Managing Software Evolution Through Semantic History Slicing

Yi Li
University of Toronto

PhD Advisor: Marsha Chechik

ASE'17 Doctoral Symposium
Oct 30, 2017

Acknowledgements

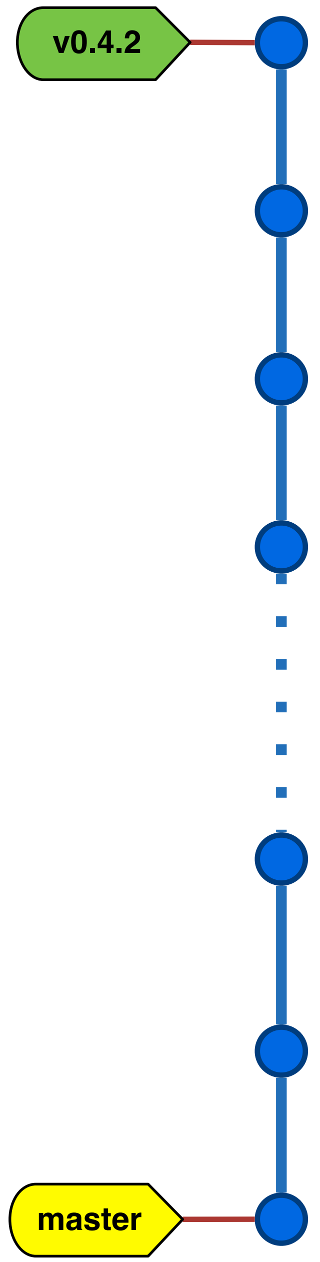
Marsha Chechik
U Toronto

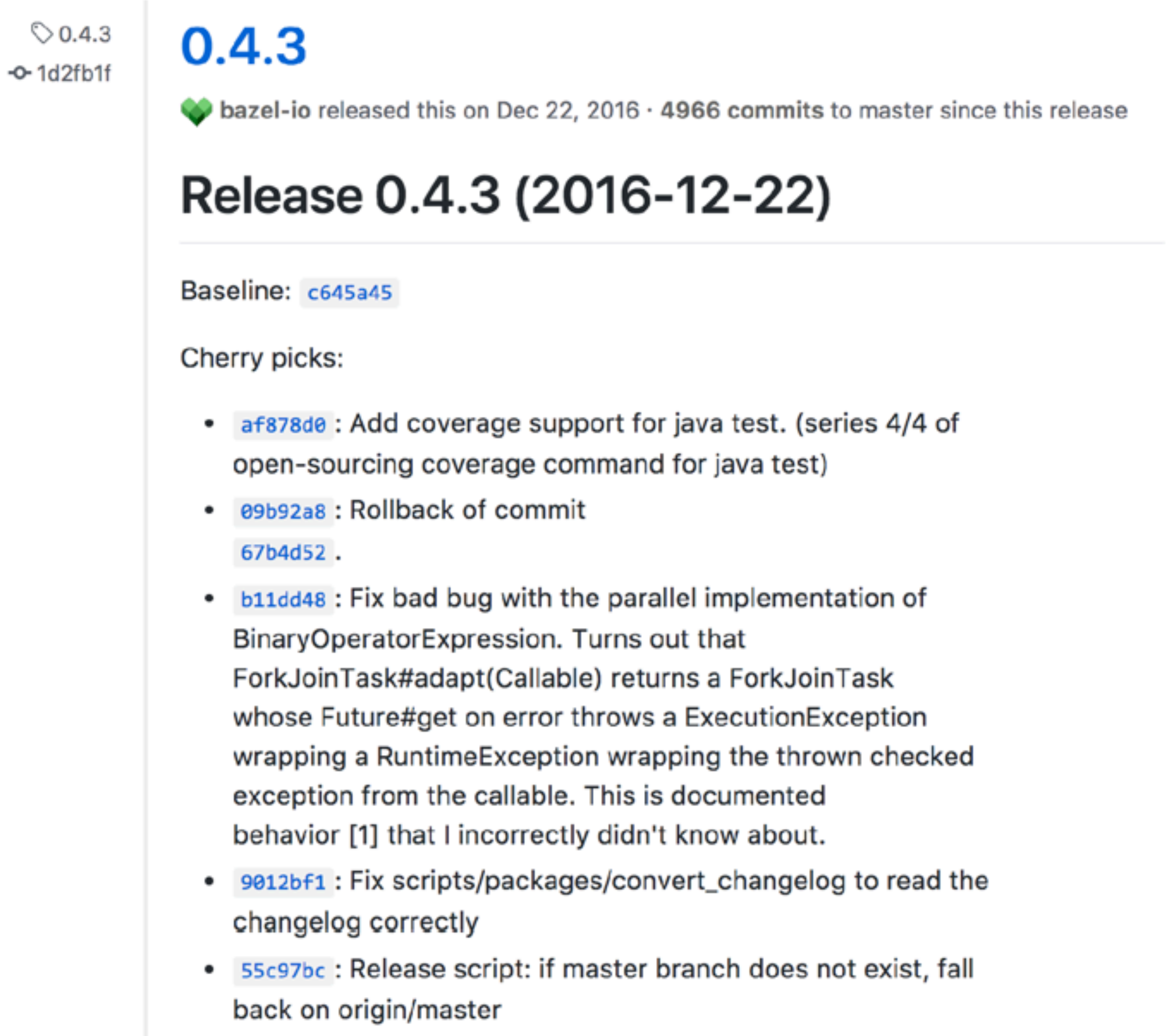
Julia Rubin
UBC

Chenguang Zhu
U Texas, Austin

Bazel

{Fast, Correct} - Choose two





0.4.3
1d2fb1f

0.4.3

bazel-io released this on Dec 22, 2016 · **4966 commits** to master since this release

Release 0.4.3 (2016-12-22)

Baseline: `c645a45`

Cherry picks:

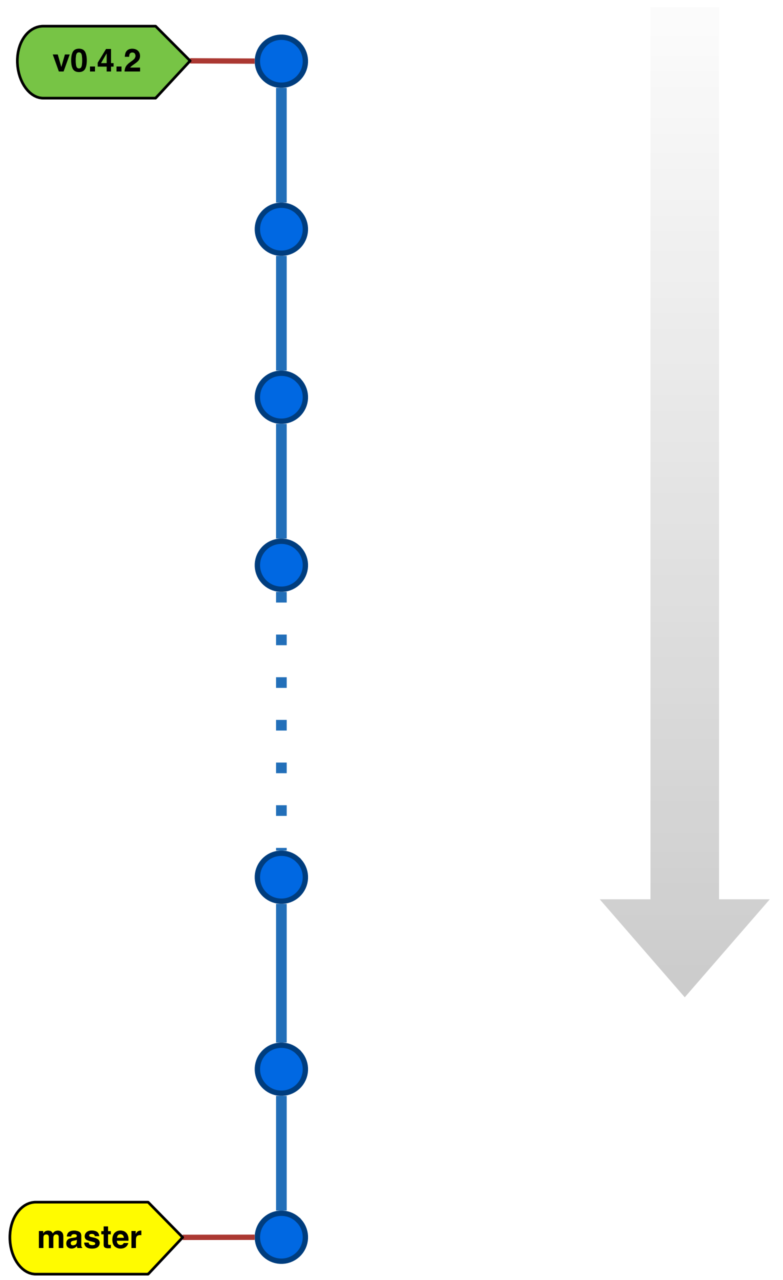
- `af878d0`: Add coverage support for java test. (series 4/4 of open-sourcing coverage command for java test)
- `09b92a8`: Rollback of commit `67b4d52`.
- `b11dd48`: Fix bad bug with the parallel implementation of BinaryOperatorExpression. Turns out that ForkJoinTask#adapt(Callable) returns a ForkJoinTask whose Future#get on error throws a ExecutionException wrapping a RuntimeException wrapping the thrown checked exception from the callable. This is documented behavior [1] that I incorrectly didn't know about.
- `9012bf1`: Fix scripts/packages/convert_changelog to read the changelog correctly
- `55c97bc`: Release script: if master branch does not exist, fall back on origin/master

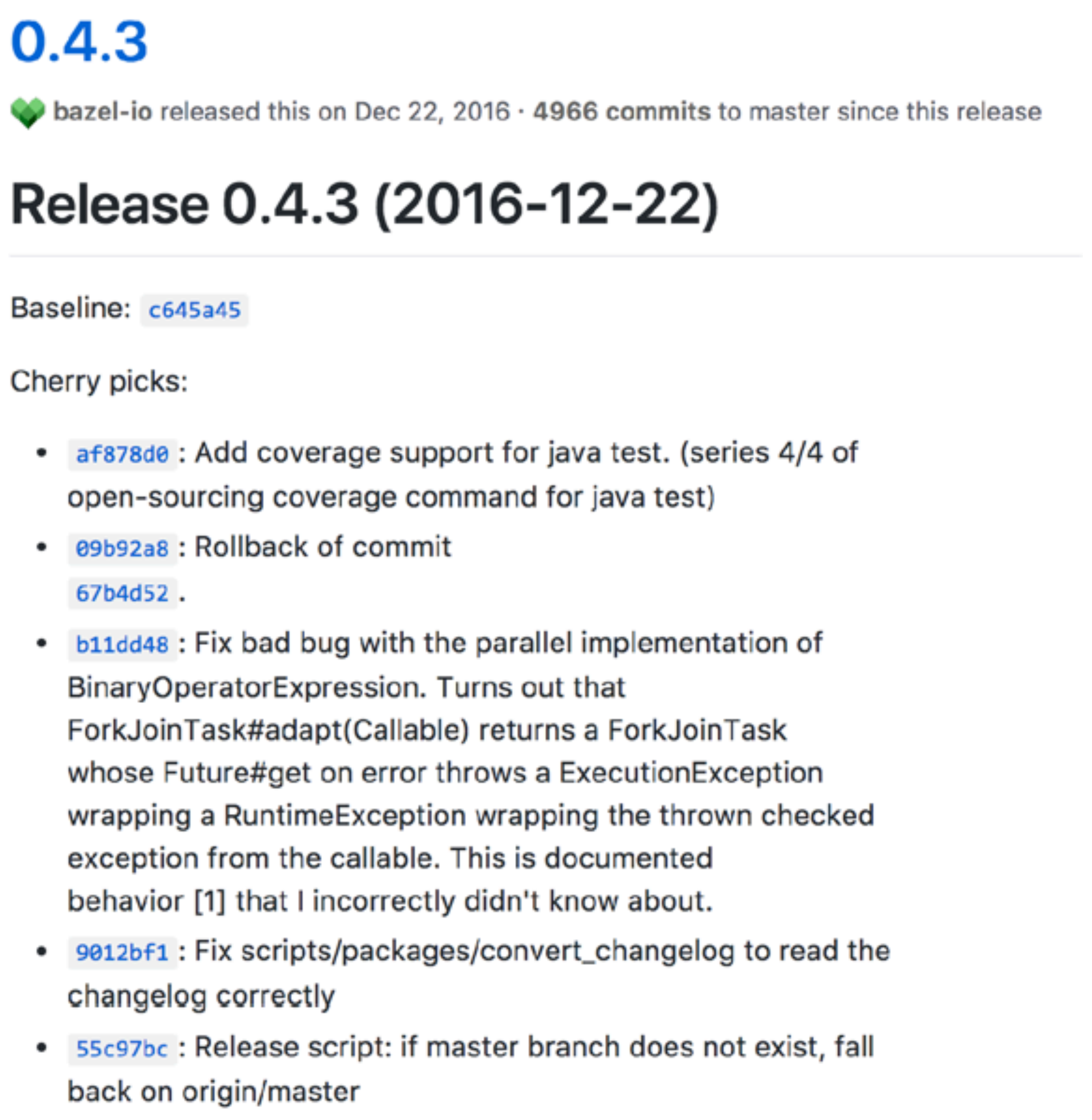
...

Bazel

{Fast, Correct} - Choose two

0.4.3
1d2fb1f





0.4.3

bazel-io released this on Dec 22, 2016 · **4966 commits** to master since this release

Release 0.4.3 (2016-12-22)

Baseline: `c645a45`

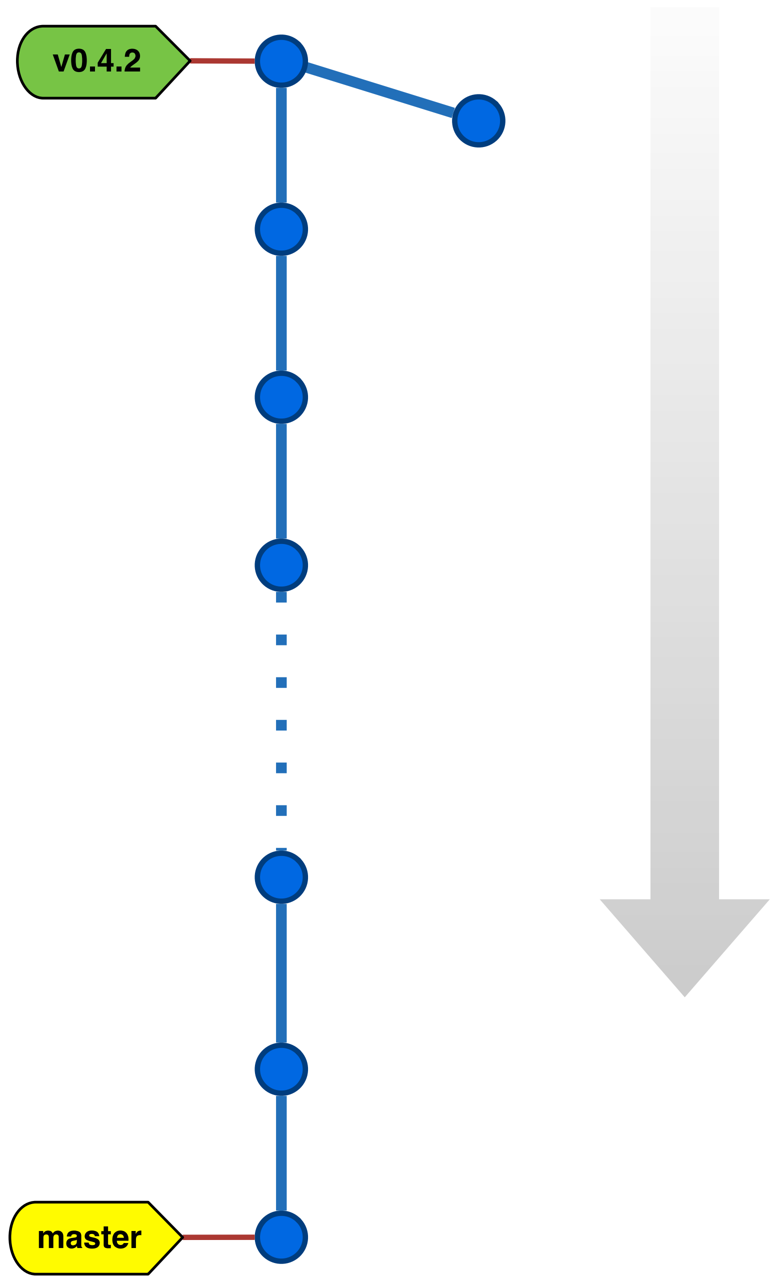
Cherry picks:

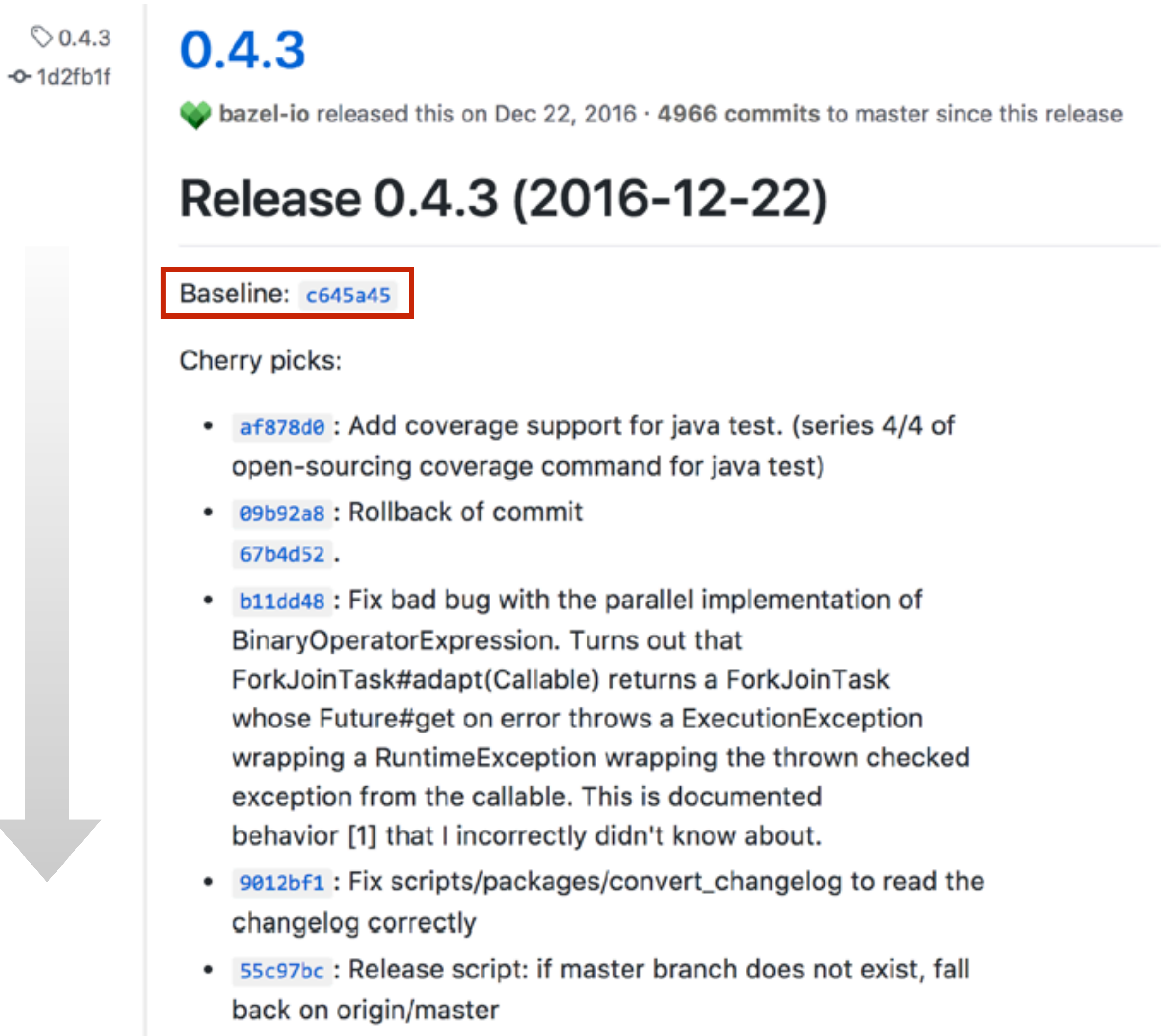
- `af878d0`: Add coverage support for java test. (series 4/4 of open-sourcing coverage command for java test)
- `09b92a8`: Rollback of commit `67b4d52`.
- `b11dd48`: Fix bad bug with the parallel implementation of BinaryOperatorExpression. Turns out that ForkJoinTask#adapt(Callable) returns a ForkJoinTask whose Future#get on error throws a ExecutionException wrapping a RuntimeException wrapping the thrown checked exception from the callable. This is documented behavior [1] that I incorrectly didn't know about.
- `9012bf1`: Fix scripts/packages/convert_changelog to read the changelog correctly
- `55c97bc`: Release script: if master branch does not exist, fall back on origin/master

...

Bazel

{Fast, Correct} - Choose two




0.4.3
1d2fb1f

0.4.3

bazel-io released this on Dec 22, 2016 · **4966 commits** to master since this release

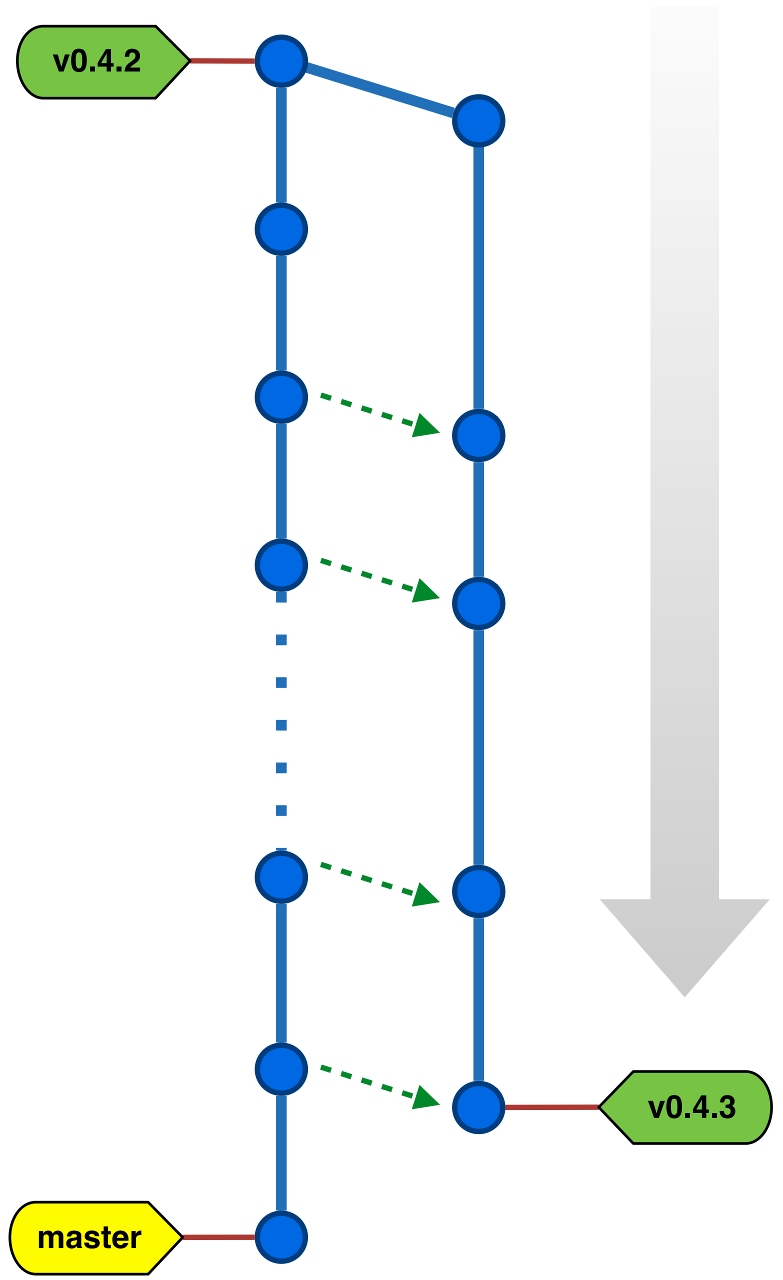
Release 0.4.3 (2016-12-22)

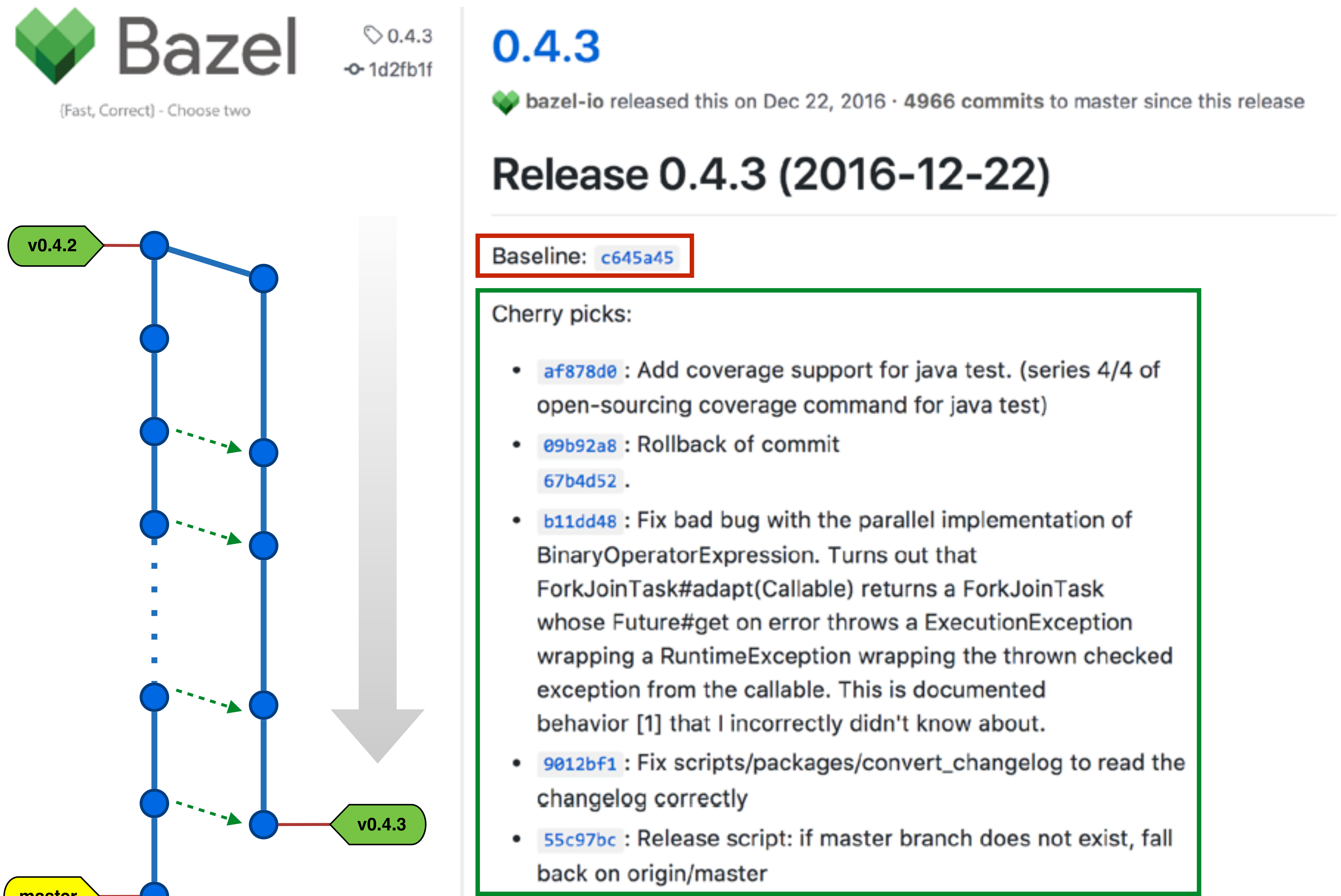
Baseline: `c645a45`

Cherry picks:

- `af878d0`: Add coverage support for java test. (series 4/4 of open-sourcing coverage command for java test)
- `09b92a8`: Rollback of commit `67b4d52`.
- `b11dd48`: Fix bad bug with the parallel implementation of BinaryOperatorExpression. Turns out that ForkJoinTask#adapt(Callable) returns a ForkJoinTask whose Future#get on error throws a ExecutionException wrapping a RuntimeException wrapping the thrown checked exception from the callable. This is documented behavior [1] that I incorrectly didn't know about.
- `9012bf1`: Fix scripts/packages/convert_changelog to read the changelog correctly
- `55c97bc`: Release script: if master branch does not exist, fall back on origin/master

• • •





Bazel

{Fast, Correct} - Choose two

0.4.3
1d2fb1f

0.4.3

bazel-io released this on Dec 22, 2016 · **4966 commits** to master since this release

Release 0.4.3 (2016-12-22)

Baseline: `c645a45`

Cherry picks:

- `af878d0`: Add coverage support for java test. (series 4/4 of open-sourcing coverage command for java test)
- `09b92a8`: Rollback of commit `67b4d52`.
- `b11dd48`: Fix bad bug with the parallel implementation of BinaryOperatorExpression. Turns out that ForkJoinTask#adapt(Callable) returns a ForkJoinTask whose Future#get on error throws a ExecutionException wrapping a RuntimeException wrapping the thrown checked exception from the callable. This is documented behavior [1] that I incorrectly didn't know about.
- `9012bf1`: Fix scripts/packages/convert_changelog to read the changelog correctly
- `55c97bc`: Release script: if master branch does not exist, fall back on origin/master

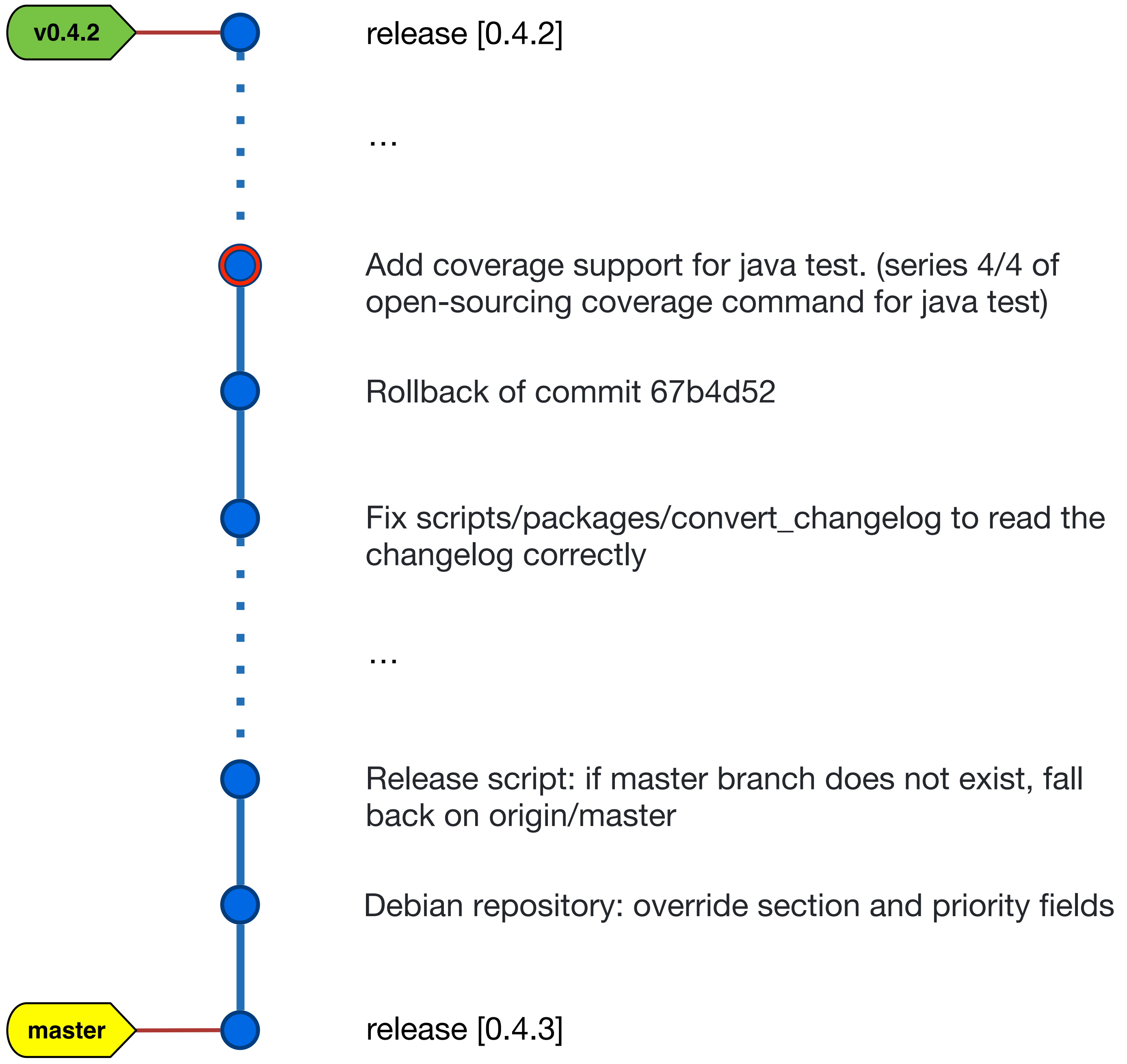
• • •

Pitfalls of Cherry-Picking

Dec 2, 2016

58 authors
279 commits
479 files changed

Dec 22, 2016



release [0.4.2]

…

Add coverage support for java test. (series 4/4 of open-sourcing coverage command for java test)

Rollback of commit 67b4d52

Fix scripts/packages/convert_changelog to read the changelog correctly

…

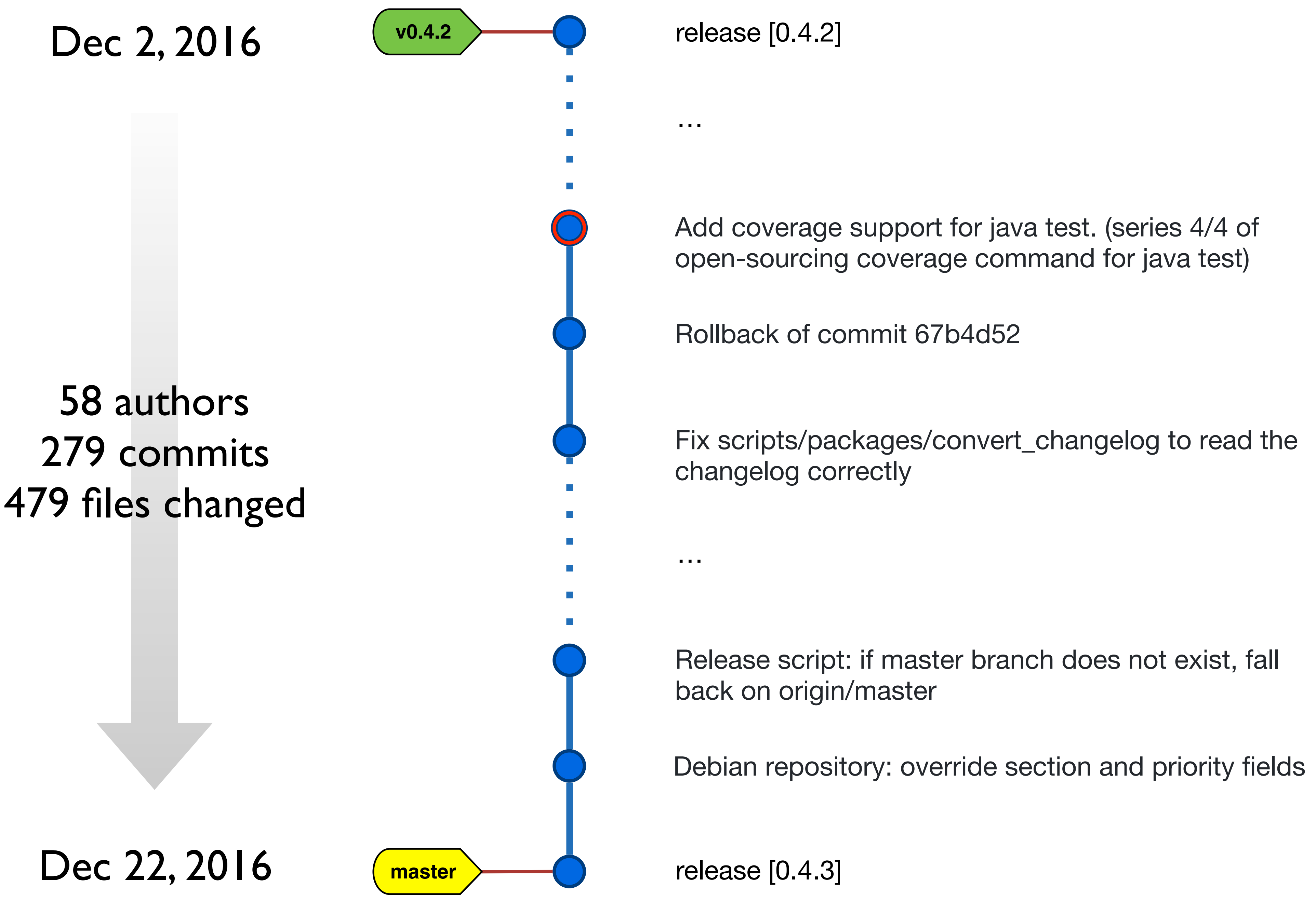
Release script: if master branch does not exist, fall back on origin/master

Debian repository: override section and priority fields

release [0.4.3]

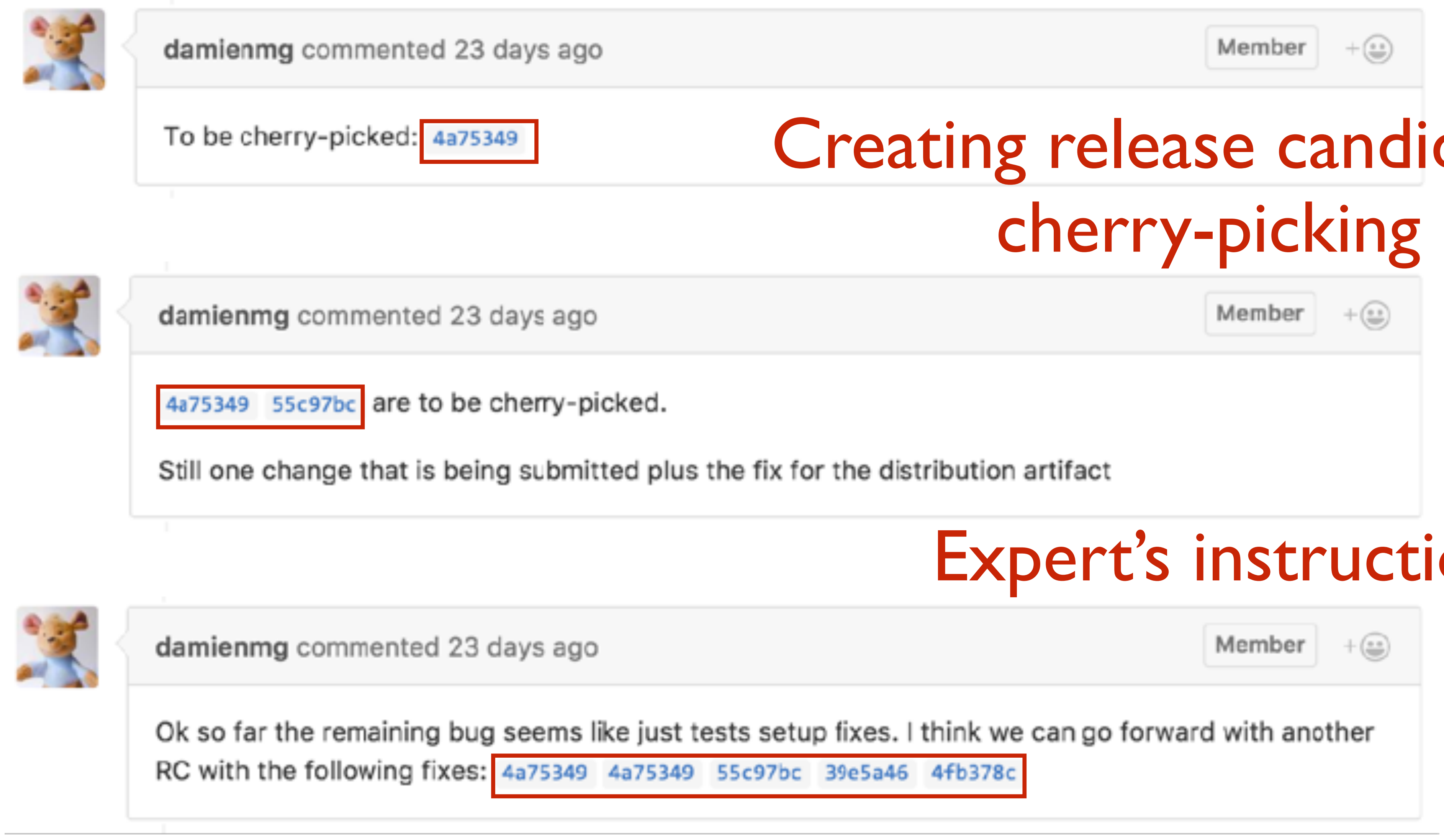
https://github.com/bazelbuild/bazel/issues/2246

Pitfalls of Cherry-Picking


Dec 2, 2016
58 authors
279 commits
479 files changed
Dec 22, 2016
v0.4.2
master
release [0.4.2]
…
Add coverage support for java test. (series 4/4 of open-sourcing coverage command for java test)
Rollback of commit 67b4d52
Fix scripts/packages/convert_changelog to read the changelog correctly
…
Release script: if master branch does not exist, fall back on origin/master
Debian repository: override section and priority fields
release [0.4.3]

https://github.com/bazelbuild/bazel/issues/2246

Pitfalls of Cherry-Picking



damienmg commented 23 days ago
Member

To be cherry-picked: 4a75349

damienmg commented 23 days ago
Member

4a75349 55c97bc are to be cherry-picked.

Still one change that is being submitted plus the fix for the distribution artifact

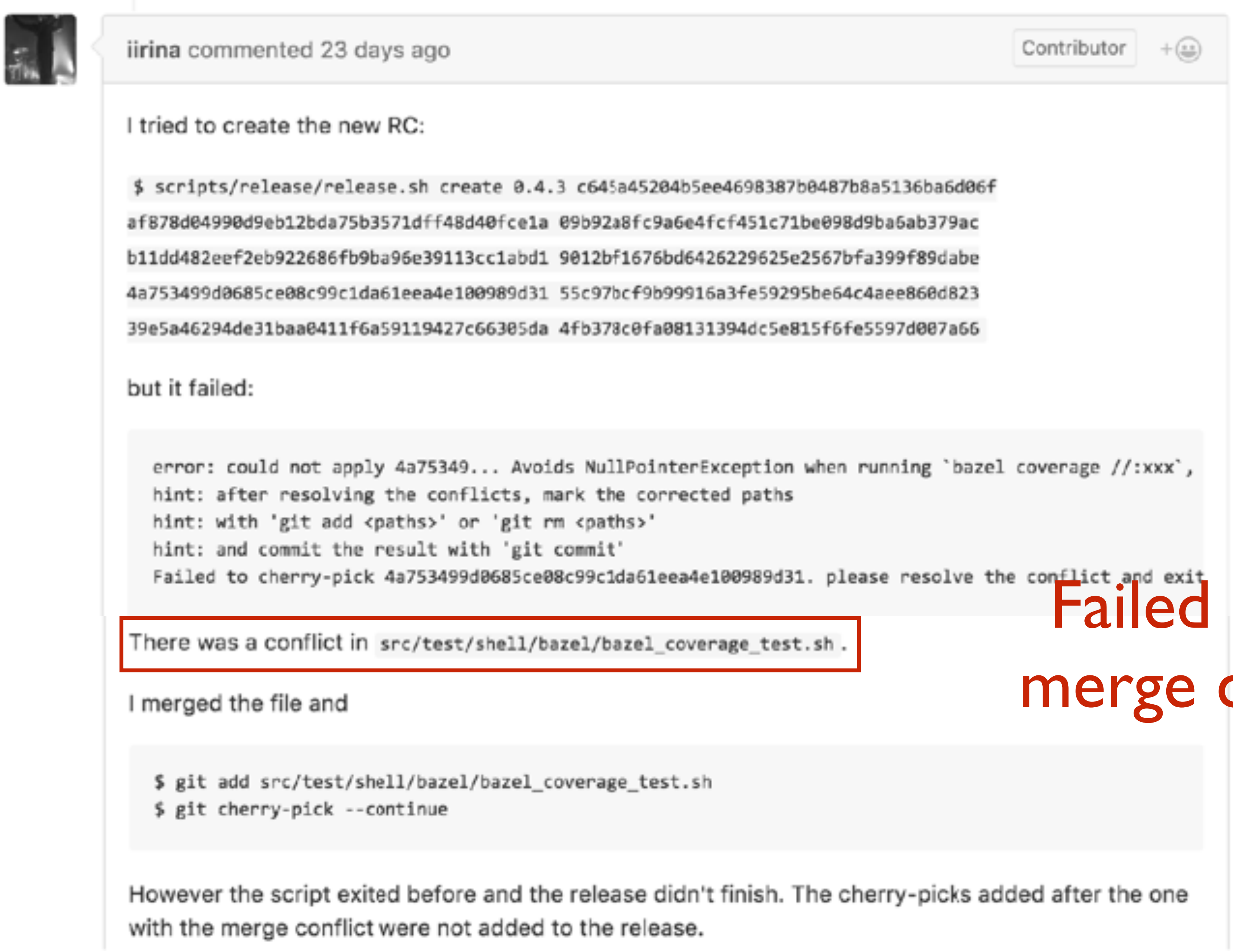
damienmg commented 23 days ago
Member

Ok so far the remaining bug seems like just tests setup fixes. I think we can go forward with another RC with the following fixes: 4a75349 4a75349 55c97bc 39e5a46 4fb378c

Creating release candidate by cherry-picking

Expert's instructions…

Pitfalls of Cherry-Picking



iirina commented 23 days ago

Contributor

I tried to create the new RC:

```
$ scripts/release/release.sh create 0.4.3 c645a45204b5ee4698387b0487b8a5136ba6d06f
af878d04990d9eb12bda75b3571dff48d40fce1a 09b92a8fc9a6e4fcf451c71be098d9ba6ab379ac
b11dd482eef2eb922686fb9ba96e39113cc1abd1 9012bf1676bd6426229625e2567bfa399f89dabe
4a753499d0685ce08c99c1da61eea4e100989d31 55c97bcf9b99916a3fe59295be64c4aee860d823
39e5a46294de31baa0411f6a59119427c66305da 4fb378c0fa08131394dc5e815f6fe5597d007a66
```

but it failed:

```
error: could not apply 4a75349... Avoids NullPointerException when running `bazel coverage //:xxx`,
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
Failed to cherry-pick 4a753499d0685ce08c99c1da61eea4e100989d31. please resolve the conflict and exit
```

There was a conflict in `src/test/shell/bazel/bazel_coverage_test.sh` .

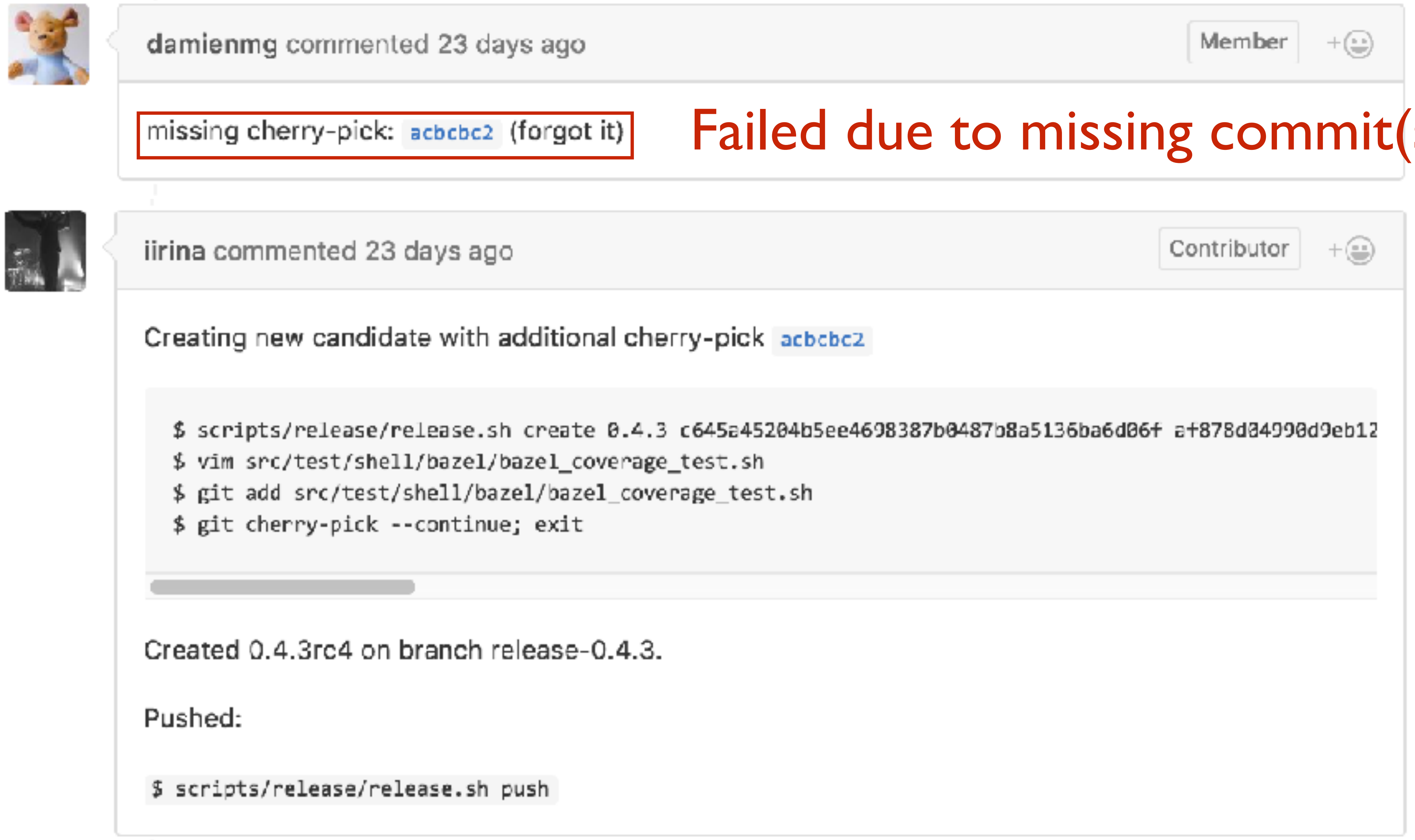
I merged the file and

```
$ git add src/test/shell/bazel/bazel_coverage_test.sh
$ git cherry-pick --continue
```

However the script exited before and the release didn't finish. The cherry-picks added after the one with the merge conflict were not added to the release.

Failed due to merge conflicts

Pitfalls of Cherry-Picking



damienmg commented 23 days ago
Member

missing cherry-pick: acbcbc2 (forgot it)

Failed due to missing commit(s)

iirina commented 23 days ago
Contributor

Creating new candidate with additional cherry-pick acbcbc2

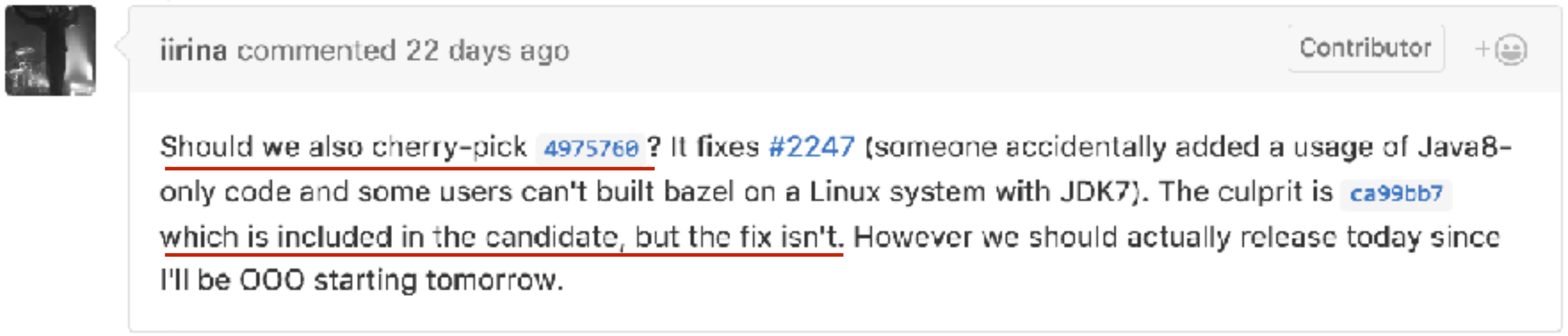
```
$ scripts/release/release.sh create 0.4.3 c645a45204b5ee4698387b0487b8a5136ba6d06f af878d84990d9eb12
$ vim src/test/shell/bazel/bazel_coverage_test.sh
$ git add src/test/shell/bazel/bazel_coverage_test.sh
$ git cherry-pick --continue; exit
```

Created 0.4.3rc4 on branch release-0.4.3.

Pushed:

```
$ scripts/release/release.sh push
```

Pitfalls of Cherry-Picking



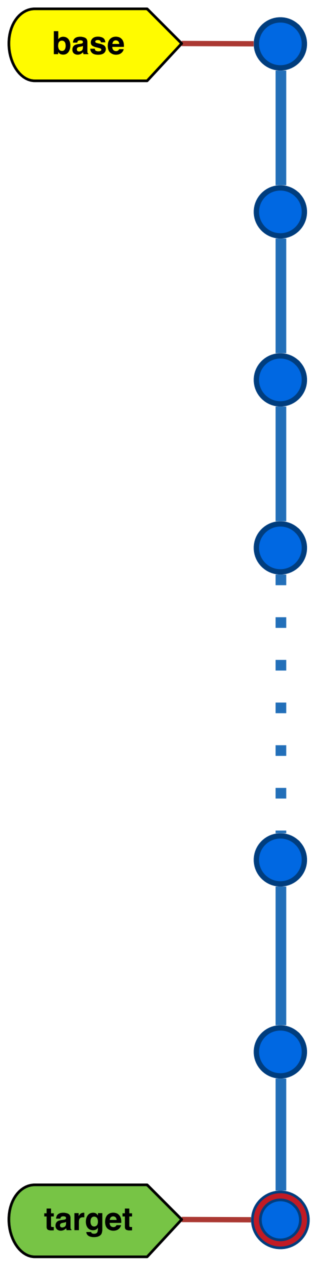
iirina commented 22 days ago

Contributor

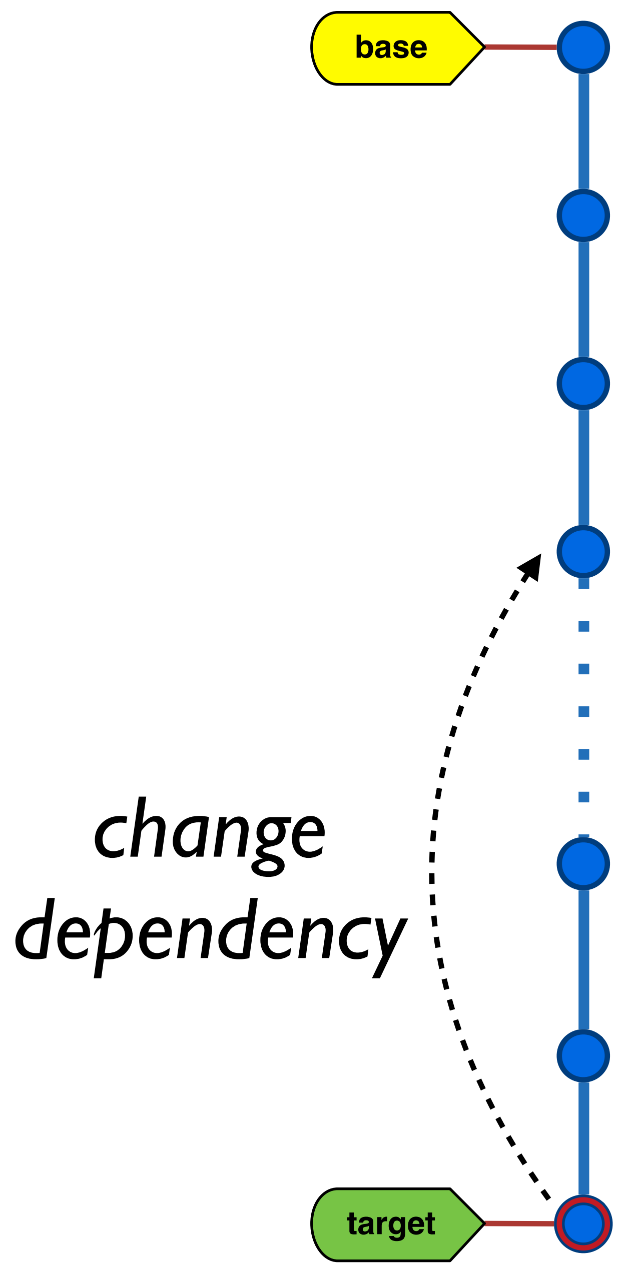
Should we also cherry-pick 4975760? It fixes #2247 (someone accidentally added a usage of Java8-only code and some users can't built bazel on a Linux system with JDK7). The culprit is ca99bb7 which is included in the candidate, but the fix isn't. However we should actually release today since I'll be OOO starting tomorrow.

Failed due to missing commit(s)
Missing a fix this time.

Why is it so hard?



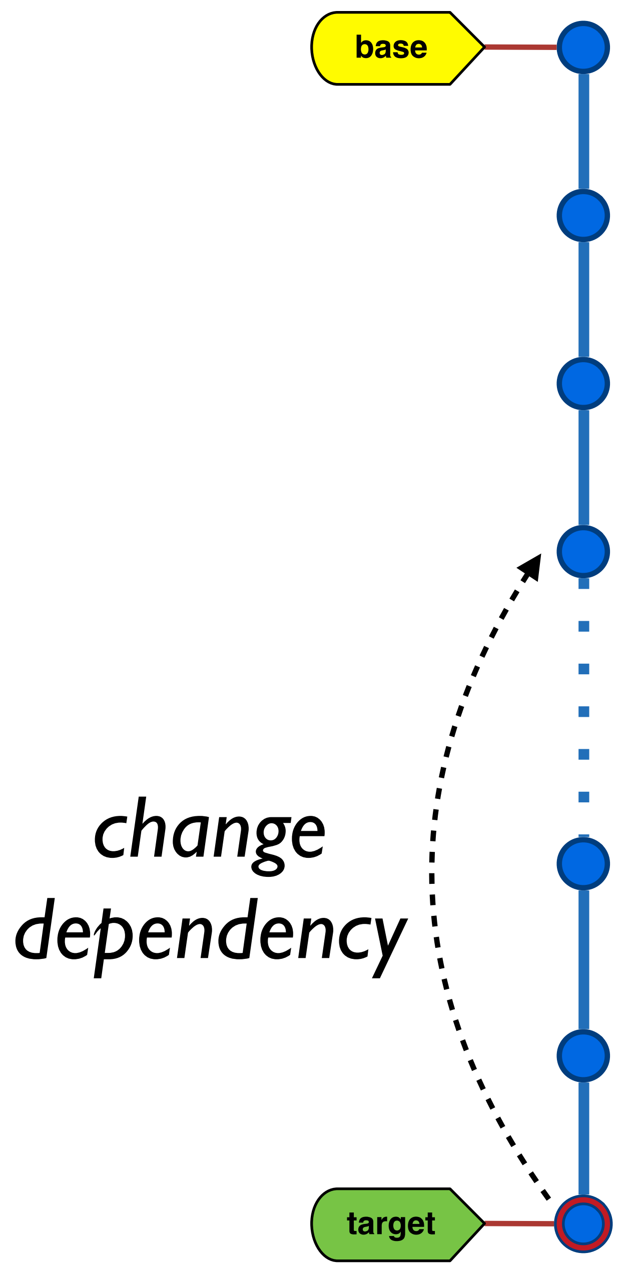
Why is it so hard?



Options?

1. ✗ Pick target commit(s)
2. Pick the entire history
3. Manually identify necessary commits

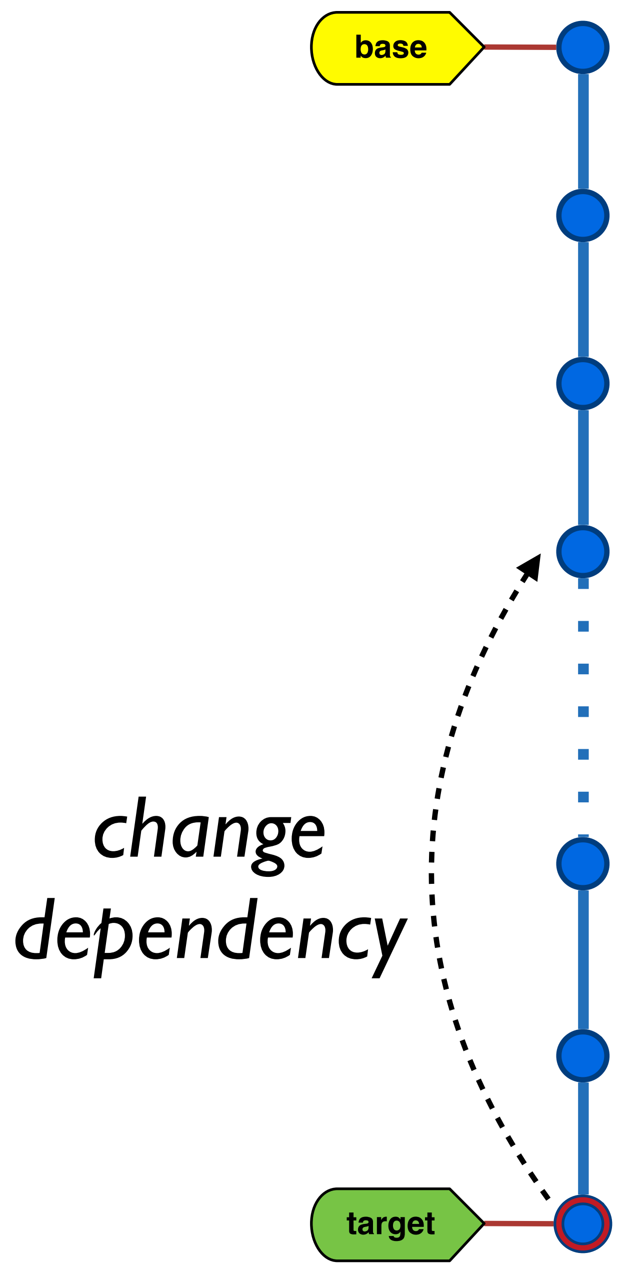
Why is it so hard?



Options?

1. ✗ Pick target commit(s)
2. ✗ Pick the entire history
3. Manually identify necessary commits

Why is it so hard?



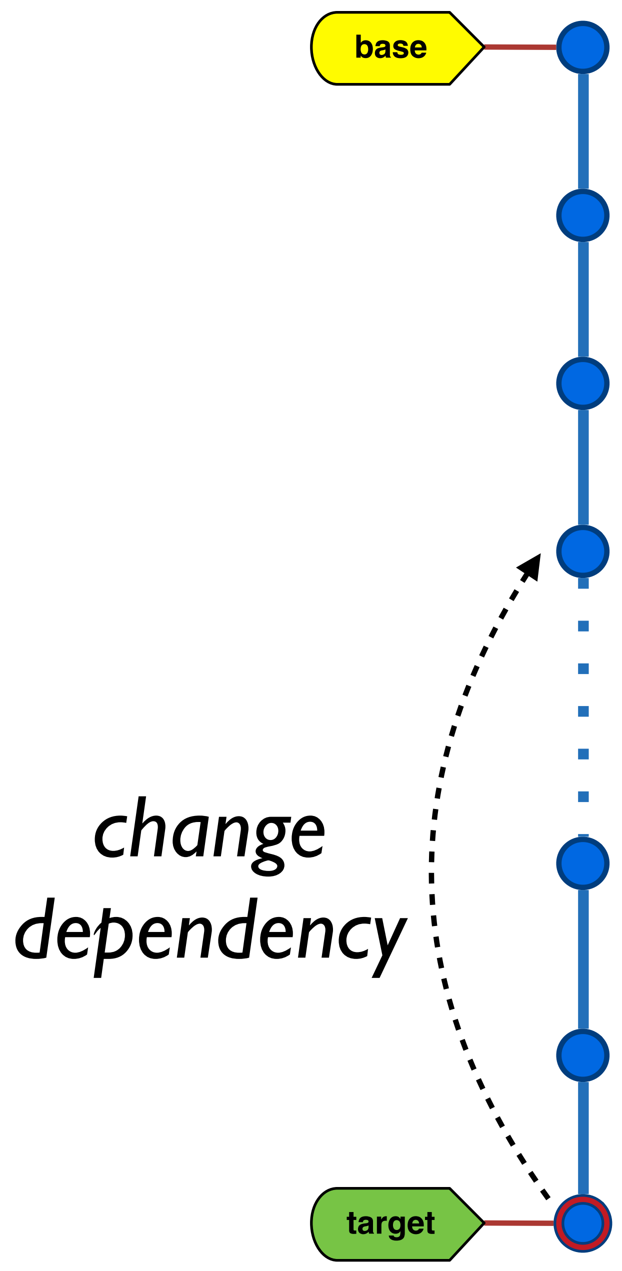
Options?

✗ 1. Pick target commit(s)

✗ 2. Pick the entire history

✗ 3. Manually identify necessary commits

Why is it so hard?



Options?

✗ 1. Pick target commit(s)

✗ 2. Pick the entire history

✗ 3. Manually identify necessary commits

Existing version control tools:

- Code treated as plain texts
- Do not understand the semantics
- User provided semantic/logical grouping is inaccurate!

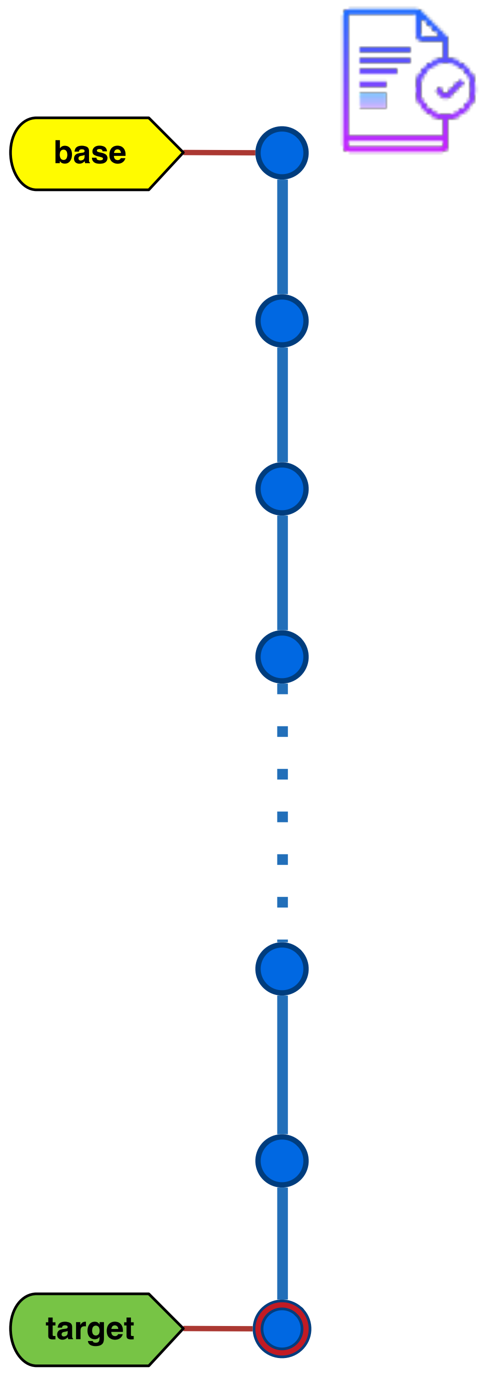
Challenges in Evolution Management

Gaps between individual *changes* and high-level system *semantics*:

- the history re-structuring challenge
- the change isolation and migration challenge
- the variability reverse engineering challenge
- and more …

This thesis aims to address some of them

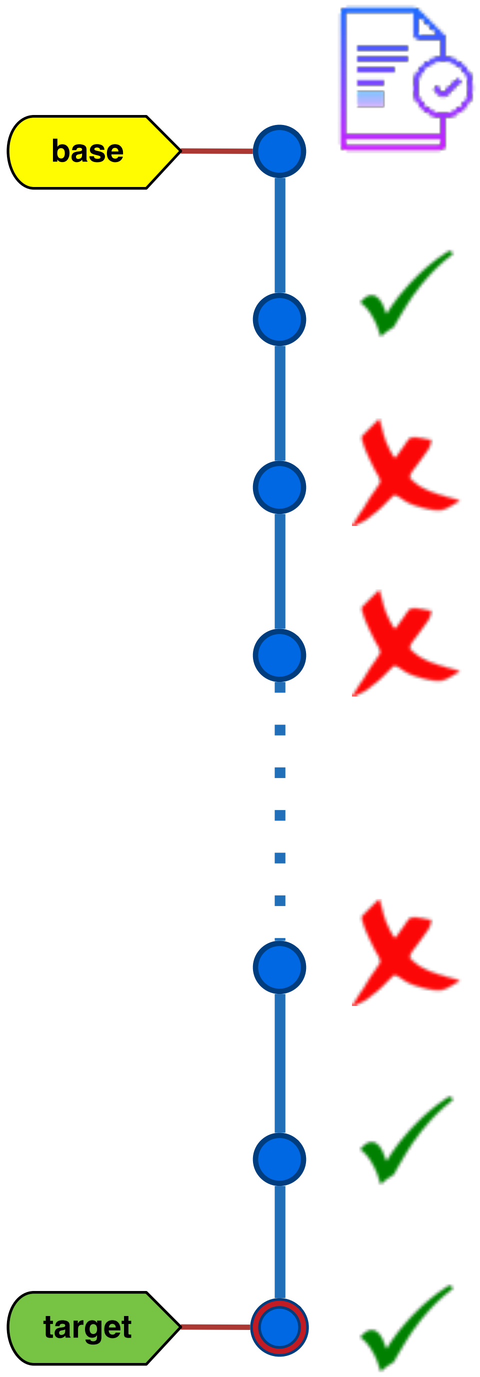
What can we do?



Exploit existing artifacts:

- Strictly *structured data*
- Well-defined language *syntax* and *semantics*
- Carefully designed *test suites*

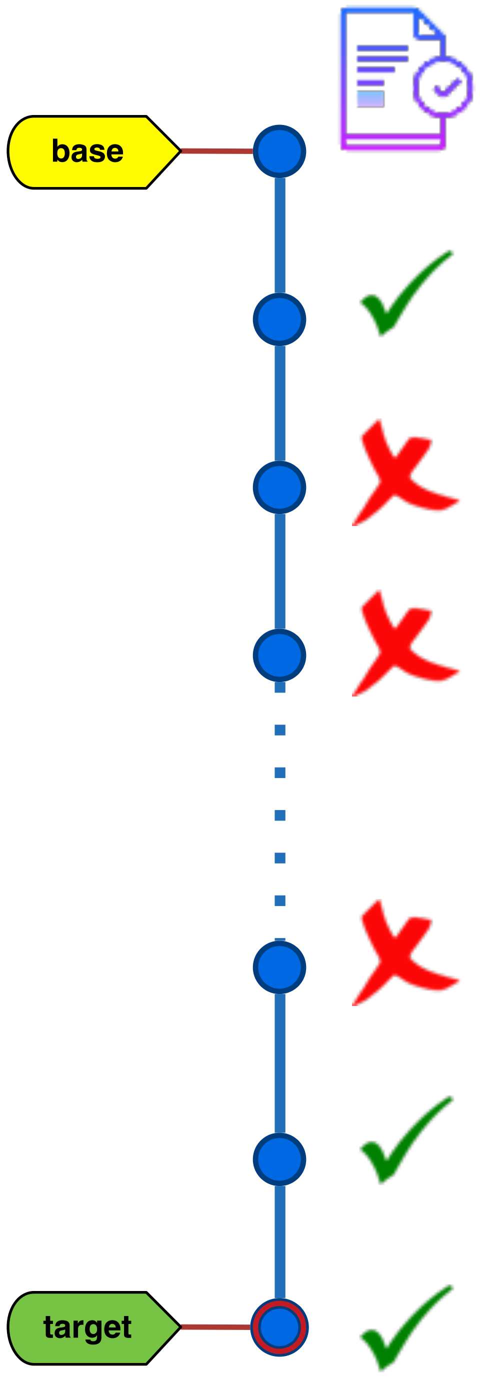
What can we do?



Exploit existing artifacts:

- Strictly *structured data*
- Well-defined language *syntax* and *semantics*
- Carefully designed *test suites*

Solution: Semantic History Slicing



Exploit existing artifacts:

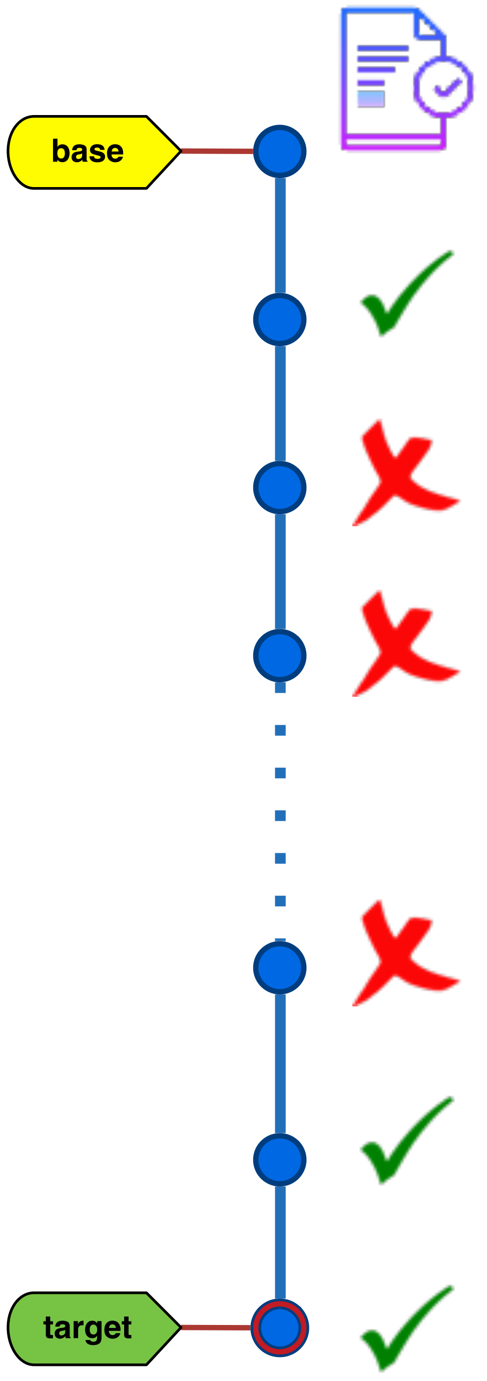
- Strictly *structured data*
- Well-defined language *syntax* and *semantics*
- Carefully designed *test suites*

History:
sequence of commits
+
Criterion:
set of tests

→

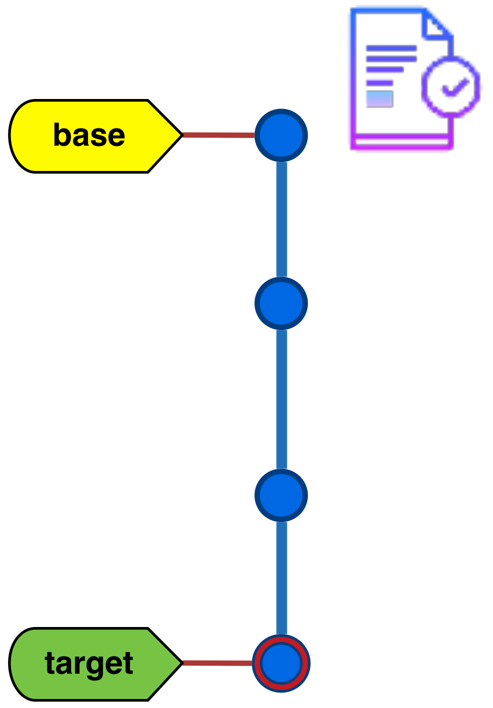
Sub-history:
well-formed: compiles
&
semantic preserving: passing tests

Solution: Semantic History Slicing

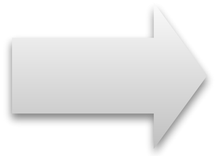


Exploit existing artifacts:

- Strictly *structured data*
- Well-defined language *syntax* and *semantics*
- Carefully designed *test suites*

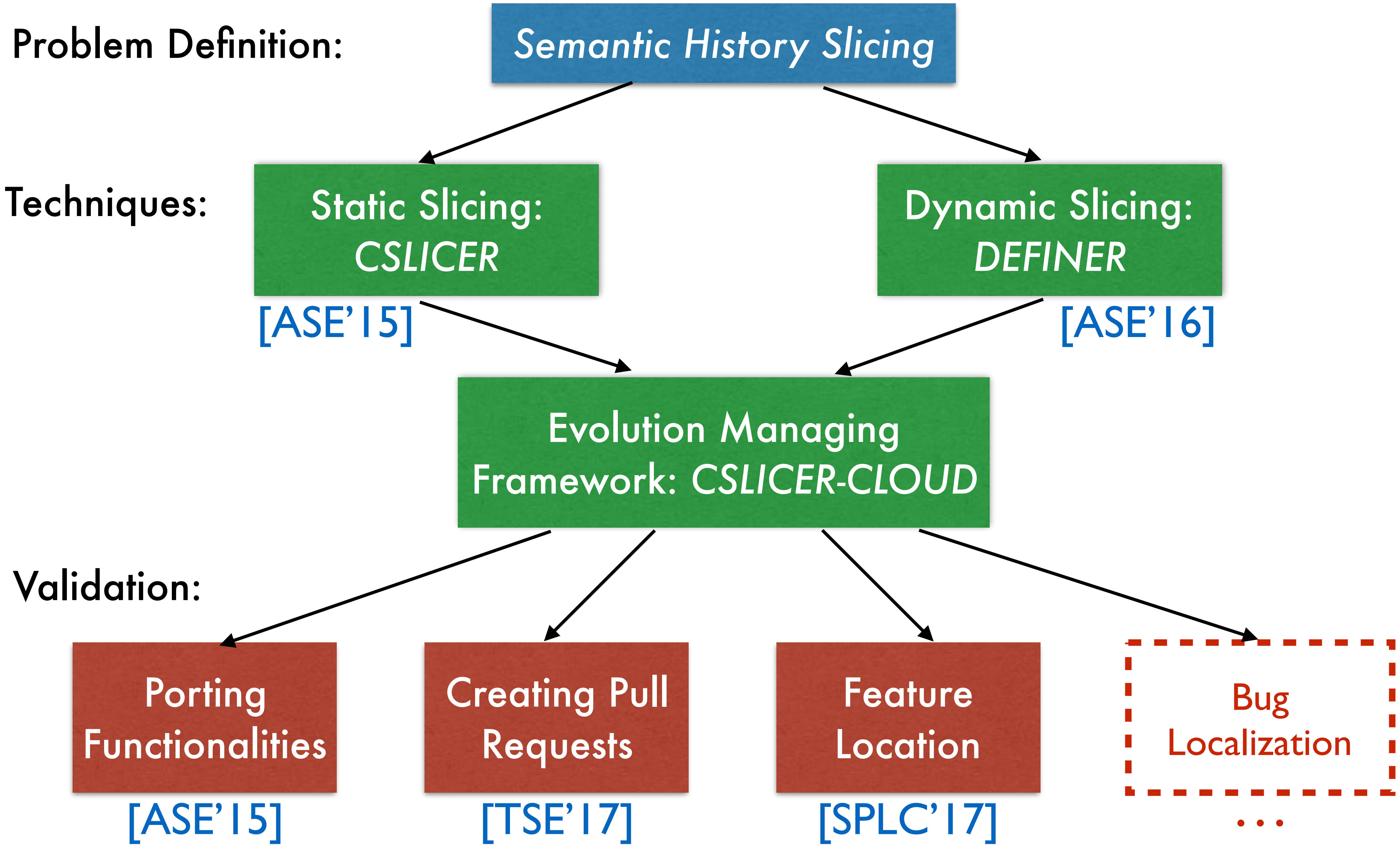


History:
sequence of commits
+
Criterion:
set of tests

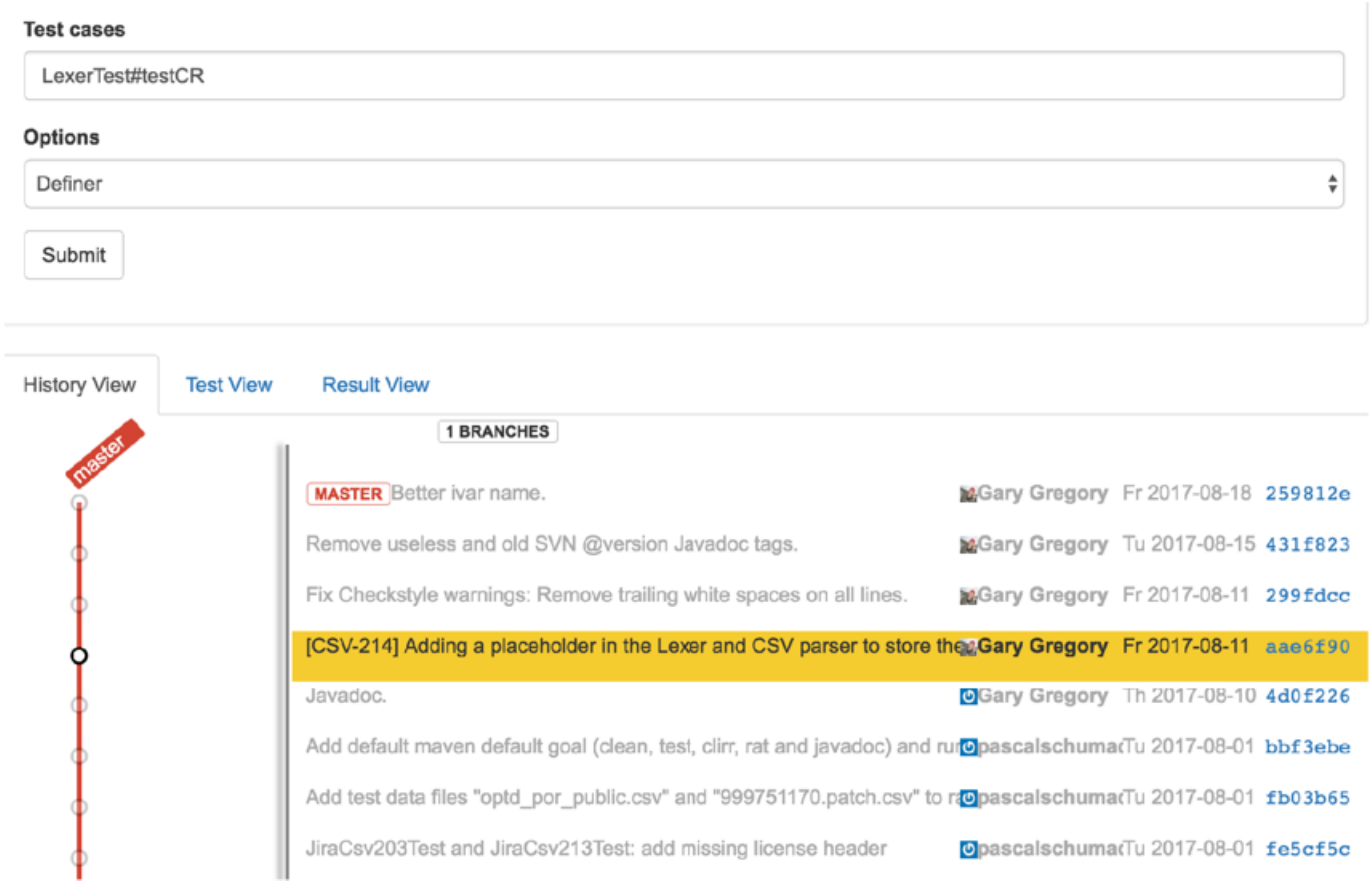


Sub-history:
well-formed: compiles
&
semantic preserving: passing tests

Research Map



CSLICER-CLOUD


Test cases

LexerTest#testCR

Options

Definer

MASTER Better ivar name.	Gary Gregory	Fr 2017-08-18	259812e
Remove useless and old SVN @version Javadoc tags.	Gary Gregory	Tu 2017-08-15	431f823
Fix Checkstyle warnings: Remove trailing white spaces on all lines.	Gary Gregory	Fr 2017-08-11	299fdcc
[CSV-214] Adding a placeholder in the Lexer and CSV parser to store the	Gary Gregory	Fr 2017-08-11	aae6f90
Javadoc.	Gary Gregory	Th 2017-08-10	4d0f226
Add default maven default goal (clean, test, clirr, rat and javadoc) and ru	pascalschuma	Tu 2017-08-01	bbf3ebe
Add test data files "optd_por_public.csv" and "999751170.patch.csv" to r	pascalschuma	Tu 2017-08-01	fb03b65
JiraCsv203Test and JiraCsv213Test: add missing license header	pascalschuma	Tu 2017-08-01	fe5cf5c

http://www.cs.toronto.edu/~liyi/cslicer

Validation

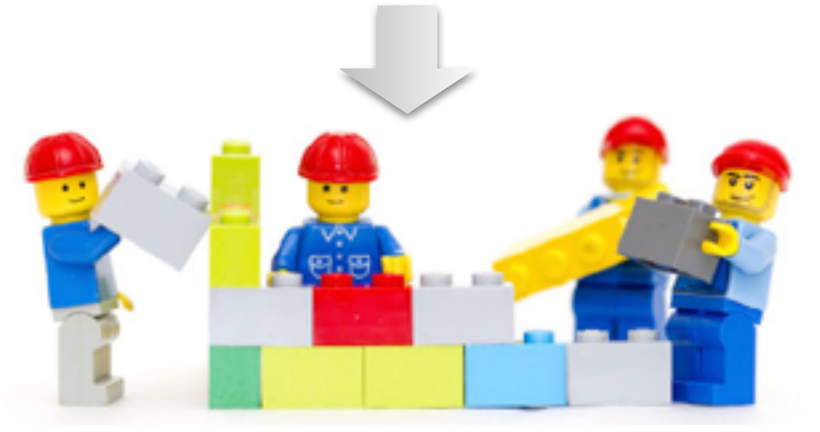
A dataset for semantic changes in version histories [MSR'17]

- 98 items of semantic change data
- Collected from 10 open-source Java projects
- Ground-truth obtained from *developer's documentation* and brute-force *minimization*
- Available at: https://github.com/Chenguang-Zhu/DoSC

Feature Location for SPLE

The "top-down" approach

core assets (features)



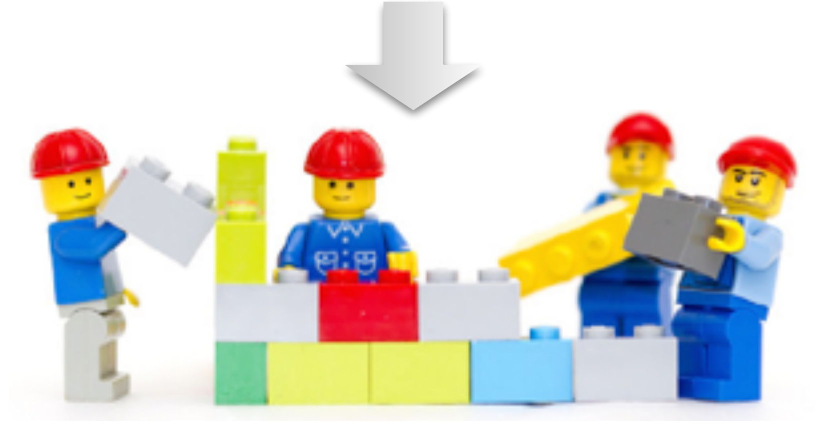
configurations + feature model

product outputs

Feature Location for SPLE

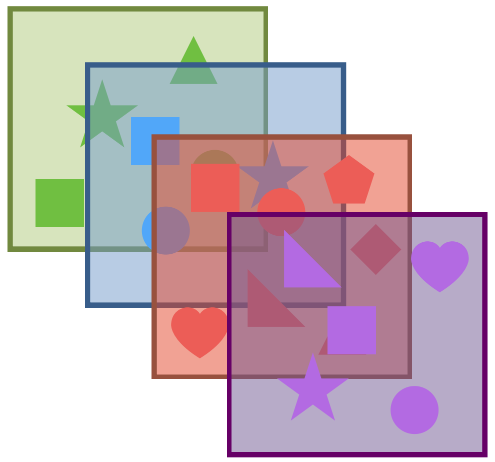
The "top-down" approach

core assets (features)



configurations + feature model

product outputs



product variants

The "bottom-up" approach

Feature Location for SPLE

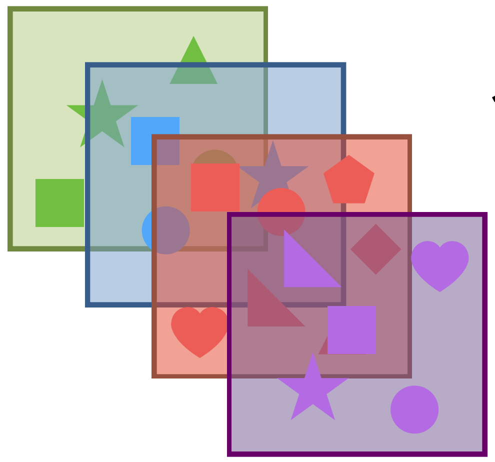
The "top-down" approach

core assets (features)

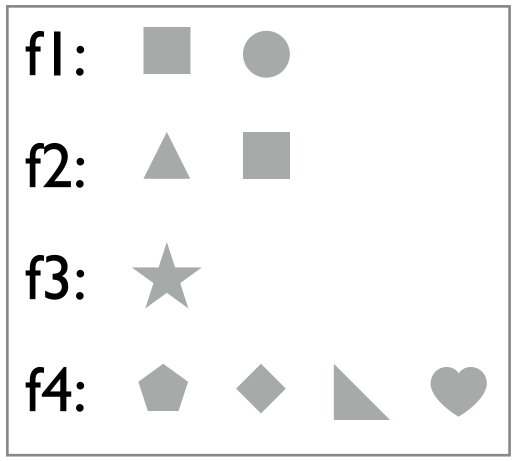
configurations + feature model

product outputs

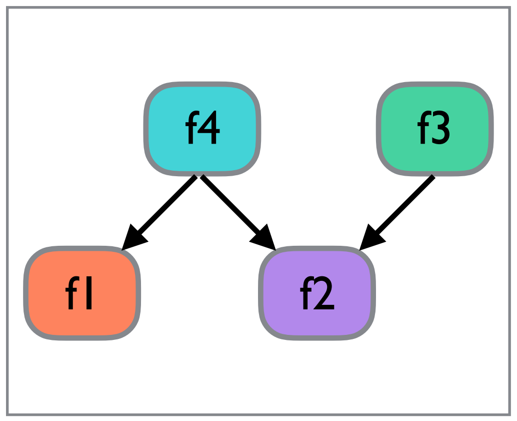
From "ad-hoc" to "systematic"



product variants



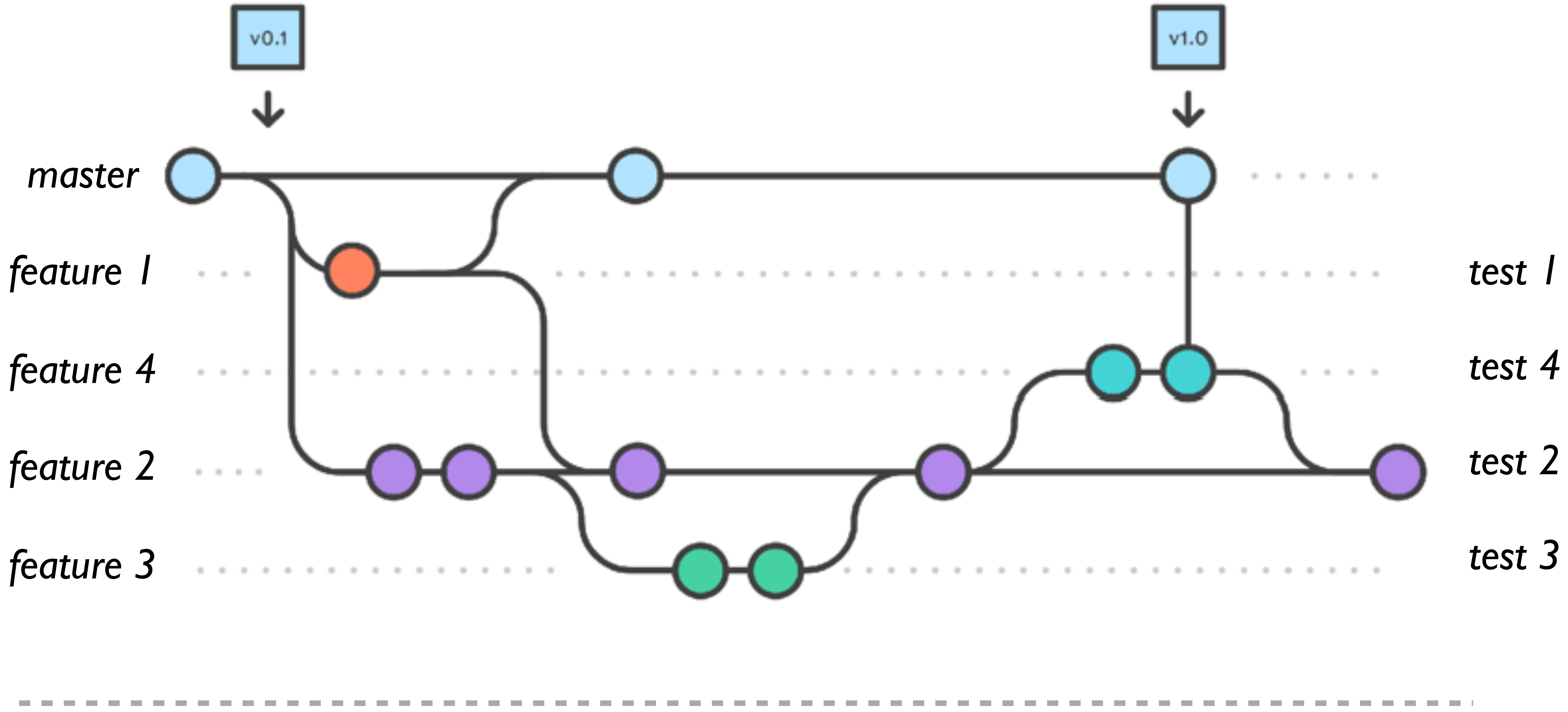
1. feature implementations (core assets)



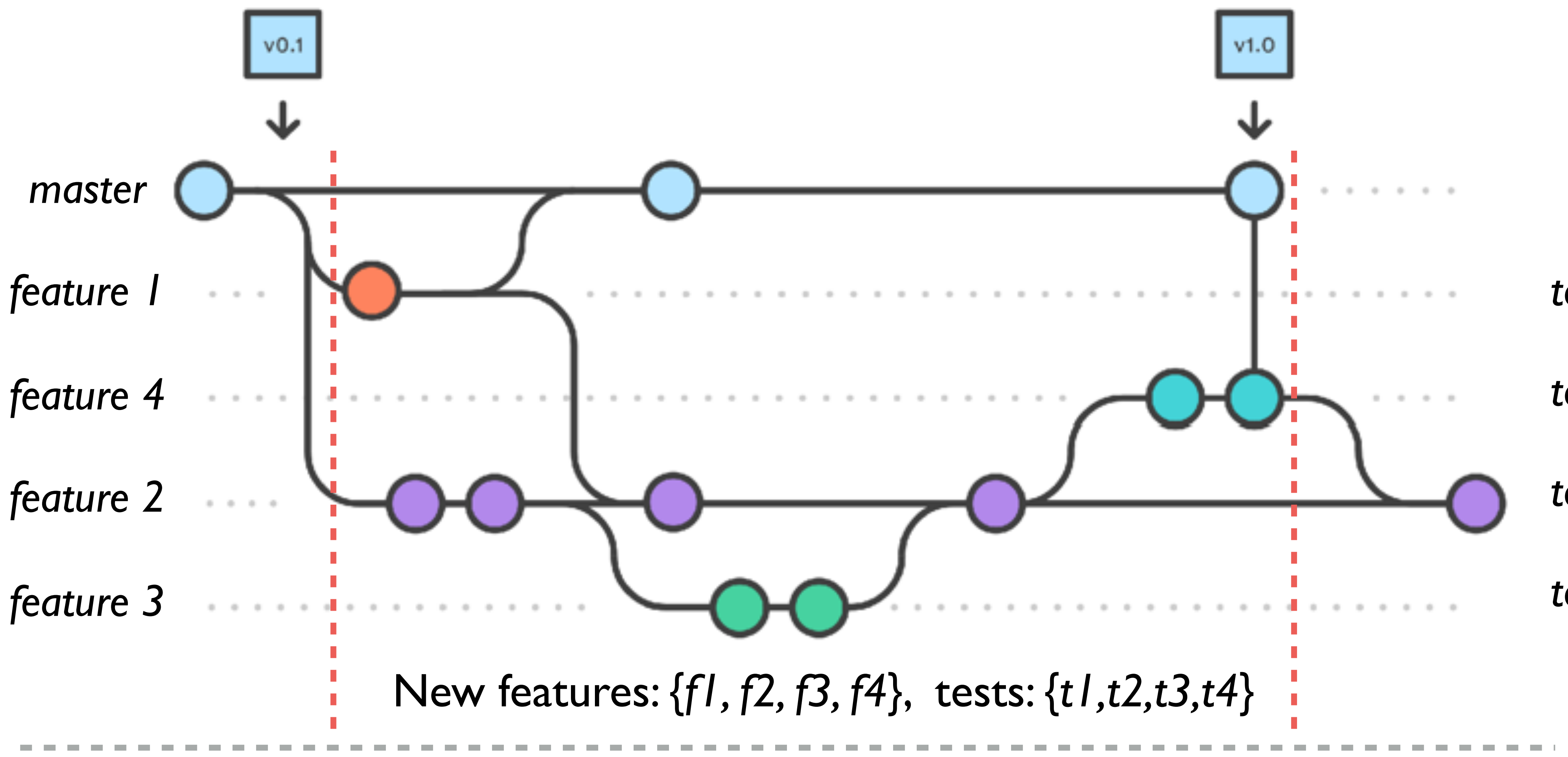
2. feature relationships (feature models)

The "bottom-up" approach

Feature Location in Version Histories

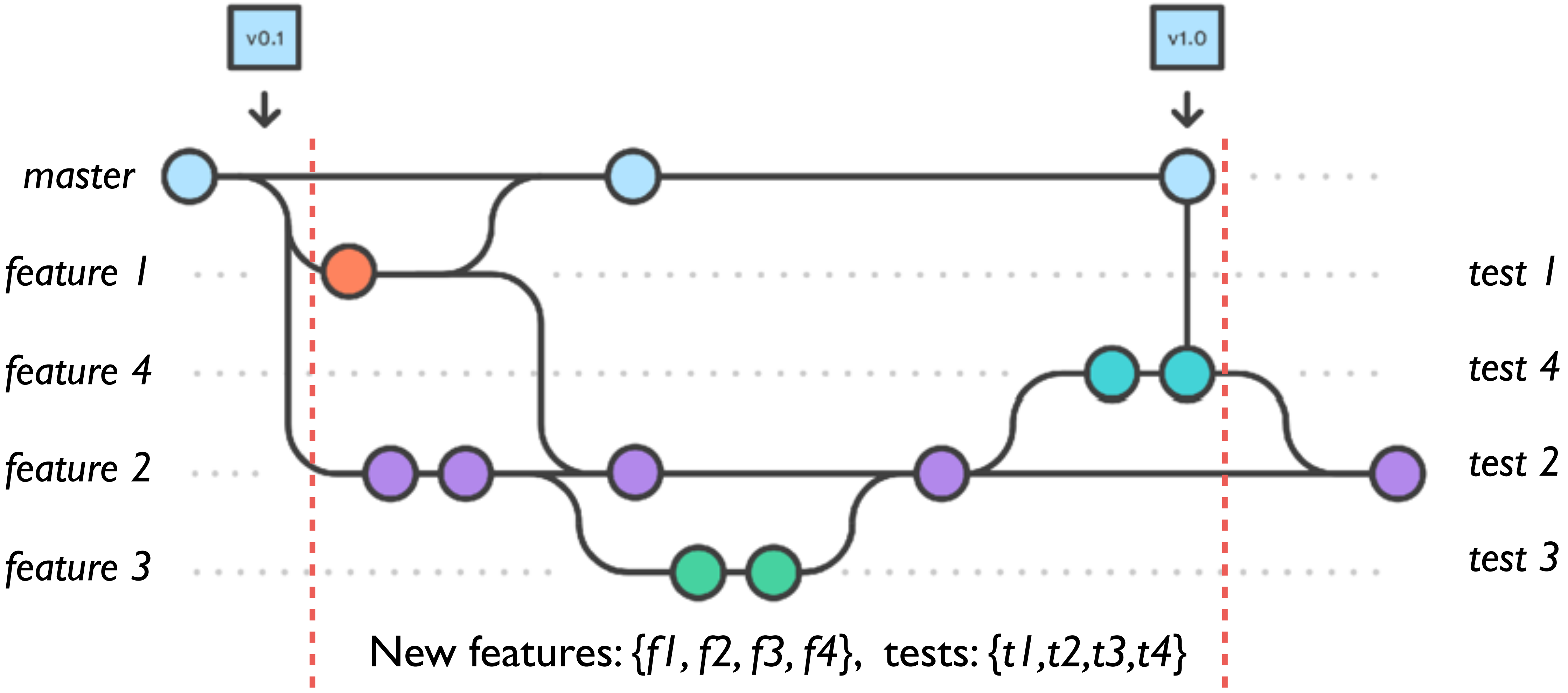


Feature Location in Version Histories

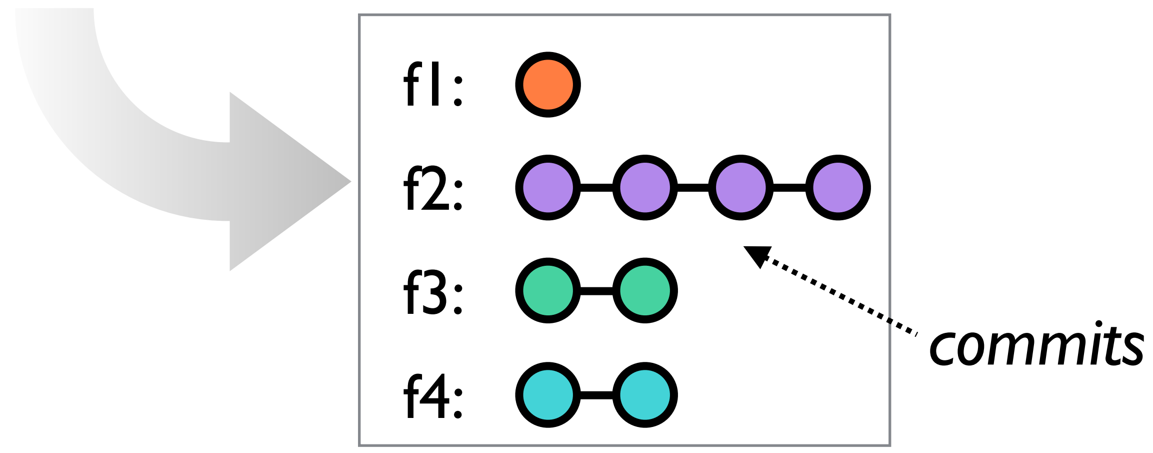


New features: {*f1*, *f2*, *f3*, *f4*}, tests: {*t1*,*t2*,*t3*,*t4*}

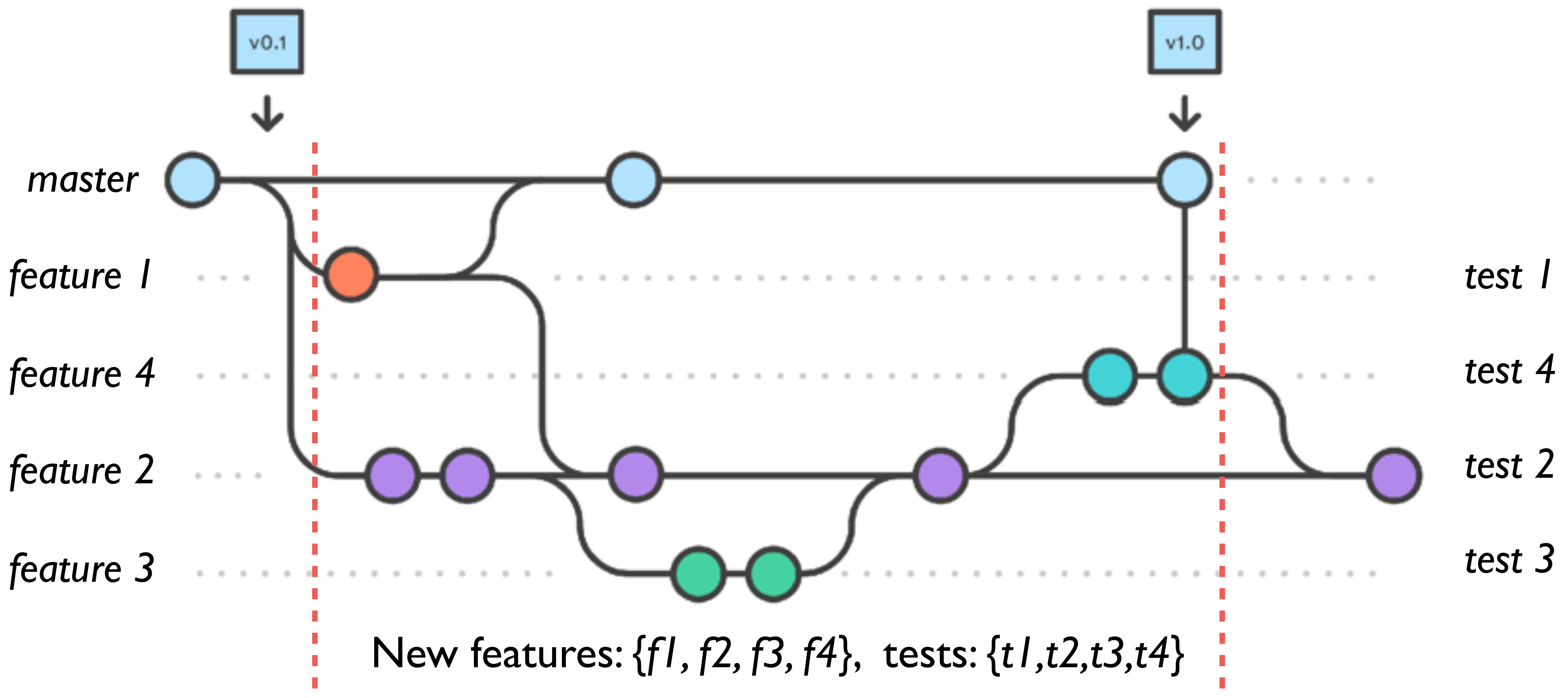
Feature Location in Version Histories



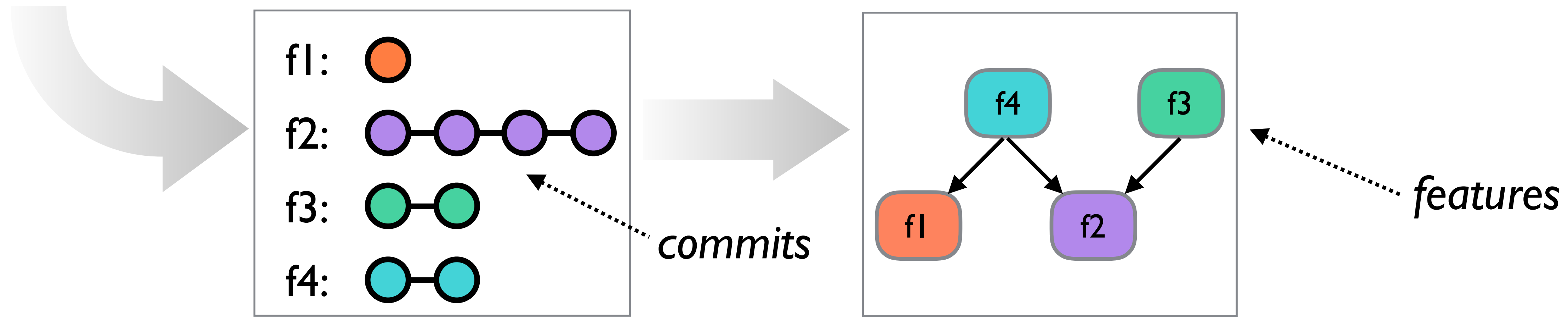
New features: {*f1*, *f2*, *f3*, *f4*}, tests: {*t1*,*t2*,*t3*,*t4*}



Feature Location in Version Histories



New features: {*f1*, *f2*, *f3*, *f4*}, tests: {*t1*,*t2*,*t3*,*t4*}



Status and Next Steps

Current Status (a few months before graduation)

- Wrapping up thesis work
- Preparing a journal paper summarizing this line of work

Future Plans (comments and discussions please!)

- Deep integration with development tool chains
  - *semantics-aware cherry-picking*
- Leveraging the social aspect of software development
  - *developer conversations*
  - *log messages*
- User studies

Related Work

History Understanding and Manipulation

- History transformation [Muslu et al.][Servant & Jones]
- Change classifications [Falleri et al.]

Change Impact Analysis

- Chianti [Ren et al.], FaultTracer [Zhang et al.]
- Ekstazi [Gligoric et al.]

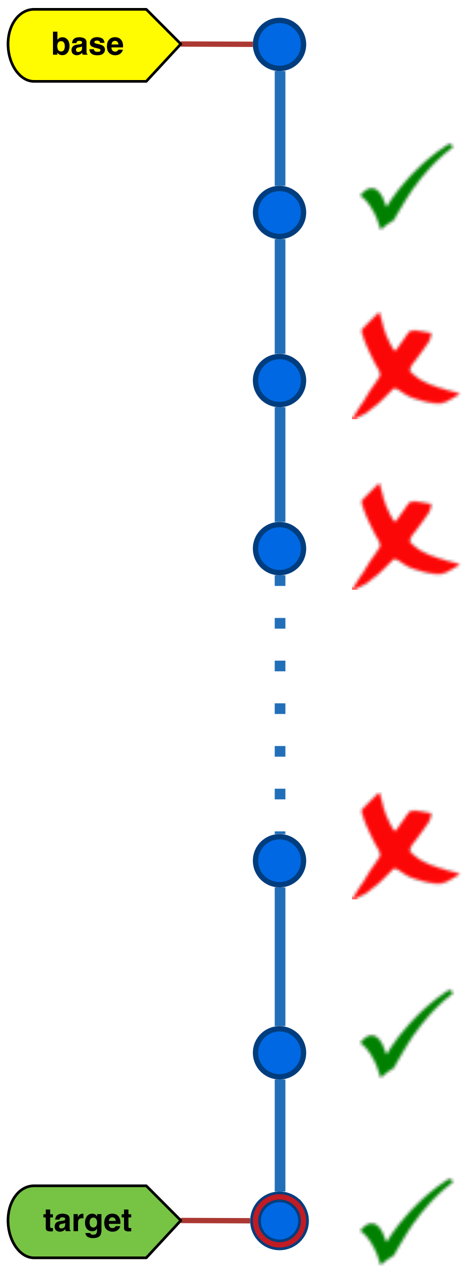
Bug Localization

- Delta debugging [Zeller et al.], Selective Bisection Debugging [Saha & Gligoric]
- Information retrieval-based approaches [Wang & Lo][Saha et al.]

Feature Location

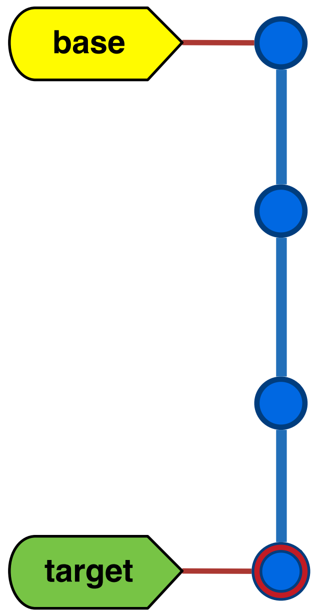
- Dynamic FL, Software Reconnaissance [Wilde & Scully]
- Managing cloned variants [Rubin et al.]

Summary



Semantic History Slicing:

- Bridging gaps between text changes and program semantics
- Many applications in evolution management
- Validation on open source Git repositories



Bibliography

Chenguang Zhu, Yi Li, Julia Rubin, and Marsha Chechik. A Dataset for Dynamic Discovery of Semantic Changes in Version Controlled Software Histories. In *Proceedings of the 14th International Conference on Mining Software Repositories*, pages 523–526, Piscataway, NJ, USA, 2017. IEEE Press.

Andreas Zeller and Ralf Hildebrandt. Simplifying and Isolating Failure-Inducing Input. *IEEE Transactions on Software Engineering*, 28(2):183–200, 2002.

Yi Li, Chenguang Zhu, Julia Rubin, and Marsha Chechik. Semantic Slicing of Software Version Histories. *IEEE Transactions on Software Engineering*, 2017.

Yi Li, Chenguang Zhu, Julia Rubin, and Marsha Chechik. FHistorian: Locating Features in Version Histories. In *Proceedings of the 21st International Systems and Software Product Line Conference - Volume A*, pages 49–58, New York, NY, USA, 2017. ACM.

Yi Li, Chenguang Zhu, Julia Rubin, and Marsha Chechik. Precise Semantic History Slicing through Dynamic Delta Refinement. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pages 495–506, Singapore, Singapore, September 2016.

Yi Li, Julia Rubin, and Marsha Chechik. Semantic Slicing of Software Version Histories. In *Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering*, pages 686–696, Lincoln, NE, USA, November 2015.

Bazel: a Fast, Scalable, Muti-Language, and Extensible Build System. https://bazel. build, 2017.

Kivanç Muşlu, Luke Swart, Yuriy Brun, and Michael D. Ernst. Development History Granularity Transformations. In *Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering*, pages 697–702, Lincoln, NE, USA, November 2015.

Shaowei Wang and David Lo. AmaLgam+: Composing Rich Information Sources for Accurate Bug Localization. *Journal of Software: Evolution and Process*, 28(10):921–942, 2016.

Yun Zhang, David Lo, Xin Xia, Tien-Duy B. Le, Giuseppe Scanniello, and Jianling Sun. Inferring Links between Concerns and Methods with Multi-Abstraction Vector Space Model. In *Proceedings of the 32nd IEEE International Conference on Software Maintenance and Evolution*, pages 110–121, Oct 2016.

Ripon Saha and Milos Gligoric. Selective Bisection Debugging. In *Proceedings of the 20th International Conference on Fundamental Approaches to Software Engineering*, pages 60–77, New York, NY, USA, 2017. Springer-Verlag New York, Inc.