

Io – prezentacja

(niezależnie od czasu).

2. Różnica między git-em a github-em; Co to jest i do czego służy repozytorium; Ciągła integracja; Przykład integracji git/githuba z dowolnym IDE (Integrated Development Environment); Przykłady współpracy członków zespołu, zarządzanie repozytorium;

Ciągła integracja

Ciągła integracja (CI – ang. Continuous Integration) – podejście do tworzenia oprogramowania, polegające na tym, że członkowie zespołu często integrują swoje zmiany kodu źródłowego z wynikami prac innych członków zespołu. Zazwyczaj integracja zachodzi przynajmniej raz dziennie.

Aspekty:

- Procesy związane z testowaniem i kompilacją oprogramowania są w pełni zautomatyzowane, co eliminuje potrzebę ręcznego testowania i kompilacji kodu.
- Programiści regularnie wprowadzają swoje zmiany do wspólnego repozytorium kodu. Prowadzi to do częstego łączenia kodu i tworzenia nowych wersji oprogramowania.
- **Testy automatyczne:** Po każdym połączeniu zmian do repozytorium, wykonywane są automatyczne testy, które pomagają zidentyfikować błędy i problemy w kodzie.
- **Raportowanie:** Wyniki testów są raportowane programistom, co umożliwia szybką reakcję na wykryte błędy i problemy.
- **Izolacja błędów:** Jeśli testy wykryją błąd, programiści mogą identyfikować i naprawiać go na bieżąco, zamiast pozostawiać go nierozwiązany na dłuższy okres czasu.
- **Łatwiejsza integracja:** Ciągła integracja ułatwia integrację zmian wprowadzonych przez różnych członków zespołu, co pomaga uniknąć konfliktów i trudności związanymi z łączeniem kodu.

Zalety:

To podejście prowadzi do znacznego ograniczenia problemów z integracją oraz umożliwia zespołom programistów szybsze tworzenie spójnego oprogramowania i ułatwia zarządzanie projektami.

Do korzyści należy:

- skrócenie cyklu dostarczania oprogramowania,
- zwiększenie stabilności aplikacji,
- zwiększenie efektywności zespołu programistycznego,
- poprawę jakości kodu źródłowego.
- szybkie i efektywne dostarczanie nowych funkcji i poprawki błędów oprogramowania.

Narzędzia:

- > systemy zarządzania kodem źródłowym (śledzenie i łączenie zmian): np. GIT/GITHUB, BITBUCKET
- > systemy wspomagające budowanie, testowanie i wdrażanie (GitLab CI, Jenkins).

Współpraca członków zespołu

Współpraca jest podstawą skutecznej pracy zespołowej. Zespół, który współpracuje ze sobą osiąga dużo więcej niż poszczególni członkowie byliby w stanie osiągnąć w pojedynkę.

Przykłady współpracy:

- ➔ **Burze mózgów w grupie:** Ćwiczenie to umożliwia wszystkim członkom zespołu podzielenie się swoimi pomysłami oraz tworzenie innowacyjnych rozwiązań dla złożonych problemów, co jest korzystne dla projektu.
- ➔ **Różnorodne zespoły:** każda osoba w zespole jest inna, a każdy członek zespołu wnosi do niego inne jakości. Tworzenie zespołów o różnorodnych talentach, poziomach umiejętności i doświadczeniu (osobistym i zawodowym) wzmacnia współpracę zespołu.
- ➔ **Szczera komunikacja i otwarte dyskusje:** skuteczna współpraca w zespole wymaga zadawania pytań, zagłębiania się w szczegóły, a nawet posiadania odmiennych opinii, aby popchnąć pracę do przodu. Otwarta i szczerza komunikacja nie zawsze jest łatwa czy nawet komfortowa, ale współpraca w zespole oznacza wspólne wypracowanie rozwiązań, słuchanie opinii wszystkich osób i dążenie do osiągnięcia celów.
- ➔ **Zespołowe rozwiązywanie problemów.**