Summary

1 Preparation 3

1.1 Batch files created: 3

1.1.1 PushToRep 3

1.1.2 GitInit 3

1.2 Initial situation 3

2 Scenarios 3

2.1 Scenario 1 3

2.2 Scenario 4 5

2.3 Scenario 5 6

3 Further Scenarios (not done yet) 7

3.1 Scenario 6: Handling two branches (1.x-master and master) 7

4 Analysis 8

4.1 Scenario 1 8

4.1.1 Before first push 8

4.1.2 Before doing merges 8

4.1.3 Após merge de aakoch 10

4.1.4 Post merge of adam’s commit and push to origin 11

4.2 Scenario 4 12

4.2.1 After scottjehl’s first commit 12

4.2.2 After scottjehl’s second commit 12

4.2.3 After scottjehl’s second merge 13

4.3 Scenario 5 14

4.3.1 Fork of a repository 14

5 Not used scenarios 14

5.1 Scenario 2 14

5.2 Scenario 3 15

5.3 Scenario 7 15

5.4 Scenario 8 15

5.5 Scenario 9 15

5.6 Scenario 10 16

# Preparation

## Batch files created:

### PushToRep

Sintax: pushToRep <repo> <commit> <branch>

Pushes history up to specified commit, to the specified repository, at the specified branch

### GitInit

Sintax: gitinit <repo> <dir>

Creates a clone of repo in the specified directory, allowing the clone to receive pushes

## Initial situation

1. Create bare clone of JQuery containing the entire history
   1. JQuery was chosen due to its size, which allows both watching the commit history in DyeVC and having moments of high parallelism.

git clone --bare "https://github.com/jquery/jquery.git" "F:\evaluation\repos\jquery-home"

1. Create simulation of central repository

mkdir F:\evaluation\repos\central-repo

cd F:\evaluation\repos\central-repo

git init --bare

1. Push first commit to central repository

cd F:\evaluation\repos\jquery-home

git push F:\evaluation\repos\central-repo 8a4a1edf047f2c272f663866eb7b5fcd644d65b3:refs/heads/master

1. Create test clone

Git clone central-repo teste

1. Configure test clone to allow receiving pushes from other clones

git config receive.denyCurrentBranch ignore

1. Push a commit from jquery-home to test clone (example)

git push "F:\evaluation\repos\teste" ffb1867a4364ea65e60dad3469e8c8eb420ebcac:refs/heads/master

1. Pushing another commit (example)

git push "F:\evaluation\repos\teste" 0645b71ee6139c19c2c5a488f16f50dc1c31e9ac:refs/heads/master

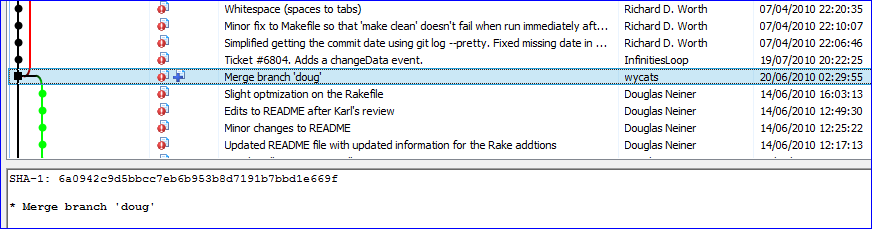
# Scenarios

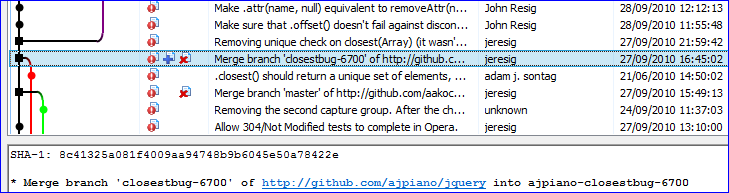
## Scenario 1

Motivation: To see how DyeVC handles topologies and how it can help to depict related partners and pending commits.

First commit: 8c41325a081f4009aa94748b9b6045e50a78422e

Last commit: 8c41325a081f4009aa94748b9b6045e50a78422e





1. pushtoRep central-repo 6a0942c9d5bbcc7eb6b953b8d7191b7bbd1e669f master
2. gitinit central-repo aakoch
3. gitinit central-repo adam
4. gitinit central-repo jeresig
5. pushToRep aakoch a088751a1b2c5761dab8de9d7da8602defb45b11 master
6. 🡺 Vide analysis “Before first push” in section 4.1.1.
7. aakoch: git push origin

Backup 1

1. pushToRep adam a2bd8a53f3a750606abea9bbb6ee2302437d42f3 closestbug-6700
2. pushToRep jeresig ea6a4813b7d996f6f7af0b61a5f1bf4ab80b291d master

🡺 Vide analysis “Before doing merges” in section 4.1.2.

1. pushTorep jeresig e63fa8beb8e285fe19fc0a1557045b80e3c63c66 aakoch-master (merge de aakoch)
2. jeresig: git merge aakoch-master
3. jeresig: git push origin
4. adam starts to track branch closestbug-6700 (git push -u origin closestbug-6700)
5. adam: git fetch origin

🡺 Vide analysis “Após merge de aakoch” aakoch in section 4.1.3.

1. jeresig: git fetch --all (bring adam’s commit to be merged)
2. pushTorep jeresig 8c41325a081f4009aa94748b9b6045e50a78422e ajpiano-closestbug-6700 (merge de adam)
3. jeresig: git checkout master
4. jeresig: git merge ajpiano-closestbug-6700
5. jeresig: git push origin

🡺 Vide analysis “Post merge of adam’s commit and push to origin” in section 4.1.4

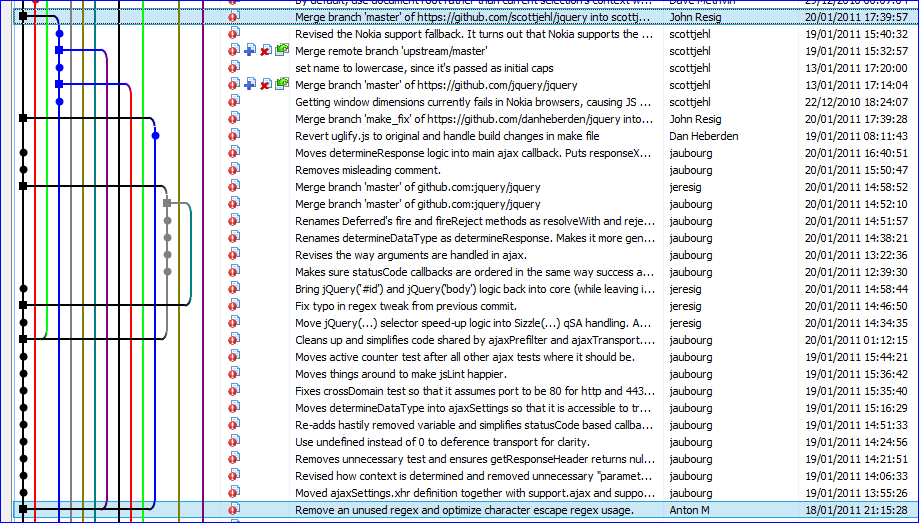
Backup 2

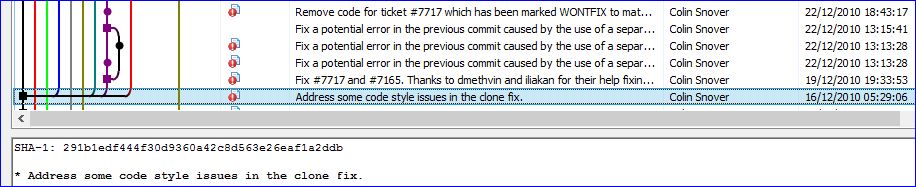
## Scenario 4

Motivation: To see merges from older commits at master

First Commit: 291b1edf444f30d9360a42c8d563e26eaf1a2ddb

Last commit: c97b6ee36a1402b6f4bb3f6893b687eb294c41df





1. gitinit central-repo scottjehl
2. pushToRep.bat scottjehl 291b1edf444f30d9360a42c8d563e26eaf1a2ddb master
3. scottjehl: git push origin
4. pushToRep.bat scottjehl 18fa1fd9da12fd6a259b422f91b663d9fbdb181e master (22/12/2010)
5. gitinit central-repo jaubourg
6. pushToRep.bat jaubourg a8fa5f2ec1030bceb9a65d0237f0c92ae4e014dd master
7. jaubourg: git push origin
8. jeresig: git pull origin master

Backup 3 (after scottjehl’s commit, before 1st merge)

🡺 Vide analysis After scottjehl’s commit in section 4.2.1

1. scottjehl tries to push changes at 13/1/2011, has newer commits
2. scottjehl: git fetch
3. pushToRep.bat scottjehl 610ab137da38106f8c464f099a304ae3795c2231 master (merges from origin)
4. scottjehl: git fetch
5. pushToRep.bat scottjehl 73d060b522ccf72847e67f56fea0e9b129ff273e master (at 13/1/2011)
6. pushToRep.bat jeresig 265cf0efa7ab3296b3fc4917b863d7b09e3d8bb4 master (18/1/2011)
7. jeresig: git push
8. 🡺 Vide analysis After scottjehl’s second commit in section 4.2.2

Backup 4

1. scottjehl tries to push changes at 19/1/2011, has newer commits and performs a new merge
2. scottjehl: git fetch
3. pushtoRep scottjehl cb1f7eeac5073748ae6200f2f960fb330ec966a8 master (19/1/2011)
4. scottjehl: git push
5. 🡺 Vide analysis After scottjehl’s second commit in section 4.2.2

Backup 5

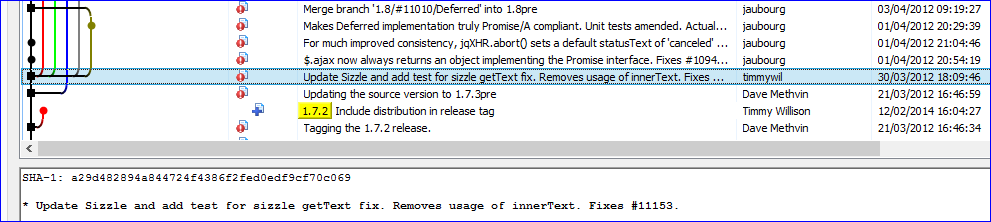
## Scenario 5

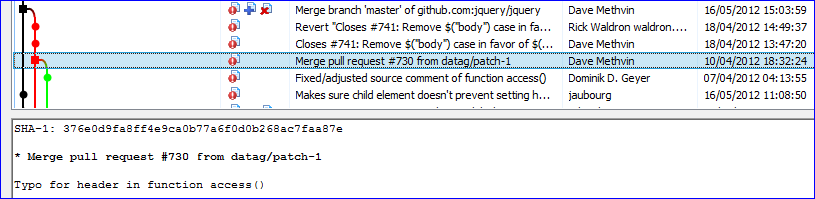
Motivation: To see how DyeVC handles pull request situations

First Commit: a29d482894a844724f4386f2fed0edf9cf70c069

Pull request merge at: 376e0d9fa8ff4e9ca0b77a6f0d0b268ac7faa87e

Last Commit: 9efe4d03df7ce417792152660a0bed36b261676e





1. pushToRep.bat jeresig a29d482894a844724f4386f2fed0edf9cf70c069 master (30/3/2012)

Simulating fork of jquery, named patch-1 and creating a clone for it, named dominik\_geyer

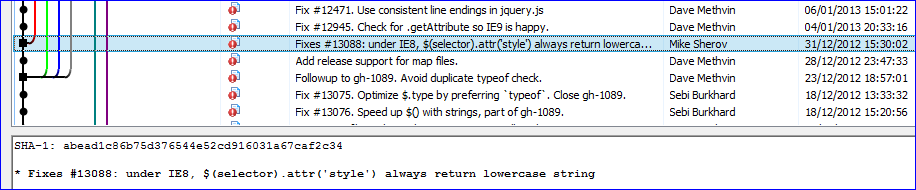
1. mkdir F:\evaluation\repos\jquery\_patch-1
2. cd F:\evaluation\repos\jquery\_patch-1
3. git init --bare
4. cd F:\evaluation\repos\jquery-home
5. git push F:\evaluation\repos\jquery\_patch-1 a29d482894a844724f4386f2fed0edf9cf70c069:refs/heads/master
6. gitinit jquery\_patch-1 dominik\_geyer
7. pushToRep.bat dominik\_geyer fccfeefb1036ac70f565c2f54b55a14fca528d57 master (7/4/2012)
8. dominik\_geyer: git push
9. dominik\_geyer: git remote add github F:/evaluation/repos/central-repo
10. Dominik\_geyer: git fetch github
11. 🡺 Vide analysis “Fork of a repository” in section 4.3.1

Backup 6

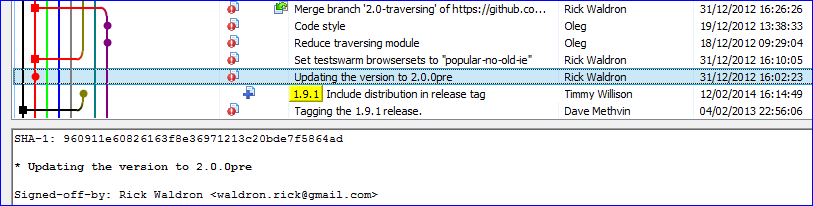
# Further Scenarios (not done yet)

## Scenario 6: Handling two branches (1.x-master and master)

First commit: abead1c86b75d376544e52cd916031a67caf2c34

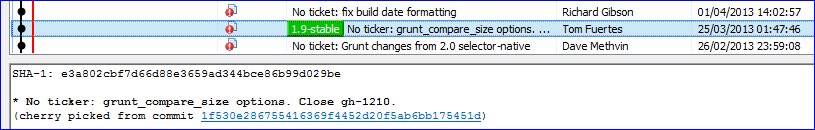


First commit as new master: 960911e60826163f8e36971213c20bde7f5864ad



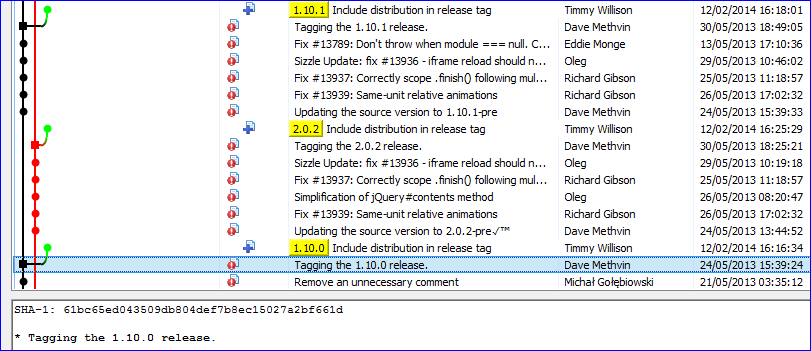
Tagging 1.9-stable, which follows as 1.x-master

1. e3a802cbf7d66d88e3659ad344bce86b99d029be



Próximo cria tags pra 1.10.0, 2.0.2 e 1.10.1

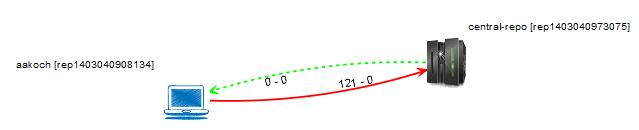
1. 61bc65ed043509db804def7b8ec15027a2bf661d

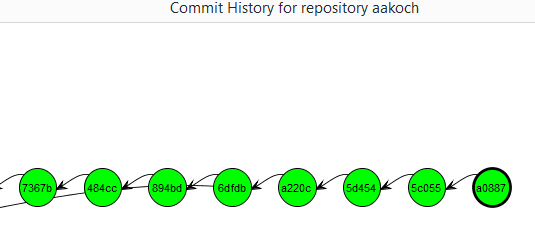


# Analysis

## Scenario 1

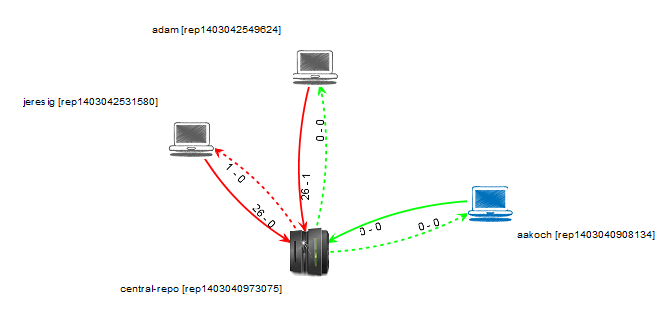
### Before first push





### Before doing merges

Topology





🡺 Adam mostra que está behind 121, mas na topologia mostra que está também ahead 1 (commit em ramo não rastreado - closestbug-6700)



🡺 jeresig está behind 1 pois nesse momento aakoch já fez push e está ahead 26 pois não fez push de seus commits

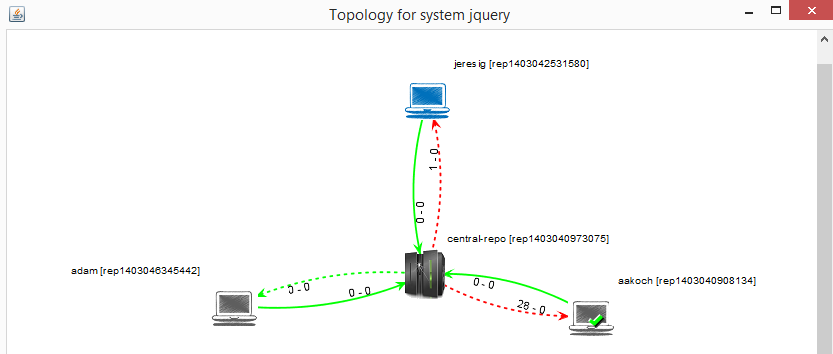


aakoch commit history: mostrando commit apontado pelo master e os 26 commits de jeresig que ele não pode receber, pois jeresig ainda não deu push.

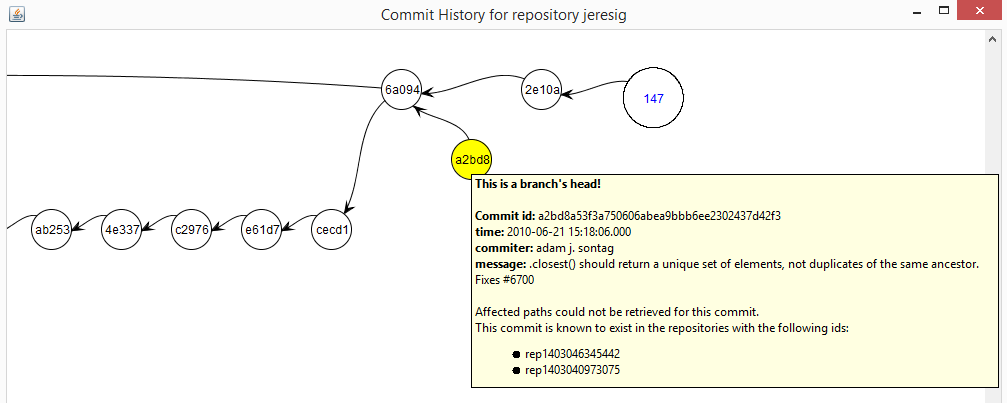


jeresig, mostrando commit não rastreado de adam, assim como commit de aakoch disponível para pull, além de 26 commits feitos localmente e os quais não foram dados push.

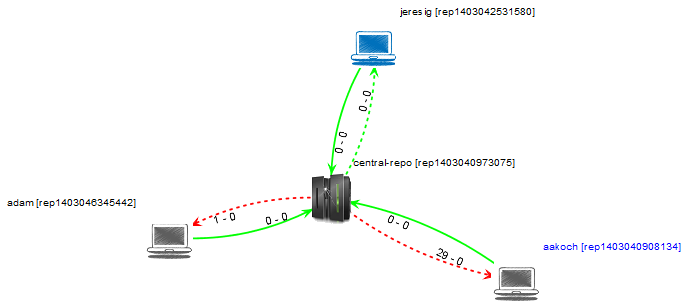
### Post aakoch merging



1 commit para baixar em jeresig, que é o commit feito por adam (agora num ramo rastreado). Vide history a seguir

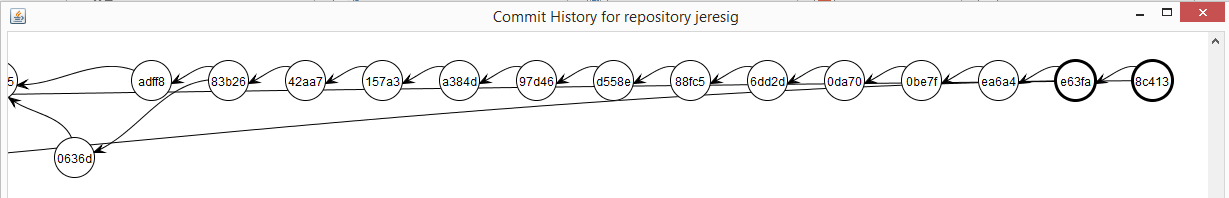


### Post merge of adam’s commit and push to origin



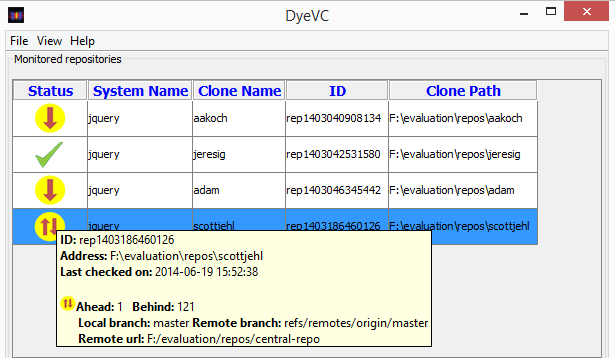
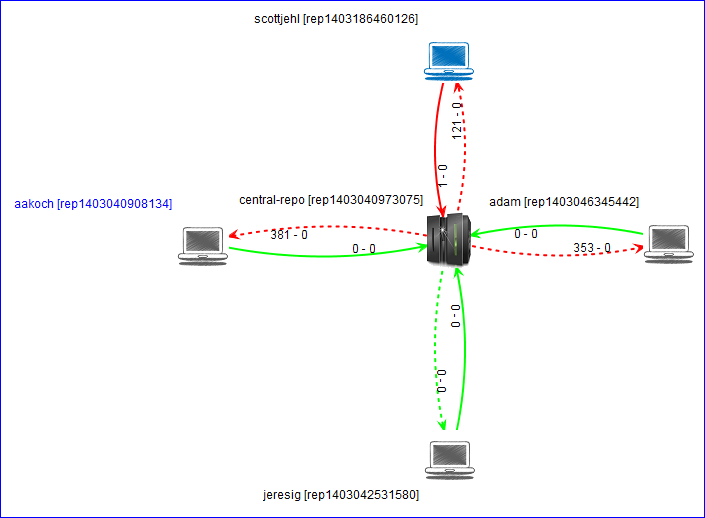
jeresig is in sync with remote

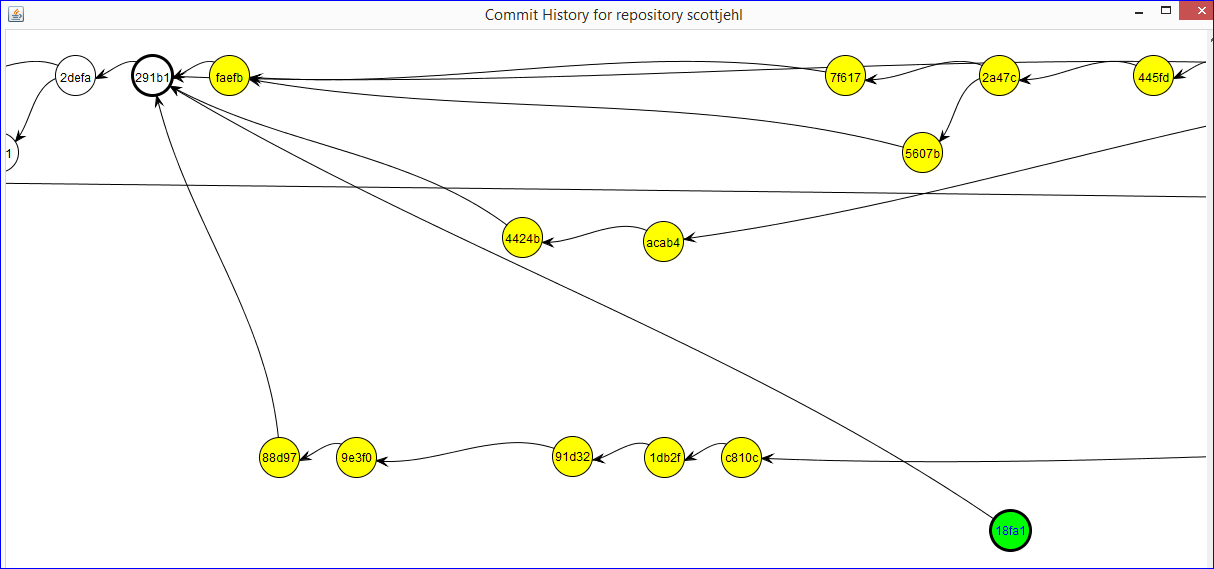
jeresig’s commit history is all white (in sync)



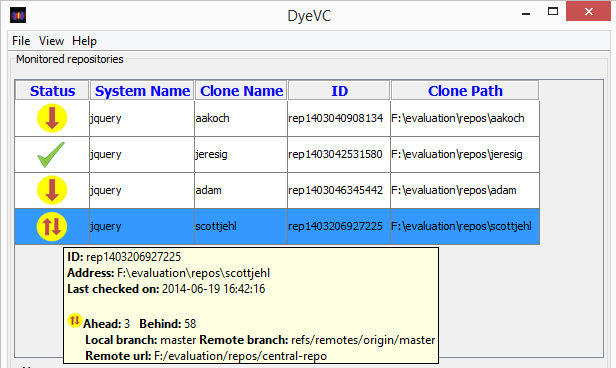
## Scenario 4

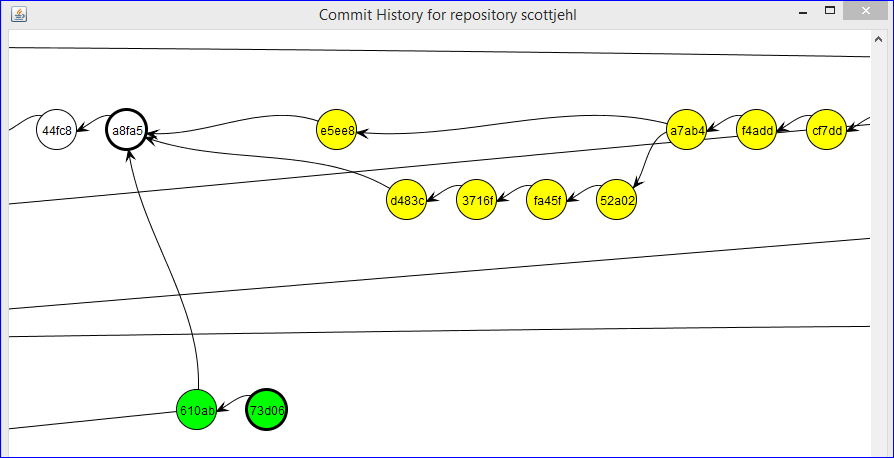
### After scottjehl’s first commit



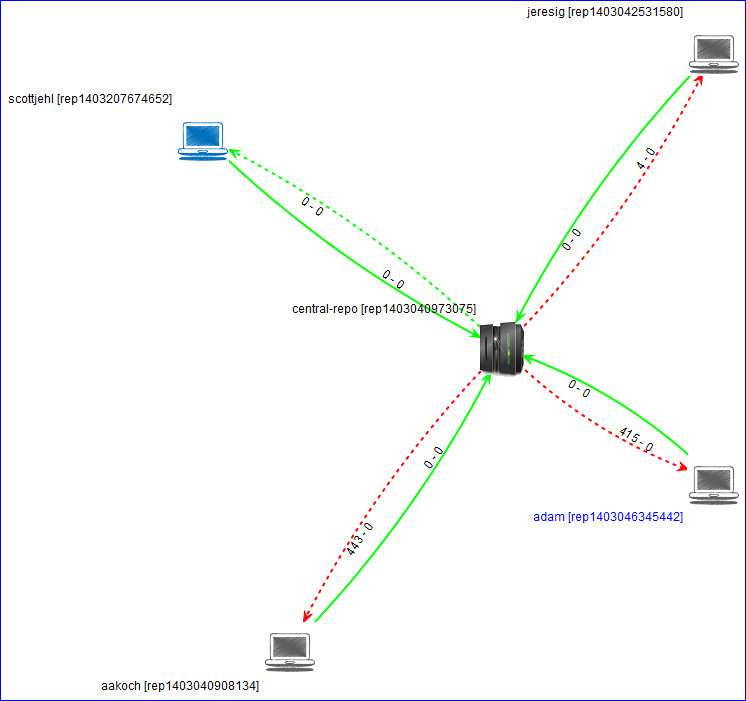
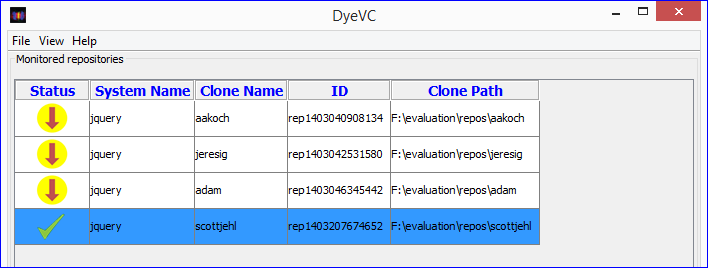


### After scottjehl’s second commit



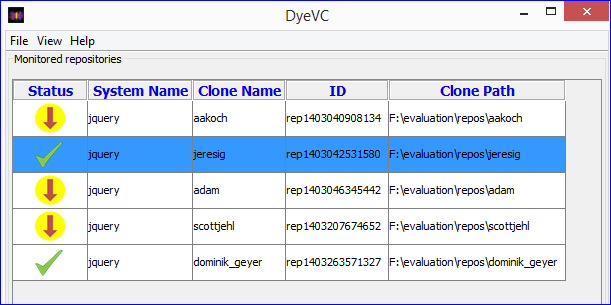
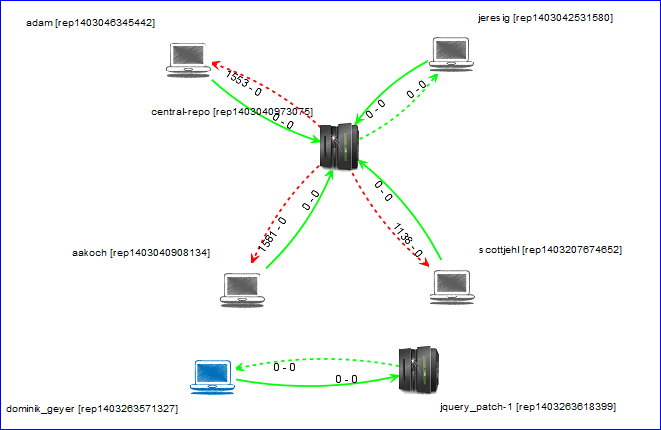


### After scottjehl’s second merge

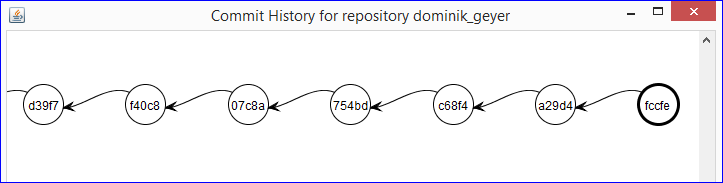


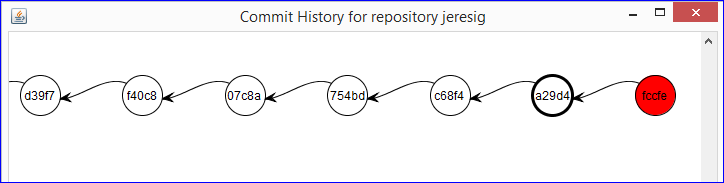
## Scenario 5

### Fork of a repository



Fork is represented as a new server, provided that the forked clone is monitored using the same project name. Both jeresig and Dominik\_geyer are in sync with their servers, but commit history shows that there is a commit in the topology that jeresig doesn’t have, and can’t pull from any partner (a pull request procedure must be in place to send this commit from jquery\_patch-1 server to central-repo server.





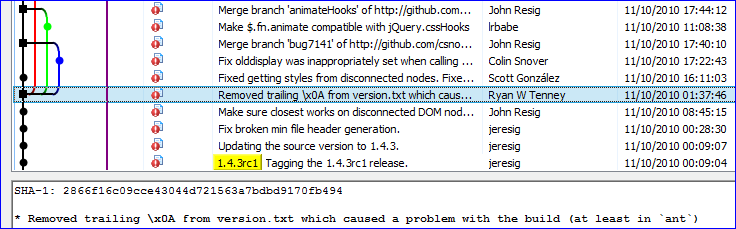
# Not used scenarios

## Scenario 2

1. 2866f16c09cce43044d721563a7bdbd9170fb494

**Reason:** Similar to scenario 2.1

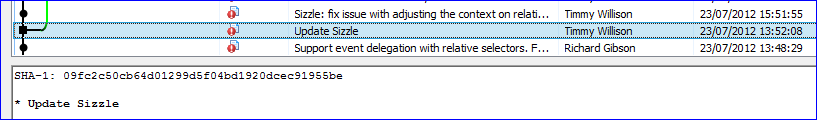
* 1. John Resig : d23f63b13ea47ac3e91906e5ac1e2503400cd903 (master)
  2. lrbabe : aa9e4db1ab5cd6f514de616070829d64d69a3428 (animateHooks)
  3. Colin Snover : 5646a4feee87b086fa4752373422622d54a8430d (bug7141)



## Scenario 3

1. 09fc2c50cb64d01299d5f04bd1920dcec91955be

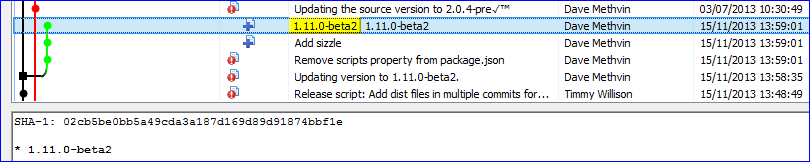
**Reason:** Similar to scenario 2.1



## Scenario 7

tag 1.11.0-beta2

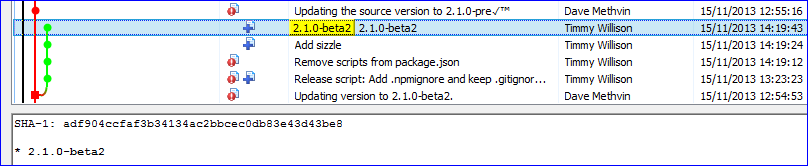
1. 02cb5be0bb5a49cda3a187d169d89d91874bbf1e



## Scenario 8

tag 2.1.0-beta2

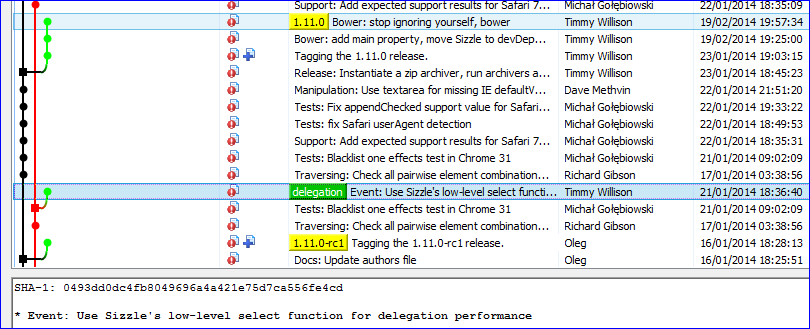
1. adf904ccfaf3b34134ac2bbcec0db83e43d43be8



## Scenario 9

Ramo delegation

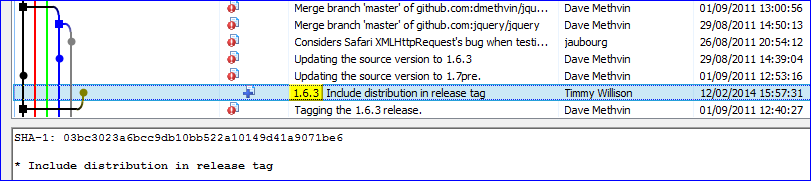
1. 0493dd0dc4fb8049696a4a421e75d7ca556fe4cd



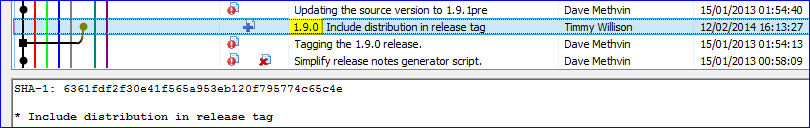
## Scenario 10

Algumas tags:

1. 03bc3023a6bcc9db10bb522a10149d41a9071be6



1. 6361fdf2f30e41f565a953eb120f795774c65c4e



1. d71f6a53927ad02d728503385d15539b73d21ac8

