Wszystko GIT

1) Co już wiemy nt GIT'a:

Jak sprawdzić wersję zainstalowanego GIT'a (*git --version*)
Jak skonfigurować adres e-mail i nazwę użytkownika (*git config*)
Jak stworzenie repozytorium (*git init*)
Jak sprawdzić listę plików, które będą "commitowane" (*git status*)
Jak dodać plik do "commita" (*git add*)
Jak dodawać komentarze przy wysyłaniu plików (*git commit*)
Jak tworzyć nowy link do repozytorium (*git remote add*)
Jak wysłać pliki do repozytorium (*git push*)
Jak klonowanie repozytorium (*git clone*)
Jak zobaczyć zmiany pomiędzy "commitami" (*git diff*)
Jak pobierać zmiany z serwera do lokalnego repozytorium (*git pull*)
Jak uzyskać pomoc (*git help*)

2) Czy potrzebujemy dodatkowych komend/funkcji do obsługi GIT'a?
Oczywiście, że nie, te które poznaliśmy podczas pierwszej prezentacji w zupełności powinny nam wystarczyć do indywidualnego zapotrzebowania, ale czy wystarczy to nam do pracy w grupie? Niekoniecznie.

3) W przypadku kiedy pracujemy w grupie przydadzą nam się podane możliwości GIT'a:

a) Możliwość przywrócenia pliku z dowolnego "commit'a" przy pomocy komendy

git checkout -- <nazwa commit'a>

Kiedy nam się przyda taka możliwość?
W momencie gdy przychodzimy do pracy, zapodajemy "pull'a" i stwierdzamy, że nasz plik jest np. pusty - ktoś mógł nadpisać nasz plik.

b) Działanie na gałęziach

git branch - pozwala wyświetlić gałęzie w lokalnym repozytorium
git branch <nazwa> - tworzy nową gałąź
git branch -d <nazwa> - usuwa gałąź o podanej nazwie

c) Scalanie/Migracja gałęzi do "master'a"

git merge - pozwala na scalenie dwóch lub więcej wersji (commit czy też branch) w jedno
git mergetool - narzędzie GUI; dokładnie to samo zadanie co powyższa komenda

d) “Tymczasowe zmiany”

git stash - Zapamiętuje wszystkie niezatwierdzone zmiany (wprowadza je do “schowka”) oraz przywraca stan z ostatniego commit’a

git stash apply - komenda odwrotna do powyższej - przywraca zapamiętane pliki (ze “schowka”)

git stash show - pokazuje różnice między “schowkiem” a aktualną wersją

e) Synchronizacja repozytorium - server -> local

git fetch - pozwala na synchronizację repozytorium bez scalania zmian

f) Synchronizacja repozytorium - local -> server

git push -f - pozwala na synchronizację repozytorium - nawet jeżeli zawiera błędy to zostanie zapisana

git rebase - pozwala przenieść “commit” z jednej gałęzi na drugą

g). Pozostałe

git remote - pozwala wyświetlić wszystkie repozytoria zdalne

git describe - pokazuje komentarz ostatniego “commit’a”

git log - pokazuje historię commitów

git shortlog - pokazuję listę osób “commitujących” wraz z komentarzami

git revert - wymazanie zmian z aktualnego “commit’a”

git blame - komentuję każdą linię w podanym pliku, aby widzieć kto wprowadził zmianę wraz z komentarzem “commit’a”