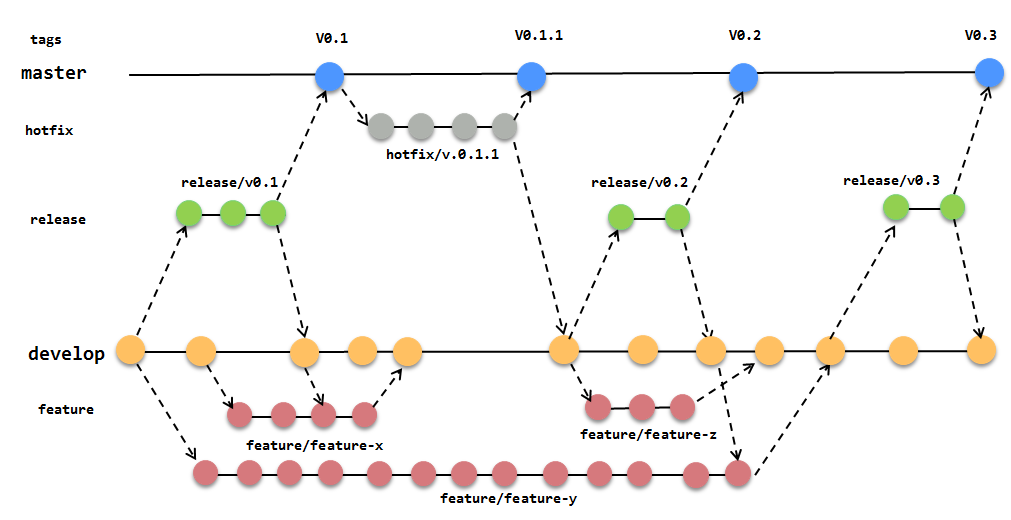
GIT FLOW

Đã là một lập trình viên thì khái niệm về git không còn xa lạ nữa. Nhắc đến git thì chúng ta không thể nhắc đến khái niệm Git Flow, như tên gọi của nó git flow tức là luồng công việc của dự án sẽ được vận hành một cách chặt chẽ, trôi chảy dựa trên mô hình phân nhánh được phát hành bởi Vincent Driessen – một trong những mô hình phổ biến của nhiều loại phần mềm. Thêm vào đó git flow cung cấp các thao tác mở rộng của git nhằm dễ dàng quản lí các dự án lớn.



Minh hoạ gt flow

Tóm tắt

Nói một cách dễ hiểu, như trên hình mỗi một dự án chúng ta sẽ có 2 nhánh chính: master và develop. Nhánh master chính là nhánh hoàn thiện được người dùng sử dụng, và nhánh develop đóng vai trò là nhánh tích hợp các tính năng đã và đang được phát triển.

Nhánh feature: các dev sẽ tạo ra nhánh riêng cho mỗi tính năng và pull *code* từ nhánh develop về code cũng như là fix bug.

Nhánh release: là nhánh tích hợp nhiều tính năng tại thời điểm checkout code từ nhánh develop sang nhánh release. Bộ phận QA sẽ test trên nhánh này

Nhánh hotfix: như tên gọi của nó là “sửa nóng”, sau khi vào production chúng ta mới phát hiện ra lỗi. Chúng ta phải checkout code từ nhánh master sang hotfix và các dev sẽ vào nhánh này fix nhanh nhất có thể và sau đó merge lại vào master.

Hoạt động chi tiết

Đối với 7 Lab chúng ta sử dụng BitBucket để quản lí dự án. Ngoài ra còn sử dụng Jira để phân công, theo dõi tiến độ công việc.

Feature Branch

Khi dev nhận task trên jira, thì việc đầu tiên là phải tạo 1 branch trên bitbucket theo tên task của mình được phân công. Sau đó checkout code từ nhánh develop về nhánh đó về code cũng như fix bug.

Sau khi làm xong chúng ta phải tạo 1 pull request trên bitbucket và giao lại cho người review để check logic có đúng chưa, thiếu trường hợp nào không, và để có thể chỉnh sửa code trước khi chuyển qua cho bộ phận QA test tính năng đó.

Sau khi được reviewer chuyển sang trạng thái TO TEST trên jira thì động thái tiếp theo của dev là phải tạo ra 1 bản RC “Release Candidate” và thêm version kèm theo (SE-7505-rc.1) – vì code có thể lỗi ta phải kèm them version để QA biết chính xác version nào để test và giao lại cho QA. Nếu bị QA đánh TEST FAILED thì dev phải chịu trách nhiệm fix lại cho đúng test case, ngược lại nếu TEST PASSED thì Jira sẽ tự động chuyển sang cho bộ phận PM (Project Manager) với trạng thái TO PRODUCTION, tiếp đó PM lại chuyển cho dev leader của component đó để merge vào develop.

Trong trường hợp task đang được thực hiện chưa hoàn thành thì lại có 1 task khác ập đến bắt buộc dev phải hoàn thành cấp tốc. Không còn cách nào khác ta phải “git stash” để lưu lại những thay đổi trên workspace. Sau đó phải checkout qua nhánh cấp tốc đó để làm.

Ngoài ra còn có trường hợp dev phải làm các sub-task dựa trên 1 tính năng cụ thể và tính năng chưa được release định kì (1 tuần 1 lần), tức là nó còn đang nằm trên nhánh feature chưa được merge với nhánh develop có thể là đang được chờ QA test hay chưa đến thời gian release định kì. Chúng ta phải checkout từ nhánh có tính năng đó chứ không phải lúc nào cũng từ nhánh develop.

Release Branch

Khi nhánh develop đã đủ các tính năng cần thiết hay gần đến ngày release định kì. Bước tiếp theo là phải tách ra một nhánh là release kèm theo version từ nhánh develop.

Mục đích của nhánh này là tích hợp tất cả các tính năng từ đầu dự án đến thời điểm hiện tại để kiểm tra lại lần cuối trước khi đưa vào production (master). Sau khi TEST PASSED của từng tính năng trên nhánh feature nhưng cũng chưa hẳn là trên nhánh release sẽ hoạt động mượt mà. Vì trên nhánh release lúc này có rất nhiều components với nhau.

Nếu có bug các dev được chỉ định sẽ checkout thì nhánh release này để fix và đặc biệt trên nhánh này không được code thêm tính năng.

Cuối cùng mọi thứ hoạt động ổn thoả, việc còn lại là merge nhánh release này vào master và không quên merge vào nhánh develop. Vì trên develop không được phép có những bug trước đó (sau khi fixed trên nhánh release). Và việc sau đó là chỉ việc checkout từ nhánh develop về code tiếp các tính năng mới.

Hotfix Branch

Không phải lúc nào bug cũng sẽ luôn được phát hiện trong lúc test, nếu đã vào master mới phát hiện ra bug, thì lúc này đây nhánh hotfix chính là cách chữa cháy.

Tại đây ta phải tạo 1 nhánh tên *hotfix 0.1.1* từ nhánh master rồi nhờ các dev vào đây sửa. Tương tự như nhánh release, sau khi fix xong ta phải merge code lại vào nhánh master và develop.