

Git

gitflow

Czego się dowiesz?

- Czym jest gitflow?
- Jak wygląda najpopularniejszy proces dostarczania kodu?

Branches w Gitflow

master - kod klienta

develop - kod rozwojowy, developerski

feature - tutaj rozwijamy konkretne zadanie

release - tutaj przygotowujemy kolejne wydanie

hotfix - tutaj naprawiane są błędy

master

master - to główny branch. Kod na tym branchu jest:

- stabilny
- przetestowany
- otagowany, oznaczony wersją (np. 1.0)

Kod, który jest na masterze, to ten, który dostają klienci. Master jest branchem, który chronimy przed nieprzemyślanymi zmianami. Ograniczamy więc liczbę osób, które mogą go modyfikować.



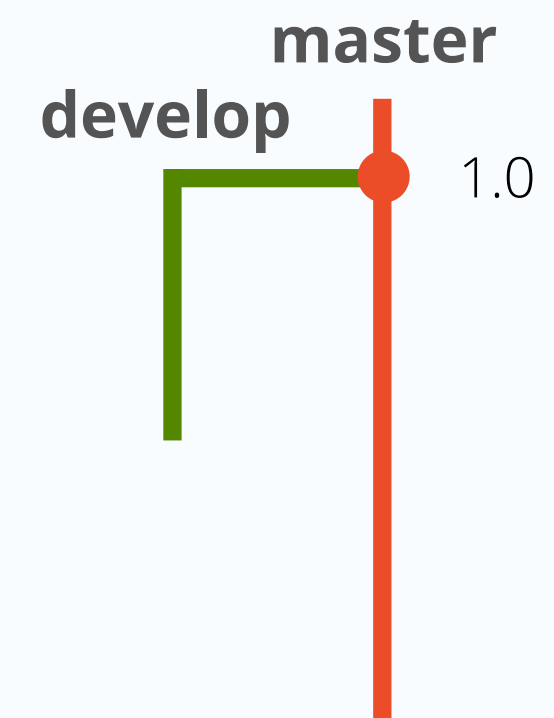
develop

develop - to branch rozwojowy. Na tym branchu:

- przechowywane są nowe (gotowe) funkcjonalności

Develop jest masterem dla programistów. Ostatecznie na develop trafiają gotowe funkcjonalności.

Develop wychodzi z mastera.



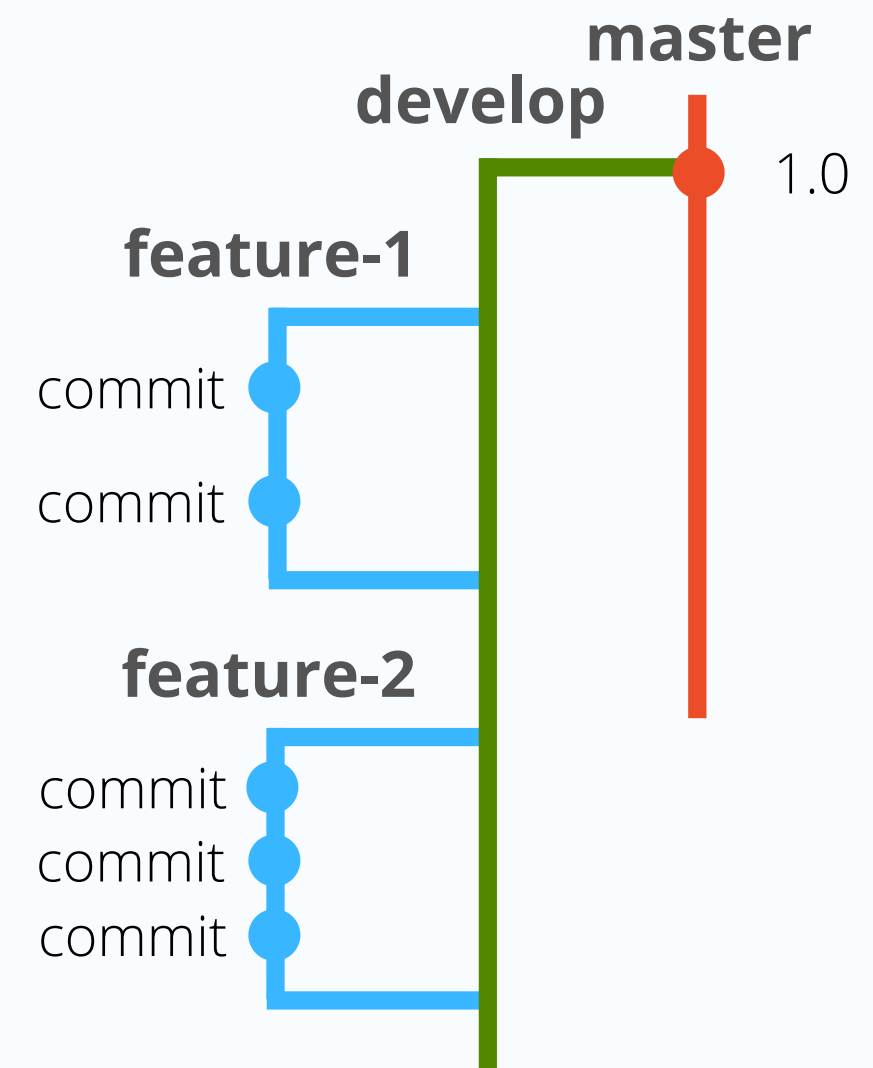
feature

feature - to branch rozwojowy. Na tym branchu:

- tworzymy konkretną funkcjonalność
- sprawdzamy różne koncepcje, które mogą zostać też porzucone

Na feature branch tworzymy kod, który docelowo trafi na develop. Dzięki wyodrębnieniu feature możemy pracować niezależnie od innych programistów.

Feature zawsze wychodzi z develop.



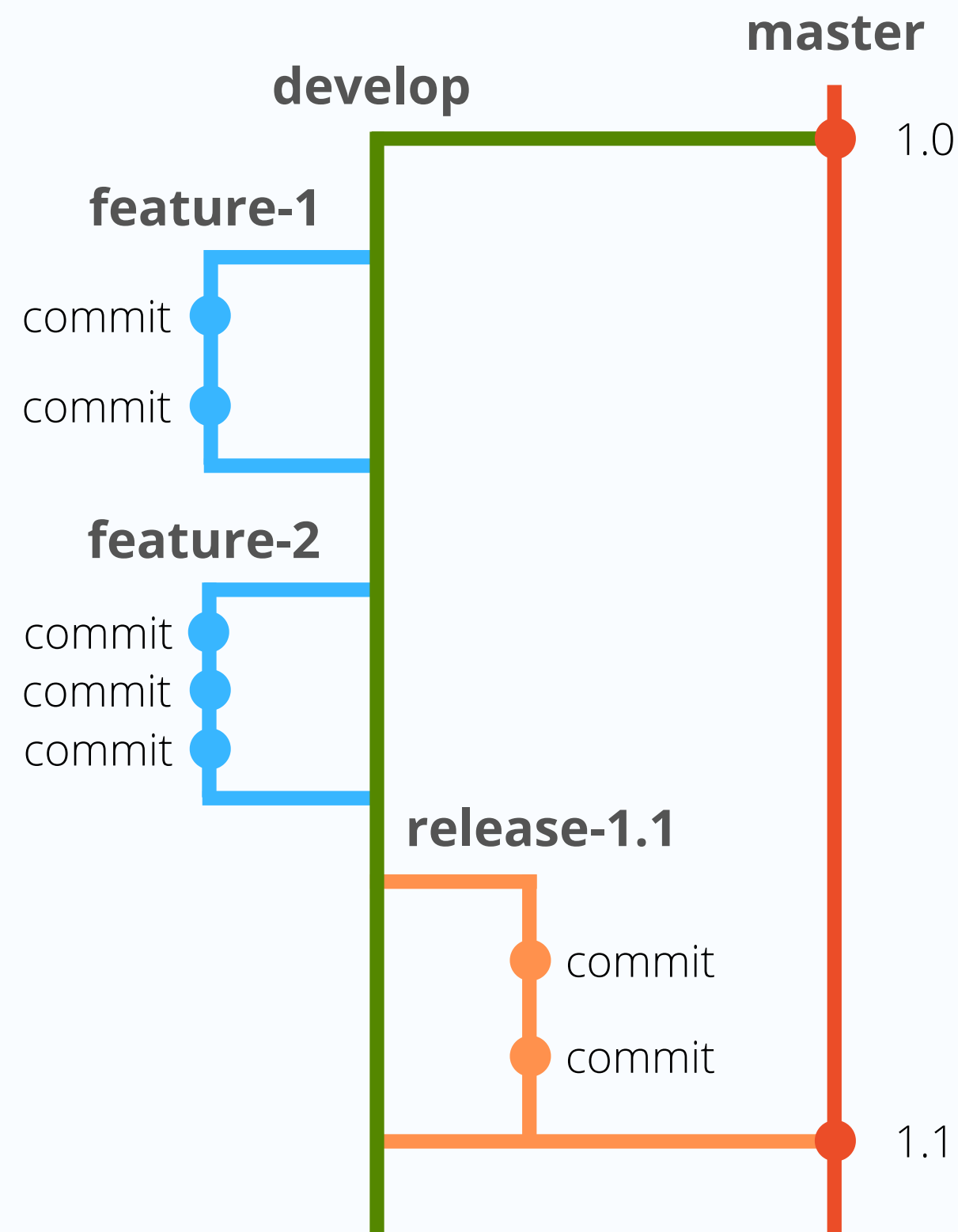
release

release - to branch wydaniowy. Na tym branchu:

- zbieramy funkcjonalności, które mają zostać wydane w następnej wersji
- testujemy funkcje po połączeniu, sprawdzamy czy działają poprawnie

Jest to ostatni branch przed dotarciem do klienta. Tutaj musimy wykryć wszystkie ewentualne problemy oraz je naprawić.

Release wychodzi z brancha develop. Dostarczany jest na master oraz develop.



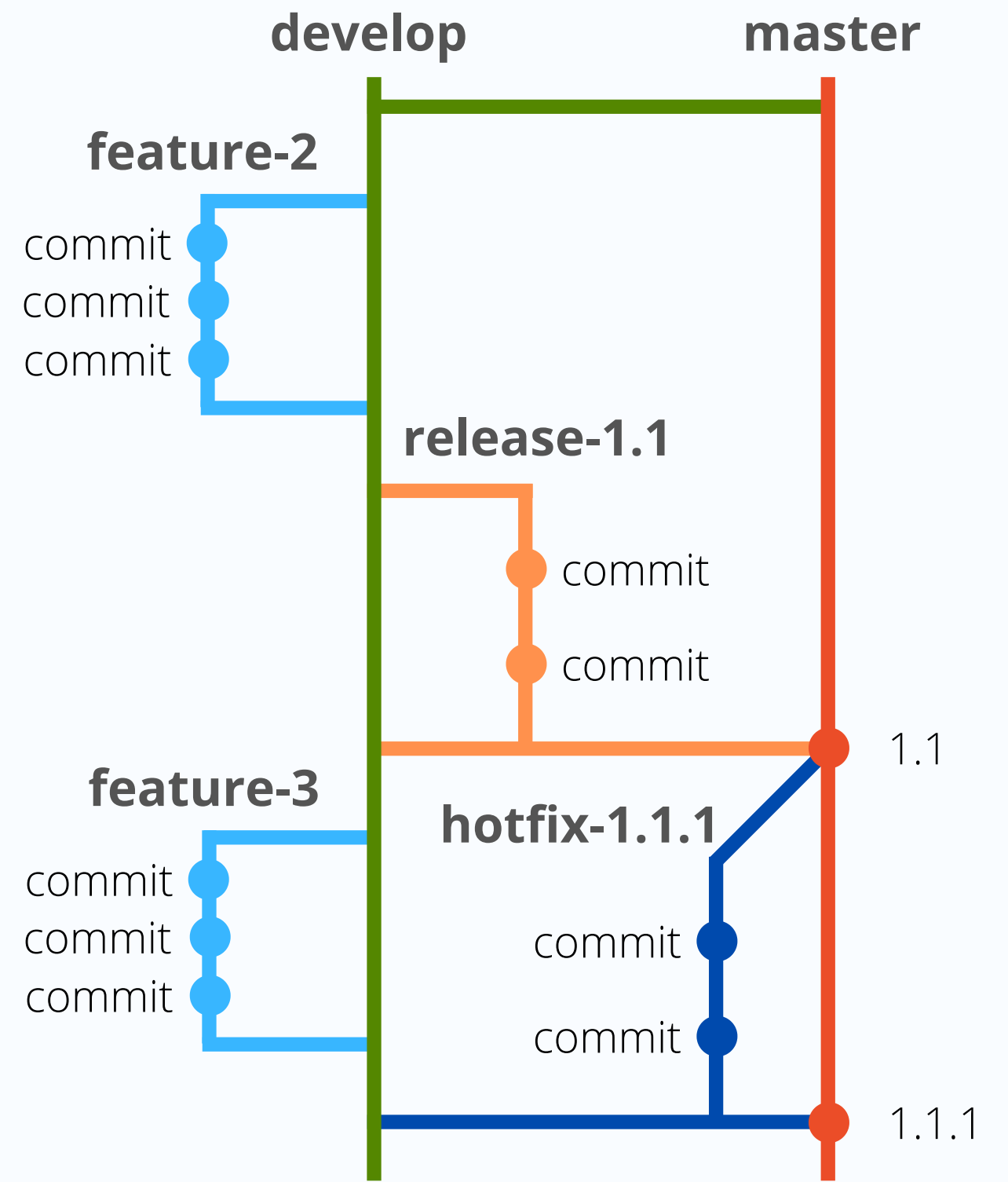
hotfix

hotfix - to branch błędów. Na tym branchu:

- naprawiamy pilne błędy, które wystąpią u klienta

Pomimo szczerych chęci bardzo często kod nie jest idealny i posiada błędy, które pojawiają się u klienta. Trzeba je szybko naprawić.

Z brancha master, z konkretnej wersji wychodzimy z hotfix. Po naprawie dostarczamy do master i develop.



Podsumowanie

Najczęściej będziesz tworzyć branche typu feature. Po ich ukończeniu dostarczysz branch do develop. Co jakiś czas wydasz wersję tworząc branch release i potem dostarczając go do mastera.

- nie commituj na master
- nie commituj na develop

Są trzy typy branchy, na których możesz commitować: **feature, release i hotfix.**

Są dwa typy branchy, do których dostarczasz: **master, develop**

Dodatkowe materiały

<https://nvie.com/posts/a-successful-git-branching-model/>

<https://nvie.com/img/git-model@2x.png>