

Manual QA

Основи Git 2

Manual QA

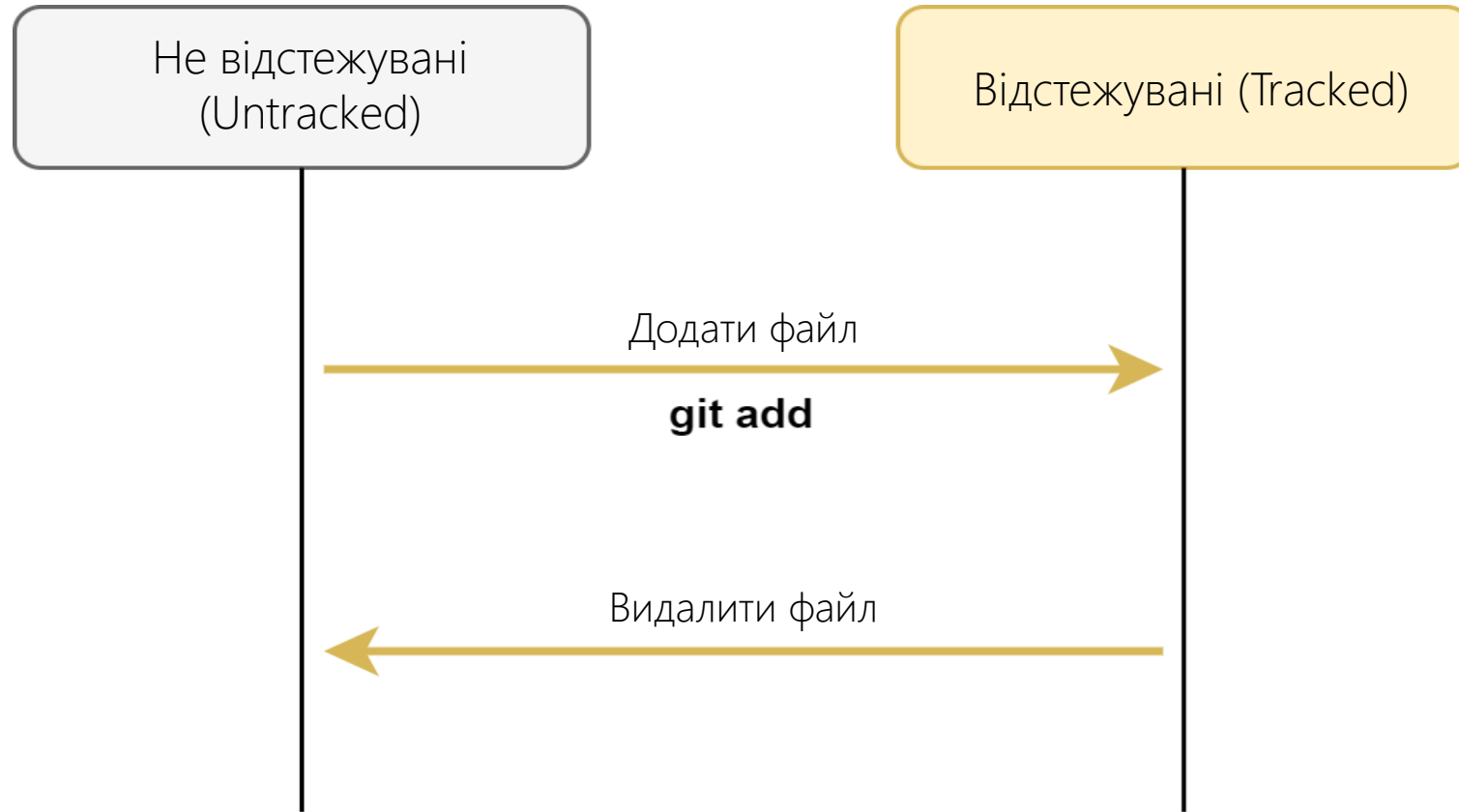
План уроку

- Робота із файлами. Стан файлів
- Віддалений репозиторій
- Розгалуження та злиття
- GitHub Flow

Робота із файлами. Стан файлів

Manual QA

Стан файлів



Manual QA

Стан файлів

```
$ git status  
On branch master
```

```
No commits yet
```

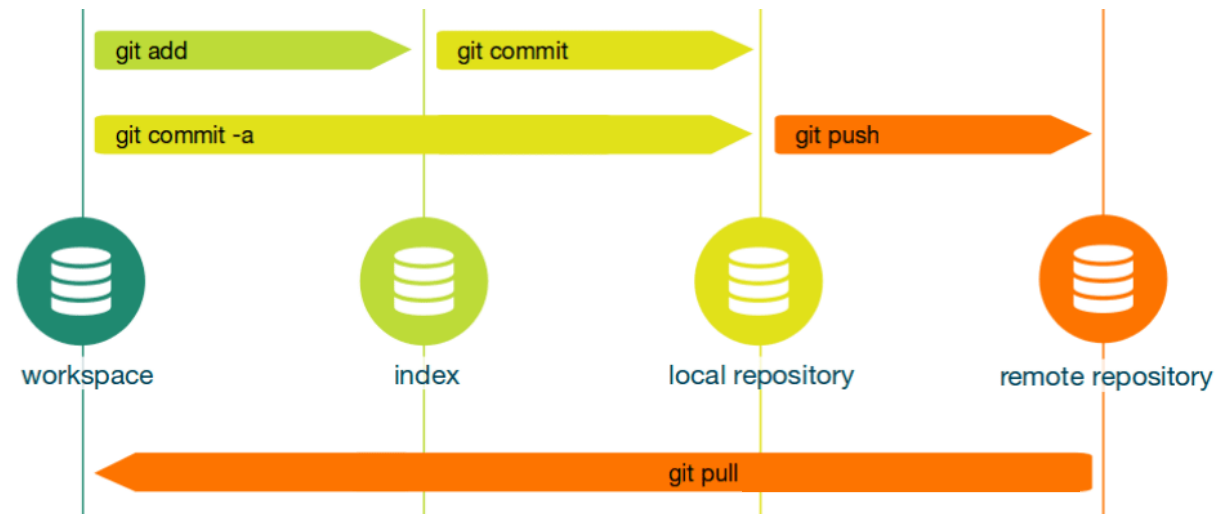
```
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    index.html
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

Manual QA

Index

Індекс — це спеціальна проміжна область, у якій зберігаються **зміни файлів** на шляху від робочої директорії до репозиторію. **Тільки файл із індексу потрапляють у коміт.**



Manual QA

git status

1. Усі файли у незміненому стані

```
$ git status
On branch master
nothing to commit, working tree clean
```

2. Файл index.html у зміненому стані

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   index.html
```

no changes added to commit (use "git add" and/or "git commit -a")

3. Файл index.html проіндексований та готовий к комміту

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index.html
```

Manual QA

.gitignore

.gitignore – файл, що дозволяє вказувати, що в репозиторії не слід відстежувати.

Що ігнорують:

- скомпільований вихідний код;
- пакети та стислі файли;
- логи;
- бази даних;
- допоміжні файли систем розробки (Visual Studio).

Manual QA

.gitignore приклад

```
# Виключити всі файли з розширення .a  
*.a
```

```
# Але відстежувати файл lib.a, навіть якщо він підпадає під виняток вище  
!lib.a
```

```
# Виключити файл TODO у кореневій директорії, але не файл у subdir/TODOTODO
```

```
# Ігнорувати всі файли в директорії build/ build/
```

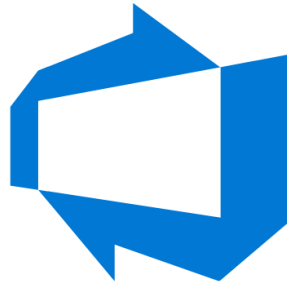
```
# Ігнорувати файл doc/notes.txt, але не doc/server/arch.txt  
doc/*.txt
```

Manual QA

Віддалений репозиторій

Manual QA

Популярні рішення



Azure DevOps



Manual QA

GitHub

Найбільший веб-сервіс для хостингу IT-проектів та їхньої спільної розробки. Веб-сервіс заснований на системі контролю версій Git та розроблений на Ruby on Rails та Erlang компанією GitHub, Inc.



Manual QA

Створення репозиторію на GitHub

Create a new repository

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

Owner *

 annmargina ▾

Repository name *

/ new_repository ✓

Great repository names are short and memorable. Need inspiration? How about **silver-rotary-phone**?

Description (optional)

new_repository description

☐



Public

Anyone on the internet 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: **None** ▾

Add a license: **None** ▾



Create repository

Manual QA

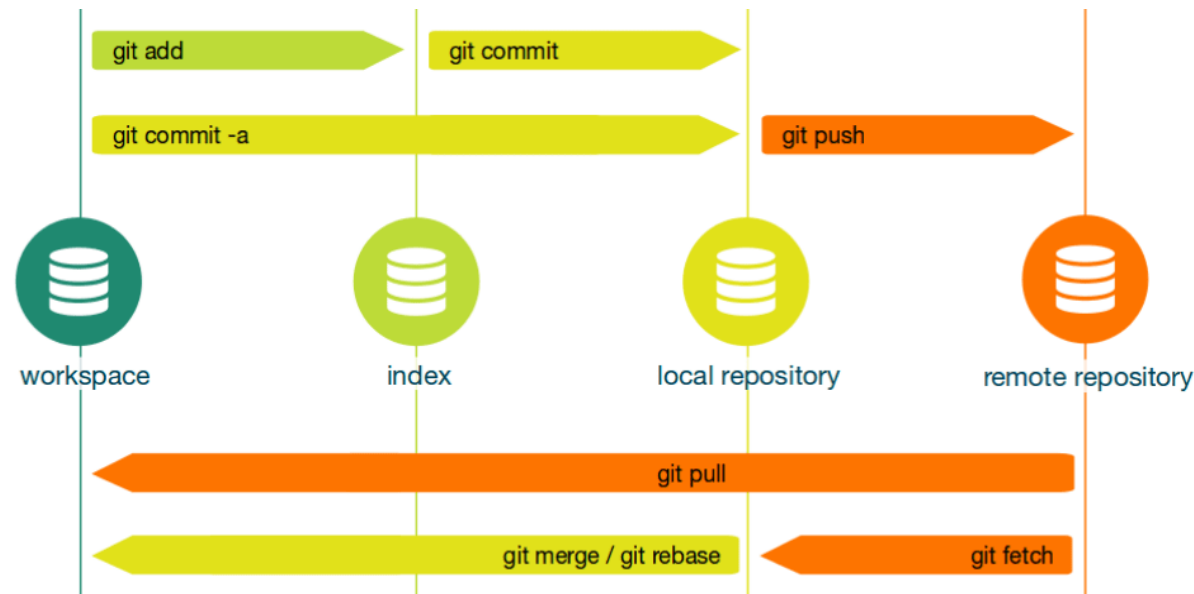
Команди

- `git clone <url>`
клонування віддаленого репозиторію.

Manual QA

Команди

- `git push <remote> <local_branch>`
відправка всіх змін до <local_branch> на віддалений репозиторій.
- `git fetch <remote>`
вилучення змін з віддаленого репозиторію до локального. Не оновлює робочу область.
- `git pull <remote>`
вилучення та інтеграція змін з віддаленого репозиторію до локального. Оновлює робочу область.

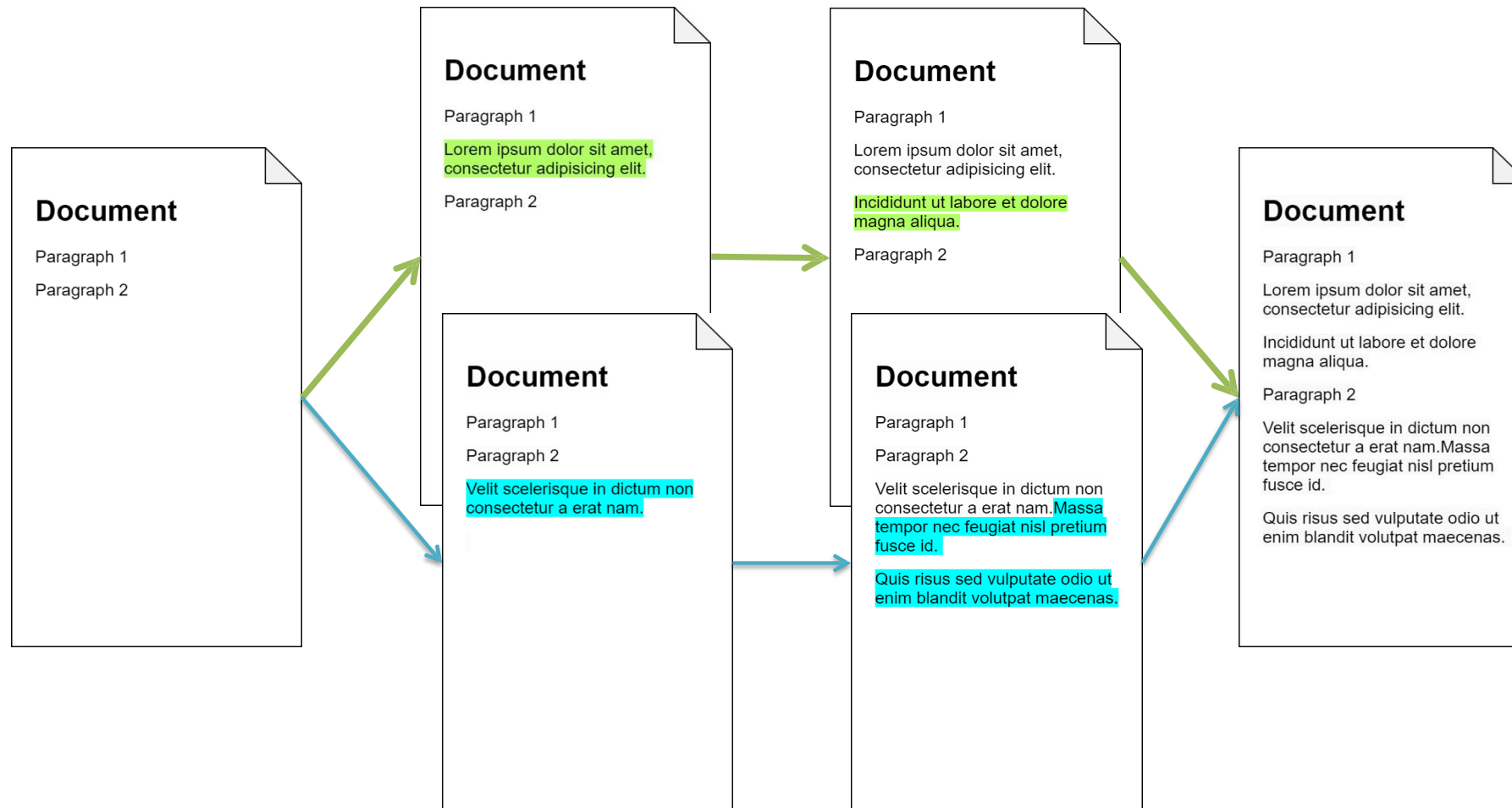


Manual QA

Розгалуження та злиття

Manual QA

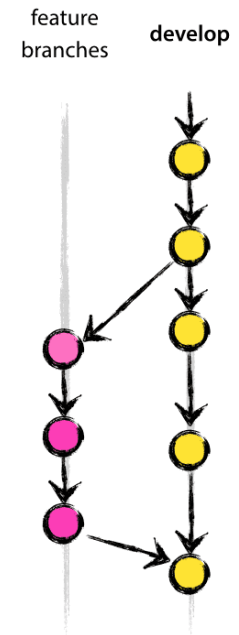
Розгалуження



Manual QA

Команди

- `git branch <name>`
Створює нову гілку, тобто новий показник на поточний коміт.
- `git checkout <branch_name>`
Перемикає на іншу гілку
- `git checkout -b <branch_name>`
Створює гілку та перемикає на неї однією командою
- `git merge <branch_name>`
Вливає цільову гілку в поточну <branch_name> - ім'я гілки зміни, яку ви хочете влити в поточну

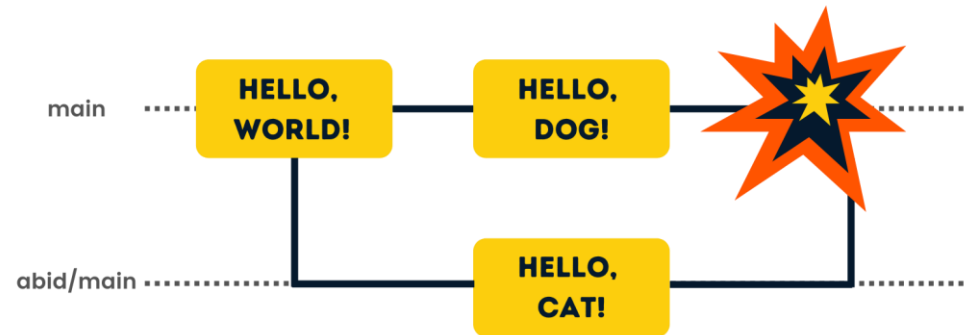


Manual QA

Конфлікти злиття

Конфлікт злиття відбувається, коли, **одну і ту ж саму** частину коду змінювали в обох гілках, що зливаються до однієї.

Якщо git самостійно не може вирішити, які зміни будуть результатом злиття – необхідно робити це вручну.



Стратегії розгалуження

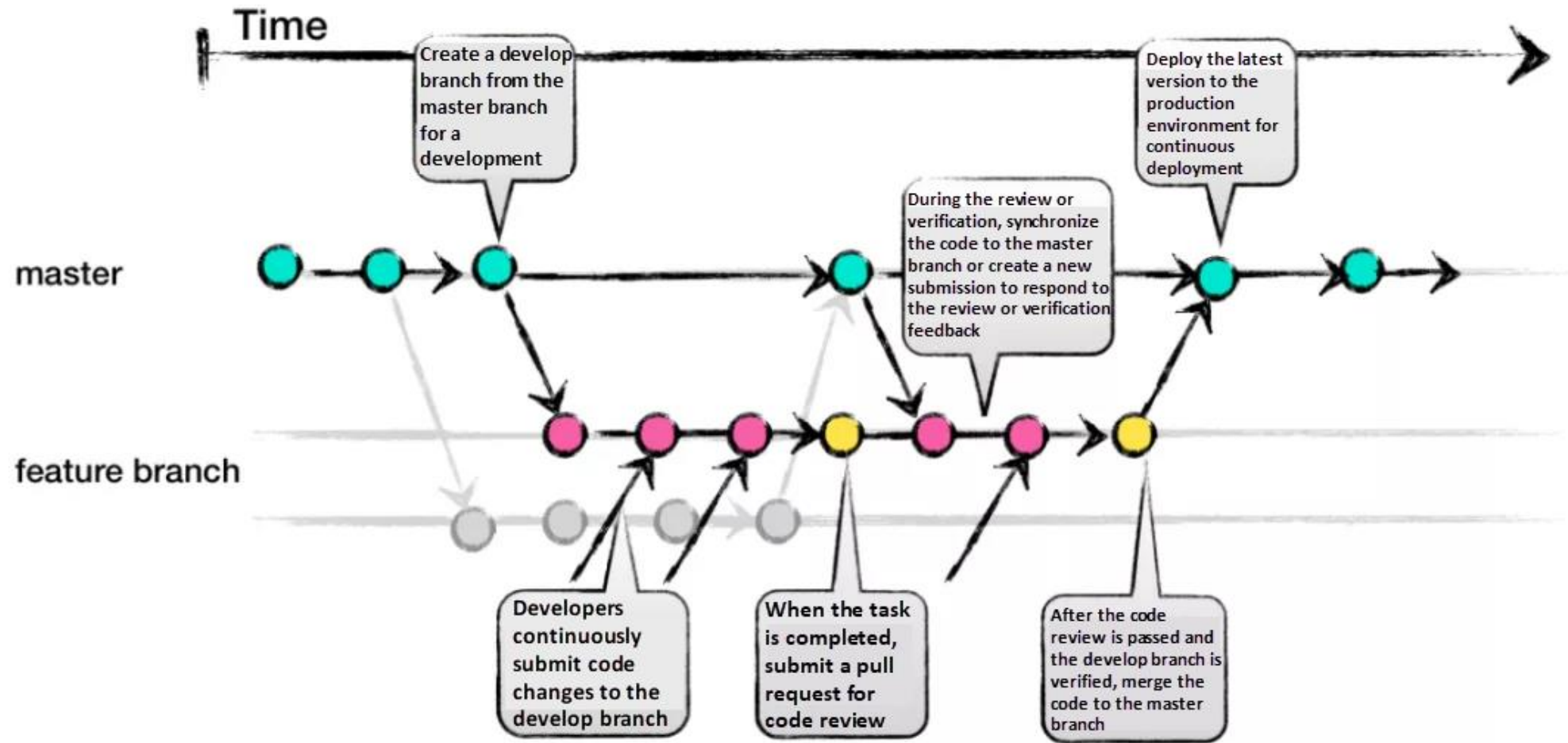
Manual QA

Стратегії розгалуження

Стратегія розгалуження – це набір правил та умов згідно з якими команда працює з git.

Manual QA

GitHub flow



Що ми сьогодні вчили

Manual QA

План уроку

- Робота із файлами. Стан файлів
- Віддалений репозиторій
- Розгалуження та злиття
- GitHub Flow

Manual QA

Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

Manual QA

Дякую за увагу! До нових зустрічей!

Інформаційний відеосервіс для розробників програмного забезпечення

