

# Система контроля версий Git

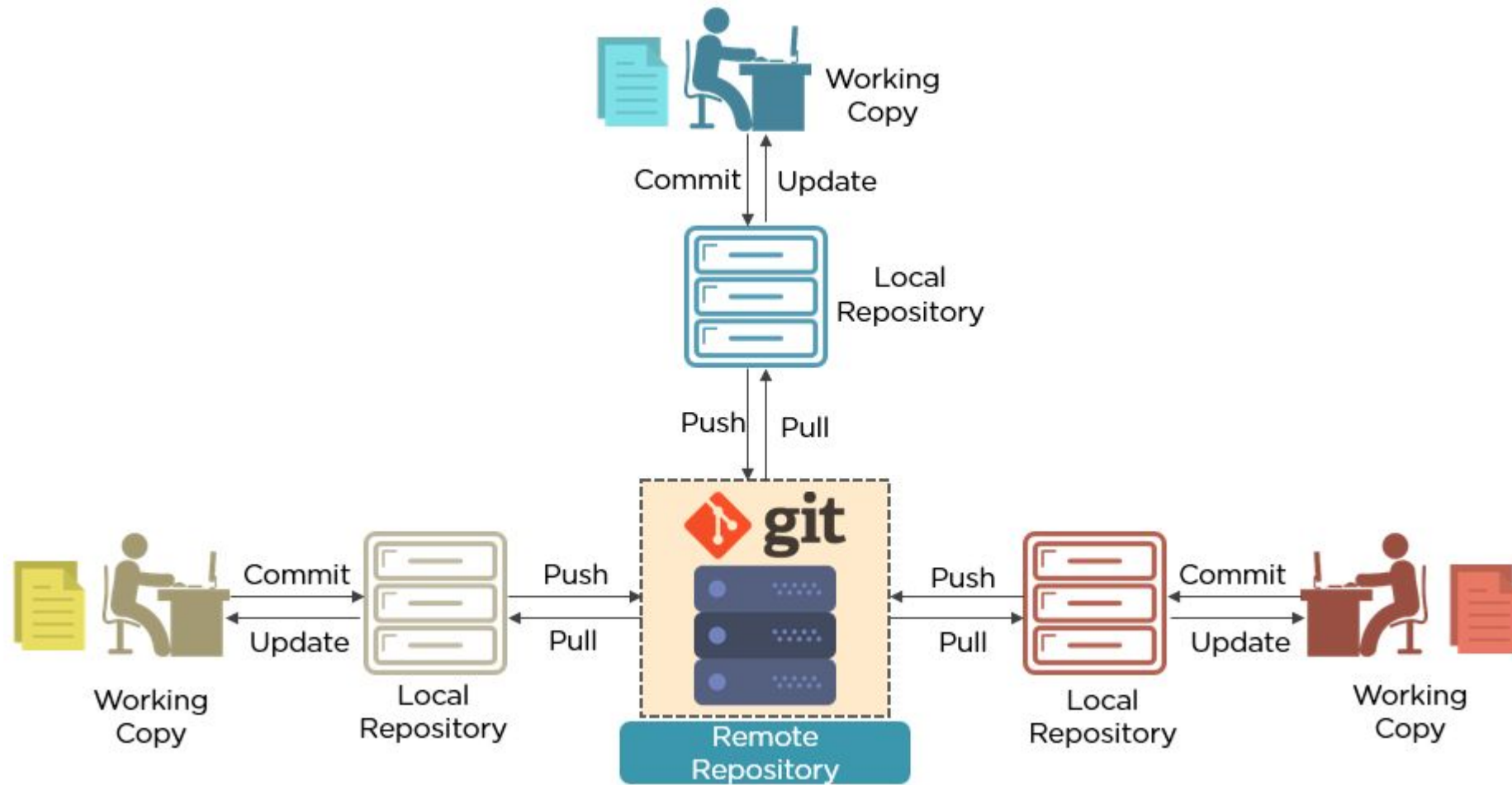
## **Задачи :**

- *Резервное копирование и восстановление*
- *Синхронизация*
- *Отмена изменений*
- *Контроль изменений*
- *Возможность командной работы*

## **Достоинства:**

- *Скорость*
- *Простота дизайна*
- *Поддержка нелинейной разработки (тысячи параллельных веток)*
- *Полная распределённость*
- *Возможность эффективной работы с большими проектами*

# Схема работы с репозитием Git



# Графические интерфейсы

Существует несколько графических интерфейсов для работы с Git

- *GitHub Desktop*
- *GitKraken*
- *SmartGit*
- *SourceTree*
- *TortoiseGit*

## **TortoiseGit – это ...**

**TortoiseGit** — графический клиент системы контроля версий для ОС семейства Microsoft Windows. Распространяется по лицензии GNU PL. Обеспечивает совместную работу членов команды. Позволяет легко вносить изменения в общий репозиторий и получать изменения от других членов команды.

<https://tortoisegit.org/about/>

# Где скачать?

<https://tortoisegit.org/download/>



**TortoiseGit**  
Windows Shell Interface to Git

[About](#)

[Download](#)

[Support](#) ▼

[Contribute](#)

[TortoiseGit.org](#) » [Download](#)

## Download

The current stable version is: **2.8.0**

For detailed info on what's new, read the [release notes](#).

[FAQ: System prerequisites and installation](#) - This version doesn't run on Windows Vista and below, use [2.4.0](#) instead.

### [Donate](#)

Please make sure that you choose the right installer for your PC, otherwise the setup will fail.

for 32-bit Windows	for 64-bit Windows
<a href="#">Download TortoiseGit 2.8.0 - 32-bit (~16.4 MiB)</a>	<a href="#">Download TortoiseGit 2.8.0 - 64-bit (~19.0 MiB)</a>

### [Pre-Release Builds](#)

Before reporting an issue, please check that your problem isn't fixed in our latest [preview release](#). Also see [What to do if a crash happened?](#)

# GitHub – это ...

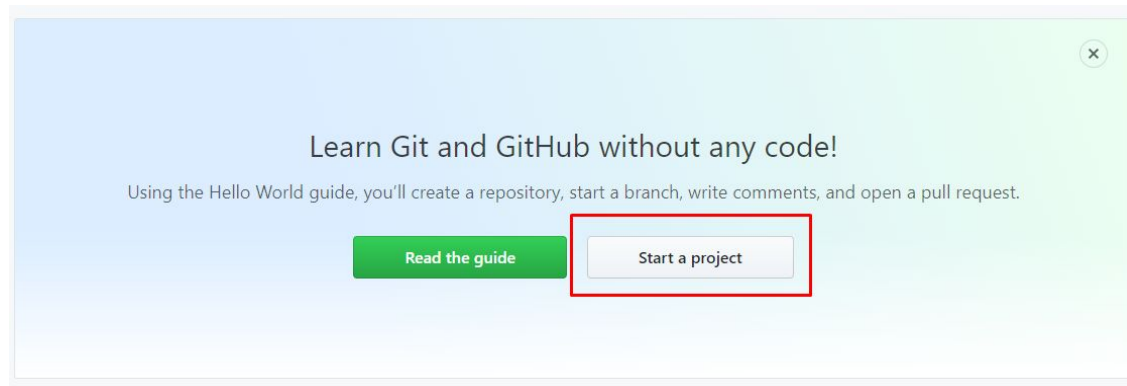
GitHub – это крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

Помимо участия в определённом проекте, GitHub позволяет пользователям общаться с единомышленниками.

# Регистрация

Все задания, которые Вы будете выполнять, должны храниться в репозитории. Это обеспечит доступ к Вашим работам людям, которые будут их проверять. Для этого Вам необходимо зарегистрироваться на портале <https://github.com/> и создать Public-репозиторий.



Owner  / Repository name \*

Great repository names are short and memorable. Need inspiration? How about [refactored-octo-invention?](#)

Description (optional)

- Public**  
Anyone can see this repository. You choose who can commit.
- Private**  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

- Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore:  Add a license:

# Клонирование репозитория

После создания репозитория в GitHub еще необходимо создать копию этого же репозитория у себя на компьютере. Для этого все клиенты, работающие с Git, предоставляют операцию клонирования. После выполнения этой операции Вы получите точную копию удаленного репозитория у себя локально. Необходимая последовательность действий представлена далее.

На странице с информацией о Вашем репозитории найдите кнопку Clone or Download, нажав на неё Вы сможете скопировать URL, который Вам понадобится для клонирования репозитория. (Рисунок 1)

Создайте на жестком диске Вашего компьютера папку, в которую будет клонироваться репозиторий. После установки TortoiseGit в контекстное меню будут интегрированы элементы для работы с Git. Нажав правой кнопкой мыши по созданной папке выберите Git Clone. Вставьте скопированный адрес репозитория в поле URL и нажмите ОК. (Рисунок 2).

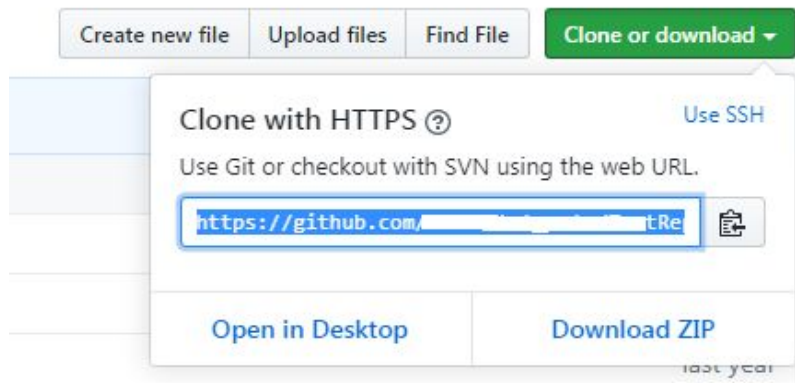


Рисунок 1. Кнопка для клонирования репозитория

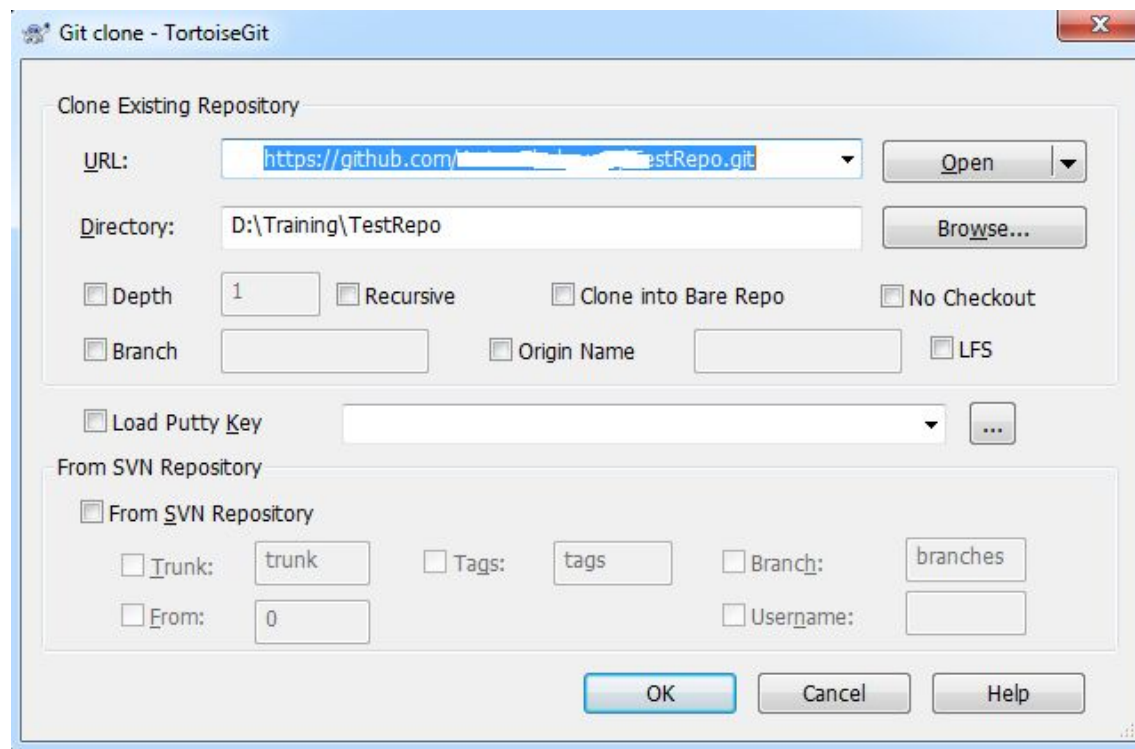


Рисунок 2. Окно TortoiseGit для клонирования репозитория



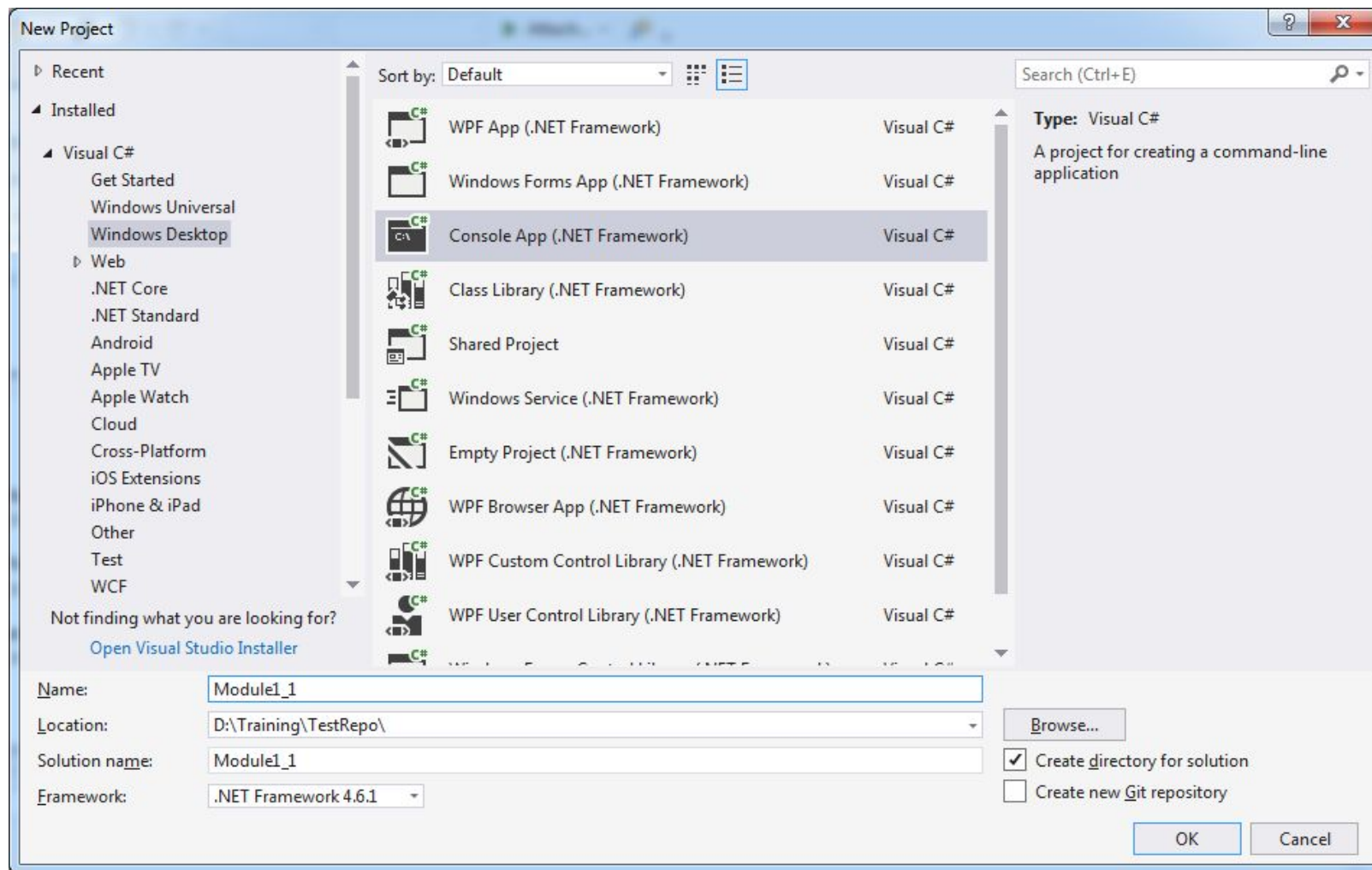
# Начало работы с локальным репозиторием

После клонирования репозитория в выбранной Вами папке создастся папка с названием эквивалентным названию Вашего репозитория. В ней необходимо проводить всю работу.

Добавьте в эту папку файл .gitignore, который Вы сможете найти в разделе Attachments в Grow. Данный файл представляет собой список путей к файлам и папкам, которые будут игнорироваться системой контроля версий. Подробнее про gitignore можно прочитать [здесь](https://git-scm.com/book/ru/v1/Ос)

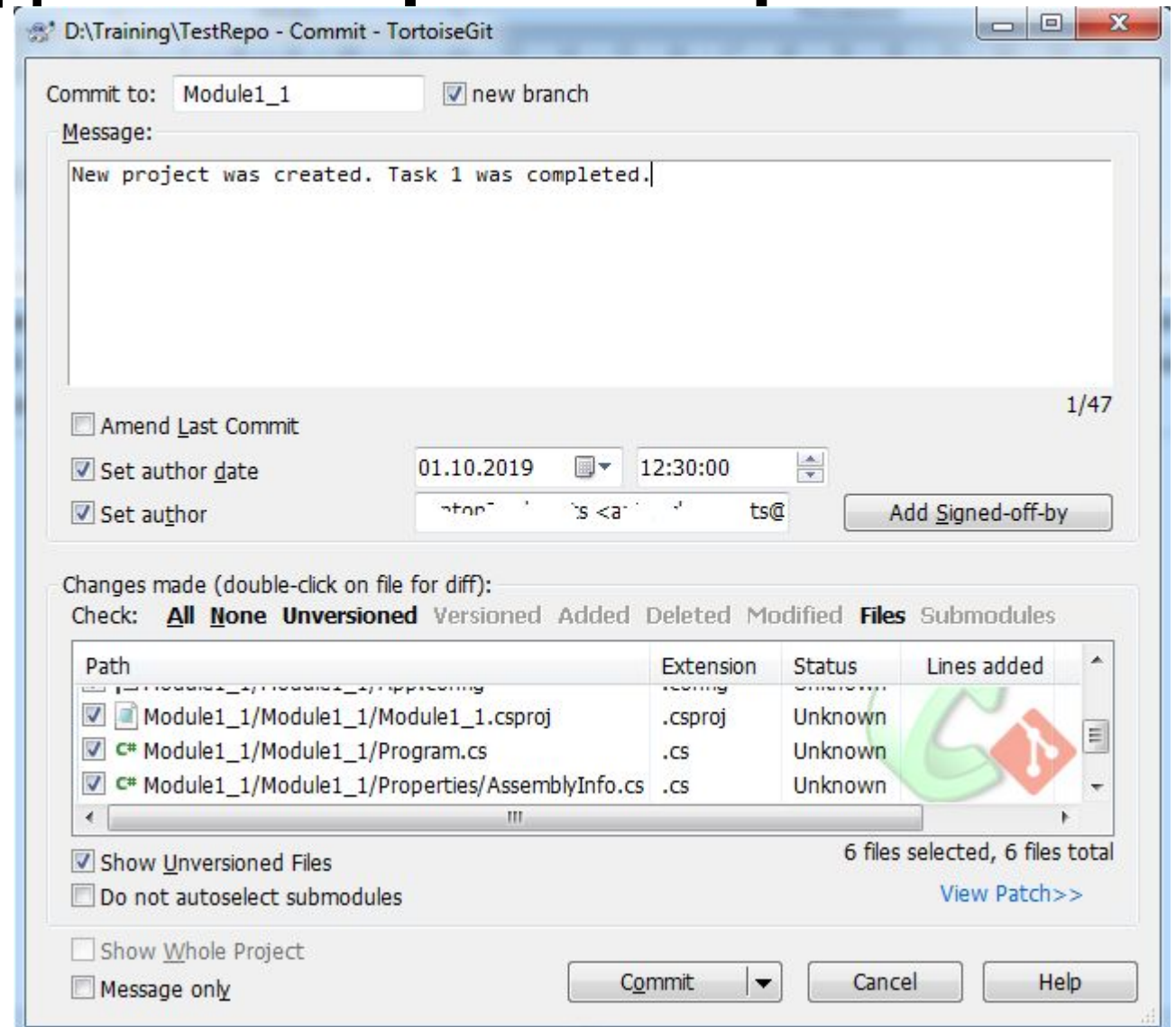
<https://git-scm.com/book/ru/v1/Ос>  
[новы-Git-Запись-изменений-в-репозиторий#Игнорирование-файлов.](#)

Создайте в этой папке консольное приложение используя Visual Studio. Обратите внимание на поля Name и Location. При именовании проектов рекомендуется придерживаться данного шаблона.



# Загрузка изменений в удаленный репозиторий

После выполнения задания из модуля 1 Вам необходимо отправить это задание на проверку. Для этого необходимо воспользоваться программой для работы с системой контроля версий. Для этого нужно нажать правой кнопкой мыши по папке с названием репозитория. После этого откроется окно, представленное на рисунке. В поле changes made отображается список новых файлов либо файлов в которых производились изменения. Обратите внимание на поле Commit to, в него необходимо ввести название ветки. Рекомендуется именовать ветки согласно указанному шаблону. В поле Message добавить комментарий к коммиту. В поле Changes made можно указать файлы, которые должны попасть в коммит. После установки автора и даты можно нажать на кнопку Commit & Push. В следующем окне в разделе Ref в поле Remote указать ветку Module1\_1. (Данной ветки нет в удалённом репозитории и она будет создана) После нажатия кнопки ОК изменения будут добавлены в удаленный репозиторий. В окне вывода результатов операции вам будет доступна ссылка на создание Pull request.

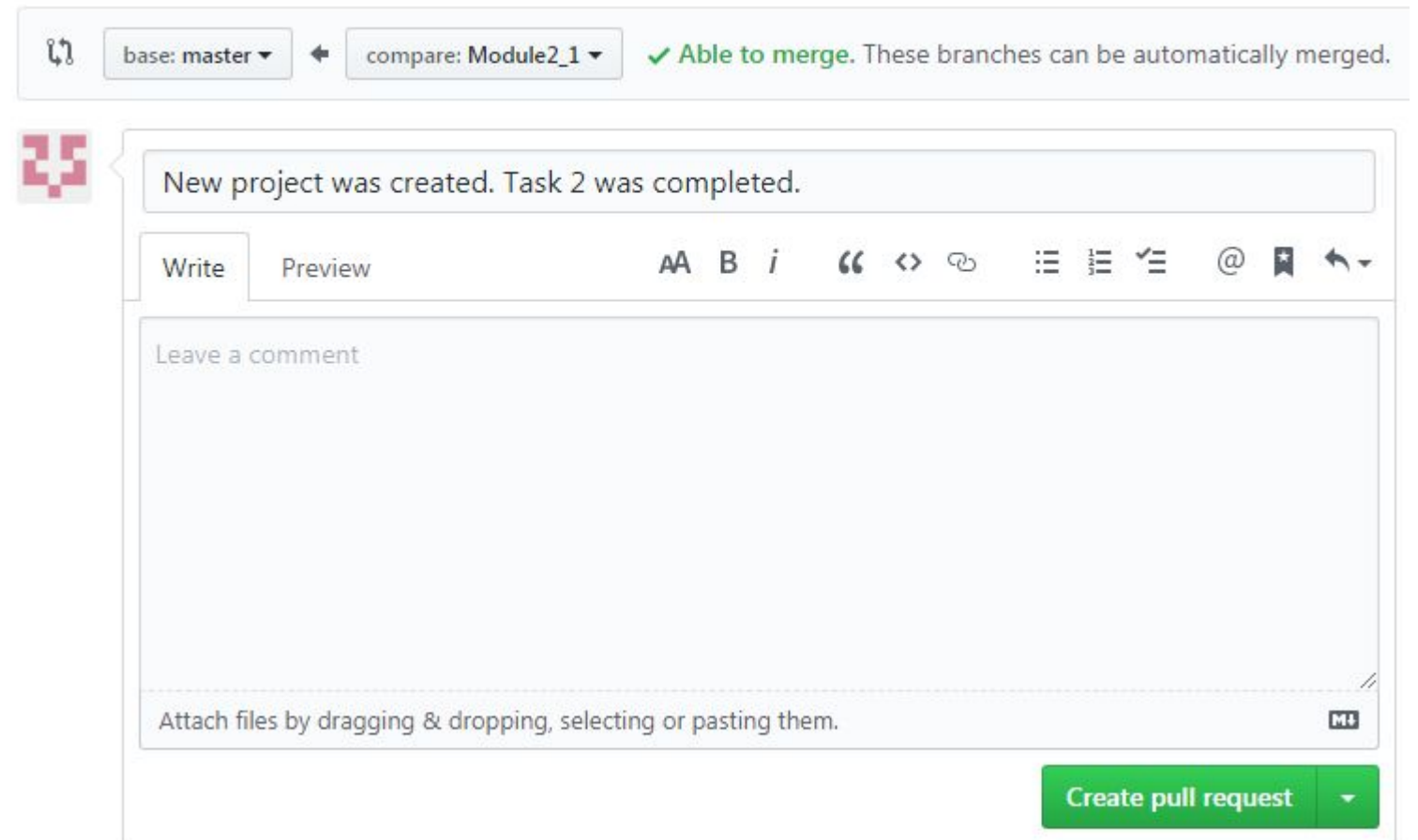


# Создание Pull Request (PR)

После перехода по ссылке для создания PR можете добавить комментарии и нажать кнопку Create pull request. Будет создан PR, ссылку на который Вам будет нужно отправить

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#)




base: master ← compare: Module2\_1 ✓ Able to merge. These branches can be automatically merged.

New project was created. Task 2 was completed.

Write Preview AA B i “ <> 🔗 ☰ ☰ ✓ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

Create pull request ▾

# Проверка задания

Возможные комментарии со стороны проверяющего можно посмотреть в разделе Conversation. После проверки задания PR будет вмержен в основную ветку (master)

New project was created. Task 2 was completed. #2

The screenshot displays a GitHub pull request interface. At the top, a green 'Open' button is visible. Below it, a navigation bar shows 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 11'. A comment box from the owner contains the text 'No description provided.' Below the comment, a commit titled 'Task1' with hash 'b3c9241' is shown. At the bottom, a green success message states: 'This branch has no conflicts with the base branch. Merging can be performed automatically.' A green button labeled 'Merge pull request' is present, along with a link to view command line instructions.