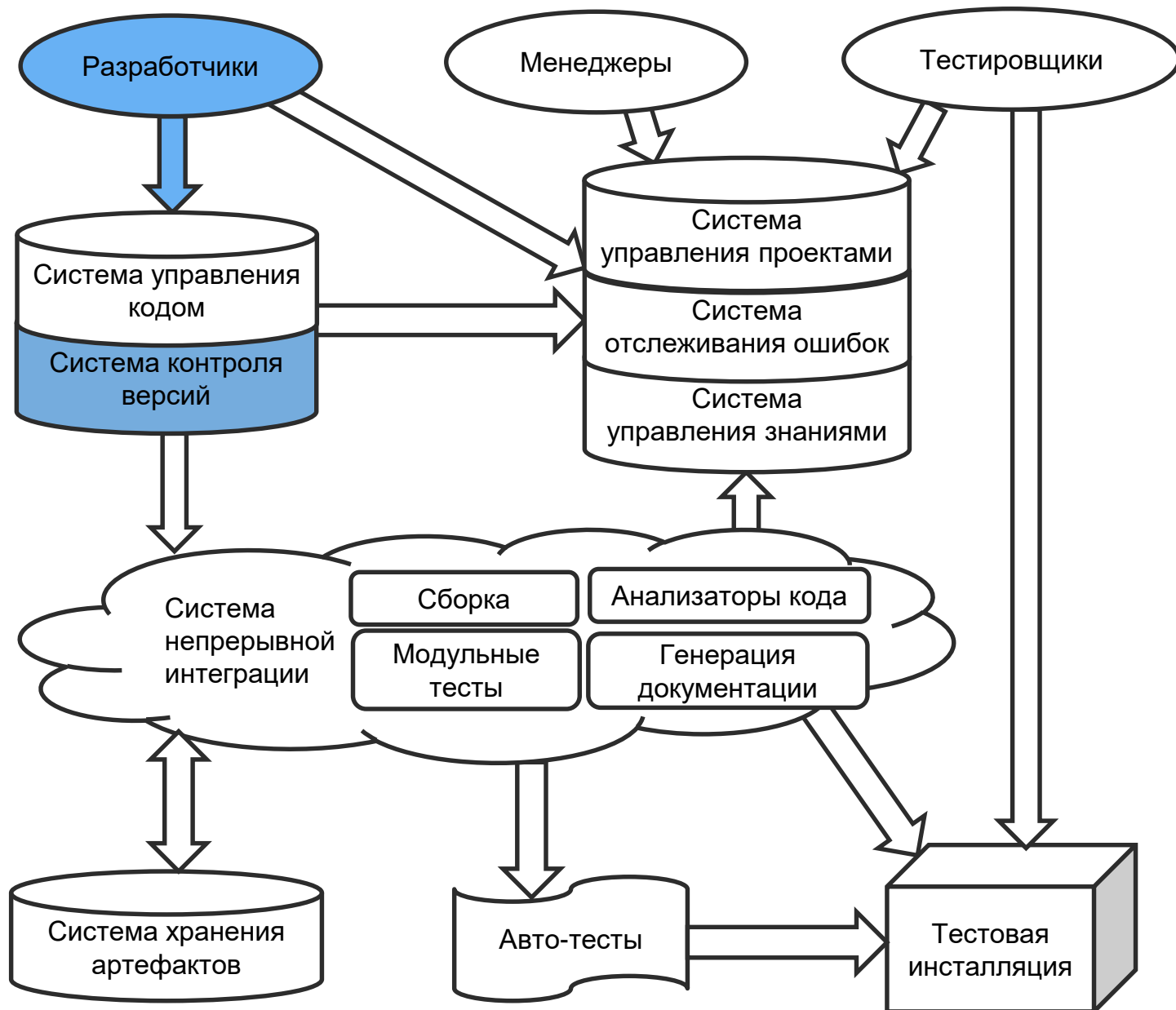


Управление производственным процессом разработки программного обеспечения

Системы контроля версий

План

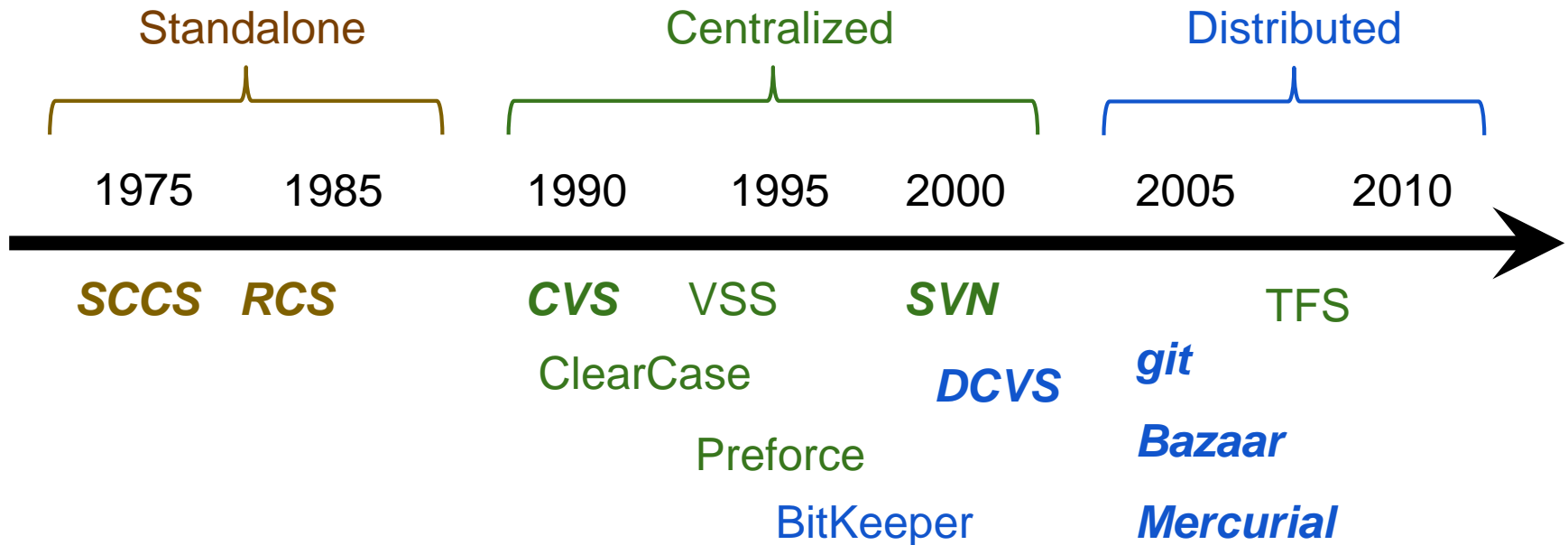
- Что такое системы контроля версий
- Основные команды
- Редко используемые возможности
- Системы контроля версий в реальных проектах



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.

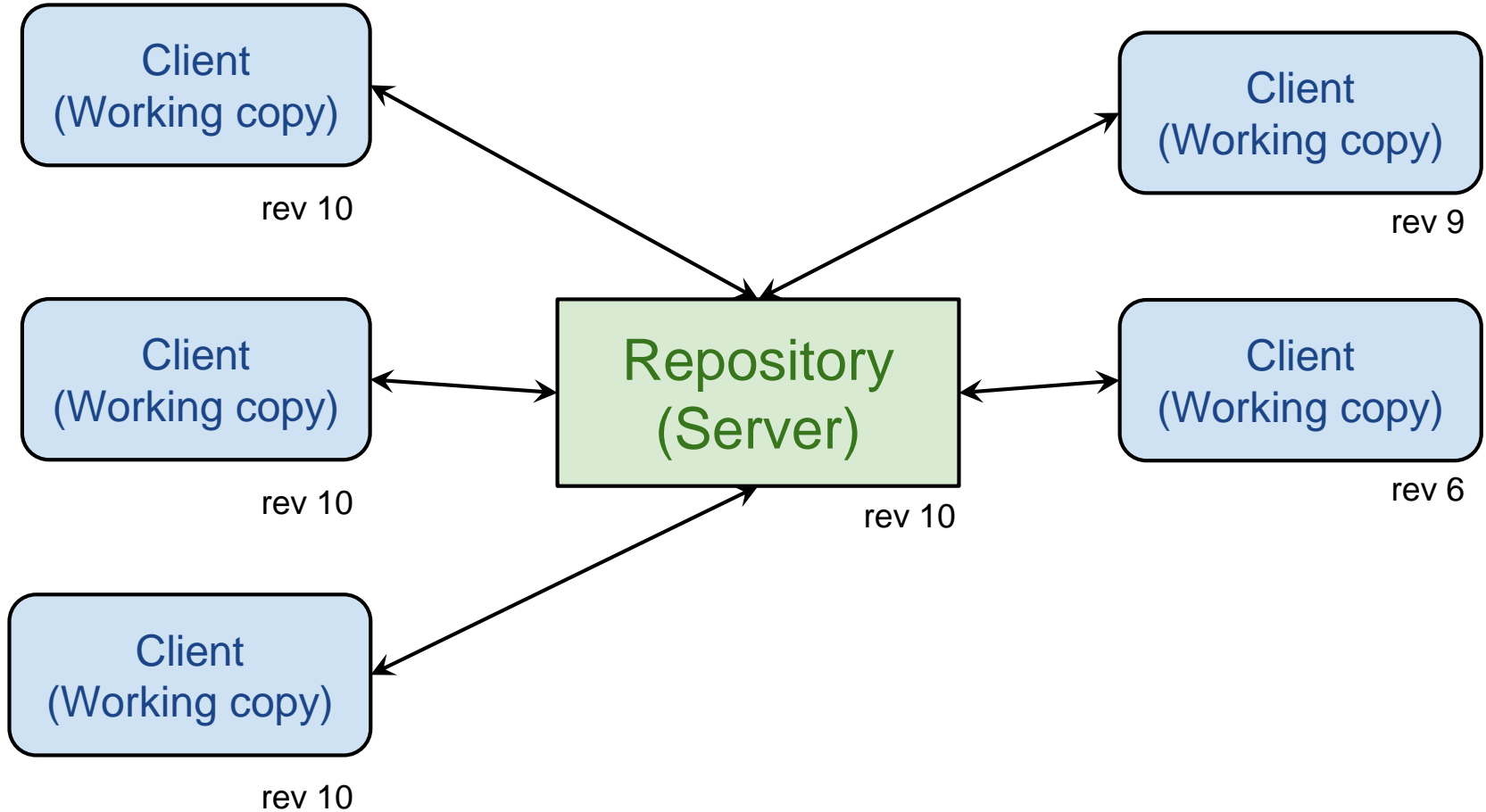
Шляпкина А.

Петров П.

Семафоров С.

Васильев В.

Centralized VCS



Достоинства и недостатки централизации

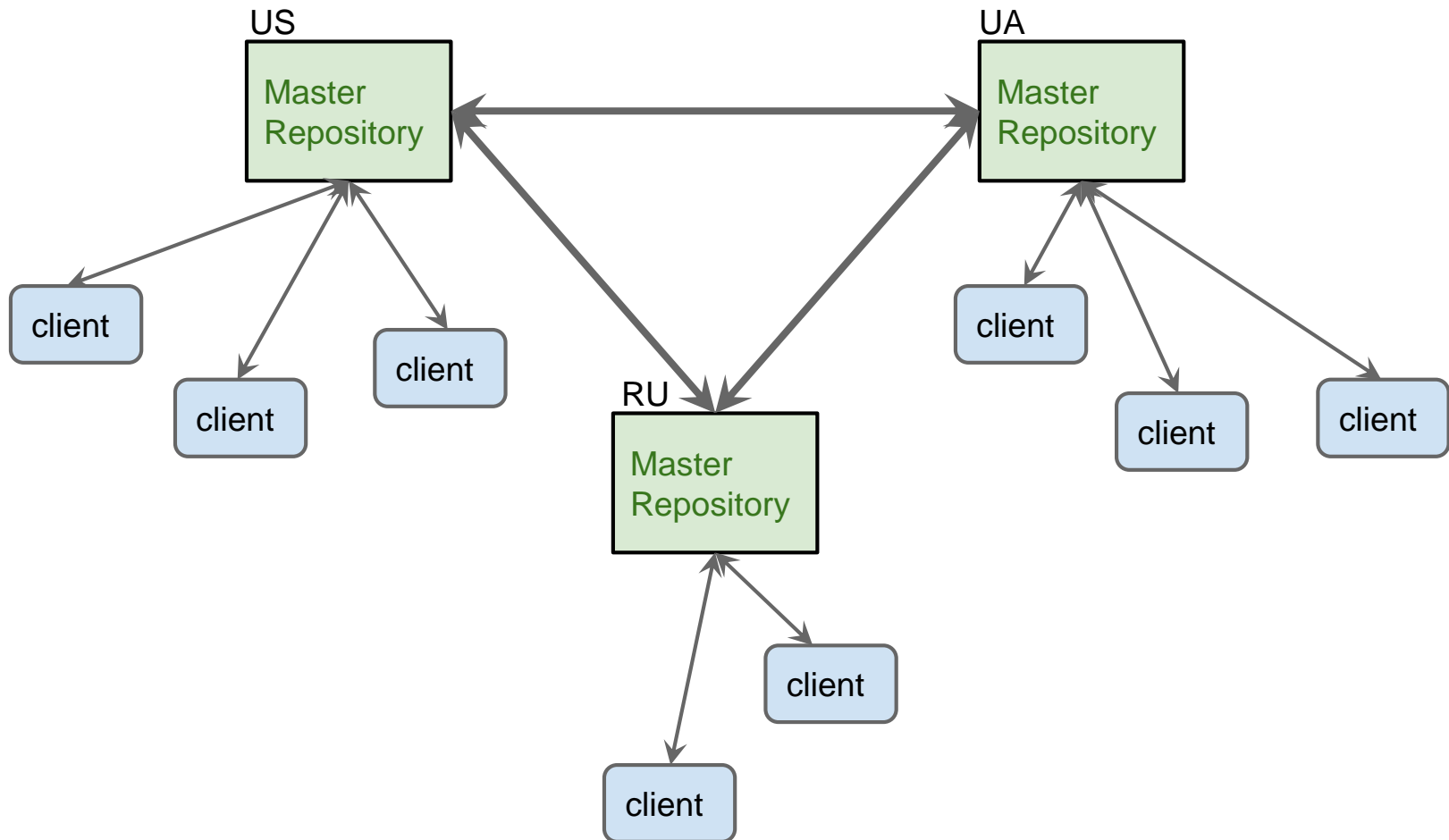
Достоинства:

- 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).

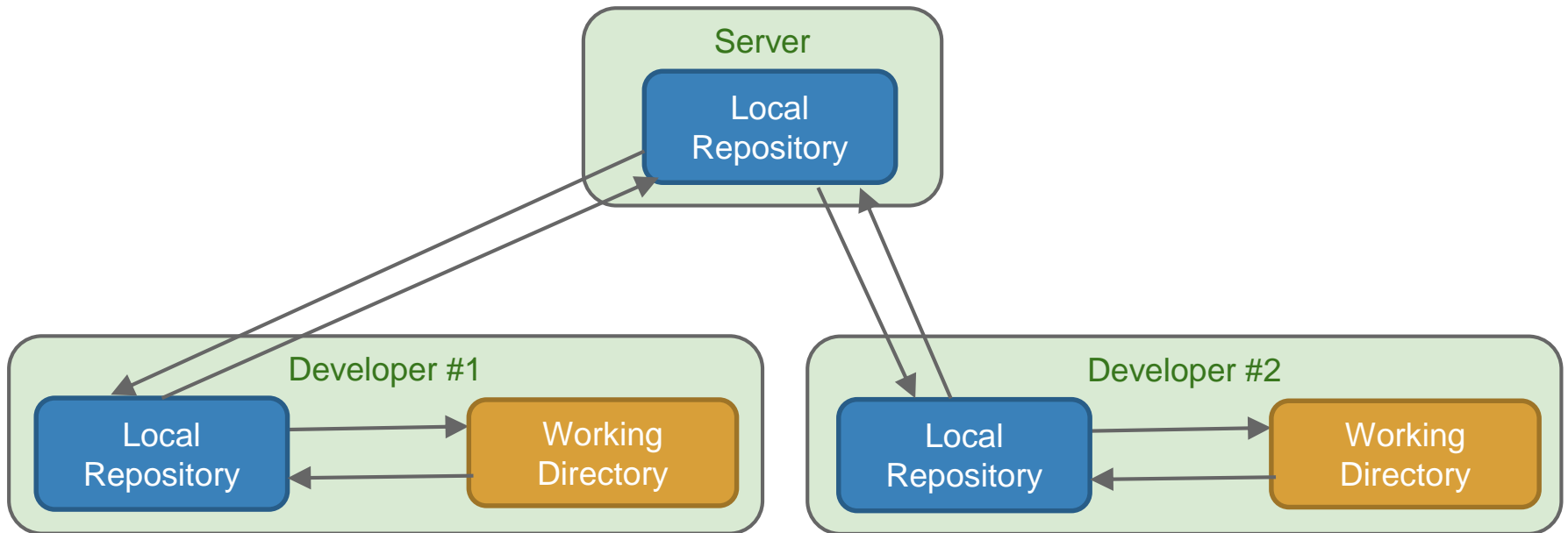
Работает, но не решает всех проблем:

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

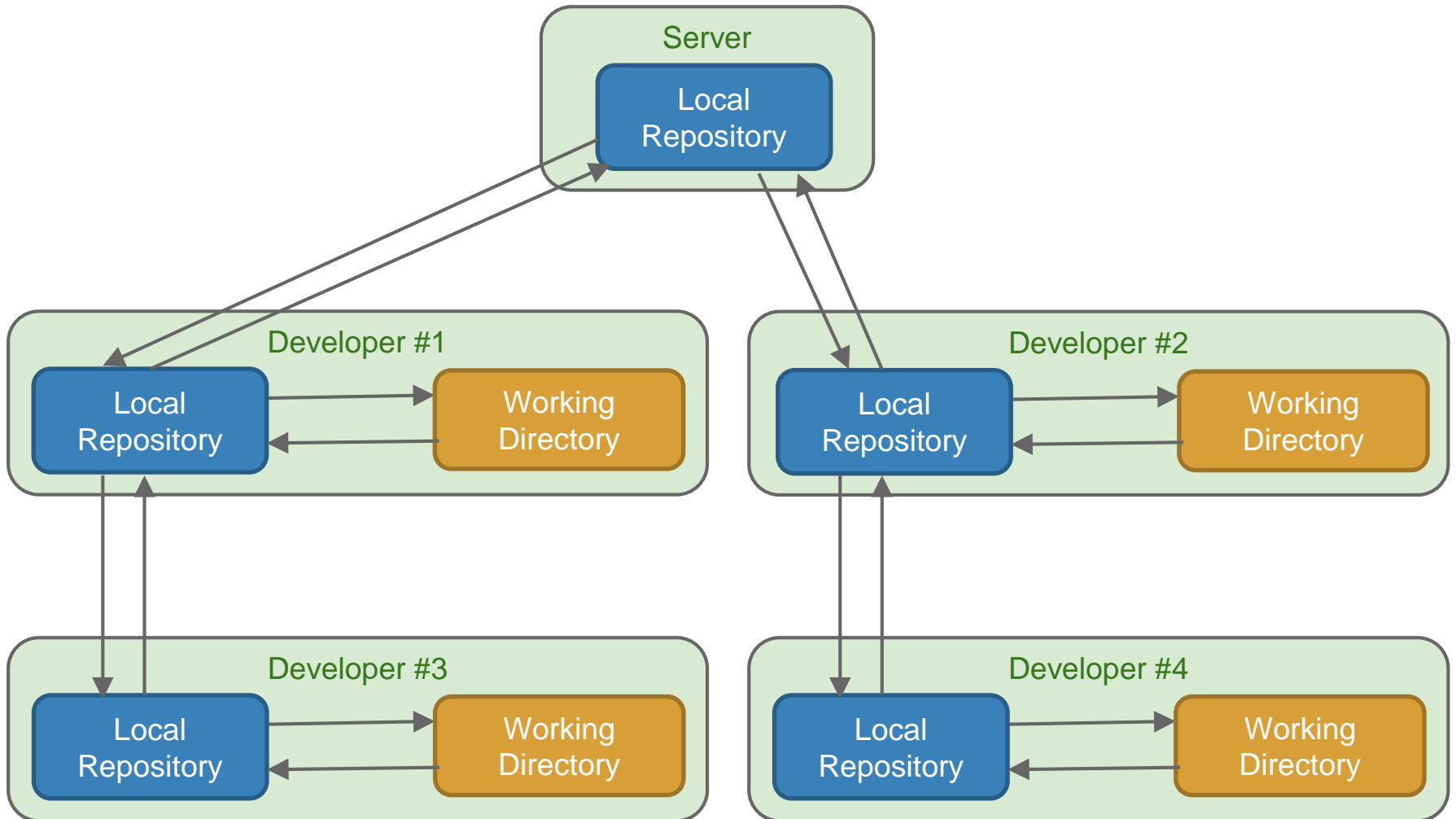
Мотивация проекта Linux kernel

- 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

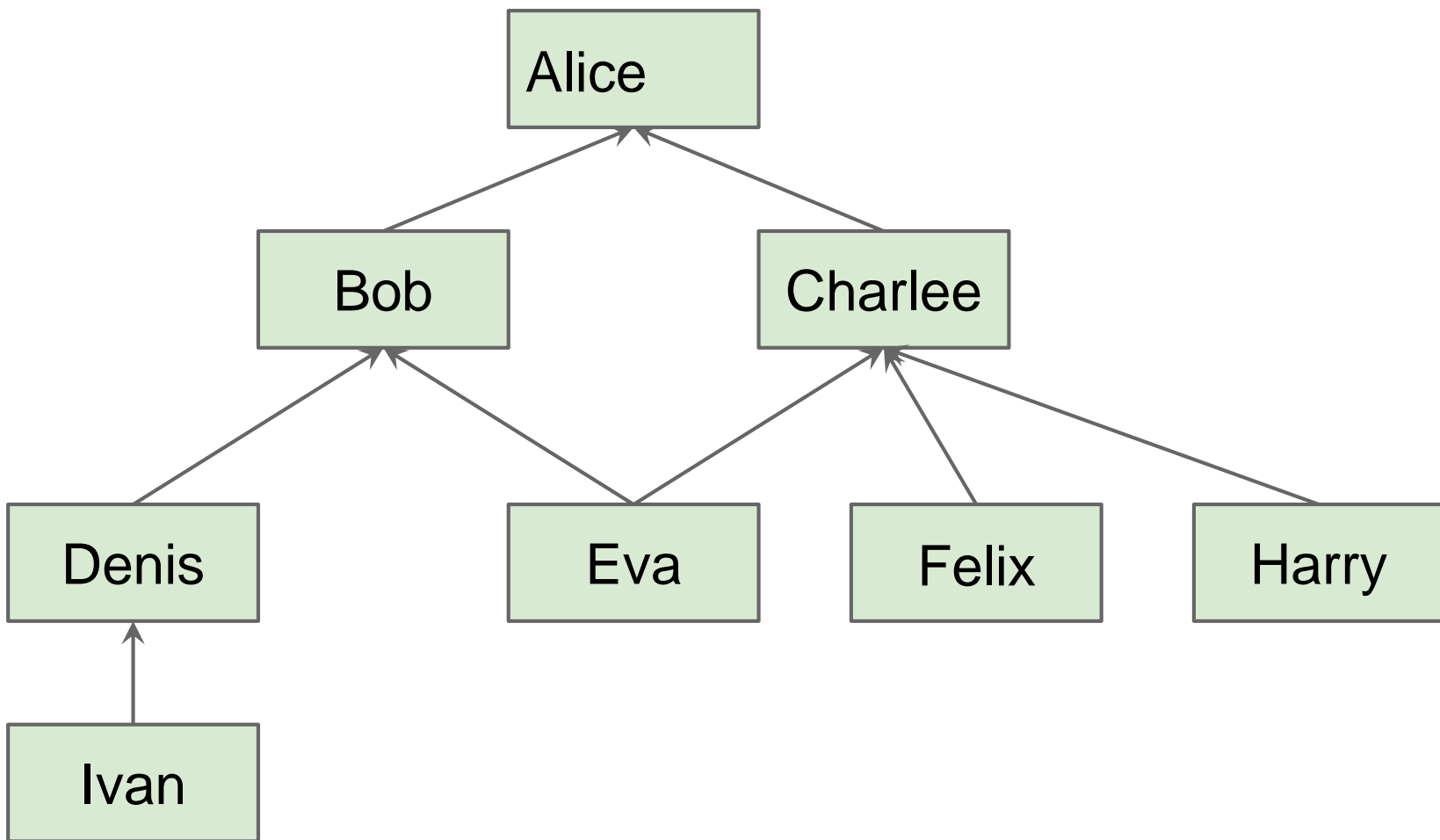


Distributed workflow

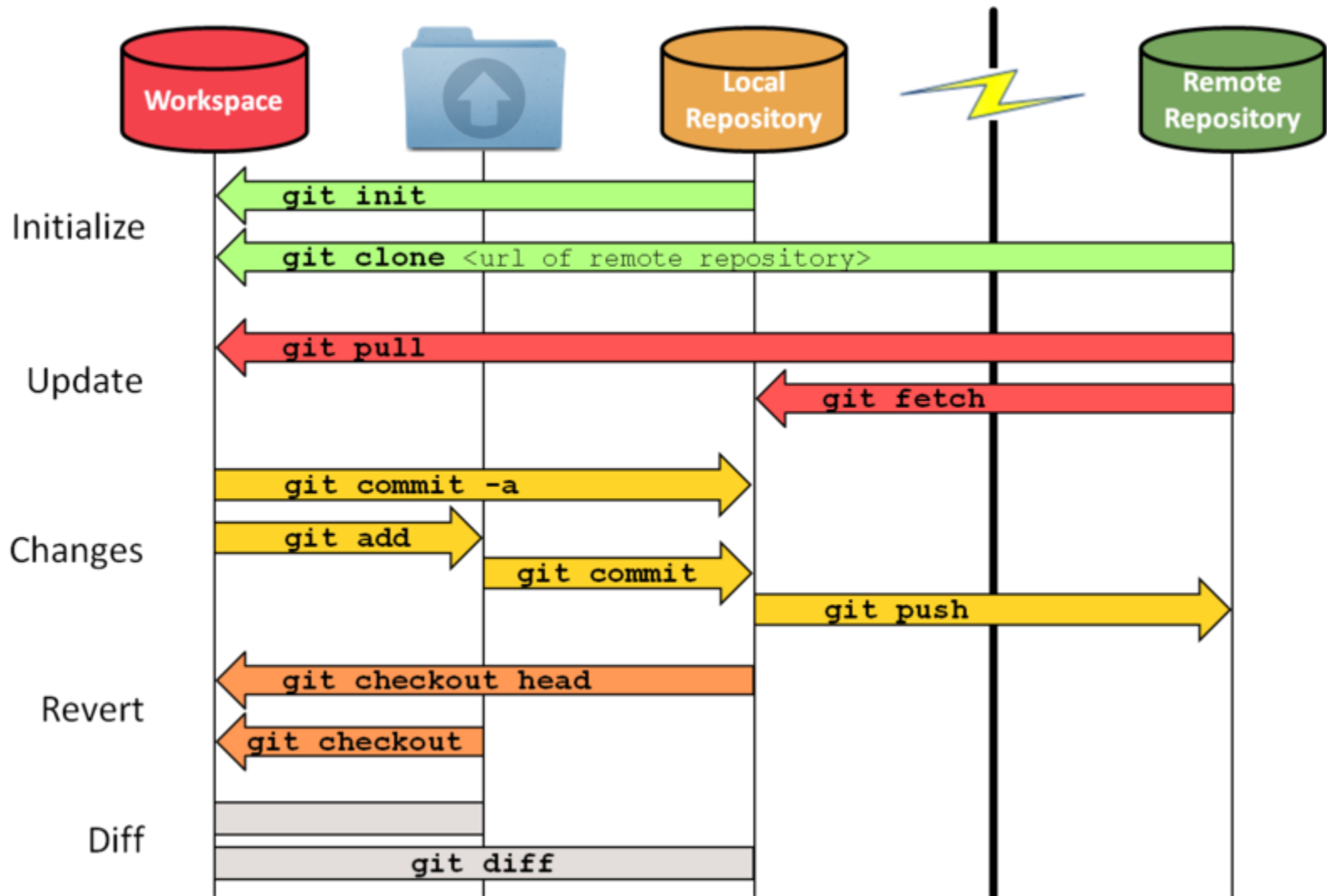


Сеть доверия

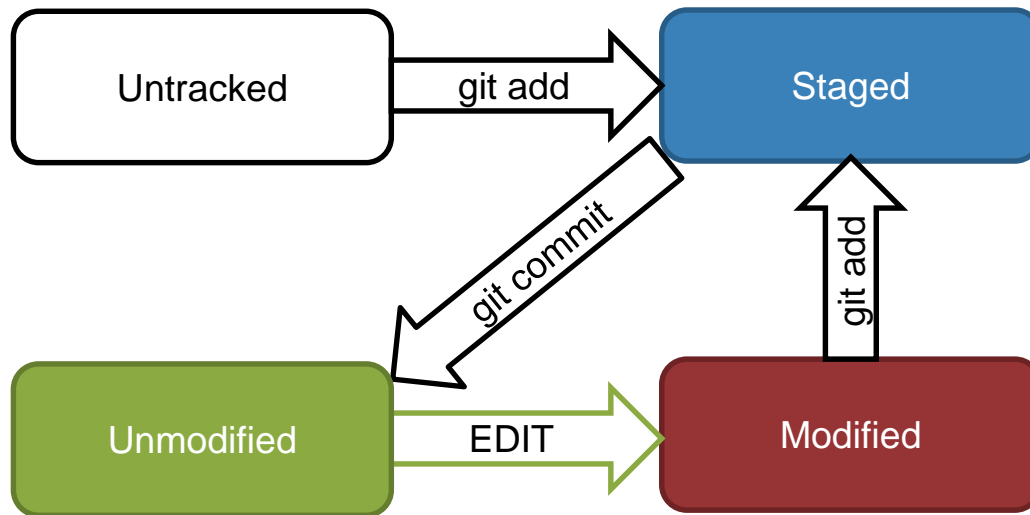
Trust network



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'
```

Требования к commit message

- Комментарий не должен быть пустым.
- Комментарий должен отвечать на вопрос что и зачем было сделано.
- Комментарий должен быть лаконичным (twitter style)
- Ссылки на задачу в багтрекере вполне достаточно
- Комментарий должен быть на английском.

Примеры хороших комментариев:

- Fixed typo in window title
- 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

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
```

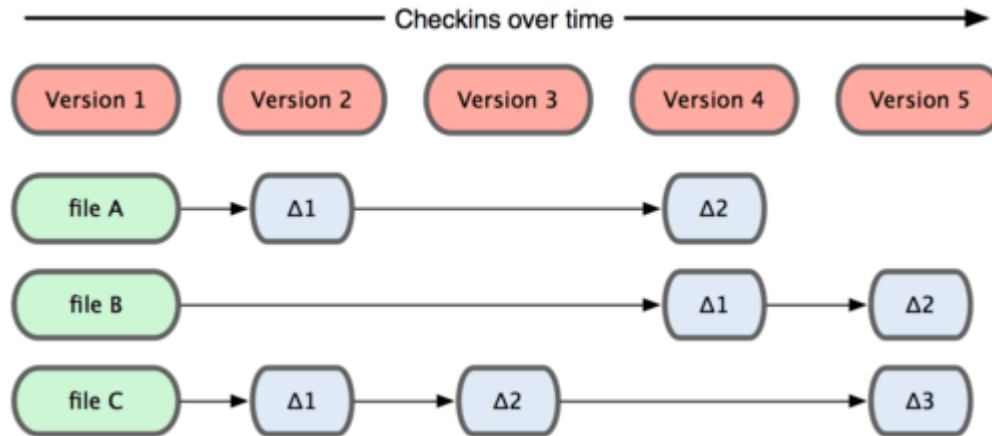
```
d30b6e8 - faster thin lines rendering
```

```
c295b70 - Added .gitignore
```

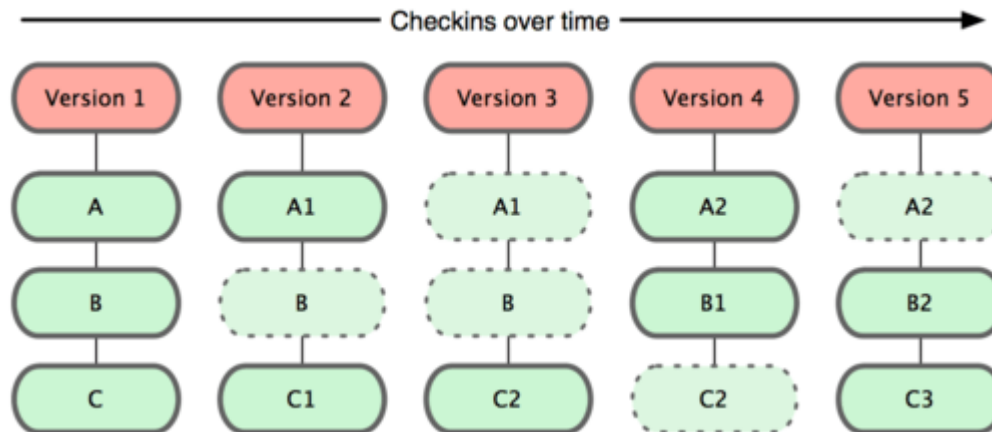
```
4956337 - Checked style with jslint
```


Хранение данных

SVN



git

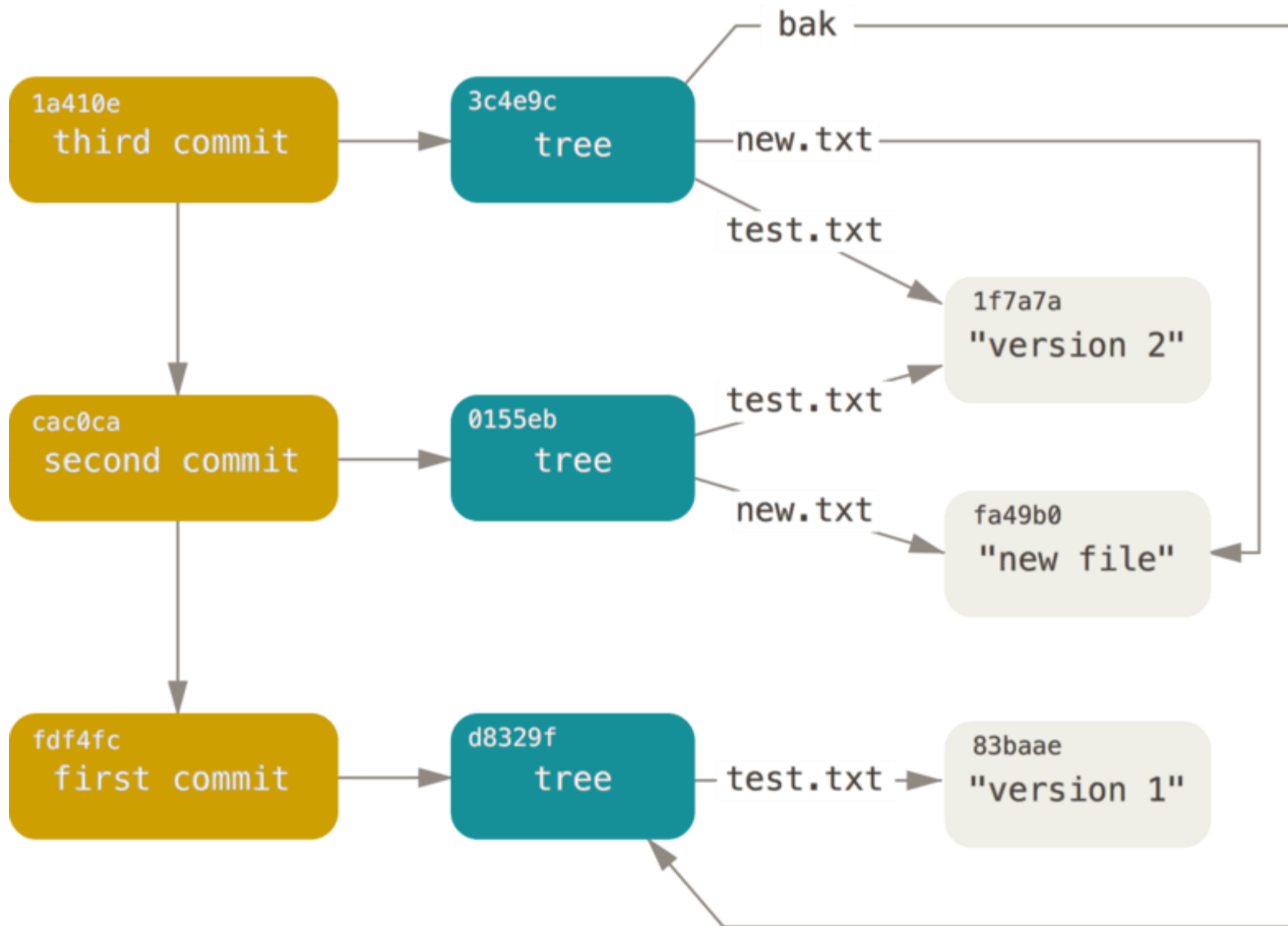


Объекты

- 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

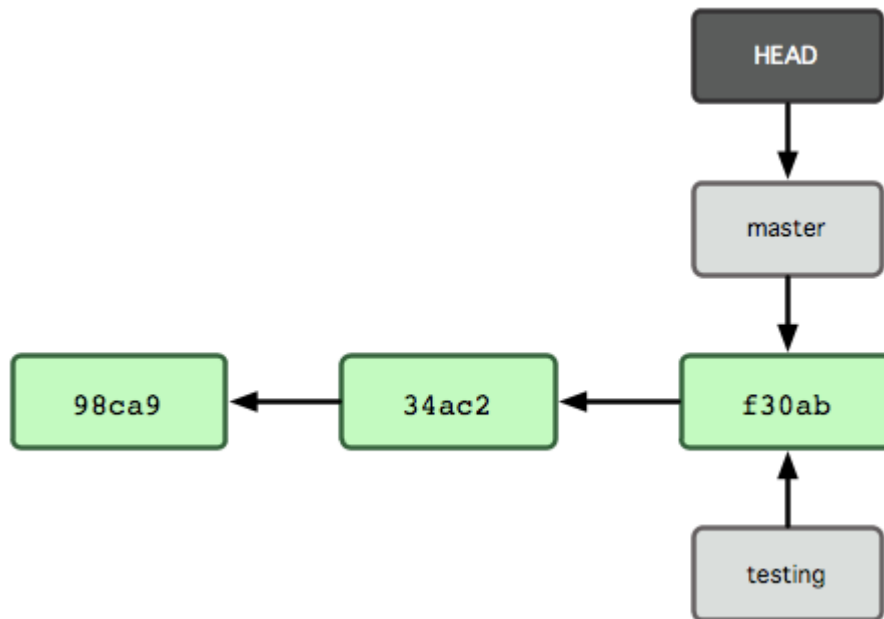
- Release branch. Параллельная разработка и поддержка старых версий.
- Feature branch. Разработка нового функционала без риска сломать работающий код.
- Частые коммиты в собственную ветку.

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

Ветвление

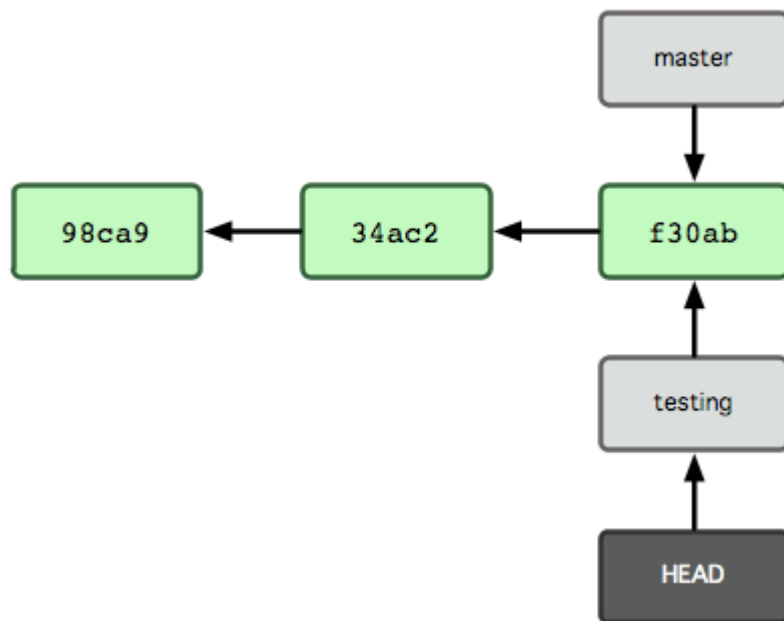
git clone <http://gitlab.dev.ccf.it.nsu.ru/students2016/example.git>

git branch testing



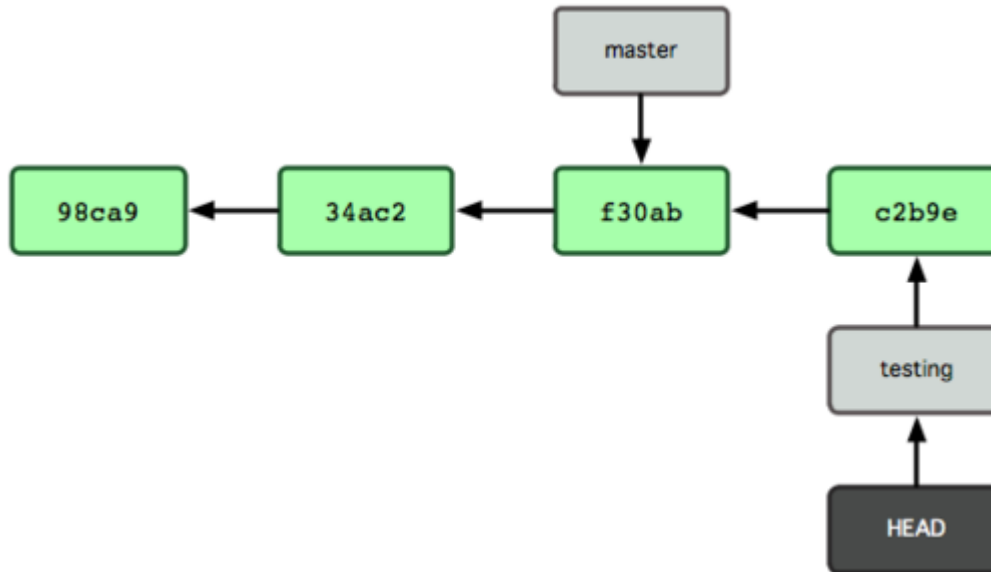
Ветвление

git checkout testing



Ветвление

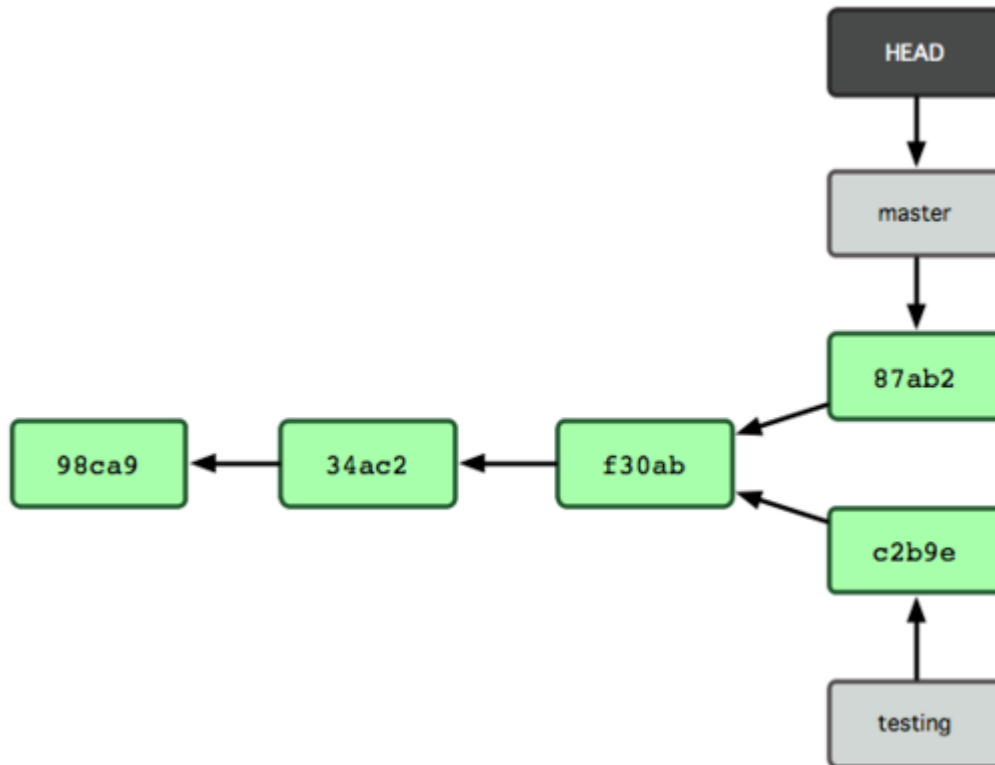
git commit -a -m 'change in the file'



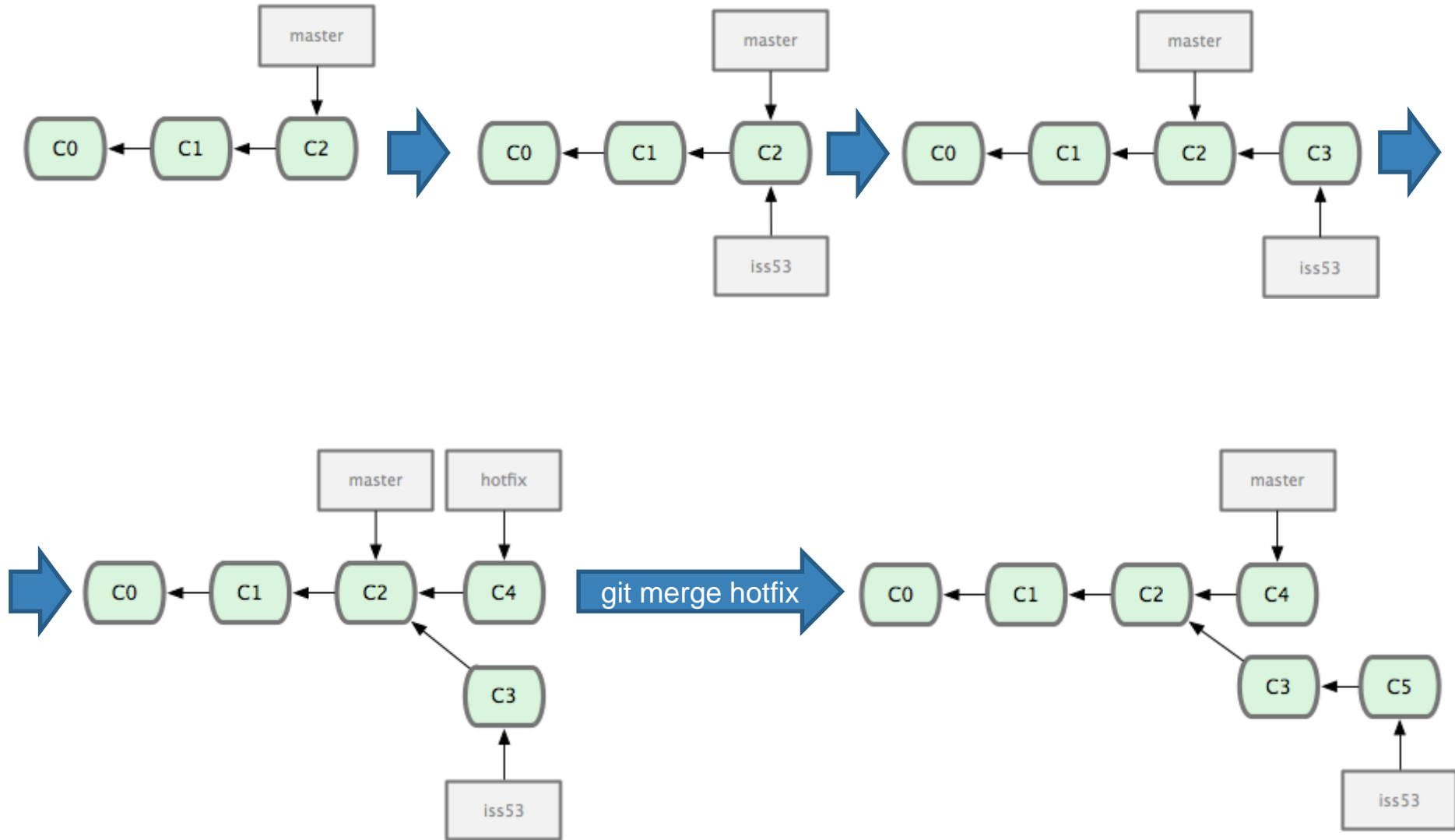
Ветвление

git checkout master

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

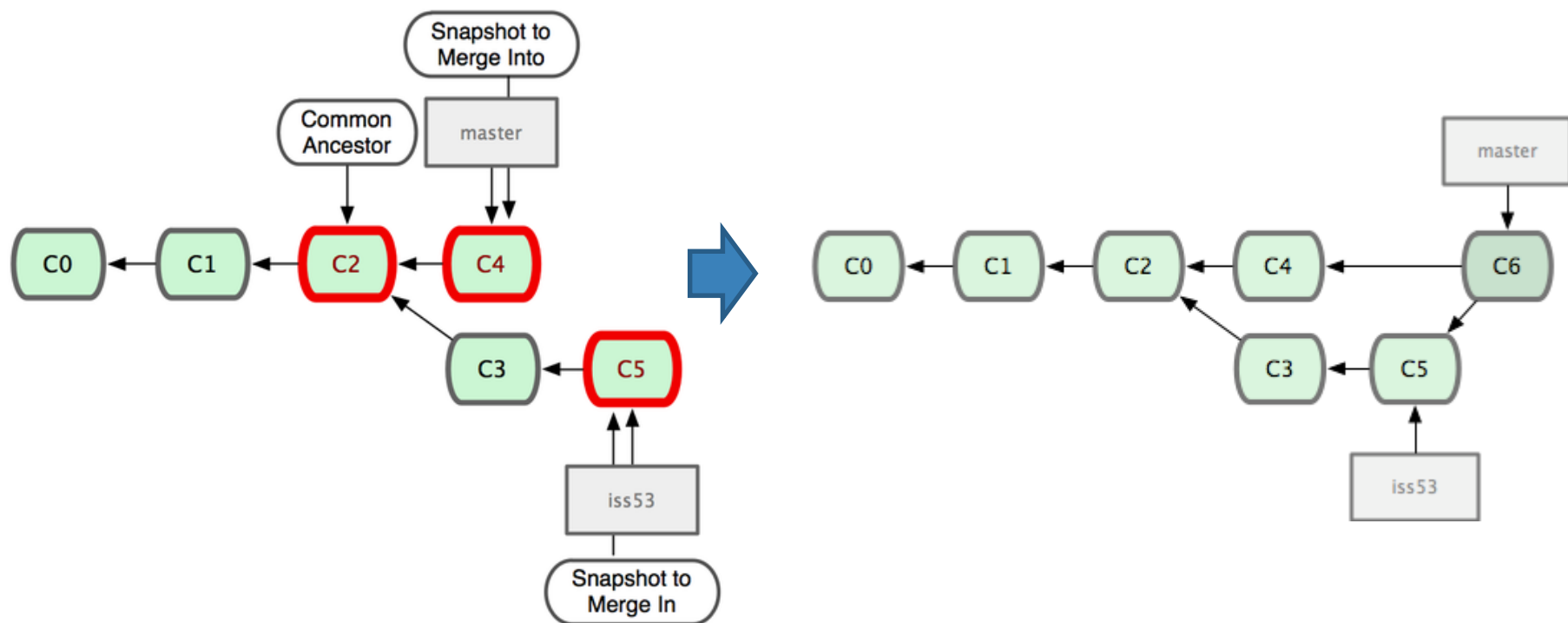


Слияние веток

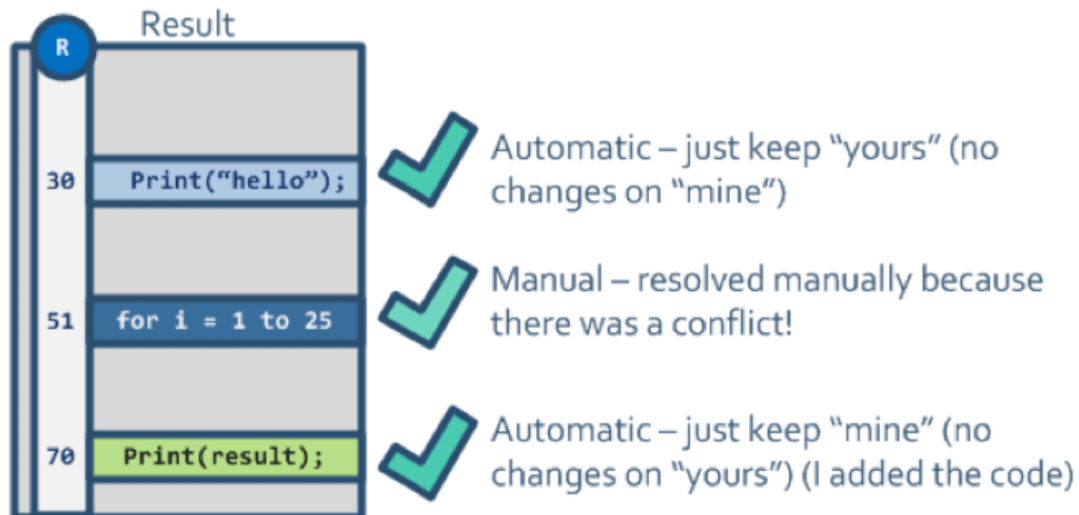
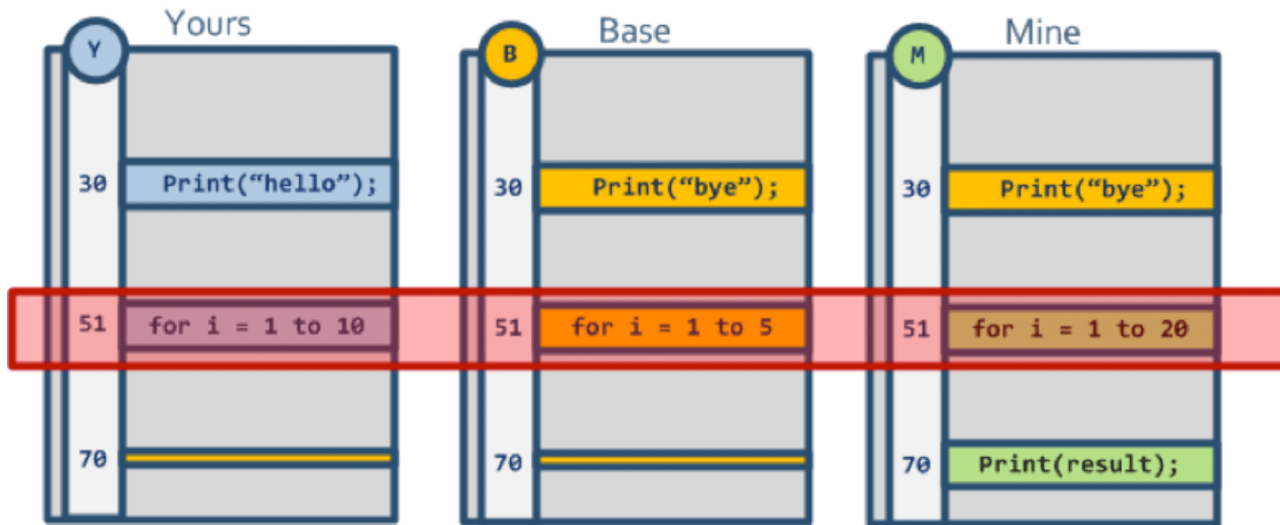


Слияние веток

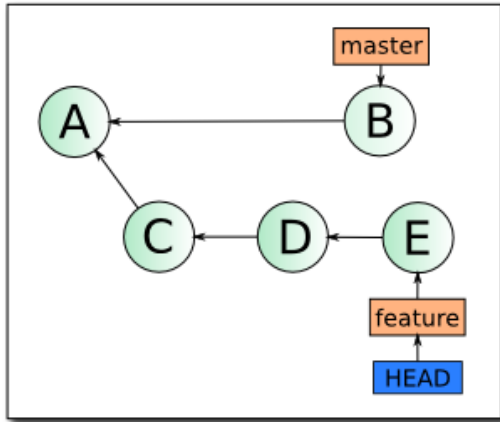
git merge iss53



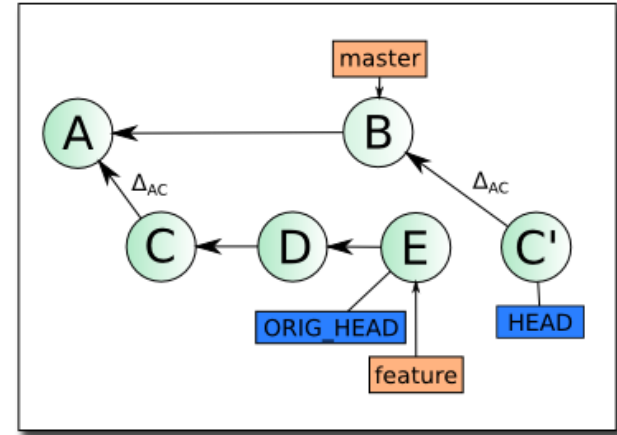
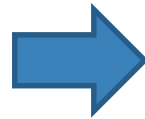
3-way merge



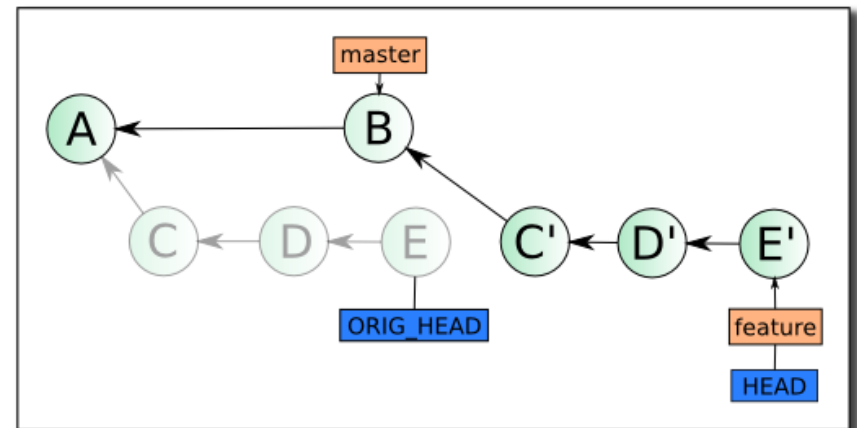
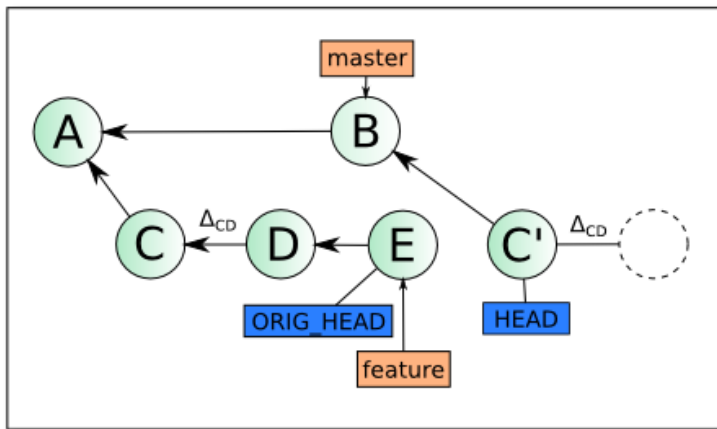
Rebase



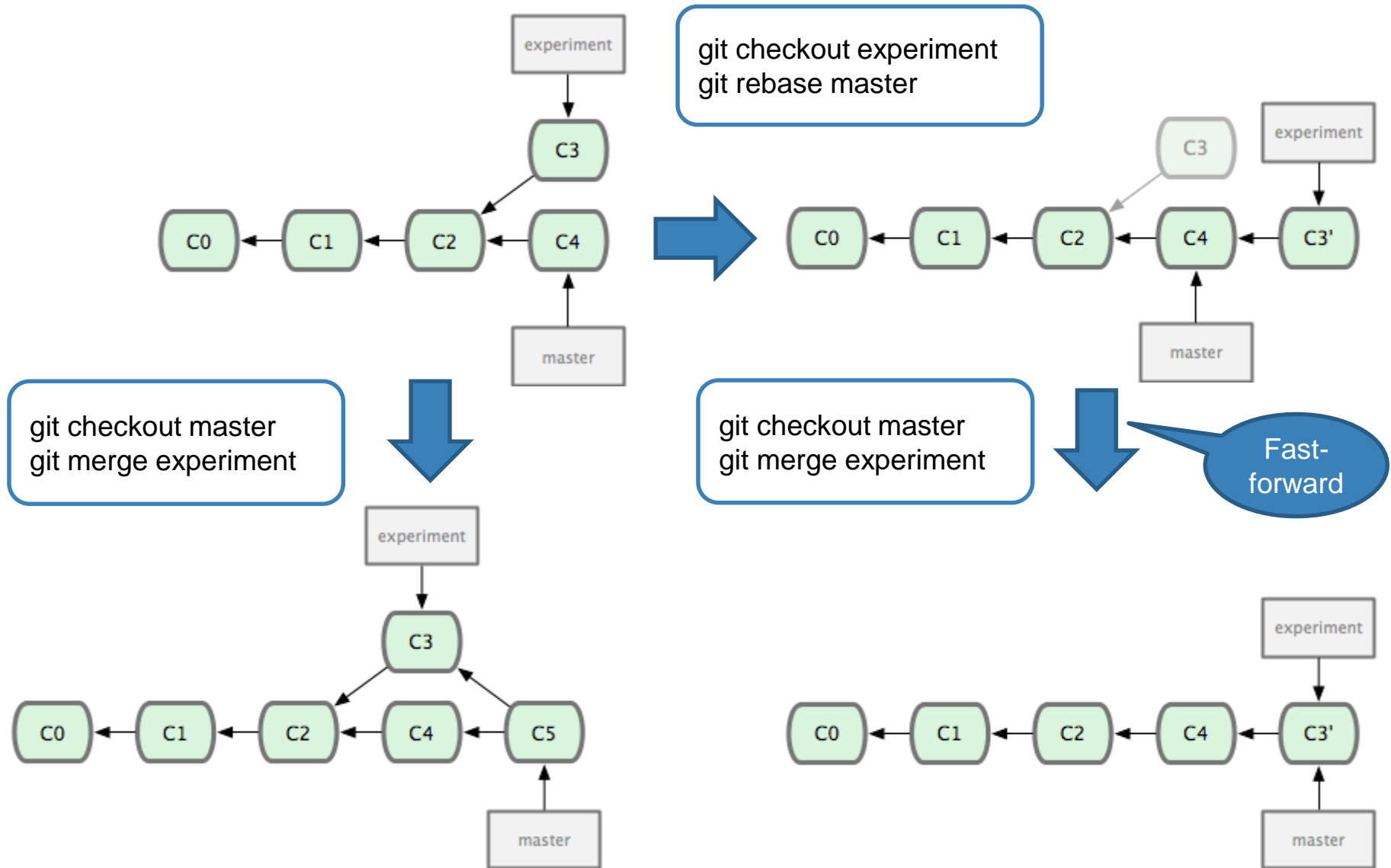
git checkout feature
git rebase master



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



Merge vs Rebase

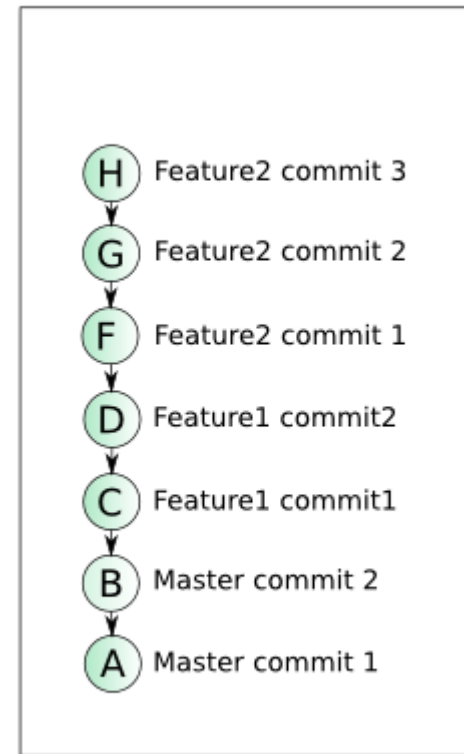
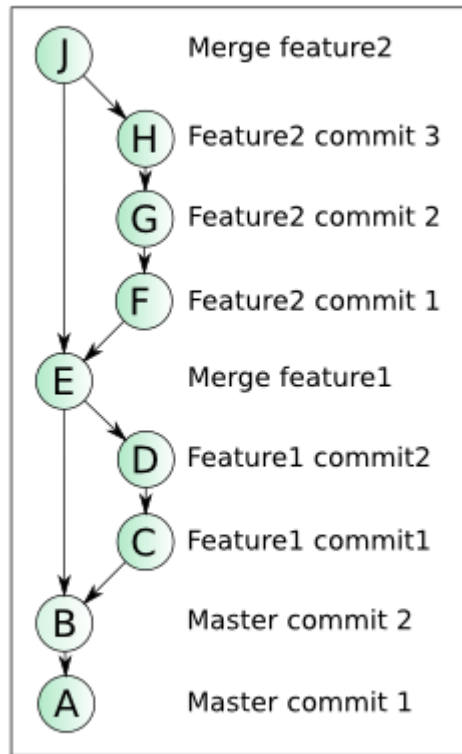


Merge vs Rebase

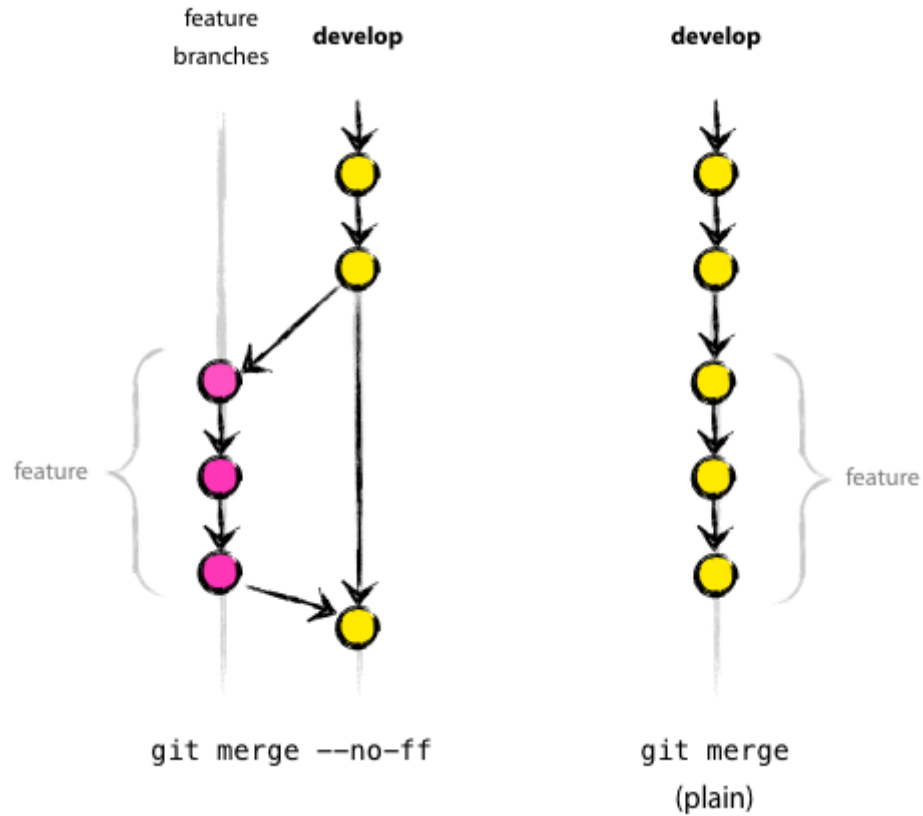
Поскольку git умеет изменять локальную историю, часто используется такой workflow:

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.

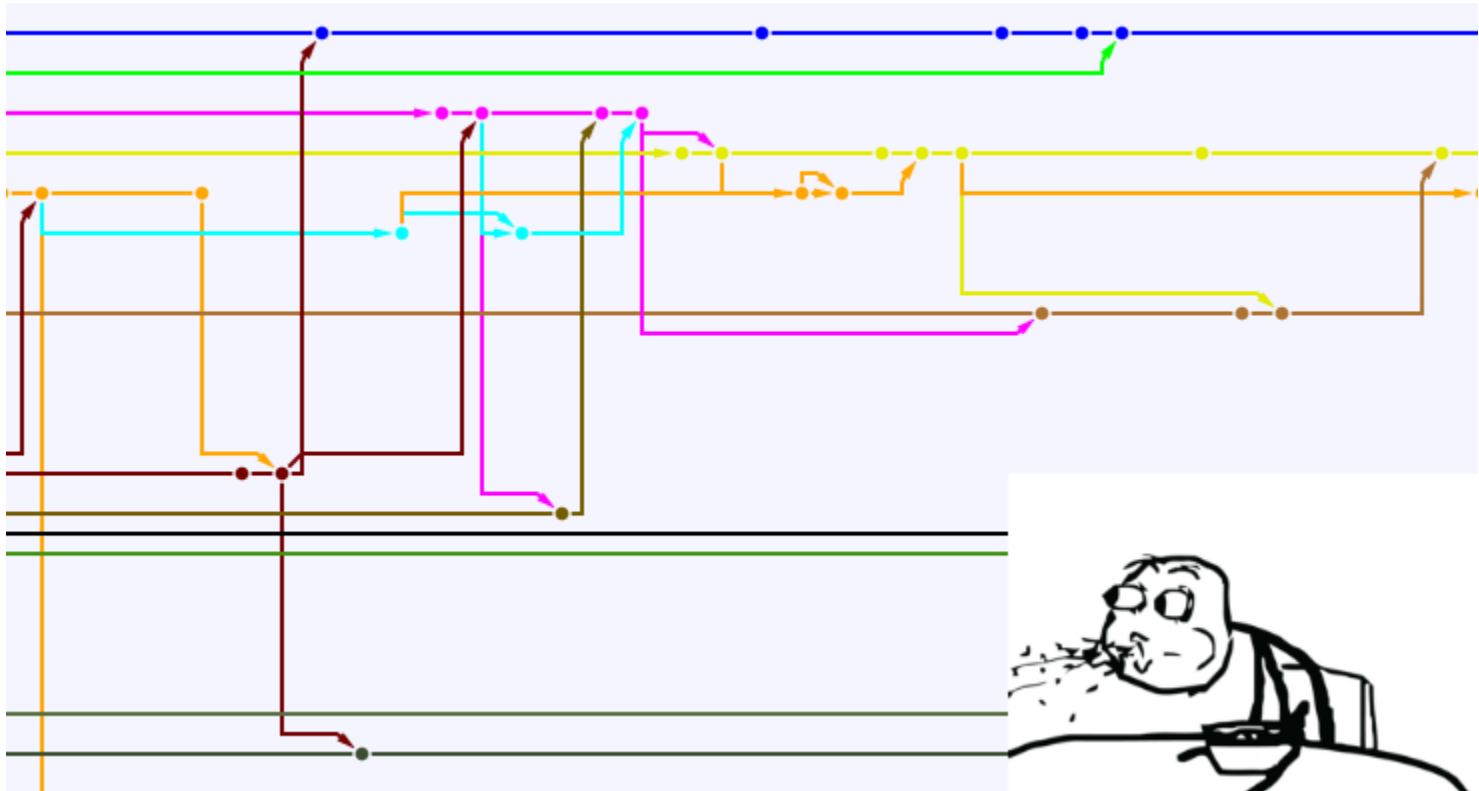
Merge vs Rebase



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

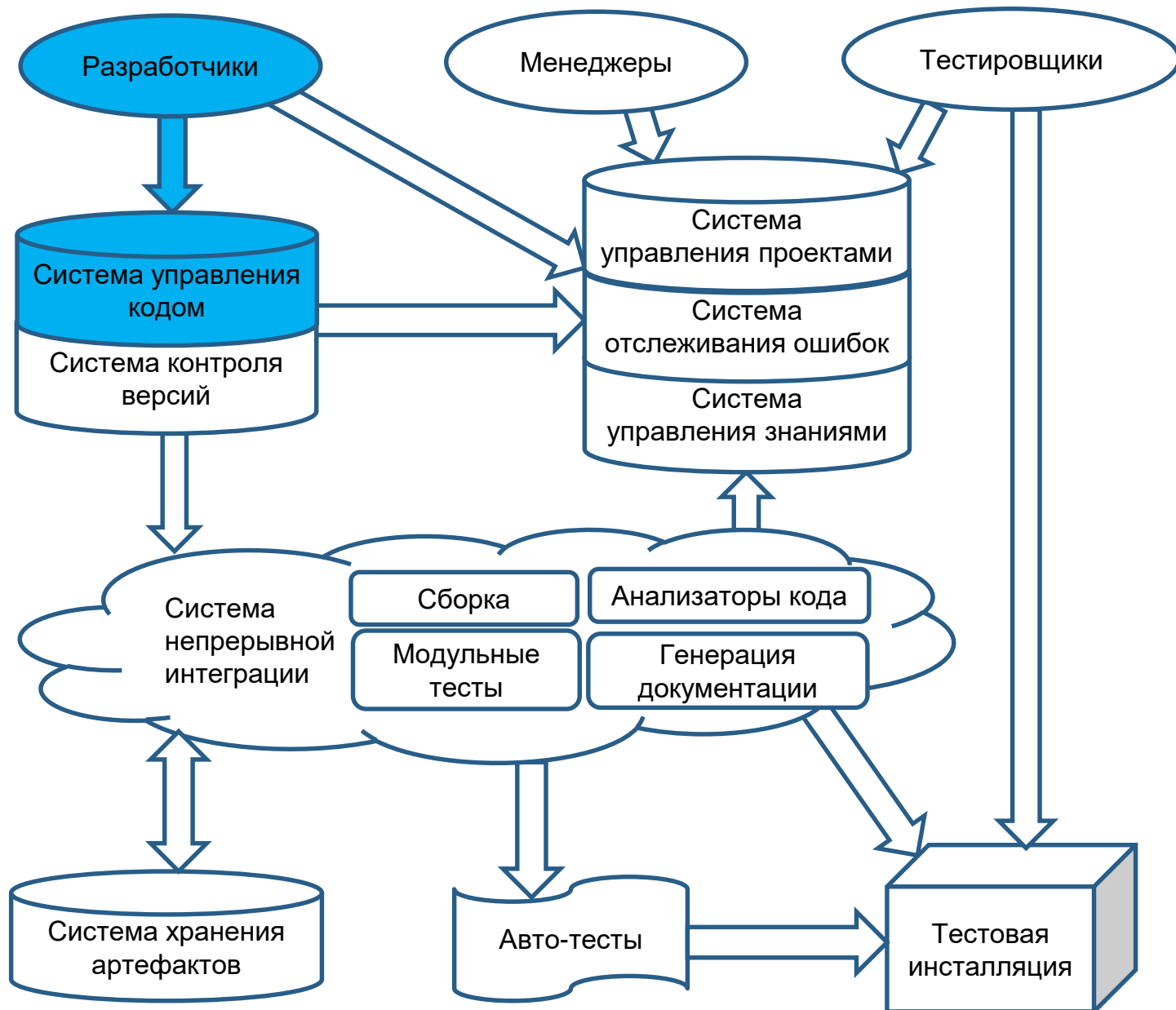
Merge Policy – **простота** использования, **запутанность** истории коммитов

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



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	READ WRITE ADMIN		×
 Alexey E	READ WRITE ADMIN		×
 Denis Anoykin	READ WRITE ADMIN		×

Groups

Select a group	Read		Add
Administrators (asteriosoft:administrators)	READ WRITE ADMIN		×
Developers (asteriosoft:developers)	READ WRITE ADMIN		×

Pull requests

Q

+

<>

Source

⬇

Commits

🔗

Branches

🔗

Pull requests

🔄

Pipelines

☁

Deployments

📄

Downloads

📁

Boards

⚙

Settings

PR-643-Targeting-by-day-parting

Check out ▾View sourceMerge...

master

change destination

PR-643-Targeting-by-day-parting

🔄 1 of 1 passed

📄 Create pull request

🔒 No restrictions edit

👤🌐

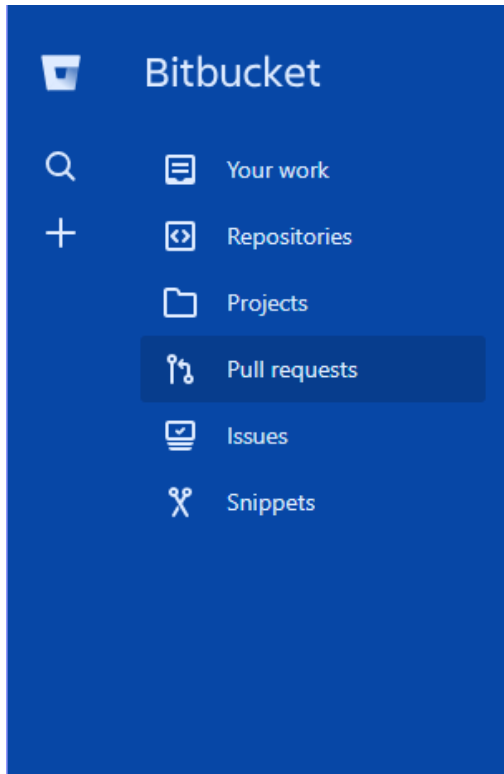
Diff

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





Pull requests

Filter by: **Reviewing** Watching Mine Teams

Repository	Author	Title	Updated	Builds
 bidder	 Alexander Zhuk...	#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 Zhuk...	#535: PR-775 , PR-776 : SSP-b...  2	2 days ago	 
 mango-x	 Alexander Zhuk...	#265: MGO-933: fix vast unwrap h...	2018-08-07	 
 rmx2	 Alexander Zhuk...	#51: EUS-96: opt out cookies	2018-02-15	 
 rmx2	 Alexander Zhuk...	#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  src/api/base/api.ts
+1 -0  src/views/report/stat-report.tsx

 src/api/base/api.ts **MODIFIED**

Side-by-side diff

View file

Comment

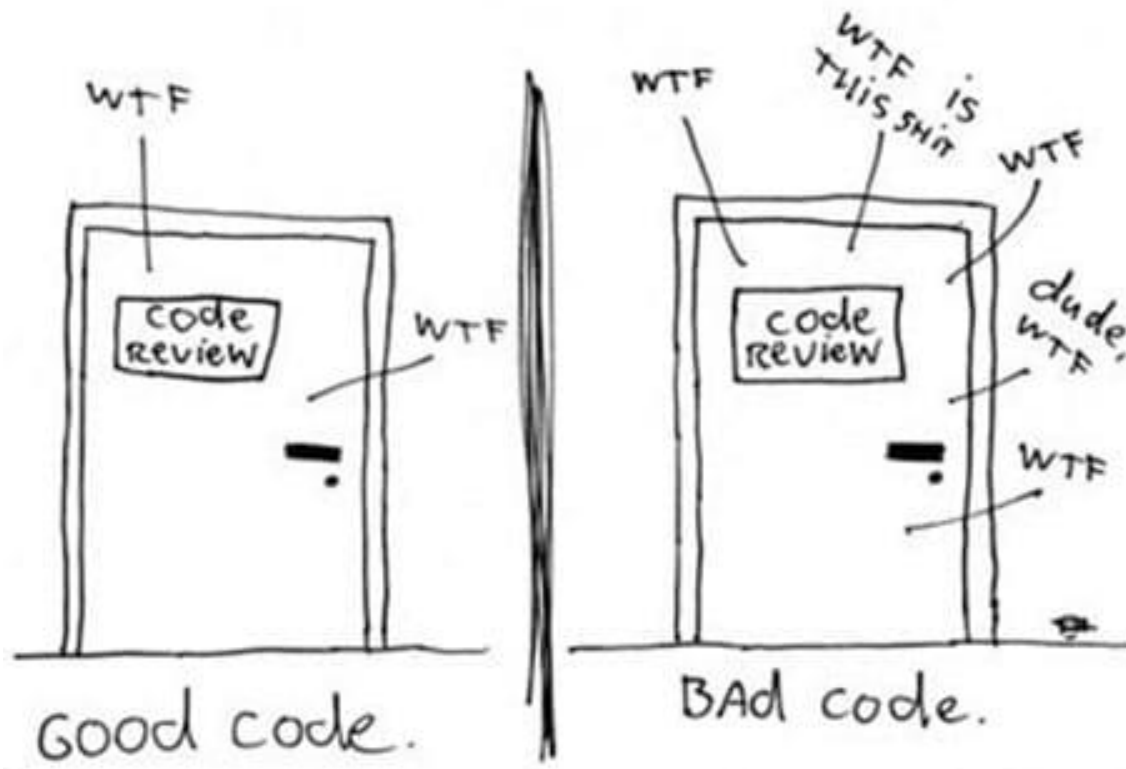
...

...

```
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

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

release/*

Write access

Nobody has write access

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

Merge via pull request

Denis Anoykin x

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

▼ Merge checks

☒ Check for at least 1 approval

☐ Check for unresolved pull request tasks

☒ Check the last commit for 1 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

Save

Merge commit

Merge commit

Squash

Fast forward

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

EditCommentAssignReOpenAdmin

Type:Story

Priority:Medium

Labels:None

Epic Link:Bidder

Sprint:PR Sept 10-11

Status:COMPLETED

Resolution:Done

Assignee:Unassigned

Reporter:Anastasia Cheshkova

Votes:0

Watchers:1

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

Drop files to attach, or browse.

Activity

AllCommentsWork logHistoryActivity

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 branchesUpdated 11/Sep/18 1:03 PM

17 commitsLatest 11/Sep/18 1:04 PM

5 pull requestsMERGEDUpdated 11/Sep/18 1:05 PM

Create branch

Agile

Continuous integration

</> openrtb

<> Source

Commits

Branches

Pull requests

Pipelines

Deployments

Downloads

Boards

Settings

soft / rtb-libraries / openrtb

Pipelines

All branches

OTHER BRANCHES

master MAIN BRANCH

deploy-test

test/deploy

deploy































What's new

Schedules

Caches

Usage

?

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

Рекомендуемая литература

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