированного изменения в браузере.

## Заключение

Организация работы по принципу, описанному в данном методическом указании, помогает не только при дистанционном варианте работы, но и при штатном режиме. Не требуется отправлять друг другу архив со всеми файлами проекта, не требуется контролировать версии и изменения, уменьшается риск запутаться в большом

количестве файлов, которые сопровождают создание изделия, руководителю и капитану удобно отслеживать прогресс и вклад каждого из участников. Данное методическое пособие не является «истиной» в последней инстанции. Автор предполагает, что многие решения могли бы быть нерациональны и неэффективно использовали все возможности описанной платформы. Автор открыт для любых предложений по изменениям и дополнениям.