Безупречная история в Git или Mercurial

Алексей Хлебников
LVEE 2014

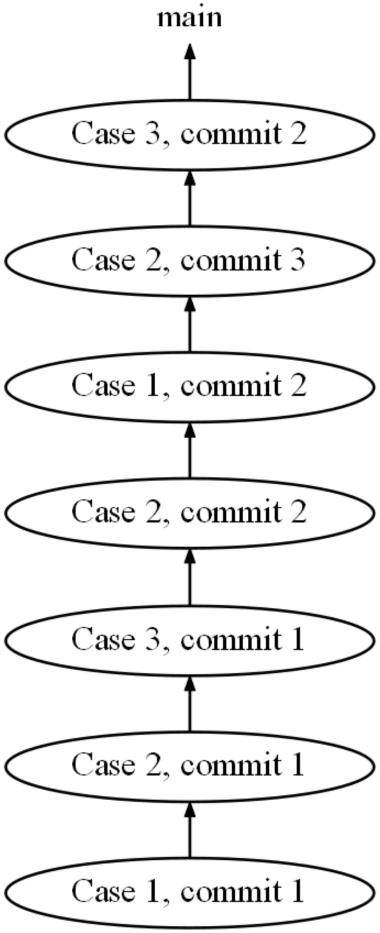
Plan

- Why branching, rebasing and squashing
- HowTo: concrete Git & Hg commands
- Release branches and delivering fixes to several branches
- Bonus 1: Pseudoproblem: too many branches
- Bonus 2: Merge conflicts and Matrix merge

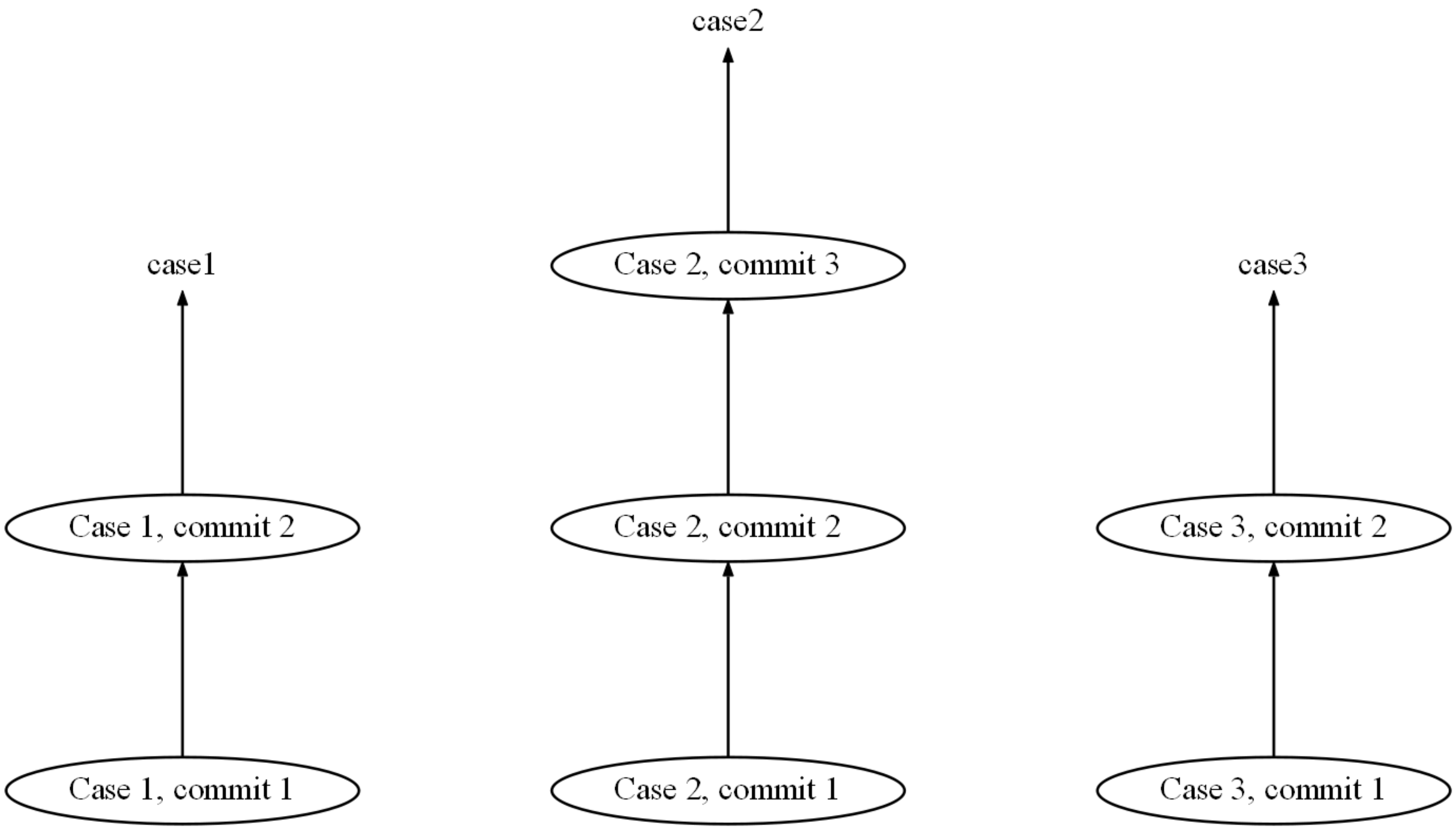
Why branching

- You can freely experiment without affecting others
- Others’ experiments do not affect you
- You make better history in VCS

Without branches: chaos



With branches: order



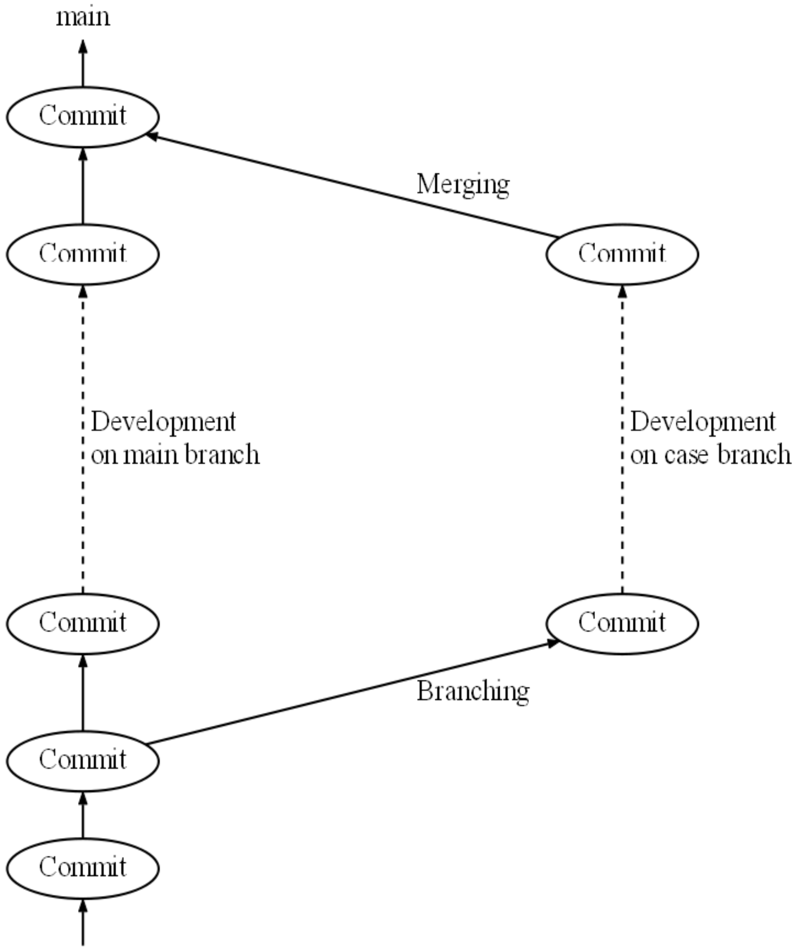
Why rebasing

- Rebasing during development:
  - Up to date with mainline
  - Smaller merge conflicts
  - Testing against updated mainline
  - Contrary to popular belief, possible without forcing after pushing

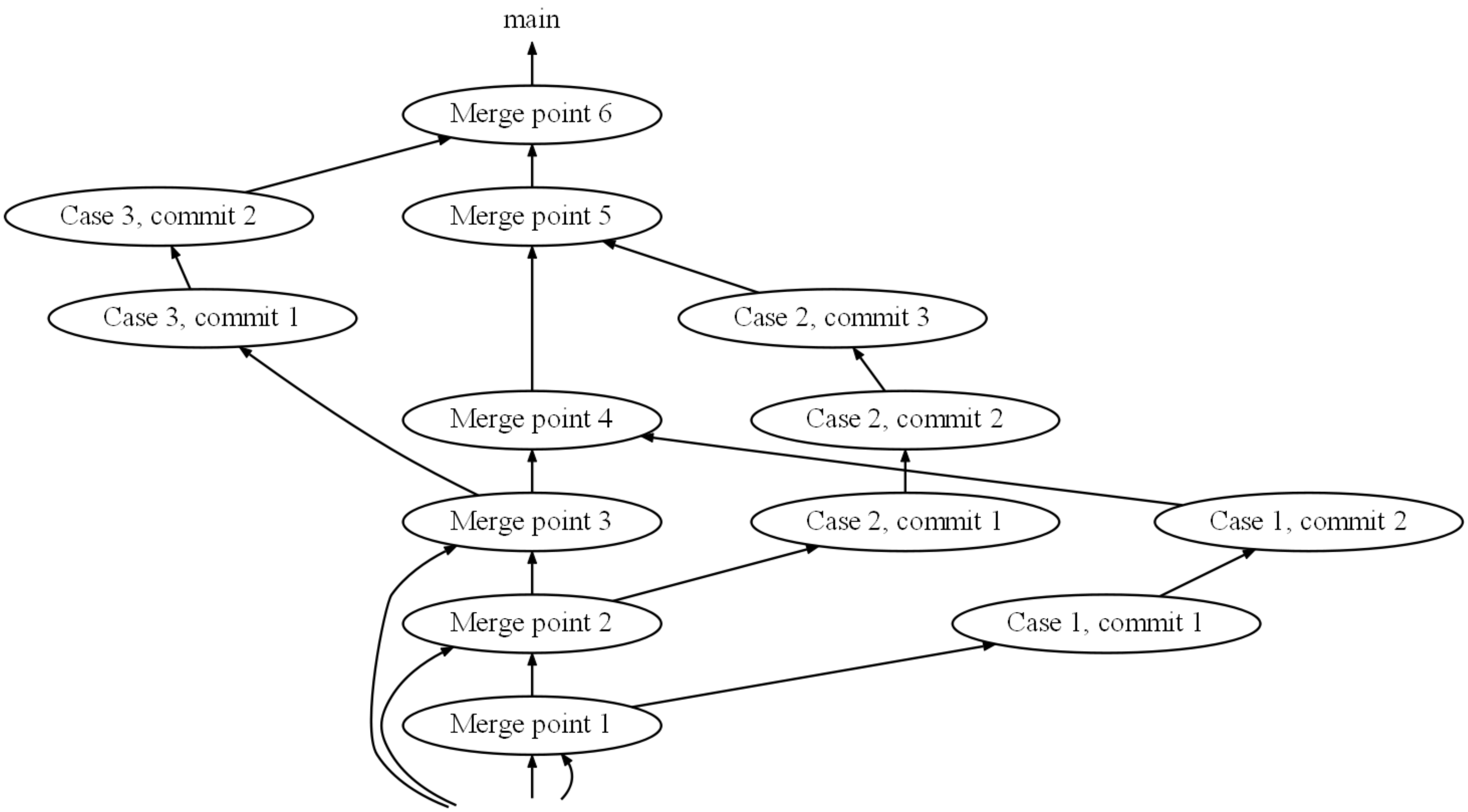
Rebase instead of merge

- Delivery to mainline by rebase instead of merge:
  - Linear history
  - Much easier to read
  - Non-problematic blame and bisect
  - Easier reversal
  - Possibility to remove too old branches (performance)

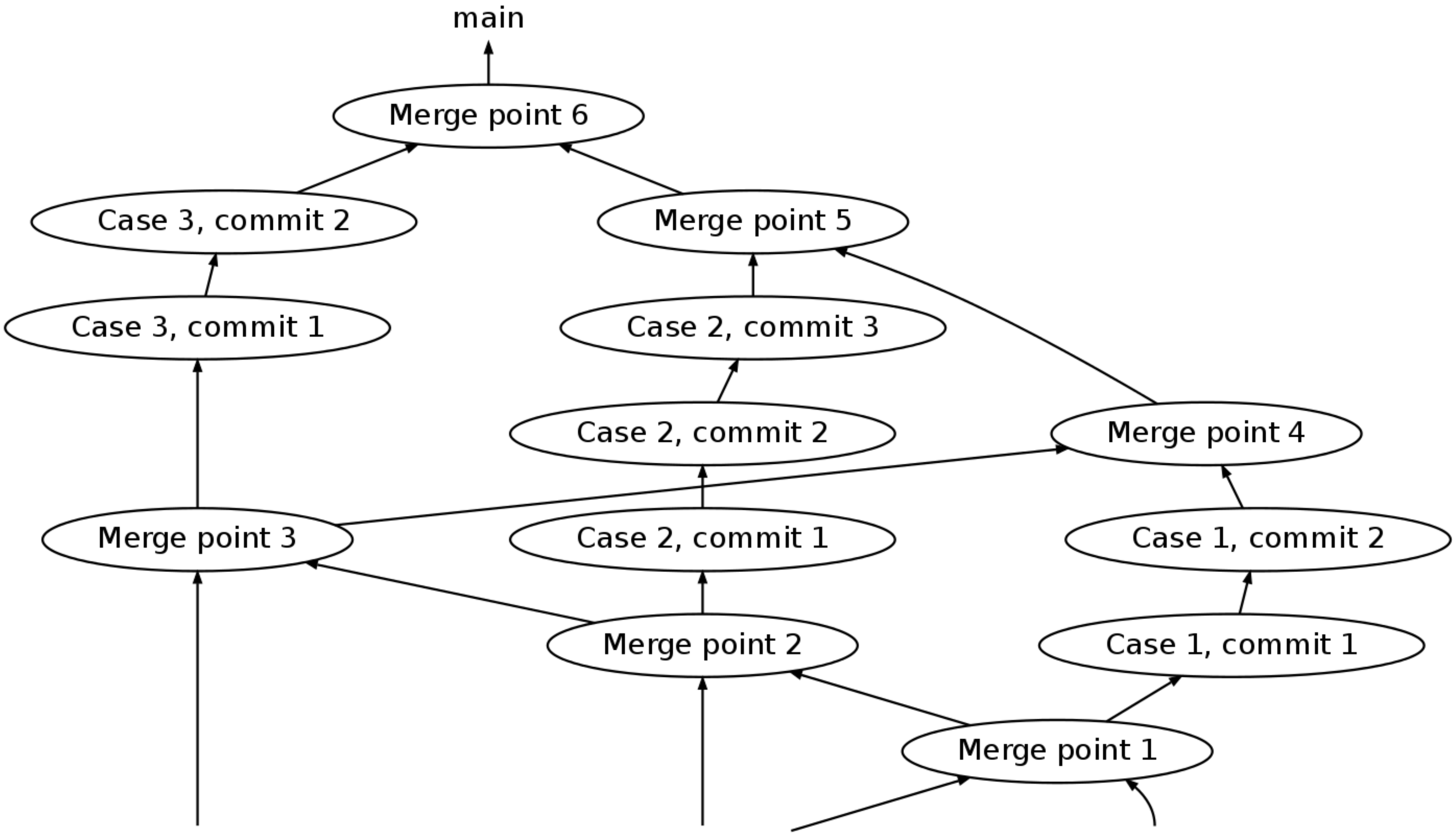
Merging: expectation, order



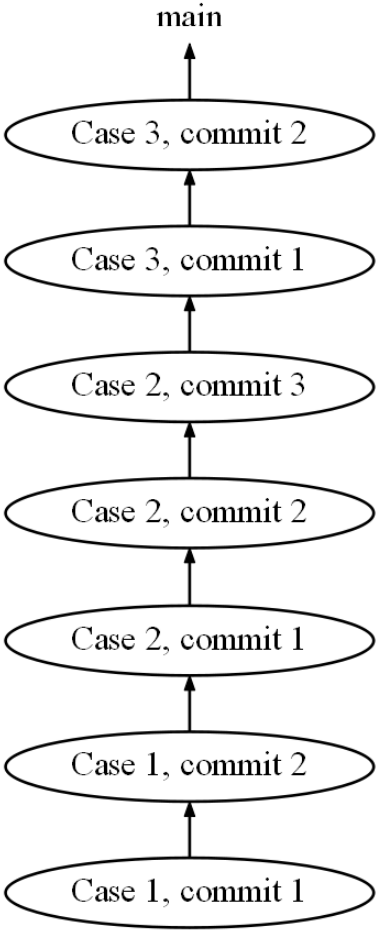
Merging: reality, chaos



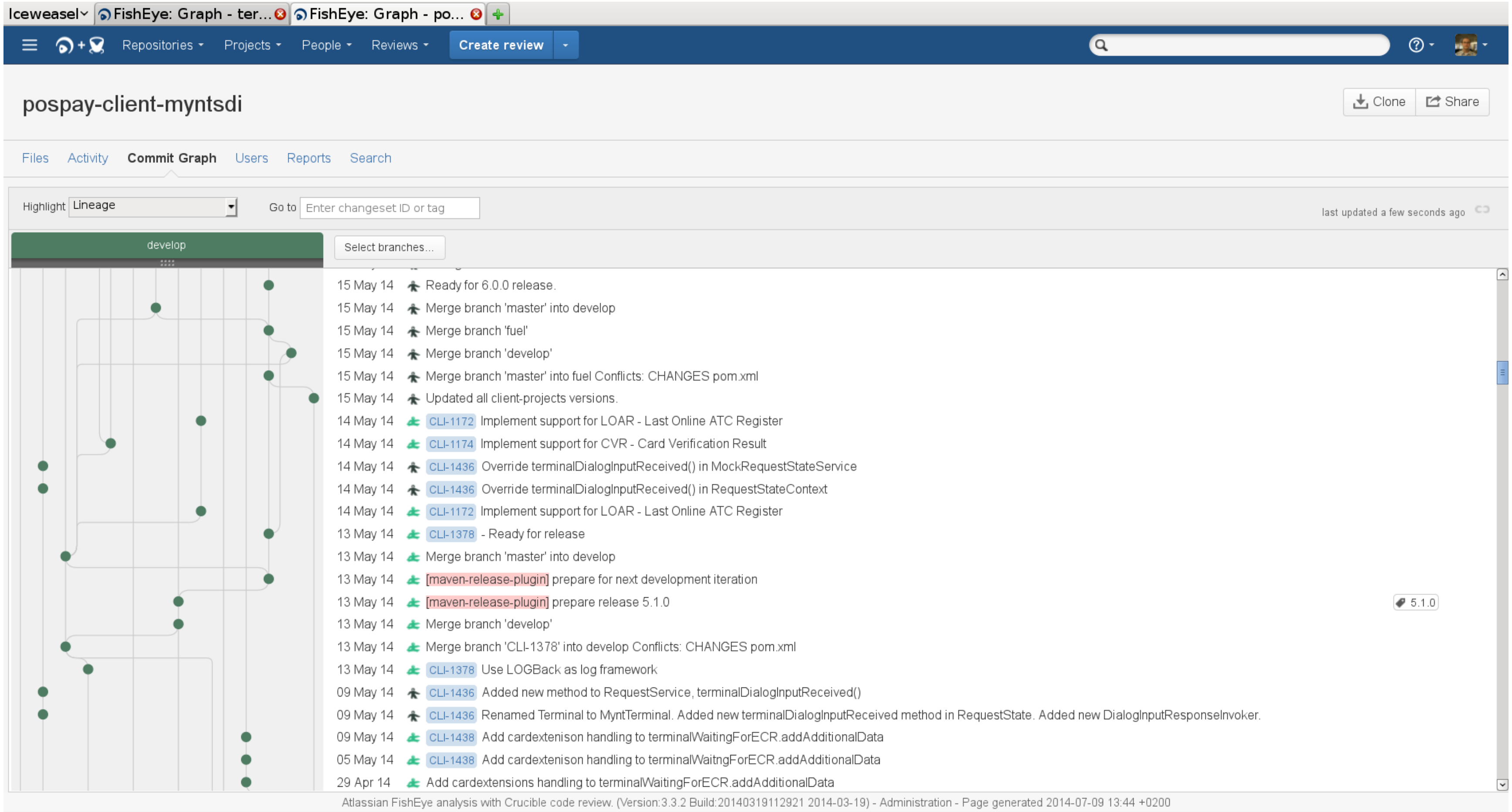
Merging: reality, chaos



Rebasing: order



Merging IRL

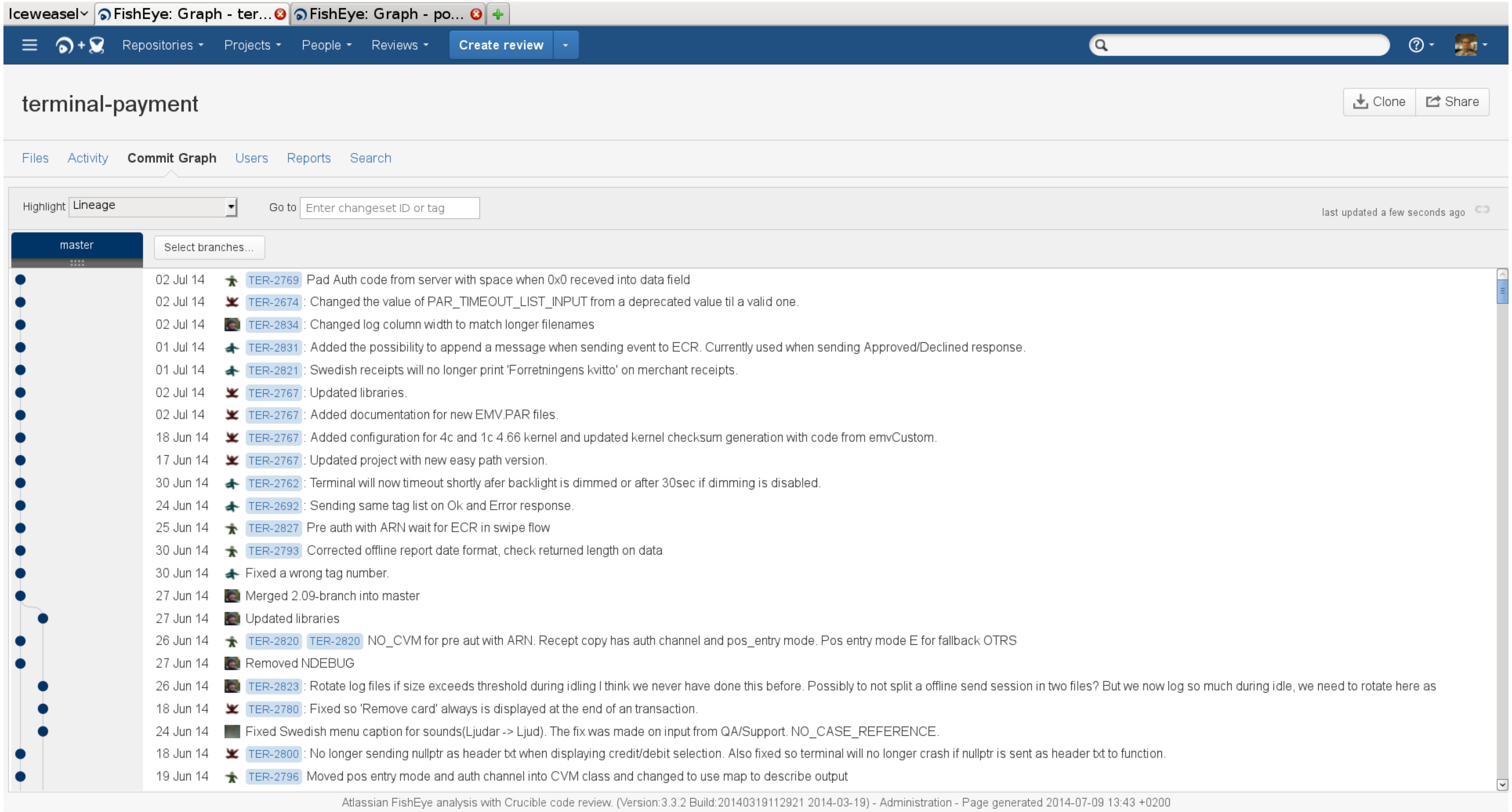


pospay-client-myntsdi

15 May 14 Ready for 6.0.0 release.
15 May 14 Merge branch 'master' into develop
15 May 14 Merge branch 'fuel'
15 May 14 Merge branch 'develop'
15 May 14 Merge branch 'master' into fuel Conflicts: CHANGES pom.xml
15 May 14 Updated all client-projects versions.
14 May 14 CLI-1172 Implement support for LOAR - Last Online ATC Register
14 May 14 CLI-1174 Implement support for CVR - Card Verification Result
14 May 14 CLI-1436 Override terminalDialogInputReceived() in MockRequestStateService
14 May 14 CLI-1436 Override terminalDialogInputReceived() in RequestStateContext
14 May 14 CLI-1172 Implement support for LOAR - Last Online ATC Register
13 May 14 CLI-1378 - Ready for release
13 May 14 Merge branch 'master' into develop
13 May 14 [maven-release-plugin] prepare for next development iteration
13 May 14 [maven-release-plugin] prepare release 5.1.0 5.1.0
13 May 14 Merge branch 'develop'
13 May 14 Merge branch 'CLI-1378' into develop Conflicts: CHANGES pom.xml
13 May 14 CLI-1378 Use LOGBack as log framework
09 May 14 CLI-1436 Added new method to RequestService, terminalDialogInputReceived()
09 May 14 CLI-1436 Renamed Terminal to MyntTerminal. Added new terminalDialogInputReceived method in RequestState. Added new DialogInputResponseInvoker.
09 May 14 CLI-1438 Add cardextenison handling to terminalWaitingForECR.addAdditionalData
05 May 14 CLI-1438 Add cardextenison handling to terminalWaitngForECR.addAdditionalData
29 Apr 14 Add cardextensions handling to terminalWaitingForECR.addAdditionalData

Atlassian FishEye analysis with Crucible code review. (Version:3.3.2 Build:20140319112921 2014-03-19) - Administration - Page generated 2014-07-09 13:44 +0200

Rebasing IRL



terminal-payment

Highlight: Lineage

Go to: Enter changeset ID or tag

last updated a few seconds ago

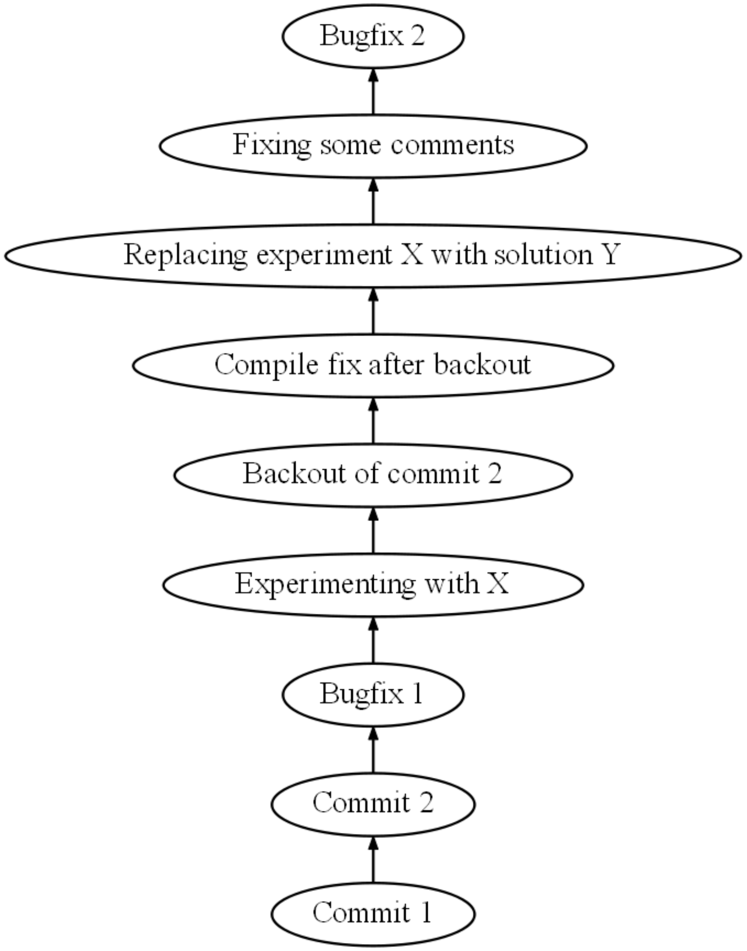
```
02 Jul 14  TER-2769 Pad Auth code from server with space when 0x0 receved into data field
02 Jul 14  TER-2674 : Changed the value of PAR_TIMEOUT_LIST_INPUT from a deprecated value til a valid one.
02 Jul 14  TER-2834 : Changed log column width to match longer filenames
01 Jul 14  TER-2831 : Added the possibility to append a message when sending event to ECR. Currently used when sending Approved/Declined response.
01 Jul 14  TER-2821 : Swedish receipts will no longer print 'Forretningens kvitto' on merchant receipts.
02 Jul 14  TER-2767 : Updated libraries.
02 Jul 14  TER-2767 : Added documentation for new EMV.PAR files.
18 Jun 14  TER-2767 : Added configuration for 4c and 1c 4.66 kernel and updated kernel checksum generation with code from emvCustom.
17 Jun 14  TER-2767 : Updated project with new easy path version.
30 Jun 14  TER-2762 : Terminal will now timeout shortly afer backlight is dimmed or after 30sec if dimming is disabled.
24 Jun 14  TER-2692 : Sending same tag list on Ok and Error response.
25 Jun 14  TER-2827 Pre auth with ARN wait for ECR in swipe flow
30 Jun 14  TER-2793 Corrected offline report date format, check returned length on data
30 Jun 14  Fixed a wrong tag number.
27 Jun 14  Merged 2.09-branch into master
27 Jun 14  Updated libraries
26 Jun 14  TER-2820 TER-2820 NO_CVM for pre aut with ARN. Recept copy has auth channel and pos_entry mode. Pos entry mode E for fallback OTRS
27 Jun 14  Removed NDEBUG
26 Jun 14  TER-2823 : Rotate log files if size exceeds threshold during idling I think we never have done this before. Possibly to not split a offline send session in two files? But we now log so much during idle, we need to rotate here as
18 Jun 14  TER-2780 : Fixed so 'Remove card' always is displayed at the end of an transaction.
24 Jun 14  Fixed Swedish menu caption for sounds(Ljudar -> Ljud). The fix was made on input from QA/Support. NO_CASE_REFERENCE.
18 Jun 14  TER-2800 : No longer sending nullptr as header txt when displaying credit/debit selection. Also fixed so terminal will no longer crash if nullptr is sent as header txt to function.
19 Jun 14  TER-2796 Moved pos entry mode and auth channel into CVM class and changed to use map to describe output
```

Atlassian FishEye analysis with Crucible code review. (Version:3.3.2 Build:20140319112921 2014-03-19) - Administration - Page generated 2014-07-09 13:43 +0200

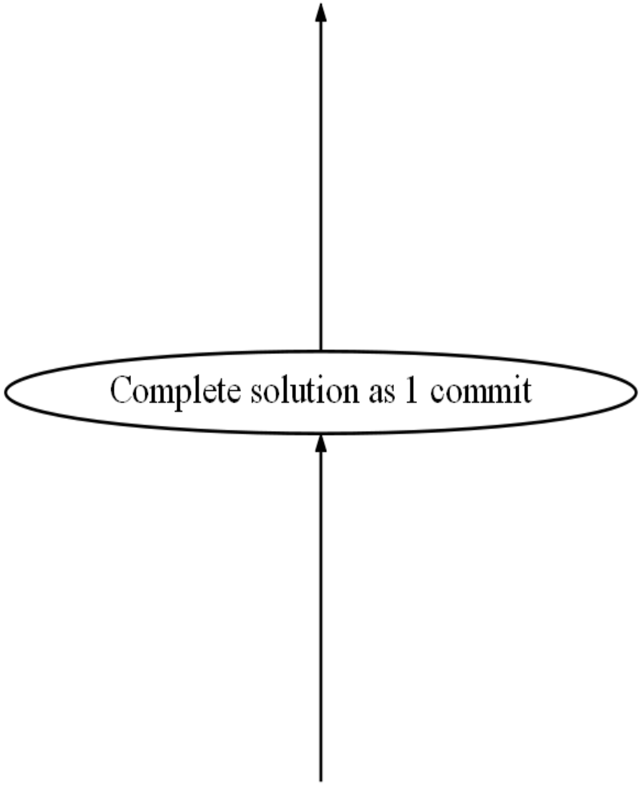
Why squashing

- Compact history
- No garbage in history
- Much more readable history
- Easier reversal
- Contrary to popular belief, possible without forcing after pushing

Not squashing: chaos

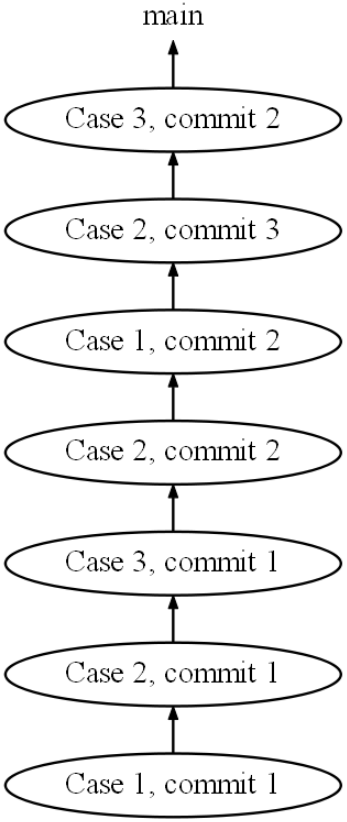


Squashing: order



Before vs After

Before: chaos



main

Case 3, commit 2

Case 2, commit 3

Case 1, commit 2

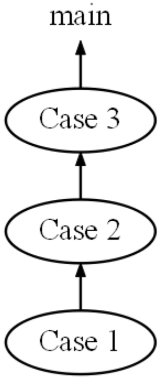
Case 2, commit 2

Case 3, commit 1

Case 2, commit 1

Case 1, commit 1

After: order



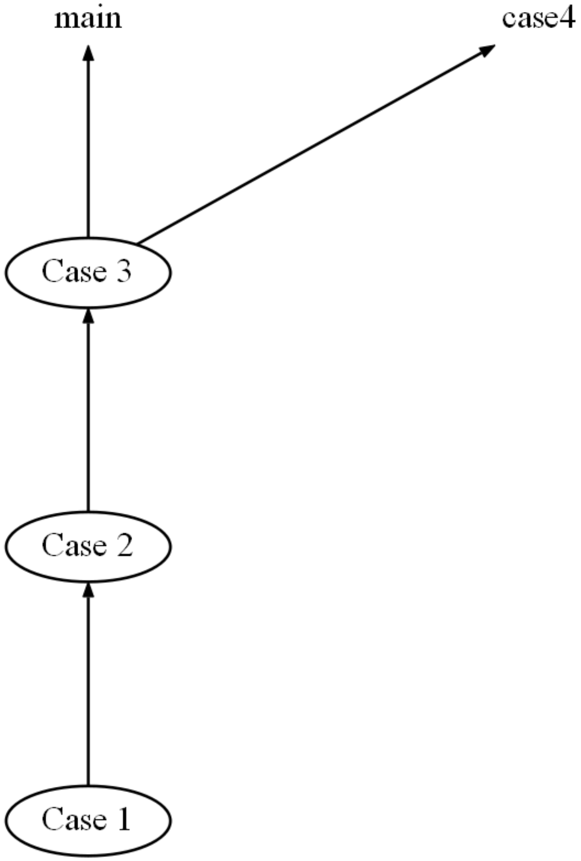
main

Case 3

Case 2

Case 1

Step 1: make a branch

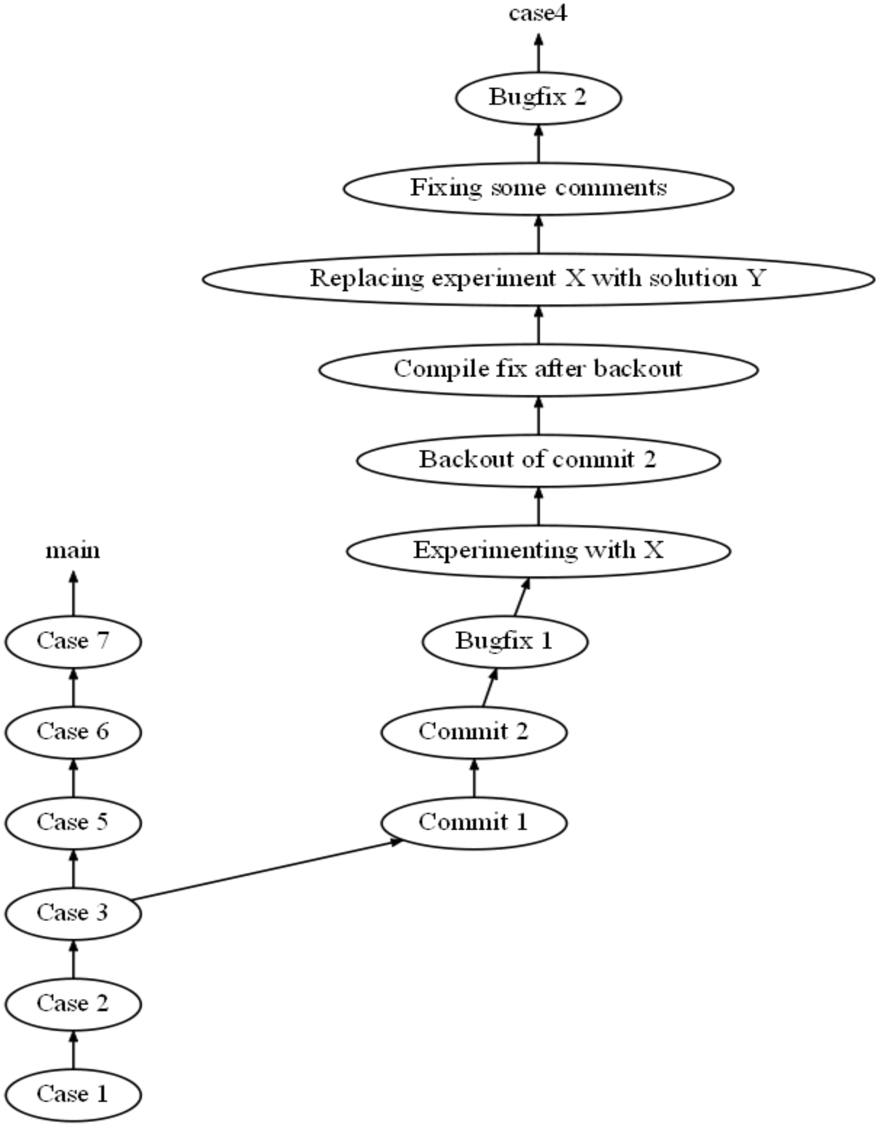


Step 1: make a branch

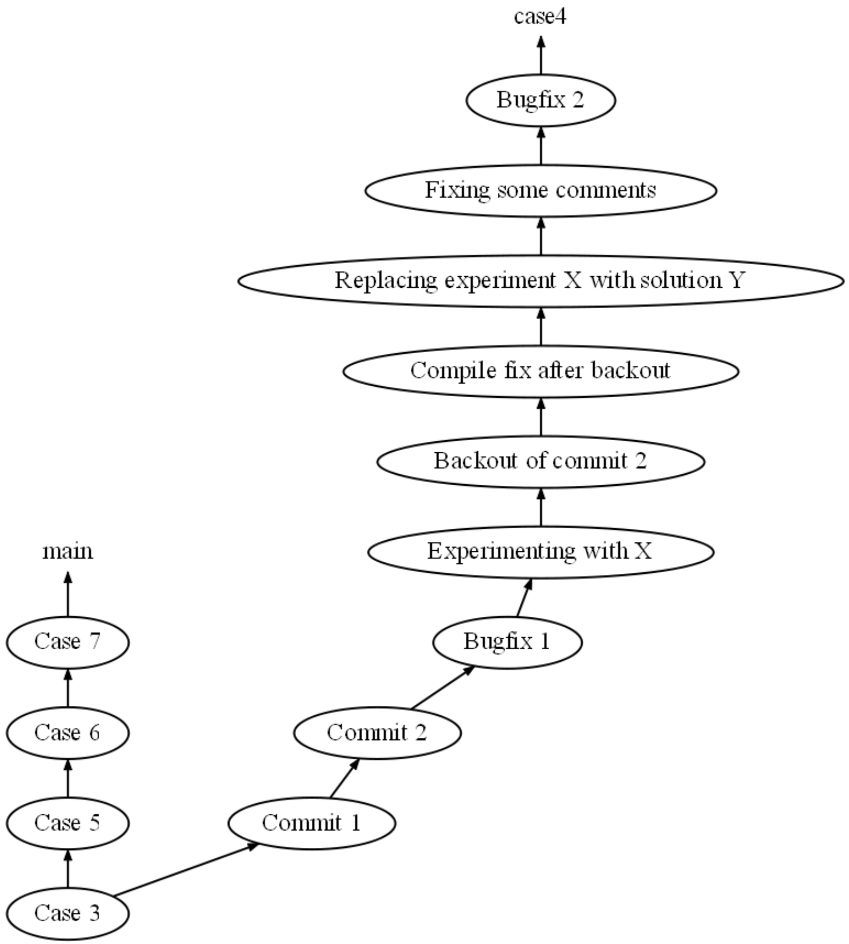
- Git
  - git checkout –b case4

- Mercurial
  - hg book case4

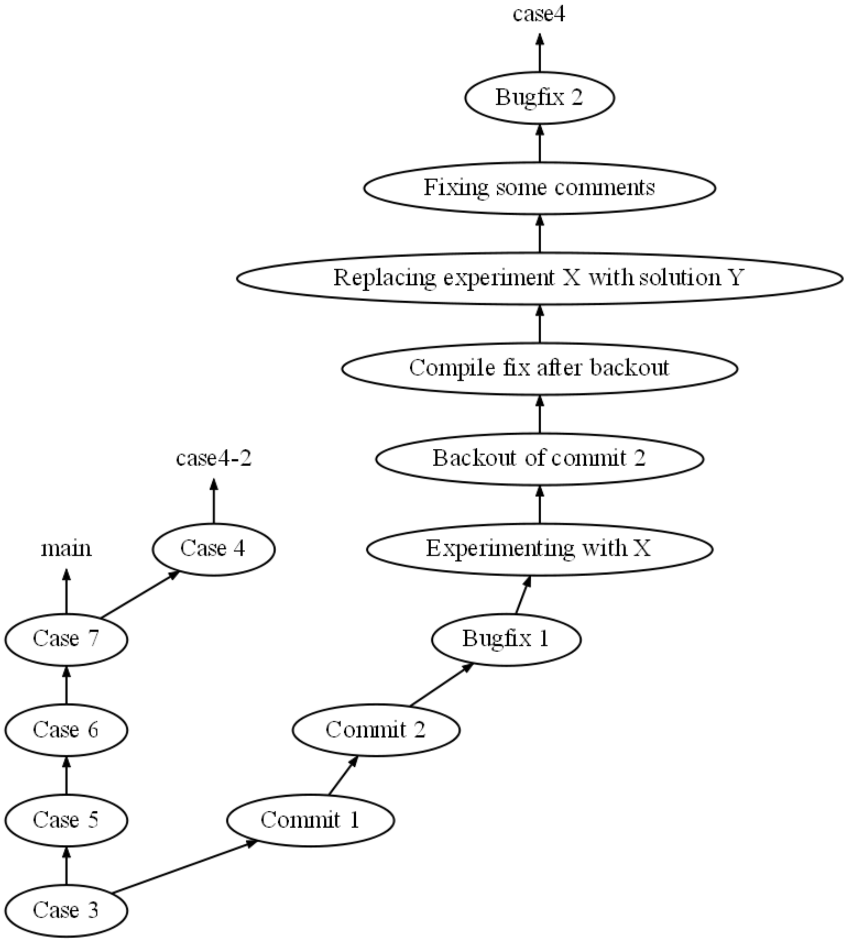
Step 2: develop



Step 3: rebase and squash



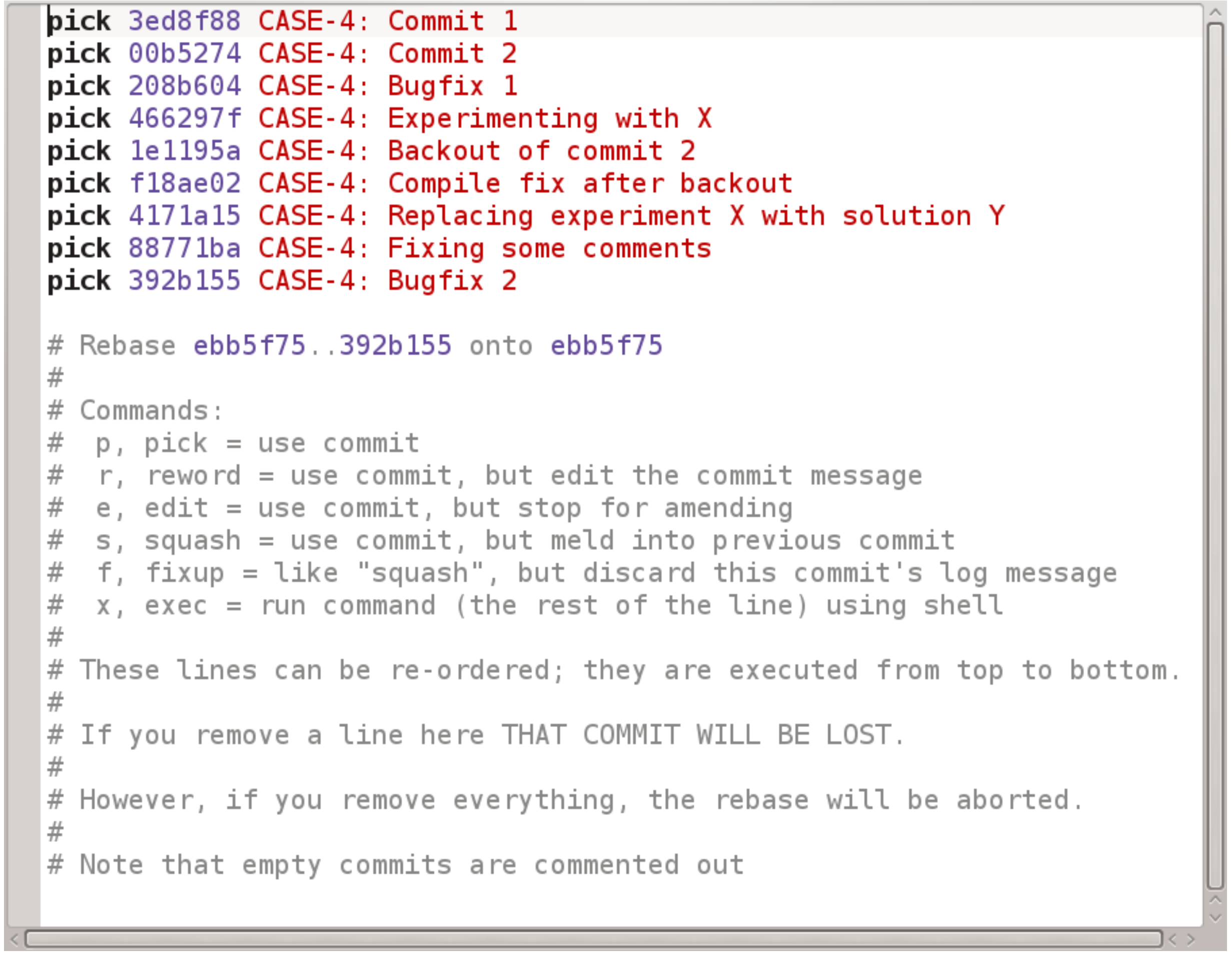
Step 3: rebase and squash



Step 3: rebase and squash

- Git
  - git checkout –b **case4-2**
  - git rebase --interactive main
- Mercurial
  - You need Rebase and Histedit extension
  - hg rebase --keep --dest main
  - hg histedit main
  - hg book **case4-2**

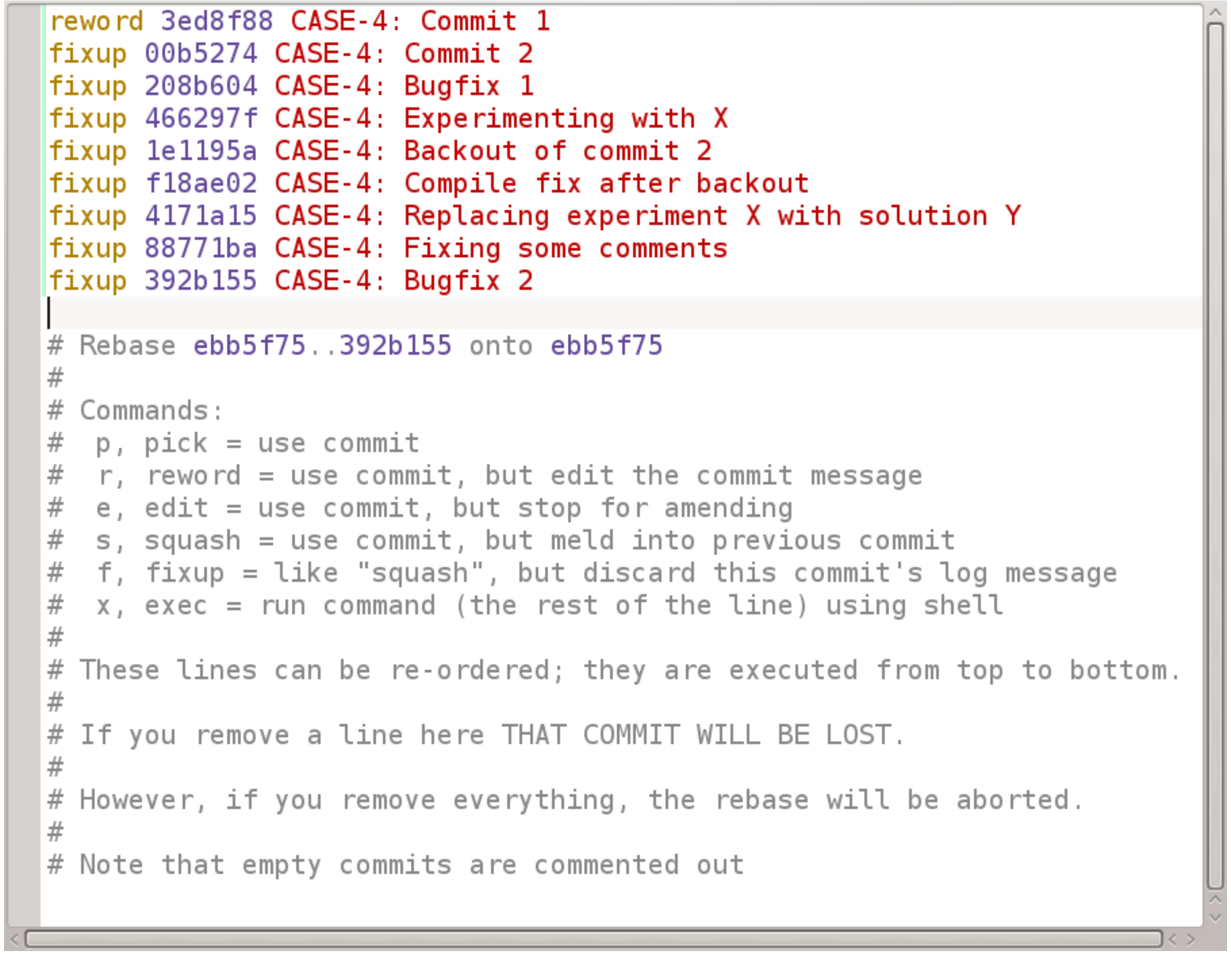
Step 3: rebase and squash



```
pick 3ed8f88 CASE-4: Commit 1
pick 00b5274 CASE-4: Commit 2
pick 208b604 CASE-4: Bugfix 1
pick 466297f CASE-4: Experimenting with X
pick 1e1195a CASE-4: Backout of commit 2
pick f18ae02 CASE-4: Compile fix after backout
pick 4171a15 CASE-4: Replacing experiment X with solution Y
pick 88771ba CASE-4: Fixing some comments
pick 392b155 CASE-4: Bugfix 2

# Rebase ebb5f75..392b155 onto ebb5f75
#
# Commands:
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

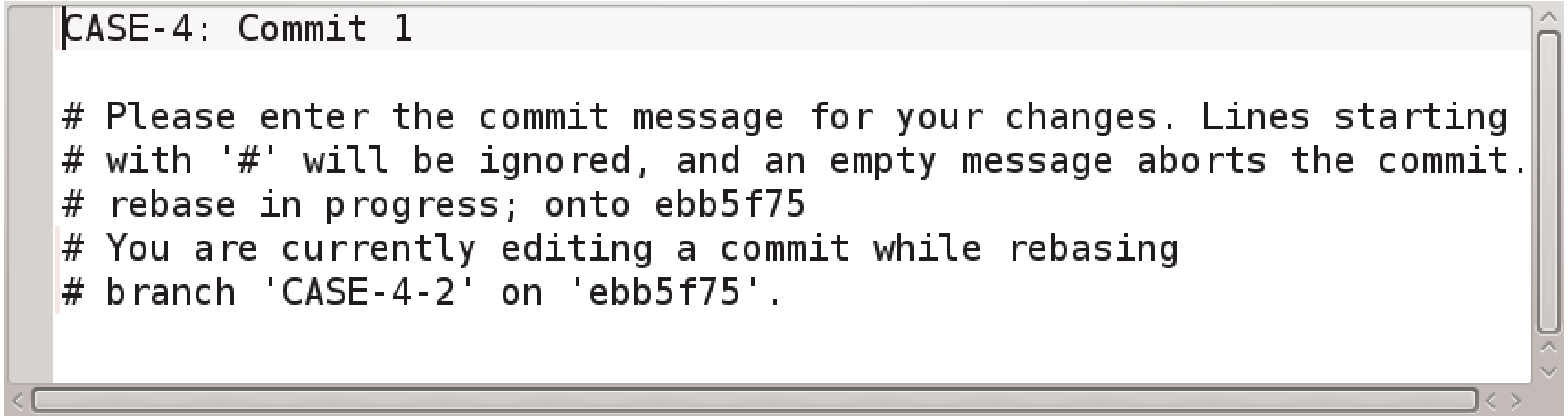
Step 3: rebase and squash



```
reword 3ed8f88 CASE-4: Commit 1
fixup 00b5274 CASE-4: Commit 2
fixup 208b604 CASE-4: Bugfix 1
fixup 466297f CASE-4: Experimenting with X
fixup 1e1195a CASE-4: Backout of commit 2
fixup f18ae02 CASE-4: Compile fix after backout
fixup 4171a15 CASE-4: Replacing experiment X with solution Y
fixup 88771ba CASE-4: Fixing some comments
fixup 392b155 CASE-4: Bugfix 2

# Rebase ebb5f75..392b155 onto ebb5f75
#
# Commands:
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

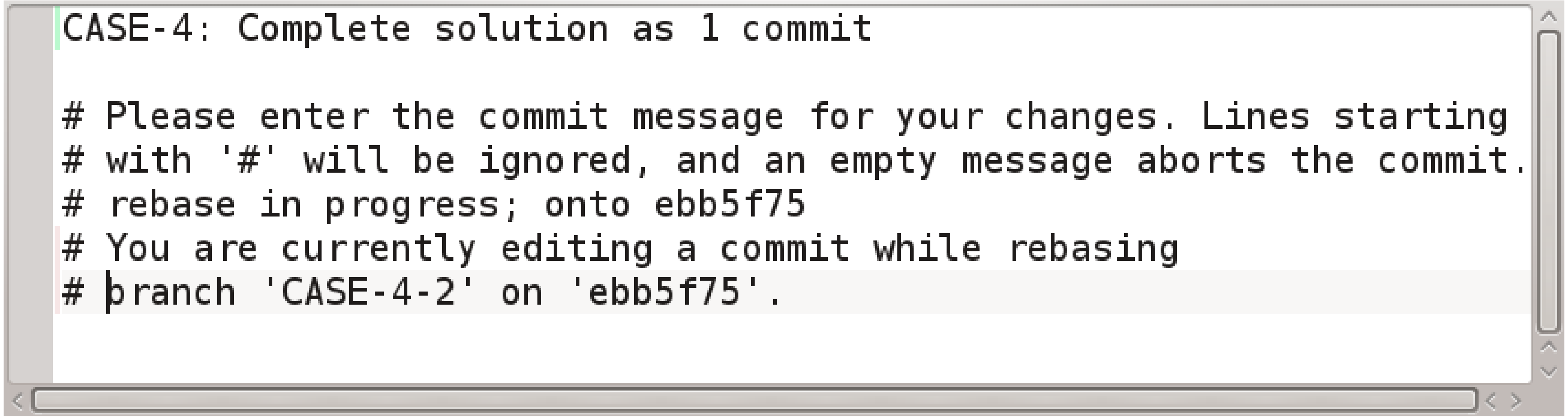
Step 3: rebase and squash



```
CASE-4: Commit 1

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# rebase in progress; onto ebb5f75
# You are currently editing a commit while rebasing
# branch 'CASE-4-2' on 'ebb5f75'.
```

Step 3: rebase and squash



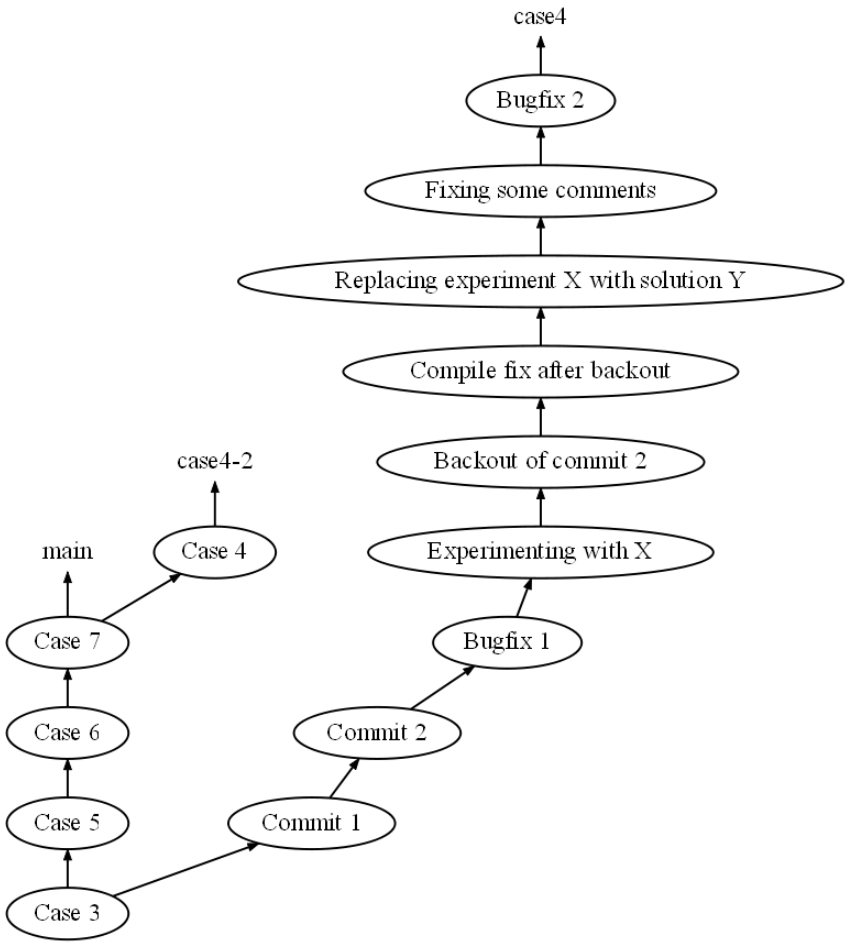
```
CASE-4: Complete solution as 1 commit

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# rebase in progress; onto ebb5f75
# You are currently editing a commit while rebasing
# branch 'CASE-4-2' on 'ebb5f75'.
```

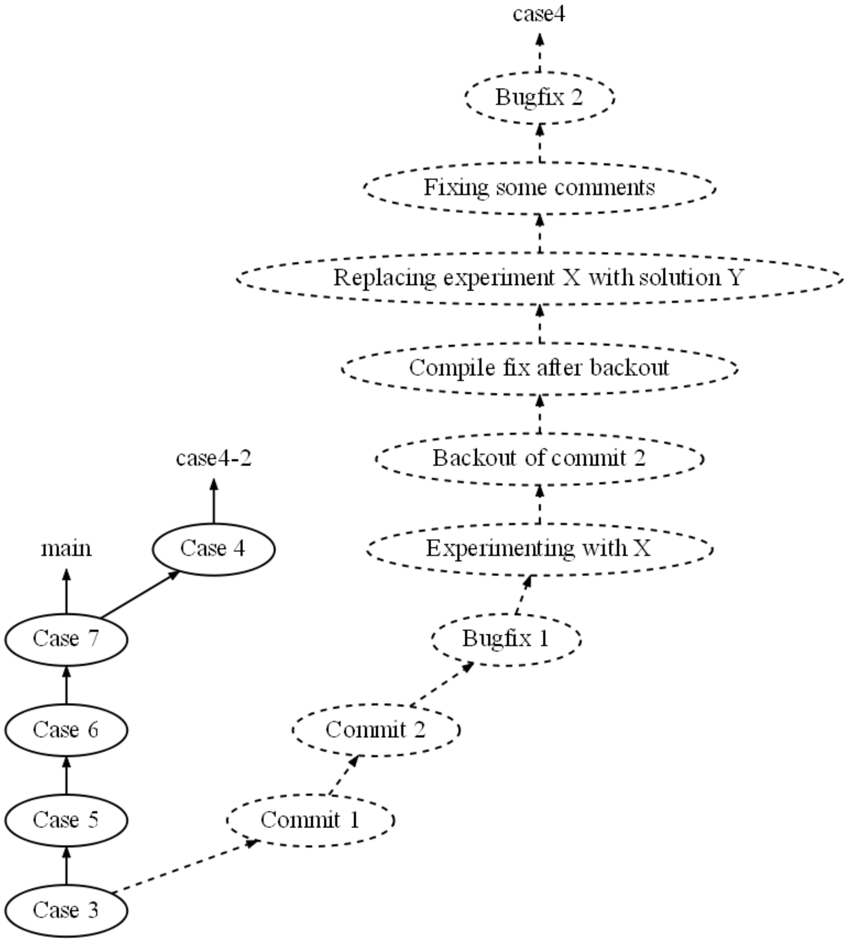
Step 3: rebase and squash

- Mercurial and rebasing
  - Rebase extension can not squash without rebasing
  - Collapse extension «collapses» with an error, i.e. does not work
  - MQ extension removes the original commits
  - Histedit extension annoys you with invocation of editor for every squashed commit
  - Transplant and Graft extensions do not squash
  - «hg diff –r rev1:rev2 | patch –p1» works

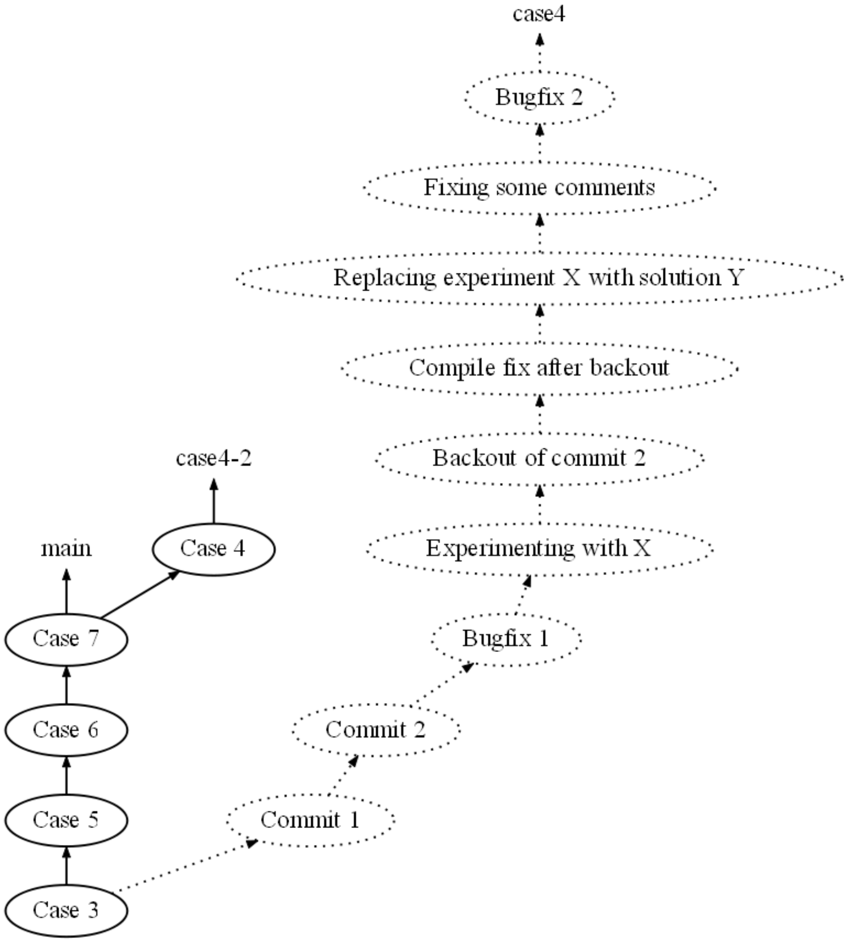
Step 4: forget about your old branch



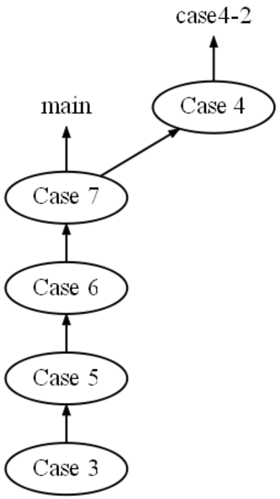
Step 4: forget about your old branch



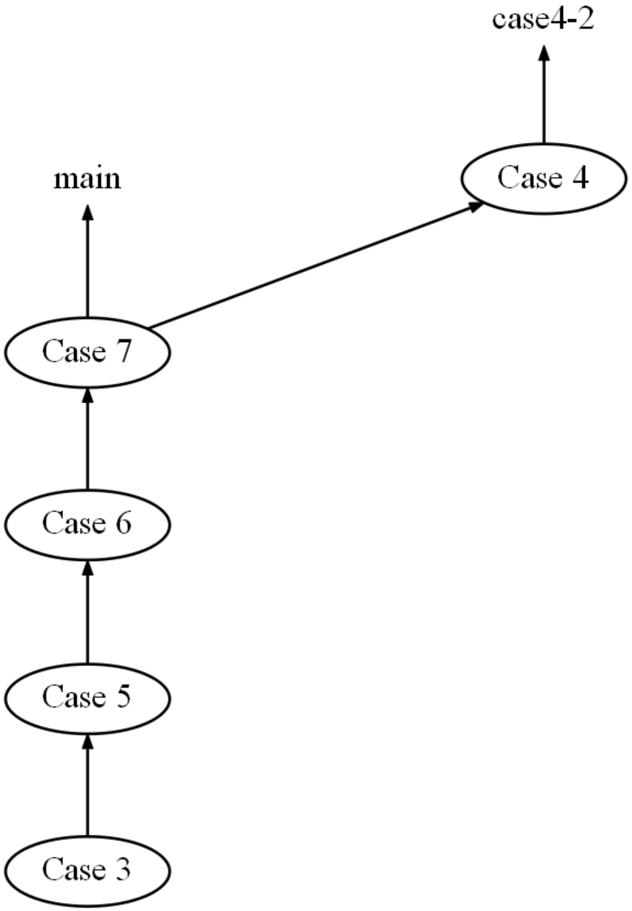
Step 4: forget about your old branch



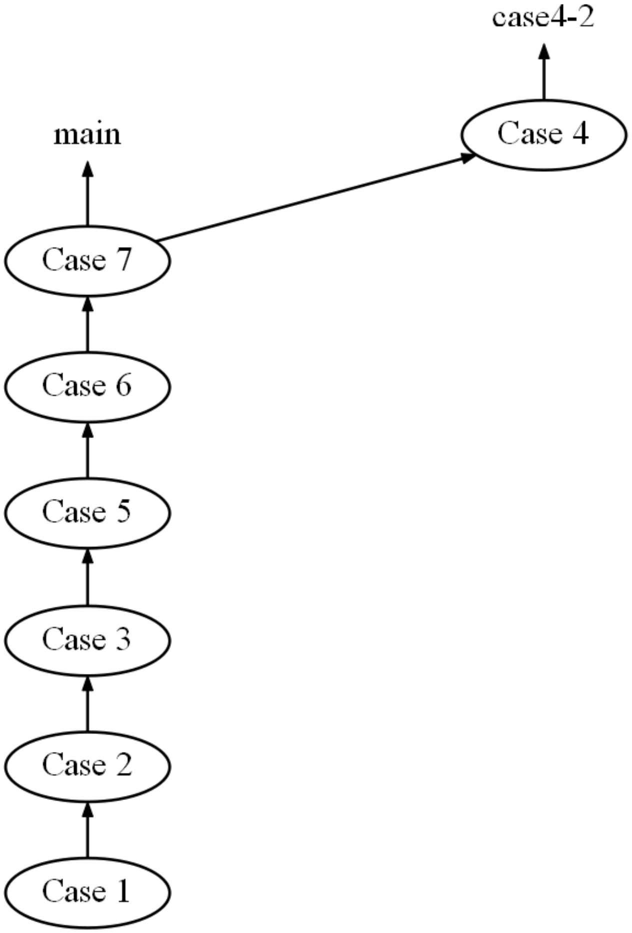
Step 4: forget about your old branch



Step 4: forget about your old branch



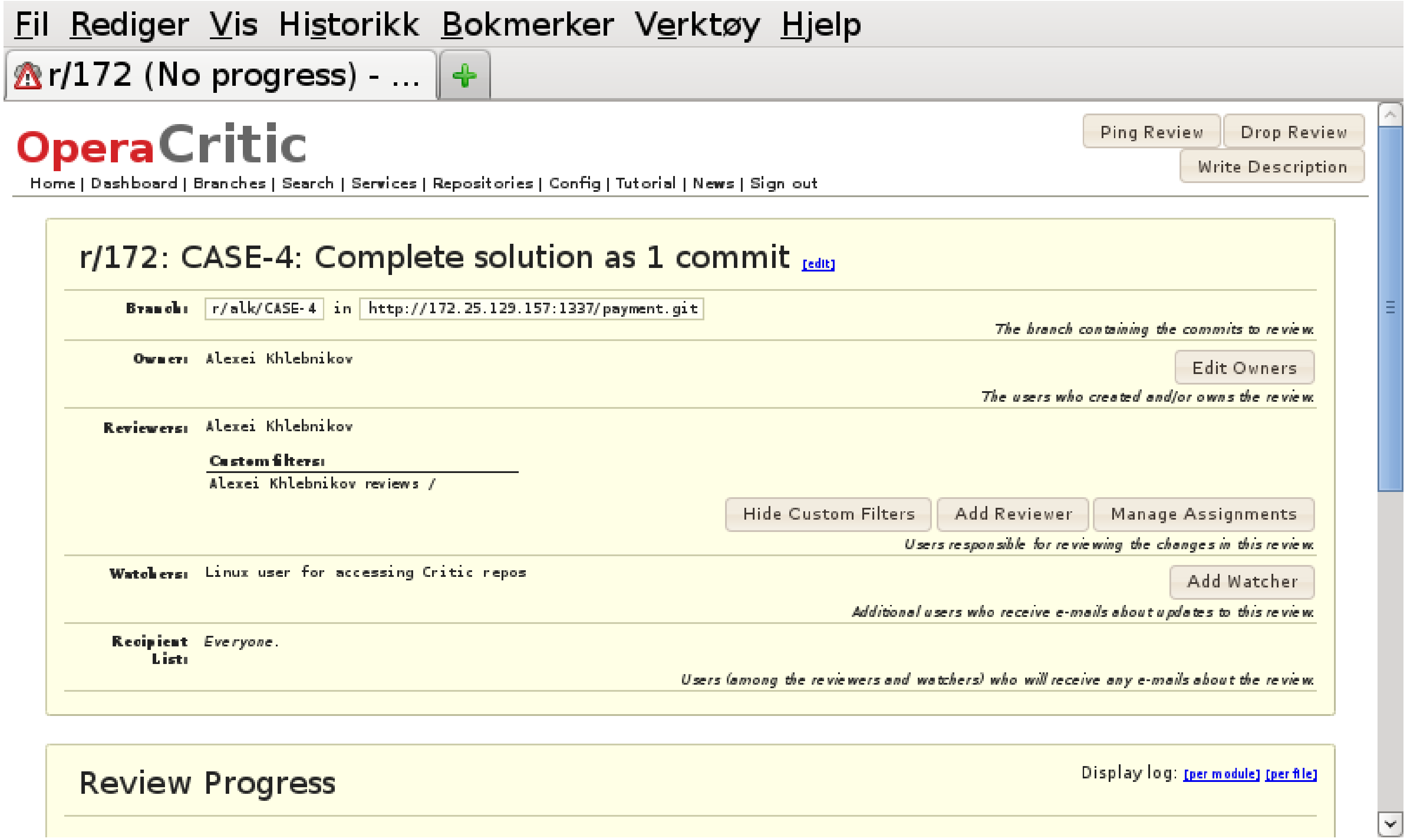
Step 4: forget about your old branch



Step 4.1: code review

- Submit code for review
  - git push critic case4-2:r/case4
- Eventually fix things during review
  - git commit -m 'Code review fixes'
  - git commit -m 'More fixes'
  - git push critic case4-2:r/case4
- Rebase and squash more if needed
  - git checkout -b case4-3
  - git rebase -i main

Step 4.1: code review

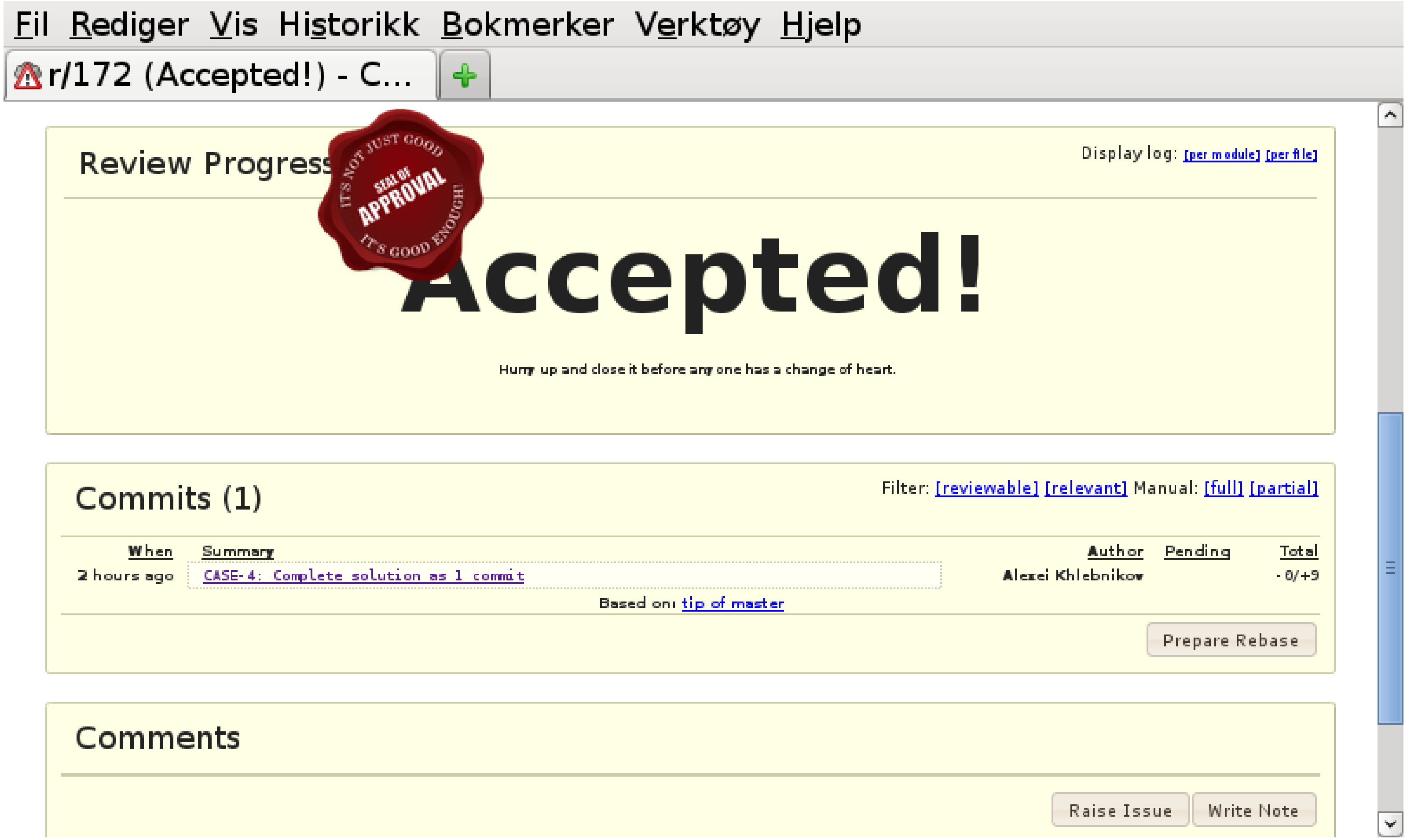

Fil Rediger Vis Historikk Bokmerker Verktøy Hjelp
r/172 (No progress) - …
OperaCritic
Home | Dashboard | Branches | Search | Services | Repositories | Config | Tutorial | News | Sign out
Ping Review
Drop Review
Write Description
[edit]
Edit Owners
Hide Custom Filters
Add Reviewer
Manage Assignments
Add Watcher
Display log: [per module] [per file]

r/172: CASE-4: Complete solution as 1 commit

Branch:	r/alk/CASE-4 in http://172.25.129.157:1337/payment.git *The branch containing the commits to review.*
Owner:	Alexei Khlebnikov *The users who created and/or owns the review.*
Reviewers:	Alexei Khlebnikov **Custom filters:** Alexei Khlebnikov reviews / *Users responsible for reviewing the changes in this review.*
Watchers:	Linux user for accessing Critic repos *Additional users who receive e-mails about updates to this review.*
Recipient List:	*Everyone.* *Users (among the reviewers and watchers) who will receive any e-mails about the review.*

Review Progress

Step 4.1: code review



Review Progress

Display log: [per module] [per file]

Accepted!

Hurry up and close it before anyone has a change of heart.

Commits (1)

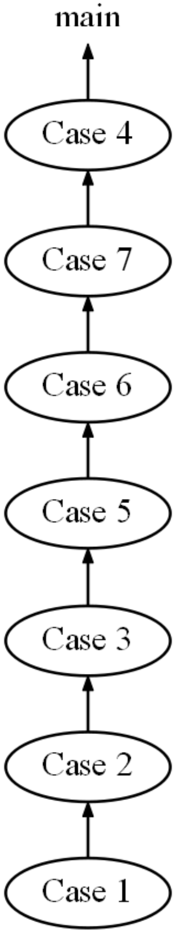
Filter: [reviewable] [relevant] Manual: [full] [partial]

When	Summary	Author	Pending	Total
2 hours ago	CASE-4: Complete solution as 1 commit	Alexei Khlebnikov		-0/+9

Based on: tip of master

Comments

Step 5: make your work part of main



Step 5: make your work part of main

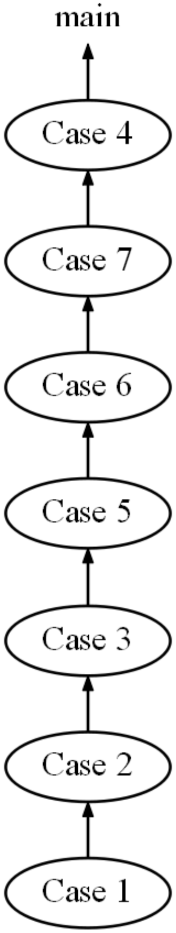
- Git
  - git push . case4-2:main

- Mercurial
  - (hg update case4-2)
  - hg book main

Step 5: make your work part of main

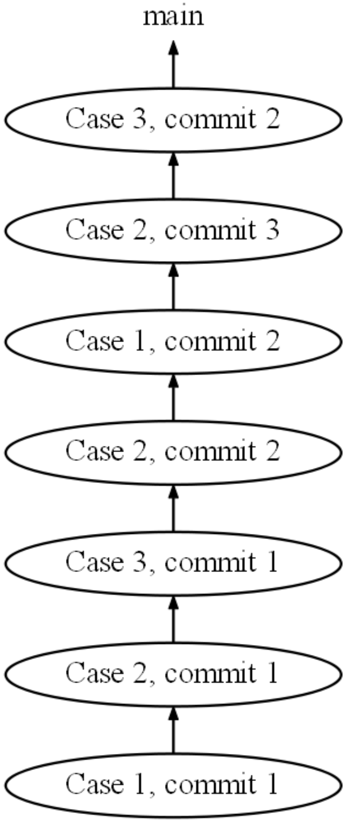
- Alternative solutions for git
  - Prerequisite: git checkout main
  - Alt 1: git reset --hard case4-2
  - Alt 2: git rebase case4-2
  - Alt 3: git cherry-pick --ff main..case4-2
  - Alt 4: git merge --ff-only case4-2

Step 5: make your work part of main



Before vs After

Before: chaos



main

Case 3, commit 2

Case 2, commit 3

Case 1, commit 2

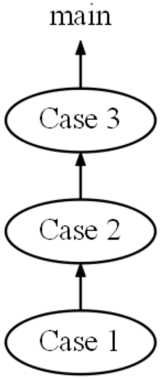
Case 2, commit 2

Case 3, commit 1

Case 2, commit 1

Case 1, commit 1

After: order



main

Case 3

Case 2

Case 1

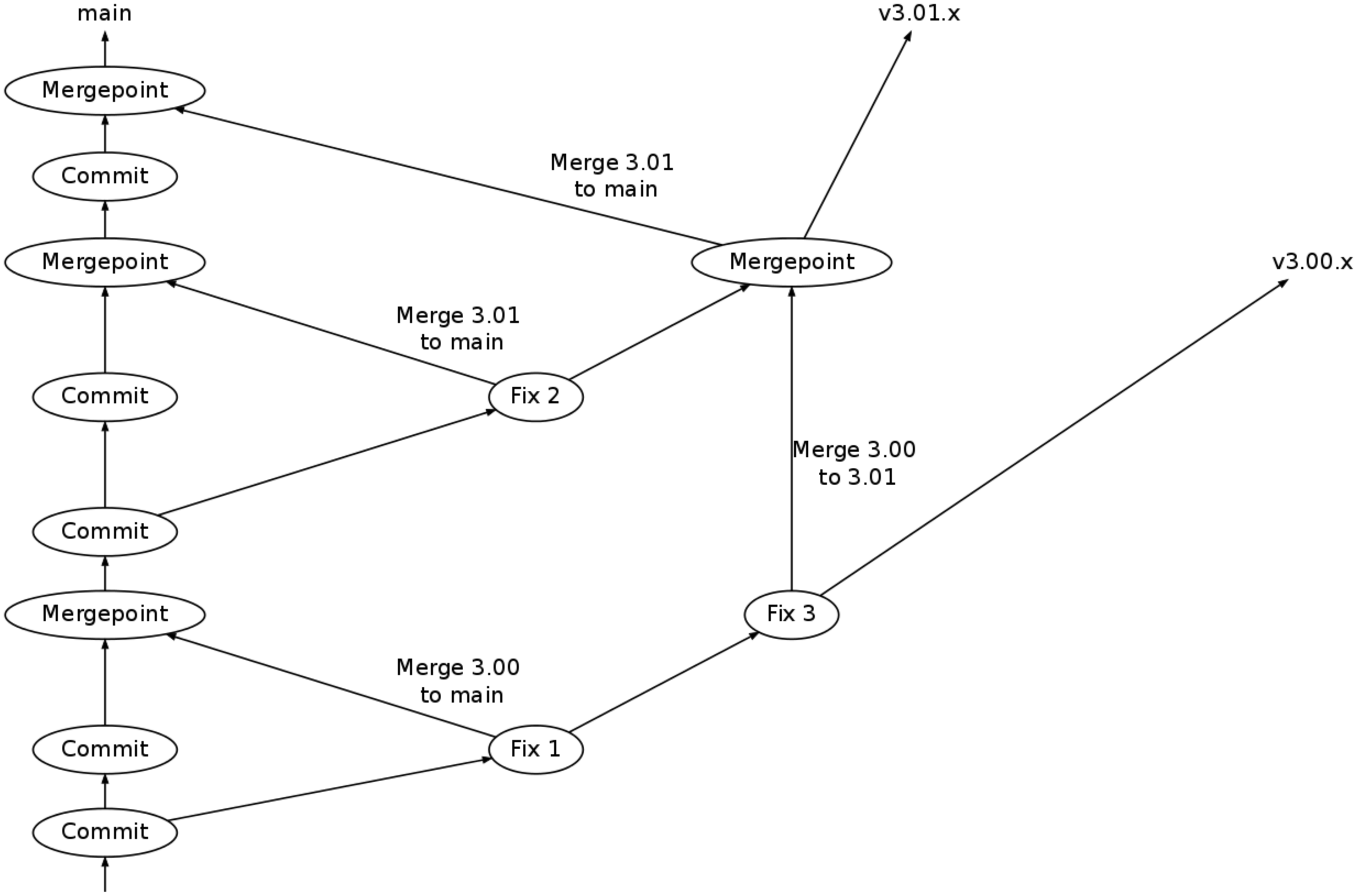
Release branches

- Make release
  - git checkout main
  - git checkout -b v3.00.x
  - git commit …
  - git tag v3.00.01

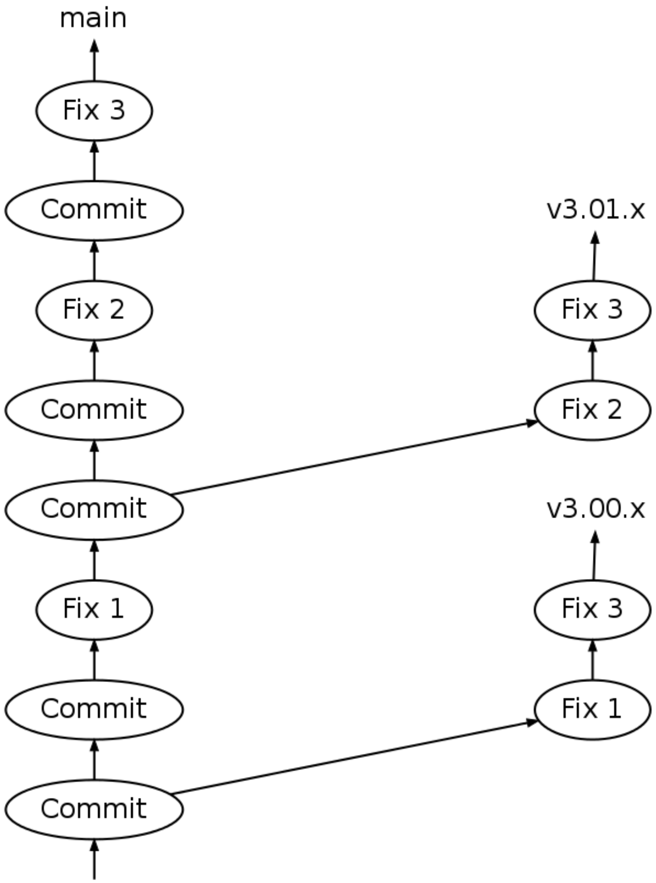
Fix on several branches

- Fix needed for branches:
  - v3.00.x
  - v3.01.x
  - main
- Branch from the oldest release branch
  - git checkout v3.00.x
  - git checkout -b case-123
  - git commit …

Delivery by merge



Delivery by cherry-pick



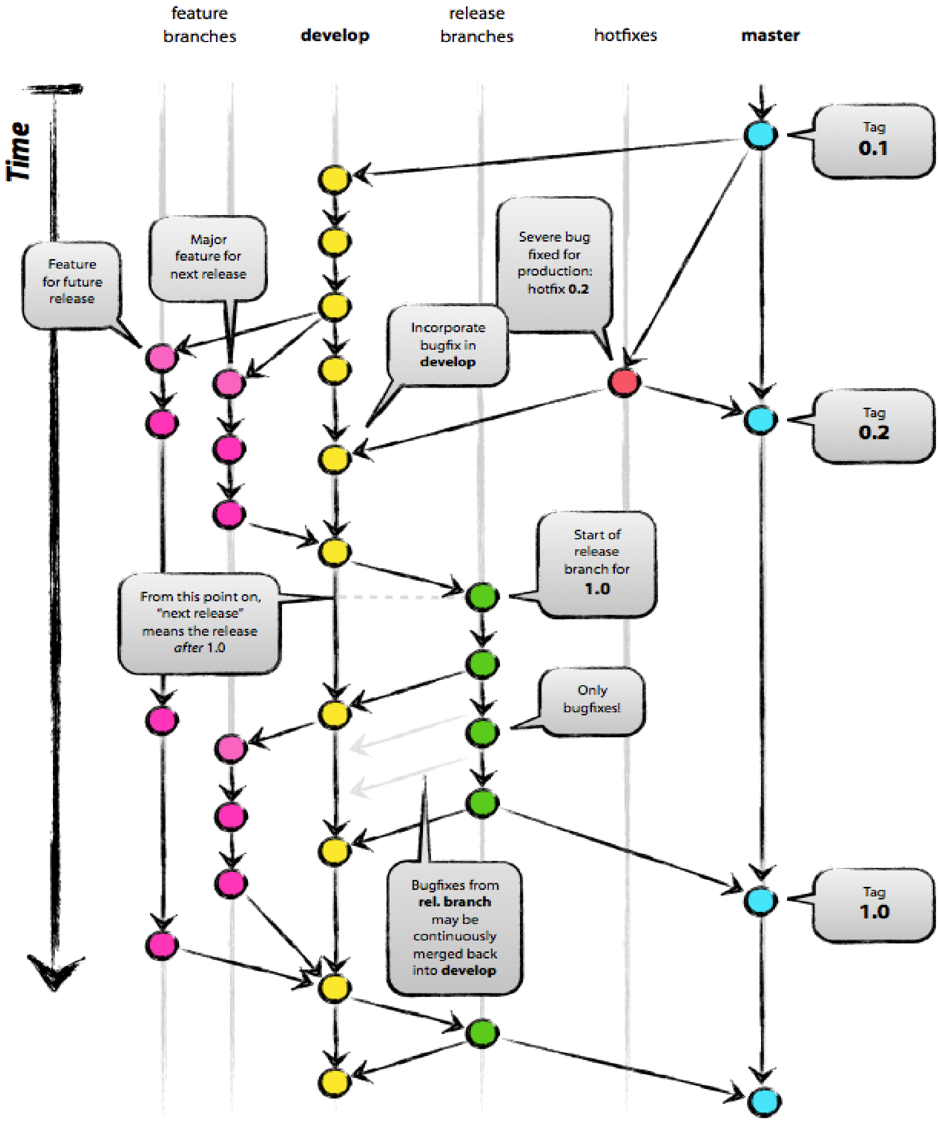
Fix on several branches

- Delivery to v3.00.x
  - git checkout -b case-123-2
  - git rebase -i v3.00.x
  - git push . case-123-2:v3.00.x
- Delivery to v.3.01.x **by merge**
  - git checkout v3.01.x
  - git merge v3.00.x
- Delivery to main **by merge**
  - git checkout main
  - git merge v3.01.x

Fix on several branches

- Delivery to v.3.01.x **by cherry-pick**
  - git checkout v3.01.x
  - git cherry-pick a1b2c3..case-123-2
- Delivery to main **by cherry-pick**
  - git checkout main
  - git cherry-pick a1b2c3..case-123-2

Git-flow


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Severe bug fixed for production: hotfix 0.2
Major feature for next release
Feature for future release
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Delivering fix to several branches: merge vs cherry-pick

- Merge:
  - Delivers ”all or nothing”
    - For example, can not deliver:
      - Fix 1 only to v3.00.x and main
      - Fix 2 only to v3.01.x and main
- Cherry-pick
  - Allows for granular delivery
    - Whatever needed in whatever order
    - Can also deliver all, if needed

Delivering fix to several branches: merge vs cherry-pick

- Merge:
  - Delivers ”all or nothing”
    - More delivery – more merge conflicts
  - Forces to resolve all merge conflicts at once
- Cherry-pick
  - Allows for granular delivery
    - Less merge conflicts
  - Allows to resolve merge conflicts in smaller chunks

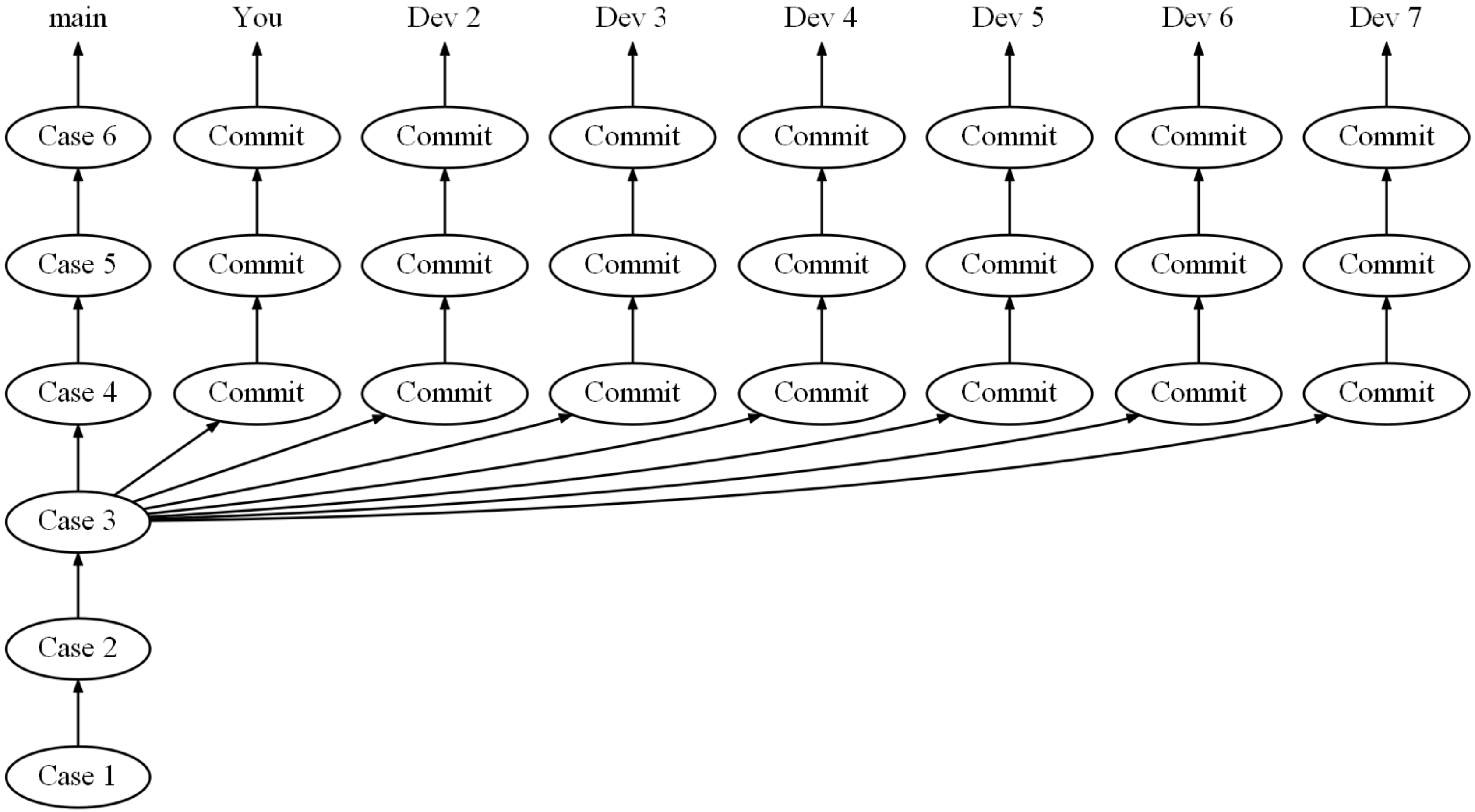
Delivering fix to several branches: merge vs cherry-pick

- Merge:
  - Delivers only from older branch to newer
    - Otherwise delivers unwanted changes and makes a mess
  - Original fix can only be developed against the oldest branch
- Cherry-pick
  - Delivers from any branch to any
  - Original fix can be developed on any branch
  - Allows customer-specific branches
  - More flexible

Delivering fix to several branches: merge vs cherry-pick

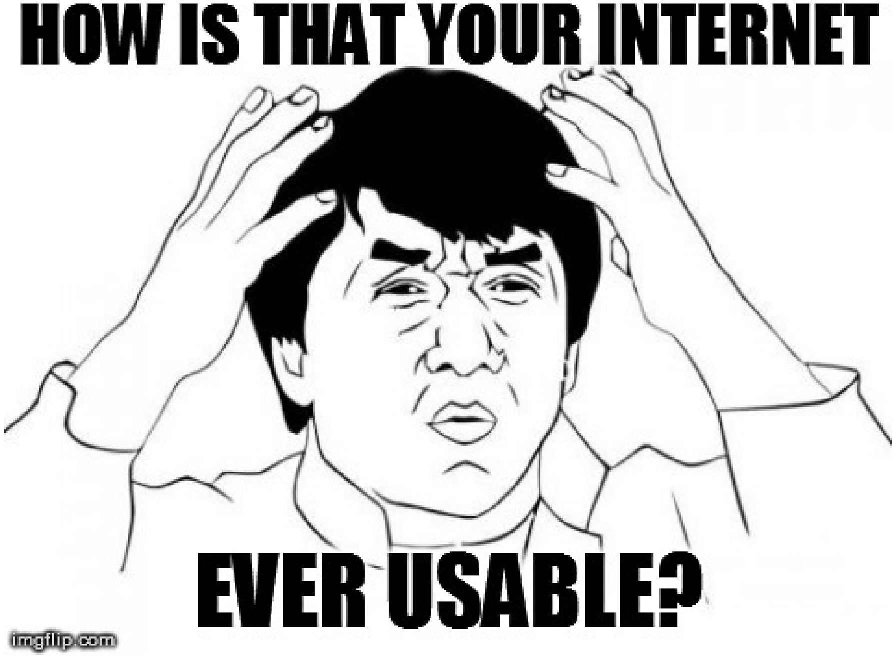
- Merge:
  - Makes multi-parent commits
    - Complicates history
    - Hinders eventual future cherry-pick and rebase
- Cherry-pick
  - Makes single-parent commits
    - History of any release branch becomes linear, i.e. easier to read
    - Does not hinder eventual future cherry-pick, rebase or even merge
    - Tree is simpler than graph

Pseudoproblem: too many branches/bookmarks



Pseudoproblem: too many sites: attacking the Internet since 1990s

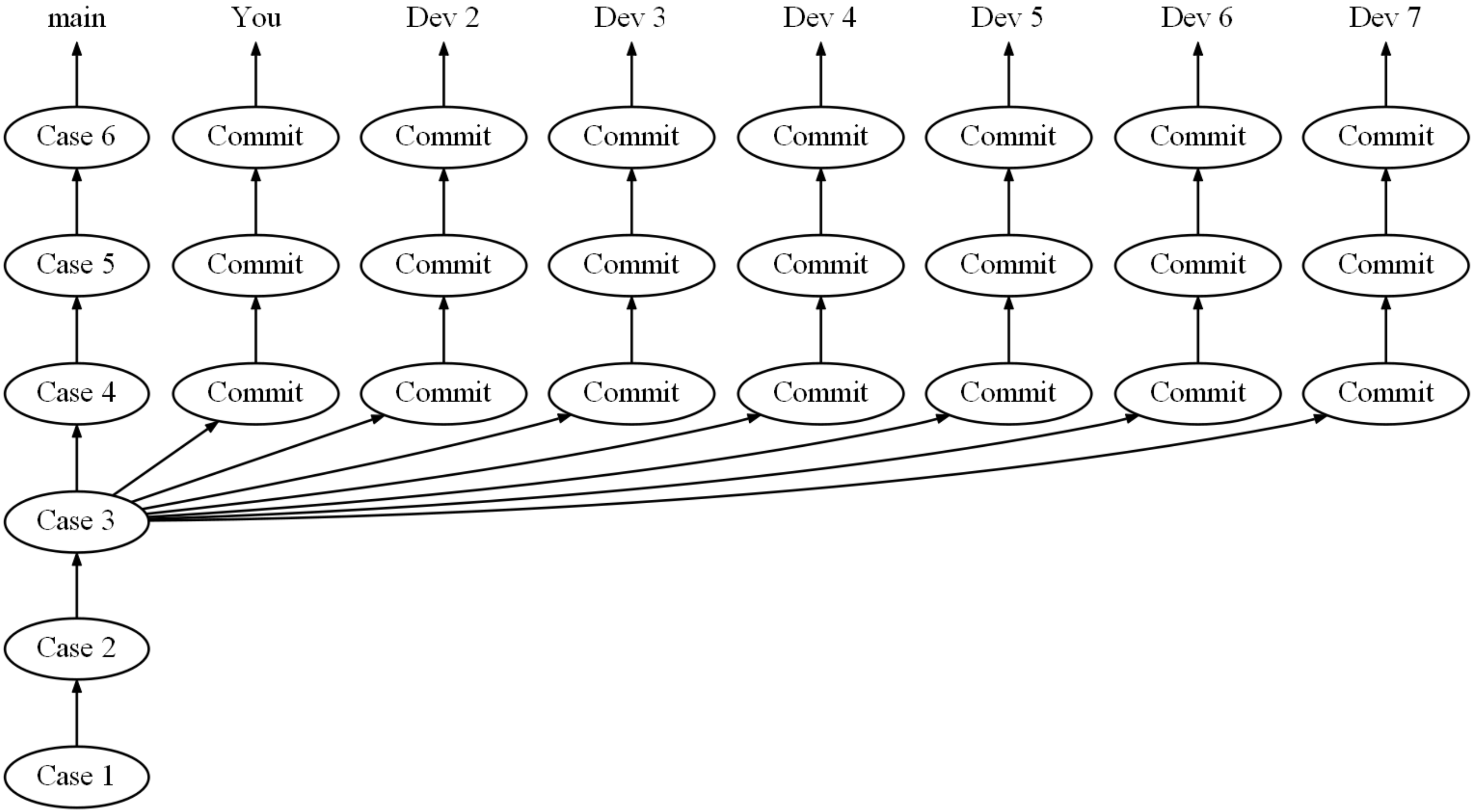
Full list of the Internet sites is too long!



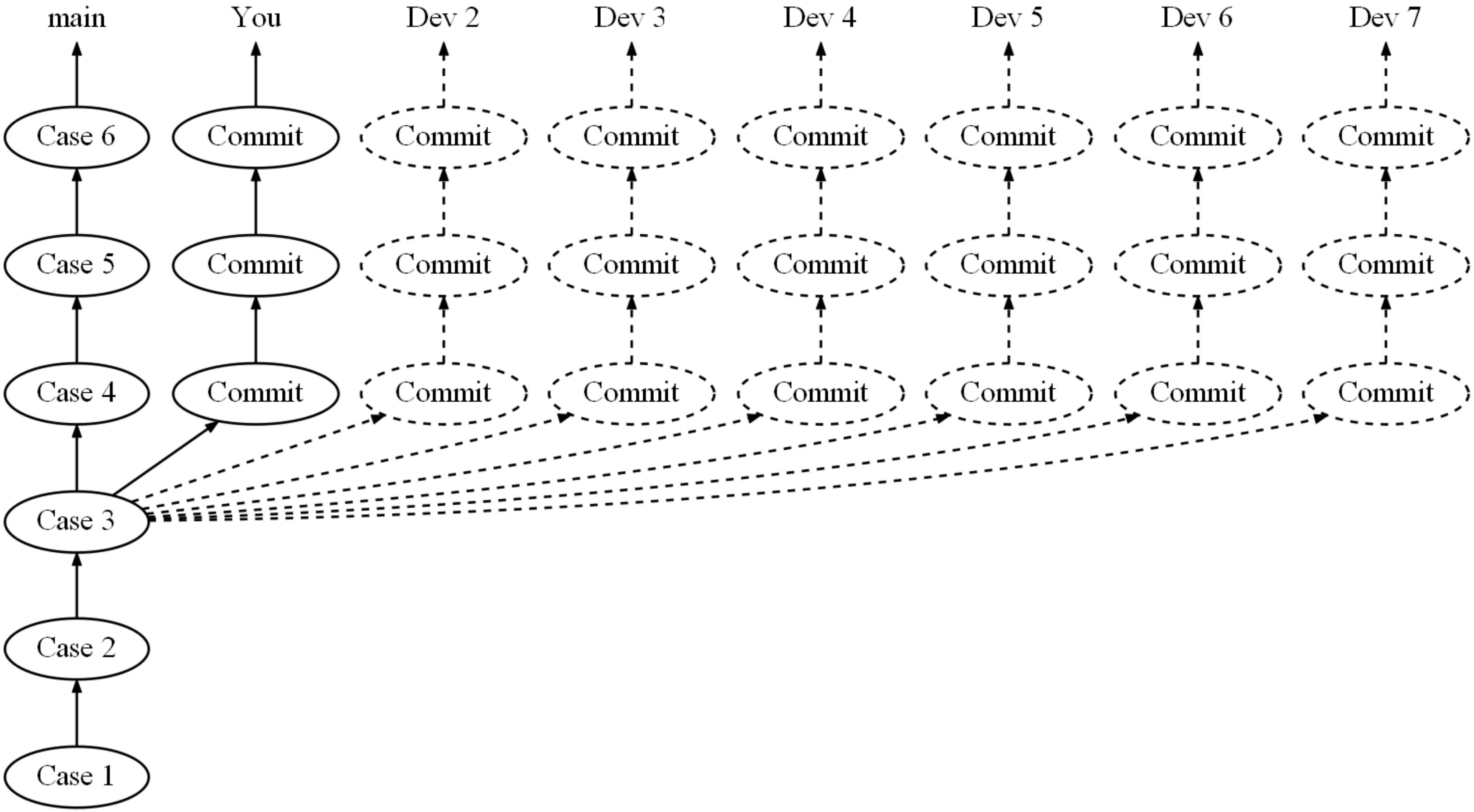
Pseudoproblem: too many sites: attacking the Internet since 1990s

Thank you, Captain Obvious!

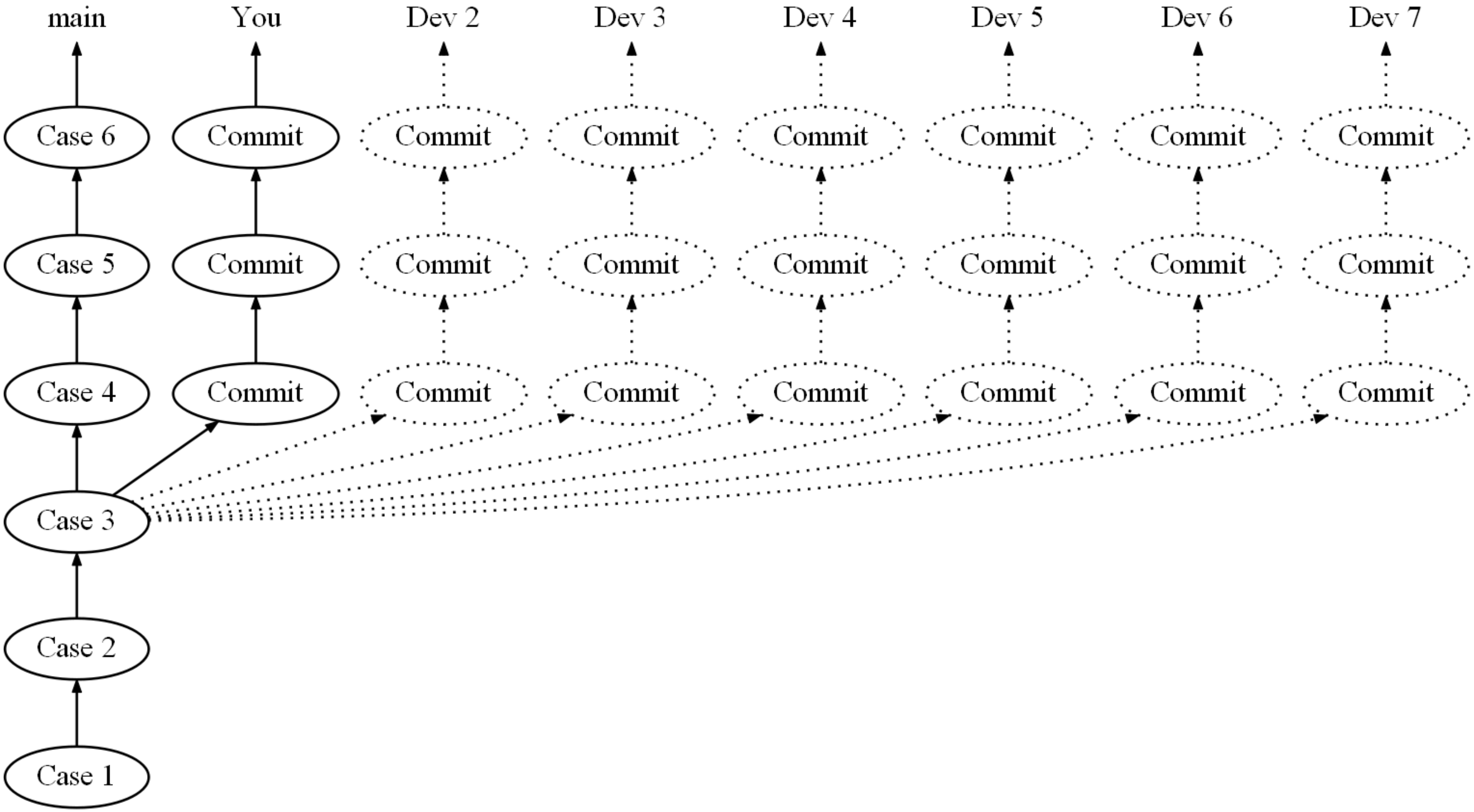
Too many branches: solution



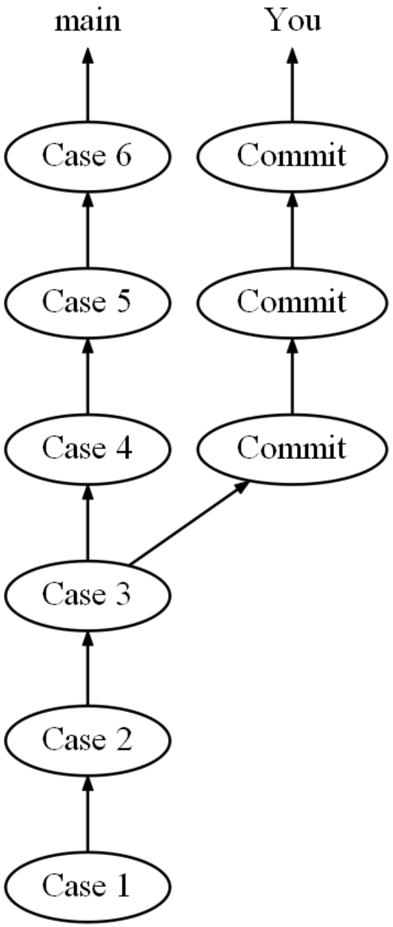
Too many branches: solution



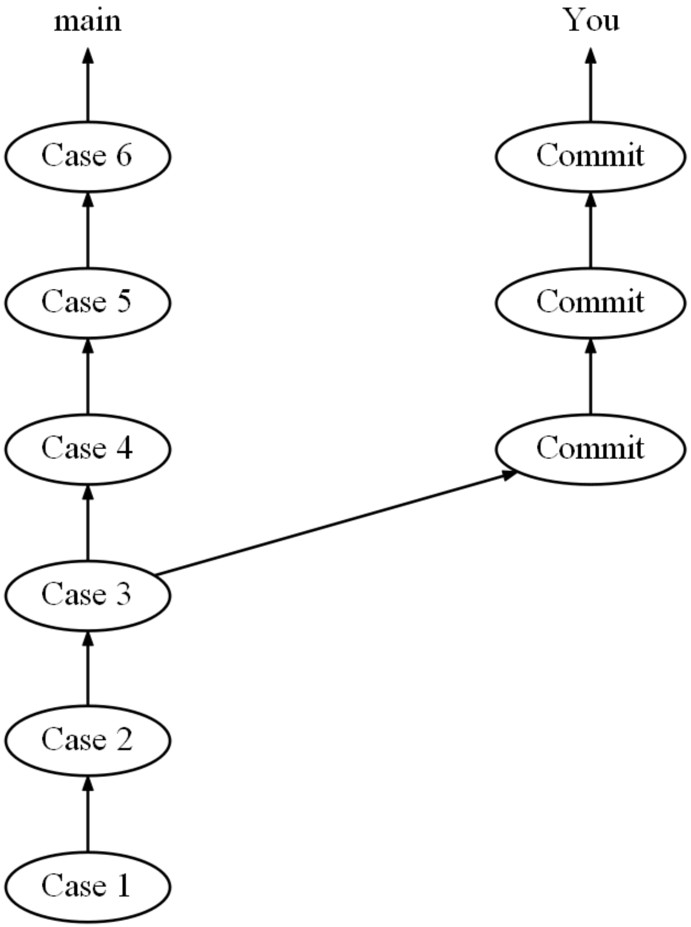
Too many branches: solution



Too many branches: solution

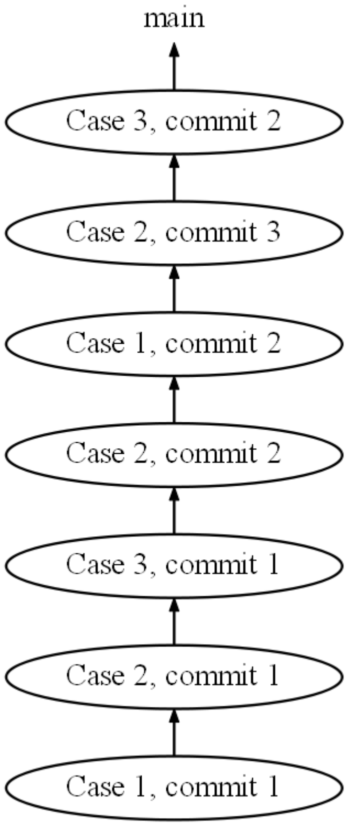


Too many branches: solution



Before vs After

Before: chaos



main

Case 3, commit 2

Case 2, commit 3

Case 1, commit 2

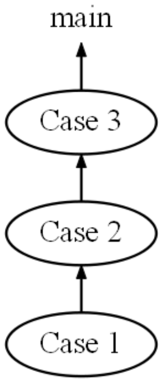
Case 2, commit 2

Case 3, commit 1

Case 2, commit 1

Case 1, commit 1

After: order



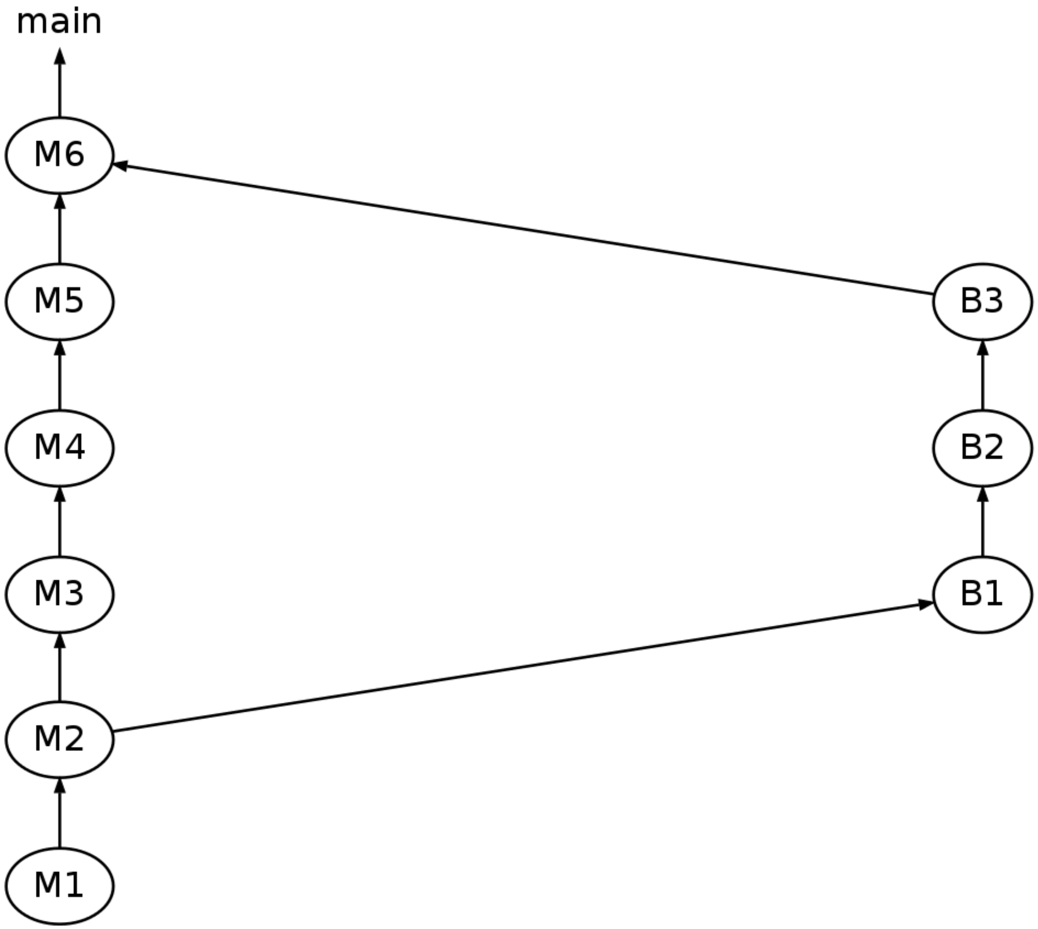
main

Case 3

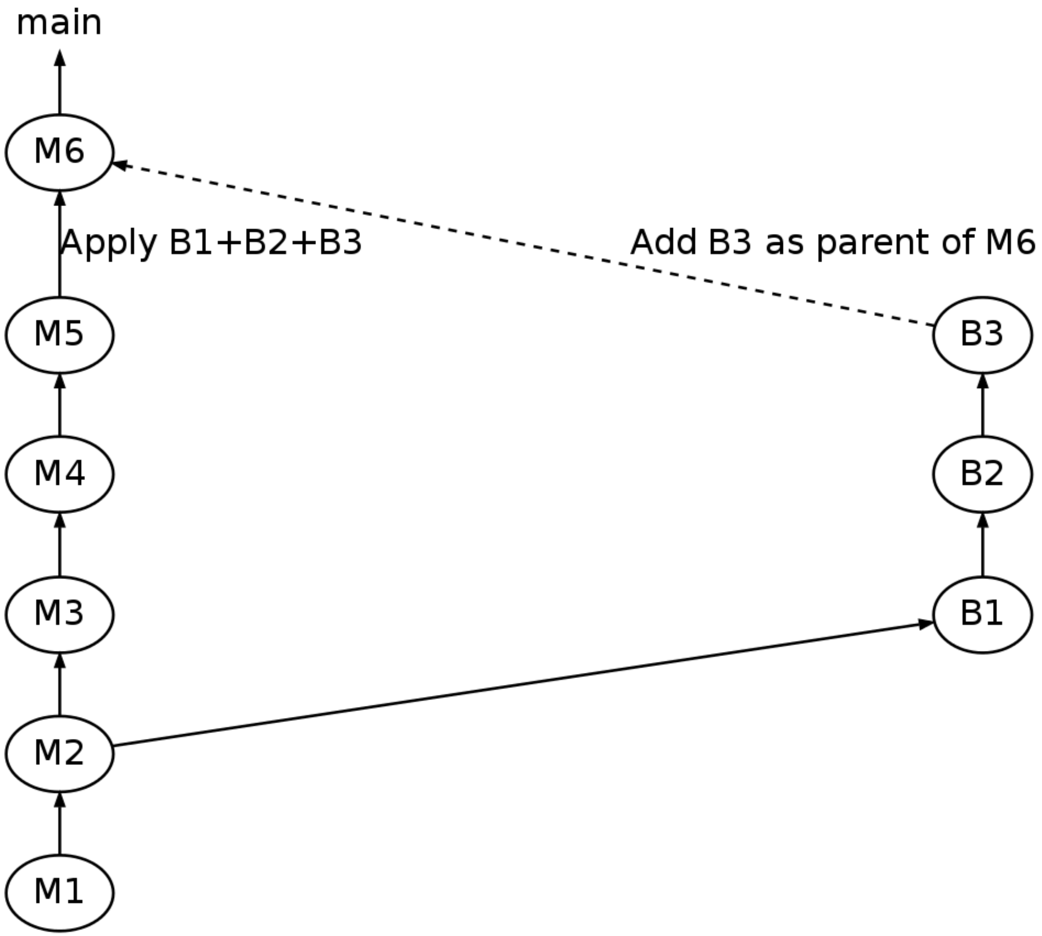
Case 2

Case 1

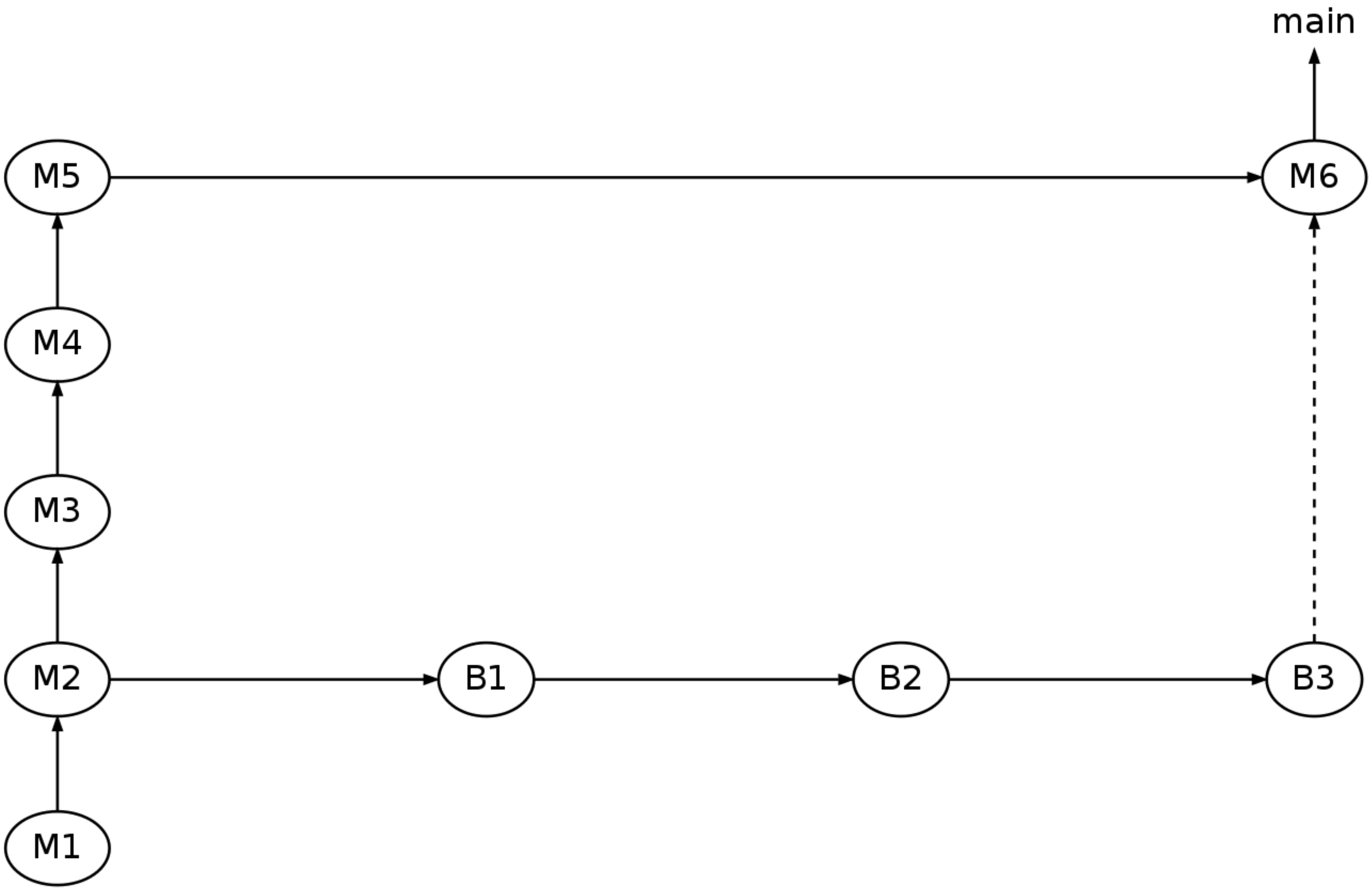
Merge conflicts: merge



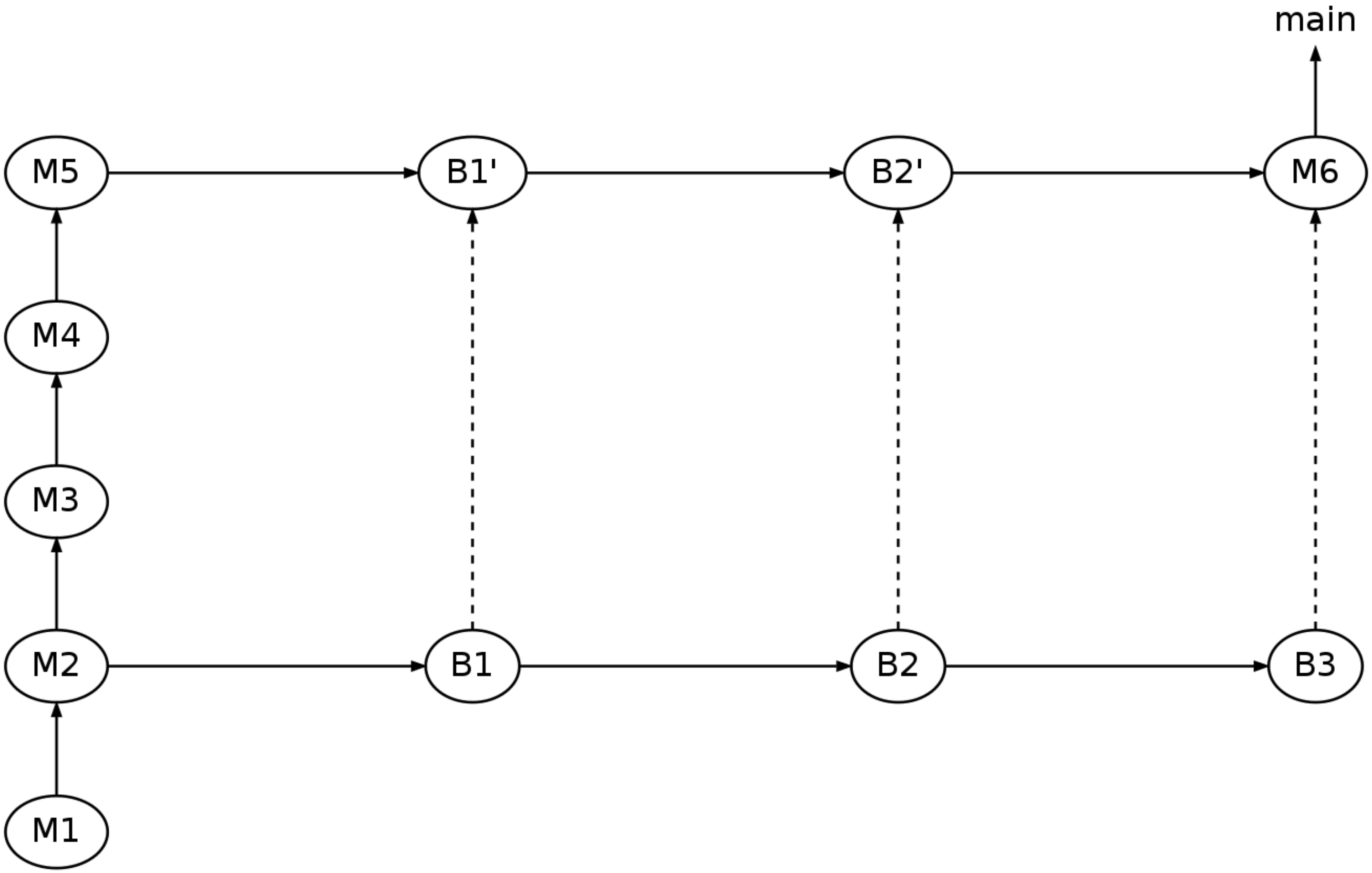
Merge conflicts: merge



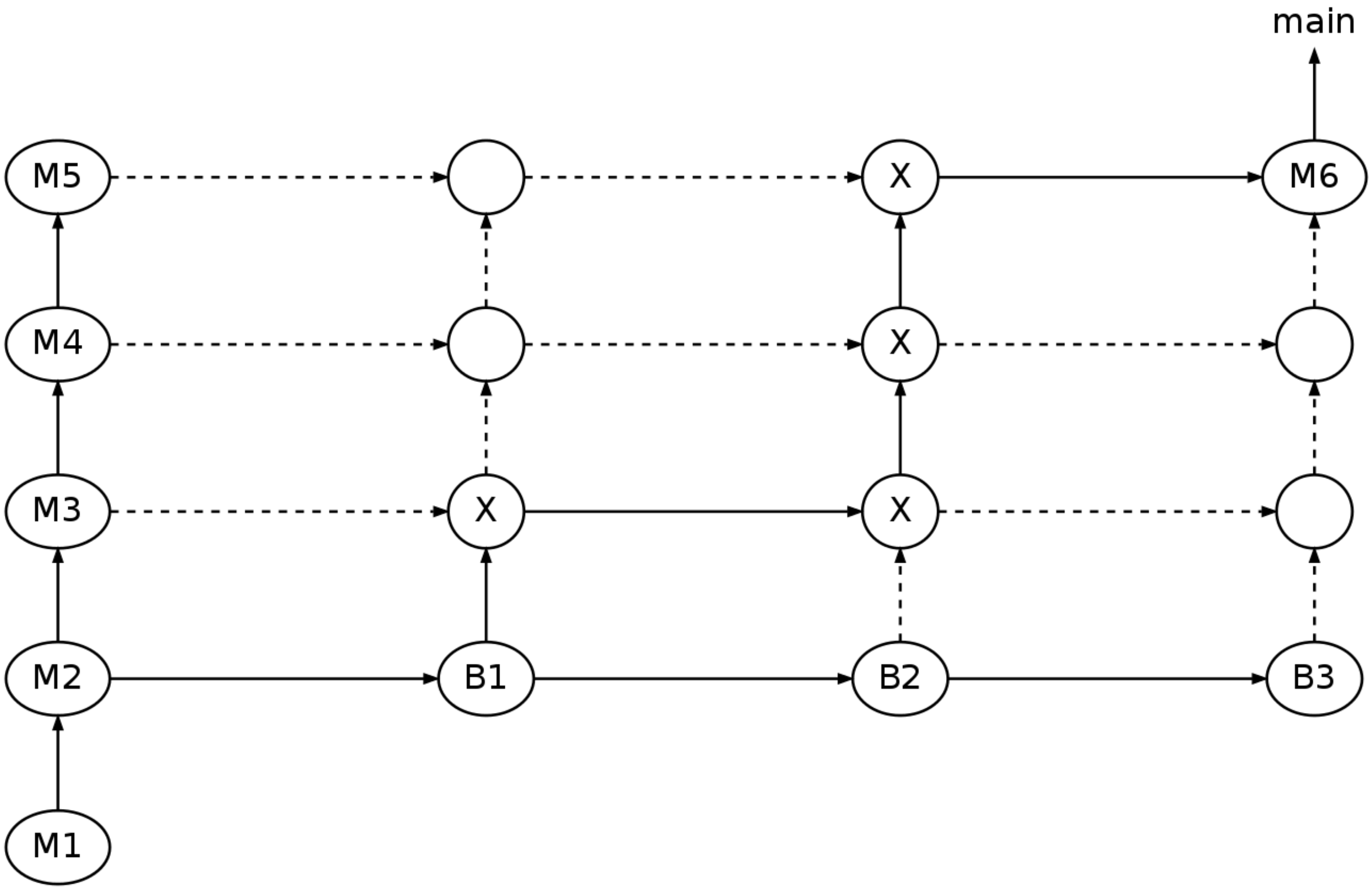
Merge conflicts: merge



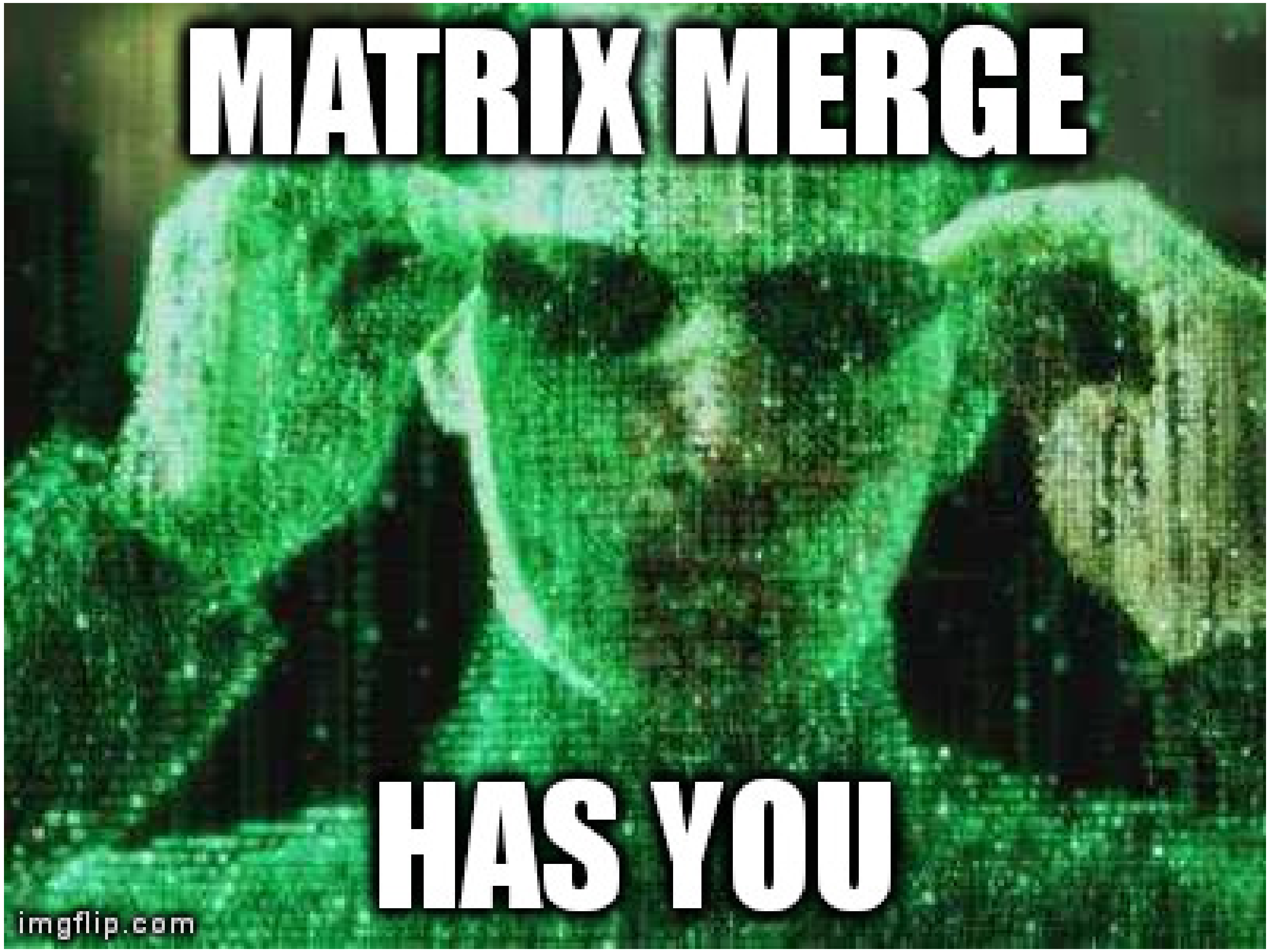
Merge conflicts: rebase



Matrix merge



Matrix merge



Merge conflicts: strategies

- Merge: merging all at once
- Rebase/cherry-pick: merging step by step
- Matrix merge:
  - Idea so far, not implemented in any tool
  - Merging step by step
  - Along (hopefully) optimal path that minimizes conflicts
  - Can merge more automatically
    - May be dangerous?
  - Does not support linear history, unlike Rebase

Спасибо за внимание

Вопросы?

(Вбросы? :)