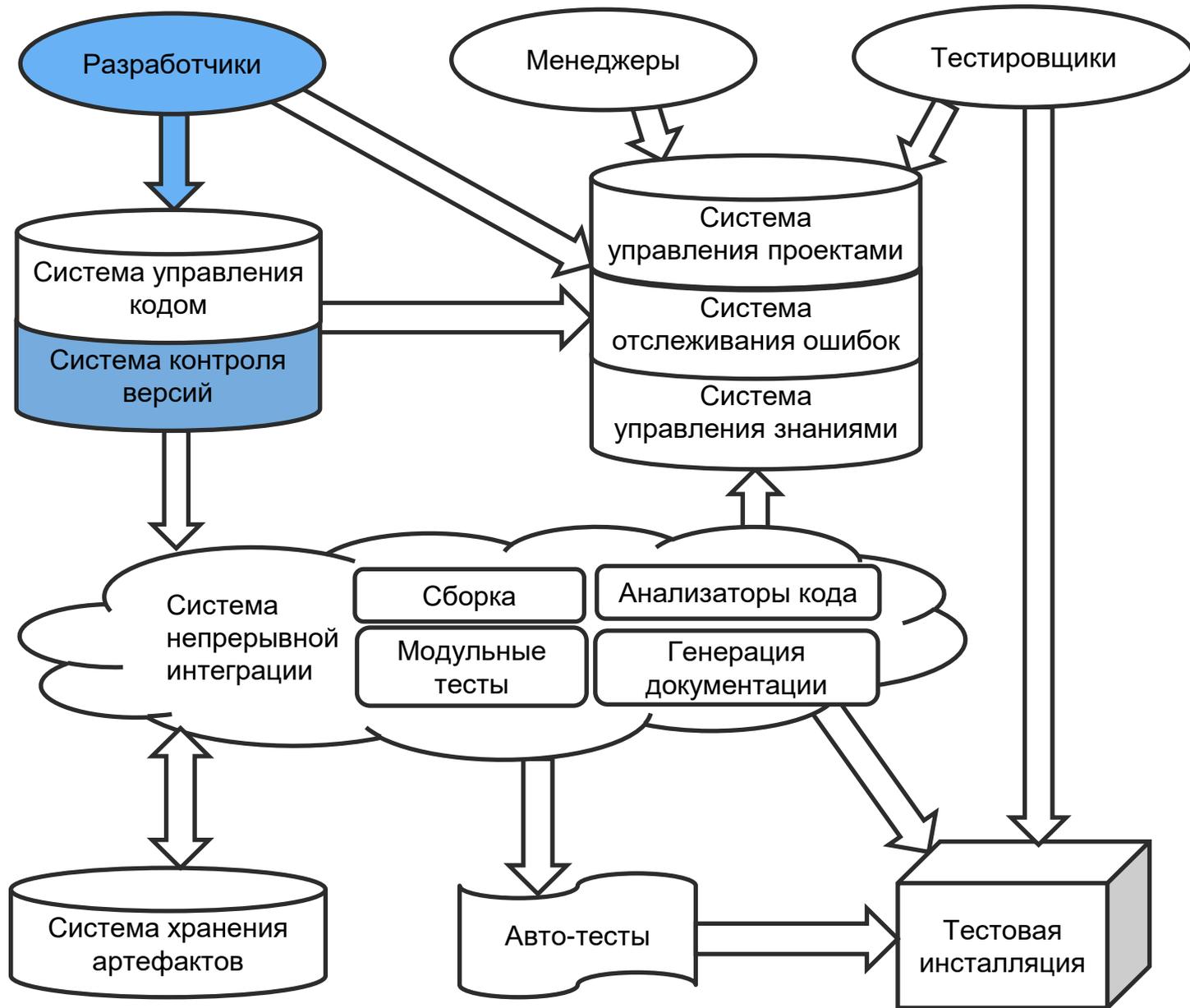


Управление производственным процессом разработки программного обеспечения

Системы контроля версий

План

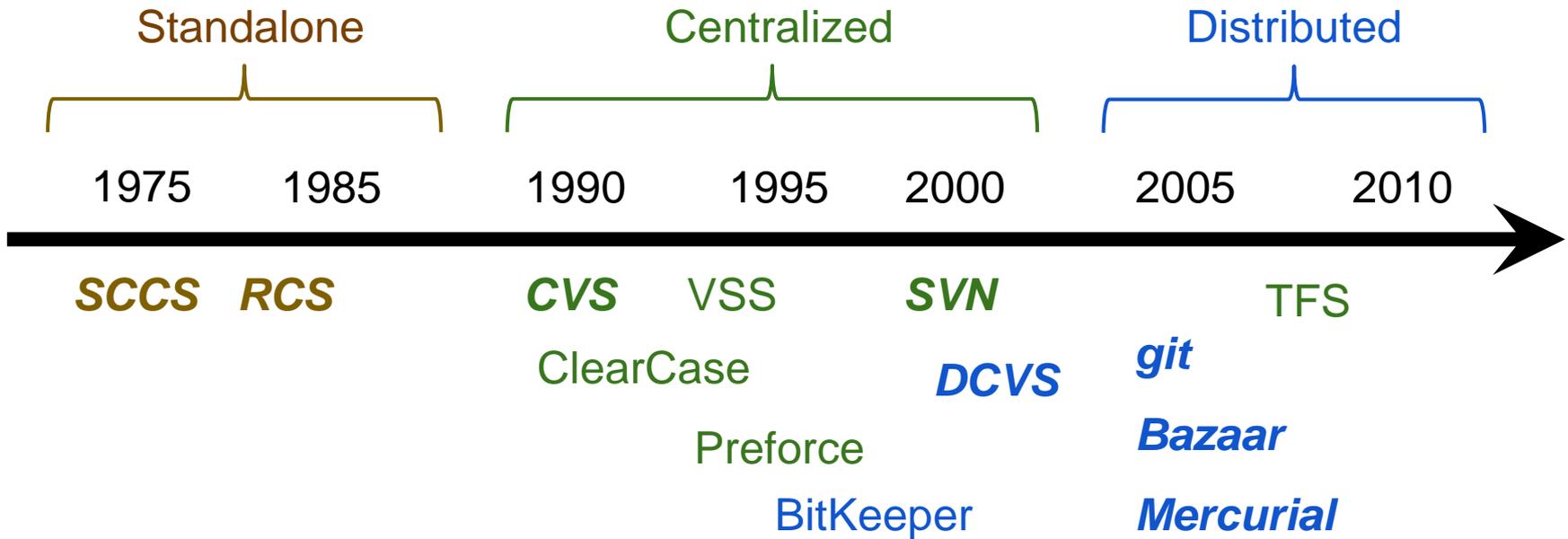
- Что такое системы контроля версий
- Основные команды
- Редко используемые возможности
- Системы контроля версий в реальных проектах



Что такое VCS

- Системы контроля версий. Version Control System, часто встречается название Revision Control System
- VCS позволяют хранить несколько версий одного и того же файла, возвращаться к более ранним версиям, отслеживать изменения.
- Версией или ревизией (revision) называется конкретное зафиксированное состояние репозитория

Основные VCS



Жирным выделены Open Source VCS

Контроль версий вручную

- Я хочу внести изменения в код, но не хочу чтобы они сломали все остальное
- Я хочу поэкспериментировать без риска все испортить
- Нужно найти что изменилось по сравнению с прошлой неделей
- В новой версии баг, нужно откатиться на предыдущую версию в которой бага нет
- etc.

Изготовим RCS из подручных материалов.

diff & patch

```
$cat students_1.txt
```

Иванов И.

Петров П.

Семафоров С.

Васечкин В.

```
$cat students_2.txt
```

Иванов И.

Шляпкина А.

Петров П.

Семафоров С.

Васильев В.

```
$diff students_1.txt students_2.txt
```

1a2

> Шляпкина А.

4c5

< Васечкин В.

> Васильев В.

```
$patch students_1.txt <students_1_2.patch
```

```
$cat students_1.txt
```

Иванов И.

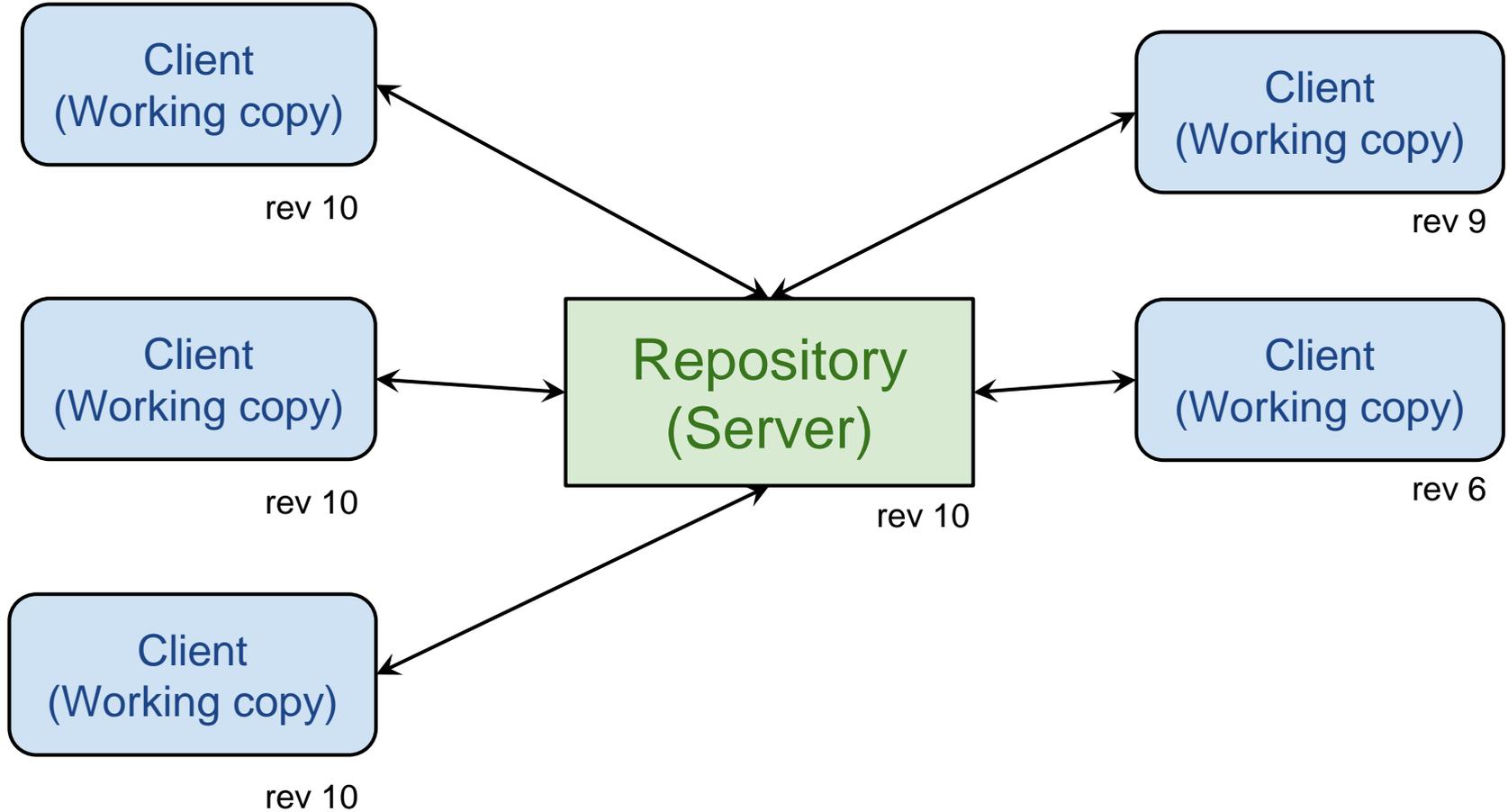
Шляпкина А.

Петров П.

Семафоров С.

Васильев В.

Centralized VCS



Достоинства и недостатки централизации

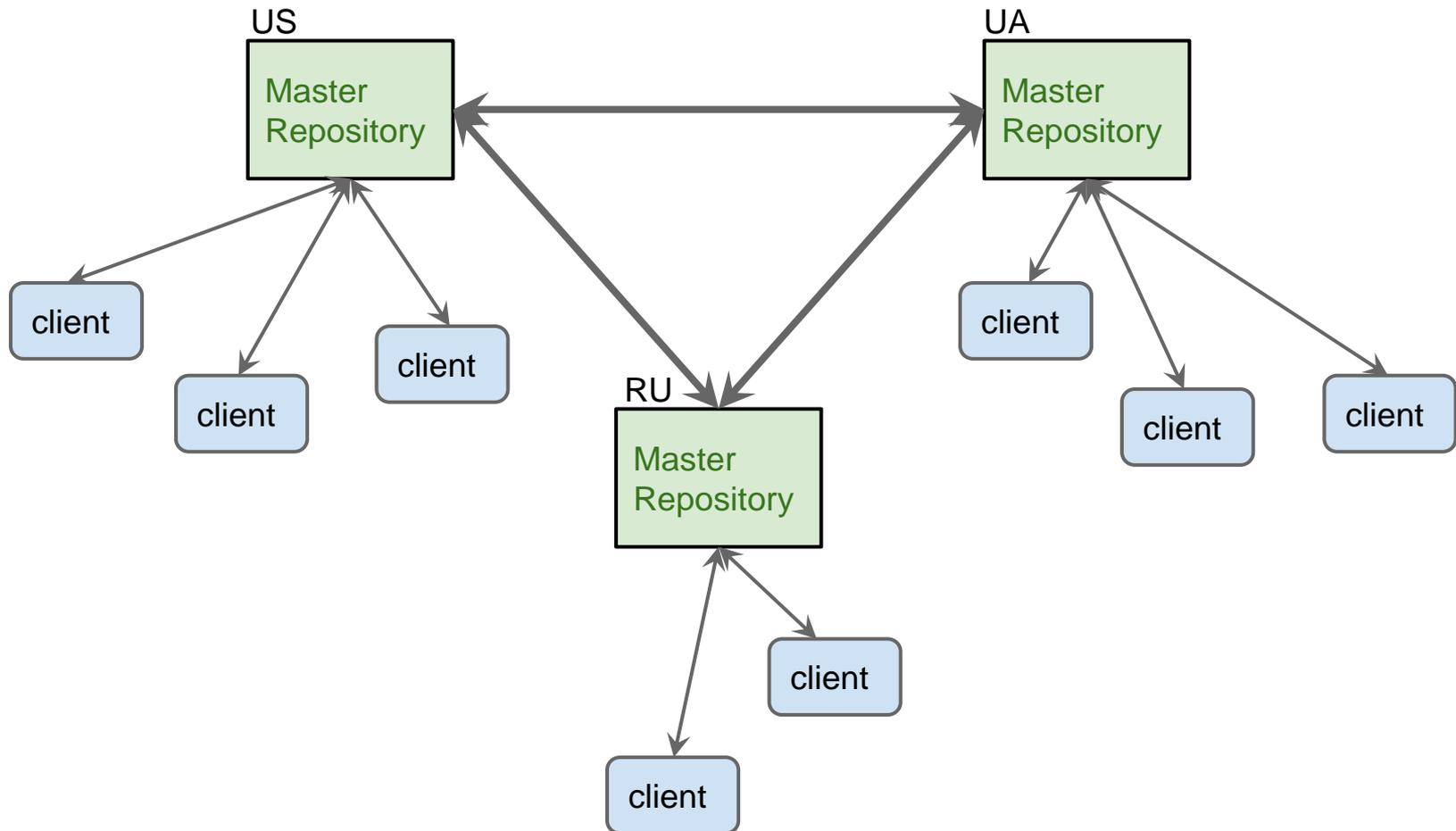
Достоинства:

- Всегда есть выделенная "эталонная" версия кода
- Централизованное администрирование
- Привычный workflow

Недостатки:

- Единая точка отказа — сервер
- Любые изменения влияют на всех пользователей
- Требуется достаточно гибкая система прав
- В очень больших или распределенных проектах централизованные механизмы работают плохо.
- Репозитории уровня корпорации трудно администрировать

Master-master репликация



Master-master репликация

Появилась довольно давно как надстройка над централизованными системами контроля версий (DCVS, SVK).

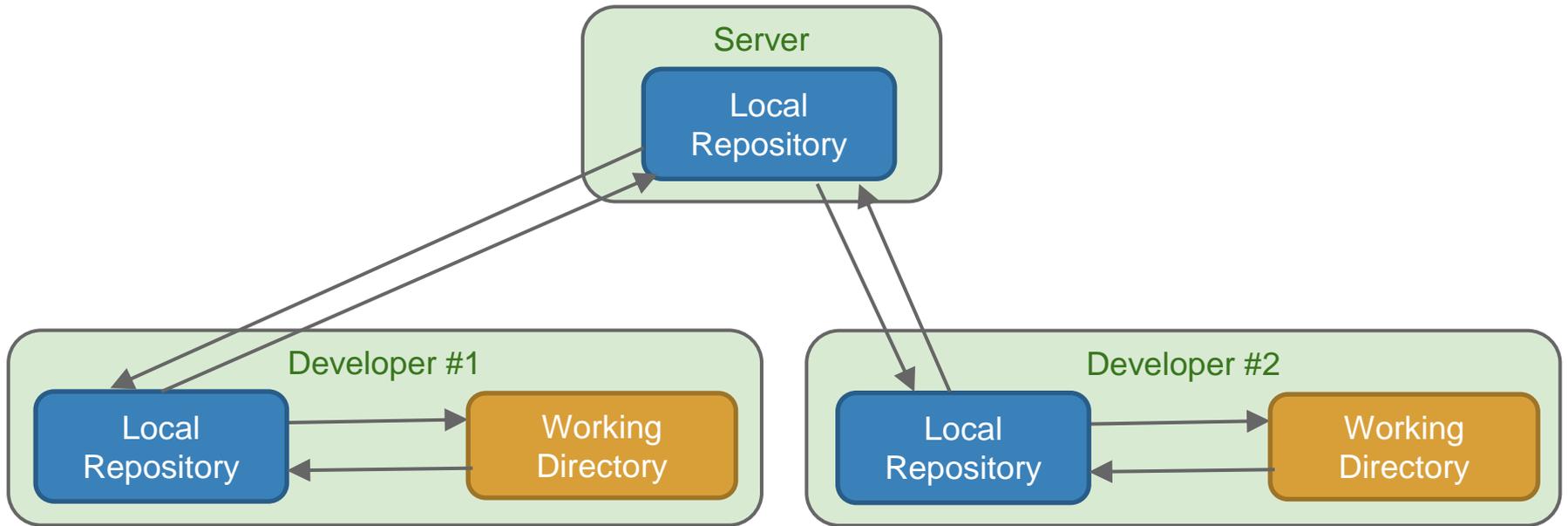
Работает, но не решает всех проблем:

- довольно сложно в администрировании
- все еще возможны конфликты правок
- все еще требуется связь с локальным сервером

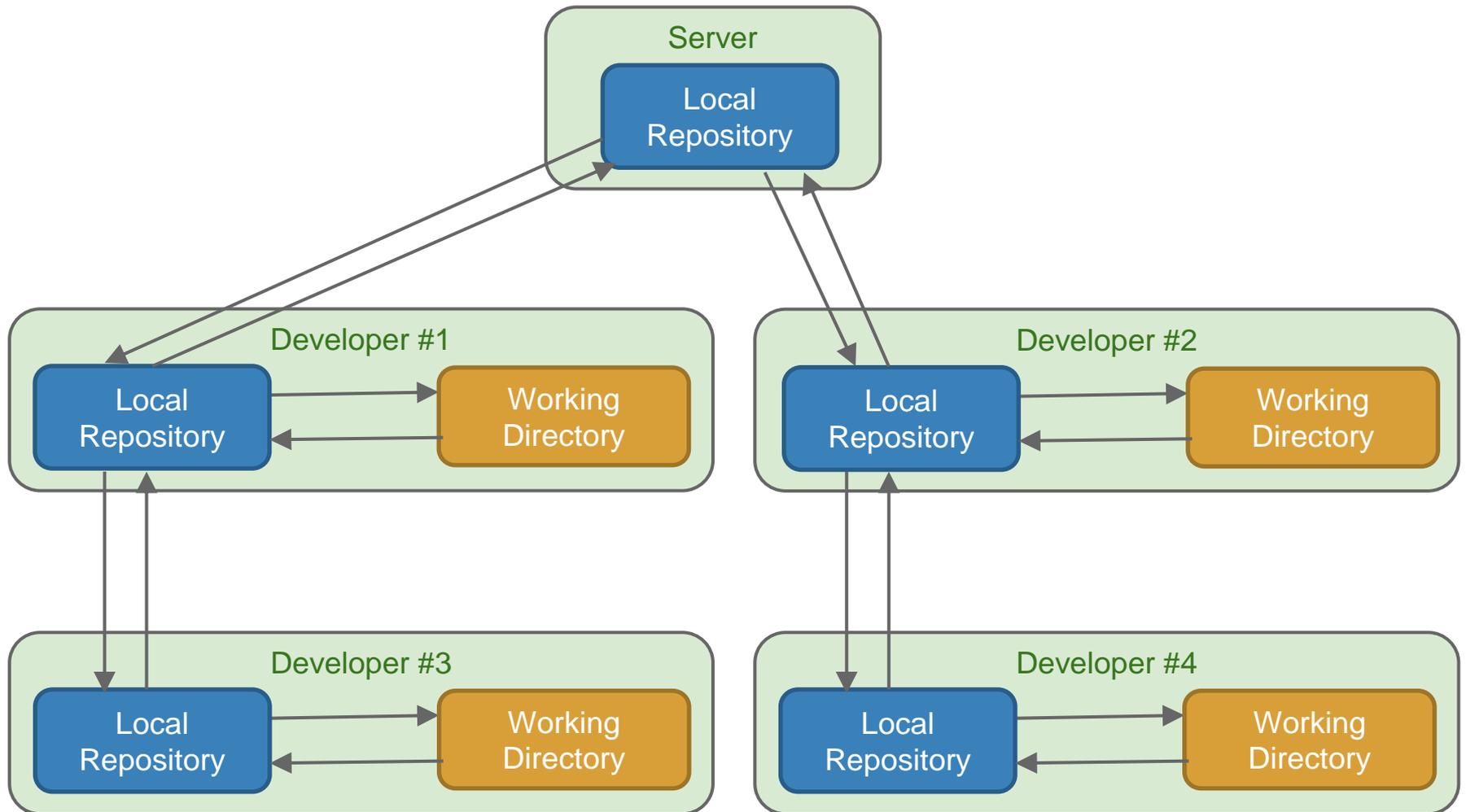
Мотивация проекта Linux kernel

- Очень большое число участников
- Сложности с управлением правами
- Собственные репозитории у участников проекта (таких как RedHat или Alt Linux)
- Очень большой объем исходного кода
- Сборка артефактов отделена от разработки

Distributed workflow

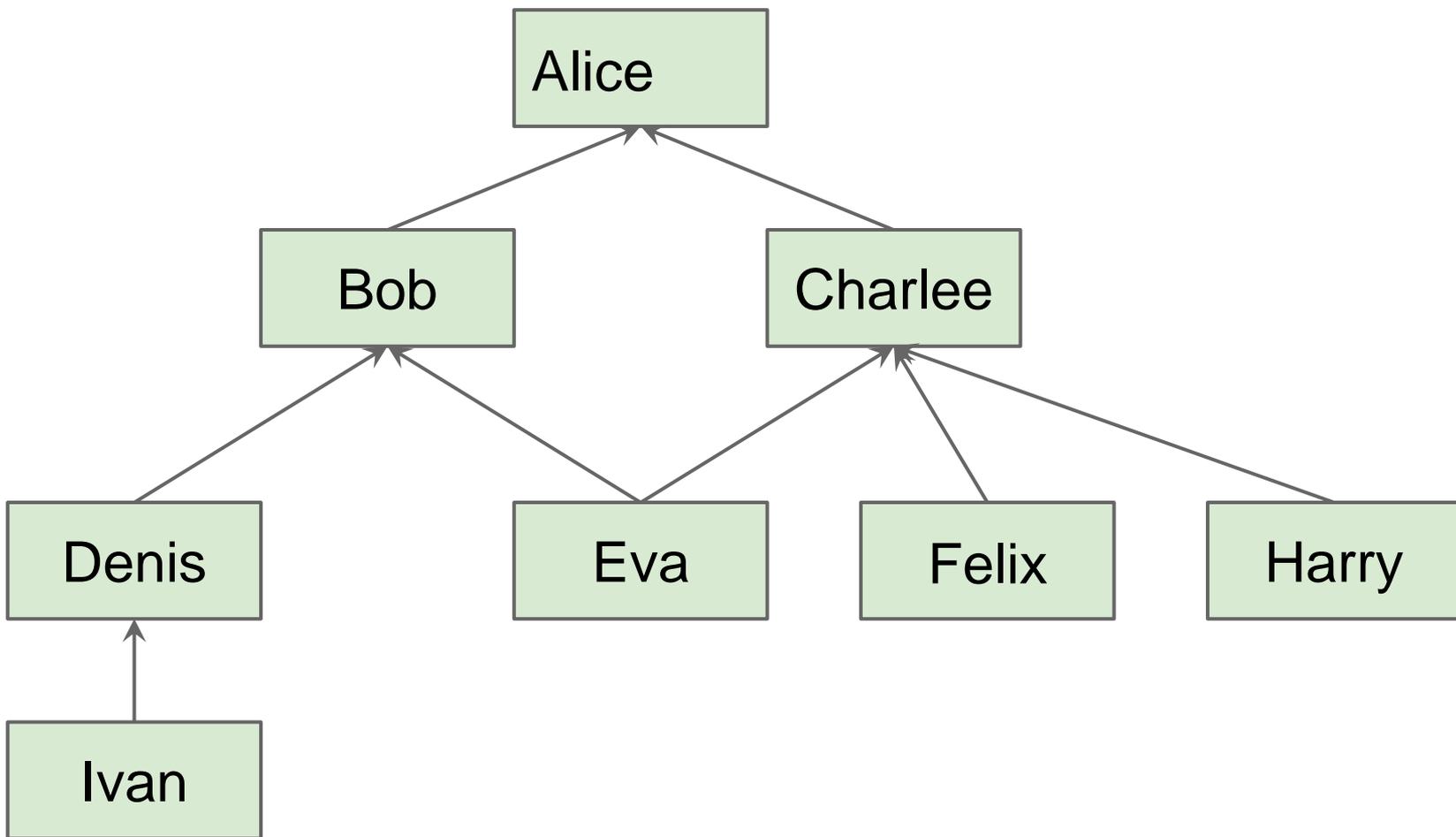


Distributed workflow

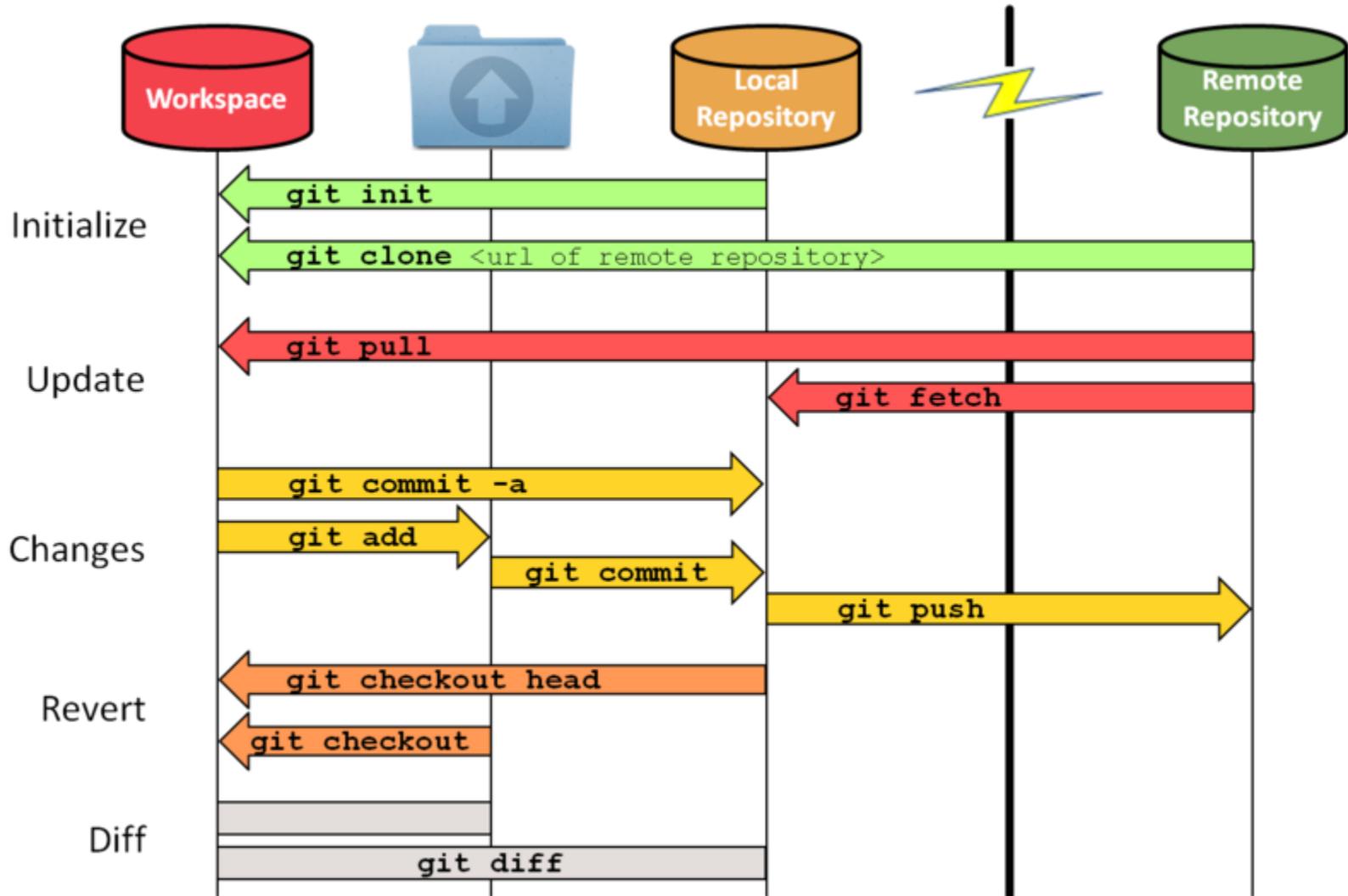


Сеть доверия

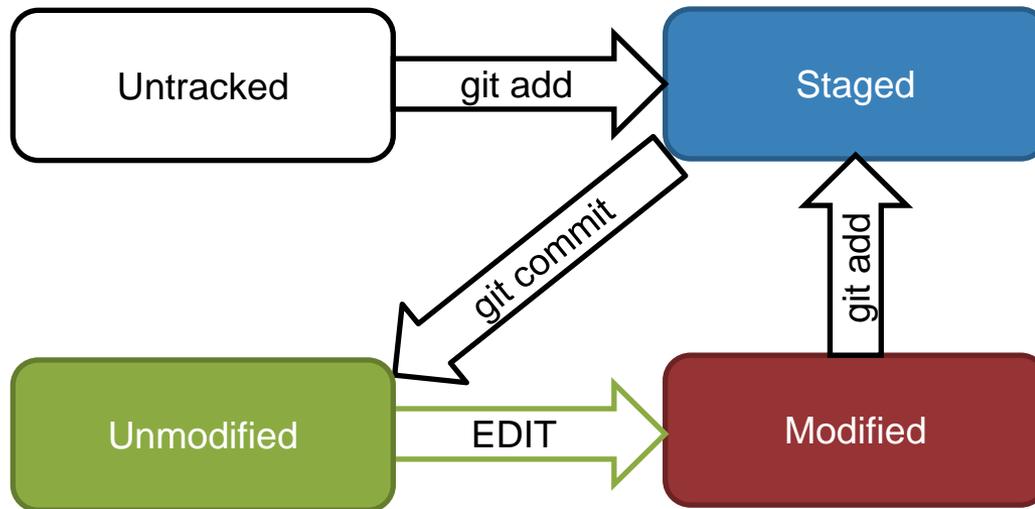
Trust network



Жизненный цикл git



Статус файла



Работа с индексом (stage)

Команда **add** добавляет измененные файлы в stage.

```
$git add filename
```

Команда **rm** помечает файл в stage как удаленный.

```
$git rm filename
```

Команда **reset** сбрасывает текущий stage.

```
$git reset HEAD filename
```

Работа с локальным репозиторием

Команда **commit** сохраняет текущий stage в локальный репозиторий.

```
$git commit
```

```
$git commit --amend
```

Команда **branch** создает/удаляет новую ветку.

```
$git branch
```

```
* master
```

```
$git branch test
```

```
$git branch
```

```
* master
```

```
  test
```

Команда **checkout** переключает рабочую копию на другую ветку.

```
$git checkout test
```

```
Switched to branch 'test'
```

Работа с удаленным репозиторием

Команда **clone** клонирует репозиторий и создает рабочую копию.

```
$git clone git@github.com:kothic/kothic-js.git
```

```
Cloning into kothic-js...
```

```
remote: Counting objects: 1771, done.
```

```
remote: Compressing objects: 100% (993/993), done.
```

```
remote: Total 1771 (delta 840), reused 1689 (delta 759)
```

```
Receiving objects: 100% (1771/1771), 6.36 MiB | 492 KiB/s, done.
```

```
Resolving deltas: 100% (840/840), done.
```

Команда **push** отправляет изменения в удаленный репозиторий.

```
$git push origin master
```

Команда **pull** забирает изменения указанной ветки из удаленного репозитория и сливает их в текущую ветку.

```
$git pull origin master
```

Команда **fetch** забирает все изменения из удаленного репозитория.

```
$git fetch
```

Временное хранилище (stash)

Команда **stash** помещает изменения из stage во временное хранилище и сбрасывает рабочую копию.

```
$git stash
```

```
Saved working directory and index state WIP on test:  
7376ecb Performance test  
HEAD is now at 7376ecb Performance test
```

```
$git status
```

```
Saved working directory and index state WIP on test:  
7376ecb Performance test  
HEAD is now at 7376ecb Performance test
```

```
$git stash apply
```

```
# On branch test  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       new file:   INSTALL  
#
```

git: что внутри

- Контентно-адресуемое key-value хранилище
- В качестве ключа используется SHA-1 хэш от содержимого
- Указатели на выделенные точки в репозитории

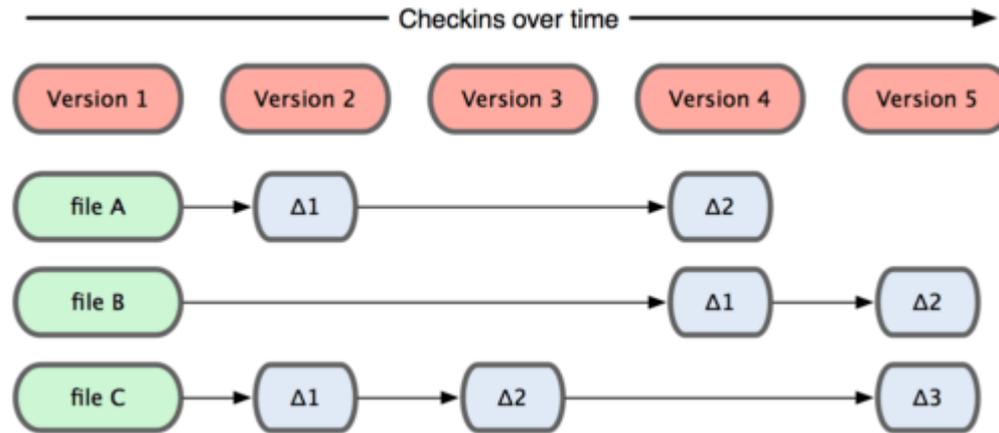
Identity management

Поскольку в распределенной среде нет центрального сервера который может назначать уникальные номера ревизий, в качестве идентификаторов ревизий используются хэши.

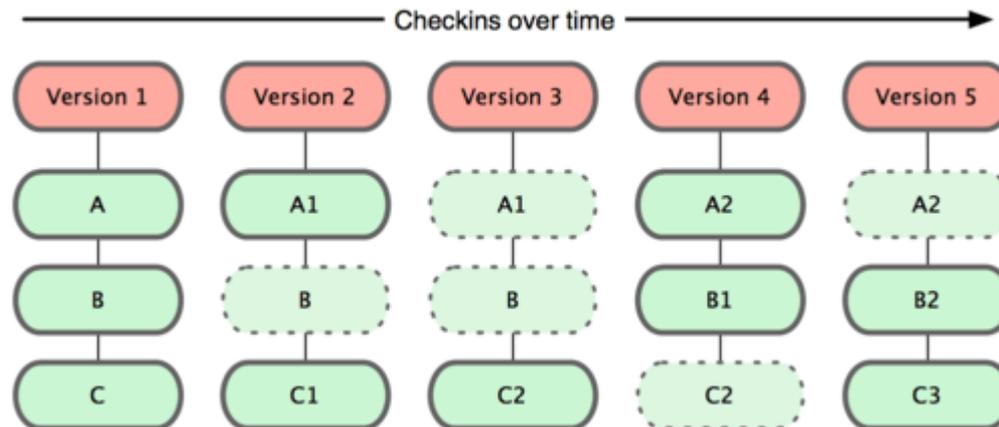
```
$git log --pretty="%h - %s"  
7376ecb - Performance test  
e8d3877 - Added style from openstreetmap.org Added mapcss  
target to Makefile  
2406a45 - Merge branch 'master' of  
github.com:kothic/kothic-js  
478ef8e - recompile  
d30b6e8 - faster thin lines rendering  
c295b70 - Added .gitignore  
4956337 - Checked style with jslint
```

Хранение данных

SVN



git

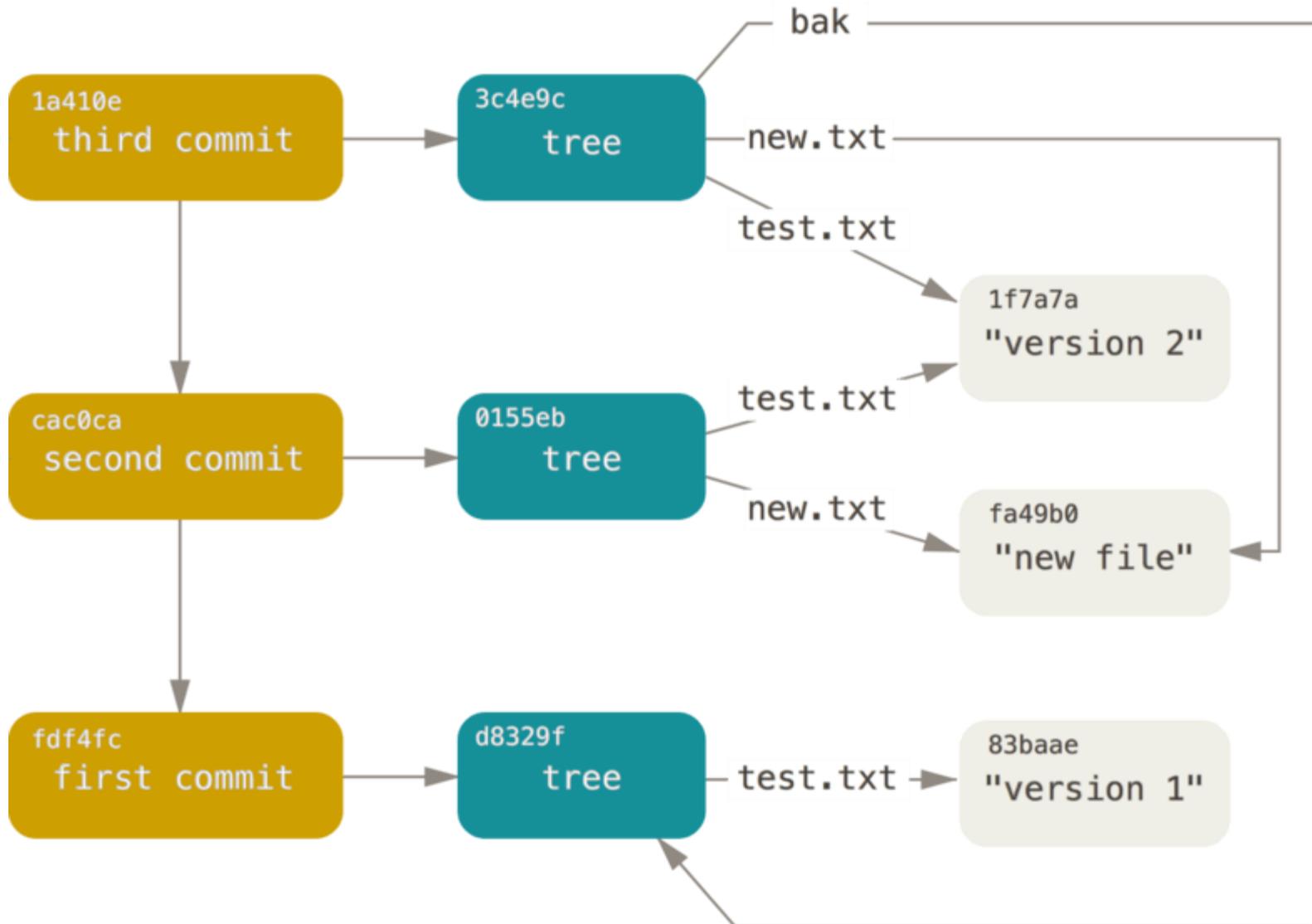


Объекты

- blob (содержимое файлов)
- tree (папки и имена файлов со ссылками на соответствующие объекты)
- commit (tree + метаданные + родительские коммиты)

Ветки это просто указатели на определенные коммиты

Хранение информации в git



Зачем нужны ветви

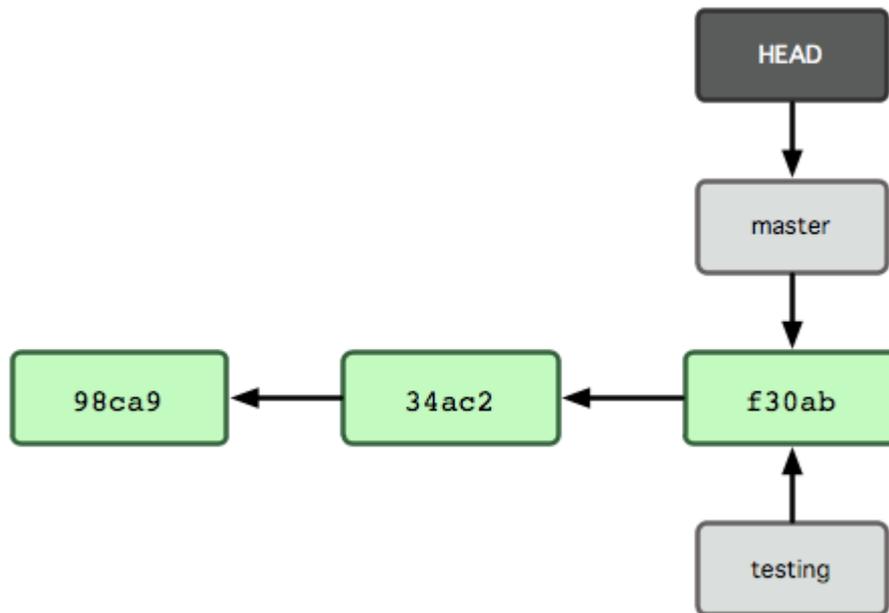
- Release branch. Параллельная разработка и поддержка старых версий.
- Feature branch. Разработка нового функционала без риска сломать работающий код.
- Частые коммиты в собственную ветку.

К сожалению у ветвей есть существенный недостаток. Операция слияния практически всегда требует ручной работы.

Ветвление

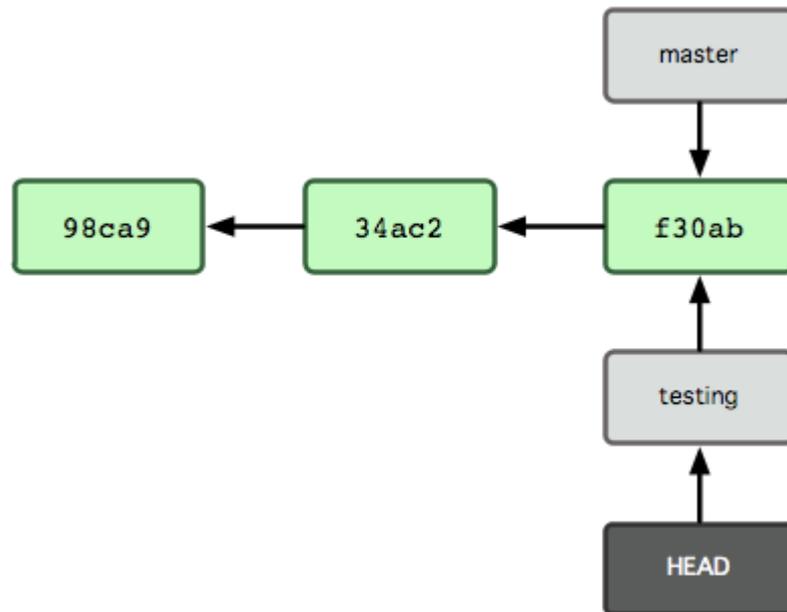
`git clone http://gitlab.dev.ccfit.nsu.ru/students2016/example.git`

`git branch testing`



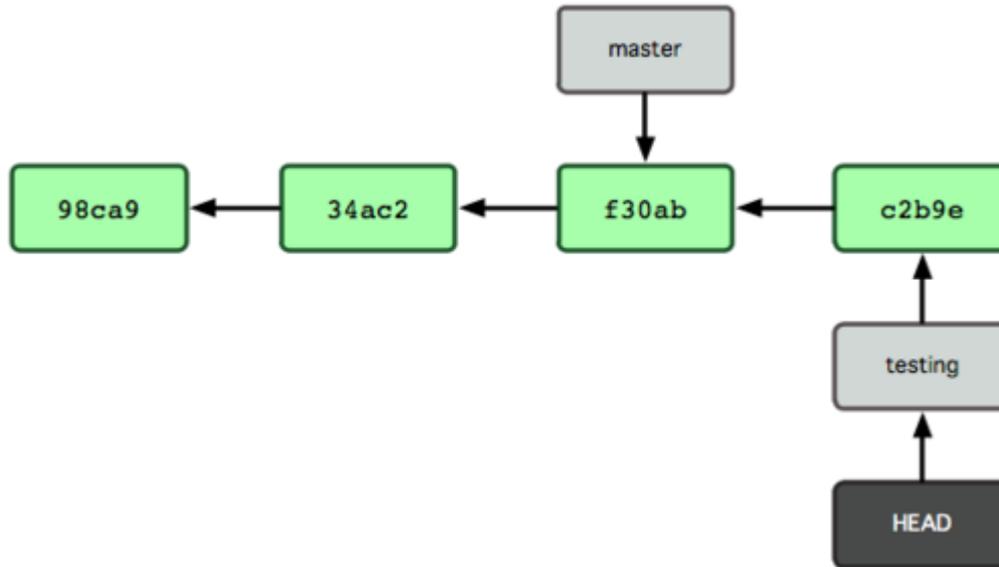
Ветвление

git checkout testing



Ветвление

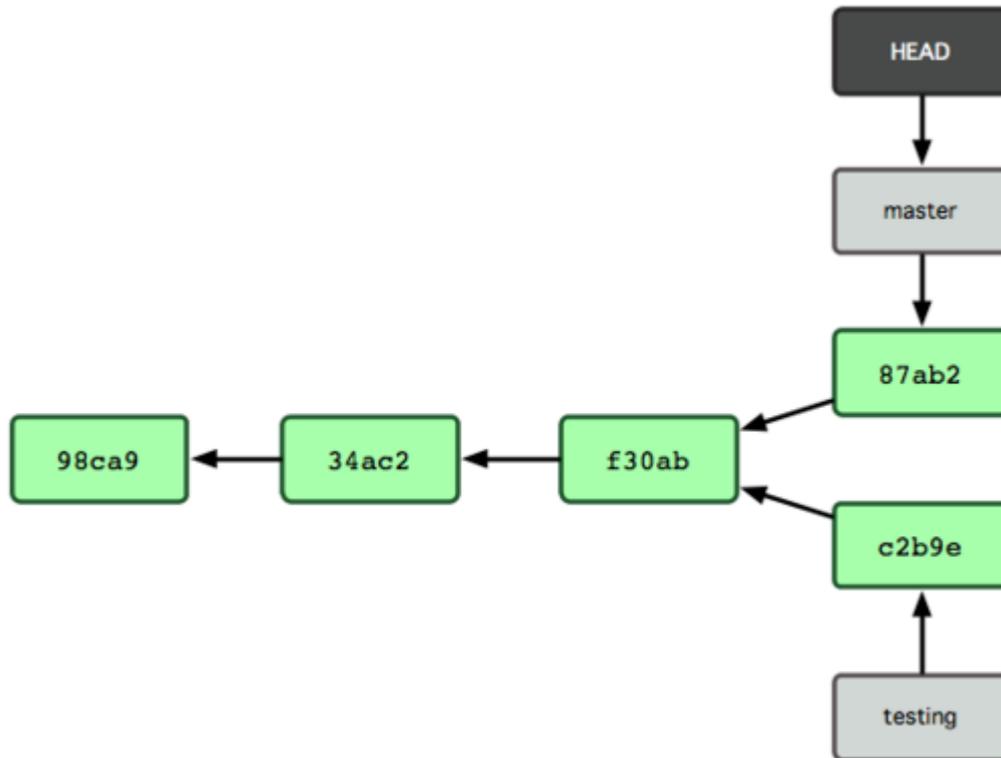
`git commit -a -m 'change in the file'`



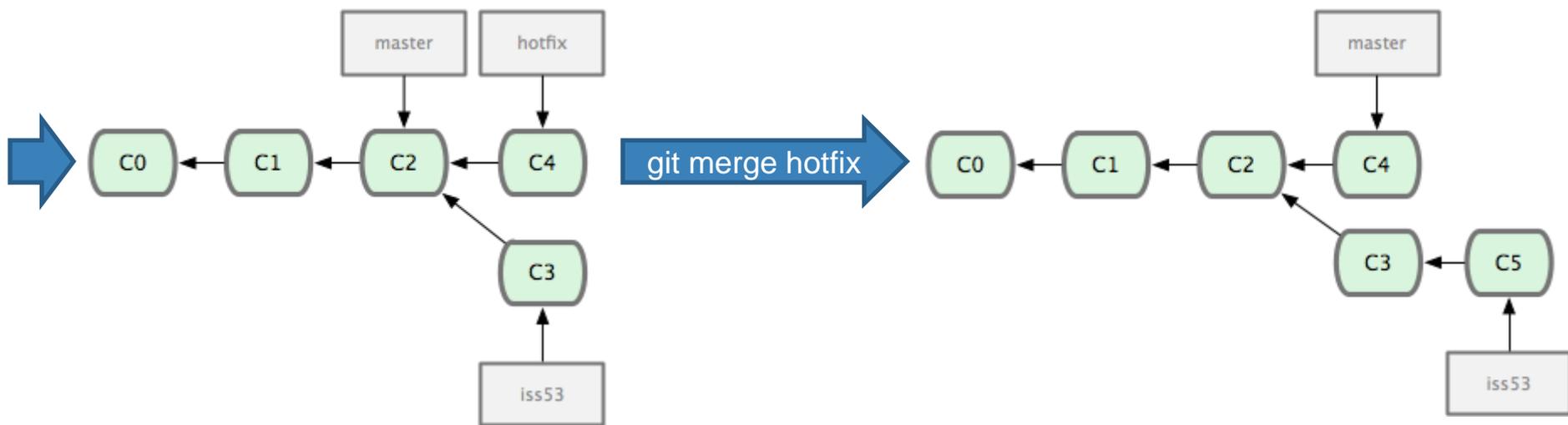
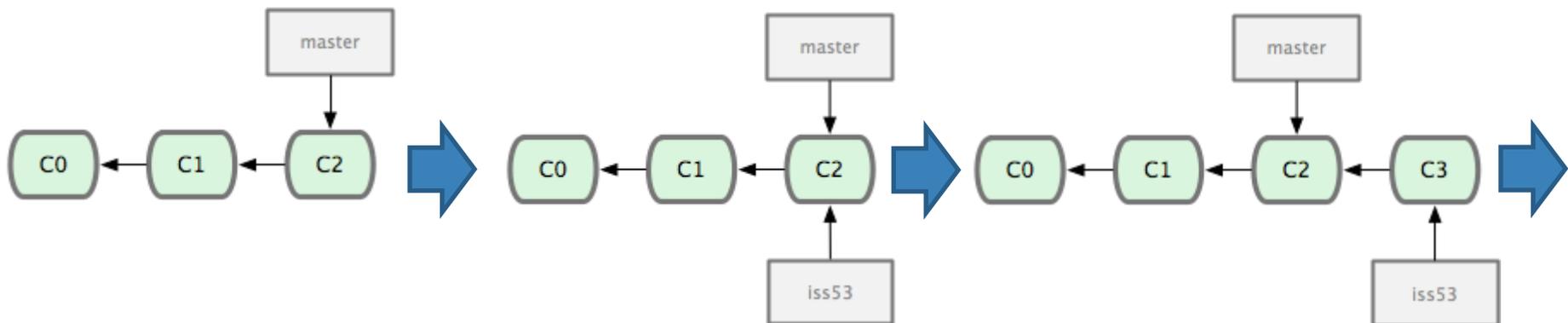
Ветвление

`git checkout master`

`git commit -a -m 'another change in the file'`

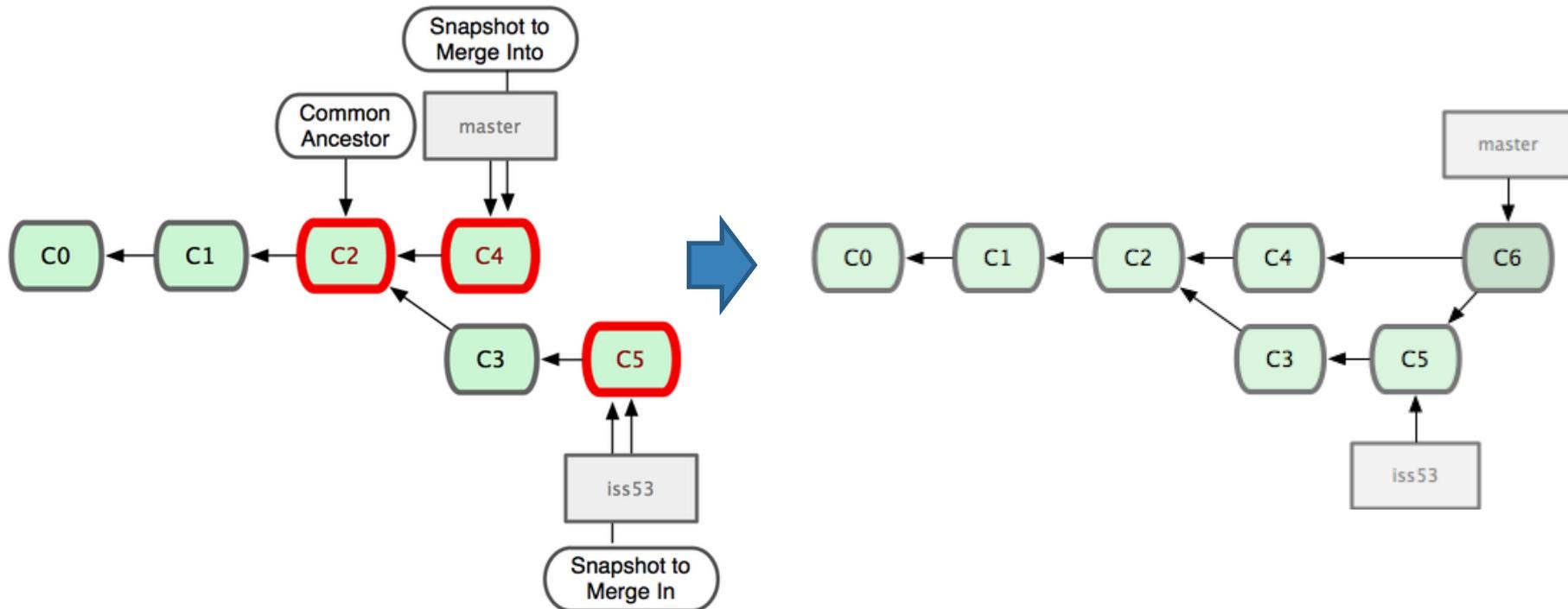


Слияние веток

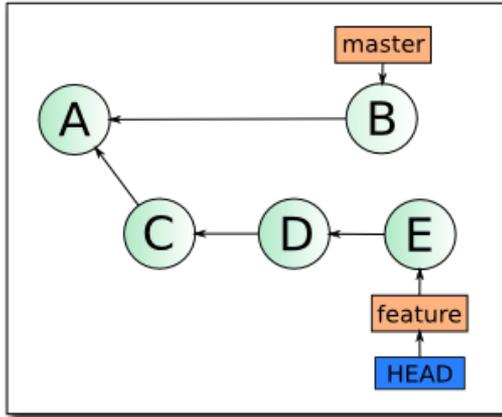


Слияние веток

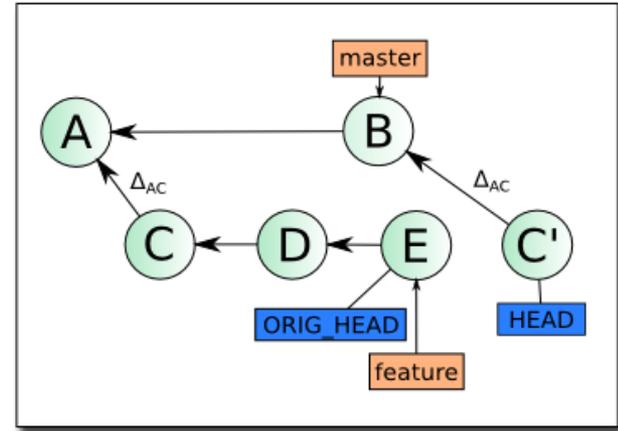
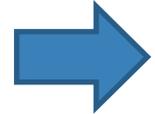
`git merge iss53`



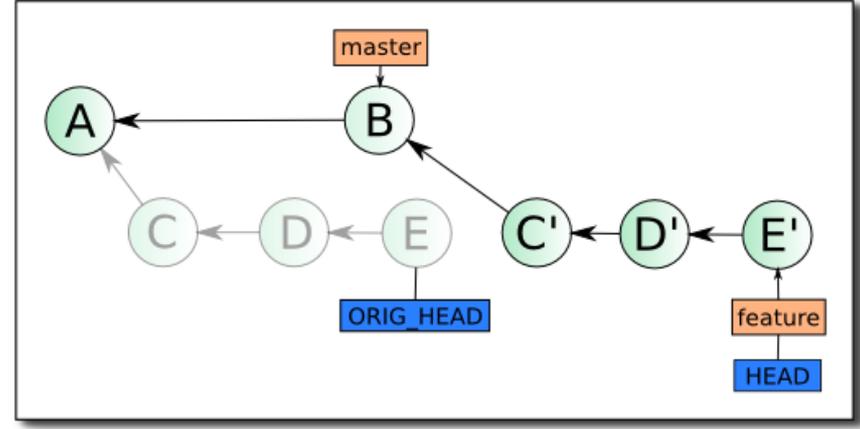
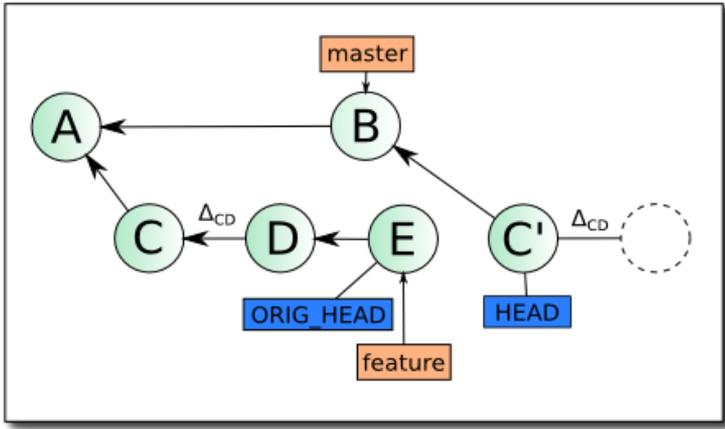
Rebase



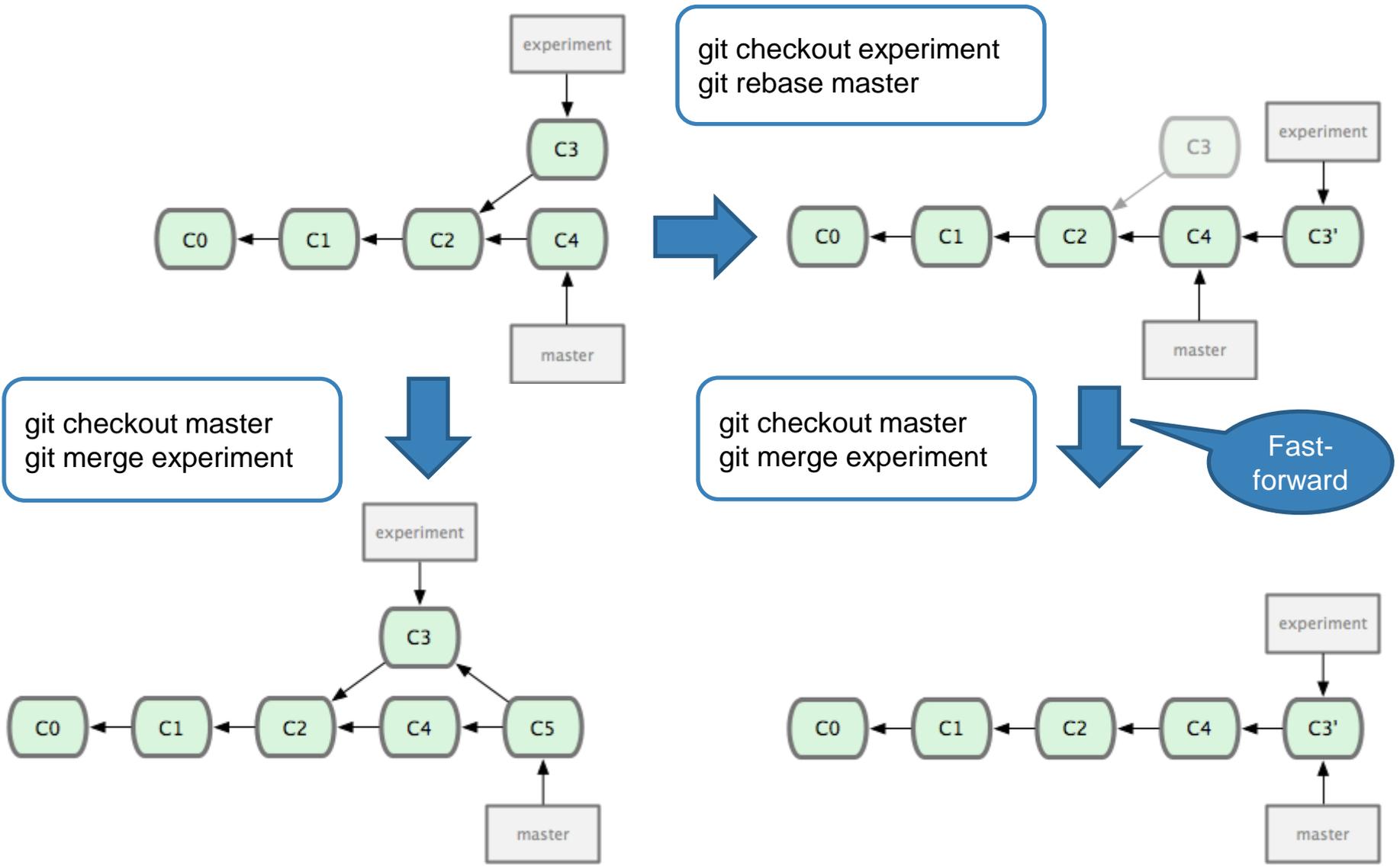
git checkout feature
git rebase master



- git rebase --abort
- git rebase --continue
- git rebase --skip



Merge vs Rebase



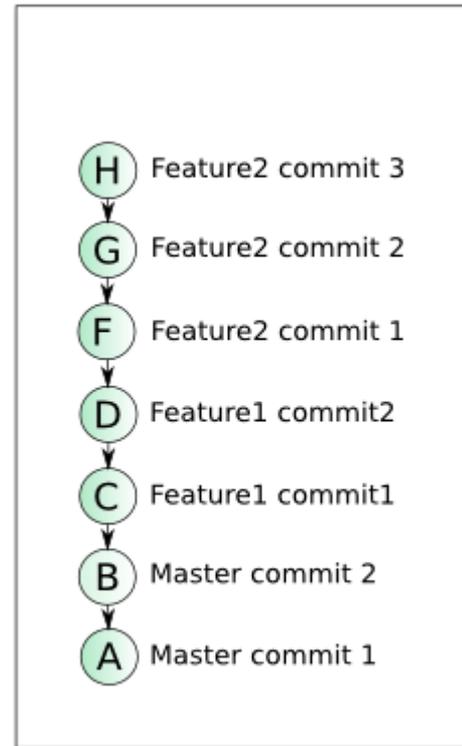
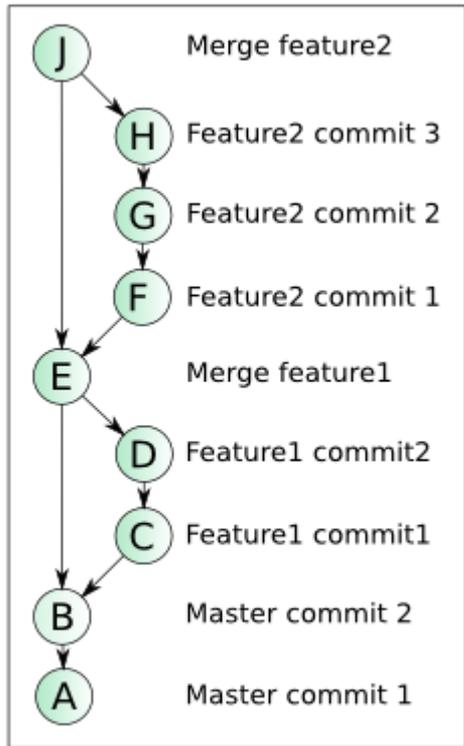
Merge vs Rebase

Поскольку git умеет изменять локальную историю, часто используется такой workflow:

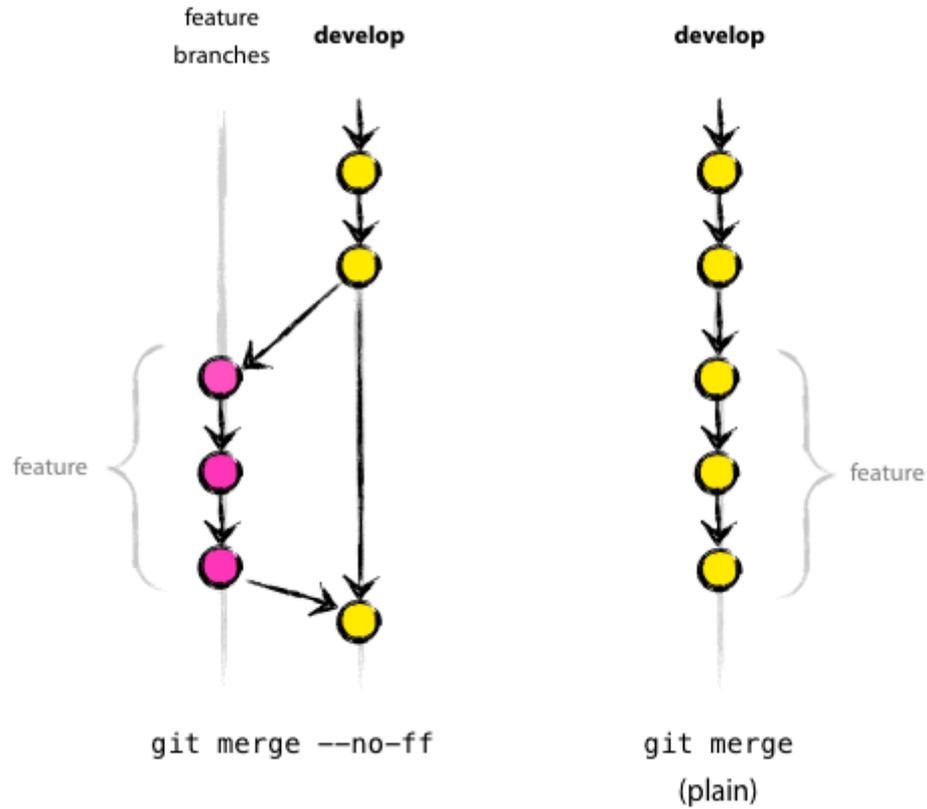
1. При разработке изменения коммитятся очень часто и довольно грязно.
2. Как только разработка завершена делается rebase к master
3. Все коммиты сбрасываются и изменения остаются только в рабочей копии
4. С помощью команд **add -i** и **commit** изменения группируются по смыслу
5. Уже причесанные изменения заливаются в удаленный репозиторий.

Примерно того же результата можно добиться через **git merge --squash newFeature && git commit -m 'Your custom commit message'**

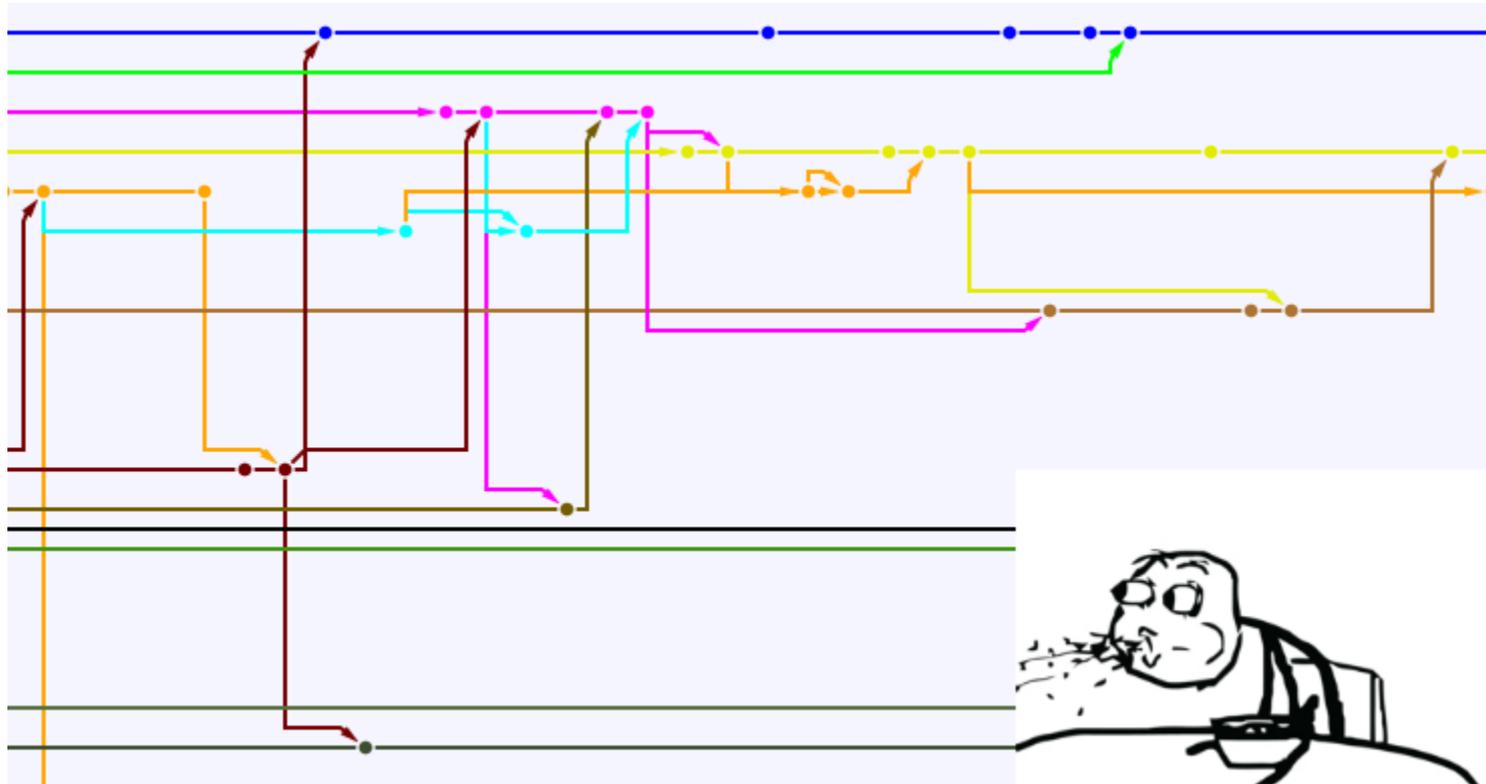
Merge vs Rebase



Merge vs Rebase



Merge vs Rebase



Merge vs Rebase

Rebase - **возможность** почистить историю приватных веток от запутанных слияний и множества мелких коммитов

Rebase Policy – **опасность** работы с публичными ветками, другие разработчики вынуждены будут проходить цепочку новых коммитов каждый раз

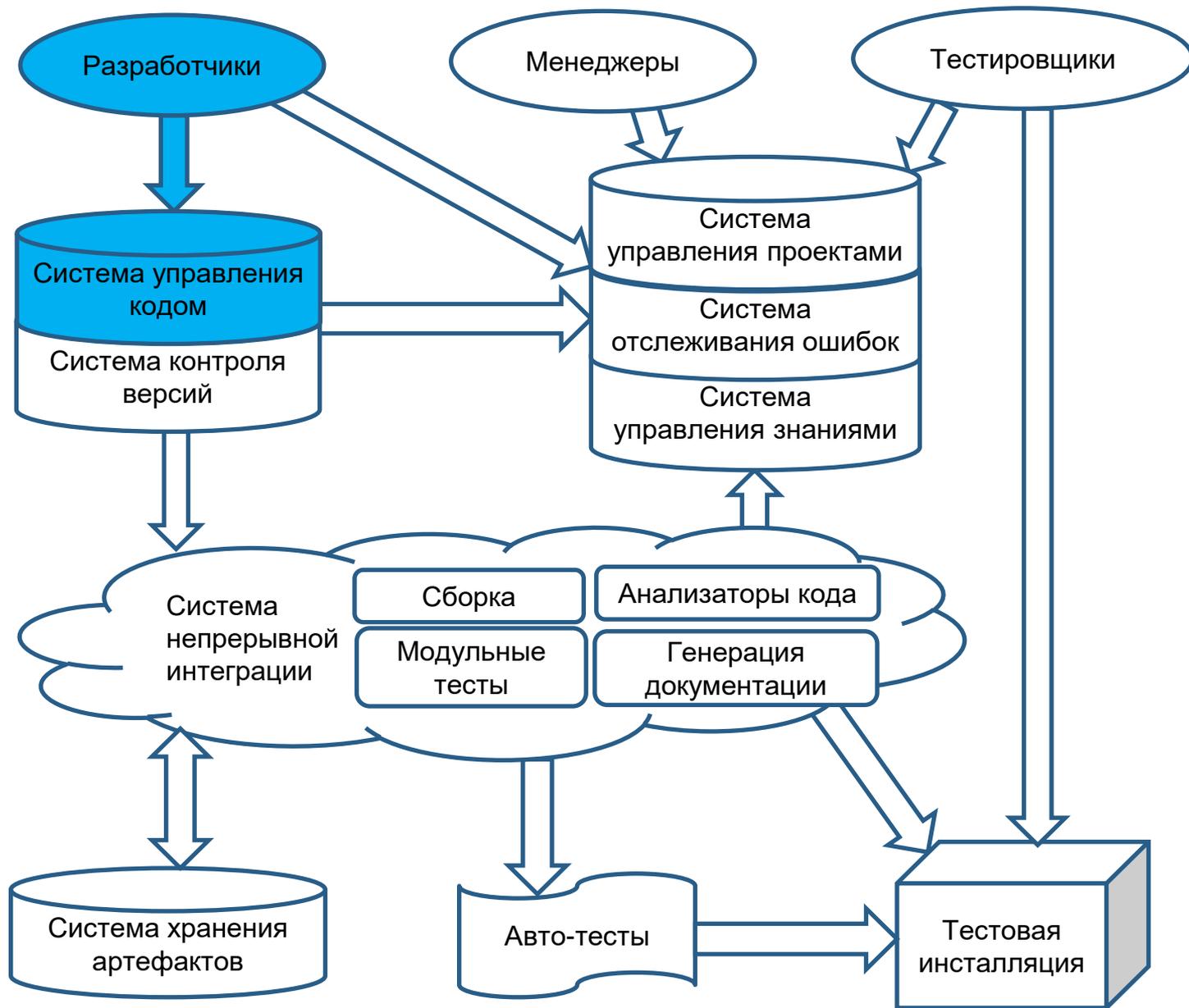
Merge Policy – **простота** использования, **запутанность** истории коммитов

Достоинства DVCS

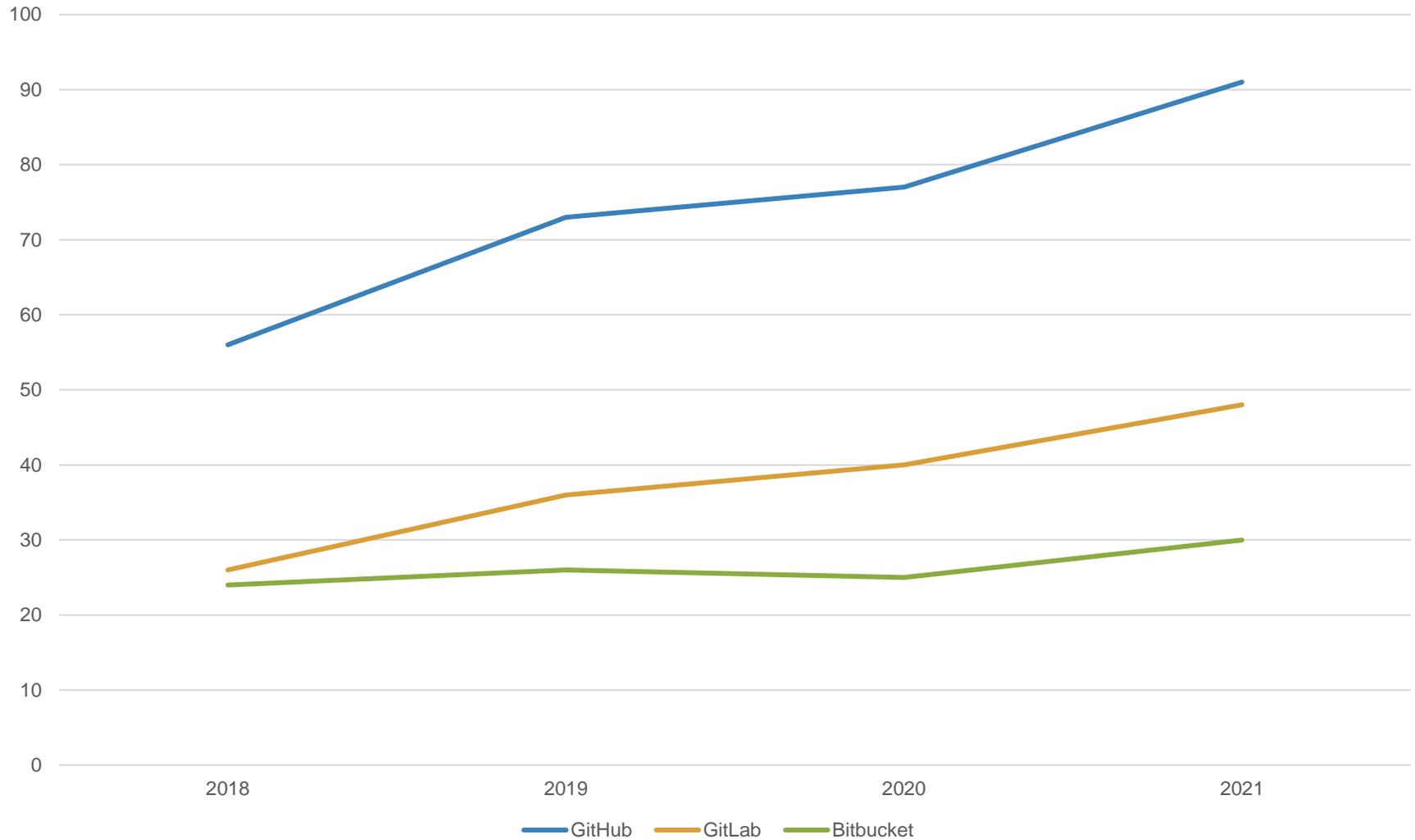
- Нет выделенного центра (технически)
- Вся история дублируется в каждом локальном репозитории
- Нет необходимости в постоянном соединении
- Нет необходимости в управлении правами
- Поддерживает любой workflow
- Каждый разработчик работает в своей песочнице
- Простое создание и слияние веток
- Локальные операции работают быстро
- Позволяет откладывать решение о коммите до момента push'a на сервер

Недостатки DVCS

- Все еще требуются бэкапы
- Нет сквозных номеров ревизий
- Достаточно сложны в использовании, по крайней мере git



Usage in % of interviewed developers



Источник: <https://www.jetbrains.com/lp/devecosystem-2021/team-tools/>

ОСНОВНЫЕ ВОЗМОЖНОСТИ

</> openrtb

- Source
- Commits
- Branches
- Pull requests
- Pipelines
- Deployments
- Downloads
- Boards
- Settings**

soft / rtb-libraries / openrtb

Settings

GENERAL

[Repository details](#)

User and group access

[Access keys](#)

[Username aliases](#)

WORKFLOW

[Branch permissions](#)

[Branching model](#) **NEW**

[Merge strategies](#)

[Default reviewers](#)

[Webhooks](#)

[Links](#)

FEATURES

[Git LFS](#)

[Wiki](#)

[Issue tracker](#)

TRELLO

[Settings](#)

CHAT NOTIFICATIONS

[Settings](#)

PIPELINES

[Settings](#)

[Environment variables](#)

[SSH keys](#)

[Integrations](#)

User and group access

Here's where you grant users and [groups](#) access to this repository. For a list of all users with access to any of your private repositories, see which users count towards your bill on the [Users on plan](#) page. [Learn more](#)

Users

| Username or email address | Read | | Add |
|---|-------------------------------------|--------------------------------------|---|
|  soft | | owner | |
|  Alexander K | <input type="button" value="READ"/> | <input type="button" value="WRITE"/> | <input type="button" value="ADMIN"/> <input type="button" value="X"/> |
|  Alexey E | <input type="button" value="READ"/> | <input type="button" value="WRITE"/> | <input type="button" value="ADMIN"/> <input type="button" value="X"/> |
|  Denis Anoykin | <input type="button" value="READ"/> | <input type="button" value="WRITE"/> | <input type="button" value="ADMIN"/> <input type="button" value="X"/> |

Groups

| Select a group | Read | | Add |
|--|-------------------------------------|--------------------------------------|---|
| Administrators (asteriosoft:administrators) | <input type="button" value="READ"/> | <input type="button" value="WRITE"/> | <input type="button" value="ADMIN"/> <input type="button" value="X"/> |
| Developers (asteriosoft:developers) | <input type="button" value="READ"/> | <input type="button" value="WRITE"/> | <input type="button" value="ADMIN"/> <input type="button" value="X"/> |

Pull requests



<> Source

⬇️ Commits

🔗 Branches

🔗 Pull requests

🔄 Pipelines

📦 Deployments

📄 Downloads

📖 Boards

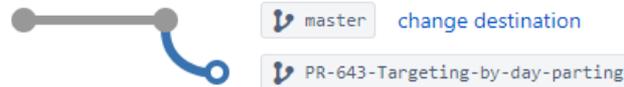
⚙️ Settings

PR-643-Targeting-by-day-parting

Check out ▾

View source

Merge



🔄 1 of 1 passed ✓

🏠 Create pull request

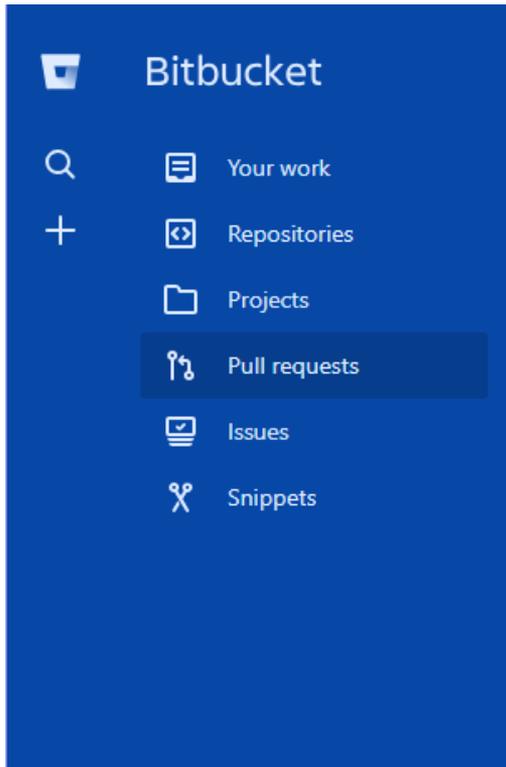
🔒 No restrictions edit



Diff Commits Merged pull requests

| Author | Commit | Message | Date | Builds |
|--|---------|---------------------------------|------------|--------|
| 140 commits behind master. Sync now. | | | | |
| 👤 Evgeniy Shushk... | 84a58db | PR-643 Targeting by day parting | 2018-07-18 | ✓ |

Pull requests



Bitbucket navigation sidebar with the following items:

- Search icon
- Your work
- + Repositories
- Projects
- Pull requests** (highlighted)
- Issues
- Snippets

Pull requests

Filter by: **Reviewing** Watching Mine Teams

| Repository | Author | Title | Updated | Builds |
|---|---|--|--------------|---|
|  bidder |  Alexander Zhukov... | #487: PR-775 , PR-776 : SSP-bundl... | 13 hours ago |   |
|  ui-frontend |  Pavel Samolovov | #129: PR-773 : Added 'Campaign B... | 15 hours ago |   |
|  ui |  Pavel Samolovov | #536: PR-773 group by campaign ... | 15 hours ago |   |
|  ui |  Alexander Zhukov... | #535: PR-775 , PR-776 : SSP-b...  2 | 2 days ago |   |
|  mango-x |  Alexander Zhukov... | #265: MGO-933: fix vast unwrap h... | 2018-08-07 |   |
|  rmx2 |  Alexander Zhukov... | #51: EUS-96: opt out cookies | 2018-02-15 |   |
|  rmx2 |  Alexander Zhukov... | #37: EUS-88: standard macros | 2018-02-12 |   |

Pull requests

PR-773: Added 'Campaign Bundle' groupby option

#129 **OPEN** PR-773-group-by-campaig... → master

Merge

Edit

Decline

Approve

0

Overview **Commits** Activity

Author  Pavel Samolovov

2 of 2 passed 

Reviewers  

Stop watching

Description *No description*

Comments (0)



What would you like to say?

Files changed (2)

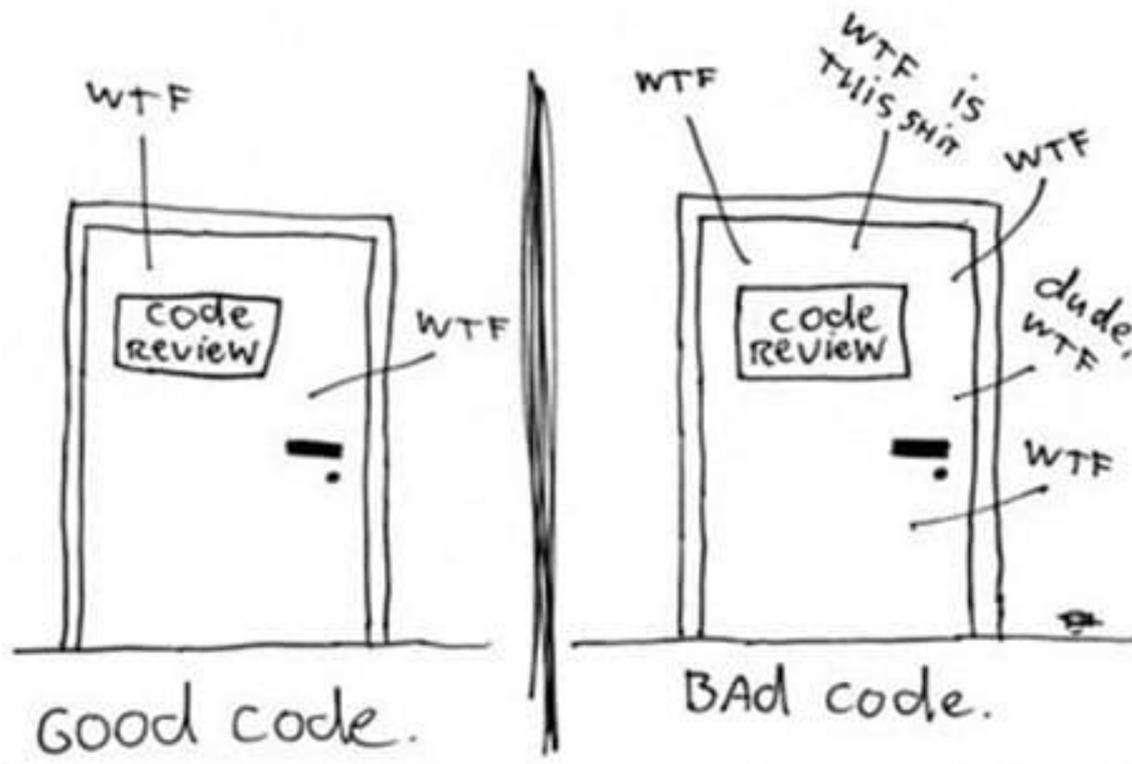
+8 -0 **M** src/api/base/api.ts
+1 -0 **M** src/views/report/stat-report.tsx

src/api/base/api.ts **MODIFIED** Side-by-side diff View file Comment ...

```
1530 1530 */
1531 1531 campaign?: string;
1532 1532 /**
1533 + *
1534 + * @type {string}
1535 + * @memberof EventStatsRow
1536 + */
```

Pull requests

The ONLY VALID MEASUREMENT
OF CODE QUALITY: WTFs/MINUTE



Pull requests

The image shows a screenshot of the GitHub repository settings page, specifically the 'Branch permissions' section. A modal dialog titled 'Add a branch permission' is open, allowing the user to configure permissions for a specific branch or pattern. The background settings page is dimmed.

Settings

- GENERAL
 - Repository details
 - User and group access
 - Access keys
 - Username aliases
- WORKFLOW
 - Branch permissions**
 - Branching model **NEW**
 - Merge strategies
 - Default reviewers
 - Webhooks
 - Links
- FEATURES
 - Git LFS
 - Wiki
 - Issue tracker
- TRELLO
 - Settings
- CHAT NOTIFICATIONS
 - Settings
- PIPELINES
 - Settings
 - Environment variables
 - SSH keys

Branch permissions

Add a branch permission

Branch or pattern:

Write access:
Users and groups who can push or merge any changes to this branch directly

Merge via pull request:
Users and groups who can merge to this branch via pull request

Merge checks

- Check for at least approval
- Check for unresolved pull request tasks
- Check the last commit for successful build and no failed builds

PREMIUM

Control merge conditions with our Premium plan. [Learn more](#)

- Prevent a merge with unresolved merge checks
- Reset approvals when the source branch is modified

Save **Cancel**

Pull requests

GENERAL

[Repository details](#)

[User and group access](#)

[Access keys](#)

[Username aliases](#)

WORKFLOW

[Branch permissions](#)

[Branching model](#) **NEW**

Merge strategies

[Default reviewers](#)

[Webhooks](#)

[Links](#)

FEATURES

[Git LFS](#)

Merge strategies

Your default merge strategy defines how you want the commit history to

Default merge
strategy

Merge commit

Merge commit

Squash

Fast forward

Save

GitHub: Pull Requests

danoykin / test Watch 0 Star 0 Fork 0

[Code](#) [Issues 2](#) [Pull requests 1](#) [Projects 1](#) [Wiki](#) [Insights](#) [Settings](#)

Filters [Labels](#) [Milestones](#) [New pull request](#)

Clear current search query, filters, and sorts

| <input type="checkbox"/> | 1 Open <input checked="" type="checkbox"/> 3 Closed | Author ▾ | Labels ▾ | Projects ▾ | Milestones ▾ | Reviews ▾ | Assignee ▾ | Sort ▾ |
|--------------------------|---|----------|----------|------------|--------------|-----------|------------|--------|
| <input type="checkbox"/> | #3 Fixes #8 opened 12 seconds ago by danoykin | | | | | | | |
| <input type="checkbox"/> | #6 Add signature after the 'Hello World' message ✓ #7 by danoykin was merged 7 hours ago | | | | | | | 1 |
| <input type="checkbox"/> | #4 add .gitignore file ✗ #5 by danoykin was merged 11 hours ago | | | | | | | |
| <input type="checkbox"/> | #1 Fix output ✗ #2 by danoykin was merged 17 hours ago | | | | | | | 2 |

ProTip! Notify someone on an issue with a mention, like: @danoykin.

GitHub: Pull requests

danoykin / test Private Watch 0 Star 0 Fork 0

[Code](#) [Issues 2](#) [Pull requests 1](#) [Projects 1](#) [Insights](#) [Settings](#)

#6 Add signature after the 'Hello World' message #7 Edit

[Open](#) danoykin wants to merge 1 commit into `master` from `issue/6-add-signature`

[Conversation 1](#) [Commits 1](#) [Checks 1](#) [Files changed 1](#)

Changes from all commits File filter... Jump to... +1 -0 Diff settings Review changes

```
1 src/main/java/HelloWorld.java
```

Show comments Copy path View file 🖨 ✎ 🗑 ⌵

@@ -4,5 +4,6 @@

```
4 public class HelloWorld {
5     public static void main(String[] args) {
6         log.info("Hello world!");
7 +     log.info("    - Your program");
```

danoykin just now
The signature is slightly incorrect. You have to write: '- Your Program'

danoykin Reply...

Resolve conversation

Start a new conversation

```
7     }
8 }
```

Интеграция с системой управления

-  openrtb
-  Source
-  Commits
-  Branches
-  Pull requests
-  Pipelines
-  Deployments
-  Downloads
-  **Boards**
-  Settings

soft / rtb-libraries / openrtb

Boards BETA

Work more collaboratively and get more done with boards

Trello's board, lists and cards enable you to organize and prioritize your projects in a fun, flexible and rewarding way.



Create a new board

or



Select an existing board ▼

Интеграция с системой управления

Add Frequency cap per app bundle

[Edit](#) [Comment](#) [Assign](#) [ReOpen](#) [Admin](#) ⌵

[↗](#) [↩](#) [⋮](#)

Type: 📖 Story Status: COMPLETED (View workflow) Assignee: 👤 Unassigned
Priority: ↑ Medium Resolution: Done Reporter: 👤 Anastasia Cheshkova
Labels: None Votes: 0 Vote for this issue
Epic Link: Bidder Watchers: 1 Start watching this issue
Sprint: PR Sept 10-11 Created: 05/Sep/18 2:03 PM
Updated: Yesterday
Resolved: 11/Sep/18 7:47 PM

Description

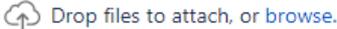
1. Add new 'Bundle' entity with properties:

- Name (unique for all bundles), VARCHAR(255)
- Daily FC Limit
- Lifetime FC Limit

2. Bidder logic:

- Check existed 'Bundle' entity for Campaign 'bundle'
- Check bundle FC settings
- Verify user FC counters per bundle
- New NBR 'BundleFC'

Attachments ⋮



Activity

All [Comments](#) [Work log](#) [History](#) [Activity](#) ⌵

 Anastasia Puzanova logged work - 11/Sep/18 7:47 PM 🔗 ✎ 🗑

Time Spent: 4 hours

<No comment>

Time Tracking +

Estimated: ▬ Not Specified

Remaining: ▬ 4h

Logged: ▬ 2d 6h 30m

Development

3 branches Updated 11/Sep/18 1:03 PM

17 commits Latest 11/Sep/18 1:04 PM

5 pull requests MERGED Updated 11/Sep/18 1:05 PM

[Create branch](#)

Agile

Требования к commit message

- Комментарий не должен быть пустым.
- В комментарии обязана быть ссылка на задачу в трекере
- Комментарий должен отвечать на вопрос что и зачем было сделано
- Комментарий должен быть лаконичным (twitter style)
- Комментарий должен быть на английском

Примеры хороших комментариев:

- Fix #7256 "flush tile cache" takes forever
See progress advancement and allow to cancel it
- See #7303: Added option to run Download dialog on startup and help line in non-expert mode
- PRD-123 Add new report column "Age"
Log user's age on every request

Именованние веток

1. В имени ветки для новой функциональности или фикса должен присутствовать номер тикета в системе управления проектом.

Примеры:

- feature/1
- feature/1-add-client-details-reporting-columns
- PRJ-123
- PRJ-123-addclient-details-reporting-columns

2. В ветке для релиза должен присутствовать полный номер релиза. Примеры:

- release/1.0.1
- release-1.0.1
- release-12

3. В основном репозитории не должны присутствовать ветки без привязки к тикету или к релизу в имени (кроме специальных веток, таких как master или develop)

Continuous integration

</> openrtb

- <> Source
- 🔗 Commits
- 🔗 Branches
- 🔗 Pull requests
- 🔄 Pipelines**
- 📦 Deployments
- 📄 Downloads
- 📊 Boards
- ⚙️ Settings

soft / rtb-libraries / openrtb

Pipelines

What's new Schedules Caches Usage ?

All branches

OTHER BRANCHES

master MAIN BRANCH

deploy-test

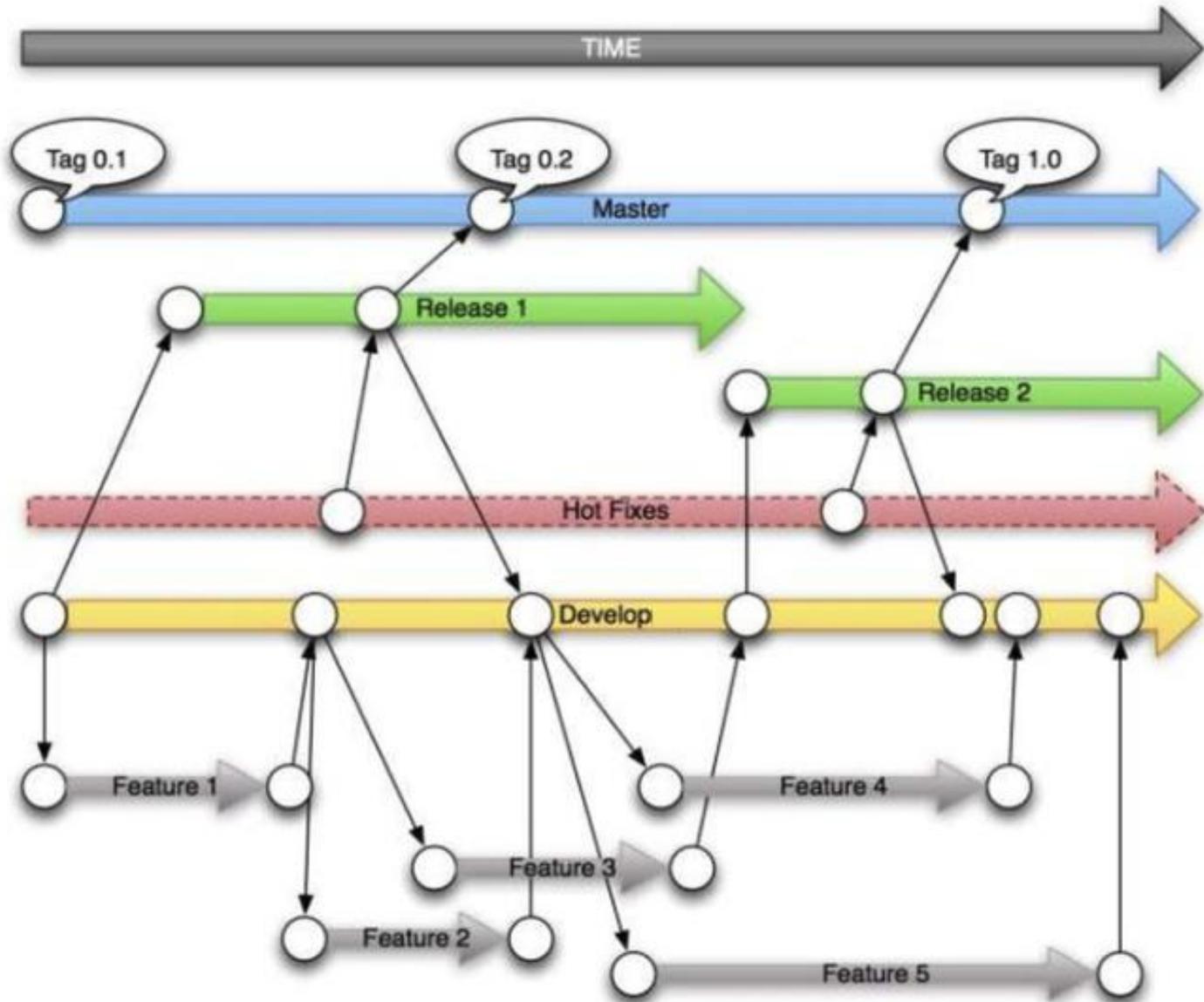
test/deploy

deploy

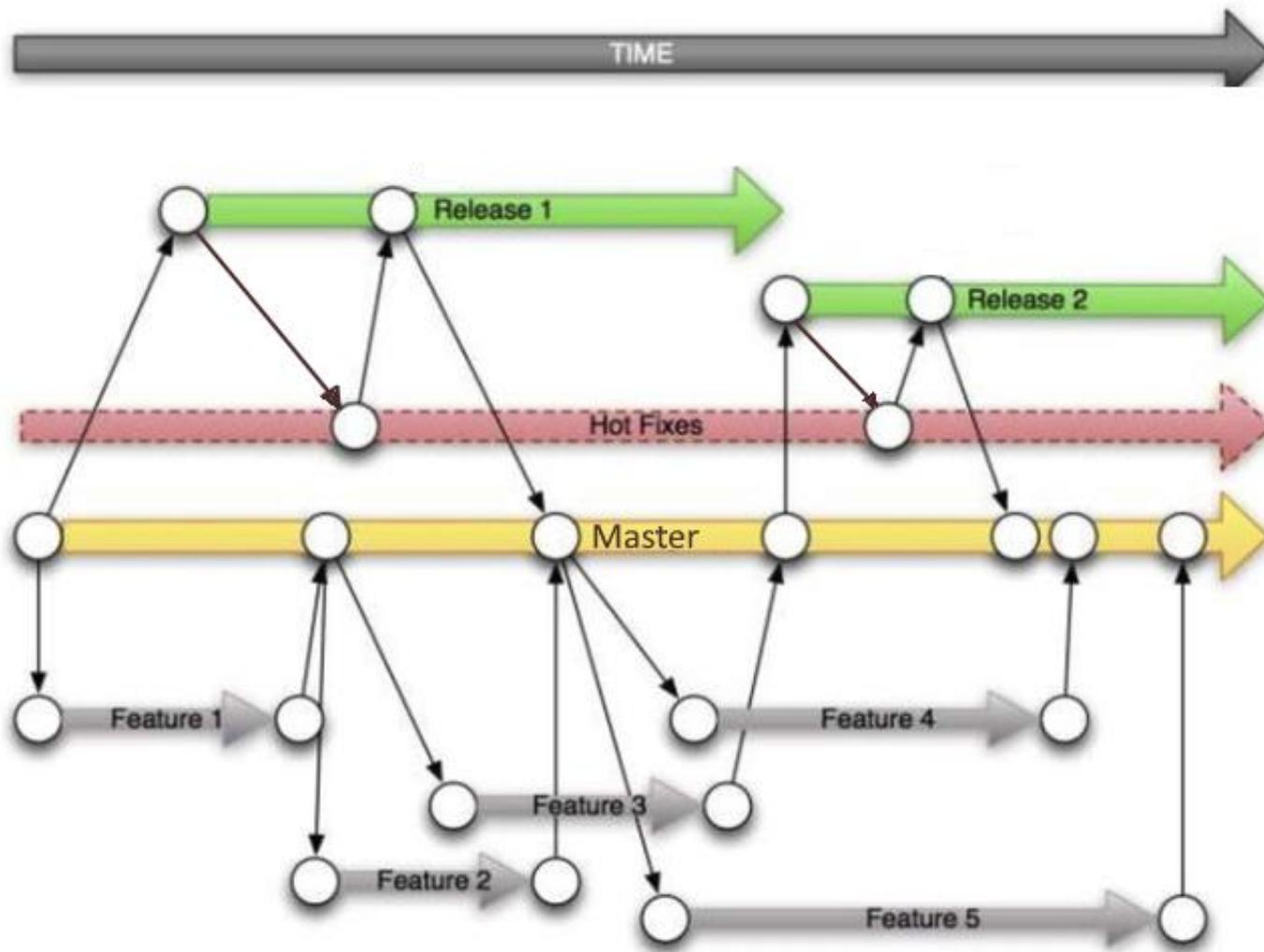
| Pipeline | Status | Started | Duration |
|---|----------------------|--------------|----------|
| #57  refactoring Alexey Emelyanov  892feea  master | ✔️ Successful | a month ago | 35 sec |
| #56  PR-555 Project migration to Java10 Alexey Emelyanov  841f7f3  master | ❌ Failed | 4 months ago | 16 sec |
| #55  PR-555 Project migration to Java10 Alexey Emelyanov  edb1a37  master | ❌ Failed | 4 months ago | 17 sec |
| #54  make all classes final Alexey Emelyanov  c1d6aff  master | ✔️ Successful | 4 months ago | 26 sec |
| #53  make all classes final Alexey Emelyanov  c1d6aff  master | ❌ Failed | 4 months ago | 22 sec |
| #52  refactoring Alexey Emelyanov  2c97504  master | ✔️ Successful | 6 months ago | 25 sec |
| #51  Merged in deploy-test (pull request #1) Deploy test Alexander Kascheev  9e2d81b  master | ✔️ Successful | 6 months ago | 37 sec |
| #50  finished Alexander Kascheev  0a76f3c  deploy-test | ✔️ Successful | 6 months ago | 18 sec |
| #49  test Alexander Kascheev  3a0fc57  deploy-test | ✔️ Successful | 6 months ago | 19 sec |
| #48  test Alexander Kascheev  56a141d  test/deploy | ✔️ Successful | 6 months ago | 17 sec |

Prev 1 2 3 4 5 6 Next

Git-flow



Упрощенный Git-flow



Рекомендуемая литература

- <http://git-scm.com/book>
- [Mercurial: The Definitive Guide](#)
- [Why Git is Better than X](#)
- [Why Switch to Bazaar?](#)
- [GitHub](#)