

Git Best Practices

Do you really use Git efficiently?

Ilya Rokhkin
2019

Git – meaning?

What does the word Git means?

GIT Overview



Article

Talk

Read

Edit

Git (slang)

From Wikipedia, the free encyclopedia

Git is a term of insult with origins in English denoting an unpleasant, silly, incompetent,

https://en.wikipedia.org/wiki/Git



T



DevIS



LOG



Duty



R



W



icinga



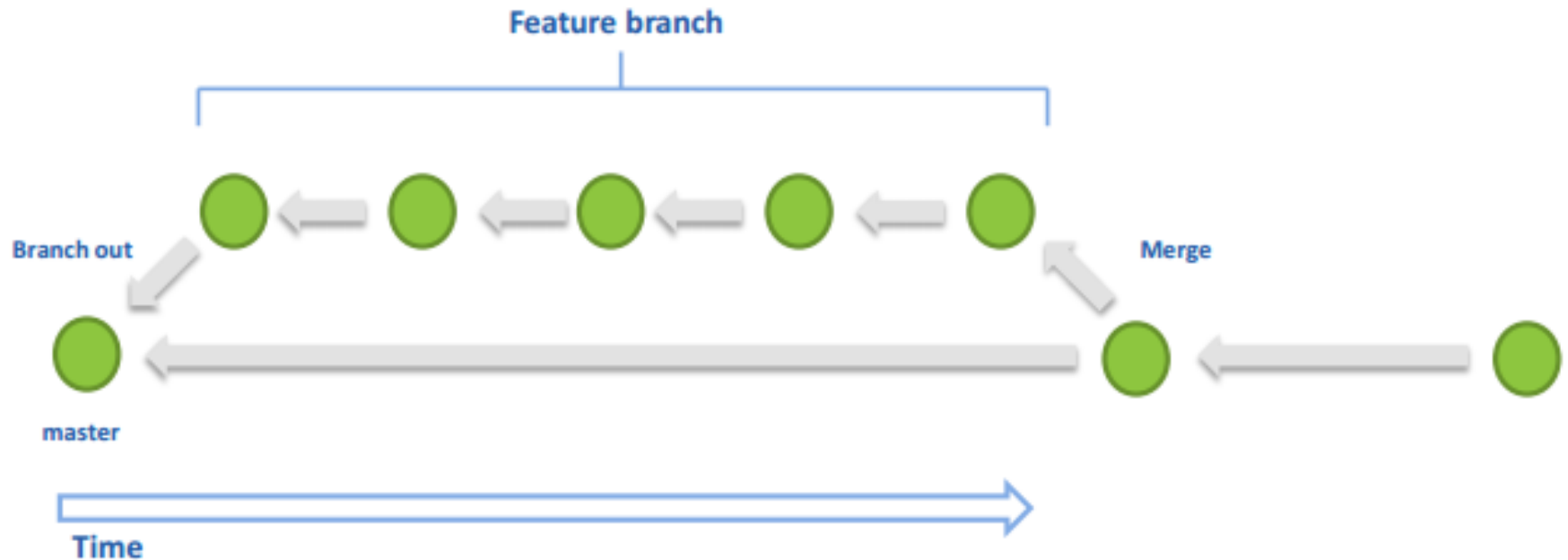
Grok

Naming [\[edit \]](#)

Torvalds quipped about the name *git* (which means *unpleasant person* in [British English](#) slang): "I'm an egotistical bastard, and I name all my projects after myself. First '[Linux](#)', now '[git](#)'."^{[23][24]}
The [man page](#) describes Git as "the stupid content tracker".^[25]

Best practices – local feature branches (topics)

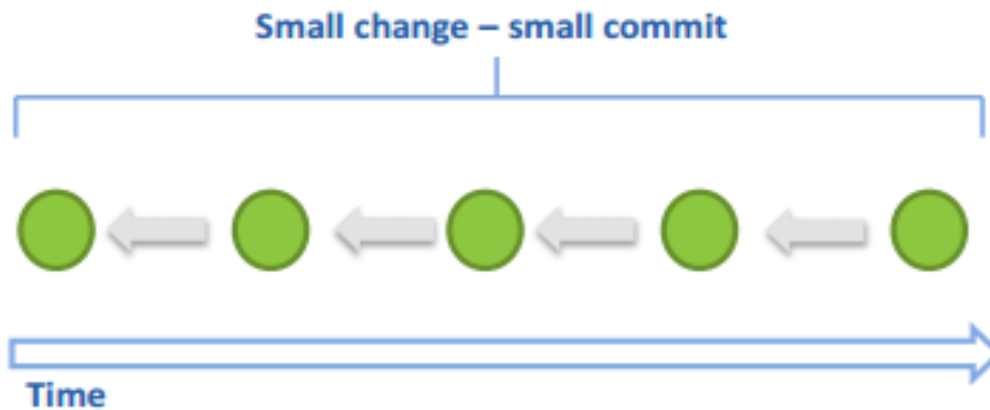
Work on feature branches locally



Demo: create feature branch, checkout, commit

Best practices – commit

Keep changes small wherever possible and commit frequently



Demo: vim, change, commit, auto-commit, git log -c, git reset

Best practices – squashing

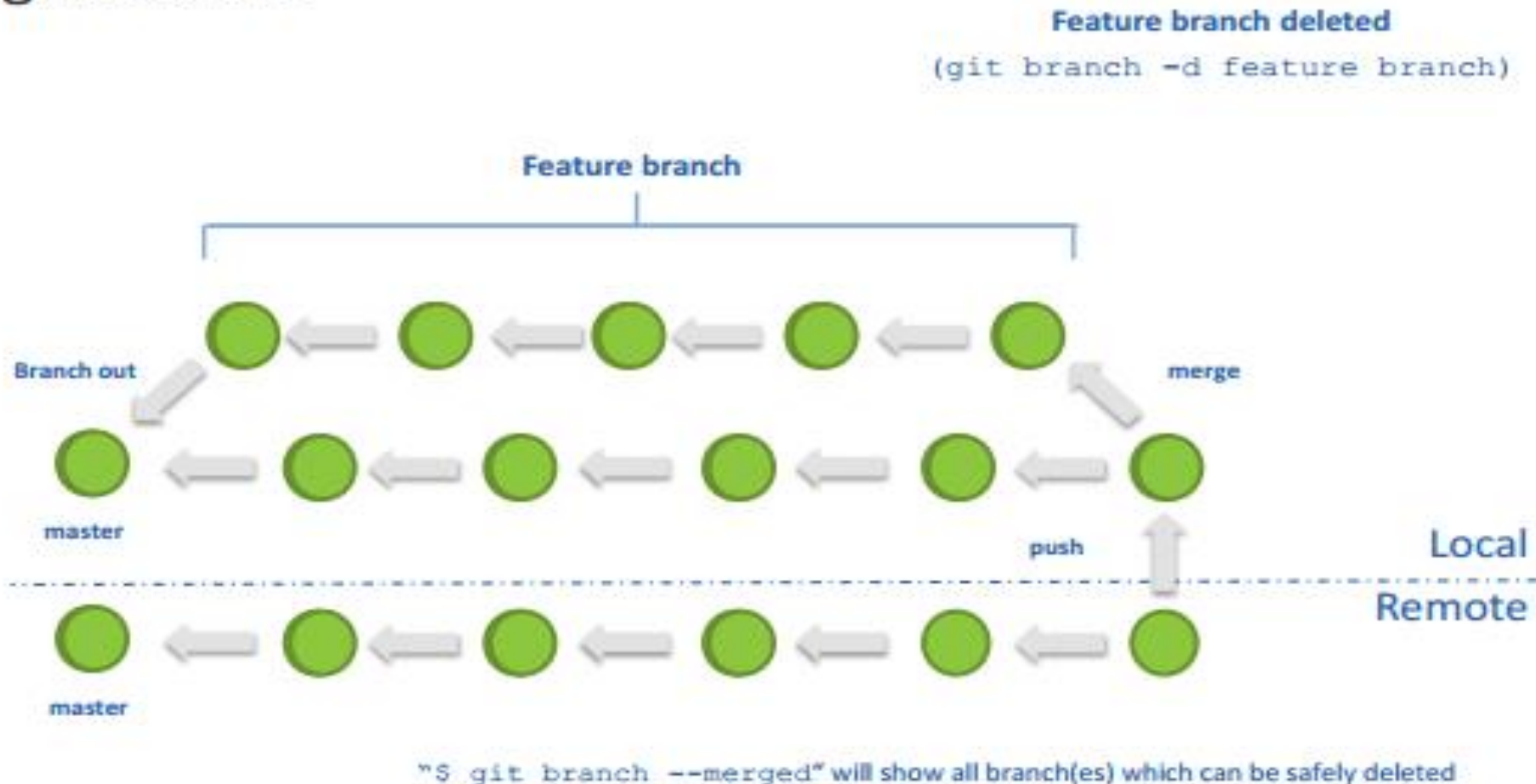
Before pushing, squash related changes together to make for better understanding by others



Demo: `git merge --squash`

Best practices – clean up local branches

Clean up local branches once the code gets pushed to target branch



Best practices - concise commit messages

Limit commit message Subject to 70 chars and it must be simple, clear, self explained – this is the line all will see!
Meaningful details put in the body of the commit message

```
commit f6ce5cc010bf6665a8f2a701e7983e0c2ac8f144
Author: Shawn O. Pierce <sop@google.com>
Date: Thu Nov 29 09:55:47 2012 -0800
```

```
Sort comments before emailing them
```

```
The order supplied by the caller can be random, ensure comments
get sorted into a sane order before they are included into the email.
```

```
Bug: issue 1692
```

```
Change-Id: ibd85e514977545d022f936a5993f2a6ef6e52321
```

Demo: Example:

<https://github.com/datreeio/node-datreeio/commits/master>

Best practices – Branch Layout

Branch layout

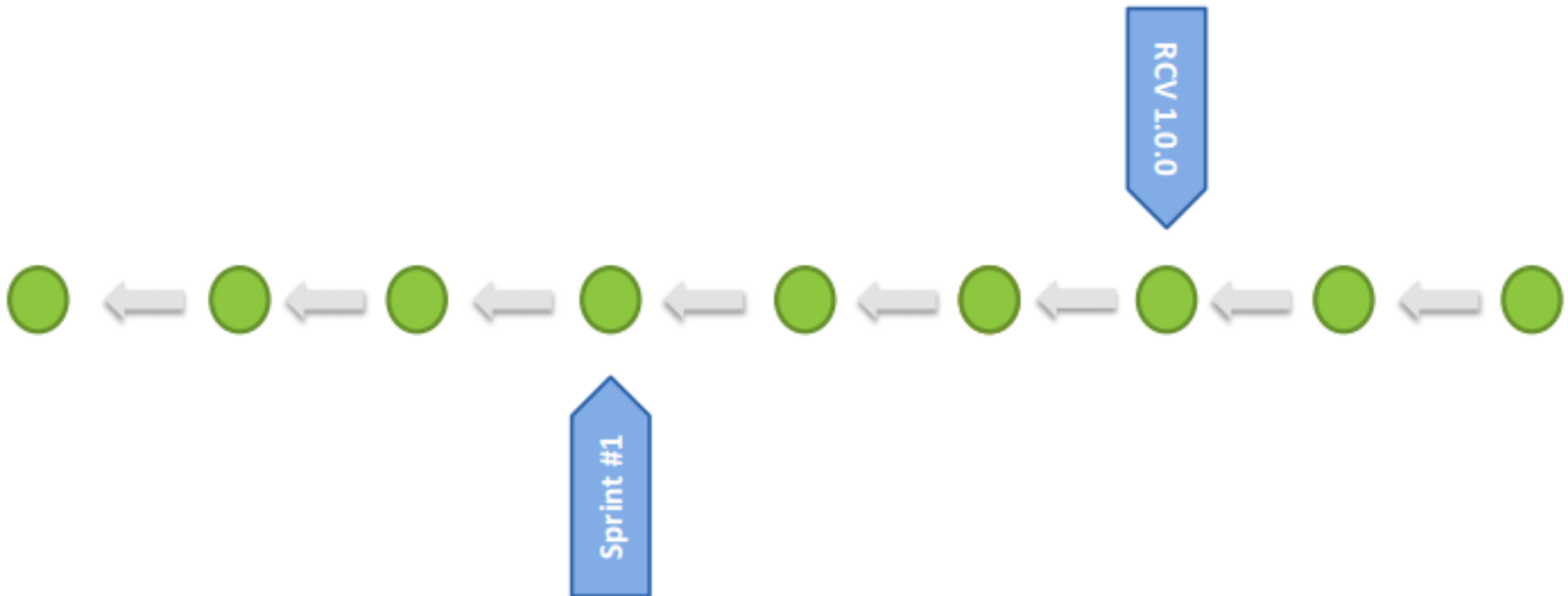
The branch layout is up to you, but there are some best practices though:

```
$ git branch # GOOD
master
* devel
feature/new-mailform
fix/off-by-one
fix/readme-grammar
```

```
$ git branch # BAD
master
* devel
new
fix
fix2
t3rrible-br@nch-name
```

Best practices – tag milestones

Tag important milestones (for history and for accessibility)



Question?

Ilya Rokhkin (Git Trainings)

rokhkin_ilya@yahoo.com

054-5224805

Thanks!