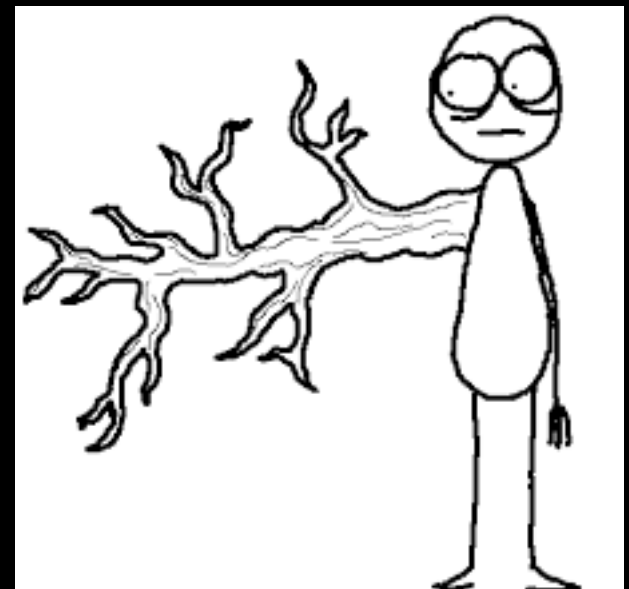


# Advanced Branching in Git

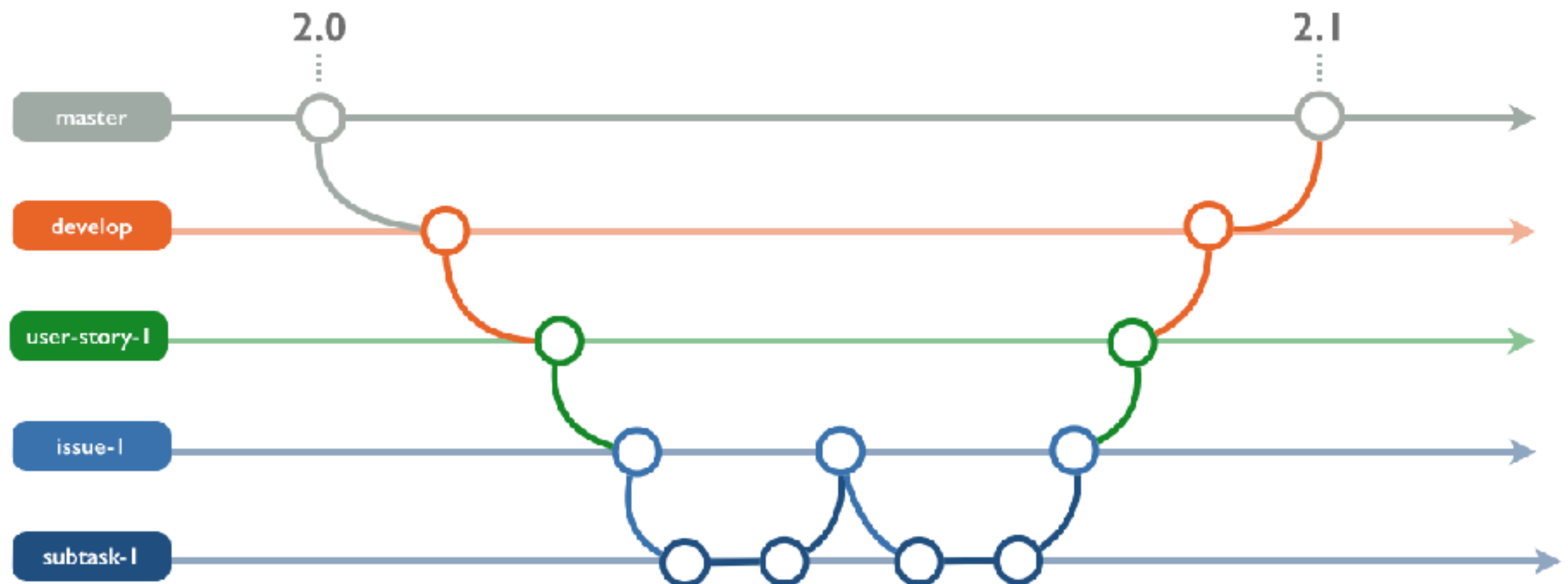
John Parker  
IAM 951



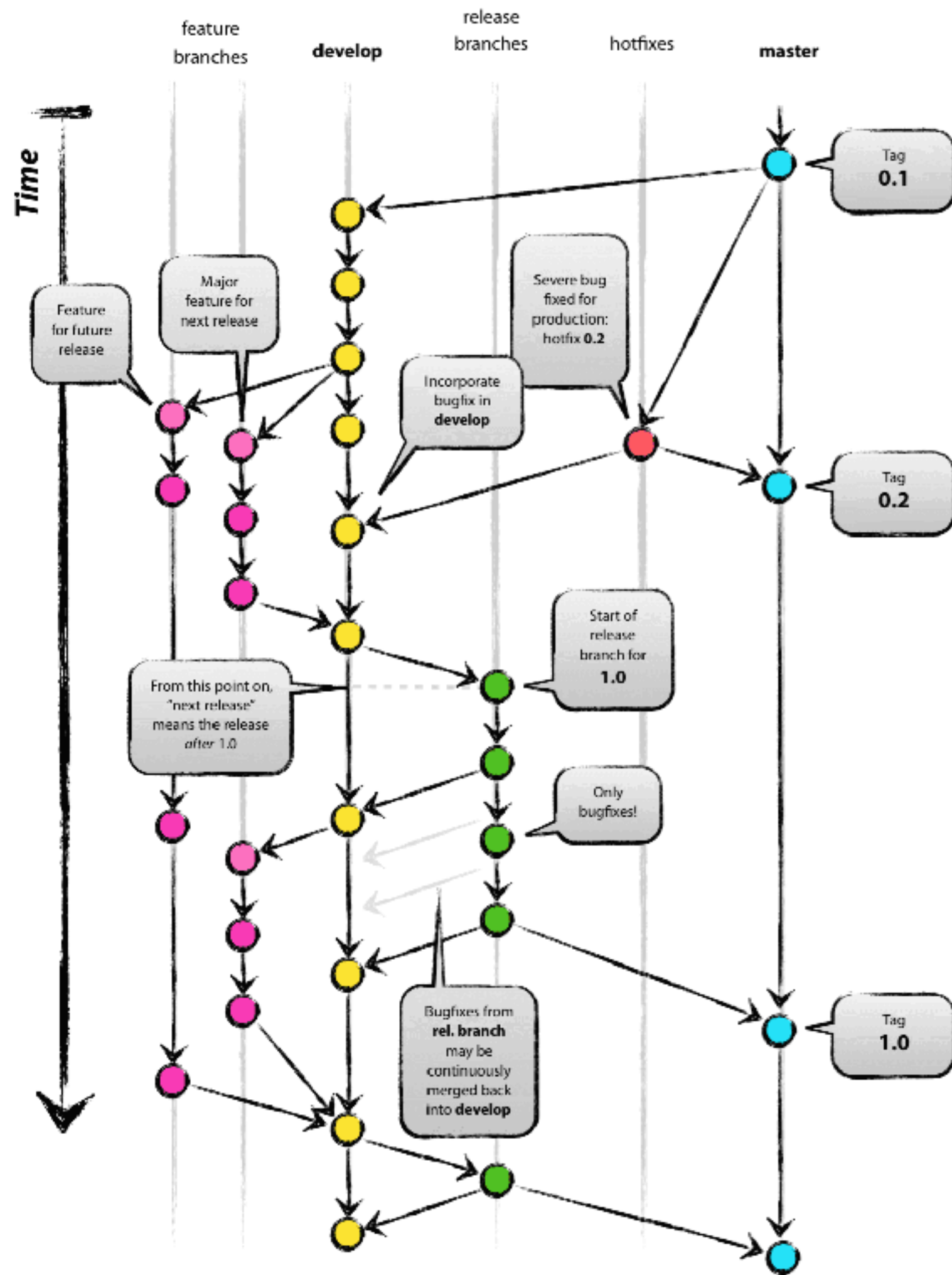
# Bad Branching



## Too many long running branches?



# Good Branching



# When to Branch?

- When it makes sense....
  - Whenever you want “Production” level software, yet still want to develop on or debug software.
  - If you are working on a team, having a plan for branching (and other features of git)

# Why Branch?

- Feature Branch
  - Company A has a product with Customers B and C. However, Customer B wants to see a certain new capability of the code - branch to start development while not affecting Customer C.
- Bug Branch

# Why Branch?

- Feature Branch
- Bug Branch
- Imagine an academic project. Able to produce a desired result. However, you see where you might be able to improve certain error, but you need polished code for either your collaborators or an upcoming deadline. Branch - if you improve the result, fantastic! Otherwise, no penalty.



# Have a Plan for Branching

- How many main/lifetime branches? Plan must fit scope of project.
- What constitutes a branch to emerge from a current branch?
- How do we name new branches/who works on them?
- How to decide when a branch is ready to be merged, and merged where?
- Don't be afraid to adjust plan as long as team is on board.

# Some models

- <http://nvie.com/posts/a-successful-git-branching-model/>
- <https://www.atlassian.com/blog/archives/trust-merge-branch-simplification-musings>