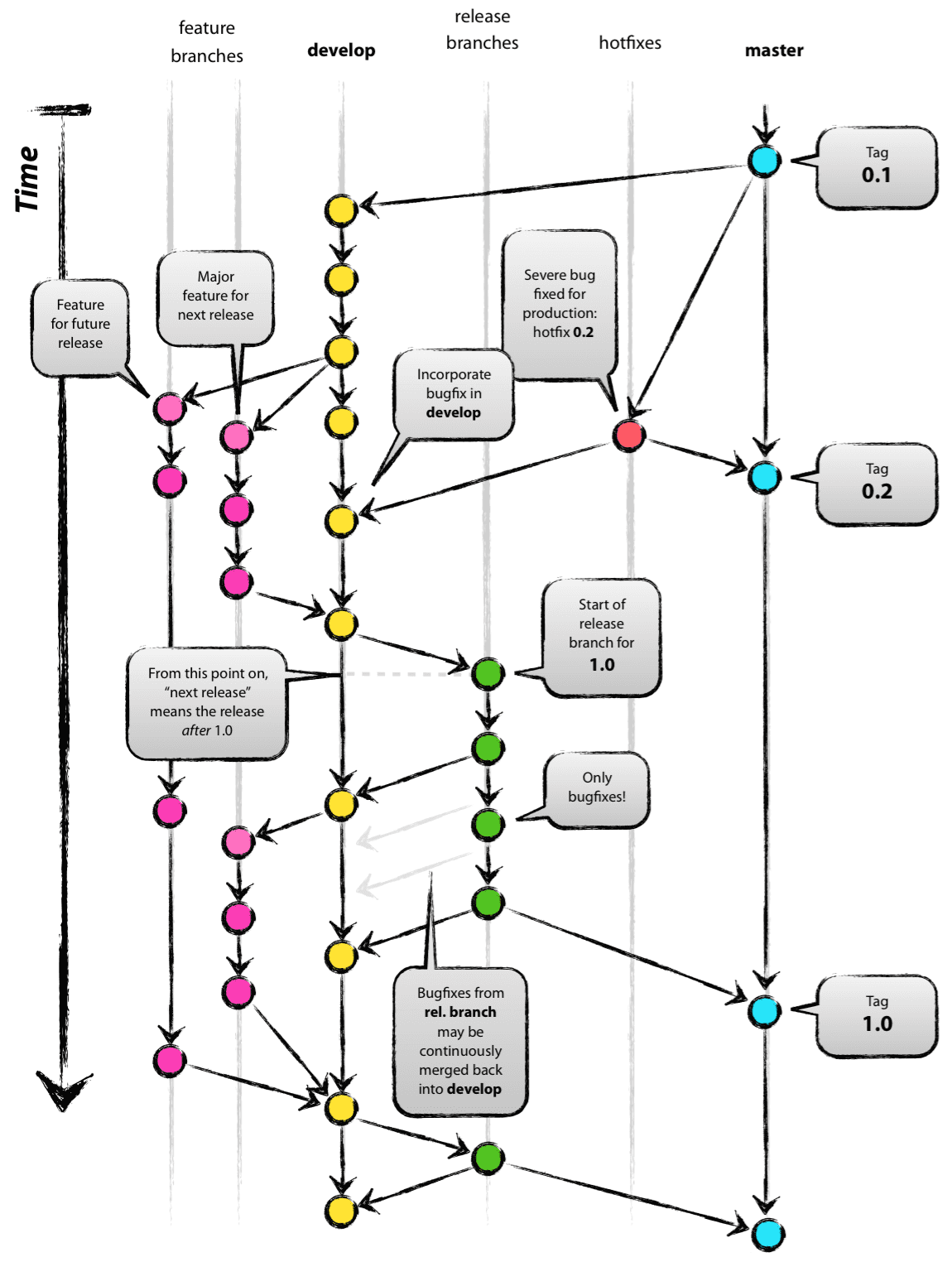
Quy trình làm việc với GIT

# Git Flow trong team

Trong một dự án có sự tham gia phát triển của một team gồm nhiều nhân sự thì việc quản lý code cần một quy trình và đặc điểm kỹ thuật rõ ràng.

Để đáp ứng được vấn đề trên thì mô hình phân nhánh git flow của Vincent Driessen ra đời năm 2010.



# Branch thường được sử dụng trong Git Flow

* **main**

Tức là nhánh main mà chúng ta thường sử dụng, nhánh này là dành cho production, được hợp nhất từ các nhánh khác, nên nhớ không nên sửa đổi trực tiếp trong nhánh này.

* **develop**

Nhánh này là nhánh phát triển chính của team dự án và chứa tất cả code sẽ được phát hành cho phiên bản tiếp theo. **Lưu ý:** nhánh này là hợp nhất của các nhánh khác (**feature branch**) ở dạng tình năng đã hoàn thành.

* **feature/[name of feature]**

Nhánh này chủ yếu được sử dụng để phát triển một chức năng mới. Sau khi quá trình phát triển hoàn tất, thì hợp nhất(merge) trở lại nhánh develop và tham gia các tính năng tiếp theo bằng cách tách một nhánh feature mới từ nhánh develop

* **release**

Khi bạn cần phát hành bản phát hành mới, chúng tôi tạo một nhánh bản release dựa trên nhánh develop. Sau khi branch release hoàn tất, chúng ta chúng ta có thể sử dụng nhánh main để release cũng được, tuy nhiên vẫn khuyến nghị sử dụng nhánh release để lưu các cấu hình môi trường chạy của production

* **hotfix**

Khi tìm thấy một lỗi mới trong dự án đang live, nghĩa là trong môi trường production thì chúng ta cần tạo một nhánh hotfix từ nhánh main hoặc release (tùy bạn production bằng nhánh nào). Sau khi hotfix hoàn tất thì merge/cherry-pick vào develop và main để những thay đổi trong Hotfix sẽ vào bản phát hành tiếp theo.

**Nếu bạn để ý**:

* **Tất cả** feature branch đều là con của develop,
* **Tất cả** hotfix đều là con của release/main
* **Tất cả** release(nếu có) đều là con của develop
* **Không một ai** code trên main
* **Không một ai** code trên develop

# Triển khai gitflow

Lưu ý git flow là một **mô hình quản lý code**, là một cách làm việc chứ không phải là một thư viện gì cả, các bạn có thể tùy biến từng trường hợp để thích hợp với dự án của mình đang quản lý, song song với mô hình gitflow thì bạn **phải có một phương pháp tổ chức code hợp lý** để **hạn chế** các thành viên trong team phát triển code trên cùng một file hay đúng hơn là k**hi hợp nhất các feature vào nhánh develop bị xung đột code/mất code.**

Nếu bạn chưa hiểu mô hình bạn nên xem lại hình ở đầu bài và thật sự hiểu mới có thể bắt đầu. Nếu bạn hiểu rồi thì đây là code CLI tương ứng:

* Tạo nhánh **develop** từnhánh **main**

**git branch develop**

**git push -u origin develop**

* Tạo nhánh **feature** từ nhánh **develop,** và **bắt đầu code** trên feature

**git checkout -b feature/feature-01 develop**

***# add code***

**git status**

**git add file-name *// git add* .**

**git commit**

* Hoàn thành **feature** vàhợp nhất(**merge**)vào **develop**

**git pull origin develop**

**git checkout develop**

**git merge --no-ff feature/feature-01**

**git push origin develop**

***# remove local branch***

**git branch -d feature/feature-01**

***# remove remote branch***

**git push origin --delete feature/feature-01**

* Tạo nhánh **release** từ **develop**

**git checkout -b release-0.1 develop**

* Hoàn thành **release** và hợp nhất(merge) vào **main** để lưu trữ phiên bản hoàn thiện mới nhất

**git checkout main**

**git merge --no-ff release-0.1**

**git push**

***# remove local branch***

**git branch -d release-0.1**

***# remove remote branch***

**git push origin --delete release-0.1**

***# add tag nếu cần***

**git tag -a v0.1**

**git push --tags**

* Tạo nhánh **hotfix** từ **main** nếu có lỗi (quy trình tương tự release)

**git checkout -b hotfix-0.1 main**

* Hoàn thành **hotfix** và merge vào **develop**

**git checkout main**

**git merge --no-ff hotfix-0.1**

**git push**

***# merge to develop***

**git checkout develop**

**git merge --no-ff hotfix-0.1**

**git push**

***# remove local branch***

**git branch -d hotfix-0.1**

***# remove remote branch***

**git push origin --delete hotfix-0.1**

Ngoài ra cũng có một bộ CLI **git extensions** hỗ trợ giúp các bạn **đỡ phải thao tác như ở trên** để tối ưu hóa thao tác khi làm việc theo mô hình này <http://danielkummer.github.io/git-flow-cheatsheet/>

**Lưu ý:** các bạn cũng có thể dùng bất cứ GUI tool cho GIT vào cũng được, miễn đáp ứng quy trình các bước làm việc, khi bạn đã **hiểu phương pháp** thì dù bạn sử dụng mã lệnh hay tool thì nó vẫn sẽ đạt được một kết quả như nhau.

HAPPY CODING.