

Jak pracować i jak żyć z modfemem na git'cie?

Zasada jest taka - wprowadzam zmiany tylko do swojej gałęzi.

W zasadzie oznacza to pracę na prywatnych gałęziach - w związku z tym będziemy **ograniczać prawa dostępu** do gałęzi innym (żeby tylko właściciel mógł dokonywać `push`).

Prawa dostępu będą ograniczane automatycznie na zasadzie schematu nazw gałęzi. Jeżeli gałąź ma prefix lub postfix w postaci inicjałów użytkownika to tylko on ma dostęp do tej gałęzi (np.: "**kb**-flow-..." albo "flow-...-**kb**").

Scenariusze:

Poniżej wersja scenariusza z konwencją pracy w kilka osób nad jedną funkcjonalnością - ale w różnych gałęziach.

CODZIENNY

1. uaktualnić wersje podstawowe (develop, ewentualnie master):

```
git checkout develop
git status
git pull
git status
```

2a. uaktualnić wersje robocze (przykład work_branch)

```
git checkout work_branch
git status
git pull
git status
```

2b. uaktualnić wersję roboczą poprawkami z develop (i master?) - ogólnie branch

```
git checkout branch
git pull
git checkout work_branch
git pull
git diff branch..work_branch
git merge --no-ff branch -m "...." (tylko jesli uznamy, że zmiany należy
dołączyć!)
git push origin work_branch (lub po prostu "git push")
```

2c. uaktualnić wersję roboczą poprawkami innych (work_branch_other jest gałęzią pracy nad tą samą funkcjonalnością innej osoby)

```
git checkout work_branch_other
git pull
git diff work_branch..work_branch_other
git checkout work_branch
git merge --no-ff work_branch_other -m "...." (tylko jeśli uznamy, że
zmiany należy dołączyć!)
git push origin work_branch (lub po prostu "git push")
```

3. praca na work_branch (lokalne modyfikacje nie zostają uwzględnione!)

4. dodać lokalne zmiany do lokalnego git-a

```
git add, git rm, git mv, np. "git add ."
git commit -a -m "...."
```

5. zapisać zmiany wersji roboczej na serwerze

```
git checkout work_branch
git push
```

ODŚWIĘTNY

1. przenieść swoje zmiany do develop (lub innej gałęzi) - ogólnie branch

```
git diff branch..work_branch
git checkout work_branch
git pull
git push (tylko jeśli chcemy własne lokalne zmiany przenieść na
serwer)
git checkout branch
git pull
git merge --no-ff work_branch
git push origin branch
git tag -a branch-work_branch -m "Merge work_branch into branch"
git push --tags
```

POCZĄTKOWY

1. utworzyć nową gałąź (new_branch):

```
git branch new_branch  
git push --set-upstream origin new_branch
```