

Renu Rajani

# Testowanie kodu w praktyce

Helion 

Packt 

Tytuł oryginału: Testing Practitioner Handbook

Tłumaczenie: Maksymilian Gutowski

ISBN: 978-83-283-3970-5

Copyright © Packt Publishing 2017

First published in the English language under the title Testing Practitioner Handbook - (9781788299541)

Polish edition copyright © 2018 by Helion SA

All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>O autorce</b>	<b>15</b>
<b>Podziękowania</b>	<b>16</b>
<b>O recenzencie</b>	<b>17</b>
<b>Zespół wydania oryginalnego</b>	<b>18</b>
<b>Wstęp</b>	<b>19</b>
Zakres tematyczny książki	19
Materiały dodatkowe	23
Do kogo ta książka jest skierowana	23
Konwencje	23
<b>Rozdział 1. Stan transformacji cyfrowej — co się zmieniło w latach 2013 – 2016?</b>	<b>25</b>
Koncentracja na wydajności i efektywności	25
Priorytety transformacji QA i testowania	26
Czynniki transformacyjne QA i testowania	26
Poziom dojrzałości cyfrowej	27
Wydatki na cyfrowe QA	27
Prognozy rozwoju aplikacji chmurowych	27
Przyrost wyzwań towarzyszących testom mobilnym i wielokanałowym IoT	28
Ograniczenie wyzwań towarzyszących testowaniu wrażeń klienta	29
Podsumowanie	30

<b>Rozdział 2. Przyszłość modeli testowych</b>	
<b>— czy prognozy wzrostu wydatków na QA są uzasadnione?</b>	<b>31</b>
<b>Podział prac nad QA</b>	<b>32</b>
Podział prac nad QA na rozwój i wsparcie produkcji	32
Intensyfikacja działań QA w zakresie rozwoju — opinia	32
Intensyfikacja działań QA w fazie projektowania	33
Intensyfikacja działań QA w fazie projektowania — opinia	33
<b>Prowadzenie (scentralizowanych i zdecentralizowanych) hybrydowych zespołów QA</b>	<b>34</b>
Prowadzenie zespołów hybrydowych QA — opinia	34
<b>Przyszłość modeli przedsięwzięć testowych</b>	<b>35</b>
Przyszłość modeli przedsięwzięć testowych — opinia	35
<b>DevOps</b>	<b>37</b>
<b>Podsumowanie</b>	<b>37</b>
<b>Źródła</b>	<b>38</b>
<b>Rozdział 3. Korzyści z zastąpienia podwykonawców testowych zarządzanymi usługami testowymi</b>	<b>39</b>
<b>Dane z World Quality Report 2016 dotyczące industrializacji i TCoE</b>	<b>39</b>
<b>Korzystanie z usług bezpośrednich podwykonawców jest powszechne</b>	<b>40</b>
<b>Największe wyzwania towarzyszące pracy z podwykonawcami</b>	<b>41</b>
<b>Korzyści z zastąpienia podwykonawców testowych zarządzanymi usługami testowymi</b>	<b>42</b>
Zachowanie wiedzy	42
Odbiór	42
Wydajność kosztowa	42
Ekonomia skali	42
Opis korzyści	43
Kalkulator oszczędności	43
<b>Podsumowanie</b>	<b>44</b>
<b>Rozdział 4. Model fabryczny cyfrowego QA</b>	<b>45</b>
<b>Fabryki usług</b>	<b>45</b>
<b>Cyfrowy model fabryki w industrializacji dostawy cyfrowego QA</b>	<b>46</b>
<b>Kluczowe aspekty korzystania z DQAF</b>	<b>47</b>
Funkcje wspierające DQAF	47
Dostarczanie usług DQAF	48
<b>Korzyści płynące z modelu DQAF</b>	<b>48</b>
<b>Podsumowanie</b>	<b>49</b>
<b>Rozdział 5. Crowdsourcing — elastyczne TCoE na żądanie</b>	<b>51</b>
<b>Testy crowdsourcingowe — trendy</b>	<b>52</b>
<b>Czym jest crowdtesting i jak to działa?</b>	<b>52</b>
<b>Modele operacyjne crowdtestingu</b>	<b>53</b>
<b>Crowdtesting w ramach TCoE</b>	<b>54</b>
Kluczowe wyzwania związane z wprowadzeniem crowdtestingu	54

Korzyści z crowdtestingu	55
Podsumowanie	55
Źródła	55
<b>Rozdział 6. Testowanie przebiega sprawniej w weekendy</b>	<b>57</b>
Testowanie weekendowe — modus operandi	57
Zalety	58
Wady	59
Trendy w stosowaniu testów crowdsourcingowych	60
W jaki sposób duże organizacje IT mogą wdrożyć program testów weekendowych?	61
Podsumowanie	61
Źródła	62
<b>Rozdział 7. Testowanie w produkcji zwinnej i poziom akceptacji podejścia zwinnego</b>	<b>63</b>
Jakie korzyści oferuje zwinność w porównaniu z modelem waterfall?	64
Rodzaje metodyk zwinnych	65
Scrum	65
Kanban	66
Jak przebiega testowanie w zwinnych sprintach?	66
Zwinność w środowiskach rozproszonych	68
Stan akceptacji zwinności — ustalenia World Quality Report 2016 – 2017	69
Wyzwania w stosowaniu metodologii agile	69
Podejścia do testowania w środowiskach zwinnych	69
Umiejętności QA i testowe potrzebne w ramach zwinności	70
Podsumowanie	71
<b>Rozdział 8. Wzrost poziomu akceptacji metodyki zwinnej i DevOps</b>	<b>73</b>
Zwiększone wykorzystanie podejścia zwinnego i DevOps	74
Rosnąca złożoność testowania w metodologii zwinnej	74
Wyzwania towarzyszące przygotowywaniu CoE na potrzeby testów zwinnych	75
W jaki sposób QA przekształca się, aby sprostać wymogom podejścia zwinnego i DevOps	76
Czym się różni testowanie w ramach DevOps od tradycyjnego?	77
Podsumowanie	78
<b>Rozdział 9. Czy rosnąca popularność DevOps podważa pozycję metodyki zwinnej?</b>	<b>79</b>
Podejście zwinne polega na szybkości	80
DevOps usuwa granice dzielące rozwój, QA i operację	81
Podejście zwinne kontra DevOps	81
Podsumowanie	82
<b>Rozdział 10. Rola automatyzacji w DevOps</b>	<b>83</b>
Znaczenie automatyzacji dla DevOps	84
Wczesna automatyzacja integracji	84
Automatyzacja — domyślna praktyka	86
Automatyzacja wdrażania	88

<b>Automatyzacja — wskaźniki i pomiar</b>	<b>88</b>
Zakres automatyzacji	88
Indeks automatyzacji	88
Poziom wielokrotnego użytku modelu i skryptów	89
Wzrost produktywności w wyniku automatyzacji	89
ROI automatyzacji	89
Śledzenie postępów automatyzacji	89
<b>Najlepsze praktyki wczesnego i ciągłego automatyzowania</b>	<b>90</b>
Uznanie automatyzacji za zwyczajną pracę rozwojową	90
Inżynieria jakości	90
Współpraca zespołów projektowych, testowych i automatyzacyjnych	90
Wybór właściwego rozwiązania	91
<b>Podsumowanie</b>	<b>91</b>
<b>Rozdział 11. Ocena stanu wdrożenia DevOps przy użyciu metody benchmarkingowej</b>	<b>93</b>
<b>Dlaczego DevOps — czynniki sprzyjające</b>	<b>94</b>
<b>Wyzwania towarzyszące wdrożeniu DevOps</b>	<b>94</b>
<b>DevOps QA benchmarking — czym jest i do czego służy</b>	<b>95</b>
<b>Obszary benchmarkingu DevOps QA</b>	<b>96</b>
Wynik oceny benchmarkingowej DevOps QA	97
<b>Rozdział 12. Usprawnienie DevOps — ChatOps</b>	<b>99</b>
<b>Trendy DevOps według WQR</b>	<b>99</b>
<b>ChatOps — historia i potrzeby</b>	<b>100</b>
<b>ChatOps — jak to działa?</b>	<b>101</b>
<b>Zastosowanie ChatOps jako platformy współpracy</b>	<b>102</b>
<b>Korzyści z ChatOps</b>	<b>102</b>
<b>Podsumowanie</b>	<b>103</b>
<b>Rozdział 13. BDD z wykorzystaniem języka Gherkin w środowisku zwinnym i DevOps</b>	<b>105</b>
<b>BDD — założenia</b>	<b>105</b>
<b>Gherkin — podstawowa składnia i przykład</b>	<b>106</b>
<b>Role w zespole</b>	<b>107</b>
<b>Korzyści z używania Gherkina</b>	<b>108</b>
<b>Podsumowanie</b>	<b>108</b>
<b>Rozdział 14. Automatyzacja zarządzania konfiguracją dla środowisk testowych DevOps</b>	<b>109</b>
<b>Wprowadzenie</b>	<b>110</b>
<b>Rodzaje środowisk testowych</b>	<b>110</b>
Usługi zarządzania konfiguracją i środowiskiem w chmurze i związane z nimi wyzwania	111
<b>Potencjalne rozwiązania kwestii automatycznego zarządzania konfiguracją</b>	<b>112</b>
<b>Korzyści z automatyzacji zarządzania konfiguracją</b>	<b>113</b>
<b>Podsumowanie</b>	<b>113</b>

<b>Rozdział 15. Zautomatyzowane zarządzanie danymi testowymi w środowisku DevOps</b>	<b>115</b>
Wprowadzenie	116
TDM w środowisku DevOps — kluczowe wyzwania	116
Automatyczne rozwiązywanie TDM dla środowiska DevOps	117
Typowe usługi TDM i architektura TDM	118
Korzyści ze zautomatyzowanego TDM	119
Najlepsze praktyki TDM	120
Podsumowanie	120
<b>Rozdział 16. Testowanie w cyklu DevOps z wykorzystaniem architektury mikrousług</b>	<b>123</b>
Czym jest architektura mikrousług?	123
Testowanie w cyklu agile/DevOps z wykorzystaniem architektury mikrousług	124
Testowanie wydajności mikrousług	126
Monitorowanie mikrousług	126
Podsumowanie	127
<b>Rozdział 17. Zautomatyzowane środowiska testowe w DevOps</b>	<b>129</b>
Zarządzanie środowiskiem testowym — kluczowe wyzwania	130
Automatyzacja środowiska testowego w DevOps	131
Korzyści z automatyzacji środowiska testowego	132
Wskaźniki środowisk testowych do uwzględnienia przy zarządzaniu wydaniem	133
Narzędzia automatyzacji środowiska testowego	134
Podsumowanie	135
<b>Rozdział 18. Wirtualizacja usług jako czynnik wspomagający DevOps</b>	<b>137</b>
Wirtualizacja usług i DevOps	137
Rola wirtualizacji usług w DevOps	139
Wirtualizacja usług — badania	140
Narzędzia do wirtualizacji usług używane w DevOps	141
Podsumowanie	141
<b>Rozdział 19. Identyfikacja przypadków do uwzględnienia w testach regresyjnych</b>	<b>143</b>
Testy regresyjne oprogramowania — wprowadzenie	144
Proces testowania regresji oprogramowania	144
Jak dobrać przypadki na potrzeby testów regresyjnych?	144
Podsumowanie	147
<b>Rozdział 20. Automatyzacja testów dostępności w środowisku DevOps</b>	<b>149</b>
Dostępność (AX) — wprowadzenie	149
AX i DevOps	150
Automatyzacja testowania AX w DevOps	151
Standardowe narzędzia AX	152
Podsumowanie	152

<b>Rozdział 21. Optymalizacja wydajności aplikacji Java</b>	<b>153</b>
<b>Wąskie gardła wydajności — główne wyzwania i rozwiązania</b>	<b>154</b>
Wielokrotne wykorzystywanie obiektów	154
Zarządzanie pulami obiektów	155
Normalizacja obiektów	155
Enumeracja stałych	155
Operatory a metody	156
Unikanie nadmiernego gromadzenia śmieci	156
Opóźniona inicjalizacja	156
<b>Narzędzia do optymalizacji wydajności Javy</b>	<b>156</b>
Profiler NetBeans do analizy aplikacji Java	157
<b>Podsumowanie</b>	<b>160</b>
<b>Rozdział 22. Testowanie aplikacji mobilnych — kluczowe wyzwania i zagadnienia</b>	<b>161</b>
<b>Oczekiwania użytkowników</b>	<b>162</b>
<b>Kluczowe wyzwania towarzyszące testowaniu aplikacji mobilnych</b>	<b>163</b>
Wrażenia użytkownika (UX)	163
Dopasowanie kontekstu	163
Zróżnicowane interfejsy urządzeń mobilnych	163
Różnorodność urządzeń i dostępność	163
Testowanie ukierunkowane na urządzenia	164
Automatyczne testowanie layoutów	164
Wyzwania towarzyszące automatyzacji testów z wykorzystaniem niestandardowych narzędzi	164
Skrócenie czasu wprowadzenia produktu na rynek	164
<b>Sposoby na zwiększenie użyteczności aplikacji mobilnych</b>	<b>165</b>
<b>Podsumowanie</b>	<b>165</b>
<b>Rozdział 23. Testowanie aplikacji analitycznych — co się zmieniło w dziedzinie SMAC</b>	<b>167</b>
<b>Zrozumienie klientów, gromadzenie danych i analiza</b>	<b>168</b>
<b>Testowanie danych</b>	<b>168</b>
<b>Testowanie aplikacji BI/BA</b>	<b>169</b>
<b>Jak przebiega testowanie w przypadku aplikacji big data/Hadoop?</b>	<b>169</b>
<b>Podsumowanie</b>	<b>171</b>
<b>Rozdział 24. Migracja aplikacji do chmury — kluczowe kwestie związane z testowaniem</b>	<b>173</b>
<b>Konieczność sprawdzenia gotowości aplikacji do migracji</b>	<b>174</b>
<b>Kluczowe obszary testowania migracji aplikacji do chmury</b>	<b>174</b>
<b>Kluczowe wyzwania towarzyszące walidacji podczas migracji aplikacji do chmury</b>	<b>174</b>
<b>Walidacja migracji aplikacji — podejście kompleksowe</b>	<b>176</b>
<b>Podsumowanie</b>	<b>177</b>



<b>Rozdział 25. Testowanie w chmurze — nowa mentalność</b>	<b>179</b>
<b>Co czeka testerów w chmurze</b>	<b>180</b>
Testowanie tego, czego nie widać	180
Uwzględnienie oddalenia geograficznego	181
Przełamywanie barier komunikacyjnych	181
Zabezpieczanie aplikacji	181
Replikacja platformy	182
Korzystanie z odpowiednich narzędzi	182
<b>Podsumowanie</b>	<b>183</b>
<b>Rozdział 26. Testowanie wydajności na żądanie w środowiskach samoobsługowych</b>	<b>185</b>
<b>Środowiska testowania wydajności na żądanie — kluczowe wyzwania i rozwiązania</b>	<b>186</b>
<b>Platforma chmurowa do tworzenia całościowych testów wydajności</b>	<b>186</b>
<b>Środowiska samoobsługowe do prowadzenia testów wydajności na żądanie</b>	<b>187</b>
<b>Kluczowe scenariusze testowania wydajności na żądanie — proponowana architektura</b>	<b>187</b>
Scenariusz nr 1 — architektura i sposoby prowadzenia testów wydajności	188
Scenariusz nr 2 — architektura i sposoby prowadzenia testów wydajności	189
<b>Podsumowanie</b>	<b>190</b>
<b>Rozdział 27. QA w dziedzinie marketingu cyfrowego</b>	<b>191</b>
<b>Rozwój marketingu cyfrowego</b>	<b>191</b>
<b>Wyzwania towarzyszące wdrażaniu marketingu cyfrowego</b>	<b>192</b>
<b>Znaczenie jakości dla marketingu cyfrowego</b>	<b>192</b>
<b>QA cyfrowych aplikacji marketingowych — kluczowe czynniki weryfikacji</b>	<b>193</b>
<b>Podsumowanie</b>	<b>194</b>
<b>Rozdział 28. Panel bezpieczeństwa dla zarządu</b>	<b>195</b>
<b>Dlaczego monitorowanie i raportowanie zagrożeń bezpieczeństwa jest konieczne?</b>	<b>195</b>
<b>Co należy monitorować i raportować?</b>	<b>196</b>
Zarządzanie bezpieczeństwem aplikacji	196
<b>Parametry wyświetlane na pulpicie</b>	<b>198</b>
<b>Konsekwencje naruszenia zabezpieczeń</b>	<b>199</b>
<b>Kluczowe wyzwania towarzyszące implementacji pulpitu</b>	<b>200</b>
<b>Podsumowanie</b>	<b>200</b>
<b>Rozdział 29. Zrobotyzowana automatyzacja w testowaniu aplikacji mobilnych</b>	<b>201</b>
<b>Roboty programowe i ich zastosowanie</b>	<b>202</b>
<b>Korzyści płynące ze zrobotyzowanej automatyzacji procesów (RPA)</b>	<b>202</b>
<b>Wykorzystanie RPA w testach mobilnych</b>	<b>203</b>
<b>Podsumowanie</b>	<b>206</b>
<b>Źródła</b>	<b>206</b>

<b>Rozdział 30. Kluczowe aspekty testowania aplikacji internetu rzeczy (IoT)</b>	<b>207</b>
Konieczność opracowania solidnej strategii testowania IoT	208
Rewolucja IoT — ustalenia World Quality Report 2016	209
Kwestie do uwzględnienia przy testowaniu IoT	210
Rodzaje testów IoT	211
Podsumowanie	211
Źródła	212
<b>Rozdział 31. Biznes algorytmiczny i testowanie w oparciu o model</b>	<b>213</b>
Rozwój analityki	214
Biznes algorytmiczny — omówienie	215
Konieczność prowadzenia testowania opartego na modelu	216
Podsumowanie	217
<b>Rozdział 32. Wykorzystanie inteligencji kognitywnej w testowaniu</b>	<b>219</b>
Inteligencja kognitywna — wprowadzenie	220
Czym są testy kognitywne?	220
Ewolucja technologii kognitywnych i dostępne platformy	220
Metodologie związane z testowaniem kognitywnym	221
Zalety badań kognitywnych w porównaniu z obecnymi metodami	222
Przypadki zastosowania właściwe dla konkretnych sektorów	222
Obsługiwane technologie	223
Powiązane koszty	223
Wyzwania związane z wdrażaniem technologii kognitywnych	224
Podsumowanie	224
<b>Rozdział 33. Fintech — testowanie i QA w nowej, przełomowej dziedzinie</b>	<b>225</b>
Ekosystem fintech	225
Usługi fintech i ich wpływ na różne sektory rynku	227
Testowanie aplikacji i rozwiązań fintechowych	228
Podsumowanie	229
<b>Rozdział 34. Technologia blockchain — zapewnianie bezpieczeństwa w biznesie</b>	<b>231</b>
Wprowadzenie do blockchajna	232
Proces blockchainowy	232
Popularne blockchajny	233
Wpływ na testowanie	233
Podsumowanie	234
<b>Rozdział 35. Technologie cyfrowych łańcuchów dostaw i zagadnienia QA</b>	<b>235</b>
Zrozumienie procesu nabywczego	236
Trendy branżowe w zarządzaniu łańcuchem dostaw	236
Nowe technologie w zarządzaniu łańcuchem dostaw	237

Nowe technologie w handlu detalicznym	239
QA w dziedzinie łańcuchów dostawczych	240
Podsumowanie	241
<b>Rozdział 36. Potencjalne innowacje w dziedzinie e-zdrowia</b>	
<b>— konsekwencje dla testowania i QA</b>	<b>243</b>
Trendy transformacji cyfrowej w sektorze opieki zdrowotnej	244
E-zdrowie w praktyce	244
Ekosystem cyfrowej opieki zdrowotnej	245
Typowe wyzwania w branży e-zdrowia	246
Potrzeba wprowadzenia solidnego QA	247
Walidacja danych w ramach podróży pacjenta	247
Podsumowanie	247
<b>Rozdział 37. Trendy w globalnym sektorze motoryzacyjnym</b>	
<b>— konsekwencje dla testowania i QA</b>	<b>249</b>
Czynniki stymulujące rozwój branży motoryzacyjnej	250
Najważniejsze trendy na chińskim rynku motoryzacyjnym	250
Wpływ globalnych trendów motoryzacyjnych na rynek chiński	251
W jaki sposób trendy w sektorze motoryzacyjnym wpływają na testowanie oprogramowania?	252
Podsumowanie	253
<b>Rozdział 38. Cyfrowa transformacja w sektorze konsumenckim i detalicznym</b>	
<b>— zagadnienia QA</b>	<b>255</b>
Bieżące wyzwania w sektorze konsumenckim i detalicznym	256
Sklepy stacjonarne i internetowe — główne trendy	256
Sklepy stacjonarne	256
Sklepy internetowe	257
Transformacja w cyfrowym sektorze konsumpcyjnym i detalicznym	258
Podróż klienta cyfrowego przez środowisko omni-channel — przykład	259
Korzyści z cyfryzacji sektora konsumenckiego i detalicznego	260
Czynniki istotne dla nabywców cyfrowych	260
Multi-channel i omni-channel — zagadnienia QA	261
Widoczność zapasów	262
Komunikacja z siecią	262
Predykcyjna analiza klientów	262
Strategia realizacji zamówień	263
Transformacja cyfrowa w sektorze konsumenckim i detalicznym — testowanie	263
Podsumowanie	264
<b>Rozdział 39. Transformacja cyfrowa w energetyce</b>	
<b>i sektorze użyteczności publicznej — zagadnienia QA</b>	<b>265</b>
Trendy technologiczne w sektorze użyteczności publicznej	266
Technologie kluczowe dla transformacji cyfrowej w sektorze energetyki i użyteczności publicznej	267

<b>Komunikacja wielokanałowa i spójność podróży klienta</b>	<b>268</b>
<b>Uwagi dotyczące testów podróży klienta</b>	<b>269</b>
<b>Uwagi dotyczące testów inteligentnej infrastruktury pomiarowej</b>	<b>270</b>
<b>Trendy cyfrowej transformacji QA w sektorze energetyki i użyteczności publicznej</b>	<b>271</b>
<b>Podsumowanie</b>	<b>273</b>
<b>Źródła</b>	<b>273</b>
<b>Rozdział 40. Inteligentna energetyka i sieci energetyczne — skuteczne testowanie</b>	<b>275</b>
<b>Wprowadzenie</b>	<b>275</b>
<b>Trendy w sektorze energetycznym i usług komunalnych</b>	<b>276</b>
Trendy w energetyce i usługach komunalnych — dane WQR 2016	278
<b>Najczęstsze wyzwania w sektorze energetycznym</b>	<b>278</b>
<b>Inteligentne liczniki i sieci</b>	<b>279</b>
<b>Proces testowania SMI</b>	<b>280</b>
<b>Podsumowanie</b>	<b>281</b>
<b>Rozdział 41. Testowanie cyfrowych aplikacji linii lotniczych — projekty responsywne</b>	<b>283</b>
<b>Branża lotnicza — obecna sytuacja</b>	<b>283</b>
<b>Rola cyfryzacji</b>	<b>284</b>
<b>QA i testowanie w branży lotniczej</b>	<b>285</b>
Testowanie wdrożenia reguł biznesowych	285
Testowanie integracji systemów	286
Testy нефunkcjonalne	286
<b>Testowanie responsywnych stron internetowych</b>	<b>286</b>
<b>Podsumowanie</b>	<b>288</b>
<b>Rozdział 42. Testowanie tablic ortogonalnych (OAT) — zastosowanie w sektorze opieki zdrowotnej</b>	<b>289</b>
<b>Wprowadzenie</b>	<b>290</b>
<b>Wyzwania towarzyszące testowaniu aplikacji H&amp;LS</b>	<b>290</b>
<b>Jak wykorzystać strategię testowania tablic ortogonalnych (OATS)?</b>	<b>291</b>
<b>Podsumowanie</b>	<b>293</b>
<b>Rozdział 43. Przyszłość konsultingu w erze przemian cyfrowych</b>	<b>295</b>
<b>Przekształcenia konsultingu w wyniku rewolucji cyfrowej</b>	<b>296</b>
<b>Konsultant w erze cyfrowej</b>	<b>296</b>
<b>Przyszłość doradztwa</b>	<b>297</b>
<b>Co bym teraz robiła inaczej?</b>	<b>298</b>
Kontakt z klientami przez cały cykl rozwoju	298
Unikanie nadmiernych wydatków na podróże służbowe	299
Crowdsourcing jako sposób na podniesienie współczynnika wykorzystania	299
Elastyczny model operacyjny firm konsultingowych	299
<b>Podsumowanie</b>	<b>300</b>

<b>Rozdział 44. Przyszłość testowania w świecie cyfrowym</b>	<b>301</b>
<b>Kluczowe trendy technologiczne kształtujące przyszłość IT</b>	<b>301</b>
Rozpowszechnione technologie i analizy predyktywne UX	302
Inteligencja kognitywna w połączonych z siecią pojazdach autonomicznych	302
Wielokanałowa łączność użytkownika i urządzenia noszone	303
Dezintermediacja — platforma łącząca nowych partnerów	304
Przeobrażenia miejsca pracy wskutek robotyzacji procesów	305
<b>Kwestie związane z testowaniem nowych technologii</b>	<b>305</b>
Wszechstronne technologie i analityka predyktywna	305
Inteligencja kognitywna — CAV	306
Wielokanałowa łączność użytkownika — technologie noszone	306
Dezintermediacja — platforma biznesowa	306
Przeobrażenia miejsca pracy wskutek robotyzacji procesów	306
<b>Podsumowanie</b>	<b>307</b>
<b>Rozdział 45. Testowanie w przyszłości — możliwości rozwoju zawodowego</b>	<b>309</b>
<b>Zawody testerskie</b>	<b>310</b>
<b>Typowe stanowiska pracy</b>	<b>311</b>
<b>Znaczenie technologii i narzędzi branżowych lub dziedzinowych w karierze testera</b>	<b>312</b>
<b>Umiejętności wymagane w testowaniu zwinnym i DevOps</b>	<b>312</b>
<b>Nowy profil testera w DevOps</b>	<b>313</b>
<b>Podsumowanie</b>	<b>314</b>
<b>Rozdział 46. Robotyka, uczenie maszynowe, internet rzeczy — konsekwencje dla indyjskiego sektora usług</b>	<b>315</b>
<b>Robotyka, uczenie maszynowe i IoT — kontekst</b>	<b>316</b>
<b>Jaki wpływ wywiera postęp technologiczny na gospodarkę indyjską?</b>	<b>317</b>
Globalny spadek cen ropy — wnioski	317
Co spowodowało zahamowanie wzrostu gospodarki chińskiej?	317
<b>Perspektywy dla indyjskiego sektora usług — opinia</b>	<b>318</b>
<b>Podsumowanie</b>	<b>319</b>
<b>Źródła</b>	<b>321</b>
<b>Rozdział 2. Przyszłość modeli testowych — czy prognozy wzrostu wydatków na QA są uzasadnione?</b>	<b>321</b>
<b>Rozdział 3. Korzyści z zastąpienia podwykonawców testowych zarządzanymi usługami testowymi</b>	<b>321</b>
<b>Rozdział 18. Wirtualizacja usług jako czynnik wspomagający DevOps</b>	<b>322</b>
<b>Rozdział 30. Kluczowe aspekty testowania aplikacji internetu rzeczy (IoT)</b>	<b>322</b>
<b>Rozdział 31. Biznes algorytmiczny i testowanie w oparciu o model</b>	<b>322</b>
<b>Rozdział 46. Robotyka, uczenie maszynowe, internet rzeczy — konsekwencje dla indyjskiego sektora usług</b>	<b>322</b>
<b>Skorowidz</b>	<b>323</b>



# Rola automatyzacji w DevOps

W dzisiejszej branży informatycznej uwzględnia się scenariusze, w których stale rozwijany kod można wdrożyć w przeciągu kilku minut, a nawet sekund. Wdrażanie zmian kodu z repozytorium kodu źródłowego w ustalonych odstępach czasowych odchodzi w zapomnienie. Poza przyspieszeniem wdrażania konieczna jest też weryfikacja jakości kodu i tym samym zapewnienie satysfakcji użytkownikowi końcowemu. Zespoły programistyczne stają zatem wobec wyzwania, jakim jest wyważenie szybkości prac z osiąganą jakością. Szybkość dotyczy prędkości, z jaką oprogramowanie jest opracowywane zgodnie z wymaganiami i ustalonymi normami. Czy istnieje jakiś sposób, aby zweryfikować, czy wdrażane oprogramowanie spełnia wymogi jakościowe?

Obecnie wiele organizacji przyswaja sobie praktyki *agile* i DevOps, które mają umożliwić im wypełnienie luki dzielącej wdrażanie kodu od weryfikacji jakości.

W tym rozdziale omówimy następujące zagadnienia:

- znaczenie automatyzacji w DevOps;
- automatyzacja w ramach wczesnej integracji;
- automatyzacja jako domyślna praktyka;
- automatyzacja wdrażania;
- automatyzacja — wskaźniki i pomiar;
- najlepsze praktyki związane z przyjęciem DevOps i stałą automatyzacją.

## Znaczenie automatyzacji dla DevOps

DevOps nieodłącznie wiąże się ze stosowaniem automatyzacji w cyklu rozwoju oprogramowania już od etapu określania wymagań aż po zakończenie fazy operacyjnej:

- Ręczne testowanie statyczne wymagań jest obecnie zastępowane **rozwojem opartym na testach akceptacyjnych** (ang. *Acceptance Test-Driven Development* — **ATDD**) i **rozwojem opartym na zachowaniu** (ang. *Behavior-Driven Development* — **BDD**).
- ATDD i BDD z wykorzystaniem Cucumber, Fitnesse lub RSpec pomagają w dokumentacji wymagań i automatyzacji.
- Automatyzacja ta jest wprowadzana przez **inżynierów tworzących oprogramowanie testowe** (ang. *Software Development Engineers in Testing* — **SDET**), którzy posiadają umiejętności związane zarówno z rozwojem oprogramowania, jak i testowaniem.
- W trakcie prowadzenia działań BDD zespół programistyczny może korzystać z narzędzi xUnit, takich jak NUnit lub Junit, w ramach pracy nad **rozwojem opartym na testach** (ang. *Test-Driven Development* — **TDD**).
- Wszystkie te testy można zintegrować z serwerem **ciągłej integracji (CI)** w celu prowadzenia ich równoległe do tworzenia oprogramowania w fazie poprzedzającej wdrożenie.

Powyższe techniki wykorzystują podejście *shift left* w odniesieniu do zapewniania jakości, w ramach którego prowadzi się testy wymagań i testy jednostkowe.

Wymagania mogą zmieniać się w trakcie cyklu rozwoju oprogramowania, a zmiany te również muszą przejść przez podobny cykl automatyzacji.

Tym niemniej można mieć pewność, że dzięki zastosowaniu powyższych rozwiązań wymagania są znane zarówno programistom, jak i testerom, co usprawnia komunikację w obrębie zespołu.

## Wczesna automatyzacja integracji

Jako że częściowa automatyzacja odbyła się już w ramach BDD i TDD, a także została zintegrowana z serwerem CI, można się zająć komponentami z poziomu usług, związanymi z nieopracowanym jeszcze **interfejsem użytkownika (UI)**.

Są też chwile, kiedy podstawową automatyzację UI można wprowadzić z wykorzystaniem makiet. Skrypty jednak muszą ulec zmianie po przygotowaniu UI. Wysilek konieczny do zmodyfikowania istniejących skryptów jest bezpośrednio powiązany z jakością makiet, które przygotowano na potrzeby wstępnej pracy automatyzacyjnej.



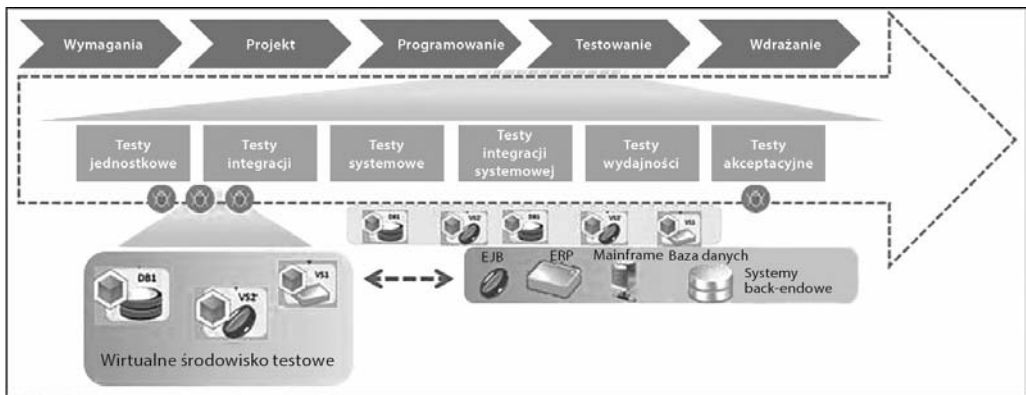
W celu umożliwienia integracji serwer CI buduje system na żądanie, według harmonogramu, czy nawet po zgłoszeniu kodu.

- Każdy programista może uruchomić i przetestować kod w swoim systemie, aby upewnić się, czy nie jest uszkodzony.
- W celu przeprowadzenia wczesnych i rozległych testów integracyjnych zespół testerów może wykorzystać możliwości wirtualizacji usług, aplikacji i sieci.

Niezależnie od tego, czy chodzi o integrację UI z usługą lub warstwą bazy danych, czy integrację jednego systemu z innymi systemami i narzędziami, wirtualizacja zapewnia różnorodne rozwiązania, koncentrując się na podejściu *shift left* w ramach testów szczegółowych i całościowych. Narzędzia do wirtualizacji usług, takie jak CA LISA, IBM Rational Integration Tester, HP Service Virtualization i Parasoft Virtualize, pomagają z następującymi kwestiami:

- skracaniem cyklu wdrażania;
- wskazywaniem błędów na wczesnym etapie cyklu życia oprogramowania.

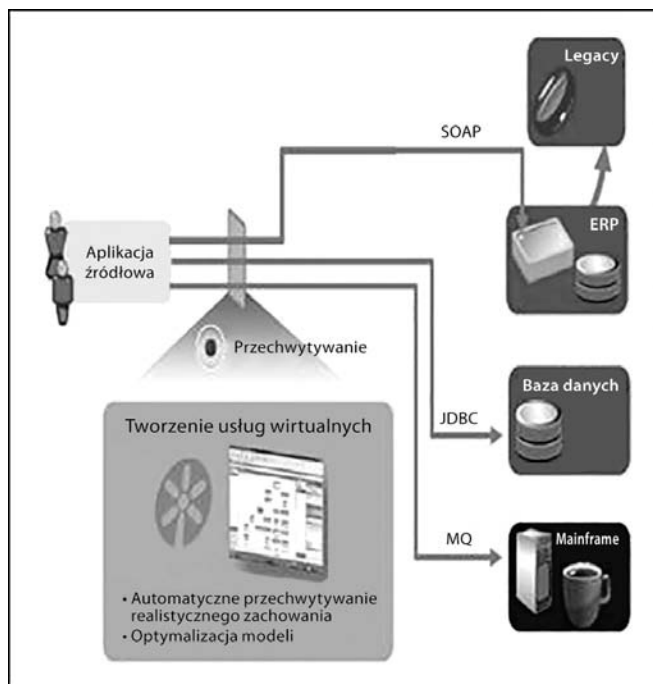
Ponieważ dedykowane środowiska testowe rzadko są dostępne na wczesnym etapie cyklu życia oprogramowania, wirtualne środowiska testowe i wirtualizację serwerową można wykorzystać do stworzenia środowiska do prowadzenia testów na poziomie usług i uruchamiania innych zautomatyzowanych skryptów. Widać to na poniższym schemacie:



Zachowanie, dane i cechy funkcjonowania zależnych i równorzędnych systemów można przechwytywać i symulować przy użyciu wirtualizacji usług.

Przechwycone usługi i symulowane dane testowe można wprowadzić do środowiska testowego w celu przedstawienia zależnego systemu na potrzeby wczesnego testu integracyjnego. Spójrz na schemat na następnej stronie.

Oprócz identyfikowania i symulowania usług automatyzacja współdziałających ze sobą interfejsów API na podstawie rozpoznawania podpisu API pozwala na wczesne wykrywanie defektów jeszcze przed opracowaniem UI. Informacje o każdej zmianie podpisu API przez zespół programistyczny powinny być przekazywane zespołowi odpowiedzialnemu za automatyzację, tak aby mógł on zaktualizować skrypty w terminie i uniknąć przekazania defektów do kolejnych etapów.



Tego rodzaju testowanie integracyjne wymaga od zespołów testowych posiadania wystarczającej wiedzy technicznej związanej z użytkowaniem narzędzi wirtualizacji usług, a także umiejętności pisania kodu testów API i interfejsu. Oto korzyści, jakie zapewnia wirtualizacja usług:

- szybszy czas wprowadzenia produktu na rynek;
- redukcja liczby usterek;
- stała dostępność usług.

## Automatyzacja — domyślna praktyka

Uczynienie automatyzacji domyślną i oczywistą praktyką w środowisku zwinnym i DevOps pomaga w organizacji poniższego procesu:

- Jeśli istniejąca funkcjonalność zostanie zepsuta przez nową partię kodu, jest to niezwłocznie zgłaszane.
- Programista zostaje powiadomiony i jest zobowiązany do naprawienia zaistniałych błędów.
- Zakres testów poszerza się, ponieważ istniejąca funkcjonalność jest sprawdzana przez automatyczne skrypty w ramach kolejnych wersji.
- Zespół testerów może skoncentrować się na sprawdzaniu poprawności nowych funkcji, zwiększając tym samym szybkość prac rozwojowych.

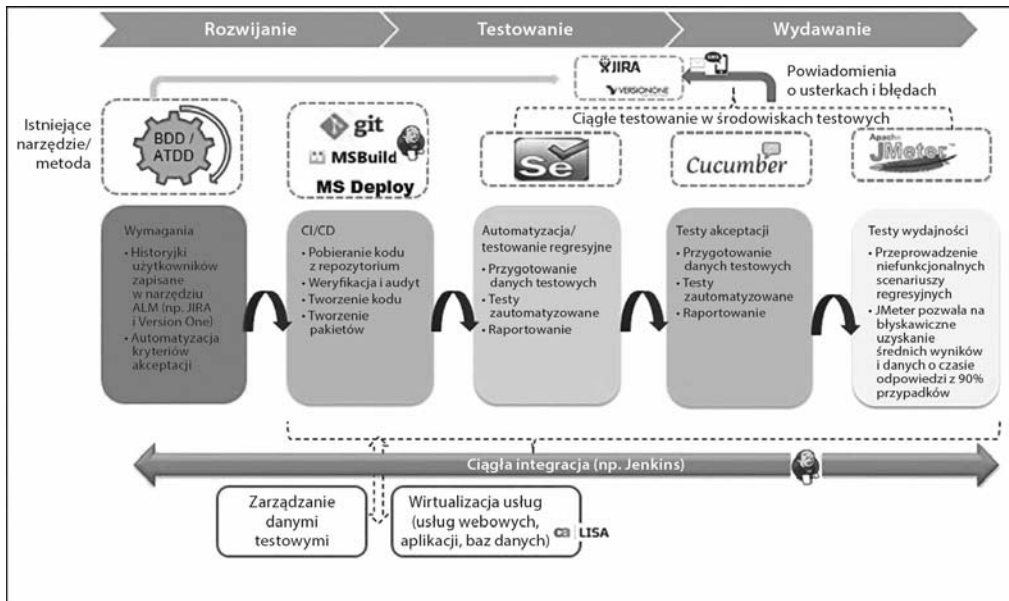
O ile TDD, ATDD, BDD i wirtualizacja usług koncentrują się na wykorzystaniu podejścia do testowania typu „od wewnątrz do zewnątrz”, o tyle podejście zakładające „testowanie po fakcie” wymaga prowadzenia równoległe niezależnych testów polegających na przyglądaniu się oprogramowaniu „od zewnątrz do wewnątrz”.

Aby zapewnić zachowanie odpowiedniej szybkości rozwoju oraz jakości testów w ramach modelu DevOps, uwzględnij poniższe punkty:

- W ramach automatycznych testów zaleca się sprawdzanie określonego, minimalnego zestawu funkcji w każdym testowanym wydaniu.
- Funkcjonalność systemu można zautomatyzować za pomocą takich narzędzi jak Selenium, HP Unified Functional Tester, Eggplant i inne.
- Dopasowanie kolejności i priorytetów funkcji przeznaczonych do automatyzacji do harmonogramu iteracji oraz wydań zapewnia wysoki poziom zwrotu z inwestycji.

Jeśli na przykład od działania formularza rejestracji użytkownika uzależnione jest wykonywanie dalszych czynności w aplikacji, warto tę funkcję zautomatyzować z uwzględnieniem pozytywnych i negatywnych scenariuszy i testować ją w każdym z wydań. Dzięki temu zyskuje się pewność, że istniejąca funkcja działa poprawnie, a zespół testowy może spokojnie przejść do sprawdzania innych funkcji, w tym zależnych. Zwiększa to również zaufanie do jakości tworzonego oprogramowania lub aplikacji.

Praktyka ciągłego tworzenia wykonywalnego kodu automatyzacyjnego na potrzeby priorytetowych funkcji z kodem testowym dla API i usług wraz z integracją tego zautomatyzowanego kodu z serwerem CI pozwala oszczędzić czas przeznaczony na wdrażanie, tym samym zapewniając wyższe ROI i krótszy czas wprowadzania produktu na rynek. Spójrz na poniższy schemat:



## Automatyzacja wdrażania

Ciągły rozwój, ciągła integracja kodu i ciągle wdrażanie w środowiskach fizycznych, wirtualnych oraz w chmurze dają możliwość ciągłego testowania. Jeśli systemy testów regresji, usług, zabezpieczeń i wydajności są zintegrowane z serwerem ciągłym, można je uruchomić od razu po wdrożeniu danej wersji w docelowym środowisku. Stworzenie pakietu automatyzacyjnego dla scenariuszy UAT i konkretnych scenariuszy produkcyjnych pomaga w prowadzeniu testów przed i po wdrożeniu oraz umożliwia skrócenie cykli przekazu informacji zwrotnych.

## Automatyzacja — wskaźniki i pomiar

Kiedy automatyzacja przeprowadzana jest dopiero po ukończeniu całego produktu, liczba wykonywanych operacji zautomatyzowanych często jest ograniczona z uwagi na brak czasu na przeprowadzenie odpowiednich prac. Po uwzględnieniu nakładu pracy przeznaczonego na rozwój i konserwację może się okazać, że ROI z automatyzacji jest niewielkie, a czasami nawet ujemne.

Do przeprowadzenia udanej inwestycji konieczne jest wczesne rozpoczęcie działań automatyzacyjnych w odniesieniu do priorytetowych funkcji i w oparciu o harmonogram wydań.

System śledzenia i monitorowania pozwala na modyfikowanie kierunku prac na podstawie zgromadzonych danych. Najważniejsze wskaźniki do śledzenia to zakres automatyzacji, indeks automatyzacji, poziom wielokrotnego użytku modelu i skryptów, wzrost produktywności w wyniku automatyzacji i ROI z automatyzacji.

## Zakres automatyzacji

Wskaźnik ten zwykle mierzy się jako stosunek pomiędzy liczbą automatyzowalnych przypadków testowych a łączną liczbą przypadków testowych w ramach zbioru testów regresyjnych i funkcjonalnych. Średnio 70% – 80% pokrycia można osiągnąć poprzez zautomatyzowanie testów funkcjonalności, regresji i testów na poziomie usługi. Automatyzacja testów całościowych może dodatkowo zwiększyć ten zakres.

## Indeks automatyzacji

Jest to stosunek liczby zautomatyzowanych skryptów do łącznej liczby przypadków testowych wybranych dla danego cyklu testowego. Im wyższy indeks, tym wyższe jest ROI i tym większe są korzyści z automatyzacji.

## Poziom wielokrotnego użytku modelu i skryptów

Dysponowanie solidnym i rozszerzalnym modelem automatyzacji ułatwia jej wdrożenie. Istotne jest również to, aby model ten był zgodny z założeniami programowania modularnego, czyli żeby możliwe było wielokrotne wykorzystanie jego skryptów i bibliotek. Pomiar tego wskaźnika można przeprowadzić dopiero po wstępnym wdrożeniu automatyzacji.

## Wzrost produktywności w wyniku automatyzacji

Liczba skryptów, które mogą zostać wytworzone i wykonane w danym okresie — np. w ciągu jednego dnia — jest wskaźnikiem produktywności zespołu. Aby uznać automatyzację za udaną, wartość tego wskaźnika powinna być wyraźnie wyższa od wskaźnika produktywności w przypadku testowania manualnego. Te dane produktywności zazwyczaj wzrastają w miarę upływu czasu, gdy zestawia się je z czynnikiem wielokrotnego użytku oraz rozwojem wiedzy dziedzinowej zespołu.

## ROI automatyzacji

W ramach automatyzacji konieczne jest poświęcenie czasu na poniższe działania:

- zaprojektowanie modelu;
- napisanie skryptów;
- wykonanie przeglądów i testów;
- prowadzenie działań konserwacyjnych w miarę wydawania kolejnych wersji, aby uwzględnić zmiany odpowiadające poszczególnym wersjom;
- wykonanie testów w każdym cyklu.

Porównanie czasu przeznaczanego na te czynności z czasem, jaki trzeba by przeznaczyć na wykonanie ich manualnie, określa ROI. Jeśli pomimo wprowadzenia skryptów automatyzujących testy nadal są wykonywane ręcznie, ROI automatyzacji należy uznać za ujemne.

Przy obliczaniu ROI należy też uwzględnić czas spędzany na analizowaniu nieudanych skryptów. Niektóre skrypty nie działają z powodu wad samej aplikacji, a ich analiza jest dość prosta. Jeśli skrypty nie działają z innego powodu, np. wskutek błędów w logice skryptu opóźnień sieci bądź wskutek opóźnień w dostawie usług zewnętrznych, a czas potrzebny do przeanalizowania tych błędów jest dłuższy, to ma to bezpośredni, niekorzystny wpływ na wydajność i ROI.

## Śledzenie postępów automatyzacji

Ponieważ praca nad automatyzacją wyjęta jest poza ogólny zakres prac, istotne jest opracowanie harmonogramu i uważne jego monitorowanie. W raportach dziennych, tygodniowych i miesięcznych powinna znajdować się sekcja przedstawiająca omówione powyżej wskaźniki automatyzacji, zagrożenia, zależności itd., aby możliwe było zapobieganie problemom.

# Najlepsze praktyki wczesnego i ciągłego automatyzowania

Tak samo jak każde inne przedsięwzięcie automatyzacja wymaga spełnienia pewnych podstawowych warunków, aby możliwe było jej przeprowadzenie. Oto niektóre z nich:

- uznanie automatyzacji za zwyczajną pracę rozwojową;
- inżynieria jakości;
- współpraca zespołów projektowych, testowych i automatyzacyjnych;
- wybór trafnych rozwiązań.

## Uznanie automatyzacji za zwyczajną pracę rozwojową

Jako że automatyzacja leży u podstaw sprawnego prowadzenia działań DevOps, konieczne jest zachowanie pewnej dyscypliny. Automatyzacja powinna być traktowana tak samo jak inne prace rozwojowe. Zespół powinien działać spójnie i stosować się do najlepszych praktyk, niezależnie od tego, czy zajmuje się projektowaniem modelu, przeglądaniem kodu, czy jednostkowymi testami kodu i skryptów. Każde nowe zadanie powinno być łatwe do włączenia w model wdrożeniowy, co przekłada się na skrócenie czasu potrzebnego na rozpoczęcie prac i ma pozytywny wpływ na ROI.

## Inżynieria jakości

Automatyzację należy rozpoczynać na jak najwcześniejszym etapie, aby móc przenieść środek ciężkości prac ze zwykłych testów na inżynierię jakości jako taką. Praktyki takie jak BDD, testowanie usług, testy API i inżynieria wydajności pomagają zwrócić uwagę na inżynierski aspekt testowania i ułatwiają znajdowanie oraz naprawianie defektów na dużo wcześniejszym etapie cyklu.

## Współpraca zespołów projektowych, testowych i automatyzacyjnych

Jeśli zapewnienie kolokacji zespołów rozwojowych i testowych nie jest możliwe, kluczowe dla synchronizacji ich działań jest prowadzenie komunikacji wirtualnej, obejmującej wykorzystanie scentralizowanych narzędzi, a także prowadzenie codziennych zebrań i utrzymywanie regularnej interakcji pomiędzy pracownikami. Wdrożenie kompilacji w środowisku testowym, znajdującym się w chmurze lub wirtualnym systemie, jest niezbędne dla ciągłego wykonywania zautomatyzowanych skryptów i gromadzenia danych zwrotnych.

## Wybór właściwego rozwiązania

Na rynku można znaleźć wiele rozwiązań służących automatyzacji. Niektóre narzędzia skupiają się tylko na jednym aspekcie działań, np. JUnit na TDD czy Selenium na testach funkcjonalnych. Kompleksowe rozwiązania automatyzacyjne, w ramach których większość tych narzędzi jest połączona centralnie z serwerem ciągłej integracji przy użyciu specjalnego kodu — takie jak Jenkins czy Bamboo — są na ogół produktami własnymi organizacji, które z nich korzystają, bądź własnością przodujących dostawców, takich jak HP.

Wybór odpowiedniego podejścia do automatyzacji i optymalnych rozwiązań jest kluczowy dla udanego oraz sprawnego wdrożenia DevOps.

## Podsumowanie

Automatyzacja jest istotnym zadaniem w obiegu pracy DevOps. Jej wczesne wykorzystanie pozwala na osiągnięcie w ramach DevOps największych korzyści. W tym rozdziale omówiliśmy zagadnienie *wczesnej integracji* oraz domyślnej automatyzacji. Automatyzacja powinna skupiać się na wszystkich działaniach: projektowaniu, budowaniu i wdrażaniu. Omówiliśmy też wskaźniki pomiarowe służące do śledzenia postępów automatyzacji. Przedstawiliśmy najlepsze praktyki do wykorzystania w ramach wczesnego wdrożenia automatyzacji i stałego jej prowadzenia.





# Skorowidz

## A

adaptacyjność, 219, 222  
agile, 69  
akcelerator, 48  
akceptacja metodyki zwinnej,  
73  
analitka, 214  
    biznesowa, BA, 27  
    predykcyjna, 305  
aplikacje  
    analityczne, 167  
    BI/BA, 169  
    big data/Hadoop, 169  
    chmurowe, 27  
    fintechowe, 228  
    H&LS, 290  
    linii lotniczych, 283  
    mobilne, 161  
architektura  
    mikrosług, 123  
    TDM, 118  
ATDD, Acceptance Test-  
    Driven Development, 84  
ATF-WAVE, 152  
automatyczne  
    testowanie layoutów, 164  
    zarządzanie konfiguracją,  
    112

automatyzacja, 83, 86  
    indeks, 88  
    najlepsze praktyki, 90  
    procesów, 202  
    ROI, 89  
    scentralizowana, 132  
    śledzenie postępów, 89  
    środowiska testowego  
        korzyści, 132  
        narzędzia, 134  
    testowania AX, 151  
    testów dostępności, 149  
    testów oprogramowania, 202  
    wdrażania, 88  
    wskaźniki i pomiar, 88  
    wybór rozwiązania, 91  
    wzrost produktywności, 89  
    zakres, 88  
    zarządzania konfiguracją, 109  
        korzyści, 113  
    zlokalizowana, 132  
    zrobotyzowana, 201  
AX, 150

## B

BA, Business Analytics, 27  
bariery komunikacyjne, 181  
BDD, Behavior-Driven  
    Development, 64, 77, 84, 105

benchmarking, 93  
    DevOps QA, 96  
bezpieczeństwo, 231  
    aplikacji, 174, 196  
    monitorowanie, 195  
    procesów biznesowych, 240  
    raportowanie zagrożeń, 195  
big data, 238  
BIT, business integration  
    testing, 280  
bitcoin, 233  
biznes algorytmiczny, 213, 215  
blockchain, 231, 232, 239  
branża  
    lotnicza, 283  
    motoryzacyjna, 250

## C

CA Service Virtualization, 141  
całkowity koszt posiadania, 113  
ChatOps, 99–102  
    korzyści, 102  
Chef, 134  
chmura, 111, 173, 267  
    testowanie aplikacji, 179  
CI, ciągła integracja, 84  
CM, configuration  
    management, 110  
CoE, 75

crowdsourcing, 51, 60, 299  
 crowdtesting, 52  
   korzyści, 55  
   modele operacyjne, 53  
 cyfrowa  
   fabryka QA, 27, 46  
   opieka zdrowotna, 245  
 cyfryzacja sektora  
   konsumenckiego  
   i detalicznego, 260  
 czynniki transformacyjne QA, 26

## D

dane WQR, 28  
 DCX, 47  
 definicja ukończenia, 82  
 DevOps, 37, 73, 76–81, 312  
   automatyczne rozwiązanie  
     TDM, 117  
   automatyzacja, 83, 86  
     środowiska testowego,  
       129, 131  
     testowania AX, 151  
     testów dostępności, 149  
 ChatOps, 99  
 ocena stanu wdrożenia, 93  
 testowanie, 123  
 trendy, 99  
 wdrożenie, 94  
 wirtualizacja usług, 137  
 zarządzanie danymi  
   testowymi, 115  
 DevOps QA benchmarking, 95  
 dezintermediacja, 306  
 dopasowanie kontekstu, 163  
 doradztwo, *Patrz także*  
   konsulting, konsultacje QA  
   kooperacyjne, 297  
   oparte na zasobach, 297  
   specjalistyczne, 297  
 dostarczanie usług DQAF, 48  
 dostępność, AX, 149  
 DQAF, 46, 47  
   dostarczanie usług, 48

  elementy składowe, 48  
   funkcje wspierające, 47  
   korzyści, 48  
   stymulatory wydajności, 48  
 drukowanie 3D, 238  
 dynamiczne procesy biznesowe,  
   240  
 dystrybucja, 279

## E

efektywność, 25  
 ekonomia skali, 42  
 ekosystem fintech, 225  
 elastyczne TCoE, 51  
 elastyczny model operacyjny,  
   299  
 energetyka, 265, 278  
 enumeracja stałych, 155  
 ethereum, 233  
 e-zdrowie, 243–246

## F

fabryki usług, 45  
 fintech, 225, 237  
   administracja rządowa, 226  
   firmy hi-tech, 226  
   tradycyjne firmy, 226

## G

Gherkin, 105, 106  
   korzyści, 108  
 Google DeepMind, 221  
 gromadzenie  
   danych, 168  
   śmieci, 156

## I

IBM Automated Accessibility  
 Tester, 152  
 IBM Digital Content Checker,  
 152

IBM Watson, 221  
 implementacja pulpitu, 200  
 industrializacja, 39  
 indyjski sektor usług, 315  
 integracja, 267  
   mediów społecznościowych,  
     194  
 inteligencja kognitywna, 219,  
   220, 306  
 inteligentna energetyka, 275  
 inteligentne  
   liczniki, 279  
   maszyny, 238  
   systemy, 267  
 intensyfikacja działań QA, 32  
   w fazie projektowania, 33  
 interaktywność, 219, 222  
 interfejsy urządzeń mobilnych,  
   163  
 internet rzeczy, IoT, 26, 207,  
   238, 315  
 inżynieria jakości, 90  
 IT  
   trendy technologiczne, 301  
 iteracyjność, 219, 222

## J

Java  
   optymalizacja wydajności,  
     156  
 język Gherkin, 105  
 JVM, Java Virtual Machines,  
 156

## K

kalkulator oszczędności, 43  
 Kanban, 65, 66, 80  
 katalog usług, 47  
 Keyless Signature  
   Infrastructure, 233  
 kluczowy wskaźnik  
   efektywności, KPI, 41, 48  
 kompatybilność środowiska, 175

kompetencje domenowe, 47  
 komunikacja wielokanałowa, 268  
 konsultacje QA, 298  
 konsulting, 295, 296  
 kontekstowość, 222  
 kontekstualność, 219  
 korzyści z crowdtestingu, 55  
 KPI, key performance indicator,  
 41

**L**

Lean, 65

**Ł**

łańcuch dostaw, 235, 237  
 łączność użytkownika, 303

**M**

marketing cyfrowy, 191  
 materiały crowdtestingowe, 53  
 metoda benchmarkingowa, 93  
 metodologia agile, 69  
 metody, 156  
 metodyka zwinna, 63, 81  
 migracja aplikacji do chmury,  
 173  
 mikrousługi, 123  
 monitorowanie, 126  
 testowanie wydajności, 126  
 model  
 DQAF, 48  
 fabryczny cyfrowego QA, 45  
 rozproszonej zwinności, 68  
 TCoE, 57  
 testowania jako usługi, TaaS,  
 46  
 waterfall, 64  
 modele  
 operacyjne crowdtestingu, 53  
 przedsięwzięć testowych, 35  
 testowe, 31  
 modelowanie, 139

monitorowanie, 195  
 mikrousług, 126  
 MTS, Managed Test Services,  
 35, 39  
 multi-channel, 261

**N**

nadzorcy łańcucha dostaw, 238  
 naruszenie zabezpieczeń, 199  
 narzędzia  
 automatyzacji  
 środowiska testowego,  
 134, 152  
 automatyzacyjne, 48  
 do optymalizacji wydajności,  
 156  
 do wirtualizacji usług, 141  
 narzędzie  
 ATF-WAVE, 152  
 CA Service Virtualization,  
 141  
 Chef, 134  
 IBM Automated  
 Accessibility Tester, 152  
 IBM Digital Content  
 Checker, 152  
 Parasoft Virtualize, 141  
 Profiler NetBeans, 157  
 Puppet, 134  
 ServiceNow, 134  
 SmartBear, 141  
 SmartFrog, 134  
 SSH, 134  
 TEMS, 134  
 nasłuch, 138  
 normalizacja obiektów, 155

**O**

OAT, orthogonal array testing,  
 289  
 oddalenie geograficzne, 181  
 OEM, original equipment  
 manufacturers, 251

omni-channel, 261  
 OpenAI, 221  
 operacja, 81  
 operatorzy, 156  
 opieka zdrowotna, 244, 289  
 opiekun crowdtestingu, 53  
 opóźniona inicjalizacja, 156  
 optymalizacja wydajności, 153

**P**

PaaS, 252  
 panel bezpieczeństwa dla  
 zarządu, 195  
 Parasoft Virtualize, 141  
 planowanie zasobów  
 przedsiębiorstwa, 295, 305  
 platforma  
 chmurowa, 186  
 crowdtestingowa, 52  
 dezintermediacyjna, 304  
 płaszczyzna  
 industrializacji, 35  
 innowacyjności, 36  
 zwykłej działalności  
 biznesowej, 35  
 podejście zwinne, 76–81  
 podwykonawca, 40, 41  
 podział prac nad QA, 32  
 pojazdy autonomiczne, 302  
 postęp technologiczny, 317  
 poziom  
 dojrzałości cyfrowej, 27  
 lojalności klientów, 167  
 priorytety  
 testowania, 26  
 transformacji QA, 26  
 proces  
 blockchainowy, 232  
 testowania SMI, 280  
 produkcja zwinna, 63  
 Profiler NetBeans, 157  
 projektowanie oparte na  
 zachowaniu, BDD, 77  
 projekty responsywne, 283

przechwytywanie, 138  
 przekształcenia konsultingu, 296  
 przemysł motoryzacyjny, 251  
 przetwarzanie w chmurze, 238  
 przypadki testowe, 145  
 Puppet, 134

## Q

QA, quality assurance, 19, 81,  
 191, 235, 240  
 QA cyfrowych aplikacji  
 marketingowych, 193

## R

raportowanie, 195  
 replikacja platformy, 182  
 responsywne strony, 286  
 roboty  
 procesowe, 204  
 programowe, 202  
 testowe, 204  
 robotyka, 315  
 robotyzacja procesów, 305, 306  
 rodzaje  
 metodyk zwinnych, 65  
 środowisk testowych, 110  
 testów IoT, 211  
 ROI automatyzacji, 89  
 role w zespole, 107  
 rozwój, 81  
 aplikacji chmurowych, 27  
 oparty na testach, 84  
 oparty na testach  
 akceptacyjnych, ATDD, 84  
 oparty na zachowaniu, BDD,  
 64  
 zawodowy, 309  
 RPA, 202  
 rynek chiński, 251

## S

scenariusze testowania  
 wydajności, 187  
 Scrum, 65, 80  
 sektor  
 energetyczny, 267, 271, 276,  
 278  
 konsumencki, 255  
 motoryzacyjny, 249, 252  
 opieki zdrowotnej, 289  
 usług  
 indyjski, 318  
 usług komunalnych, 276  
 użyteczności publicznej,  
 265–271  
 ServiceNow, 134  
 sieci energetyczne, 275  
 SIT, system integration testing,  
 280  
 sklepy  
 internetowe, 257  
 stacjonarne, 256  
 SMAC, 167, 252  
 SmartBear, 141  
 SmartFrog, 134  
 sprawdzanie  
 jakości komunikatów, 193  
 jakości treści, 193  
 sprzedaż  
 all-channel, 259  
 omni-channel, 259  
 SSH, 134  
 stabilność środowiska, 133  
 stan wątku, 157  
 strategia testowania IoT, 208  
 symulacja, 139  
 system zarządzania  
 wiedzą, 41  
 przestojami, 279

## Ś

środowiska  
 rozproszone, 68  
 samoobsługowe, 185, 187  
 testowe, 110  
 środowisko omni-channel, 259

## T

TaaS, Testing as a Service, 46  
 tablice ortogonalne, 289, 291  
 TCoE, 39  
 crowdtesting, 54  
 na żądanie, 51  
 TDD, Test-Driven  
 Development, 84  
 TDM, 116  
 korzyści, 119  
 najlepsze praktyki, 120  
 technologia blockchain, 231  
 technologie  
 kognitywne, 224  
 w handlu detalicznym, 239  
 w zarządzaniu  
 łańcuchem dostaw, 237  
 telemedycyna, 244  
 TEM, test environment  
 management, 129  
 TEMS, 134  
 test  
 inteligentnej infrastruktury  
 pomiarowej, 270  
 podróży klienta, 269  
 wydajności, 211  
 testerzy, 310, 313  
 testowanie, 26, 57  
 aplikacji  
 analitycznych, 167  
 BI/BA, 169  
 fintechowych, 228  
 H&LS, 290  
 internetu rzeczy, 207  
 mobilnych, 161, 201  
 bazy danych, 211

bezpieczeństwa, 169  
 danych, 168  
 integracji, 124, 175
 

- biznesowej, BIT, 280
- systemowej, SIT, 280, 286

 IoT, 208, 210  
 jako usługa, 53  
 kompatybilności, 211  
 layoutów, 164  
 migracji aplikacji do chmury, 174  
 niezawodności, 211  
 nowych technologii, 305  
 pracy awaryjnej, 169  
 regresji oprogramowania, 144  
 responsywnych stron, 286  
 sieci, 211  
 skalowalności, 211  
 SMI, 280  
 tablic ortogonalnych, OAT, 289, 291  
 ukierunkowane na
 

- urządzenia, 164

 użyteczności, 169, 211  
 w branży lotniczej, 285  
 w chmurze, 179  
 w cyklu agile/DevOps, 77, 123, 124  
 w oparciu o model, 213  
 w przyszłości, 309  
 w środowiskach zwinnych, 69, 74  
 wdrożenia reguł
 

- biznesowych, 285

 wrażeń klienta, 29  
 wydajności, 124, 169
 

- mikrousług, 126
- na żądanie, 185, 187

 zabezpieczeń, 211  
 testy
 

- akceptacyjne systemów, 280
- crowdsourcingowe, 52
- dostępności, 149

dynamiczne, 216  
 eksploracyjne, 124  
 funkcjonalne, 211  
 jednostkowe, 124  
 kognitywne, 220, 221  
 kontraktowe, 124  
 niefunkcjonalne, 286  
 oparte na modelu, 216  
 regresyjne, 143
 

- przypadki testowe, 145

 skuteczne, 275  
 weekendowe
 

- wady, 59
- wdrażanie, 61
- zalety, 58

 zwinne, 75, 312  
 transformacja
 

- cyfrowa, 25
- QA, 26

 tworzenie całościowych testów
 

- wydajności, 186

## U

uczenie maszynowe, 315  
 UI, interfejs użytkownika, 84  
 urządzenia noszone, 303  
 usługi
 

- fintech, 227
- komunalne, 278
- pay-per-use, 48
- TDM, 118
- zarządzane, 42
  - ekonomia skali, 42
  - kalkulator oszczędności, 43
  - odbiór, 42
  - opis korzyści, 43
  - wydajność kosztowa, 42
  - zachowanie wiedzy, 42
  - zarządzania konfiguracją, 111

 UX, user experience, 163, 302

## W

walidacja
 

- interfejsów, 194
- migracji aplikacji, 176

 wczesna automatyzacja
 

- integracji, 84

 wdrażanie technologii
 

- kognitywnych, 224

 weryfikacja jakości danych, 194  
 wielokrotne wykorzystanie
 

- zasobów, 216
- obiektów, 154
- procesów, 240

 wirtualizacja usług, 137, 140  
 WQR, World Quality Report, 25  
 rewolucja IoT, 209  
 stan akceptacji zwinności, 69  
 trendy DevOps, 99  
 wrażenia użytkownika, UX, 163  
 wskaźniki środowisk testowych, 133  
 wydajność, 25, 153
 

- kosztowa, 42
- procesora, 158
- środowiska, 133

 wydatki na QA, 31  
 wytyczne dotyczące dostępu do sieci, 149

## X

XP, 65

## Z

zaawansowana infrastruktura
 

- pomiarowa, 279

 zabezpieczanie aplikacji, 181  
 zachowanie wiedzy, 42  
 zakupy wielokanałowe, 259  
 zapewnianie jakości, QA, 19  
 zarządzane usługi testowe, MTS, 39

zarządzanie  
  bezpieczeństwem aplikacji, 196  
  danymi liczników, 279  
  danymi testowymi, 115  
  incydentami, 197  
  konfiguracją, CM, 110  
  łańcuchem dostaw, 236, 237  
  poprawkami, 197

  pulami obiektów, 155  
  środowiskiem testowym, 129  
  zmianami konfiguracji, 197

zastępowanie podwykonawców, 42

zawody testerskie, 310

zdalne monitorowanie pacjentów, 244

zespoły hybrydowe QA, 34

zużycie pamięci, 158

zwinne sprinty, 66

zwinność, 63, 64

  środowiska, 133

  w środowiskach rozproszonych, 68

# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

## Podoba Ci się ten kod? Przetestuj, zanim pochwalisz!

Doświadczeni deweloperzy wiedzą, że testowanie oprogramowania jest procesem żmudnym i wymagającym, niemniej rzetelnie przeprowadzone testy są konieczne, aby uznać kod za godny zaufania. Co więcej, testowanie kodu jest tą gałęzią techniki, która rozwija się bardzo dynamicznie. Ma to związek z rosnącymi konsekwencjami ewentualnych incydentów, przerw w działaniu oprogramowania czy wręcz ataków hakerskich. Oznacza to, że tester musi być na bieżąco z nowinkami ze swojej branży.

Ta książka jest nieodzowną pomocą każdego testera. Stanowi wyczerpujące źródło potrzebnych informacji, prezentuje przeróżne perspektywy, praktyki, trendy, narzędzia i rozwiązania związane z testowaniem aplikacji. Omówiono różne modele działań testowych i cykle życia oprogramowania. Pokazano nieoczywiste aspekty analizy jakości oraz prowadzenia testów w przypadku mediów społecznościowych, urządzeń mobilnych, danych analitycznych i technologii chmury. Opisano technologie, które najprawdopodobniej będą kształtować przyszłość testów oprogramowania. W publikacji nie zabrakło informacji o najlepszych praktykach, które warto stosować w celu zapewnienia stabilności i bezpieczeństwa oprogramowania.

### W tej książce między innymi:

- najnowsze trendy w dziedzinie działań testowych i modeli operacyjnych
- automatyzacja testów
- uczenie maszynowe i systemy kognitywne: testowanie i wykorzystywanie w testach
- testowanie na potrzeby różnych branż: ochrony zdrowia, marketingu cyfrowego i innych
- testowanie aplikacji na potrzeby rozwoju IoT (internetu rzeczy)

**Renu Rajani** ma ponad 25 lat doświadczenia w dziedzinie usług IT. Wspecjalizowała się w takich zagadnieniach jak dostawa, przekształcenia, dostarczanie rozwiązań technicznych i zarządzanie outsourcingiem. Pracowała dla IBM, Citi, Capgemini i KPMG Consulting; była nagradzana za osiągnięcia w zakresie technik testowania oprogramowania. Jest też starszym menedżerem projektu IBM z certyfikacją DPE/SM. Ma certyfikaty: ITIL V3, CAIIB oraz PMP.

	<i>Sprawdź nasze szkolenia</i>	<b>KOD KORZYŚCI</b> Sięgnij po więcej! ▶	
 <a href="http://helion.pl">helion.pl</a>		ISBN 978-83-283-3970-5	
 <b>0 801 339900</b>	AKADEMIA IT & BUSINESS		
 <b>0 601 339900</b>	<a href="http://WWW.SZKOLENIA.HELION.PL">WWW.SZKOLENIA.HELION.PL</a>	9 788328 339705	
<b>INFORMATYKA W NAJLEPSZYM WYDANIU</b>		Cena: 59,00 zł	

**Packt**