

Spis treści

O autorce	13
Dedykacja	13
Podziękowania	13
Przedmowa	16
Wstęp	18
CZĘŚĆ 1: CZYM JEST DEVOPS	21
ROZDZIAŁ 1: Wprowadzenie do DevOps	23
Czym jest DevOps?	23
DevOps wyewoluował z Agile'a	24
DevOps skupia się na ludziach	24
U podstaw DevOps leży kultura firmy	25
Uczysz się, obserwując swój proces i zbierając dane	25
Przekonanie jest kluczem do przyjęcia DevOps	25
Małe, przyrostowe zmiany są bezcenne	26
Korzyści płynące z DevOps	26
Model CALMS	27
Rozwiązywanie problemu sprzecznych interesów	29
ROZDZIAŁ 2: Projektowanie własnej organizacji	31
Ocena stanu zdrowia kultury	32
Wprowadzenie DevOps	33
Ustalanie wartości DevOps	34
Zachęcaj do pracy zespołowej	34
Redukuj strukturę silosową	35
Stosuj myślenie systemowe	35
Przyjmuj porażki	36
Przed wszystkim komunikacja!	36
Przyjmuj informację zwrotną	37
Automatyzuj procesy (w razie potrzeby)	37
Modelowanie kultury organizacyjnej	38
Unikanie najgorszych elementów kultury świata technologii	39
Tworzenie własnej wizji	40
Zachęcanie do wyznawania swoich wartości	42
Oceny	43
Nagrody	44

ROZDZIAŁ 3: Identyfikacja obszarów marnotrawstwa	47
Na czym polega siedem rodzajów marnotrawstwa?	48
Niepotrzebny proces	48
Czekanie	49
Zbędny ruch	49
Koszty wad	49
Nadprodukcja	49
Transport	50
Nadmierne zapasy	50
Zrozumienie pojęcia marnotrawstwa w DevOps	50
Usuwanie marnotrawstwa	52
Odkrywanie wąskich gardeł	52
Skoncentruj się na wpływie	55
 ROZDZIAŁ 4: Przekonanie współpracowników do wypróbowania DevOps	 57
Lęk przed zmianą	58
Przekonanie otoczenia do przejścia na DevOps	59
Zdobycie wsparcia kadry kierowniczej	62
Wywołanie narastającej fali zmian w grupie inżynierskiej	63
Zarządzanie menedżerami średniego szczebla	64
Przekonywanie uparciuchów	64
Zrozumienie krzywej akceptacji	65
Popychanie w kierunku zmian	67
Reagowanie na głosy sprzeciwu	69
Poruszanie się nad przepaścią	69
Pytanie „dlaczego?”	70
 ROZDZIAŁ 5: Przeprowadzenie pomiaru Twojej organizacji	 73
Mierzenie swoich postępów	74
Aspekty ilościowe w DevOps	75
Gromadzenie danych	79
Opracowanie wewnętrznych studiów przypadku	80
 CZĘŚĆ 2: STWORZENIE POTOKU	 85
 ROZDZIAŁ 6: Przyjęcie nowego cyklu życia rozwoju oprogramowania	 87
Zaproszenie wszystkich do stołu	88
Zmiana procesów: od linii do cyklu	88
Przesunięcie eksploatacji „w lewo”: myślenie o infrastrukturze	92
Przesunięcie „w lewo” również wdrożeń	93
Naśladowanie produkcji poprzez środowisko testowe	94
 ROZDZIAŁ 7: Planowanie z wyprzedzeniem	 95
Wyjście poza model Agile	96

Wyzwania związane z prognozowaniem	97
Identyfikacja wyzwań i ograniczeń projektowych	98
Gromadzenie wymagań	99
Projektowanie MVP	100
Odkrycie problemu, który ma zostać rozwiązany za sprawą Twojego MVP	102
Identyfikacja klienta	102
Obserwacja konkurencji	103
Ustalanie priorytetów w zakresie funkcji	103
Projektowanie doświadczeń użytkownika	104
Testowanie hipotezy	105
Beta czy nie beta?	106
Określenie klienta przez zaprojektowanie persony	106
Co to jest persona?	107
Projektowanie persony	107
ROZDZIAŁ 8: Projektowanie funkcji z perspektywy DevOps	109
Praca nad projektem	110
Projektowanie pod kątem DevOps	113
Projektowanie oprogramowania na potrzeby zmian	113
Ciągłe ulepszanie oprogramowania	114
Dokumentowanie oprogramowania	115
Tworzenie architektury kodu pod kątem sześciu zdolności DevOps	116
Łatwość utrzymania	116
Skalowalność	117
Bezpieczeństwo	118
Użyteczność	120
Niezawodność	121
Elastyczność	121
Dokumentowanie decyzji projektowych	122
Unikanie pułapek architektonicznych	122
ROZDZIAŁ 9: Tworzenie kodu	125
Komunikowanie się na temat kodu	125
Inżynieria w zakresie błędów	128
Pisanie kodu łatwego w utrzymaniu	128
Testowanie kodu	128
Debugowanie kodu	129
Kod logowania	129
Pisanie niezmiennego kodu	130
Tworzenie czytelnego kodu	130
Wzorce programowania	131
Programowanie obiektowe	131
Programowanie funkcyjne	131
Wybór języka	132
Unikanie antywzorców	133
Rozwój w kierunku DevOps	134

Pisanie czystego kodu	135
Zrozumienie biznesu	135
Słuchanie innych	135
Skupianie się na właściwych rzeczach	136
Pogodzenie się z brakiem komfortu	136
Ustanowienie dobrych praktyk	137
Organizowanie kodu źródłowego	137
Pisanie testów	137
Dokumentowanie funkcji	138
Przeglądanie kodu przez współpracowników	139

ROZDZIAŁ 10: Automatyzacja testów przed wydaniem **141**

Testowanie nie jest opcjonalne	141
Automatyzacja testów	142
Testowanie w różnych środowiskach	143
Środowisko lokalne	144
Środowisko deweloperskie	145
Środowisko badawcze	145
Środowisko testowe imitujące produkcyjne	146
Środowisko produkcyjne	146
Wyjście poza testy jednostkowe	147
Testy jednostkowe: To żyje!	147
Testy integracyjne: Czy wszystkie elementy współpracują ze sobą?	148
Testy regresji: Czy po zmianach kod zachowuje się tak samo?	148
Testy wizualne: Czy wszystko wygląda tak samo?	149
Testy wydajności	149
Ciągłe testowanie	149

ROZDZIAŁ 11: Wdrażanie produktu **151**

Wydanie kodu	151
Ciągła integracja i dostarczanie	152
Korzyści płynące z CI/CD	152
Wdrażanie CI/CD	153
Zarządzanie wdrożeniami	155
Właściwa automatyzacja	155
Wersjonowanie	155
Łagodzenie skutków niepowodzeń	158
Wycofanie zmian	158
Naprawa bez cofania zmian	159
Demokratyzacja wdrożeń	159
Wybór stylu wdrażania	160
Blue-green: nie tylko kolor jezior	160
Kanarek Schrödingera: wdrożenie nie umarło (a może jednak?)	162
Stopniowe wdrażanie zmian	163
Przełączanie za pomocą flag funkcji	164
Monitorowanie systemów	165
Rozumienie telemetrii	165

Zapis zachowania	166
SLA, SU oraz SLO	167

CZĘŚĆ 3: ŁĄCZENIE CYKLU W CAŁOŚĆ **169**

ROZDZIAŁ 12: Wdrażanie szybkiej iteracji **171**

Nadawanie priorytetów temu, co ważne	172
Ważne i pilne	173
Ważne, niepilne	173
Pilne, mało ważne	175
Ani ważne, ani pilne	177
Zwiększanie prędkości	177
Poprawa wydajności	180
Wyeliminowanie doskonałości	181
Projektowanie małych zespołów	181
Śledzenie swojej pracy	182
Zmniejszenie tarcia	183
Dostosowanie alertów do możliwości poznawczych ludzi	183

ROZDZIAŁ 13: Tworzenie pętli informacji zwrotnej w kontekście klienta **185**

Tworzenie procesu przekazywania informacji zwrotnych przez klientów	186
Tworzenie pętli informacji zwrotnej	187
Przyjmuj	187
Analizuj	188
Komunikuj	189
Zmieniaj	189
Zbieranie informacji zwrotnej	190
Badania satysfakcji	190
Studia przypadku	191
„Dogfooding”, czyli wypróbuj to na własnej skórze	192
Prośba o ciągłą informację zwrotną	194
Net promoter score (NPS)	194
Odnalezienie rytmu	195

ROZDZIAŁ 14: DevOps nie jest zespołem (wyjawszy sytuacje, kiedy jest) **199**

Tworzenie zespołów DevOps	200
Zharmonizowanie zespołów funkcjonalnych	200
Wyznaczanie do DevOps oddzielnego zespołu	201
Tworzenie wielofunkcyjnych zespołów produktowych	202
Przeprowadzanie szybkich (ale nie zbyt szybkich) rozmów kwalifikacyjnych	204
Podjęcie decyzji o wyborze nazwy stanowiska	205
Rekrutacja nigdy się nie kończy	207
Znalezienie właściwych ludzi	208
Przekazanie wspianiałych kandydatów	209

Ocena zdolności technicznych	209
Powrót do białej tablicy	209
Oferowanie testów domowych	210
Przegląd kodu	211
Szybkie zwalnianie	212
Arogancki palant	212
Męczennik	213
Niewydolny pracownik	213

ROZDZIAŁ 15: Wzmocnienie pozycji inżynierów **215**

Skalowanie zespołów inżynierskich za pomocą metodyki DevOps	215
Trzy etapy przedsiębiorstwa	216
Motywowanie inżynierów	220
Badania nad motywacją	221
Motywacja do podejścia DevOps	222
Niepoleganie na nagrodach zewnętrznych	222
Autonomia	223
Mistrzostwo	223
Poczucie sensu	224
Spraw, żeby praca była przyjemna	224
Umożliwienie ludziom wyboru zespołu	225
Pomiar motywacji	225

CZĘŚĆ 4: KAIZEN W PRAKTYCE JAKO SZTUKA CIĄGŁEGO DOSKONALENIA **227**

ROZDZIAŁ 16: Produktywne podejście do niepowodzeń **229**

Szybkie porażki w świecie technologii	230
Bezpieczna porażka	230
Kontrola awarii	231
Akceptowanie błędu ludzkiego (i wyeliminowanie poczucia winy)	232
Ponoszenie porażek we właściwy sposób	232
Utrzymanie postawy ukierunkowanej na rozwój	233
Tworzenie przestrzeni wolności do ponoszenia porażek	234

ROZDZIAŁ 17: Przygotowanie się na incydenty **239**

Walka z „błędem człowieka” za pomocą automatyzacji	240
Skoncentrowanie się na systemach: realistyczna automatyzacja	241
Używanie narzędzi do automatyzacji w celu uniknięcia problemów z integracją kodu	243
Obsługiwanie wdrożeń i infrastruktury	244
Ograniczanie nadmiernego projektowania	245
Naprzemienne dyżury pod telefonem na ludzkich zasadach	246
Kiedy dyżury stają się niehumanitarne	247
Ludzkie oczekiwania względem dyżurów	248
Zarządzanie incydentami	249
Uczynienie konsekwentności celem	250

Przyjęcie standardowych procesów	251
Ustalenie realistycznego budżetu	251
Ułatwienie reagowania na incydenty	252
Reagowanie na nieplanowane zakłócenia	254
Empiryczny pomiar postępu	257
Średni czas naprawy (MUR)	257
Średni czas między awariami (MTBF)	258
Koszt przypadający na incydent (CPI)	258

ROZDZIAŁ 18: Przeprowadzanie przeglądów po incydencie **261**

Wyjście poza analizę przyczyn źródłowych	262
Przechodzenie przez incydent	263
Wnioski z przeglądów	264
Zaplanuj to natychmiast	264
Zaangażowanie wszystkich pracowników	264
Brak kosztów ofiarnych	265
Przegląd osi czasu	265
Zadawanie trudnych pytań	266
Uznanie efektu pewności wstecznej	267
Sporządzanie notatek	268
Tworzenie planu	268

CZĘŚĆ 5: NARZĘDZIA DO PRAKTYKOWANIA DEVOPS **269**

ROZDZIAŁ 19: Przyjęcie nowych narzędzi **271**

Integracja z otwartym oprogramowaniem	272
Otwarcie się na innowacje w ramach społeczności	272
Licencjonowanie otwartego oprogramowania	273
Decyzja o wyborze otwartego oprogramowania	274
Przejsie na inne języki	276
Kompilowanie i interpretowanie języków	276
Paralelizacja i wielowątkowość	277
Programowanie funkcjonalne	278
Zarządzanie pamięcią	279
Roztropne wybieranie języków	280

ROZDZIAŁ 20: Zarządzanie rozproszonymi systemami **283**

Praca z monolitami i mikrouslugami	284
Stosowanie architektury monolitycznej na wczesnym etapie projektu	285
Ewolucja w kierunku mikrouslug	286
Projektowanie świetnych interfejsów API	288
Co znajduje się w API	288
Koncentracja na spójnym projekcie	289
Kontenery: oferują znacznie więcej niż maszyny wirtualne	292
Zrozumienie kontenerów i obrazów	292
Wdrażanie mikrouslug do kontenerów	293
Porównanie orkiestratorów: harmonizacja ula	295

Konfiguracja kontenerów	297
Monitorowanie kontenerów: utrzymuj je przy życiu, dopóki ich nie zabijesz	297
Zabezpieczanie kontenerów: te skrzynki potrzebują zamka	299

ROZDZIAŁ 21: Migracja do chmury 301

Automatyzacja DevOps w chmurze	301
Przeniesienie kultury DevOps do chmury	302
Uczenie się przez przyjęcie	302
Korzyści z usług w chmurze	303
Cumulus, cirrus i stal: rodzaje chmur	304
Chmura publiczna	304
Chmura prywatna	305
Chmura hybrydowa	305
Chmura jako usługa	306
Infrastruktura jako usługa	306
Platforma jako usługa	306
Oprogramowanie jako usługa	307
Wybór najlepszego dostawcy usług w chmurze	308
Amazon Web Services (AWS)	309
Microsoft Azure	309
Google Cloud Platform (GCP)	309
Znajdowanie narzędzi i usług w chmurze	310

CZĘŚĆ 6: DEKALOGI 313

ROZDZIAŁ 22: Dziesięć (plus) powodów, dla których DevOps ma znaczenie 315

Akceptacja ciągłych zmian	315
Przyjęcie chmury	316
Zatrudnianie najlepszych	316
Utrzymanie konkurencyjności	317
Rozwiązywanie ludzkich problemów	317
Stawianie wyzwań przed pracownikami	318
Niwelowanie różnic	318
Ponoszenie porażki z wdziękiem	318
Ciągłe doskonalenie	319
Automatyzacja trudu	320
Przyspieszenie dostawy	320

ROZDZIAŁ 23: Dziesięć największych pułapek DevOps 321

Nadanie kulturze niższego priorytetu	321
Pozostawienie innych w tyle	322
Zapominanie o dopasowaniu zachęt	323
Siedzenie cicho	323
Zapominanie o pomiarach	324
Mikrozarządzanie	324

Zbyt wiele zmian w zbyt krótkim czasie	325
Zły wybór narzędzi	325
Strach przed porażką	326
Bycie zbyt rygorystycznym	327

oprac. BPK