

Spis treści

Wstęp	11
Przedmowa	13
1. Wprowadzenie do testowania full stack	19
Testowanie full stack w celu uzyskania wysokiej jakości	21
Testowanie shift-left	22
Dziesięć umiejętności testowania full stack	25
Kluczowe wnioski	29
2. Ręczne testowanie eksploracyjne	30
Bloki konstrukcyjne	32
Schematy stosowane w testach eksploracyjnych	32
Odkrywanie funkcjonalności	40
Strategia ręcznego testowania eksploracyjnego	43
Zrozumienie aplikacji	43
Eksploracja częściami	46
Powtarzanie testów eksploracyjnych w wielu fazach	46
Ćwiczenia	48
Testowanie API	48
Testowanie interfejsu webowego	54
Perspektywy: higiena środowiska testowego	60
Kluczowe wnioski	62
3. Automatyczne testowanie funkcjonalne	64
Bloki konstrukcyjne	66
Wprowadzenie w tematykę testów typu mikro i makro	66
Strategia automatycznego testowania funkcjonalnego	71
Ćwiczenia	73
Testy funkcjonalne na bazie interfejsu użytkownika	74
Testy usług	90
Testy jednostkowe	94
Dodatkowe narzędzia do testowania	97
Pact	97
Karate	101
Narzędzia AI/ML w automatycznych testach funkcjonalnych	102
Perspektywy	104
Antywzorce do wyeliminowania	104
Stuprocentowe pokrycie testami automatycznymi!	105
Kluczowe wnioski	107

4. Ciągłe testowanie	108
Bloki konstrukcyjne	109
Wprowadzenie w tematykę ciągłej integracji	109
Procesy ciągłej integracji, ciągłego wdrażania i ciągłego testowania	110
Zasady i etykieta	114
Strategia ciągłego testowania	115
Korzyści	119
Ćwiczenie	120
Git	121
Jenkins	123
Cztery kluczowe wskaźniki	127
 5. Testowanie danych	 130
Bloki konstrukcyjne	131
Bazy danych	133
Pamięci podręczne	137
Systemy przetwarzania wsadowego	138
Strumienie zdarzeń	139
Strategia testowania danych	141
Ćwiczenia	143
SQL	143
JDBC	148
Apache Kafka i Zerocode	151
Dodatkowe narzędzia do testowania	158
Kontenery testowe	158
Deequ	159
Kluczowe wnioski	161
 6. Testowanie wizualne	 162
Bloki konstrukcyjne	163
Wprowadzenie w tematykę testów wizualnych	163
Przypadki użycia o kluczowym znaczeniu dla projektu i jego wymagań biznesowych	164
Strategia testowania frontendu	167
Testy jednostkowe	167
Testy integracyjne (na poziomie komponentów)	168
Testy migawkowe	169
Funkcjonalne testy „od-końca-do-końca”	170
Testy wizualne	171
Testowanie w wielu przeglądarkach	171
Testowanie wydajności frontendu	173
Testowanie dostępności	173
Ćwiczenia	173
BackstopJS	174
Cypress	178
Dodatkowe narzędzia do testowania	181
Applitools Eyes — narzędzie oparte na sztucznej inteligencji	181
Storybook	182

Perspektywy: wyzwania związane z testowaniem wizualnym	183
Kluczowe wnioski	184
7. Testowanie zabezpieczeń	185
Bloki konstrukcyjne	187
Popularne rodzaje cyberataków	188
Model zagrożenia STRIDE	191
Luki w zabezpieczeniach aplikacji	193
Modelowanie zagrożeń	196
Strategia testowania zabezpieczeń	203
Ćwiczenia	206
OWASP Dependency-Check	206
OWASP ZAP	207
Dodatkowe narzędzia do testowania	213
Wtyczka Snyk IDE	214
Hak pre-commit Talisman	214
Chrome DevTools i Postman	215
Perspektywy: bezpieczeństwo to nawyk	216
Kluczowe wnioski	217
8. Testy wydajności	218
Bloki konstrukcyjne testowania wydajności backendu	218
Wydajność, sprzedaż i wolne weekendy są ze sobą powiązane!	219
Proste cele wydajności	220
Czynniki wpływające na wydajność aplikacji	220
Kluczowe wskaźniki wydajności	222
Rodzaje testów wydajności	223
Rodzaje wzorców obciążenia	225
Etapy testowania wydajności	226
Ćwiczenia	230
Krok 1. Zdefiniuj docelowe wskaźniki KPI	230
Krok 2. Zdefiniuj przypadki testowe	231
Kroki 3-5. Przygotuj dane, środowisko i narzędzia	232
Krok 6. Utwórz skrypty przypadków testowych i uruchom je z wykorzystaniem narzędzia JMeter	233
Dodatkowe narzędzia do testowania	240
Gatling	240
Apache Benchmark	241
Bloki konstrukcyjne testów wydajności frontendu	242
Czynniki wpływające na wydajność frontendu	243
Model RAIL	244
Metryki wydajności frontendu	245
Ćwiczenia	246
WebPageTest	247
Lighthouse	250
Dodatkowe narzędzia do testowania	252
PageSpeed Insights	252
Wtyczka DevTools przeglądarki Chrome	253

Strategia testowania wydajności	254
Najważniejsze wnioski	256
9. Testowanie dostępności	257
Bloki konstrukcyjne	258
Sylwetki użytkowników związanych z ułatwieniami dostępu	258
Ekosystem ułatwień dostępu	260
Przykład: czytniki ekranu	261
WCAG 2.0: zasady przewodnie i poziomy	262
Standardy zgodności poziomu A	263
Frameworki programistyczne z obsługą ułatwień dostępu	266
Strategia testowania dostępności	266
Lista kontrolna elementów dostępności w historyjkach użytkowników	266
Narzędzia do automatycznej inspekcji ułatwień dostępu	268
Testowanie ręczne	268
Ćwiczenia	270
WAVE	270
Lighthouse	272
Moduł Node wtyczki Lighthouse	275
Dodatkowe narzędzia testowe	276
Moduł Node PA11y CI	276
Axe-core	276
Perspektywy: dostępność jako kultura	277
Najważniejsze wnioski	277
10. Testowanie wymagań wielofunkcyjnych	279
Bloki konstrukcyjne	280
Strategia testowania wymagań CFR	282
Funkcjonalność	283
Wygoda użytkowania	284
Niezawodność	286
Wydajność	287
Możliwości wsparcia	287
Inne metody testowania wymagań CFR	288
Inżynieria chaosu	288
Testowanie architektury	292
Testowanie infrastruktury	293
Testowanie zgodności z przepisami	296
Perspektywy: możliwości ewolucji i próba czasu!	299
Najważniejsze wnioski	300
11. Testowanie mobilne	301
Bloki konstrukcyjne	302
Wprowadzenie do krajobrazu mobilnego	302
Architektura aplikacji mobilnych	307
Strategia testowania mobilnego	308
Ręczne testowanie eksploracyjne	310
Funkcjonalne testy automatyczne	311

Testowanie danych	311
Testy wizualne	312
Testy zabezpieczeń	312
Testy wydajności	313
Testowanie dostępności	314
Testy wymagań CFR	315
Ćwiczenia	316
Appium	317
Wtyczka Appium Visual Testing	323
Dodatkowe narzędzia testowe	326
Database Inspector w środowisku Android Studio	326
Narzędzia do testowania wydajności	326
Narzędzia do testowania zabezpieczeń	329
Accessibility Scanner	330
Perspektywy: piramida testowania aplikacji mobilnych	331
Najważniejsze wnioski	332
12. Nie tylko testowanie	333
Najważniejsze zasady testowania	333
Zapobieganie defektom zamiast ich wykrywania	333
Testy empatyczne	335
Testy na poziomie mikro i makro	335
Szybkie informacje zwrotne	336
Ciągłe sprzężenie zwrotne	337
Mierzenie wskaźników jakości	337
Kluczem do jakości są komunikacja i współpraca	339
Umiejętności miękkie pomagają w budowaniu nastawienia na jakość	340
Podsumowanie	342
13. Wprowadzenie do testowania w nowych technologiach	343
Sztuczna inteligencja i uczenie maszynowe	344
Wprowadzenie do uczenia maszynowego	344
Testowanie aplikacji ML	346
Blockchain	348
Wprowadzenie do pojęć związanych z blockchainem	349
Testowanie aplikacji Blockchain	351
Internet rzeczy	353
Wprowadzenie do pięciowarstwowej architektury IoT	354
Testowanie aplikacji IoT	355
Rzeczywistość rozszerzona i wirtualna	357
Testowanie aplikacji AR i VR	357
Skorowidz	359