

Przedmowa Martina Fowlera (17)

Wprowadzenie (19)

Podziękowania (27)

O autorach (28)

CZĘŚĆ I. PODSTAWY (29)

Rozdział 1. Problem dostarczania oprogramowania (31)

- Wstęp (31)
- Niektóre powszechnie występujące błędne wzorce wydawania oprogramowania (32)
 - o Antywzorec: ręczne wdrażanie oprogramowania (33)
 - o Antywzorec: wdrożenie w środowisku zbliżonym do środowiska produkcyjnego dopiero po zakończeniu programowania (35)
 - o Antywzorec: ręczne zarządzanie konfiguracją środowiska produkcyjnego (37)
 - o Czy możemy to poprawić? (38)
- Jak mamy osiągnąć nasz cel? (39)
 - o Każda zmiana powinna uruchamiać proces pozyskiwania informacji zwrotnej (40)
 - o Informacja zwrotna musi być uzyskiwana możliwie szybko (41)
 - o Zespół odpowiedzialny za wdrożenie musi wyciągnąć praktyczne wnioski z otrzymanej informacji zwrotnej (43)
 - o Czy ten proces się skaluje? (43)
- Jakie płyną z tego korzyści? (44)
 - o Przyznanie zespołom większej władzy (44)
 - o Ograniczenie liczby błędów (45)
 - o Obniżenie poziomu stresu (47)
 - o Elastyczność wdrożenia (47)
 - o Ćwiczenie czyni mistrza (48)
- Kandydat do wydania (49)
 - o Każde zaewidencjonowanie prowadzi do potencjalnego wydania (50)
- Zasady dostarczania oprogramowania (50)
 - o Stwórz powtarzalny, niezawodny proces dostarczania oprogramowania (50)
 - o Automatyzuj, co tylko się da (51)
 - o Przechowuj wszystko w systemie kontroli wersji (51)
 - o Jeśli to boli, rób to częściej i szybciej zmierz się z bólem (52)
 - o Wbuduj jakość w proces wytwarzania (52)
 - o Gotowe oznacza wydane (53)
 - o Wszyscy są odpowiedzialni za udostępnianie oprogramowania (53)
 - o Ciągłe doskonalenie (54)
- Podsumowanie (54)

Rozdział 2. Zarządzanie konfiguracją (57)

- Wstęp (57)
- Stosowanie systemów kontroli wersji (58)
 - o W systemie kontroli wersji przechowuj absolutnie wszystko (59)
 - o Wprowadzaj zmiany regularnie do głównej gałęzi projektu (61)
 - o Posługuj się czytelnymi opisami zakresu zmian (62)
- Zarządzanie zależnościami (63)
 - o Zarządzanie bibliotekami zewnętrznymi (63)
 - o Zarządzanie modułami (64)

- Zarządzanie konfiguracją oprogramowania (64)
 - o Konfiguracja i elastyczność (65)
 - o Typy konfiguracji (66)
 - o Zarządzanie konfiguracją aplikacji (67)
 - o Zarządzanie konfiguracją szeregu aplikacji (70)
 - o Zasady zarządzania konfiguracją aplikacji (71)
- Zarządzanie środowiskami (72)
 - o Narzędzia do zarządzania środowiskami (75)
 - o Zarządzanie procesem zmiany (75)
- Podsumowanie (76)

Rozdział 3. Ciągła integracja (77)

- Wstęp (77)
- Wdrażanie ciągłej integracji (78)
 - o Czego potrzebujesz na początek? (78)
 - o Podstawowy system ciągłej integracji (79)
- Warunki wstępne ciągłej integracji (81)
 - o Ewidencjonuj regularnie (81)
 - o Stwórz obszerny i kompleksowy zestaw zautomatyzowanych testów (81)
 - o Niech proces kompilacji i testowania będzie możliwie krótki (82)
 - o Zarządzanie środowiskiem programistycznym (83)
- Stosowanie systemów ciągłej integracji (84)
 - o Podstawowa funkcjonalność (84)
 - o Wodotryski (85)
- Kluczowe praktyki (87)
 - o Nie ewidencjonuj niczego w popsutej kompilacji (87)
 - o Zawsze testuj lokalnie wszystkie zmiany przed ich zatwierdzeniem albo zleć to serwerowi CI (87)
 - o Zanim podejmiesz pracę, poczekaj na powodzenie testów towarzyszących przekazywaniu zmian (88)
 - o Nigdy nie idź do domu, dopóki kompilacja nie działa poprawnie (89)
 - o Zawsze bądź przygotowany na powrót do poprzednich wersji (90)
 - o Ustaw sobie limit czasu na poprawki przed cofnięciem zmian (90)
 - o Nie wyłączaj testów, które zakończyły się niepowodzeniem (91)
 - o Weź odpowiedzialność za wszystkie szkody powstałe w wyniku zmian (91)
 - o Programowanie sterowane testami (91)
- Zalecane praktyki (92)
 - o Praktyki programowania ekstremalnego (XP) (92)
 - o Odrzucanie kompilacji ze względu na naruszenie architektury (92)
 - o Odrzucanie kompilacji ze względu na powolność testów (93)
 - o Odrzucanie kompilacji ze względu na ostrzeżenia i niewłaściwe formatowania kodu (94)
- Zespoły rozproszone (95)
 - o Wpływ na proces (95)
 - o Scentralizowana ciągła integracja (96)
 - o Problemy techniczne (97)
 - o Podejścia alternatywne (97)
- Rozproszone systemy kontroli wersji (99)
- Podsumowanie (101)

Rozdział 4. Wdrożenie strategii testów (103)

- Wstęp (103)
- Typy testów (104)
 - o Testy biznesowe wspierające proces wytwarzania oprogramowania (105)
 - o Testy technologiczne wspierające programowanie (108)
 - o Testy biznesowe umożliwiające krytyczną analizę projektu (108)
 - o Testy technologiczne umożliwiające krytyczną analizę projektu (110)
 - o Obiekty zastępcze (110)
- Sytuacje i strategie z prawdziwego życia (111)
 - o Na początku projektu (111)
 - o W środku projektu (112)
 - o Kod zastany (113)
 - o Testy integracyjne (115)
- Proces (117)
 - o Zarządzanie zaległymi błędami (118)
- Podsumowanie (119)

CZĘŚĆ II. POTOK WDROŻEŃ (121)

Rozdział 5. Anatomia potoku wdrożeń (123)

- Wstęp (123)
- Czym jest potok wdrożeń? (124)
 - o Podstawowy potok wdrożeń (128)
- Praktyki związane z potokiem wdrożeń (130)
 - o Kompiluj binaria tylko raz (130)
 - o W każdym środowisku wdrażaj w taki sam sposób (132)
 - o Testuj wdrożenia testami dymnymi (134)
 - o Wdrażaj na kopii środowiska produkcyjnego (134)
 - o Każda zmiana powinna być natychmiast przekazywana do kolejnej fazy potoku (135)
 - o Jeśli jakakolwiek część potoku nie działa, zatrzymaj potok (136)
- Faza przekazywania zmian (136)
 - o Najlepsze praktyki fazy przekazywania zmian (138)
- Bramka automatycznych testów akceptacyjnych (139)
 - o Najlepsze praktyki fazy zautomatyzowanych testów akceptacyjnych (141)
- Kolejne fazy testowania (142)
 - o Testy ręczne (143)
 - o Testy нефunkcjonalne (144)
- Przygotowanie do wydania (144)
 - o Automatyzacja wdrożenia i wydania (145)
 - o Wycofywanie się ze zmian (147)
 - o Budowanie na sukcesie (148)
- Implementacja potoku wdrożeń (148)
 - o Tworzenie modelu strumienia wartości i szkieletu systemu (148)
 - o Automatyzacja procesu kompilacji i wdrażania (149)
 - o Automatyzacja testów jednostkowych i analiza kodu (150)
 - o Automatyzacja testów akceptacyjnych (151)
 - o Rozwijanie potoku (151)
- Miary (152)

- Podsumowanie (155)

Rozdział 6. Skrypty kompilacji i wdrożenia (157)

- Wstęp (157)
- Przegląd narzędzi kompilacji (158)
 - o Make (160)
 - o Ant (161)
 - o NAnt i MSBuild (162)
 - o Maven (162)
 - o Rake (163)
 - o Buildr (164)
 - o Psake (164)
- Reguły i praktyki pisania skryptów kompilacji i wdrożenia (165)
 - o Stwórz skrypt dla każdej fazy potoku wdrożeń (165)
 - o Zastosuj właściwą technologię do wdrożenia aplikacji (165)
 - o W każdym środowisku wdrażaj za pomocą tych samych skryptów (166)
 - o Skorzystaj z systemu zarządzania pakietami systemu operacyjnego (167)
 - o Zapewnij idempotentność procesu wdrożenia (168)
 - o Rozwijaj system wdrożeniowy przyrostowo (169)
- Struktura projektu dla aplikacji, których celem jest wirtualna maszyna Javy (170)
 - o Układ projektu (170)
- Tworzenie skryptów wdrożenia (173)
 - o Wdrażanie i testowanie warstw (174)
 - o Testowanie konfiguracji środowiska (175)
- Rady i wskazówki (176)
 - o Zawsze stosuj ścieżki względne (176)
 - o Wyeliminuj etapy ręczne (177)
 - o Wbuduj możliwość prześledzenia drogi od binariów do systemu kontroli wersji (177)
 - o Nie ewidencjonuj binariów w systemie kontroli wersji jako części kompilacji (178)
 - o Cele testowe nie powinny eliminować kompilacji (178)
 - o Ogranicz aplikację za pomocą zintegrowanych testów dymnych (179)
 - o Porady i wskazówki dotyczące .NET (179)
- Podsumowanie (179)

Rozdział 7. Faza przekazywania zmian (181)

- Wstęp (181)
- Zasady i praktyki fazy przekazywania zmian (182)
 - o Dostarczaj szybkiej i użytecznej informacji zwrotnej (182)
 - o Co powinno przerywać fazę przekazywania zmian? (184)
 - o Nadzoruj uważnie fazę przekazywania zmian (184)
 - o Przekaż odpowiedzialność programistom (185)
 - o W bardzo dużych zespołach przypisz komuś funkcję mistrza kompilacji (186)
- Wyniki fazy przekazywania zmian (186)
 - o Repozytorium artefaktów (186)
- Zasady i praktyki dotyczące zestawu testów fazy przekazywania zmian (189)
 - o Unikaj interfejsu użytkownika (190)

- o Stosuj wstrzykiwanie zależności (190)
- o Unikaj bazy danych (190)
- o Przy testach jednostkowych unikaj asynchroniczności (191)
- o Wykorzystywanie obiektów zastępczych (191)
- o Minimalizacja stanu w testach (194)
- o Pozorowanie czasu (195)
- o Nic na siłę (195)
- Podsumowanie (196)

Rozdział 8. Zautomatyzowane testy akceptacyjne (197)

- Wstęp (197)
- Dlaczego zautomatyzowane testy akceptacyjne są tak ważne? (198)
 - o Jak tworzyć zestawy poddających się utrzymaniu testów akceptacyjnych? (200)
 - o Testowanie graficznego interfejsu użytkownika (202)
- Tworzenie testów akceptacyjnych (203)
 - o Rola analityków i testerów (203)
 - o Analiza w projektach iteracyjnych (203)
 - o Kryteria akceptacyjne jako wykonywalne specyfikacje (204)
- Warstwa sterownika aplikacji (207)
 - o Jak wyrażać swoje kryteria akceptacyjne? (209)
 - o Wzorzec sterownika okna: uniezależnianie testów od GUI (210)
- Implementacja testów akceptacyjnych (212)
 - o Stan w testach akceptacyjnych (212)
 - o Ograniczenia procesu, hermetyzacja i testowanie (214)
 - o Zarządzanie asynchronicznością i przekroczeniem czasu przyznanego na daną operację (215)
 - o Stosowanie obiektów zastępczych (217)
- Faza testów akceptacyjnych (220)
 - o Utrzymywanie poprawności testów akceptacyjnych (221)
 - o Testy wdrożenia (223)
- Wydajność testów akceptacyjnych (225)
 - o Refaktoryzacja często wykonywanych zadań (225)
 - o Współdziel kosztowne zasoby (226)
 - o Testowanie równoległe (227)
 - o Stosowanie przetwarzania rozproszonego (227)
- Podsumowanie (229)

Rozdział 9. Testowanie wymagań niefunkcjonalnych (231)

- Wstęp (231)
- Zarządzanie wymaganiami niefunkcjonalnymi (232)
 - o Analiza wymagań niefunkcjonalnych (233)
- Programowanie z myślą o wydajności (234)
- Pomiar wydajności (236)
 - o Jak definiować sukces i porażkę w testach wydajnościowych? (238)
- Środowisko testów wydajnościowych (239)
- Automatyzacja testów wydajnościowych (243)
 - o Testowanie wydajności poprzez interfejs użytkownika (245)

- o Nagrywanie interakcji przez usługę lub publiczne API (246)
 - o Stosowanie szablonów nagranych interakcji (246)
 - o Stosowanie stubów testów wydajnościowych do produkcji testów (248)
- Dodawanie testów wydajnościowych do potoku wdrożeń (249)
- Dodatkowe korzyści płynące z systemu testów wydajnościowych (251)
- Podsumowanie (252)

Rozdział 10. Wdrażanie i wydawanie aplikacji (253)

- Wstęp (253)
- Tworzenie strategii udostępniania oprogramowania (254)
 - o Plan wydania (255)
 - o Udostępnianie produktów użytkownikom (256)
- Wdrażanie i promocja aplikacji (256)
 - o Pierwsze wdrożenie (256)
 - o Szkicowanie procesu udostępniania oprogramowania i promowania kompilacji (257)
 - o Promocja konfiguracji (260)
 - o Orkiestracja (260)
 - o Wdrożenia w środowiskach tymczasowych (261)
- Wycofywanie się z wdrożeń i wydania bez przestojów (262)
 - o Wycofywanie się poprzez powtórne wdrożenie wcześniejszej dobrej wersji (262)
 - o Wydanie bez przestoju (263)
 - o Wdrożenia niebiesko-zielone (263)
 - o Wydanie kanarkowe (265)
- Poprawki awaryjne (267)
- Ciągłe wdrażanie (268)
 - o Ciągłe udostępnianie oprogramowania instalowanego przez użytkownika (269)
- Rady i wskazówki (271)
 - o Ludzie odpowiedzialni za wdrożenie powinni być zaangażowani w tworzenie procesu wdrożenia (271)
 - o Loguj działania związane z wdrożeniem (272)
 - o Nie kasuj starych plików, tylko je przenieś (272)
 - o Za wdrożenie odpowiada cały zespół (273)
 - o Aplikacje serwerowe nie powinny mieć interfejsu graficznego (273)
 - o Przy nowym wdrożeniu pamiętaj o rozgrzewce (273)
 - o Szybko odrzucaj błędne wersje (274)
 - o Nie dokonuj zmian bezpośrednio w środowisku produkcyjnym (274)
- Podsumowanie (274)

CZĘŚĆ III. EKOSYSTEM DOSTARCZANIA OPROGRAMOWANIA (277)

Rozdział 11. Zarządzanie środowiskami i infrastrukturą (279)

- Wstęp (279)
- Rozumienie potrzeb zespołu eksploatacji systemów IT (281)
 - o Dokumentacja i audyt (282)
 - o Ostrzeżenia o nienormalnych zdarzeniach (282)
 - o Planowanie ciągłości dostarczania usług IT (283)
 - o Korzystaj z technologii znanej zespołowi eksploatacji systemów IT (284)

- Opracowywanie modelu infrastruktury i zarządzanie nią (284)
 - o Kontrola dostępu do infrastruktury (286)
 - o Wprowadzanie zmian w infrastrukturze (287)
- Zarządzanie dostarczaniem i konfiguracją serwerów (288)
 - o Dostarczanie serwerów (289)
 - o Bieżące zarządzanie serwerami (290)
- Zarządzanie konfiguracją middleware'u (295)
 - o Zarządzanie konfiguracją (295)
 - o Zbadaj produkt (297)
 - o Przeanalizuj, w jaki sposób middleware obsługuje stan (298)
 - o Poszukaj API konfiguracji (298)
 - o Zastosuj lepszą technologię (299)
- Zarządzanie usługami infrastrukturalnymi (299)
 - o Systemy wieloadresowe (300)
- Wirtualizacja (301)
 - o Zarządzanie środowiskami wirtualnymi (303)
 - o Środowiska wirtualne i potok wdrożeń (305)
 - o Wysoce równoległe testowanie ze środowiskami wirtualnymi (307)
- Przetwarzanie w chmurze (309)
 - o Infrastruktura w chmurze (310)
 - o Platformy w chmurze (311)
 - o Jedno rozwiązanie nie musi być odpowiednie dla wszystkich (312)
 - o Krytyka przetwarzania w chmurze (312)
- Monitorowanie infrastruktury i aplikacji (313)
 - o Gromadzenie danych (314)
 - o Rejestrowanie zdarzeń (315)
 - o Tworzenie tablic wskaźników (316)
 - o Monitoring sterowany zachowaniami (318)
- Podsumowanie (318)

Rozdział 12. Zarządzanie danymi (321)

- Wstęp (321)
- Pisanie skryptów baz danych (322)
 - o Inicjalizacja baz danych (322)
- Zmiana przyrostowa (323)
 - o Wersjonowanie bazy danych (323)
 - o Zarządzanie zharmonizowanymi zmianami (325)
- Wycofywanie się do poprzedniej wersji baz danych i wydania bez przestojów (326)
 - o Wycofywanie się bez utraty danych (327)
 - o Uniezależnianie wdrożenia aplikacji od migracji bazy danych (328)
- Zarządzanie danymi testowymi (329)
 - o Imitowanie bazy danych na potrzeby testów jednostkowych (330)
 - o Zarządzanie zależnościami między testami a danymi (331)
 - o Izolacja testu (331)
 - o Przygotowanie i rozmontowanie (332)
 - o Spójne scenariusze testowe (332)
- Zarządzanie danymi i potok wdrożeń (333)
 - o Dane w fazie przekazywania zmian (333)
 - o Dane w testach akceptacyjnych (334)

- o Dane w testach wydajnościowych (335)
 - o Dane w innych fazach testów (336)
- Podsumowanie (337)

Rozdział 13. Zarządzanie modułami i zależnościami (339)

- Wstęp (339)
- Utrzymywanie aplikacji w stanie zdolności do wydania (340)
 - o Ukryj nową funkcjonalność, dopóki nie zostanie ukończona (341)
 - o Wprowadzaj wszystkie zmiany przyrostowo (343)
 - o Rozgałęzianie przez abstrakcję (343)
- Zależności (345)
 - o Piekło zależności (346)
 - o Zarządzanie bibliotekami (347)
- Moduły (349)
 - o Jak dzielić bazę kodu na moduły? (349)
 - o Droga modułów przez potok wdrożeń (352)
 - o Potok integracyjny (353)
- Zarządzanie schematem zależności (355)
 - o Tworzenie schematów zależności (355)
 - o Potokowanie schematów zależności (357)
 - o Kiedy powinniśmy wyzwać kompilację? (360)
 - o Ostrożny optymizm (360)
 - o Zależności cykliczne (362)
- Zarządzanie binariami (363)
 - o Jak powinno działać repozytorium artefaktów? (363)
 - o W jaki sposób potok wdrożeń powinien współdziałać z repozytorium artefaktów? (364)
- Zarządzanie zależnościami za pomocą Mavena (365)
 - o Refaktoryzacja zależności Mavena (367)
- Podsumowanie (368)

Rozdział 14. Zaawansowana kontrola wersji (369)

- Wstęp (369)
- Krótka historia kontroli wersji (370)
 - o CVS (370)
 - o Subversion (371)
 - o Komercyjne systemy kontroli wersji (373)
 - o Wyłącz pesymistyczne blokowanie (373)
- Rozgałęzianie i scalanie (375)
 - o Scalanie (376)
 - o Gałęzie, strumienie i ciągła integracja (377)
- Rozproszone systemy kontroli wersji (380)
 - o Czym jest rozproszony system kontroli wersji? (380)
 - o Krótka historia rozproszonego systemu kontroli wersji (382)
 - o Rozproszone systemy kontroli wersji w środowiskach korporacyjnych (382)
 - o Korzystanie z rozproszonych systemów kontroli wersji (383)
- Strumieniowe systemy kontroli wersji (385)
 - o Czym są strumieniowe systemy kontroli wersji? (385)

- o Modele wytwarzania oprogramowania z wykorzystaniem strumieni (387)
 - o Widoki statyczne i dynamiczne (389)
 - o Ciągła integracja z systemami kontroli wersji opartymi na strumieniach (389)
- Programuj na gałęzi głównej projektu (390)
 - o Dokonywanie złożonych zmian bez rozgałęziania (391)
- Gałąź na potrzeby wydania (393)
- Rozgałęzienia według kryterium funkcji (394)
- Rozgałęzianie pod kątem zespołu (397)
- Podsumowanie (400)

Rozdział 15. Zarządzanie ciągłym dostarczaniem oprogramowania (403)

- Wstęp (403)
- Model dojrzałości zarządzania konfiguracją i wydaniem (405)
 - o Jak posługiwać się modelem dojrzałości (405)
- Cykl życia projektu (407)
 - o Identyfikacja (408)
 - o Zapoczątkowywanie (409)
 - o Inicjalizacja (410)
 - o Wytwarzanie i wdrażanie (411)
 - o Eksploatacja (414)
- Proces zarządzania ryzykiem (414)
 - o Podstawy zarządzania ryzykiem (415)
 - o Harmonogram zarządzania ryzykiem (415)
 - o Jak wykonać ćwiczenie z zakresu zarządzania ryzykiem? (416)
- Częste problemy z dostarczaniem oprogramowania - objawy i przyczyny (417)
 - o Rzadkie lub wadliwe wdrożenia (418)
 - o Kiepska jakość aplikacji (418)
 - o Kiepsko zarządzany proces ciągłej integracji (420)
 - o Słabe zarządzanie konfiguracją (420)
- Zgodność z regulacjami i audyt (421)
 - o Przewaga automatyzacji nad dokumentacją (422)
 - o Narzucanie możliwości śledzenia zmian (422)
 - o Praca w silosach (423)
 - o Zarządzanie zmianą (424)
- Podsumowanie (425)

Bibliografia (427)

Skorowidz (429)