

**Site reliability engineering : jak Google zarządza systemami produkcyjnymi / Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy. – Gliwice, cop. 2018**

Spis treści

<b>Przedmowa</b>	<b>13</b>
<b>Wstęp</b>	<b>15</b>
<b>Część I. Wprowadzenie</b>	<b>23</b>
<b>1. Wprowadzenie</b>	<b>25</b>
Zarządzanie usługami przez administratorów systemów	25
Podejście do zarządzania usługami w Google'u — Site Reliability Engineering	26
Założenia podejścia SRE	29
Koniec początku	34
<b>2. Środowisko produkcyjne w Google'u z perspektywy SRE</b>	<b>35</b>
Sprzęt	35
Oprogramowanie „organizujące” pracę sprzętu	37
Inne systemy oprogramowania	40
Infrastruktura dla oprogramowania	40
Nasze środowisko programistyczne	41
Shakespeare — przykładowa usługa	42
<b>Część II. Zasady</b>	<b>45</b>
<b>3. Akceptowanie ryzyka</b>	<b>47</b>
Zarządzanie ryzykiem	47
Pomiar ryzyka związanego z usługą	48
Tolerancja ryzyka dla usługi	50
Uzasadnienie stosowania budżetu błędów	54
<b>4. Poziomy SLO</b>	<b>59</b>
Terminologia związana z poziomem usług	59
Wskaźniki w praktyce	62
Cele w praktyce	65
Umowy w praktyce	68
<b>5. Eliminowanie harówki</b>	<b>69</b>
Definicja harówki	69
Dlaczego ograniczenie harówki jest korzystne?	71

Co kwalifikuje się jako prace inżynierskie?	72
Czy harówka zawsze jest czymś złym?	72
Wniosek	74
<b>6. Monitorowanie systemów rozproszonych</b>	<b>75</b>
Definicje	75
Po co monitorować?	76
Wyznaczanie rozsądnych oczekiwań względem monitorowania	77
Symptomy a przyczyny	78
Monitorowanie czarnoskrzynkowe i białoskrzynkowe	79
Cztery złote sygnały	79
Uwzględnianie wartości skrajnych (lub narzędzia pomiarowe i sprawność)	81
Określanie odpowiedniej szczegółowości pomiarów	81
Tak proste jak to możliwe, ale nie prostsze	82
Łączenie opisanych zasad	82
Monitorowanie długoterminowe	83
Wnioski	86
<b>7. Ewolucja automatyzacji w Google'u</b>	<b>87</b>
Wartość automatyzacji	87
Wartość dla zespołów SRE w Google'u	89
Przypadki zastosowania automatyzacji	90
Wyautomatyzuj się z pracy — automatyzuj WSZYSTKO!	93
Łagodzenie problemów — stosowanie automatyzacji do uruchamiania klastrów	95
Borg — narodziny komputera na skalę hurtowni	101
Podstawową cechą jest niezawodność	102
Zalecenia	103
<b>8. Inżynieria udostępniania</b>	<b>105</b>
Rola inżyniera udostępniania	105
Filozofia	106
Ciągłe budowanie i wdrażanie	108
Zarządzanie konfiguracją	111
Wnioski	112
<b>9. Prostota</b>	<b>115</b>
Stabilność a elastyczność systemu	115
Cnota nudy	116
Nie oddam mojego kodu!	116
Wskaźnik „negatywne wiersze kodu”	117
Minimalne interfejsy API	117
Modułowość	117
Udostępnianie prostych zmian	118
Prosty wniosek	118

<b>Część III. Praktyki</b>	<b>119</b>
<b>10. Praktyczne alarmy na podstawie szeregów czasowych</b>	<b>123</b>
Powstanie systemu Borgmon	124
Narzędzia pomiarowe w aplikacji	125
Zbieranie eksportowanych danych	126
Przechowywanie danych w obszarze szeregów czasowych	126
Sprawdzanie reguł	129
Alarmy	132
Podział systemu monitorowania	133
Monitorowanie czarnoskrzynkowe	134
Zarządzanie konfiguracją	135
Dziesięć lat później...	136
<b>11. Dyżury na wezwanie</b>	<b>139</b>
Wprowadzenie	139
Życie inżyniera dyżurnego	140
Zrównoważone dyżury	141
Poczucie bezpieczeństwa	142
Unikanie nieodpowiedniego obciążenia operacyjnego	144
Wnioski	145
<b>12. Skuteczne rozwiązywanie problemów</b>	<b>147</b>
Teoria	148
Praktyka	150
Wyniki negatywne to magia	156
Studium przypadku	158
Ułatwianie rozwiązywania problemów	162
Wnioski	162
<b>13. Reagowanie kryzysowe</b>	<b>163</b>
Co robić, gdy w systemie wystąpi awaria?	163
Kryzys wywołany testami	164
Sytuacje kryzysowe spowodowane zmianami	165
Sytuacje kryzysowe spowodowane procesem	167
Dla każdego problemu istnieje rozwiązanie	169
Wyciągaj wnioski z przeszłości i nie powtarzaj tych samych błędów	170
Wnioski	170
<b>14. Zarządzanie incydentami</b>	<b>173</b>
Niezarządzane incydenty	173
Anatomia niezarządzanego incydentu	174
Aspekty procesu zarządzania incydentami	175
Zarządzany incydent	176
Kiedy ogłaszać incydent?	177
Podsumowanie	178

<b>15. Kultura analizy zdarzeń – wyciąganie wniosków z niepowodzeń</b>	<b>179</b>
Filozofia analizy zdarzeń w Google'u	179
Współpracuj i dziel się wiedzą	181
Wprowadzanie kultury analizy zdarzeń	182
Wnioski i wprowadzane usprawnienia	184
<b>16. Śledzenie przestojów</b>	<b>185</b>
Escalator	185
Outalator	186
<b>17. Testowanie niezawodności</b>	<b>191</b>
Rodzaje testów oprogramowania	192
Tworzenie środowiska testowania i środowiska budowania	198
Testowanie w dużej skali	199
Wnioski	210
<b>18. Inżynieria oprogramowania w SRE</b>	<b>211</b>
Dlaczego inżynieria oprogramowania w zespołach SRE ma znaczenie?	211
Studium przypadku. Auxon — wprowadzenie do projektu i przestrzeń problemowa	213
Planowanie przepustowości na podstawie celów	215
Wspomaganie inżynierii oprogramowania w SRE	223
Wnioski	227
<b>19. Równoważenie obciążenia na poziomie frontonu</b>	<b>229</b>
Moc nie jest rozwiązaniem	229
Równoważenie obciążenia z użyciem systemu DNS	230
Równoważenie obciążenia na poziomie wirtualnych adresów IP	232
<b>20. Równoważenie obciążenia w centrum danych</b>	<b>235</b>
Scenariusz idealny	236
Identyfikowanie problematycznych zadań — kontrola przepływu i „kulawe kaczki”	237
Ograniczanie puli połączeń za pomocą tworzenia podzbiorów	239
Reguły równoważenia obciążenia	244
<b>21. Obsługa przeciążenia</b>	<b>251</b>
Pałapki związane z „liczbą zapytań na sekundę”	251
Limity na klienta	252
Ograniczanie liczby żądań po stronie klienta	253
Poziom krytyczności	255
Sygnały poziomu wykorzystania	256
Obsługa błędów przeciążenia	257
Obciążenie wynikające z połączeń	260

Wnioski	261
<b>22. Radzenie sobie z awariami kaskadowymi</b>	<b>263</b>
Przyczyny awarii kaskadowych i projektowanie z myślą o ich uniknięciu	264
Zapobieganie przeciążeniu serwerów	268
Powolny rozruch i pusta pamięć podręczna	276
Warunki wywołujące awarie kaskadowe	279
Testowanie pod kątem awarii kaskadowych	280
Pierwsze kroki w obliczu awarii kaskadowych	283
Uwagi końcowe	285
<b>23. Zarządzanie krytycznym stanem — zapewnianie niezawodności za pomocą konsensusu w środowisku rozproszonym</b>	<b>287</b>
Uzasadnienie uzgadniania konsensusu — niepowodzenie koordynacji systemów rozproszonych	289
Jak działa konsensus w środowisku rozproszonym?	291
Wzorce architektury systemu związane z konsensem w środowisku rozproszonym	292
Wydajność uzgadniania konsensusu w środowisku rozproszonym	297
Wdrażanie rozproszonego systemu opartego na konsensusie	305
Monitorowanie rozproszonych systemów uzgadniania konsensusu	312
Wnioski	314
<b>24. Okresowe szeregowanie prac w środowisku rozproszonym za pomocą crona</b>	<b>315</b>
cron	315
Prace crona a idempotencja	316
cron w dużej skali	317
Budowanie crona w Google'u	318
Podsumowanie	325
<b>25. Potoki przetwarzania danych</b>	<b>327</b>
Początki wzorca projektowego „potok danych”	327
Początkowy wpływ big data na prosty wzorzec potoku danych	327
Wyzwania związane ze wzorcem „okresowo uruchamiany potok danych”	328
Problemy powodowane przez nierównomierny podział pracy	328
Wady okresowo uruchamianych potoków w środowiskach rozproszonych	329
Wprowadzenie do systemu Workflow Google'a	332
Etapy wykonywania w systemie Workflow	334
Zapewnianie ciągłości biznesowej	335
Podsumowanie i uwagi końcowe	336
<b>26. Integralność danych — wczytywanie tego, co zostało zapisane</b>	<b>337</b>
Ścisłe wymagania z zakresu integralności danych	338

Cele zespołów SRE w Google'u w zakresie integralności i dostępności danych	342
Jak zespoły SRE Google'a radzą sobie z problemami z integralnością danych?	346
Studia przypadków	357
Ogólne zasady SRE stosowane w obszarze integralności danych	363
Wnioski	365
<b>27. Niezawodne udostępnianie produktów w dużej skali</b>	<b>367</b>
Inżynieria koordynowania udostępniania	368
Konfigurowanie procesu udostępniania	370
Tworzenie listy kontrolnej udostępniania	373
Wybrane techniki niezawodnego udostępniania	377
Powstawanie zespołu LCE	381
Wnioski	384
<b>Część IV. Zarządzanie</b>	<b>385</b>
<b>28. Szybkie przygotowywanie inżynierów SRE do dyżurów i innych zadań</b>	<b>387</b>
Zatrudniłeś nowych inżynierów SRE. Co dalej?	387
Początkowe pouczające doświadczenia — argument na rzecz przewagi struktury nad chaosem	389
Rozwój świetnych ekspertów od inżynierii odwrotnej i improwizatorów	393
Pięć praktyk dla przyszłych dyżurnych	395
Dyżury i inne zadania — rytuał przejścia i ciągłe uczenie się	400
Końcowe myśli	401
<b>29. Radzenie sobie z zakłóceniami</b>	<b>403</b>
Zarządzanie obciążeniem operacyjnym	404
Czynniki wpływające na sposób zarządzania zakłóceniami	404
Niedoskonałe maszyny	405
<b>30. Angażowanie inżyniera SRE w celu wyeliminowania przeciążenia operacyjnego</b>	<b>411</b>
Etap 1. Poznaj usługę i kontekst	412
Etap 2. Przedstawianie kontekstu	414
Etap 3. Motywowanie do zmian	415
Wnioski	417
<b>31. Komunikacja i współpraca w zespołach SRE</b>	<b>419</b>
Komunikacja — spotkania produkcyjne	420
Współpraca w ramach zespołów SRE	423
Studium przypadku z obszaru współpracy w zespołach SRE — Viceroy	425
Współpraca z zespołami innymi niż SRE	429
Studium przypadku — przeniesienie DFP do F1	430

Wnioski	432
<b>32. Zmiany w modelu angażowania się zespołów SRE</b>	<b>433</b>
Zaangażowanie zespołów SRE — co, jak i dlaczego?	433
Model PGP	434
Model angażowania się zespołów SRE	434
Przeglądy gotowości produkcyjnej — prosty model oparty na PGP	436
Ewolucja prostego modelu opartego na PGP — wczesne zaangażowanie	439
Zmiany w rozwoju usług — frameworki i platforma SRE	441
Wnioski	446
<b>Część V. Wnioski</b>	<b>447</b>
<b>33. Lekcje z innych branż</b>	<b>449</b>
Poznaj naszych branżowych weteranów	450
Testowanie gotowości i odporności na katastrofy	451
Kultura analizy zdarzeń	454
Eliminowanie powtarzalnej pracy i kosztów operacyjnych dzięki automatyzacji	456
Ustrukturyzowane i racjonalne podejmowanie decyzji	457
Wnioski	459
<b>34. Podsumowanie</b>	<b>461</b>
<b>A Tabela dostępności</b>	<b>465</b>
<b>B Zbiór dobrych praktyk dotyczących usług produkcyjnych</b>	<b>467</b>
<b>C Przykładowy dokument ze stanem incydentu</b>	<b>473</b>
<b>D Przykładowa analiza zdarzenia</b>	<b>475</b>
<b>E Lista kontrolna LCE</b>	<b>479</b>
<b>F Przykładowe notatki ze spotkania produkcyjnego</b>	<b>481</b>
<b>Bibliografia</b>	<b>483</b>
<b>Skorowidz</b>	<b>494</b>