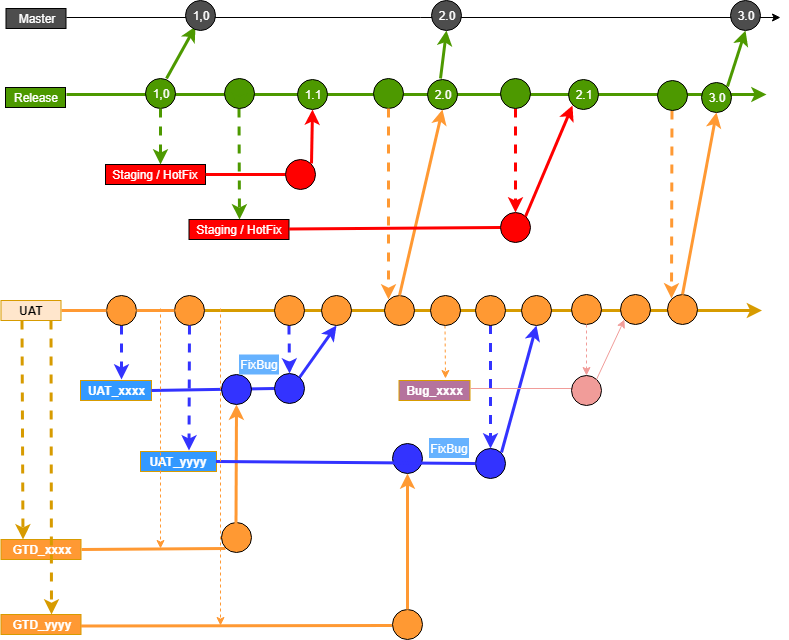


**Mô hình phân nhánh GIT**

**Phiên bản : 3.0**

**06/2019**

1. **Mô hình tổng quát**

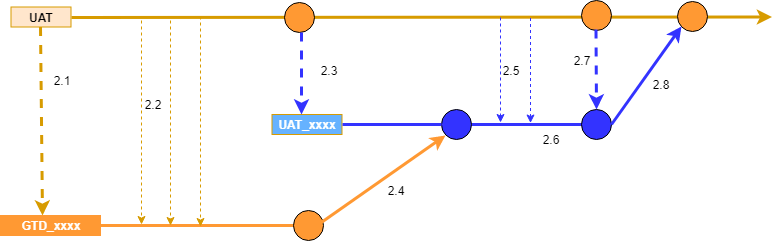
****

Repo trung tâm sẽ có 3 branch chính hoạt động mãi mãi :

1. **Master** : Source baseline. Stable theo từng version release. Chịu trách nhiệm backup và restore khi nhánh release có vấn đề.
2. **Release** : Nhánh dùng để build staging / production. Nhánh này phản ánh tình trạng hiện tại của production.
3. **UAT** : Nhánh phản ánh sự thay đổi mới nhất trong quá trình phát triển tính năng. Nơi chuẩn bị cho đợt release tiếp theo.

Các branch phụ có thể xóa sau khi deploy và test pass trên môi trường production.

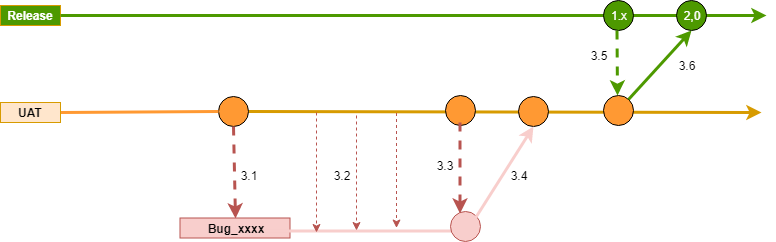
1. **Branch Feature**

****

Cách thức hoạt động:

* 1. Khi bắt đầu phát triển feature mới, dev sẽ clone source từ branch UAT và đặt tên có tiền tố theo task được được giao trên jira. Ví dụ tên branch : **GTD\_1234\_FE\_MassOffer**. Trường hợp nhiều dev làm cùng 1 task (vừa có FE vừa có BE) thì các dev làm chung trong 1 branch này, tránh trường hợp 1 dev code 1 branch riêng rồi khi test không biết merge vào đâu.
  2. Mỗi ngày dev sẽ pull code từ UAT về để đảm bảo code luôn được update mới nhất (Team Lead có trách nhiệm nhắc nhở đôn đốc).
  3. Sau khi dev phát triển xong tính năng và self check xong, dev sẽ thông báo cho Lead review, sau đó Lead sẽ request QC test. QC kiểm tra resource, nếu có người test và máy UAT trống sẽ request devops deploy trên máy UAT đó để test. Dev clone source ra từ UAT đặt tên theo quy định : tiền tố UAT + task trên jira. Ví dụ **UAT\_GTD\_1234\_FE\_MassOffer**
  4. Team Lead Merge source từ nhánh feature dev lên (trong ví dụ này là merge từ **GTD\_1234\_FE\_MassOffer🡪 UAT\_GTD\_1234\_FE\_MassOffer**). Devops deploy branch này lên máy UAT trống.
  5. Trong quá trình test tính năng này (gọi là test feature hay test Round 01) nếu có bug sẽ fix trên branch này luôn. Mỗi ngày dev sẽ lấy code mới nhất từ UAT về để update.
  6. Dev fix tất cả bug QC phát hiện.
  7. Sau khi QC báo test feature done. Team Lead sẽ merge code từ branch UAT xuống branch UAT Feature để xử lý conflict.
  8. Team lead merge code sẽ từ branch UAT feature vào UAT để test tích hợp (gọi là Integration test hay test Round 02). Devops deploy UAT.

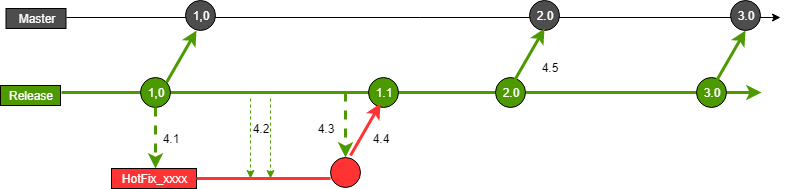
1. **Branch UAT**

****

Cách thức hoạt động :

* 1. QC báo bug khi Integration (test Round 02), dev sẽ clone source từ branch UAT đặt tên theo rule : Bug + bugId trên Jira. Ví dụ **Bug\_GTD\_1234\_bamboo**.
  2. Dev sẽ tiến hành fix bug trên branch này, nếu quá trình fix bug kéo dài, thì mỗi ngày dev cũng sẽ thực hiện việc Pull code từ UAT xuống để update.
  3. Khi bug được fix xong, dev sẽ request merge vào UAT để test. Team lead review và merge code từ UAT xuống, xử lý conflict.
  4. Team lead merge code vào UAT. Devops deploy UAT để test bug.
  5. Sau khi QC test bug này xong, sẵn sàng deploy. Team Lead sẽ merge code từ release xuống để update code mới nhất. Sau đó request devops merge vào Release.
  6. Khi có confirm từ QC, devops thực hiện merge và deploy lên production.

1. **Branch Release**

****

Cách thức hoạt động:

* 1. Khi có hotfix, dev sẽ clone source từ nhánh release, đặt tên theo rule : HotFix + idJira. Ví dụ: **HotFix\_GTD\_5678\_FE\_Bamboo\_Promotion**
  2. Nếu HotFix kéo dài thì mỗi ngày dev cũng sẽ pull code từ Release xuống.
  3. Sau khi fix xong, dev tạo merger request từ Release xuống branch HotFix. Team Lead review merge code, xử lý conflict, Devops tiến hành deploy branch này lên môi trường **staging** để QC test. Sau khi test pass sẽ tiến hành qua bước tiếp theo.
  4. Team lead tạo merge request lên Release. Devops tiến hành deploy release.
  5. QC retest trên release. Khi done sẽ request devops backup lên nhánh master (chỉ nên tiến hành backup ở các tính năng lớn).

**Ghi chú : Tại một thời điểm chỉ nên tiến hành 1 HotFix để clear từng tính năng. Không nên có quá nhiều HotFix**. **Các HotFix không phụ thuộc vào nhau có thể tiến hành đồng thời, nhưng tất cả phải pass được trên staging mới release.**