



ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

Не забыть включить запись!





Меня хорошо видно и слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

Системы контроля версий (Git)



Алена Шилина

Преподавательница курса
Middle разработчица .Net (C#)

Правила вебинара



Активно участвуем, включаем камеры



Задаем вопросы голосом или в чат



Off-topic обсуждаем в канале группы



На вопросы из чата могу ответить не сразу

Маршрут

Системы контроля версий




Git



CLI



GUI, GitHub, GitLab

The background of the slide is an aerial photograph of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network of white lines and dots, resembling a molecular or digital structure, is visible across the blue area, particularly concentrated around the text.

Работали ли вы с системами
контроля версий?

Системы контроля версий

О системе контроля версий

Что такое «система контроля версий» и почему это важно? Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии. Для контроля версий файлов в этой книге в качестве примера будет использоваться исходный код программного обеспечения, хотя на самом деле вы можете использовать контроль версий практически для любых типов файлов.

Если вы графический или web-дизайнер и хотите сохранить каждую версию изображения или макета (скорее всего, захотите), система контроля версий (далее СКВ) — как раз то, что нужно. Она позволяет вернуть файлы к состоянию, в котором они были до изменений, вернуть проект к исходному состоянию, увидеть изменения, увидеть, кто последний менял что-то и вызвал проблему, кто поставил задачу и когда и многое другое. Использование СКВ также значит в целом, что, если вы сломали что-то или потеряли файлы, вы спокойно можете всё исправить. В дополнение ко всему вы получите всё это без каких-либо дополнительных усилий.

Задачи, которые решают СКВ

- Версионирование файлов (состояние проекта)
- История изменений (состояние доработок, когда, кем, какие изменения были сделаны)
- Совместная работа

Версионирование файлов



Name

- v1.0
- v2.0
- v2.1
- v2.2


- project
- project-revised
- project-final
- project-final-for-real

Name

- project
- projectt
- projecttttttt
- aaaaaaaaaaaaaaaaaaaaaa...



История изменений

-  Removed a gitignore specifier from the .gitignore file
- Added 'Lecture 26 Files'
- Added 'Lecture 25 Tuples'
- Added 'Lecture 23 Dictionaries'
- Reorganized/Fixed all of the code/comments on all of the lectures that I have done up to this point
- Added 'Lecture 21 Lists'
- Added 'Lecture 19 Print Formatting'
- Changed the numbering of the files
- Added 'Lecture 16 Strings'
- Reorganized the repository
- Added 'Remove Numbers v.1.py'
- Added 'Lecture 14 Numbers'
- Deleted an unneeded folder

История изменений

The screenshot shows a web browser window displaying a GitHub pull request for the file `documentation/pull/3340/files`. The browser's address bar shows the URL `https://github.com/owncloud/documentation/pull/3340/files`. The page header indicates the pull request has 316 additions and 255 deletions. The diff view is split into two columns: the original code on the left and the proposed changes on the right. The left column has a pink background, and the right column has a green background. The diff shows several changes to the documentation, including updates to the FAQ section and the 'Why Did ownCloud Add Code Signing?' section. The changes are summarized in the table below:

Line	Original Code (Left)	Proposed Changes (Right)
13	past, invalid	when updating ownCloud.
14	-updates were a significant source of errors when updating ownCloud.	
15	FAQ	FAQ
16	---	---
17	Why Did ownCloud Add Code Signing?	Why Did ownCloud Add Code Signing?
18	~~~~~	~~~~~
21	-By supporting Code Signing we add another layer of security by ensuring that	+By supporting Code Signing we add another layer of security which ensures that
22	-nobody other than authorized persons can push updates for applications, and	+nobody, other than authorized individuals, can push updates for applications.
23	-ensuring proper upgrades.	+This ensures proper upgrades.
24	Do We Lock Down ownCloud?	Do We Lock Down ownCloud?
25	~~~~~	~~~~~
28	-The ownCloud project is open source and always will be. We do not want to make	+The ownCloud project is open source and always will be.
29	-it more difficult for our users to run ownCloud. Any code signing errors on	+We do not want to make it more difficult for our users to run ownCloud.
30	-upgrades will not prevent ownCloud from running, but will display a warning on	+Any code signing errors on upgrades will not prevent ownCloud from running, but will display a warning on the Admin page.
31	-the Admin page. For applications that are not tagged "Official" the code signing	+For applications that are not tagged "Official" the code signing process is optional.
32	-process is optional.	
34	Not Open Source Anymore?	Not Open Source Anymore?
35	~~~~~	~~~~~
37	-The ownCloud project is open source and always will be. The code	+The ownCloud project is open source and always will be

Совместная работа

The screenshot displays the Git Log interface in IntelliJ IDEA. The left sidebar shows the local branch '201.6668' and the remote 'origin'. The main panel shows a list of commits, with the most recent one highlighted: 'Fixed UI label text for Debugger + DFA integration' by 'anna.gasparyan' on '25/03/2020, 17:45'. The commit message includes a cherry-picked commit hash and the author's email. The right sidebar shows the files changed in the selected commit: 'intellij.java.debugger' and 'resources/messages/JavaDebuggerBundle.properties'.

Git: Log: all

Local

- 201.6668
- master
- 201

Remote

- origin

Fixed UI label text for Debugger + DFA integration

anna.gasparyan 25/03/2020, 17:45

(cherry picked from commit cfb79aec39cf4289a1459ba8ec9b8d3bc17f9e7)

25cae7cb anna.gasparyan
<anna.gasparyan@jetbrains.com> on 25/03/2020 at 17:45
committed on 25/03/2020 at 19:47

HEAD 201.6668 origin/201.6668

In 3 branches: HEAD, 201.6668, origin/201.6668

Почему это важно

- Позволяет команде разработчиков работать над одним и тем же проектом
- Минимизирует конфликты в разработке
- Автоматически создает архив каждой версии проекта, включающий в себя все изменения

Системы контроля версий



Git

- <https://git-scm.com/>
- <https://git-scm.com/downloads>
- <https://git-scm.com/docs>
- <https://git-scm.com/book/ru/v2>
- <https://en.wikipedia.org/wiki/Git>
- <https://habr.com/ru/companies/ruvds/articles/599929/>
- <https://www.atlassian.com/ru/git/glossary#commands>



<https://selectel.ru/blog/tutorials/how-to-install-git-to-windows/>

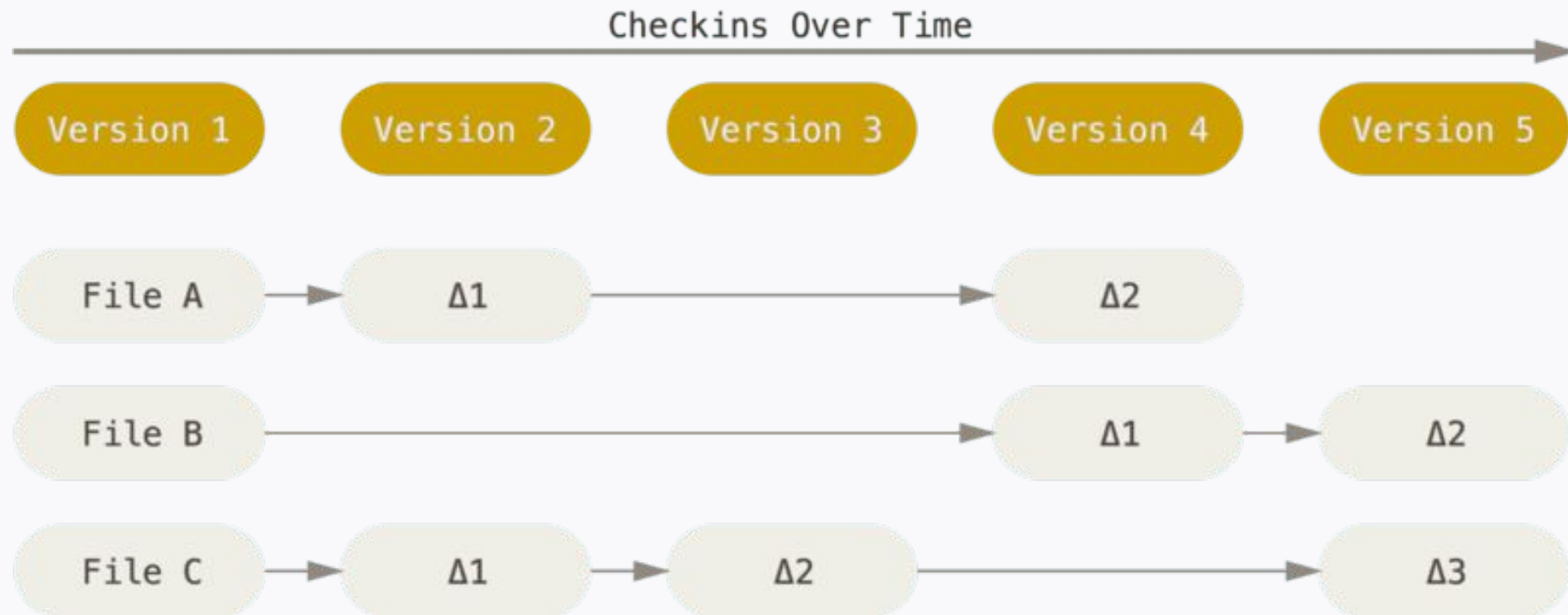
Git – это **распределенная** система версий. Системы контроля версий бывают локальными, централизованными или распределенными.

- **Локальная система хранит файлы на одном устройстве.** В ней удобно работать с большими проектами, но сложно взаимодействовать с удаленной командой.
- **Централизованная использует общий сервер.** В такой системе налажена удалённая работа, но всё привязано к одному серверу. Любой сбой или взлом может повредить файлы проекта.
- **Распределенная использует общее облачное хранилище и локальные устройства участников команды.** В такой системе также налажена удаленная работа, но если с файлами основного репозитория что-то случится – проект легко восстановить из копии любого участника команды.

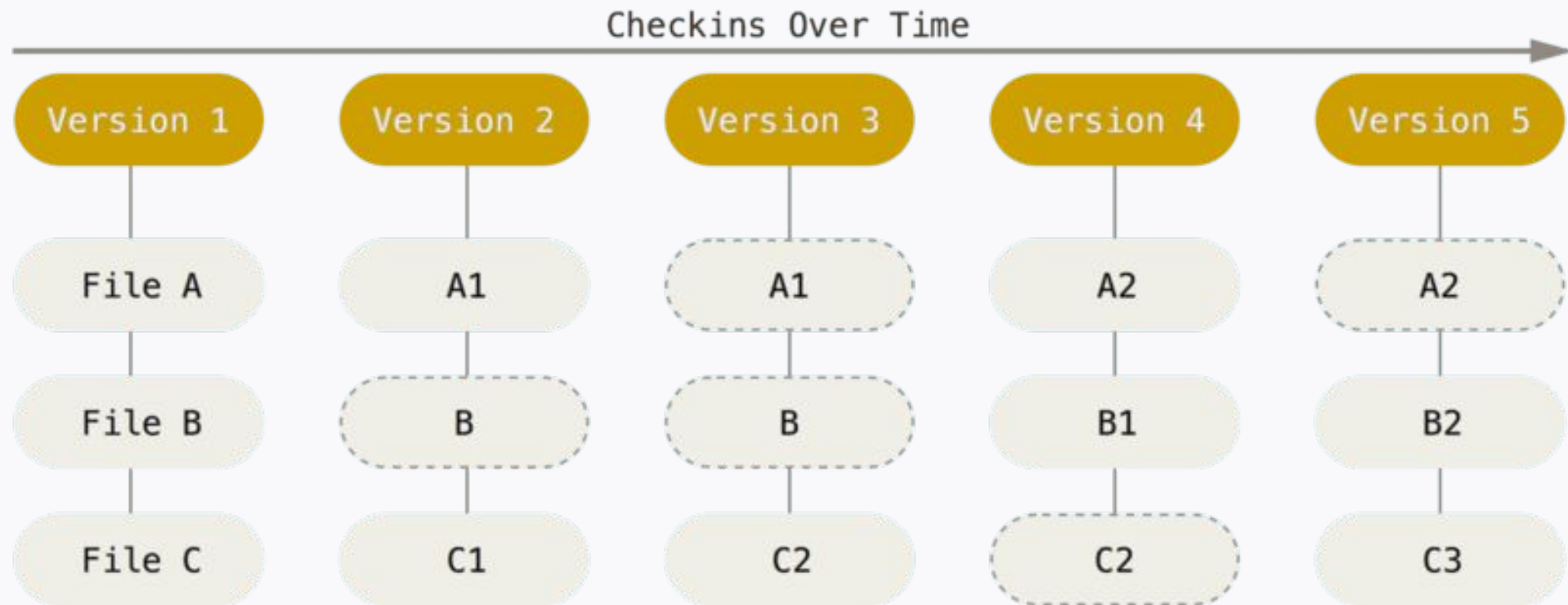
Что дает Git

- Скорость
- Простая архитектура
- Хорошая поддержка нелинейной разработки
- Полная децентрализация
- Возможность эффективного управления большими проектами

Устройство других СКВ



Снимки, а не различия





Репозиторий




Материал из Википедии — свободной энциклопедии

[\[править \]](#) | [\[править код \]](#)

Репозитóрий^[1] (от *англ.* *repository* — хранилище) — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по *сети*.

.gitignore и dotnet new gitignore

 .gitignore  807 B

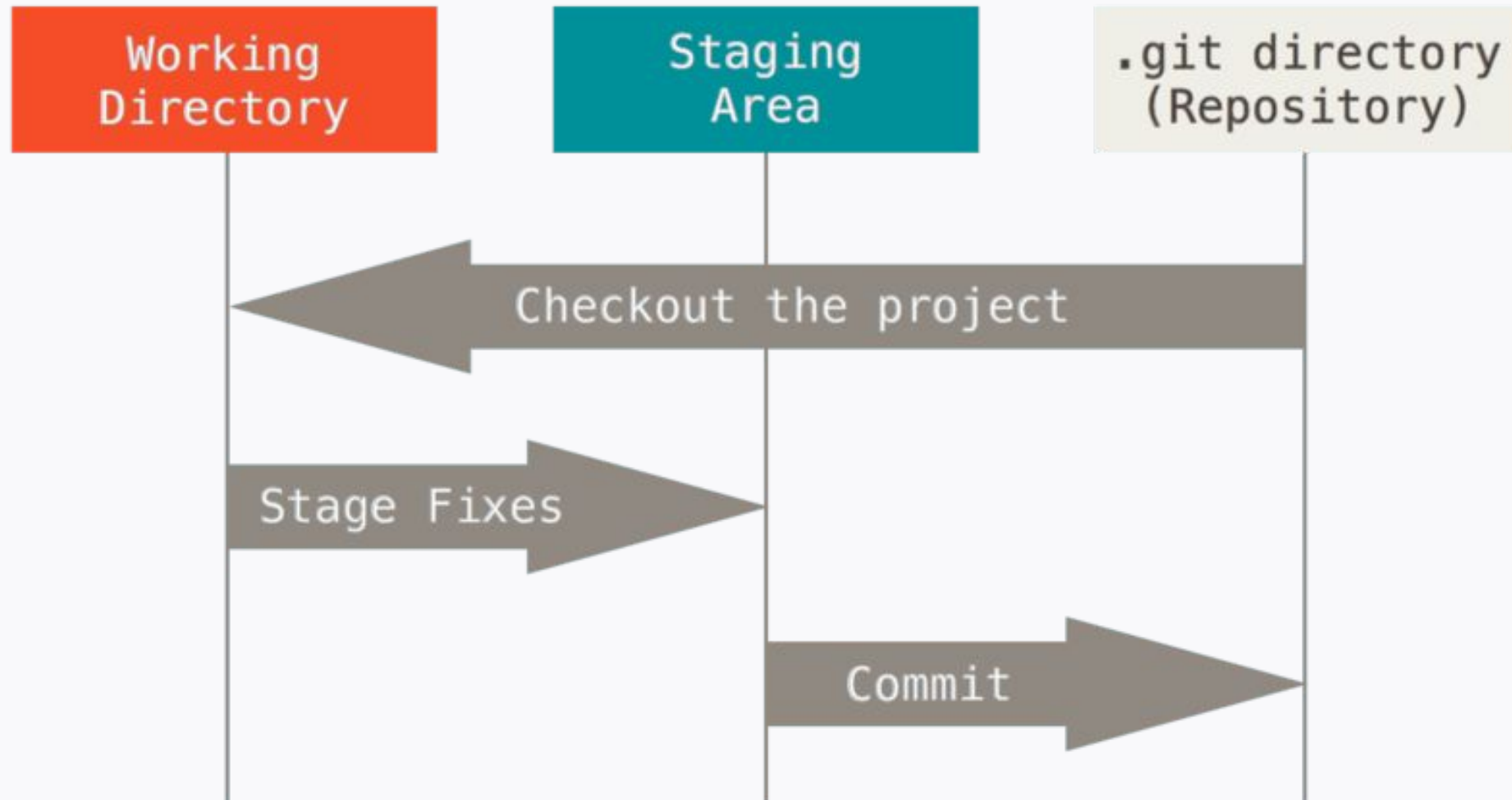
Blame Edit ▾ Replace Delete   

```
1  ## Ignore Visual Studio temporary files, build results, and
2  ## files generated by popular Visual Studio add-ons.
3
4  # User-specific files
5  *.suo
6  *.userosscache
7  *.sln.docstates
8  *.csproj.user
9  *.DotSettings.user
10
11 # Build results
12 [Dd]ebug/
13 [Dd]ebugPublic/
14 [Rr]elease/
15 [Rr]eleases/
16 bin/
17 obj/
18 x86/
19 x64/
20
21 # Visual Studio 2015 cache/options directory
22 .vs/
23
24 # MSTest test Results
25 [Tt]est[Rr]esult*/
26 [Bb]uild[Ll]og.*
27
```

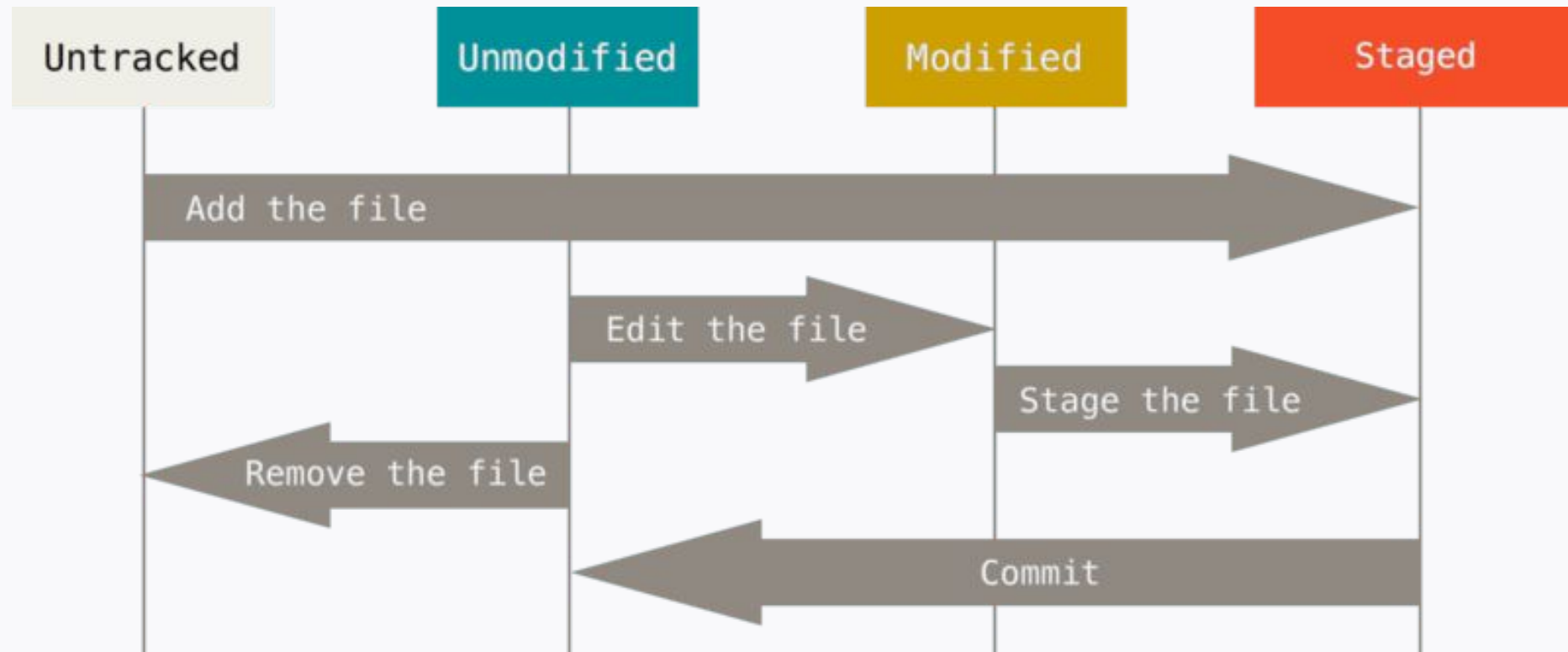
Основные состояния файла

- **Untracked** – вообще не отслеживаем файл
- **Unmodified** – вообще не измененный файл
- **Modified** (изменен) – файл, которые изменился, но еще не был зафиксирован
- **Staged** (индексирован) – измененный файл в его текущей версии, отмеченный для включения в следующий коммит
- **Committed** (зафиксирован) – файл уже сохранен в вашем локальном репозитории

Секции проекта



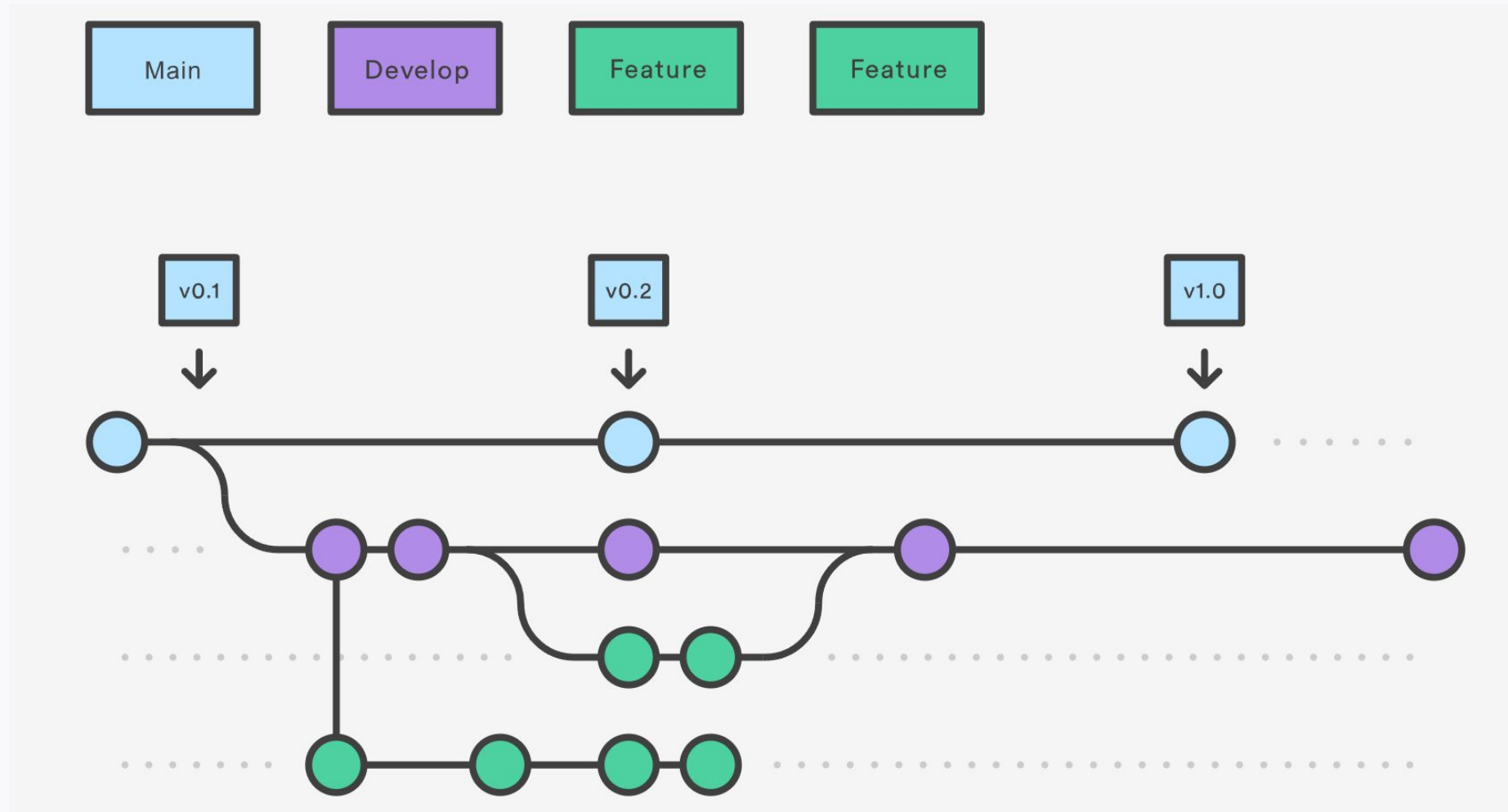
Состояния файлов



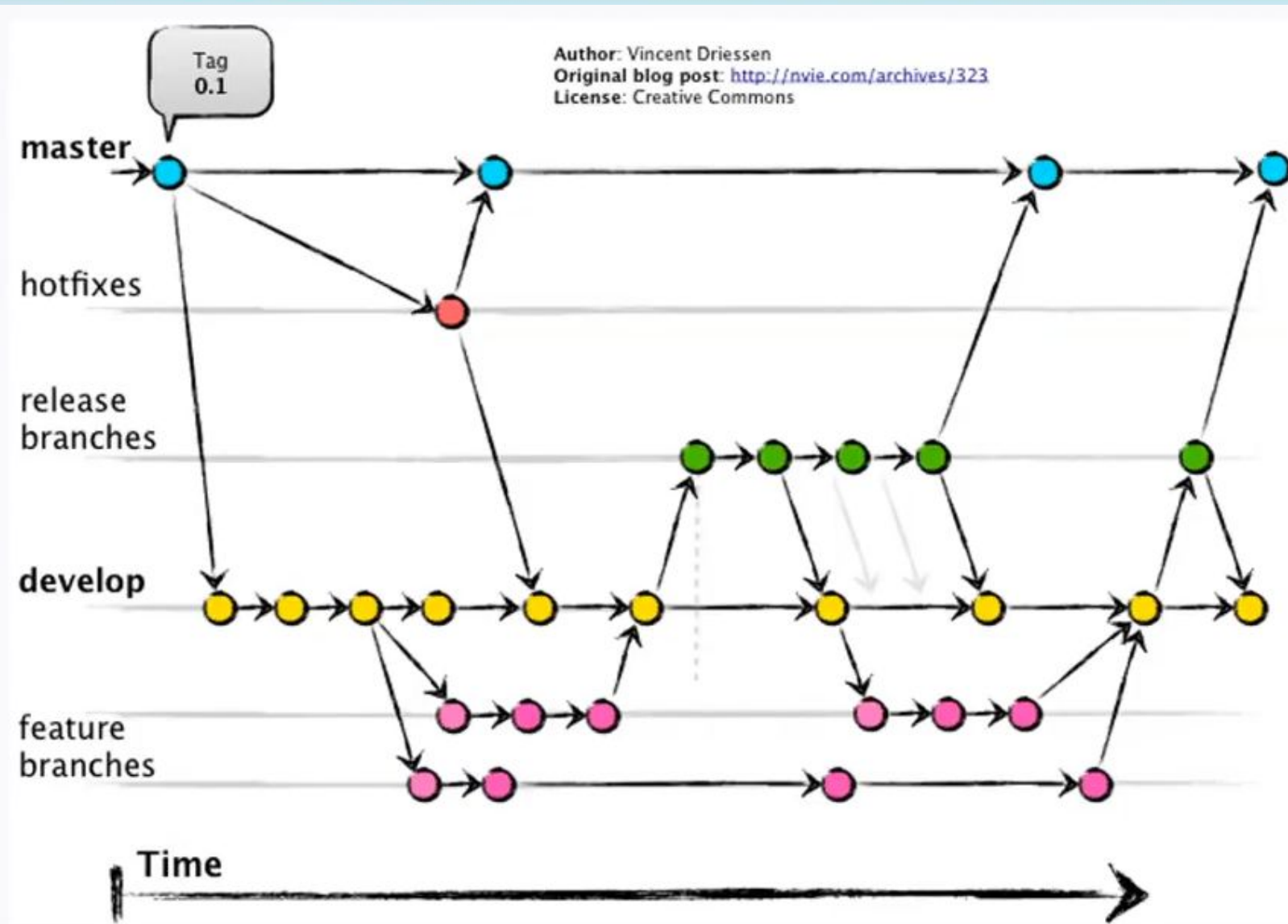
The image features a high-angle, aerial view of a city skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is rendered in a monochromatic blue color scheme. A semi-transparent network pattern, consisting of interconnected dots and lines, is overlaid on the image, particularly visible in the central area where the text is located. The word "DEMO" is prominently displayed in the center in a large, white, sans-serif font.

DEMO

Ветки и GitFlow



Ветки и GitFlow



Хостинг репозиториев и GUI



Github



Bitbucket



Gitlab

[Сравнение хостингов для проектов свободного программного обеспечения](https://git-scm.com/downloads/guis)

<https://git-scm.com/downloads/guis>

The image features a high-angle, aerial view of a dense urban landscape, likely New York City, characterized by numerous skyscrapers and buildings. The entire scene is rendered in a monochromatic blue and green color palette. A semi-transparent network of white lines, resembling a digital or data connection map, is overlaid on the image, particularly concentrated around the central text. The word "DEMO" is prominently displayed in the center in a large, white, sans-serif font.


DEMO

Рефлексия

- git init
- git clone <>
- git status
- git log
- git add
- git commit (-m "<>", --amend)
- git fetch / git pull
- git push (-u origin <>)
- git branch
- git checkout (-b) <>
- git merge <>
- git revert

The background of the slide is a blue-tinted aerial photograph of a dense city skyline, likely New York City. Overlaid on this image is a semi-transparent network pattern consisting of numerous small dots connected by thin lines, creating a web-like structure. The text "Вопросы?" is centered in the middle of the slide in a large, white, sans-serif font.

Вопросы?

The background of the slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or technological aesthetic. The text is centered in white, bold, sans-serif font.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за занятие!



Алена Шилина

Преподавательница курса
Middle разработчица .Net (C#)