

Kod szkolenia: **DEVOPS**

Tytuł szkolenia: **DevOps - zwinne dostarczanie aplikacji**

Dni: 3

Opis:

Adresaci szkolenia

Szkolenie przeznaczone jest dla programistów, administratorów systemowych lub ich liderów, których zadaniem jest zapewnienie szybkiego dostarczenia przyrostu funkcjonalności na środowisko produkcyjne uwzględniając development, testy, release oraz deployment

Cel szkolenia

Szkolenie w sposób przekrojowy omawia tematykę DevOps, zarówno pod kątem zrozumienia jej idei, jak i jej praktycznej realizacji poprzez poznanie narzędzi pozwalających na automatyzację procesu wytwarzania oprogramowania, dostarczania go do instalacji na środowisku oraz tworzenia infrastruktury.

Uczestnik szkolenia po jego ukończeniu powinien:

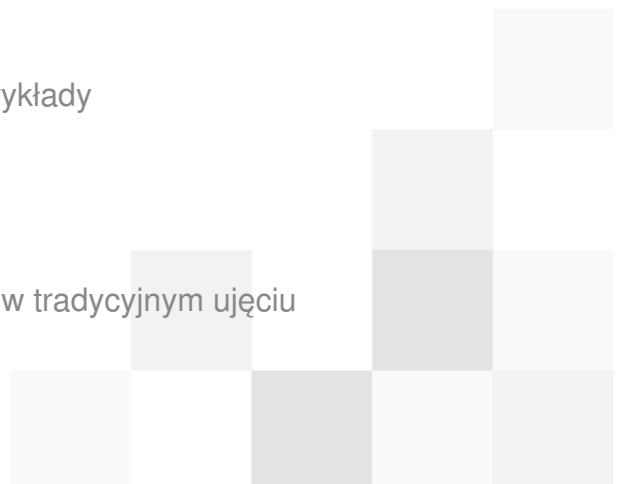
- rozumieć ideę DevOps,
- być efektywnym członkiem zespołu DevOps,
- sugerować zmiany w istniejącej organizacji, by sprzyjała powstawaniu zespołów DevOps,
- rozumieć koncepcję automatyzacji jako narzędzia podnoszącego efektywność zespołu i organizacji,
- rozumieć ideę Continuous Integration i móc ją praktycznie zastosować za pomocą narzędzia Jenkins,
- rozumieć ideę automatyzacji infrastruktury i móc ją praktycznie zastosować za pomocą narzędzi Vagrant i Puppet (alternatywnie: Chef),
- rozumieć zarządzanie infrastrukturą w chmurze na przykładzie Amazon EC2.

Parametry szkolenia

3 x 8h w proporcji: 70% warsztaty, dyskusje; 30% wykłady

Program szkolenia:

1. ITIL a DevOps.
2. Rozwój produktu z perspektywy developera w tradycyjnym ujęciu



- a. rozwój nowych produktów, funkcjonalności, bugfixów
 - b. oczekiwanie tygodniami na pojawienie się kodu na produkcji
 - c. jednoczesne utrzymywanie kodu produkcyjnego i do wdrożenia na produkcję
 - d. różnice między środowiskiem developerskim a produkcyjnym
3. Rozwój produktu z perspektywy administratora systemu w tradycyjnym ujęciu
- a. zapewnienie SLA i HA
 - b. utrzymanie rosnącej liczby środowisk ze względu na rosnącą liczbę produktów i funkcjonalności
 - c. różnice przy administracji kilkoma, kilkudziesięcioma a kilkuset środowiskami
 - d. wdrożenie nowego kodu na produkcję i jego konfiguracja, planowanie okien wdrożeniowych
 - e. diagnoza problemów w związku ze zmianą
4. Rozwój produktu z perspektywy członka zespołu DevOps (developera i administratora)
- a. wspólny cel i wspólna odpowiedzialność
 - b. automatyzacja testów
 - c. automatyzacja przepływów pracy (ang. workflows)
 - d. automatyzacja infrastruktury
 - e. ciągły pomiar funkcjonowania aplikacji
 - f. rozwój małych przyrostów produktu a szybkość i częstotliwość wdrożeń
 - g. spójność środowiska developerskiego i produkcyjnego
5. Recepta na sukces DevOps = zmiana myślenia + narzędzia
- a. Continuous Integration za pomocą Jenkins
 - i. ciągle budowanie
 - konfiguracja zadań
 - integracja z SVN / git
 - integracja z Maven
 - ii. ciągle testowanie
 - testy integracyjne z wykorzystaniem Embedded Jetty
 - testy z wykorzystaniem przegładarki na przykładzie Selenium
 - testy wydajnościowe na przykładzie JMeter
 - testy Web Service'ów na przykładzie Soap UI
 - iii. ciągła inspekcja
 - badanie pokrycia kodu testami
 - wykrywanie i prezentacja błędów z wykorzystaniem FindBugs
 - integracja z Sonar
 - b. Automatyzacja infrastruktury za pomocą Vagrant i Puppet (alternatywnie: Chef)
 - i. Vagrant
 - instalacja
 - zarządzanie maszynami wirtualnymi
 - ii. Puppet
 - instalacja
 - podstawowa konfiguracja
 - zarządzanie użytkownikami i grupami, deklaracja zależności
 - moduły

architektura master-agent
integracja Vagrant-Puppet
przykładowa konfiguracja środowiska z serwerem Apache,
Tomcat, aplikacją i bazą danych MySQL

- c. Ciągły pomiar funkcjonowania aplikacji na przykładzie Nagios
 - i. konfiguracja serwera
 - ii. konfiguracja klientów
 - iii. raporty
 - iv. powiadomienia e-mail, SMS itd.
 - v. automatyczna reakcja na wykryty błąd
- 6. DevOps w chmurze
 - a. Amazon EC2 w pigułce
 - i. założenie konta
 - ii. konfiguracja i uruchamianie maszyn wirtualnych
 - iii. przegląd możliwości platformy
 - b. Heroku w pigułce
 - i. automatyczne budowanie kodu
 - ii. automatyczny deployment
- 7. Lean w procesie wytwarzania oprogramowania
 - a. podstawy
 - i. przepływ pracy, jego wizualizacja i miara
 - ii. istota strat
 - iii. WIP
 - b. gra symulacyjna Kanban

