



Tłumaczenie publikacji "99 rzeczy, które możesz zrobić, aby stać się lepszym testerem".¹

1. Poznaj swojego klienta. Kto płaci za Twoją pracę i czego od tej pracy oczekuje.
2. Popracuj jakiś czas na pierwszej linii wsparcia. Dowiesz się więcej o ludziach, którzy używają Twojego oprogramowania.
3. Nigdy nie przestawaj się uczyć.
4. Uświadom sobie, że nigdy nie dowiesz się wszystkiego.
5. Naucz się, kiedy warto automatyzować, a kiedy nie.
6. Każdego dnia kwestionuj to, w jaki sposób pracujesz.
7. Miej odwagę dzielić się Twoimi odkryciami z każdym, kto rządzi i walczy o ważne zmiany.
8. Pracuj w parze z programistą.
9. Testuj tak wcześnie, jak to możliwe.
10. Pamiętaj, że zawsze chodzi o ludzi.
11. Naucz się pytać/kwestionować.
12. Naucz się wyjaśniać/tłumaczyć.
13. Zrozum model biznesowy i wyzwania biznesowe.
14. Nie trać z oczu końcowych celów.
15. Zrozum myślenie systemowe - świat jest zbiorem systemów, które na siebie oddziałują.
16. Zrozum modelowanie mentalne, metody naukowe oraz projektowanie eksperymentów.

¹ Źródło publikacji: <http://www.ministryoftesting.com/wp-content/uploads/2013/07/99ThingsEbook.pdf>



17. Bądź krytyczny, ale nie krytykuj.
18. Twórz mapy myśli (mindmap).
19. Ucz się na błędach innych...
20. ...ale najpierw ucz się na własnych błędach.
21. Bierz udział albo przemawiaj na testerskich i programistycznych konferencjach.
22. Zrób kurs(y) "Black Box Software Testing".
23. Śledź innych testerów na Twitterze.
24. Dziel się ideami i doświadczeniami, zbieraj informację zwrotną i ćwicz swoje umiejętności debatowania.
25. Ucz się innych języków. W mowie i piśmie.
26. Ucz innych testowania.
27. Naucz się programowania.
28. Zaczynij blog testerski.
29. Komentuj na blogach testerskich.
30. Czytaj artykuły, blogi, fora.
31. Dołącz do projektu open source jako tester.
32. Dołączyłeś już do "Software Testing Club!"?
33. Słuchaj tego, co klient ma do powiedzenia.
34. Słuchaj swojego zespołu. Nie tylko po to, aby ich zrozumieć, ale również po to, by dowiedzieć się, czego nie rozumieją.
35. Przeczytaj sylabus ISTQB od początku do końca. Pomyśl o nim jako o zamkniętej skrzynce. A teraz myśl "out-of-box".
36. Nie unikaj technicznych rozmów i informacji.
37. Powtarzaj sobie regularnie podstawy.
38. Zawsze używaj najlepszej wymyślonej metodologii z zachowaniem zdrowego rozsądku.
39. Zbuduj dobre relacje z programistami.
40. Podejmij ryzyko samodzielnego myślenia.
41. Naucz się używać technik takich jak analiza wartości granicznych, klasy równoważności, itd.
42. Eksploruj własne idee poprzez wpisy na blogu, dyskusje, wystąpienia na wydarzeniach. Wysłuchaj opinii innych osób i użyj ich do rozszerzenia własnych.
43. Czytaj noty wydania produktu, którego używasz (lub nawet takich, których nie używasz). Zawierają listę naprawionych defektów. Sprawdź czy te defekty nie występują w twoim produkcie.
44. Ucz się z defektów, które znajdowane są na produkcji. Dowiedz się, jak to możliwe, że testowanie je ominęło.
45. Słuchaj. Wtedy twórz własną opinię.



46. Miej swój umysł otwarty na nowe techniki i narzędzia.
47. Zastanów się nad trzema rzeczami, jakie twój szef, kolega z pracy i twój uczeń mogą zrobić lepiej w ich pracy... i zastosuj te usprawnienia sam.
48. Czytaj o testowaniu tak dużo, ile dasz radę. Następnie przeczytaj tak dużo, ile dasz radę publikacji z obszarów spoza testowania: projektowanie, tworzenie kodu, psychologia, antropologia, itd.
49. Kiedy analizujesz system nie zapominaj o tej jego części, którą jest człowiek.
50. Eksploruj system!
51. Sprawdź produkty konkurencji i porównaj je ze swoim.
52. Nie ufaj nikomu! Nie każdy mówi dokładnie to, co naprawdę myśli. Czym więcej rozmawiasz, tym więcej wierzysz, ale nigdy nie ufasz na więcej niż 95%.
53. Znajdź odpowiedź na pytania: dlaczego potrzebujemy testerów? co to znaczy "dobre testowanie"? co powinienem udokumentować, dlaczego i dla kogo...? Najważniejsza nie jest odpowiedź, ale proces docierania do niej.
54. Zrozum swojego klienta i to, czego potrzebuje.
55. Pamiętaj, że ludzie nie zawsze potrzebują tego, co myślą, że potrzebują. Przygotuj się na analizowanie "wokół" problemu, aby znaleźć dla niego rozwiązanie. Bez bezpośredniego zadawania pytań.
56. Jeśli społeczność nie istnieje, zbuduj ją sam. Miejsce, w którym będziesz mógł/mogła rozmawiać o testowaniu, dzielić się pomysłami i stawać się lepszym testerem.
57. Wykorzystaj swoją odwagę do obrony swoich racji. Upewnij się jednak, że twoja hipoteka jest ubezpieczona :)
58. Stosuj myślenie jakościowe jak styl życia, nie ograniczaj się jedynie do godzin pracy.
59. Żyj jak tester. Bądź testerem. Bądź testerem w każdym aspekcie twojego życia. W pracy, w życiu prywatnym, w związku i w hobby. 24/7, 365 dni w roku (jeden dzień odpoczynku w roku z 366 dniami).
60. Nie przestawaj stawać się lepszym testerem!
61. Spróbuj dostarczać rozwiązanie tak jak znajdujesz problem.
62. Myśl o tylu perspektywach jak to tylko możliwe.
63. Kiedy raportujesz, unikaj bycia postrzeganym jako wróg. Wiń raczej systemy i aplikacje niż jednostki. Trzymaj się faktów. Podkreślaj pozytywne i wysiłek potrzebny do poprawienia problemu przez osoby.
64. Naucz się różnicy między ważnością a priorytetem.
65. Upewnij się, że wszyscy w organizacji znają różnicę między ważnością a priorytetem.
66. Zaakceptuj, że nie wszystkie zaraportowane przez Ciebie defekty zostaną naprawione.
67. Każdy w projekcie jest odpowiedzialny za jakość.
68. Testowanie może być przyjemne, więc pozostań pozytywny i zarażaj entuzjazmem testowania.
69. Zaakceptuj, że programiści mają inne nastawienie (mindset).

70. Bądź odważny. Jesteś prawdopodobnie jedyną osobą mówiącą: "Czy jesteś pewien, że chcesz wydać ten produkt...?"
71. Nie powtarzaj się. Nie powtarzaj tych samych akcji, nie idź tą samą ścieżką, w tej samej kolejności. Złam swoje przyzwyczajenia.
72. Poproś doświadczonego testera o informację zwrotną.
73. Jeśli jesteś doświadczonym testerem, pomagaj tym mniej doświadczonym testerom. Tyle, ile potrafisz. Pomożesz im, pomożesz sobie.
74. Jeśli jesteś doświadczonym testerem, opowiedz tym mniej doświadczonym testerom o swoim problemie (oczywiście zawodowym). Mają świeże, nowe spojrzenie.
75. Redefiniuj boks, aby pomóc sobie wyjść z boksu (out-of-box).
76. Pozyskaj perspektywę innej osoby do pomocy.
77. Zostaw swoje EGO w domu - może ten doskonały defekt nie zostanie naprawiony przed wydaniem softu. Zaufaj osobie, która podejmuje taką decyzję - wie ona lepiej niż ty, dlaczego nie warto tego poprawiać.
78. Zaakceptuj fakt, że będąc testerem będzie się wymagało od Ciebie więcej niż tylko bycia testerem. Ktoś może Cię zapytać czy ten soft ma zostać wydany czy też nie.
79. Respektuj programistów, projektantów, właścicieli produktu i wszystkich innych.
80. Zdobądź szacunek wykonując świetną robotę i naucz się komunikować rezultaty swojej pracy.
81. Nigdy nie podsycaj ognia w wojnie między testerami a programistami.
82. Bierz udział w nie testerskich, ale ciągle technologicznych wydarzeniach.
83. Redukuj uprzedzenia i niezamierzoną ślepotę.
84. Naucz się robić efektywne notatki i dokumentować testy na różne sposoby - modele, mapy myśli, szkice i inne, które pomogą Ci zrozumieć testowany obiekt i przyjąć inną perspektywę.
85. Zaplanuj osobisty rozwój w ciągu tygodnia. 5%-10% czasu lub około połowy dnia na pogłębianie wiedzy poprzez czytanie, praktykowanie i uczenie się.
86. Patrz z szerszej perspektywy. W jaki sposób testowanie dodaje wartości twojemu zespołowi, projektowi i organizacji?
87. Naucz się wyrażać regularnych.
88. Naucz się używać linii poleceń. Shell-a i programów wsadowych również.
89. Naucz się języka skryptowego. Naucz się, jak zautomatyzować powtarzalne zadania i procesy.
90. Stań się mistrzem Excel-a. Użyj funkcji, logiki, formatowania warunkowego dla lepszej analizy.
91. Przestań iść zgodnie ze skryptem i zacznij myśleć.
92. Wyjdź na miasto z programistami, projektantami i kierownikami.
93. Zrozum biznes i potrzeby klienta, a nie tylko wymagania.
94. Zrozum obszar domenowy twojej firmy oraz konkurentów na rynku.

95. Nie próbuj znajdować błędów czy defektów - spróbuj znaleźć problemy i ich ofiary. Testowanie to coś więcej niż tylko sprawdzanie.
96. Bądź przygotowany na to, że nie znajdziesz wszystkich defektów, ale próbuj.
97. Bądź przygotowany na to, że nie wszystkie zaraportowane defekty zostaną naprawione.
98. Testuj to, co ma znaczenie.
99. Zakwestionuj pozostałe rady dla testera i rozważ ich słuszność w kontekście twojej pracy.
100. Zawsze próbuj sprawdzić, co się stanie jeśli naruszysz zasady.