

DevOps

2022.09.06

Opracował: Mikołaj Ziółek
ziolk.mikosz@protonmail.com

Czym jest DevOps	3
Model C.A.M.S.	3
• Culture	3
• Automation	3
• Measurement	3
• Sharing	3
'Trzy Ścieżki DevOps'	4
Agile	5
Manifest Agile	5
Scrum	5
Kanban	6
Lean	7
Continuous Delivery & Integration & Deployment	8
Integration	8
Delivery	8
Deployment	8
Narzędzia DevOps	10
Version Control	10
Git	10
Bitbucket	10
Configuration Management	10
Chef	10
Ansible	10
Puppet	10
Continuous Integration Systems	10
Jenkins	10
Bamboo	10
TeamCity	10
BUILD	10

<i>Maven</i>	10
<i>Packer</i>	10
<i>Make or Rake</i>	10
<i>Unit Testing</i>	10
<i>Jenkins</i>	11
<i>Selenium</i>	11
<i>Integration Testing</i>	11
<i>Cucumber</i>	11
<i>Protractor</i>	11
<i>Sauce Labs</i>	11
<i>Robot</i>	11
<i>Local Testing</i>	11
<i>Vagrant</i>	11
<i>Otto</i>	11
<i>Docker Compose</i>	11
<i>Artifact Repository</i>	11
<i>Artifactory</i>	11
<i>Nexus</i>	11
<i>Docker Hub</i>	11
<i>Amazon S3</i>	11
<i>Deployment</i>	11
<i>Rundeck</i>	11
<i>UrbanCode</i>	11
<i>Deployinator</i>	12

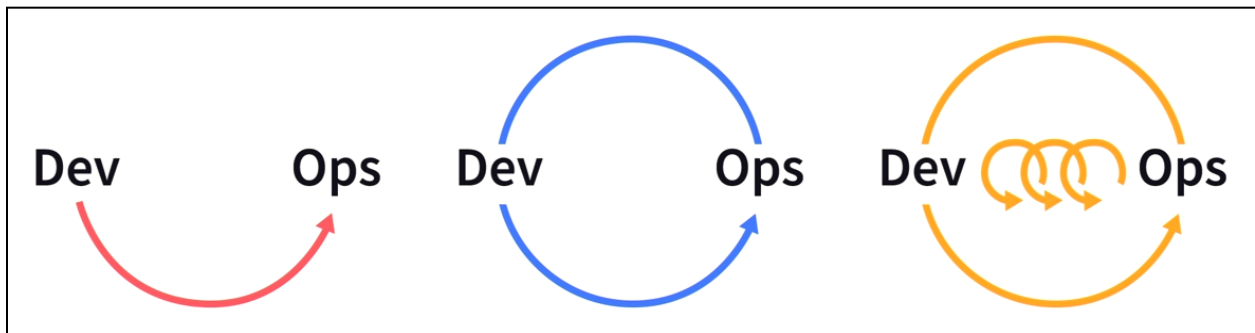
Czym jest DevOps

Albo raczej kim jest - to osoba wewnątrz projektu, która stara się ułatwić życie reszcie developerów poprzez m. in. **automatyzację procesów** przez nich wykonywanych, **nadzorowanie oprogramowania** lub samej **infrastruktury**, tworzenie pipeline'ów **CI/CD**.

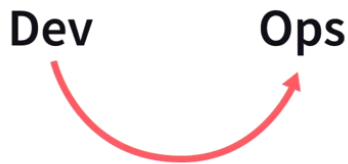
Model C.A.M.S.

- **Culture**
 - Sedno tego czym zajmuje się DevOps jest związane z ludźmi i biznesem. Pomijając wszelkie zaawansowane narzędzia, z którym pracują, głównym zadaniem jest stworzenie odpowiedniej kultury w organizacji aby zapewnić rozwiązania dobrej jakości.
- **Automation**
 - Automatyzacją nazwiemy na przykład kawałek technologii, która wykonuje dane zadanie **bez** dodatkowej ingerencji człowieka. Celem tego zabiegu jest przyspieszenie i ułatwienie powtarzalnej pracy.
- **Measurement**
 - Metryka różnych aspektów będących częścią projektu może być pomocna, pomiary wartości t.j. czas cyklu zadania, koszta, satysfakcja pracowników mogą ułatwić znalezienie lepszego punktu widzenia na potencjalne problemy, może poprawić zaangażowanie zespołu, i ustalić ogólny cel.
- **Sharing**
 - ***"Discrete continuous improvement"***
 - *Dzielenie się wiedzą, pomysłami i nawet problemami jest bardzo ważne w życiu DevOps'a. Dzięki temu ktoś może się nauczyć czegoś nowego, lub ten ktoś udzieli Ci rady odnośnie problemu, z którym się zmagasz, lub zespołowi spodoba się Twój nowy pomysł i zostanie wdrożony.*

'Trzy Ścieżki DevOps'



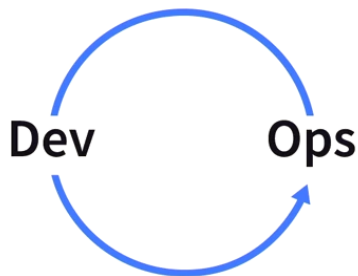
Systems Thinking



Ścieżka pierwsza mówi o tym, że na początku powinniśmy się skupić na **ogólnym wyniku** projektowego łańcucha wartości. Musimy pamiętać o tym, że optymalizacja jednego z elementów systemu, może skutkować pogorszeniem drugiego, a w końcu pogorszeniem się ogólnego wyniku.

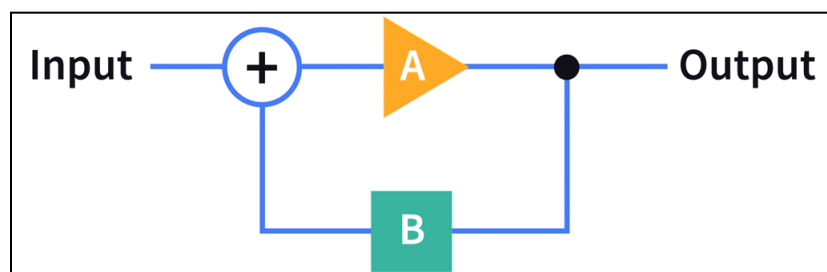
Np. Dodanie więcej serwerów do gry komputerowej, nie uwzględniając mocy przerobowej naszej bazy danych, która nie poradzi sobie z większą ilością zapytań może skutkować zawieszeniem serwera lub jego upadkiem.

Amplify Feedback Loops



Ścieżka druga nawiązuje do tworzenia, skracania i wzmacniania pętli pozyskiwania informacji zwrotnych (**feedback loop**) między częściami organizacji.

Feedback loop'em nazywamy proces, który bierze pod uwagę własne wyniki przy podejmowaniu decyzji, co dalej.



Continuous Experimentation



Ścieżka trzecia przypomina nam o tworzeniu kultury pracy, która pozwala zarówno na ciągłe eksperymentowanie, jak i uczenie się.

W tym podejściu skupiamy się na pracy nad sobą, doskonalenie umiejętności poprzez powtarzanie ćwiczeń, szukanie wyzwań i próbowanie nowych rozwiązań.

Agile

„Agile to zestaw wartości i zasad, w których procesy pracy, metody, współpraca i dostarczanie są **stale ulepszone** i **dostosowywane** do zmieniającego się kontekstu. Zespół wykonuje pracę w **małych**, ale eksploatacyjnych przyrostach”.

Manifest Agile

Osoby i interakcje ponad procesami i narzędziami

Działające oprogramowanie ponad obszerną dokumentacją

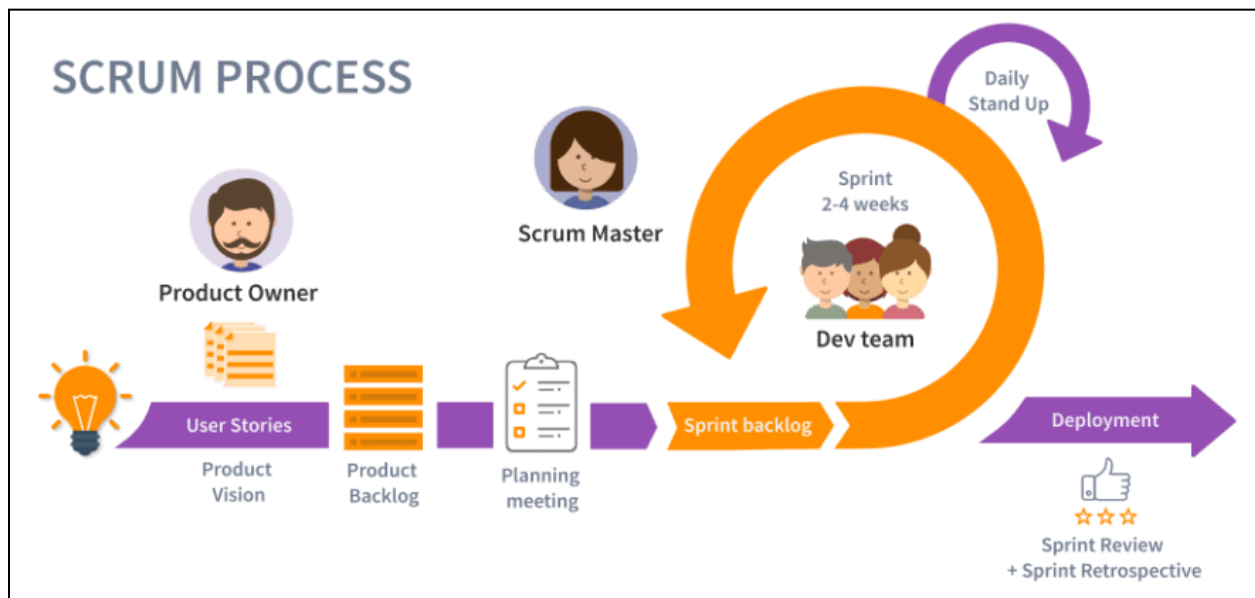
Współpraca z klientem ponad negocjacjami umowy

Reagowanie na zmianę ponad realizację planu

Oznacza to, że podczas gdy przedmioty po prawej stronie **mają wartość**, bardziej cenimy przedmioty po lewej stronie.

Scrum

Scrum to framework, który pomaga zespołom współpracować ze sobą. Często jest uważany za framework **Agile** do zarządzania projektami, który zawiera w sobie zestaw narzędzi, ról i spotkań (standups), aby pomóc zespołom w organizacji pracy i zarządzaniu nią.

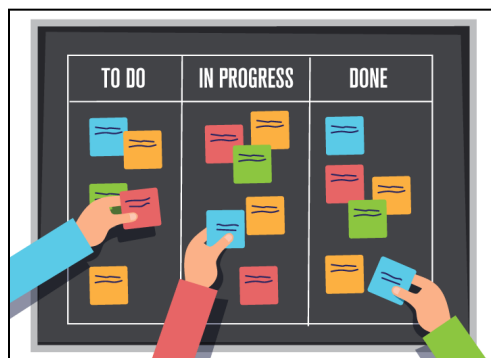


Dodatkowo Scrum zawiera w sobie **sprinty**.

Sprint to krótki, ograniczony czasowo okres, w którym zespół scrumowy pracuje nad wykonaniem określonej pracy

Kanban

Kanban jest to również framework stosowany przy podejściu Agile, wymaga komunikacji w czasie rzeczywistym i przejrzystości pracy. Elementy owej pracy są reprezentowane poprzez karteczki samoprzylepne, bądź wizualnie na tablicy Kanban w dedykowanym narzędziu. Dzięki takiemu podejściu, każdy z zespołu ma wgląd do aktualnego stanu każdego z elementów projektu.



Lean

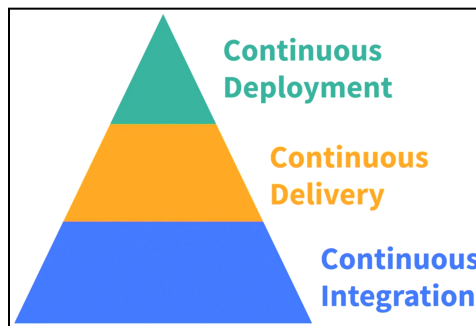
Metodologią Lean nazwiemy zbiór wiedzy, który jest rozwijany w oparciu o produkcję i skupia się na dostarczaniu lepszych produktów poprzez ciągłe doskonalenie.

- „Zrób coś, gdy ktoś tego potrzebuje, zamiast mieć nadzieję, że ktoś tego chce. Oznacza to, że produkt powinien być wykonany tylko wtedy, gdy jest na niego zapotrzebowanie.
- Poświęć trochę czasu na podniesienie jakości w okresach braku popytu, zwiększa to również produkcję i wydajność.
- Zamiast wykonywać daną pracę wielokrotnie, skup się na poprawie jakości wykonywanej pracy

Lean Development skupia się na ciągłym **podnoszeniu jakości i satysfakcji klienta**.

Różnica pomiędzy podejściem Lean a Agile polega na tym, że przy metodologii **Lean** zespoły **zwiększają szybkość zarządzając przepływem** (flow) (zwykle poprzez ograniczanie pracy w toku), podczas gdy w **Agile** zespoły **kładą nacisk na małe rozmiary partii**, aby zapewnić szybką dostawę (często w sprintach).

Continuous Delivery & Integration & Deployment



Integration

Ciągła integracja wiąże ze sobą automatyczne budowanie i testowanie całej aplikacji np.: podczas każdej aktualizacji kodu źródłowego.

Delivery

Ciągłe dostarczanie jest praktyką wdrażania wszystkich zmian na środowisku produkcyjnym (lub podobnym) oraz przeprowadzania testów automatycznych.

Deployment

Ciągłe wdrażanie polega na automatycznym rozlokowywaniu zmian, które przeszły testy.



Narzędzia DevOps

☐ Version Control

☒ **Git**

☐ GitLab

☐ GitHub

☒ **Bitbucket**

☐ Configuration Management

☐ **Chef**

☒ **Ansible**

☐ **Puppet**

☐ Continuous Integration Systems

☒ **Jenkins**

☐ **Bamboo**

☐ **TeamCity**

☐ BUILD

☒ **Maven**

☒ **Packer**

☐ **Make or Rake**

☐ Unit Testing

☒ **Jenkins**

☐ **Selenium**

☐ **Integration Testing**

☐ **Cucumber**

☐ **Protractor**

☐ **Sauce Labs**

☐ **Robot**

☐ **Local Testing**

☐ **Vagrant**

☐ **Otto**

☐ **Docker Compose**

☐ **Artifact Repository**

☒ **Artifactory**

☒ **Nexus**

☐ **Docker Hub**

☐ **Amazon S3**

☐ **Deployment**

☐ **Rundeck**

☐ **UrbanCode**

☐ Deployinator