**Problems**

Dự án có nhiều thành phần tham gia:

* ~ 40 Developers từ Rikkei
* ~ 20 Developers Nội bộ
* Nhóm Testers, BA, PM, Nhân viên kinh doanh Trung tâm BC, Nhóm Khách hàng Vinaphone

Khoảng thời gian giữa 2 lần release thường rất ngắn (tối thiểu 1 tuần 1 bản cập nhật nhỏ, 1 tháng sẽ có một bản cập nhật lớn)

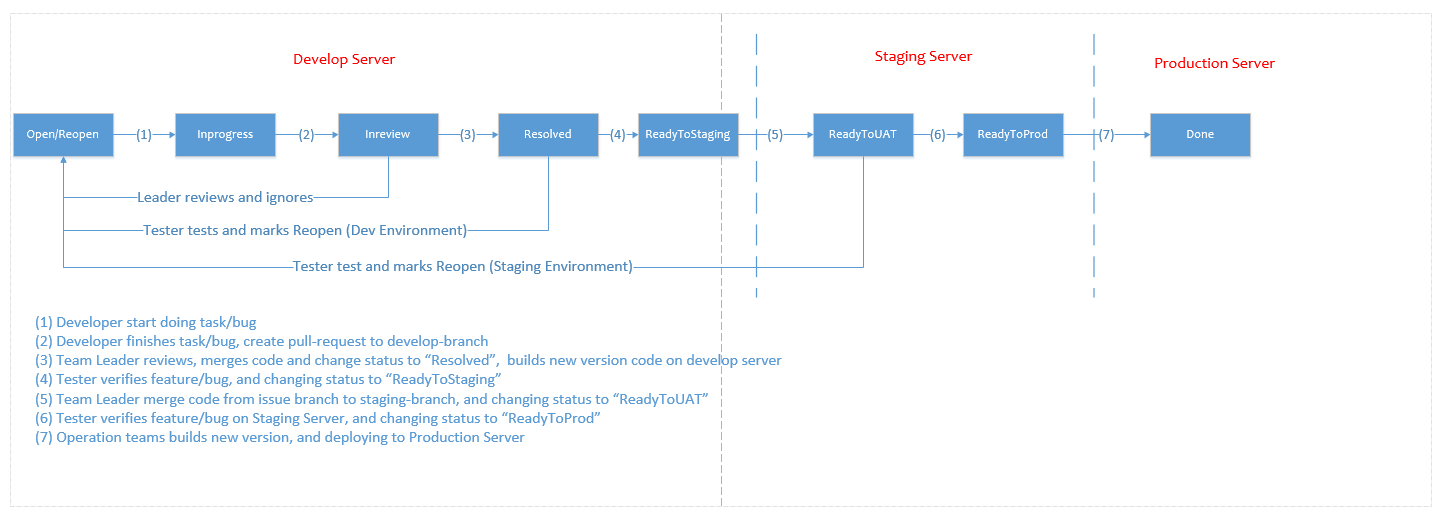
Đã có sự cố xảy ra về vấn đề kiểm soát mã nguồn, trạng thái issues dẫn đến việc phải rollback trên môi trường Production (mất rất nhiều thời gian, nguồn lực, lời xin lỗi đến khách hàng để khắc phục vấn đề)

**Methodologies**

Nhóm dự án mong muốn áp dụng một cách chặt chẽ và hiệu quả CI/CD vào quá trình phát triển phần mềm

**Solutions**

***Quy trình Jira***



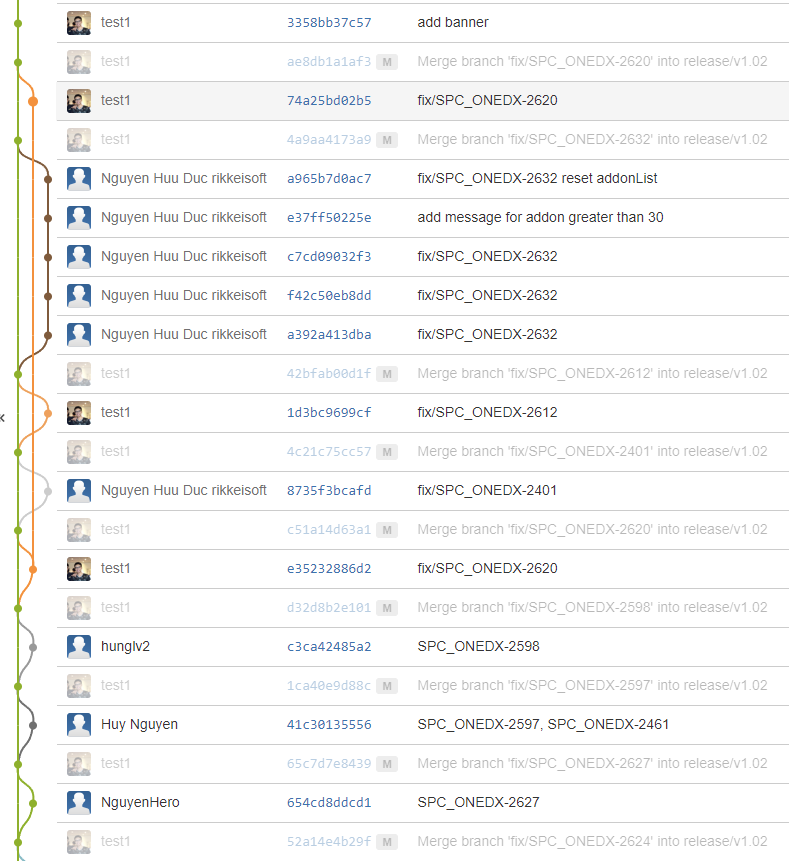
***Quy trình Git***

Một số câu hỏi:

* Rebase tốt hơn hay là Merge tốt hơn?
* Vấn đề review code

Nguyên tắc mà technical lead đưa ra yêu cầu các thành viên tuân thủ:

* Luôn lấy code mới nhất về nhánh base(nhánh ban đầu dùng để tạo nhánh của mình) trước khi commit lên nhánh của mình.
* Mọi commit đều cần phải chứa id task jira tương ứng.
* Tận dụng rebase (tránh việc dùng pull và merge không đúng mục đích) để duy trì đường git thẳng, tránh các commit merge thừa không cần thiết (ví dụ hình dưới). Việc này tạo ra nhiều phức tạp không đáng có khi phải merge code trong một dự án nhiều developer, nhiều thành phần tham gia.



**Implementation**

Một số câu lệnh Git được sử dụng trong dự án (tuyệt đối không sử dụng pull)

* fetch
* rebase (rebase –continue, rebase –skip, rebase –abort)
* merge –squash, merge –no-commit
* cherry-pick (cherry-pick –no-commit, cherry-pick –continue, cherry-pick –abort)
* reset –hard, reset –soft
* stash, stash apply
* push, push tags origin
* checkout, checkout -b
* tag -a

Một số case study sử dụng Git trong dự án

1. Phát triển mới một tính năng (tính năng lớn cần nhiều developers cùng tham gia)

* Team lead tạo ra nhánh code mới từ nhánh staging (ví dụ branch với tên feature/SPC\_ONEDXINTERNAL-680)
* Các thành viên trong nhóm tạo ra branch nhánh từ branch cha feature/SPC\_ONEDXINTERNAL-680 và tiến hành làm việc trên các branch cá nhân này (ví dụ task SPC\_ONEDXINTERNAL-681, SPC\_ONEDXINTERNAL-682 là 2 trong số nhiều subtask của SPC\_ONEDXINTERNAL-680 sẽ có các branch tương ứng feature/SPC\_ONEDXINTERNAL-681, feature/SPC\_ONEDXINTERNAL-682)
* Hàng ngày thành viên làm subtask đều đặn cập nhật code từ nhánh cha về, tránh việc code bị out-of-date quá lâu
  + Trường hợp chưa tạo commit, có các file thuộc danh sách thay đổi

git stash

git fetch

git rebase origin/feature/SPC\_ONEDXINTERNAL-680

git stash apply

* + Trường hợp đã tạo commit ở local

git fetch

git rebase origin/feature/SPC\_ONEDXINTERNAL-680

Trường hợp bị conflict trong khi rebase thì thực hiện fix tất cả các conflict sau đó tiếp tục git add, git rebase –continue

Tiếp tục bị conflict thì tiến hành lặp lại bước trên

* Tiến hành tạo commit, push lên nhánh subtask
* Lưu ý trước khi push lên nhánh subtask luôn luôn rebase code mới nhất từ nhánh cha về (origin/feature/SPC\_ONEDXINTERNAL-680)
* Tiến hành tạo pull-request để teamlead review code

1. Merge code của toàn bộ nhánh (chứa code của 1 nhóm tính năng lớn) vào nhánh chứa code của môi trường test (ví dụ nhánh internal\_test).

* git checkout internal\_test
* git fetch
* git reset origin/internal\_test –hard
* git merge –squash origin/feature/SPC\_ONEDXINTERNAL-680
* Tiến hành fix các conflict nếu có
* git add . (add các file sau khi đã fix conflict)
* git commit -m “feature/SPC\_ONEDXINTERNAL-680: tính năng ….”
* git push origin internal\_test

1. Khi tiến hành test các chức năng của nhánh feature/SPC\_ONEDXINTERNAL-680, có thể xuất hiện lỗi, cần có các commit sửa lỗi (ví dụ task bug SPC\_ONEDXINTERNAL-720

* Thành viên tạo nhánh nhánh fix/SPC\_ONEDXINTERNAL-720 và tiến hành fix bug trên nhánh này
  + Tiến hành fix bug trên nhánh đã tạo
  + Chủ động rebase với nhánh SPC\_ONEDXINTERNAL-680 để lấy code mới nhất
  + Tạo commit và push lên nhánh SPC\_ONEDXINTERNAL-720
  + Tạo pull-request vào nhánh SPC\_ONEDXINTERNAL-680, liên hệ teamlead để review code và đưa code nhánh SPC\_ONEDXINTERNAL-720 vào nhánh chính SPC\_ONEDXINTERNAL-680
* Các bản cập nhật fix bug cần phải được đưa vào trong nhánh internal\_test, build và triển khai
  + Tìm commit cuối cùng được merge từ nhánh SPC\_ONEDXINTERNAL-680 vào nhánh internal\_test
  + tiến hành pick các commit mới sang bằng lệnh git cherry-pick <<mã hash của commit>>
  + Trong quá trình cherry-pick có thể xảy ra conflict, tiến hành fix các conflict
  + git add . (add các file đã fix conflict)
  + git cherry-pick –continue
  + git push origin internal\_test

1. Sau khi đã test thành công trên môi trường test, cần chuyển code nhánh SPC\_ONEDXINTERNAL-680 lên môi trường UAT để test UAT. Tương tư như trường hợp 2, tiến hành merge –squash code của nhánh SPC\_ONEDXINTERNAL-680 vào nhánh staging
2. Sau khi đã hoàn thiện test UAT trên môi trường staging, tiến hành đánh tag cho code trên nhánh staging, sau đó triển khai production

Về mặt nguyên tắc thì cần chuyển code đã triển khai vào nhánh production, nhưng thực tế có thể chỉ cần quản lý các tag đã được đánh version trong branch staging