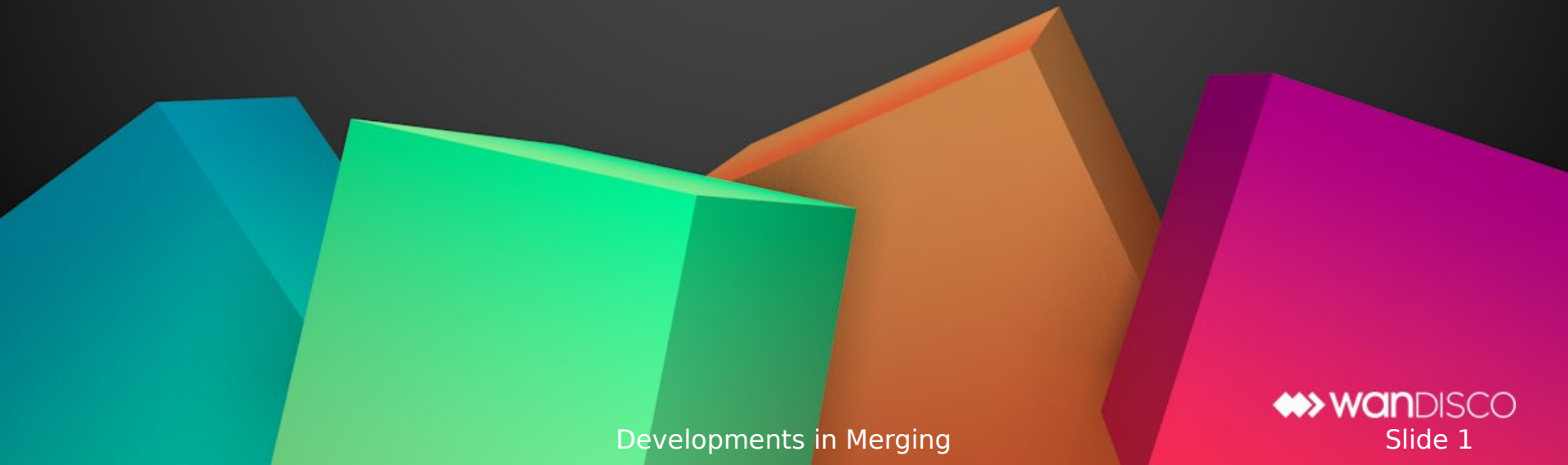
SUBVERSION & GIT

LIVE

Developments in Merging

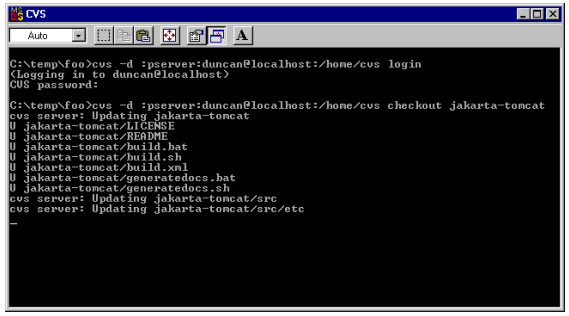
Julian Foad
WANdisco

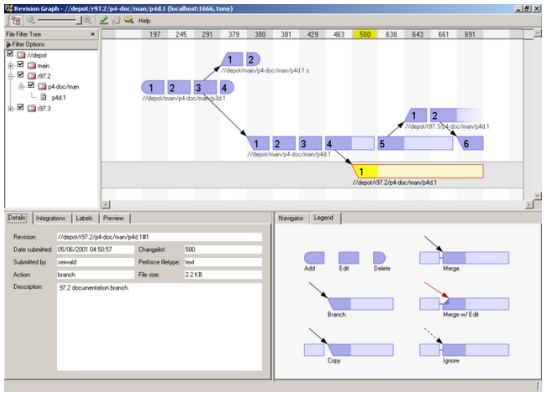


Topics

- How I got interested
- Models, intent and assumptions
- Automatic reintegrate
- Starting on move tracking

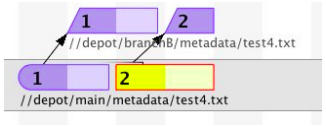
sepura
RELIABLE • ROBUST • SECURE

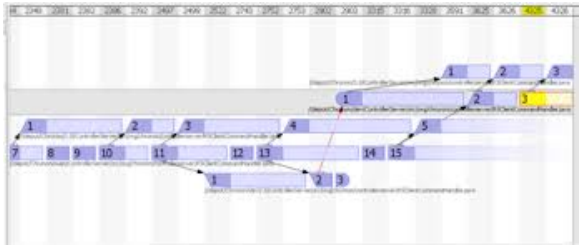


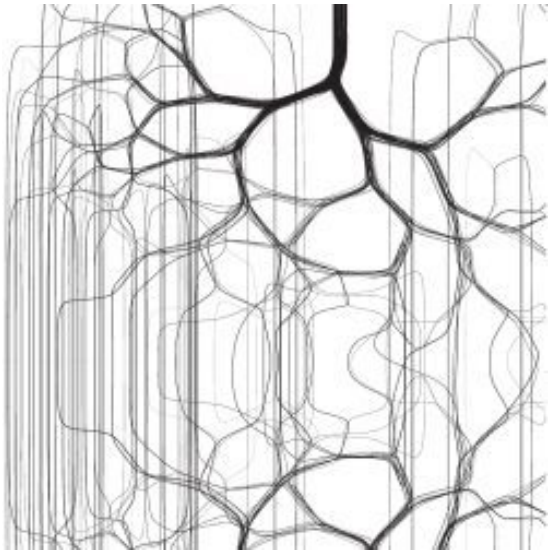


PERFORCE

Version everything.


1
2
//depot/branchB/metadata/test4.txt
1
2
//depot/main/metadata/test4.txt



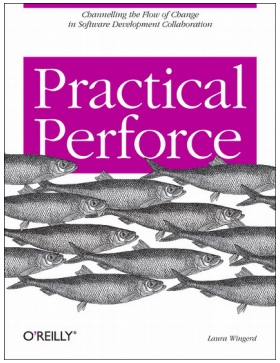


Convergence vs. Divergence

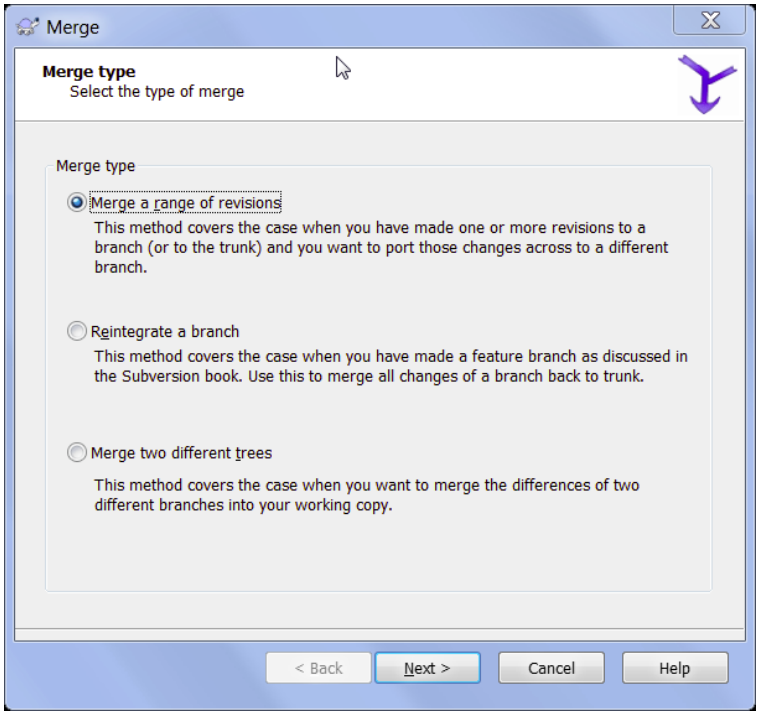
Purposeful Merging with Perforce

Laura Wingerd • Perforce Software • www.perforce.com

Perforce User Conference EUROPEAN LONDON

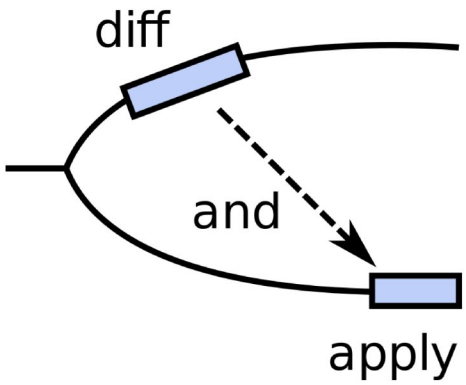


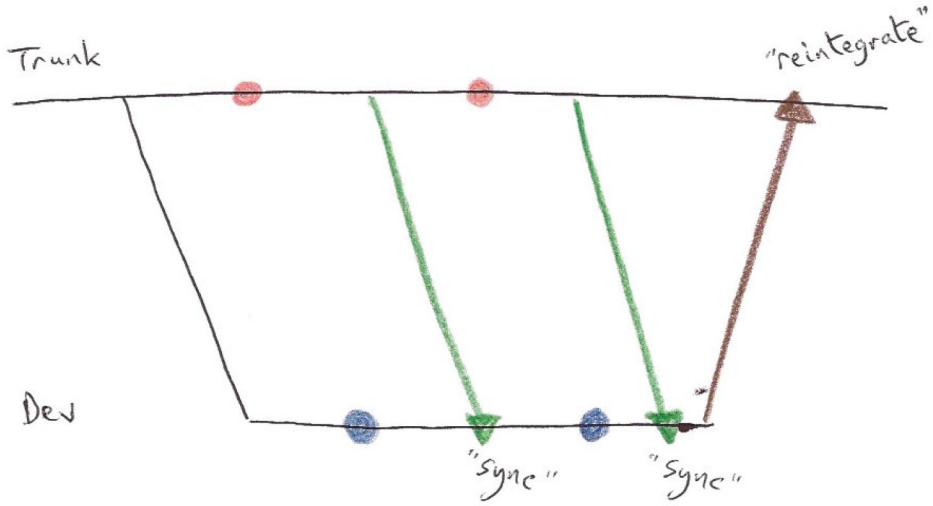
TortoiseSVN

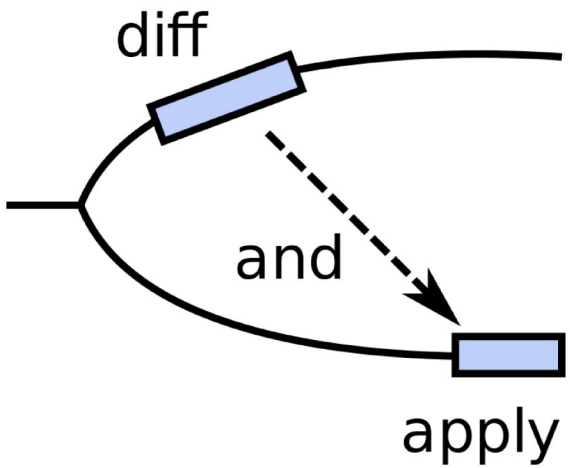


Who cares?
Just make it DTRT!

Models, Intent, Assumptions



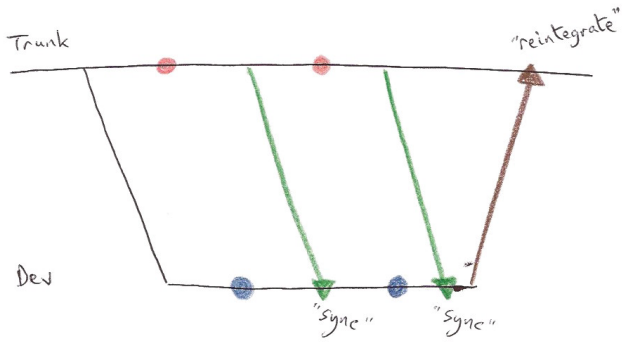




Explicit
– source -r X:Y

Assumptions
– tree shape

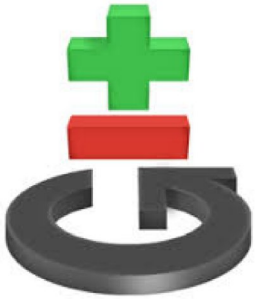
Model
– related branch
– no renames

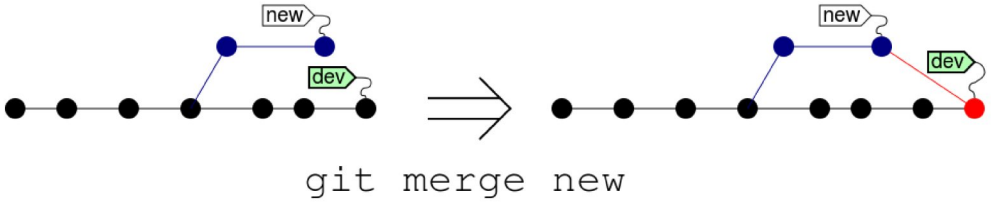


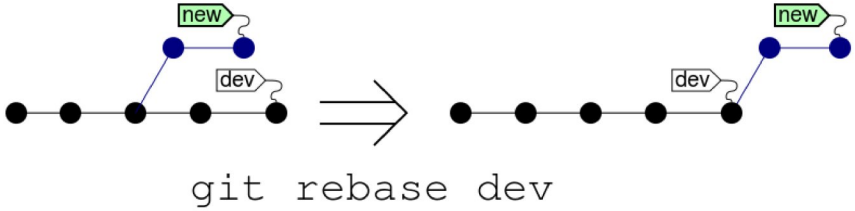
Model
– Dev-branch scenario
– All changes → main

Assumptions
– sync many times
– reintegrate once
– record-only = “don't want”

Git's Model





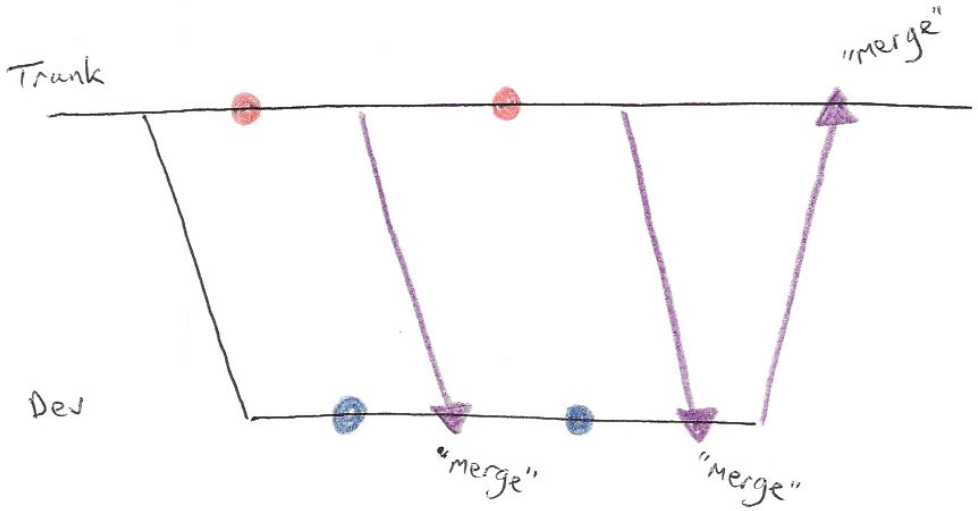


Images: David Drysdale <lurklurk.org>

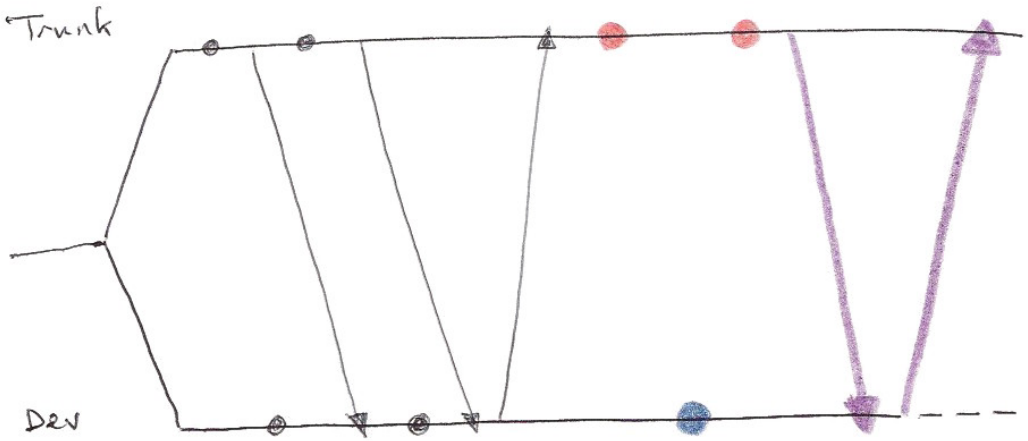
History of Subversion Merge

1.0 Diff & Apply
1.5 Merge Tracking
1.5 Reintegrate
1.8 Automatic reintegrate

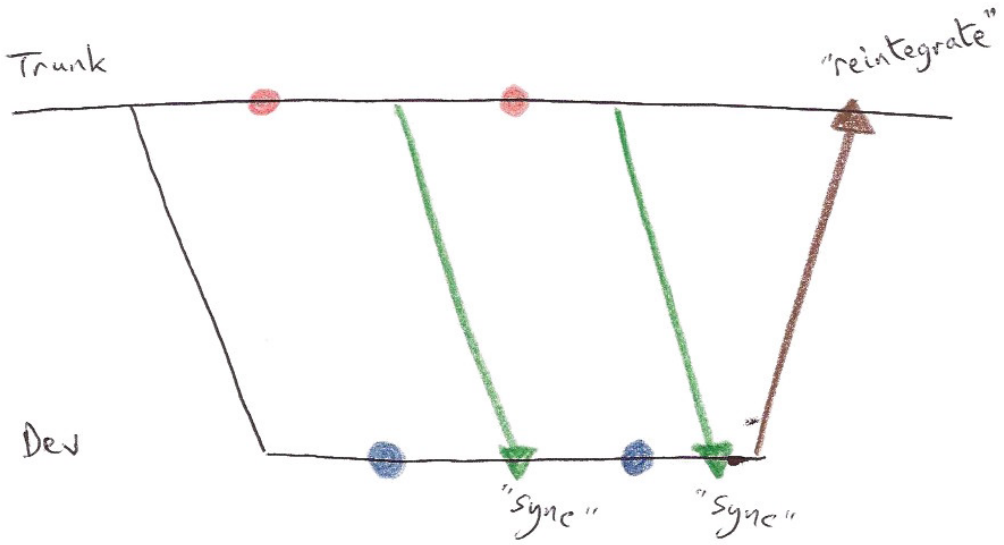
Automatic Reintegrate – Goal (1)



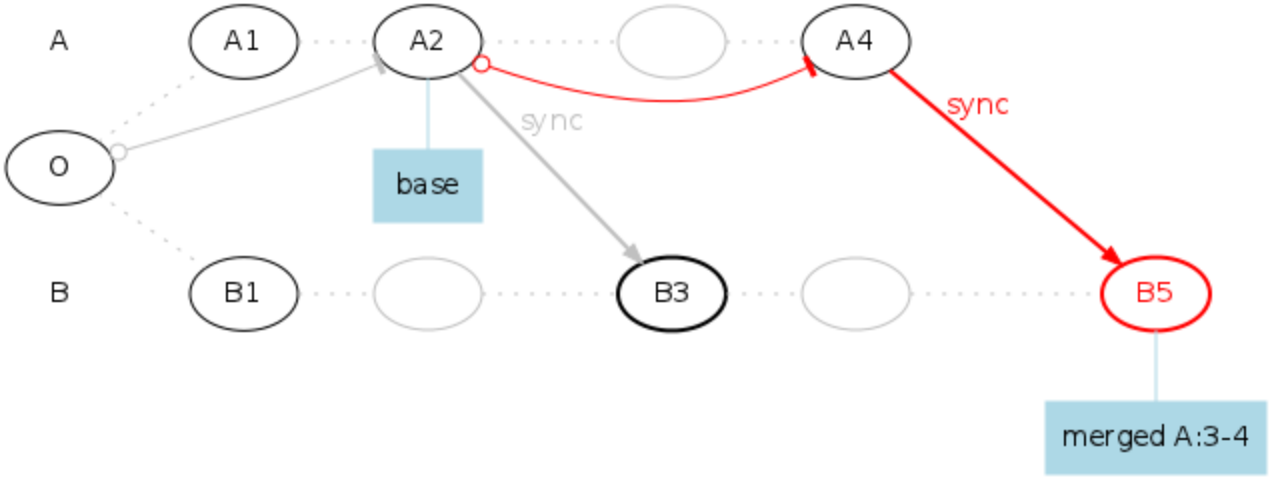
Automatic Reintegrate – Goal (2)



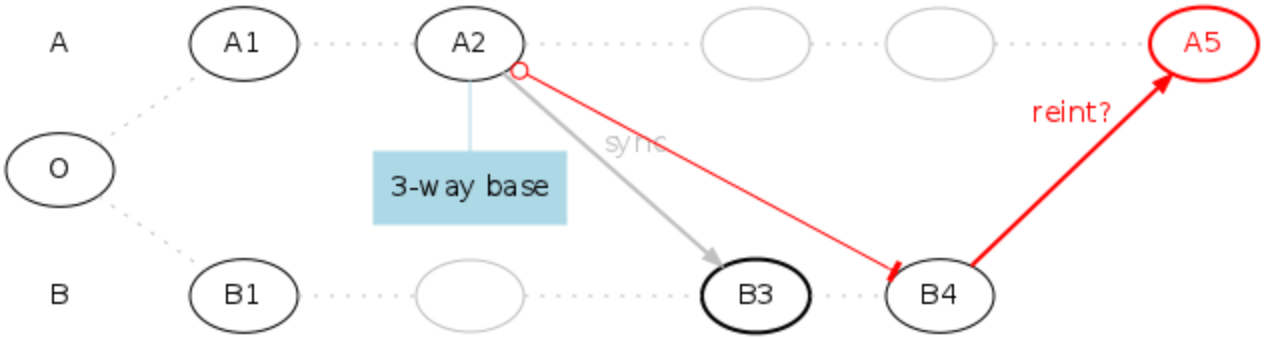
Merging in Svn 1.7



Sync vs. Reintegrate (1)



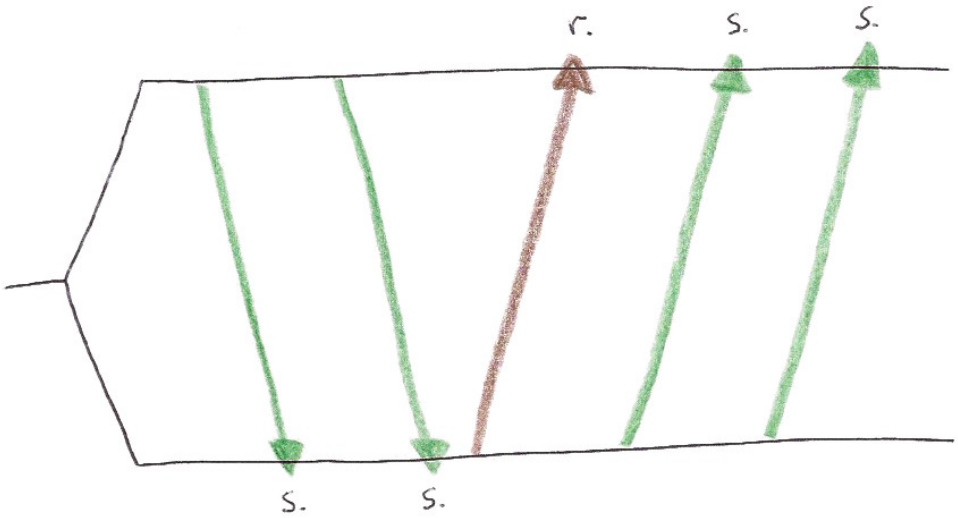
Sync vs. Reintegrate (2)



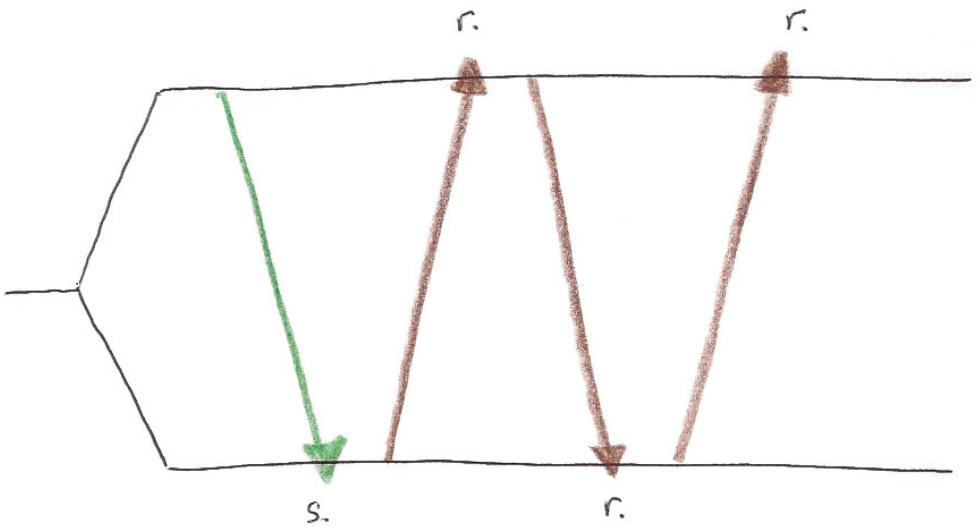
Sync vs. Reintegrate (3)

	sync	reintegrate
base node	on source branch	on target branch
skip cherry-picked revs?	yes	no
fill in partly-merged subtrees?	yes	no
handle local mods in the WC?	yes	no

Same Direction = Sync



Change Direction = Reintegrate



How does it work?

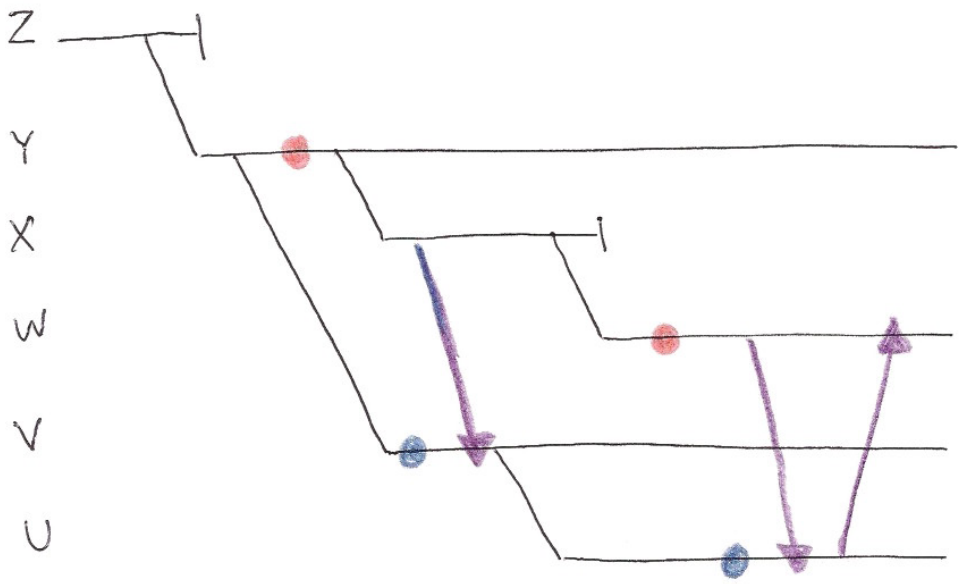
Find the best base

- Find the latest rev of A synced to B and of B synced to A.

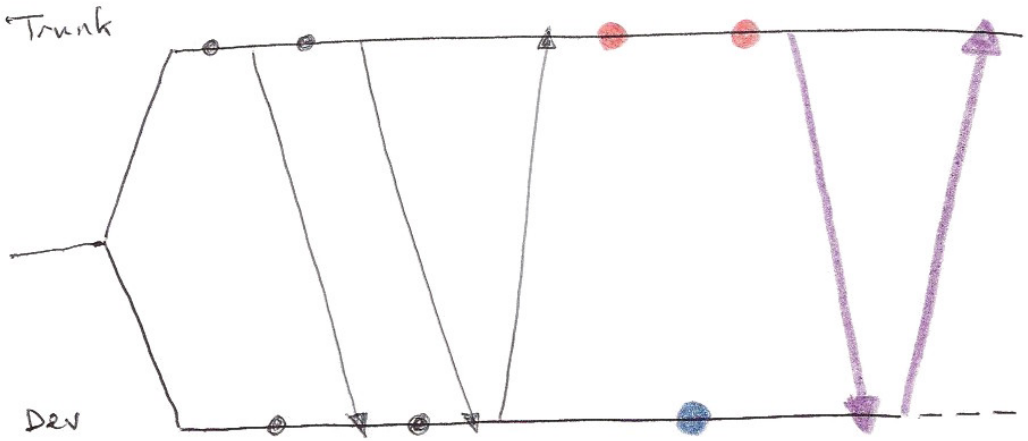
Choose the more recent base.

- Select which code to run
  - Run sync if base on source
  - Run reintegrate if base on target

Confusing



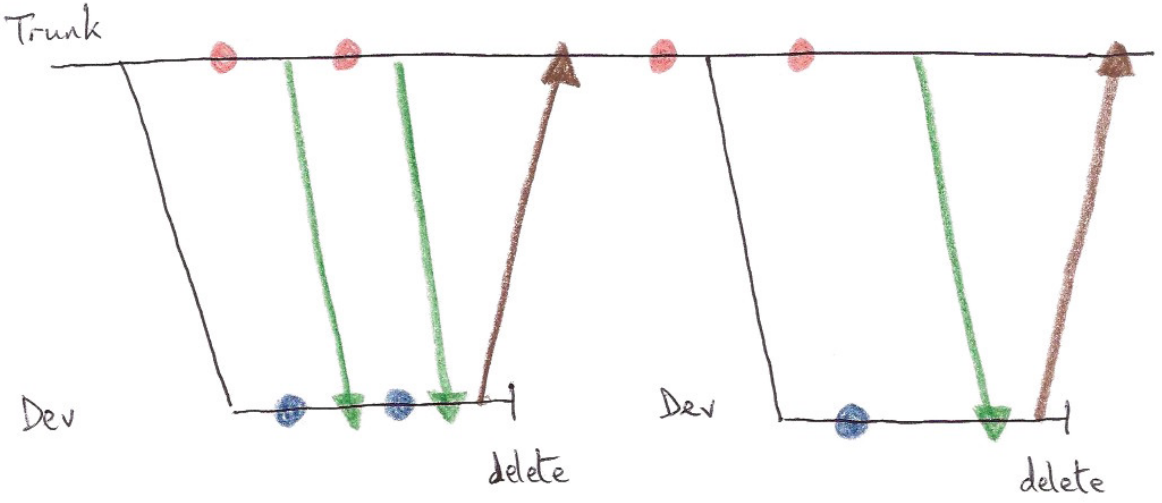
Continue Work on Branch



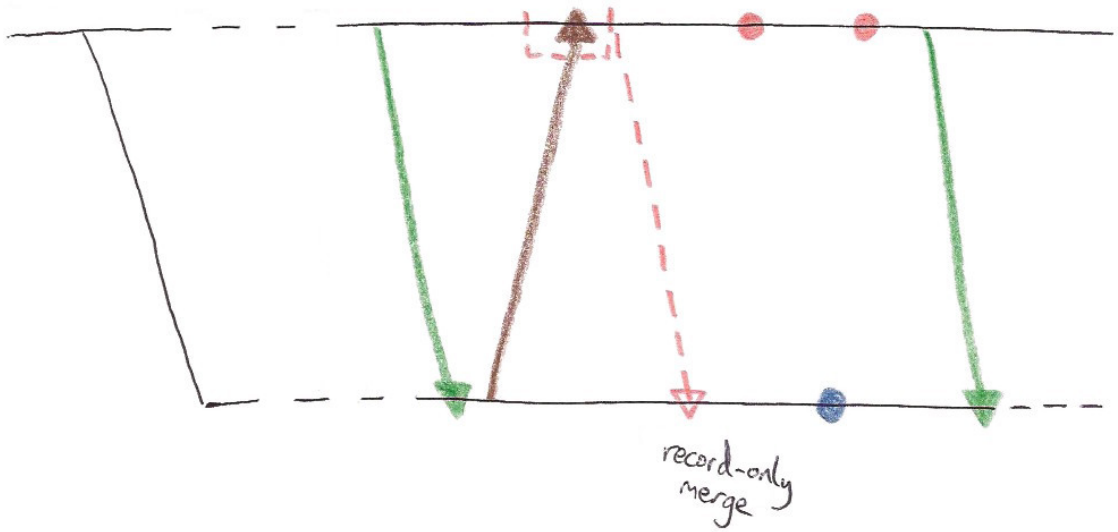
Continue – Options in 1.7

- Delete
- Keep Alive

Continue – Delete and Re-branch



Continue – Keep-Alive



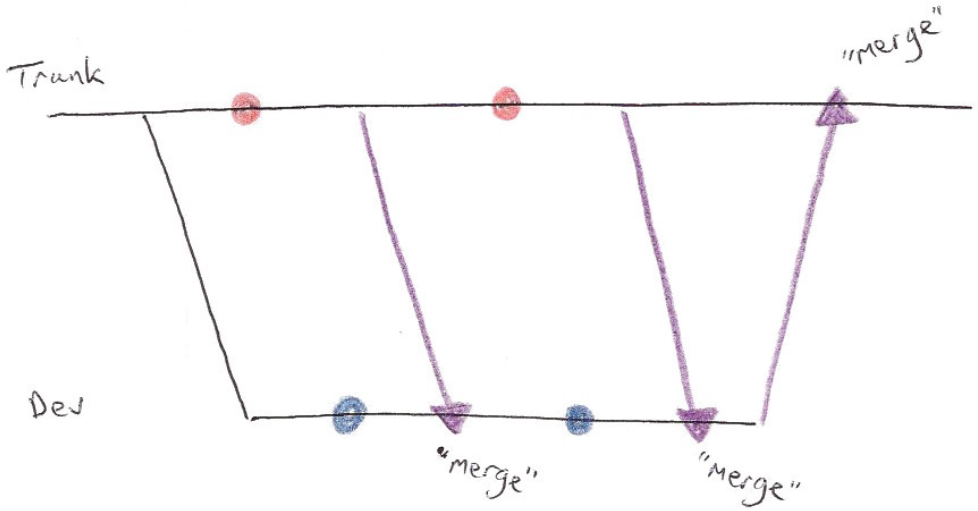
Limitations

not yet symmetric inside
- limitations NOT symmetric
- results are symmetric

reintegrate (change-direction) merges
- no cherry-picked revisions
- no subtree-specic mergeinfo
- no local mods in WC
- no sparse WC

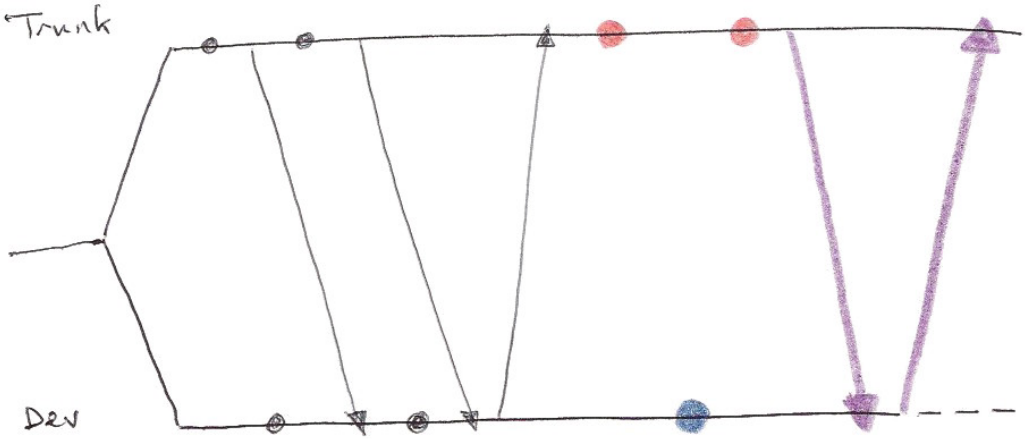
in line with usage & best practice

Automatic Reintegrate



Automatic Reintegrate

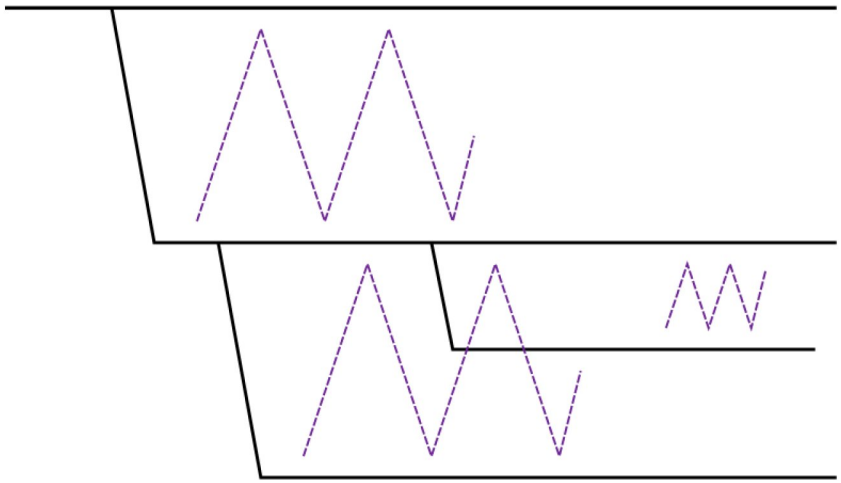
Just make it DTRT!

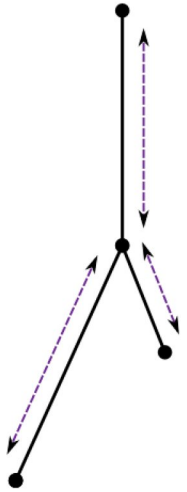


Summary so far

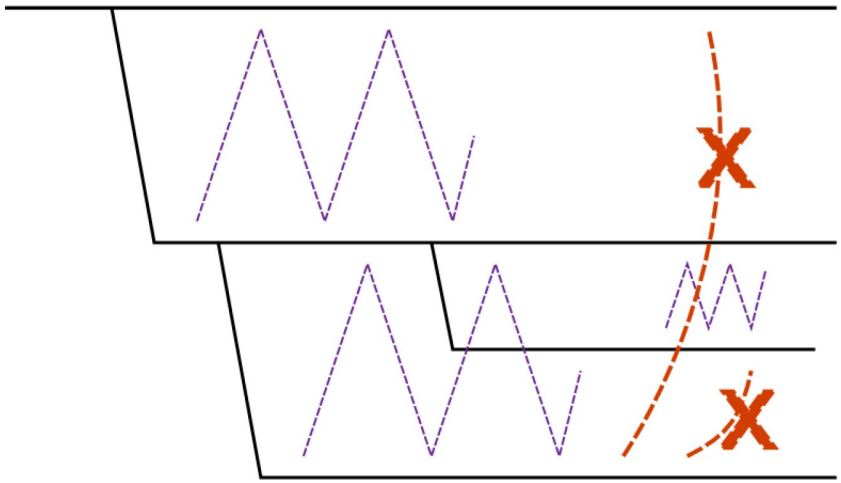
- Models & assumptions
- Automatic reintegrate

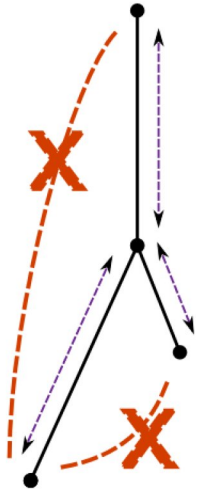
Merge between Two Branches





Merge between Two Branches





Move Tracking

- incoming move, local mods
  - obvious
- incoming move, local move of parent
  - obvious... to us

Unix 'mv' command

- mv foo bar
- mv A/B/C X/Y/Z

Take file/dir from path 1, put at path 2

- For merging, don't think of *path*
  - but name in parent dir.
  - Those need not conflict.
- mv A/B X/B || mv A/B/foo A/B/bar
  - (name 'B' in dir A) → (same in X)
  - (name 'foo' in dir B) → ('bar' in same)
  - together: A/B/foo → X/B/bar

- WC & 'update' work on paths.
  - update brings incoming rename A→X
  - client: 'I have A@4 and A/foo@5'
  - server: 'delete A; add X; add X/foo'
- Merge, Update, Commit
  - all need the same awareness of moves

- Tricky cases include
  - mv A X; mkdir A; mv X A/X; commit
  - Remember & transmit whole sequence?
  - Condense to sequence with one mv?
  - Represent the result not as a sequence?

Summary

- Models & assumptions
- Automatic reintegrate
- Move tracking