# Podstawy inżynierii oprogramowania

*Od kodera do inżyniera*

*Nathaniel Schutta i Dan Vega*

*przekład: Marek Włodarz*

APN Promise
Warszawa 2026

O'REILLY

# Spis treści

Przedmowa . . . . . xiii

Podziękowania . . . . . xix

## Część I. Podstawowe umiejętności

**1. Od programisty do inżyniera . . . . . 3**

Inżynier pod (dowolną) inną nazwą . . . . . 3
Podstawy są ważne . . . . . 4
Wiele dróg, aby zostać inżynierem oprogramowania . . . . . 5
Czego cię nauczono, a co potrzebujesz wiedzieć . . . . . 6
Etos leniwego programisty . . . . . 7
Wartość świeżego spojrzenia . . . . . 9
Nie rozwiązuj zbyt szybko . . . . . 10
Zastosuj złotą regułę do oprogramowania . . . . . 13
Podsumowanie . . . . . 13
W praktyce . . . . . 14
Dodatkowe materiały . . . . . 14

**2. Czytanie kodu . . . . . 17**

Wyzwania stawiane przez istniejący kod . . . . . 17
Błędy poznawcze . . . . . 19
Podchodząc do nieznanego kodu . . . . . 20
Archeologia programistyczna . . . . . 21
Skuteczne strategie czytania kodu . . . . . 26
Wykorzystywanie funkcji IDE . . . . . 27
Czytanie testów . . . . . 32
Ćwiczenie czyni mistrza . . . . . 34
Podsumowanie . . . . . 35
W praktyce . . . . . 36
Dodatkowe źródła . . . . . 36

**3. Pisanie kodu . . . . . 37**
Nie wynajduj koła od nowa . . . . . 38
Czym jest dobry kod? . . . . . 39
Mniej znaczy więcej . . . . . 41
Zerowe prawo informatyki . . . . . 42
Unikaj kodu szablonowego . . . . . 42
Preferuj kompozycję, nie dziedziczenie . . . . . 42
Preferuj krótkie metody . . . . . 43
Pisz kod, który ma być czytany . . . . . 43
Problem komentarzy do kodu . . . . . 45
Testy jako dokumentacja . . . . . 46
Unikaj (nazbyt) sprytnego kodu . . . . . 47
Przeglądy kodu . . . . . 48
Unikanie „przeglądów" poprzez odfajkowanie . . . . . 48
Nikt nie lubi być krytykowany . . . . . 49
Wykuwanie zaufania . . . . . 50
Nauka nowych języków . . . . . 50
Podsumowanie . . . . . 52
W praktyce . . . . . 52
Dodatkowe źródła . . . . . 53

## Część II. Praktyki techniczne

**4. Modelowanie . . . . . 57**
Czym jest modelowanie oprogramowania i po co je wykonujemy? . . . . . 57
Jakich diagramów potrzebujemy? . . . . . 60
Diagramy kontekstowe . . . . . 61
Diagramy komponentów . . . . . 62
Diagramy klas . . . . . 63
Diagramy sekwencji . . . . . 63
Diagramy wdrożenia . . . . . 65
Modele danych . . . . . 67
Dodatkowe diagramy . . . . . 67
Najlepsze praktyki modelowania . . . . . 68
Zachowaj prostotę . . . . . 68
Znaj swoich odbiorców . . . . . 69
Ostrożnie z kolorami . . . . . 70
Wprowadź standardy i szablony . . . . . 70

Narzędzia . . . . . 70
Podsumowanie . . . . . 73
W praktyce . . . . . 73
Dodatkowe źrodła . . . . . 73

**5. Zautomatyzowane testowanie . . . . . 75**

Zalety automatycznych testów . . . . . 75
Działają jako dokumentacja . . . . . 76
Ulepszają utrzymywanie . . . . . 76
Wzmacniają pewność siebie . . . . . 77
Zapewniają spójność i powtarzalność . . . . . 78
Typy testów zautomatyzowanych . . . . . 79
Testy jednostkowe . . . . . 80
Testy integracyjne . . . . . 80
Testy całościowe . . . . . 80
Jaką kombinację testów trzeba napisać? . . . . . 81
Czego nie należy testować . . . . . 81
Pokrycie kodu . . . . . 82
Pisanie testów . . . . . 83
Zaczynamy . . . . . 83
Asercje . . . . . 84
Pisanie testów jednostkowych . . . . . 84
Imitowanie . . . . . 86
Pisanie testów integracyjnych . . . . . 87
Pisanie testów całościowych . . . . . 88
Podsumowanie . . . . . 90
W praktyce . . . . . 90
Dodatkowe źródła . . . . . 92

**6. Badanie i modyfikowanie nieznanych systemów . . . . . 93**

Poznawanie nieznanych baz kodu . . . . . 93
Zacznij od szerszego spojrzenia . . . . . 94
Zrozumieć projekt . . . . . 94
Przegląd dostępnej dokumentacji . . . . . 94
Poznawanie przepływu wykonania . . . . . 97
Stopniowe budowanie modeli mentalnych . . . . . 108
Przykładowy proces . . . . . 112
Bezpieczne wykonywanie zmian . . . . . 113
Bezpieczna refaktoryzacja . . . . . 113
Reguła skauta . . . . . 115

Małe, odwracalne zmiany . . . 116
Podsumowanie . . . 121
W praktyce . . . 121
Dodatkowe źródła . . . 122

## Część III. Projektowanie i tworzenie aplikacji

**7. Projektowanie interfejsu użytkownika . . . 125**
Projektowanie dla każdego . . . 125
Czym jest użytkowalność? . . . 126
Czym jest dostępność? . . . 128
Czym są lokalizacja i internacjonalizacja? . . . 129
Znaj swoich użytkowników . . . 130
Dodatkowi użytkownicy . . . 131
Nie jesteś swoim użytkownikiem . . . 132
Wpływ kultury . . . 133
Maksymalizowanie użytkowalności . . . 134
Zasady projektowania . . . 135
Kontrast . . . 136
Powtarzanie . . . 136
Wyrównanie . . . 137
Sąsiedztwo . . . 137
Stosowanie zasad projektowania . . . 139
Spraw, aby właściwe działanie było działaniem oczywistym . . . 142
Dobre komunikaty błędów . . . 147
Działania destrukcyjne . . . 148
Podsumowanie . . . 151
W praktyce . . . 151
Dodatkowe źródła . . . 151

**8. Praca z danymi . . . 153**
Typy i formaty danych . . . 154
Dane strukturalne kontra niestrukturalne . . . 154
Powszechnie używane formaty danych . . . 157
Specjalistyczne uwarunkowania danych . . . 160
Efektywne przechowywanie danych . . . 163
Typy baz danych i ich przypadki użycia . . . 163
Utrwalanie i zarządzanie danymi . . . 169
Połączenia z bazą danych i transakcje . . . 173

Modele spójności i strategie buforowania ... 174
Planowanie pod kątem wzrostu danych ... 181
Zapytania i wydajność danych ... 182
Pisanie wydajnych zapytań ... 183
Podstawowa optymalizacja zapytań ... 183
Zapytania parametryzowane ... 184
Zarządzanie indeksami ... 185
Obsługa wielkich zbiorów wyników ... 186
Narzędzia i najlepsze praktyki ... 187
Migracje i transformacje danych ... 193
Podstawy przenoszenia danych ... 193
Obsługa zmian schematu ... 195
Podsumowanie ... 198
W praktyce ... 199
Dodatkowe źródła ... 200

**9. Architektura oprogramowania ... 201**

Czym jest architektura? ... 201
Kompromisy ... 203
Architektura kontra projektowanie ... 205
Atrybuty jakościowe ... 206
Identyfikowanie atrybutów jakościowych ... 207
Uzgadnianie stanowisk z interesariuszami ... 208
Style architektoniczne ... 210
Zwinny architekt ... 210
Funkcje dopasowania ... 212
Diagramy architektoniczne ... 213
Rejestr decyzji architektonicznych ... 214
Podsumowane ... 216
W praktyce ... 216
Dodatkowe źródła ... 217

**10. Do produkcji ... 219**

Złożoność środowisk produkcyjnych ... 219
Użytkownicy są nieprzewidywalni ... 220
„Ale u mnie działa” ... 221
Budowanie kodu gotowego do produkcji ... 224
Optymalizacja wydajności ... 224
Konfiguracje dostosowane do środowiska ... 226
Obsługa błędów i rejestrowanie ... 229

Podstawy zabezpieczeń . . . 231
Potok wdrażania . . . 238
Środowiska wdrożeniowe . . . 239
Strategie kontroli wersji . . . 240
Automatyzacja wdrażania . . . 245
Strategie wdrażania . . . 248
Ciągła integracja i ciągłe wdrażanie . . . 250
Monitorowanie i utrzymywanie systemu produkcyjnego . . . 254
Monitorowanie . . . 254
Konserwacja systemu . . . 255
Podsumowanie . . . 257
W praktyce . . . 258
Dodatkowe źródła . . . 259

## Część IV. Rozwój osobisty

**11. Zwiększanie produktywności . . . 263**
Optymalizacja środowiska deweloperskiego . . . 263
Znaj swoje narzędzia . . . 263
Zbuduj swój własny miecz świetlny . . . 265
Wykorzystaj potęgę wiersza polecenia . . . 266
Wykorzystaj moc skrótów klawiszowych . . . 268
Strategiczna automatyzacja . . . 271
Ciągłe dążenie do produktywnych nawyków . . . 271
Wspólna nauka . . . 271
Osobiste zarządzanie wiedzą . . . 272
Podsumowanie . . . 274
W praktyce . . . 275
Dodatkowe źródła . . . 275

**12. Nauka uczenia się . . . 277**
Wkuwanie nie działa . . . 278
Nabywanie umiejętności . . . 279
Nawyk uczenia się . . . 280
Uczenie się z pomocą AI . . . 283
Lęk przed pominięciem czegoś . . . 284
Gdzie powinieneś zainwestować swój czas? . . . 286
Praktyka innowacji . . . 288
Briefingi architektoniczne . . . 290

Bądź życzliwy dla siebie. . . . . 291
Podsumowanie . . . . . 292
W praktyce . . . . . 292
Dodatkowe źródła . . . . . 293

**13. Kompetencje miękkie w świecie techniki . . . . . 295**

Współpracująca komunikacja . . . . . 296
Kanały komunikacyjne . . . . . 296
Firmowy głuchy telefon . . . . . 303
Znaj swoich odbiorców . . . . . 305
Ćwiczenie wpływu . . . . . 305
Rozumienie i przedstawianie wartości . . . . . 306
Strategiczne podejście do wpływu. . . . . 306
Zarządzanie interesariuszami. . . . . 308
Zarządzanie czasem . . . . . 309
Harmonogram twórcy. . . . . 310
Pozostawanie przy zadaniu. . . . . 311
Podsumowanie . . . . . 312
W praktyce . . . . . 312
Dodatkowe źródła . . . . . 312

**14. Zarządzanie karierą. . . . . 313**

Zaplanuj swoją ścieżkę kariery. . . . . 313
Odkryj, czym się pasjonujesz. . . . . 314
Eksplorowanie różnych dziedzin i technologii . . . . . 314
Eksperymenty z projektami pobocznymi. . . . . 314
Eksplorowanie opcji kariery. . . . . 316
Ustalanie celów pośrednich . . . . . 320
Świadome zdobywanie umiejętności . . . . . 321
Dostosowywanie wyborów zawodowych do etapów życia . . . . . 324
Droga po ścieżce kariery. . . . . 327
Celebruj i rejestruj swoje zwycięstwa . . . . . 327
Radzenie sobie z syndromem oszusta. . . . . 328
Budowanie własnej społeczności profesjonalnej. . . . . 330
Pielęgnowanie relacji zawodowych . . . . . 333
Wygrać na kolejnej rozmowie kwalifikacyjnej . . . . . 335
Równowaga pomiędzy pracą a życiem . . . . . 342
Podsumowanie . . . . . 346
W praktyce . . . . . 347
Dodatkowe źródła . . . . . 348

**15. Inżynier oprogramowania ze wsparciem AI. . . . . 349**
Czym naprawdę jest sztuczna inteligencja? . . . . . 350
Demistyfikacjia terminologii AI . . . . . 350
Możliwości i ograniczenia AI. . . . . 356
AI jako partner w programowaniu . . . . . 359
Autonomiczny asystent (chatbot) . . . . . 360
Asystent inline w IDE . . . . . 360
Agentowe środowiska AI . . . . . 362
Podstawy inżynierii promptów . . . . . 363
Czym jest inżynieria promptów?. . . . . 364
Podstawowe techniki inżynierii promptów . . . . . 364
Zaawansowane techniki inżynierii promptów. . . . . 367
Jak AI może przekształcić inżynierię oprogramowania . . . . . 369
Czy AI zabierze mi pracę? . . . . . 369
Przeglądy kodu „na wibracjach” . . . . . 371
AI jako mnożnik siły: od pisania kodu po rozwiązywania problemów . . . . . 374
Podsumowanie . . . . . 375
W praktyce . . . . . 376
Dodatkowe źródła . . . . . 377

Indeks . . . . . 379

O autorach . . . . . 389