



BRANCHING PATTERNS AND PULL REQUEST

María Suárez Hevia UO270437
Miguel Menéndez Rodríguez UO269871

BRANCHING PATTERNS

Existen dos categorías principales:

Patrones de integración: muchos desarrolladores combinan su trabajo en un conjunto coherente.

Patrones de producción: usar ramas para llegar de un código base a un producto funcionando (en producción).

Ramas basicas

Source Branching: Crear una copia de la actual y guardar en ella todos los nuevos cambios. (develop)

MainLine: La mainline es una rama activa, por lo que se va actualizando constantemente. Por tanto, si está cuidada, dicho código puede lanzarse a producción. (master)

Healthy Branch: Rama sin errores, sobre la que se suelen hacer tests.

Patrones de integración

MainLine Integration: pull de mainline, merge y si es seguro push a mainline.

Feature Branching: se crea una rama nueva para cada nueva funcionalidad, requisito o historia de usuario.

Continuous Integration: se hace commit tan pronto como se tenga un código sin errores que se pueda compartir, normalmente de menos de un día de trabajo.

Pre-Integration Review: Cada commit será revisado antes de ser aceptado.

Patrones de producción

Release Branch: De forma típica, esta rama copiará lo que hay en la mainline, sin ningún tipo de adición. Los desarrolladores se encargarán de encontrar errores y arreglarlos, haciendo un merge hacia la mainline.

Maturity Branch: Esta rama se utilizaría una vez que el código sea lo suficiente estable y sin fallos. De esta manera, en esta rama estaría la versión más estable del programa.

Environment Branch: Esta rama se usa para probar el producto en distintos entornos. Así, se abre una nueva rama cada que haya una prueba en un entorno diferente, cambiando pequeñas partes del código relativas a la configuración del programa.

Hotfix Branch: Se usa cuando hay un bug importante en el programa que necesita un arreglo urgente. Se coge la última versión lanzada al público, teniendo usualmente una mayor prioridad que el resto de ramas. Cuando esté arreglado, se hace el merge en mainline.

Si están utilizando una release branch, los cambios se pueden hacer sobre esta y llevarlos a la mainline.

Release Train: Este enfoque se basa en crear ramas de una forma periódica para su lanzamiento. Por tanto, cuando se lance esa versión del programa, se crea una nueva rama para la próxima versión, y la anterior se convierte en una release branch, enfocándose en los errores existentes.

Release-Ready Mainline: Si el código en mainline tiene apenas errores, se puede lanzar directamente a producción, sin tener que crear más ramas. Para cada versión lanzada se crearía un tag.

Otro tipo

Experimental Branch: Se crean cuando se quiere probar algo nuevo pero que no suele ser añadido a la mainline.

Future Branch: Poco común (aparece a veces con Continuous Integration), se usa cuando un cambio es demasiado intrusivo en el código base. Al contrario que la feature branch, solo hay una al mismo tiempo.

Collaboration Branch: Se usa cuando se utiliza Mainline y se necesita enseñar el código realizado a otras personas del equipo. Dura hasta que se integra el código en mainline.

Team Integration Branch: Permite a los mismos miembros de un equipo juntar su código antes de integrarlo con el código del resto de miembros del proyecto. Sería el equivalente de mainline para un equipo.

PULL REQUEST

Un pull request es una petición que un usuario hace al grupo de propietarios de un repositorio para poder incorporar sus commits (propuestas o cambios en el código del proyecto) a este.

Una vez solicitado el grupo podrá aceptarlo o rechazarlo y además pueden hacer comentarios sobre él con posibles cambios, mejoras u opiniones.

BIBLIOGRAFÍA

<https://martinfowler.com/articles/branching-patterns.html>

<https://openwebinars.net/blog/git-flow-tipos-de-ramas/>

<https://cleventy.com/que-es-git-flow-y-como-funciona/>

<https://buddy.works/blog/5-types-of-git-workflows>

<https://aprendegit.com/que-es-un-pull-request/>

<https://styde.net/pull-request-en-github/>

<https://martinfowler.com/bliki/PullRequest.html>