

Zadanie w grupach – Git integracja

Backstory

Jesteście zespołem, macie przed sobą projekt do napisania w C. Macie problem, coś do policzenia. Jakaś kosmiczna liczba iteracji, zgodnie z przewidywaniami liczyło to się będzie miesiącami. Pracodawca zleca wam odnalezienie odpowiedzi na pytanie jak przyspieszyć pracę z pętlą i którą pętlę powinniście wybrać?

Szef oczekuje raportu. Co? Gdzie? Jak? I dlaczego do takiego przypadku coś jest lepsze, a do takiego proponowalibyście inną drogę.

Tip, podczas robienia tych zadań możecie rozmawiać, nawet powinniście, w trakcie wręcz powinniście wymieniać się informacjami.

Podzielcie się na zespoły, wybierzcie dla siebie jedną **rolę** i spróbujcie zrealizować ten mini projekt

Spis treści

Zadanie w grupach – Git integracja	1
Backstory	1
ETAP 1.....	2
Researcher:.....	2
Programista:	3
DevOPS	3
ETAP 2.....	3

ETAP 1

Researcher:

- W jaki sposób można mierzyć czas wykonania funkcji w języku C, tak aby możliwe było sensowne wyświetlenie tej informacji.

-Przygotuj szablon programu umożliwiający mierzenie czasu wykonania funkcji (dla wielu funkcji) umożliwiający zapisywanie danych do pliku (nazwa funkcji,id_funkcji,najszybsze wykonanie, najwolniejsze wykonanie, czas średni(dla 10 wywołań, a potem dla 100 000)

- czy kolejność komend/procedur ma znaczenie w kontekście szybkości funkcji? Jeśli tak to jakie zaproponowałbyś rozwiązania w zależności od zastosowań?

Przypadki:

-w pętli wywołujemy funkcję wyświetlającą "Hello world!" na konsoli

- szyfrujemy ciąg znaków przy pomocy szyfru Cezara(dla nas +3, możemy wyjść poza litery w ASCII)

- możesz dodać własne

- Która pętla jest najszybsza? I dlaczego pozostałe są wolniejsze?

Korzystając z dostępnych źródeł opracuj ten temat(stwórz dostępną dla wszystkich mini-dokumentację/sprawozdanie). Korzystając ze zdobytej wiedzy, wspomagaj pozostałych członków zespołu. Twoim zadaniem jest napisanie skryptu, który przerobi dane z pliku zawierającego czasy, na tabelkę tworzoną w oparciu o język markdown, która będzie samoistnie się aktualizowała po każdym zdarzeniu „on push” w repozytorium.

Programista:

-Używając dostępnych źródeł informacji postaraj się zoptymalizować program, w taki sposób, aby zachowując jednoznaczne działanie wykonywał się szybciej i z mniejszym użyciem zasobów.

-zachowując założenia podpunktu pierwszego przygotuj program wyświetlający na konsoli „Hello World”

-zachowując założenia podpunktu pierwszego przygotuj funkcję realizującą szyfrowanie (Szyfr Cezara +3)

- Korzystając ze skryptu Researchera odpowiedz na poniższe pytania:

- Ile czasu zajmuje wykonanie 100 000.krotnie programu HelloWorld?
- Ile czasu zajmuje wykonanie 100 000.krotnie szyfrowania ciągu znaków składającego się z 5 elementów?
- Jakie są średnie czasy, najmniejszy czas oraz największy czas dla każdego z tych programów?

- Zautomatyzuj przesyłanie danych na repozytorium(i w domyśle udostępni je w taki sposób, aby Researcher miał do nich dostęp)

Korzystając ze zdobytej wiedzy, wspomagaj pozostałych członków zespołu.

DevOPS

- Porozmawiaj z pozostałymi członkami zespołu spróbuj zrozumieć ich zadania oraz przygotuj poprawną strukturę repozytorium, które będzie bazą dla ich współpracy.

- Ustal jednolite nazewnictwo plików, folderów, funkcji, klas etc. Dla wszystkich członków zespołu

- Wedle uznania ogranicz uprawnienia, komendy oraz inne wymogi dotyczące zdalnego repozytorium

- Skonfiguruj plik .gitignore (dobrze jest porozmawiać z pozostałymi osobami w zespole)

- Udostępni miejsce pracy pozostałym osobom z zespołu.

- W przypadku pojawienia się konfliktów w repozytorium to ty jesteś osobą, która podejmuje ostateczne decyzje

- Przygotuj środowisko pod automatyczne uruchamianie testów jednostkowych.

- W przypadku gdyby ktoś naruszył początkową strukturę projektu w sposób niezgodny z założeniami, napraw ten stan rzeczy.

Korzystając ze zdobytej wiedzy, wspomagaj pozostałych członków zespołu.

ETAP 2

Przechodzimy do rozmów na forum, podpytujemy tych którzy zrobili najszybszą wersję. Spróbujmy wywołać dyskusję, a jeśli to nie wyjdzie to szybki quiz po jednym dla każdego i machniemy dalej.