

Why we should use a GitFlow?

Feb. 3, 2020

Paul FIORENTINO

GitFlow?

GitFlow?



Git

GitFlow?



Git + Flow

Why we should use a GitFlow?

In the beginning it was



Why we should use a GitFlow?

SVN

Single repository / multiple clients

Simple purpose, **simple** tool

SO SIMPLE



Why we should use a GitFlow?

SVN

Simple, yep, but limited

Client/Server → needs permanent connection

<ctrl> + S =  for everyone

But above all...

Why we should use a GitFlow?



Why we should use a GitFlow?

```
#> git init
```

```
#> git commit -m "A whole new world"
```

Why we should use a GitFlow?

GIT

Multiple repositories → Local repository

Why we should use a GitFlow?

GIT

Multiple repositories → Local repository

No server needed



There is no Internet connection

You can try to diagnose the problem by taking the following steps:

Go to **Applications > System Preferences > Network > Assist me to test your connection**.

Try:

Why we should use a GitFlow?



Why we should use a GitFlow?

And now ?

Save your work

Save it now!

Save it again

What should I do with my current work?

SAVIT!!!

Why we should use a GitFlow?



Why we should use a GitFlow?

GIT the other side of the coin

Way more **complex** than SVN

Complex != complicated

But If you don't pay attention...

Why we should use a GitFlow?



Why we should use a GitFlow?

GIT

Add processes

Working frame

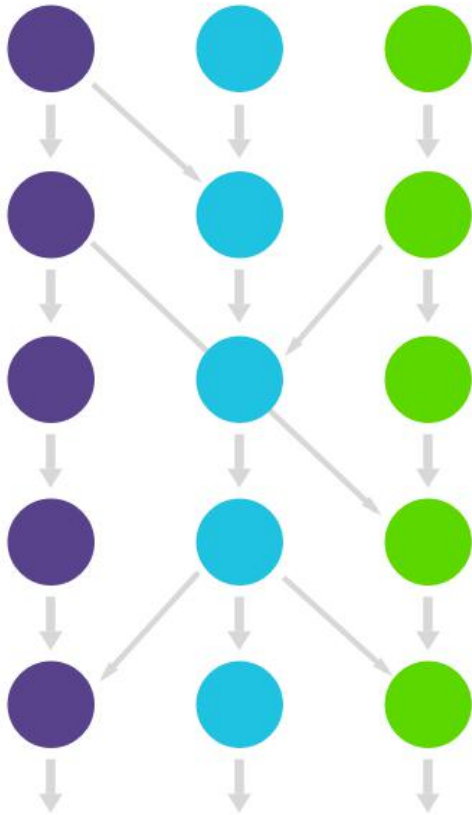
Good practices

(delete as appropriate)

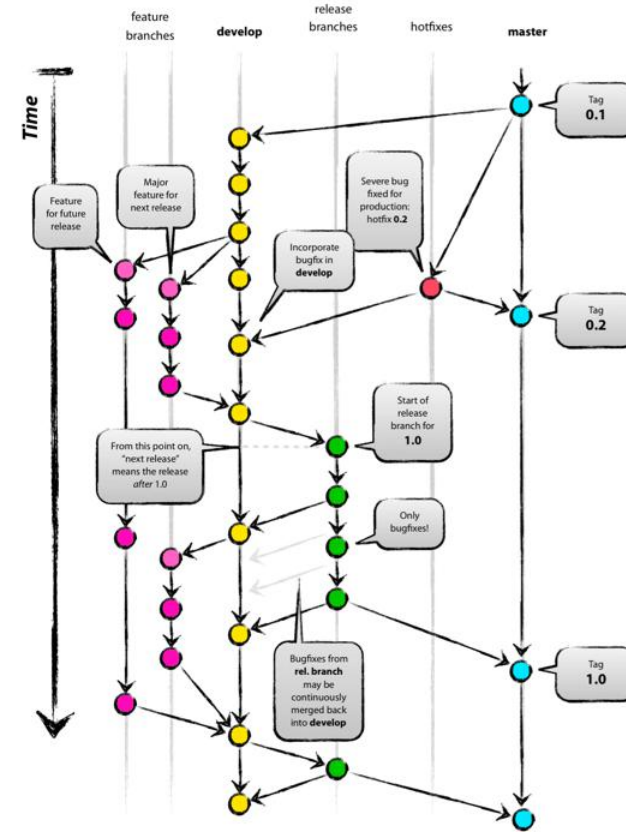
GitFlow?

Why we should use a GitFlow?

GitFlow is **that**:



or **that**:



Why we should use a GitFlow?

Looks confused/confusing/messy (no filiation with Lionel)?

But in reality... Nope, in reality you're totally **right**!

Why we should use a GitFlow?

Main principles

- Branch **everything**
- **`develop`** branch based
- **Strong isolation** of current sprint developments

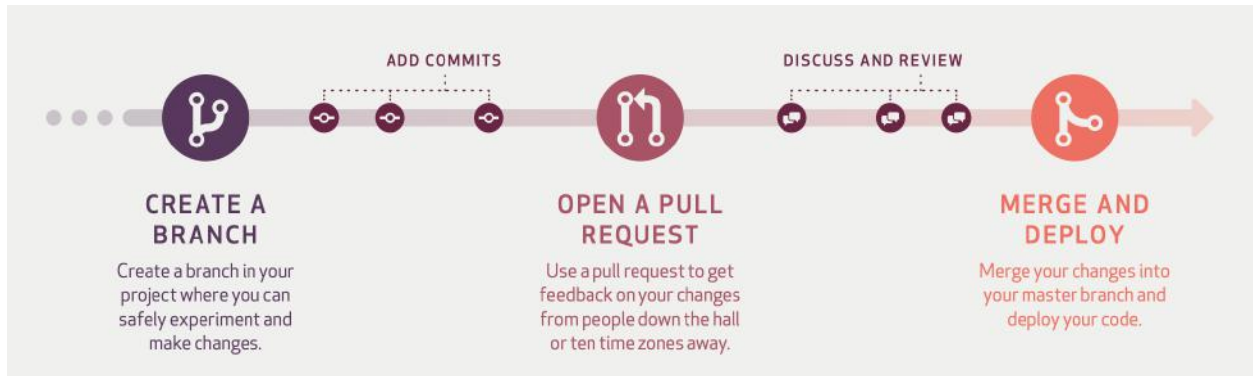
Global thoughts

- Redundant branches/concepts (develop, master, release, tags...)
- Hard to really masterise
- Risky hotfixes
- A lot of manual operations/manipulations
- Is it really useful to keep that amount of different branches (open discussion)?

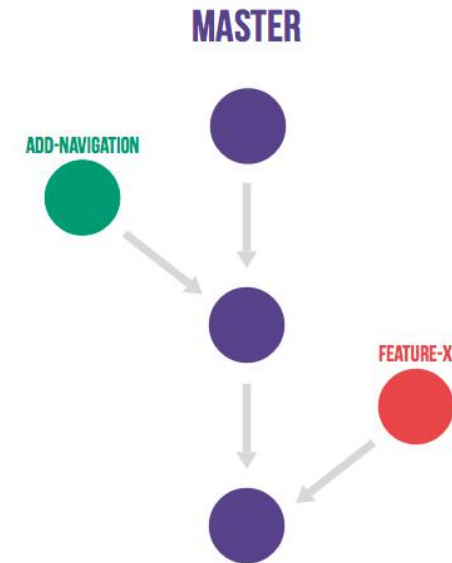
GitHub* teams thought about that
GitHub Flow!

Why we should use a GitFlow?

GitHub Flow is **that**:



or **that**:



Why we should use a GitFlow?

Looks too simple/minimalist?

But in reality... Nope, in reality you're totally **right**!

Why we should use a GitFlow?

Main principles

- Branch nothing but **features**
- **`master`** branch based
- Release **quick**, release **often**
- **Limit** a the extreme **operations**/manipulations

Global thoughts

- Keep branching system minimalist
- Really simple to understand
- Straightforward
- Minimize the unreleased code
- Needs a high organization maturity about Git/CI
- Release each time you merge on master? Really?
- What about deployment, release, the real life... (open discussion)?

GitLab teams thought about that

(And said "Haha n00bs" to GitHub engineers)

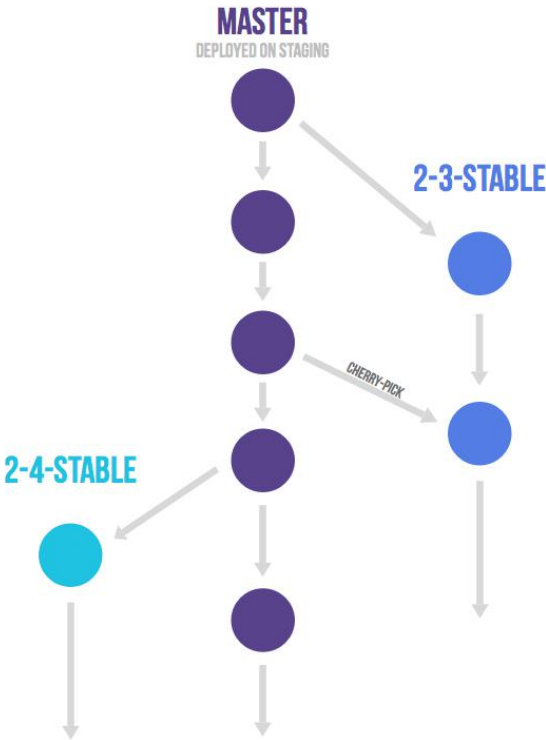
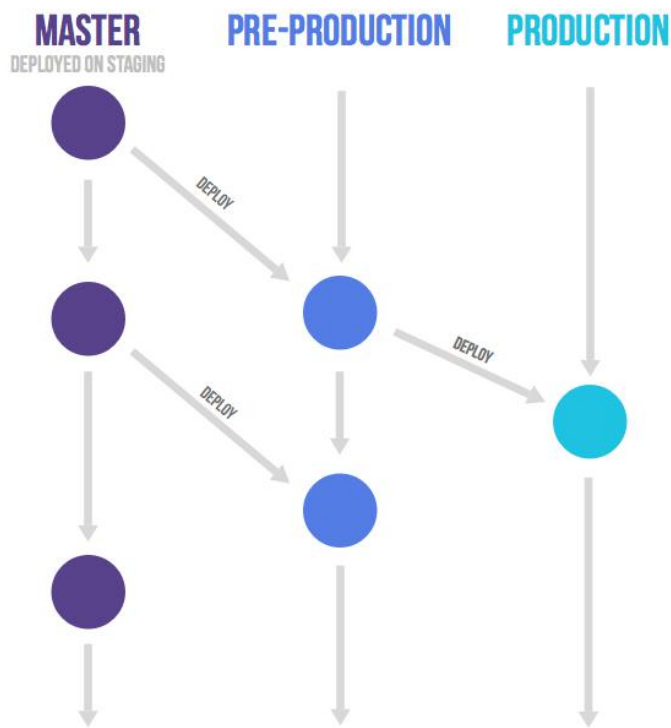
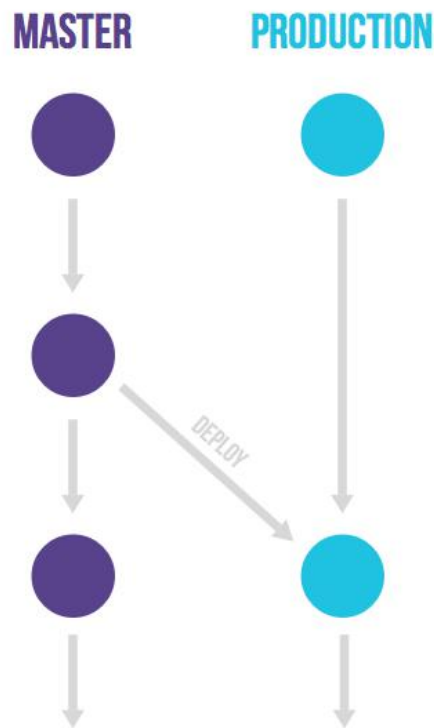
GitLab Flow!

Why we should use a GitFlow?

GitLab Flow is **that**:

or **that**:

or **that**:



Why we should use a GitFlow?

Looks the better mix?

But in reality... Nah... you're not totally **right**!

Why we should use a GitFlow?

Main principles

- Branch what you **need**
- **`master`** branch based
- Release **quick**, release **often**
- **Smart** and **flexible** way to use a Git<random-company> Flow

Global thoughts

- Sexy in prime approach
- Needs a good organization maturity about Git/CI to initialize
- Simple to understand and really fluid to work with
- Minimize the unreleased code
- One branch by release seems to be a little overkill... (open discussion)?
- Freedom can easily leads to messy implementation

WHO ARE WE?



PROGRAMMERS!!!



WHAT DO WE WANT?



EXAMPLES!!!



WHEN DO WE GET?



NOW!!!



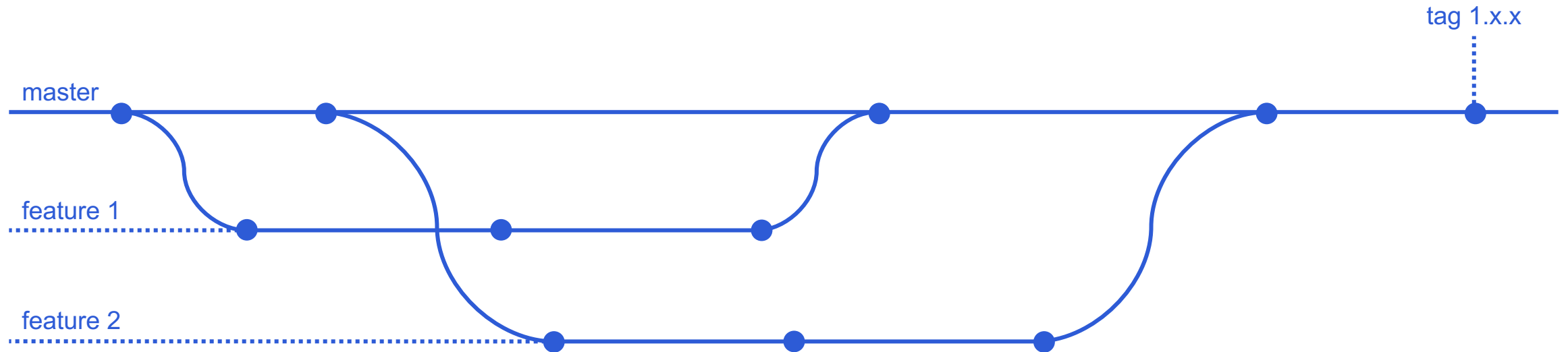
Why we should use a GitFlow?

Scenario's script (by Michael Bay 💣💥)

- In the beginning was the **master branch**
- ~~God~~ Client said, "I command **feature** to be"
- Haters gonna hate, developers gonna **develop**
- And feature started to be (**released**)
- It's not a bug, it's a feature! no? Ok it's **bugfix** time! Yay!

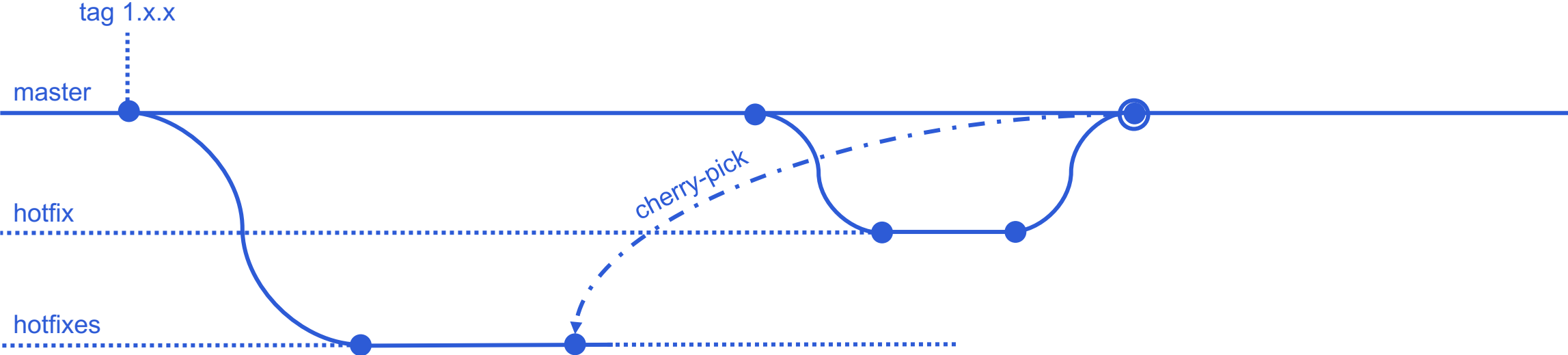
Why we should use a GitFlow?

Development/everyday flow



Why we should use a GitFlow?

Hotfixing flow



Conclusion

Why we should use a GitFlow?

The better flow, your flow

- Good approach (I think)
 - Be **YAGNI** (as always)
 - **Feature** branch == **User Story** (not ticket) (GF, GHF, GLF)
 - “**upstream first**” strategy (GHF, GLF)
 - **master** as the only source of truth (GHF, GLF)
 - Tend to the **trunk based** dev (GHF, GLF)
 - **Lighten** the **release** process (GHF)

Why we should use a GitFlow?

Sources:

<https://blog.xebia.fr/2018/03/28/gitflow-est-il-le-workflow-dont-jai-besoin>

https://docs.gitlab.com/ee/workflow/gitlab_flow.html

<https://guides.github.com/introduction/flow/index.html>

<https://blog.octo.com/git-dans-la-pratique-22>

<https://imgflip.com/>

<https://www.google.com/search>

Thank you