

Politechnika Śląska
Wydział Matematyki Stosowanej
Kierunek Informatyka
Studia stacjonarne I stopnia

Projekt inżynierski

Automatyzacja w procesie tworzenia oraz eksploatacji oprogramowania komputerowego

Kierujący projektem:
dr inż. Zdzisław Sroczyński

Autorzy:
Artur Kasperek
Michał Płonka
Patryk Musiol

Gliwice 2020

Projekt inżynierski:

Automatyzacja w procesie tworzenia oraz eksploatacji oprogramowania komputerowego

kierujący projektem: dr inż. Zdzisław Sroczyński

autorzy: Artur Kasperek, Michał Płonka, Patryk Musiol

Podpisy autorów pracy

.....
.....
.....

Podpis kierującego projektem

.....

Oświadczenie kierującego projektem inżynierskim

Potwierdzam, że niniejszy projekt został przygotowany pod moim kierunkiem i kwalifikuje się do przedstawienia go w postępowaniu o nadanie tytułu zawodowego: inżynier.

Data

Podpis kierującego projektem

Oświadczenie autorów

Świadomy/a odpowiedzialności karnej oświadczam, że przedkładany projekt inżynierski na temat:

Automatyzacja w procesie tworzenia oraz eksploatacji oprogramowania komputerowego

został napisany przeze mnie samodzielnie.

Jednocześnie oświadczam, że ww. projekt:

- nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (j.t. Dz.U. z 2018 r. poz. 1191, z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym, a także nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,
- nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów wyższej uczelni lub tytułów zawodowych.
- nie zawiera fragmentów dokumentów kopiowanych z innych źródeł bez wyraźnego zaznaczenia i podania źródła,
- złożona w postaci elektronicznej jest tożsama z pracą złożoną w postaci pisemnej.

Podpisy autorów pracy

Artur Kasperek, nr albumu:1234,(podpis:)

Michał Płonka, nr albumu:1234,(podpis:)

Patryk Musiol, nr albumu:1234,(podpis:)

Gliwice, dnia

Spis treści

Wstęp	7
1. Automatyzacja a branża IT	9
2. Wirtualizacja i orkiestracja	11
3. Hudson oraz Jenkins - klasyczne narzędzia do CI/CD	13
4. Platformy SaaS z wbudowanym CI/CD	15
5. Testy a continues integration	17
6. Podsumowanie	19
Literatura	21

Wstęp

TODO dodać wstęp

1. Automatyzacja a branża IT

mógłbym tu opisać ogólnie całą branżę IT oraz jak w przestrzeni lat ważny stał się CI/CD. Tutaj mógłbym umieścić trochę informacji o tym, że w projektach scrumowych szczególnie ważny jest CI/CD. Mógłbym opisać ogólnie, że dzisiaj dominują rozwiązania typu PaaS gdzie zespół nie musi już konfigurować od zera serwera CI/CD. Myślę że wstęp mógłby zawierać bardziej szczegółowe informacje o tym czym jest continues integration oraz czym jest continues delivery. Wydaje mi się, że nie ma sensu robić osobnego działu o tym czym te dwie rzeczy są, bo byłoby to lanie wody

2. Wirtualizacja i orkiestracja

myślę, że opisanie tego jak wirtualizacja pomaga w CI/CD zasługuje na swój rozdział. Szczególnie mógłbym ten rozdział poświęcić dockerowi z faktu, że zdominował rynek. Warto tutaj byłoby napisać też o rozwiązaniach które były używane w przeszłości a zostały wyparte jak vagrant. Dodatkowo w rozdziale możemy opisać na czym polega orkiestracja - Kubernetes, docker swarm

3. Hudson oraz Jenkins - klasyczne narzędzia do CI/CD

tutaj mógłbym zamieścić trochę informacji historycznych jak język java zrewolucjonował inżynierię oprogramowania i dalej rozwinać, że pokłosiem tego było powstanie Jenkinsa. Myślę, że mógłbym tutaj opisać najbardziej uniwersalne rzeczy które są w nim używane. Wydaje mi się że mógłbym poszukać alternatyw do Jenkinsa by rozszerzyć ten dział. Jako część tego rozdziału mógłbym zawrzeć problem gdzie konieczne jest użycie Jenkinsa i to byłoby zrobione jako część praktyczna. Wbrew pozorom do niektórych zastosowań Jenkins nadal jest używany, np do środowisk gdzie QA musi odpalać środowisko z różnymi parametrami

4. Platformy SaaS z wbudowanym CI/CD

tutaj mógłbym rozwinąć rozdział 3 i powiedzieć, że coraz mniej używa się Jenkinsa i obecnie popularne są platformy jak Github Actions, CircleCI czy gitlabCI, które są triggerowane przez commity, pushe czy też tworzenie tagów z pomocą systemu wersji git. Myślę, że mógłbym tutaj zrobić nawiązanie do części praktycznej pracy: 4a) continues delivery w React Native na przykładzie GitHub actions 4b) continues deployment na przykładzie aplikacji backendowej w javascriptcie 4c) continues deployment strony internetowej hostowanej przez GitHub actions za pomocą circleCI W każdym z tych podrozdziałów opisałbym jaki jest ogólny problem i dalej opisałbym już konkretne rozwiązanie

5. Testy a continues integration

mógłbym tutaj powiedzieć więcej co to są testy jednostkowe, integracyjne oraz e2e. Następnie mógłbym powiedzieć trochę więcej, że dzięki poprawnemu CI nasza aplikacja będzie miała mniej błędów oraz będzie trudniej o regresję. Tutaj mógłbym nawiązać do części praktycznej i opisać problem w którym przydają się testy e2e. Myślę, że mógłbym te testy napisać w jakimś frameworku cypress.js i skonfigurować to na GitHub actions,

6. Podsumowanie

myślę, że cały wydźwięk podsumowania powinien być nastawiony na to że dzięki CI/CD programiści mogą efektywnie działać, ich kod szybko znajduje się na wersji produkcyjnej i dzięki odpowiednio skonfigurowanemu workflow mogą być bardziej pewni, że ich zmiany nie zepsują istniejących funkcjonalności

Literatura

[1] Jakaś pozycja literatury

[2] Jakaś pozycja literatury