onepress

WYDANIE II

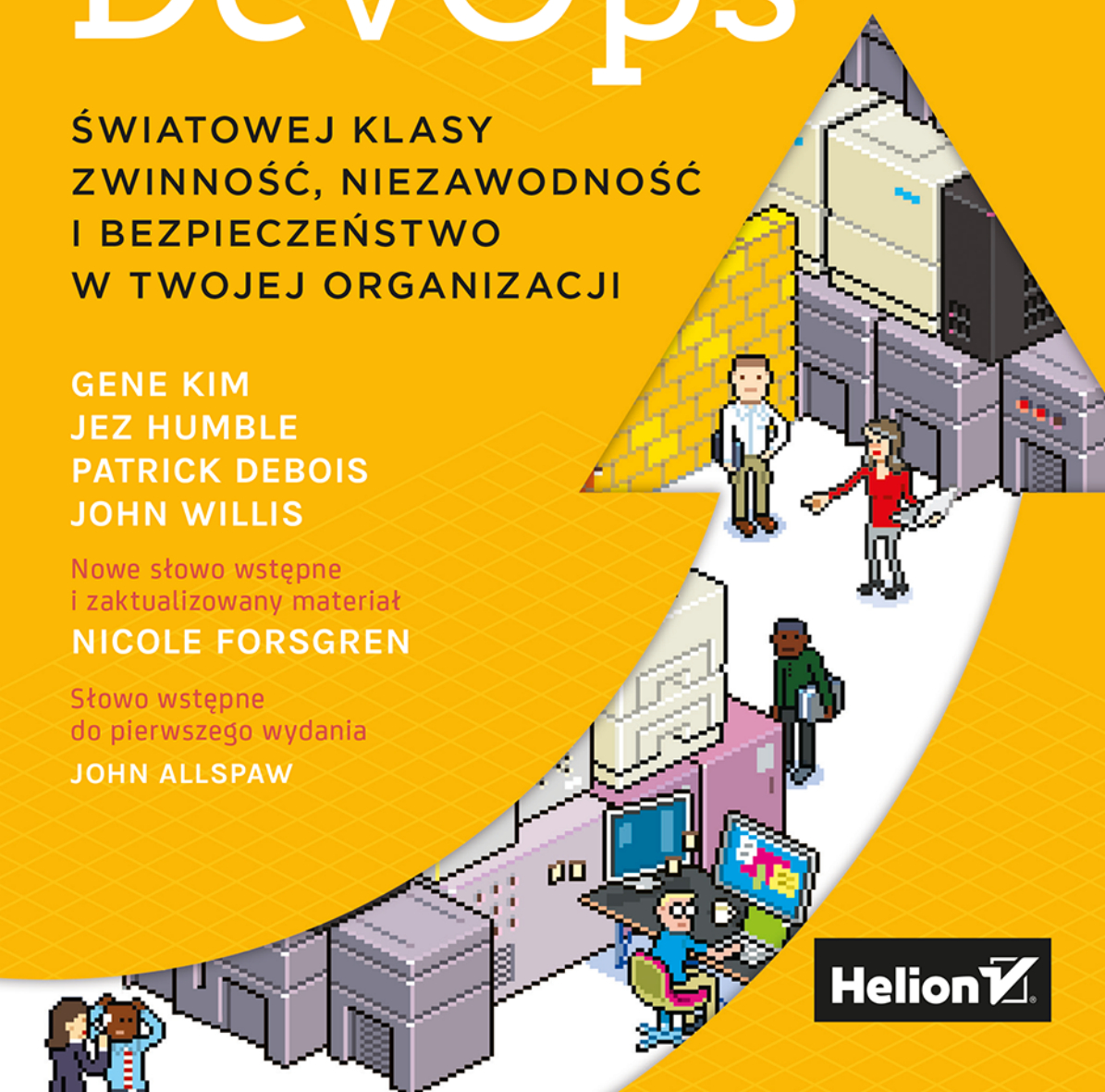
DevOps

ŚWIATOWEJ KLASY ZWINNOŚĆ, NIEZAWODNOŚĆ I BEZPIECZEŃSTWO W TWOJEJ ORGANIZACJI

GENE KIM
JEZ HUMBLE
PATRICK DEBOIS
JOHN WILLIS

Nowe słowo wstępne
i zaktualizowany materiał
NICOLE FORSGREN

Słowo wstępne
do pierwszego wydania
JOHN ALLSPAW



Helion

Tytuł oryginału: The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, 2nd Edition

Tłumaczenie: Radosław Meryk

ISBN: 978-83-283-9686-9

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/devop2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

SPIS TREŚCI

1

Agile, ciągłe dostarczanie i trzy drogi

W tym rozdziale zaprezentowano wstęp do teorii produkcji Lean, a także **trzy drogi** — zasady, na podstawie których można wywnioskować wszystkie obserwowane zachowania DevOps.

Skoncentrujemy się tutaj przede wszystkim na teorii i zasadach opisujących wiele dziesięcioleci doświadczeń firm produkcyjnych, organizacji wysokiej niezawodności, modeli zarządzania bazujących na wysokim zaufaniu i innych czynników i nurtów, z których wywodzą się praktyki DevOps. W pozostałych rozdziałach książki przedstawiono wynikowe konkretne zasady i wzorce oraz ich praktyczne zastosowanie do strumienia wartości technologii.

STRUMIEŃ WARTOŚCI PRODUKCJI

Jedną z podstawowych koncepcji Lean jest strumień wartości. Zdefiniujemy go najpierw w kontekście produkcji, a następnie dokonamy jego ekstrapolacji w celu zastosowania go do DevOps oraz strumienia wartości technologii.

Karen Martin i Mike Osterling w swojej książce *Value Stream Mapping: How to Visualize Work and Align Leadership for Organizational Transformation* zdefiniowali strumień wartości jako „sekwencję działań podejmowanych przez organizację w celu realizacji zlecenia klienta" lub „sekwencję działań wymaganych do zaprojektowania, wyprodukowania i dostarczenia towaru lub usługi do klienta z uwzględnieniem przepływów informacji i materiałów"[1].

W działalności produkcyjnej strumień wartości często jest łatwy do zaobserwowania: zaczyna się po otrzymaniu zamówienia klienta i dostarczeniu surowców do hali produkcyjnej. Aby umożliwić szybkie i przewidywalne terminy realizacji w każdym strumieniu wartości, zwykle kładzie się ciągły nacisk na tworzenie bezproblemowego i równomiernego przepływu pracy przy użyciu takich technik, jak niewielkie partie materiałów, zmniejszenie produkcji niezakończonej (ang. *Work In Process* — **WIP**), przeciwdziałanie przeróbkom, aby mieć pewność, że nie przekazujemy defektów do centrów pracy w dole strumienia, oraz ciągłe optymalizowanie systemu pod kątem osiągania globalnych celów.

STRUMIEŃ WARTOŚCI TECHNOLOGII

Te same zasady i wzorce, które umożliwiły szybki przepływ pracy w procesach fizycznych, odnoszą się również do prac nad technologią (oraz do wszystkich rodzajów pracy umysłowej). W infrastrukturze DevOps strumień wartości technologii zwykle określamy jako proces niezbędny do przekształcenia hipotezy biznesowej na korzystającą z technologii usługę, która dostarcza wartości dla klienta.

Dane wejściowe dla procesu to sformułowanie celu biznesowego, koncepcji, pomysłu lub hipotezy. Proces zaczyna się w chwili, gdy zaakceptujemy prace w rozwoju poprzez dodanie ich do zbioru zadań do wykonania (ang. *backlog*).

Od tego momentu zespoły deweloperskie stosujące typowe podejście Agile lub proces iteracyjny najczęściej przekształcają tę koncepcję na historyjki użytkowników oraz jakąś postać specyfikacji funkcji, które następnie są implementowane w kodzie aplikacji albo tworzonej usłudze. Następnie kod jest przekazywany do repozytorium kontroli wersji, gdzie każda zmiana jest testowana i integrowana z pozostałą częścią systemu oprogramowania.

Ponieważ wartość jest tworzona tylko wtedy, kiedy usługi są uruchamiane w produkcji, to musimy zadbać nie tylko o szybki przepływ, ale także o to, aby wdrożenie nie spowodowało chaosu i zakłóceń, takich jak przerwy w dostawie usług, słaba jakość usług czy też błędy w zabezpieczeniach lub zgodności z przepisami.

KONCENTRACJA NA CZASIE REALIZACJI WDRAŻANIA

W pozostałej części książki skoncentrujemy uwagę na czasie realizacji wdrożenia — podzbiorze strumienia wartości opisanego powyżej. Ten strumień wartości zaczyna się w chwili, gdy dowolny inżynier* w strumieniu wartości (który obejmuje dział rozwoju, walidacji (QA), operacji IT i bezpieczeństwa informacji) prześle zmianę do repozytorium kontroli wersji, a kończy, gdy ta zmiana zostanie pomyślnie zastosowana

* Idąc dalej, określenie „inżynier" odnosi się do wszystkich inżynierów pracujących w strumieniu wartości, nie tylko deweloperów.

w produkcji, zapewni wartość dla klienta oraz wygeneruje przydatne sprzężenie zwrotne i dane telemetryczne.

Pierwszy etap prac, który obejmuje projektowanie i rozwój, jest zbliżony do fazy rozwoju produktu w Lean (ang. *Lean Product Development*). Jest bardzo zróżnicowany i wysoce niepewny, często wymaga wysokiego stopnia kreatywności i wykonania pracy, która może nigdy nie zostać wykorzystana ponownie. W związku z tym czasy przetwarzania są na tym etapie bardzo zmienne. Dla odróżnienia drugi etap prac, który obejmuje testowanie i uruchamianie, jest zbliżony do produkcji Lean (ang. *Lean Manufacturing*). Wymaga on kreatywności i wiedzy oraz dążenia do tego, aby był przewidywalny i mechanistyczny. Celem jest uzyskanie efektów pracy przy jak najmniejszej zmienności (np. krótkie i przewidywalne czasy realizacji, prawie zerowa liczba defektów).

Zamiast dużych partii pracy przetwarzanych sekwencyjnie przez strumień wartości projektowania i rozwoju, a następnie strumień wartości testowania i uruchamiania (tak jak w dużym projekcie realizowanym według metodologii kaskadowej lub podczas rozwijania długowiecznych gałęzi funkcji) naszym celem jest realizowanie fazy testowania i uruchamiania w tym samym czasie, w którym są realizowane projektowanie i rozwój. Takie działania umożliwiają szybki przepływ i zapewniają wysoką jakość. Metoda kończy się sukcesem, gdy pracujemy niewielkimi partiami, budując jakość we wszystkich częściach strumienia wartości[*].

Definiowanie czasu realizacji a czas przetwarzania

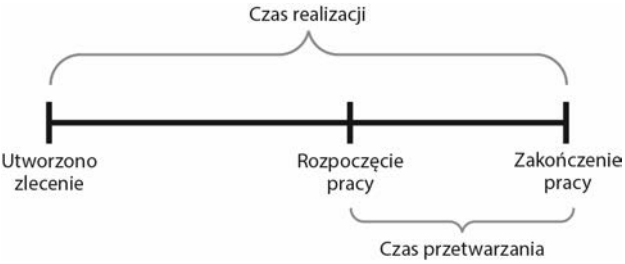
W społeczności Lean czas realizacji (ang. *lead time*) jest jednym z dwóch parametrów powszechnie stosowanych do pomiaru wydajności w strumieniu wartości. Drugim jest czas przetwarzania — ang. *processing time* (czasami znany jako *czas zadania* — ang. *task time*)[†2].

O ile czas realizacji zaczyna się w momencie złożenia zamówienia, a kończy się, gdy zamówienie zostanie zrealizowane, o tyle czas przetwarzania rozpoczyna się dopiero wtedy, gdy zaczynamy pracę nad zleceniem klienta — w szczególności pomijany jest czas, gdy zlecenie oczekuje w kolejce na przetworzenie (rysunek 1.1).

---

[*] W rzeczywistości w przypadku stosowania takich technik jak TDD testowanie jest wykonywane przed napisaniem jakiejkolwiek linijki kodu.

[†] Będziemy w tej książce faworyzowali termin „czas przetwarzania" z tych samych powodów, jakie wymienili Karen Martin i Mike Osterling: „Aby zminimalizować zamieszanie, będziemy unikali używania terminu *czas cyklu*, ponieważ istnieje dla niego kilka definicji równoznacznych, np. *czas przetwarzania* lub *częstotliwość wyników*".

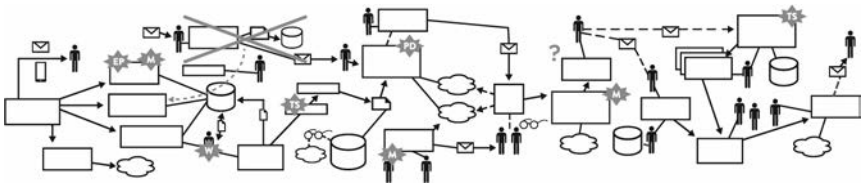


Rysunek 1.1. Czas realizacji a czas przetwarzania w operacji wdrażania

Ponieważ klient odczuwa czas realizacji, to zwykle na nim, a nie na czasie przetwarzania koncentrujemy wysiłek zmierzający do usprawniania procesu. Jednak proporcje pomiędzy czasem przetwarzania a czasem realizacji odgrywają rolę ważnego wskaźnika wydajności — uzyskanie szybkiego przepływu w krótkim czasie prawie zawsze wymaga skrócenia czasu oczekiwania pracy w kolejce.

Typowy scenariusz. Czasy realizacji wdrażania wymagające miesięcy

W biznesie czasami występują sytuacje, gdy czasy realizacji wdrożeń wymagają miesięcy. Jest to szczególnie powszechne w dużych, złożonych organizacjach, pracujących ze ściśle powiązanymi, monolitycznymi aplikacjami, często z ograniczonymi środowiskami testów integracyjnych, długimi czasami wykonania testów i wdrożeń w środowisku produkcyjnym, wysokim stopniem uzależnienia od ręcznego testowania oraz wielu wymaganych procesów zatwierdzania. W takim przypadku strumień wartości może wyglądać tak, jak pokazano na rysunku 1.2:



Rysunek 1.2. Strumień wartości technologii z czasem wdrażania wynoszącym trzy miesiące (źródło: Damon Edwards, „DevOps Kaizen", 2015)

W przypadku długich czasów realizacji wdrożeń na prawie każdym etapie strumienia wartości wymagana jest „heroiczna" praca. Może się zdarzyć, że pod koniec projektu, gdy zachodzi potrzeba scalenia zmian wszystkich zespołów rozwojowych, okaże się, że nic nie działa. W efekcie uzyskujemy kod, który się nie buduje, i nie przechodzą żadne testy. Rozwiązanie każdego problemu wymaga wielu dni lub tygodni prowadzenia dochodzenia w celu ustalenia, co spowodowało awarię i w jaki sposób trzeba ją naprawić. Dodatkowym efektem jest słaby poziom obsługi klientów.

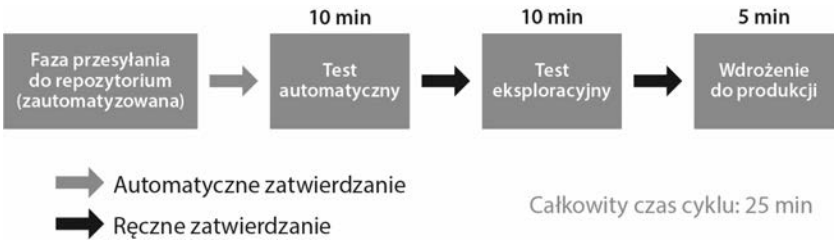
Idealne środowisko DevOps. Czas realizacji wdrażania rzędu minut

W idealnym środowisku DevOps deweloperzy otrzymują szybkie i stałe sprzężenie zwrotne dotyczące ich pracy, co pozwala im szybko i samodzielnie implementować, integrować i walidować swój kod oraz wdrażać go w środowisku produkcyjnym (samodzielnie albo przez innych).

Można to osiągnąć przez iteracyjne wprowadzanie do repozytorium kontroli wersji niewielkich zmian, wykonywanie dla nich zautomatyzowanych testów badawczych oraz wdrażanie ich do produkcji. Dzięki temu można uzyskać wysoki stopień zaufania, że wprowadzone zmiany będą działać w środowisku produkcyjnym zgodnie z przeznaczeniem, a wszelkie problemy zostaną szybko wykryte i rozwiązane.

Najłatwiej to osiągnąć, gdy mamy architekturę, która jest modułowa, dobrze zhermetyzowana i preferuje luźne powiązania. Dzięki temu niewielkie zespoły mogą pracować w warunkach wysokiego stopnia autonomii, a awarie są niewielkie, dotyczą zamkniętego fragmentu i nie powodują globalnych zakłóceń.

W takim scenariuszu czas realizacji wdrożenia jest mierzony w minutach lub w najgorszym przypadku w godzinach. Uzyskana mapa strumienia wartości powinna mieć postać podobną do przedstawionej na rysunku 1.3:



Rysunek 1.3. Strumień wartości technologii z czasem realizacji rzędu minut

OBSERWACJA WSKAŹNIKA %C/A JAKO PARAMETRU KONIECZNOŚCI WYKONYWANIA PRZERÓBEK

Trzecim kluczowym parametrem w strumieniu wartości technologii oprócz czasów realizacji i przetwarzania jest wskaźnik procentu **C/A** (ang. *complete and accurate* — dosł. „gotowe i dokładne"). Parametr ten odzwierciedla jakość wyników każdego etapu strumienia wartości.

Karen Martin i Mike Osterling opisali ten parametr w następujący sposób: „Wartość wskaźnika % C/A można uzyskać, zadając klientom dalszego etapu strumienia pytanie o procent przypadków, gdy otrzymują pracę w postaci gotowej do wykorzystania, co oznacza, że mogą oni wykonywać swoją pracę bez konieczności korygowania dostarczonych informacji, dodawania brakujących danych lub wyjaśniania sformułowań, które mogłyby i powinny być czytelniejsze"[3].

CIĄGŁE UCZENIE

Metryki przepływu pracy do pomiaru dostarczania wartości biznesowej

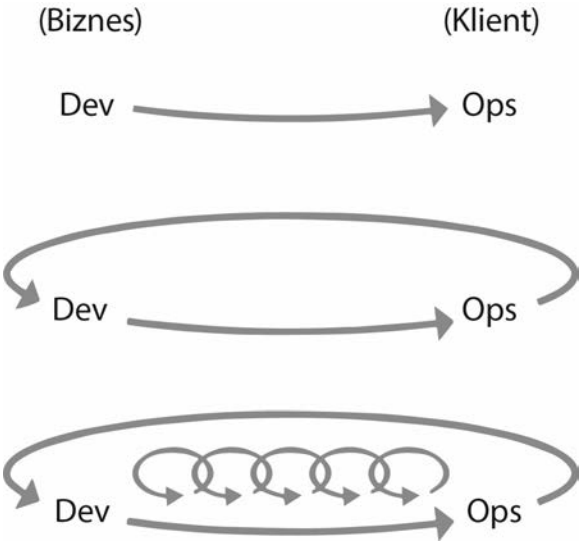
Podczas pomiaru kompleksowej wartości dowolnego strumienia wartości ważne jest unikanie metryk pośrednich (np. liczby wierszy scommitowanego kodu lub częstotliwości wdrożeń). Chociaż te wskaźniki mogą pokazywać miejsca, gdzie możliwe są lokalne optymalizacje, nie łączą się bezpośrednio z wynikami biznesowymi, takimi jak przychody.

Korzystanie z metryk przepływu pracy zapewnia wgląd w kompleksową wartość dostarczania oprogramowania, dzięki czemu produkty programowe i strumienie wartości są wyraźnie widoczne. W książce *Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework* dr Mik Kersten wymienił następujące metryki przepływu: szybkość przepływu, wydajność przepływu, czas przepływu, obciążenie przepływu i rozkład przepływu[4]:

- **Szybkość przepływu** — liczba elementów przepływu (np. elementów roboczych), które zostały ukończone w określonym czasie. Metryka ta pomaga odpowiedzieć na pytanie o to, czy dostarczanie wartości przyspiesza.
- **Wydajność przepływu** — stosunek liczby opracowanych elementów przepływu do całkowitego czasu, który upłynął. Identyfikuje nieefektywności takie jak długi czas oczekiwania i pomaga zespołom sprawdzić, czy praca w górnej części strumienia przetwarzania jest w stanie oczekiwania, czy nie.
- **Czas przepływu** — jednostka wartości biznesowej uzyskiwanej przez interesariusza za pośrednictwem strumienia wartości produktu (tzn. funkcjonalności, defekty, zagrożenia i długi). Pomaga zespołom sprawdzić, czy czas na osiągnięcie wartości skraca się.
- **Obciążenie przepływu** — liczba aktywnych lub oczekujących elementów przepływu w strumieniu wartości. Ta metryka jest podobna do tzw. pracy w toku (ang. *work in progres* — WIP), ale dotyczy elementów przepływu pracy. Wysoka wartość obciążenia przepływu prowadzi do nieefektywności i zmniejszenia szybkości przepływu lub wydłużenia czasu przepływu. Pomaga zespołom sprawdzić, czy popyt przewyższa podaż.
- **Rozkład przepływu** — proporcja każdego typu elementu przepływu w strumieniu wartości. Każdy strumień wartości, w zależności od potrzeb, pozwala śledzić i dostosowywać ten rozkład, co pozwala zmaksymalizować dostarczaną wartość biznesową.

TRZY DROGI. ZASADY LEŻĄCE U PODSTAW DEVOPS

W *Projekcie Feniks* zaprezentowano trzy drogi jako zbiór podstawowych zasad, z których wywodzą się wszystkie obserwowane zachowania i wzorce DevOps (rysunek 1.4).



Rysunek 1.4. Trzy drogi
(źródło: Gene Kim, *The Three Ways: The Principles Underpinning DevOps*, blog IT Revolution Press, 9 sierpnia 2016, *http://itrevolution.com/the-three-ways-principles-underpinning-devops/*)

Pierwsza droga umożliwia szybki przepływ pracy od lewej do prawej, od działu rozwoju, poprzez dział operacji, do klienta. Aby zmaksymalizować przepływ, należy spowodować, aby praca była widoczna, zmniejszyć rozmiary partii i interwały pracy, tworzyć jakość poprzez przeciwdziałanie przekazywania defektów do centrów pracy w dole strumienia oraz stale optymalizować pracę, by osiągać globalne cele.

Dzięki przyspieszeniu przepływu przez strumień wartości technologii możemy skrócić czas realizacji zleceń wewnętrznych lub żądań klientów — zwłaszcza czas potrzebny do wdrożenia kodu w środowisku produkcyjnym. Dzięki temu podnosimy jakość pracy, jak również przepustowość i zwiększamy szanse na pokonanie konkurencji.

Praktyki stosowane do osiągnięcia tego celu obejmują procesy ciągłego budowania, integracji, testowania i wdrażania, tworzenie środowisk na żądanie, ograniczanie pracy w toku (WIP) oraz budowanie systemów i organizacji, które zapewniają bezpieczeństwo zmian.

Druga droga umożliwia szybki i stały przepływ informacji zwrotnych od prawej do lewej na wszystkich etapach strumienia wartości. Wzmocnienie sprzężenia zwrotnego jest potrzebne do przeciwdziałania wystąpieniu podobnych problemów w przyszłości

lub wspomożenia procesu ich wykrywania oraz eliminowania skutków. W ten sposób tworzymy jakość u źródła i generujemy albo „osadzamy" wiedzę tam, gdzie jest ona potrzebna. To pozwala tworzyć coraz bezpieczniejsze systemy pracy, gdzie problemy są znajdowane i rozwiązywane wcześnie, zanim wywołają katastrofalne skutki.

Dzięki dostrzeganiu problemów natychmiast po ich wystąpieniu oraz dzięki ich gromadzeniu do czasu znalezienia skutecznych środków zaradczych stale skracamy i wzmacniamy pętle sprzężeń zwrotnych — najważniejszy mechanizm prawie wszystkich nowoczesnych metodologii usprawniania procesów. To maksymalizuje możliwości uczenia się i doskonalenia organizacji.

Trzecia droga polega na stworzeniu generatywnej kultury wysokiego zaufania, która pozwala na dynamiczne, zdyscyplinowane i naukowe podejście do eksperymentowania i podejmowania ryzyka. To wspomaga proces czerpania nauki — zarówno z sukcesów, jak i porażek. Ponadto dzięki ciągłemu skracaniu i wzmacnianiu pętli sprzężenia zwrotnego tworzymy coraz bezpieczniejsze systemy pracy. Jesteśmy lepiej przygotowani na podejmowanie ryzyka i eksperymentowanie, co pomaga uczyć się szybciej niż konkurencja i zwyciężać na rynku.

W ramach trzeciej drogi projektujemy również nasz system pracy w taki sposób, aby można było pomnożyć efekty nowej wiedzy — przekształcić lokalne odkrycia na globalne usprawnienia. Bez względu na to, gdzie ktoś wykonuje swoją pracę, robi to z wykorzystaniem coraz większego, kolektywnego doświadczenia wszystkich osób w organizacji, a także w historii organizacji.

CIĄGŁE UCZENIE

Trzy drogi — wyniki badań

Trzy drogi to nie tylko dobry pomysł — badania wykazały, że przyjęcie tych strategii prowadzi do doskonałych wyników i przynosi pożytek zarówno organizacjom, jak i ich pracownikom.

Dane uzyskane podczas trwających sześć lat badań pod przewodnictwem współautorki tej książki dr Nicole Forsgren, których efektem były wydane przez firmy Puppet i DORA raporty *State of DevOps* 2014 – 2019 i które zostały później opublikowane w książce *Accelerate: The Science of Lean and DevOps*, pokazują, że najlepsze wyniki daje połączenie możliwości i praktyk takich jak ciągła integracja, ciągłe testowanie, ciągłe wdrażanie i praca w małych partiach (pierwsza droga), szybkie informacje zwrotne i monitorowanie (druga droga) oraz kultura generatywna (trzecia droga)[5].

Trzy drogi pomagają zespołom uzyskać elitarną wydajność, dzięki czemu mogą one szybciej wydawać bardziej niezawodne oprogramowanie. Przyczyniają się również do zwiększenia przychodów, osiągnięcia lepszej pozycji na rynku i poprawy zadowolenia klientów ich organizacji. Wysokowydajne zespoły dają dwukrotnie większe szanse na osiągnięcie celów wydajności lub ich przekroczenie. Stosowanie trzech dróg poprawia również komfort pracy członków zespołu. Badania opublikowane w raportach *State of DevOps* dowodzą, że przyjęcie tych praktyk łagodzi problemy związane z wypaleniem zawodowym i zmniejsza poczucie presji związanej z wdrażaniem[6].

STUDIUM PRZYPADKU

Zbliżamy się do wysokości przelotowej — transformacja DevOps firmy American Airlines (część 1., 2020)

Inspiracją transformacji DevOps w firmie American Airlines była seria pytań, z których pierwsze brzmiało po prostu: „Co to jest DevOps?".

Maya Leibman, wiceprezes wykonawczy i dyrektor ds. informatyki w American Airlines, podczas konferencji DevOps Enterprise Summit w Londynie w 2020 r. powiedziała: „Naprawdę zaczynaliśmy od zupełnych podstaw"[7].

Najpierw zespół przeprowadził badania, ale co najważniejsze, przestał wymyślać wymówki. Początkowo większość przykładów zastosowania DevOps pochodziła od natywnych firm w branży cyfrowej, takich jak Netflix i Spotify. Zespołowi łatwo było dyskredytować ich osiągnięcia — w końcu te firmy „urodziły się w chmurze". Ale gdy na pokładzie zameldowały się bardziej tradycyjne przedsiębiorstwa, czyli firmy takie jak Target, Nordstrom i Starbucks, do członków zespołu American Airlines dotarło, że żadne wymówki nie istnieją.

Zespół rozpoczął od realizacji następującego planu:

1. Wyznaczenie konkretnych celów.
2. Sformalizowanie zbioru narzędzi.
3. Sprowadzenie trenerów i mentorów z zewnątrz.
4. Eksperymentowanie i automatyzacja.
5. Przeprowadzenie szczegółowego szkolenia praktycznego (aby uczyć się w trakcie transformacji).

Wszystkie te działania były związane z ostatecznym celem, którym było szybsze dostarczanie wartości. Jak powiedziała Leibman:

> **Wiele razy zdarzało się, że partner biznesowy z czymś przychodził, zgłaszał nowy pomysł w rodzaju „chcielibyśmy zrobić coś takiego, ale to zajmie działowi IT sześć miesięcy lub nawet rok". To mnie po prostu dobijało.**

Tak więc impulsem stojącym za transformacją w gruncie rzeczy było dążenie do rozwiązania problemu czekania na efekty w nieskończoność. Wiedzieliśmy, że istnieje lepszy sposób pracy, który pomoże nam to osiągnąć[8].

Następnie zidentyfikowano wyniki, które trzeba mierzyć[9]:

- częstotliwość wdrażania,
- czas cyklu wdrożeniowego,
- współczynnik awarii po wprowadzeniu zmian,
- czas cyklu rozwoju,
- liczba incydentów,
- średni czas do naprawy (ang. *mean time to recover* — MTTR).

Wczesne sukcesy w mapowaniu strumienia wartości pomogły członkom zespołu lepiej zrozumieć złożone procesy systemu i dały motywację. Na tych sukcesach zbudowano energię pozwalającą znaleźć nowe sposoby podejścia do rozwiązywania problemów i usprawniania procedur. Zostały przeprowadzone również kompleksowe szkolenia w dziale IT.

Początkowe sukcesy, zapoznanie się z DevOps i rozpoczęcie faktycznego stosowania niektórych zasad tej metodologii doprowadziły do postawienia drugiego poważnego pytania związanego z problemami na drodze do DevOps: „Czy dział finansów to przyjaciele, czy wrogowie?".

Stosowany proces zatwierdzania finansowania był uciążliwy i długotrwały. Cykle zatwierdzania ciągnęły się przez wiele miesięcy. „Zwykle opisywałam go jako proces, który ma na celu skłonienie cię do poddania się" — powiedziała Leibman[10].

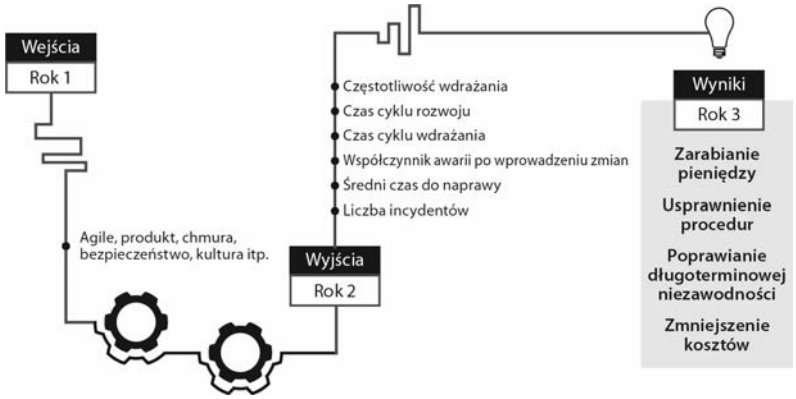
Proces wyglądał następująco[11]:

- Żaden projekt nie mógł być zatwierdzony bez udziału działu finansów.
- Projekty były zatwierdzane, ale nie zatrudniano nowych pracowników, którzy mogliby te projekty zrealizować (i nie zmieniano żadnych innych priorytetów).
- Wnioski były rozpatrywane jednakowo niezależnie od ich skali i możliwych zagrożeń.
- Były one traktowane jednakowo nawet wtedy, gdy dotyczyły najważniejszych zadań korporacyjnych i nie było wątpliwości, że zostaną zrealizowane.
- Projekty były często realizowane przed ich zatwierdzeniem.

Z konieczności zmian zdawali sobie sprawę nawet pracownicy działu finansów, ale z powodu braku zaufania między finansami a IT powstawała blokada. Aby rzucić światło na to, gdzie są wydawane pieniądze, i zbudować zaufanie do finansów, zespół podjął się mapowania kosztów: przypisał wszystkie koszty do swoich produktów, włącznie z kosztami ich utrzymania.

Na podstawie tych danych zespół IT potrafił precyzyjniej wskazać miejsca, gdzie faktycznie inwestowano pieniądze, i mógł postawić sobie pytanie, czy było to najlepsze wykorzystanie dostępnych środków. Z kolei dział finansów zyskał wymaganą przezroczystość i mógł się przekonać, że nie ma zbyt dużego marnotrawstwa.

Ta przezroczystość pozwoliła zbudować zaufanie potrzebne do eksperymentowania. Dział finansów przydzielił czterem zespołom produktowym ustalony budżet na rok. Zespoły zdefiniowały cele i kluczowe rezultaty (ang. *objectives and key results* — OKR) i wykorzystały budżet na najważniejsze priorytety, które ich zdaniem pozwoliły osiągnąć te cele. Pozwoliło to zespołom przetestować efekty przed faktycznym wdrożeniem i skupić się na odpowiedzialności i wynikach, natomiast dział finansowy zyskał jeszcze większą przezroczystość.



Rysunek 1.5. Transformacja DevOps w firmie American Airlines
Źródło: za zgodą Rossa Clantona.

Ten sukces pozwolił zespołowi skalować nowy model na wszystkie swoje produkty i zdefiniować nowy proces finansowania. „To był ogromny akcelerator w naszej transformacji" — powiedziała Leibman[12].

Dzięki przekonaniu działu finansów do transformacji i nowym procedurom firma American Airlines odkryła trzecie pytanie w swojej podróży do DevOps: „Skąd wiadomo, jaki jest wynik?". Z każdym, nawet niewielkim sukcesem zespół chciał lepiej zrozumieć, jak sobie radzi. Innymi słowy: członkowie zespołu chcieli wiedzieć, jakie osiągają wyniki.

Zespół American Airlines w pierwszym roku swojej podróży do DevOps skupiał się na wejściach: poznawaniu Agile (DevOps), produktów, chmury obliczeniowej, zabezpieczeń itp. W drugim roku zaczęto bardziej koncentrować się na wyjściach, w tym na wskaźnikach, które zaczęto mierzyć, takich jak częstotliwość wdrażania i średni czas do naprawy. Wreszcie w trzecim roku zaczęto się skupiać nie tylko na wejściach i wyjściach, ale także na wynikach (rysunek 1.5). „Ostatecznie chcieliśmy się dowiedzieć, co tak naprawdę chcemy osiągnąć" — powiedziała Leibman.

Wymyślono następujące wyniki: zarabianie pieniędzy, usprawnianie procedur, poprawianie długoterminowej niezawodności i obniżanie kosztów[13].

> **W pierwszym roku jednym z naszych celów było to, aby X% osób wzięło udział w szkoleniach Agile. To w gruncie rzeczy stanowiło nasz wkład (czyli wejście). W drugim roku, kiedy zaczęliśmy bardziej koncentrować się na wyjściach, cel zmienił się na X% zespołów, które podniosą swoją dojrzałość w zakresie zwinności z takiego poziomu do takiego. Kiedy dotarliśmy do**

trzeciego roku, zwinność zespołów nie była już naszym celem. Zdaliśmy sobie sprawę, że wejścia i wyjścia są świetne, trzeba je mierzyć, ale ostatecznie należy skoncentrować się na wynikach[14].

To doprowadziło do postawienia czwartego pytania w transformacji DevOps: „Czym jest produkt?". Było jasne, że nadszedł czas na opracowanie taksonomii. Okazało się to jednym z najtrudniejszych momentów w transformacji. Było wiele opinii i nie istniała jedna prawidłowa odpowiedź. W końcu postanowiliśmy po prostu zacząć, przelać coś na papier, zorganizować wokół tego jakiś proces i usprawnić go zgodnie z tym, czego się nauczyliśmy. To wszystko doprowadziło do piątego pytania: „Czy przeprowadzona transformacja to coś znacznie więcej niż DevOps?". Aby odpowiedzieć na to pytanie i pokazać konkretne przykłady sukcesu produktu, w dalszej części książki będziemy towarzyszyć firmie American Airlines w jej transformacji.

To studium przypadku ilustruje zastosowanie trzech dróg: wykorzystanie mapowania strumienia wartości w celu optymalizacji przepływu pracy, wybór wyników do zmierzenia w celu szybszego uzyskiwania informacji zwrotnych i stworzenie procesu szkolenia w celu budowania kultury ciągłego uczenia się i eksperymentowania.

PODSUMOWANIE

W tym rozdziale opisaliśmy pojęcia strumieni wartości oraz czasu realizacji jako jedne z kluczowych parametrów skuteczności zarówno w strumieniu wartości produkcji, jak i technologii. Zapoznaliśmy się także z wysokopoziomowymi pojęciami każdej z trzech dróg — zasad, które leżą u podstaw DevOps.

W kolejnych rozdziałach opiszemy te zasady szczegółowo. Pierwszą jest **przepływ**. Należy dążyć do stworzenia szybkiego przepływu pracy w każdym strumieniu wartości — niezależnie od tego, czy dotyczy to produkcji, czy technologii. Praktyki umożliwiające szybki przepływ zostaną opisane w części III.

DevOps. Sięgaj doskonałości, zdobywaj zaufanie, wygrywaj na rynku!

Początkowo DevOps dotyczyło branży IT i miało zapobiegać konfliktom zespołów w ramach firmy. Dziś zasady i wzorce DevOps pozwalają na przełamanie problemów, które bez interwencji powodowały opóźnienia w dostarczaniu produktów, ich niską jakość i powiększanie się długu technicznego, a także mniejszą wydajność, rosnące niezadowolenie pracowników i ich wypalenie zawodowe. Dziś DevOps jest uważane za właściwe dla każdej organizacji, która musi zwiększyć przepływ pracy technicznej przy zachowaniu jakości, niezawodności i poczucia bezpieczeństwa klientów.

To drugie, zaktualizowane wydanie podręcznika ułatwiającego przekształcanie wewnętrznych procesów w firmie zgodnie z najlepszymi praktykami DevOps. Omówiono tu podstawowe zasady DevOps, dzięki którym zwiększa się produktywność, komfort pracy pracowników, a także zwinność, czyli zdolność do reagowania na zmiany otoczenia. Nie zabrakło też wyników najnowszych badań i praktycznych doświadczeń. W tym wydaniu uwzględniono najnowsze pomocnicze dane, jak również dodatkowe zasoby, narzędzia i techniki, dzięki którym transformacja DevOps okaże się dużo prostsza. Opisano ponadto, jak można rozszerzać zasady DevOps poza dział IT i ścisłe kierownictwo firmy — tak, aby dotyczyły całej organizacji.

onepress
IT REVOLUTION
Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-283-9686-9
9 788328 396869
Cena: 99,00 zł