

» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl
© Helion 1991-2010

Przetestuj ją sam! Steve Krug o funkcjonalności stron internetowych

Autor: [Steve Krug](#)

Tłumaczenie: Paweł Koronkiewicz

ISBN: 978-83-246-2693-9

Tytuł oryginału: [Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems](#)

Format: B5, stron: 160



Testuj i usuwaj problemy

Niezawodne wskazówki, JAK:

- testować DOWOLNE projekty – od „szkiców na serwetce” po gotowe witryny i aplikacje
- wyszukiwać NAJWAŻNIEJSZE problemy (nigdy nie znajdziesz czasu na usunięcie wszystkich)
- USUWAĆ te problemy, trzymając się zasady „Jak najmniej pracy™”

Testy funkcjonalności to – jak powszechnie wiadomo – niezawodny sposób na udoskonalenie Twoich produktów. Niestety, zwykle koszty takiej operacji zniechęcają do ich prowadzenia. STOP! Nie tym razem! W tym właśnie momencie do akcji wkracza Steve Krug z kontynuacją swojego bestsellera "Nie każ mi myśleć". Tym razem zdradzi Ci triki bardziej przystępnego podejścia do testowania funkcjonalności. Może ich użyć praktycznie każdy twórca witryny, aplikacji czy innego produktu. Nie musisz być ekspertem, żeby zrobić to dobrze. To prostsze, niż myślisz, praktycznie nie grożą Ci błędy i będziesz w stanie zacząć... od teraz!

Po pierwsze dowiesz się, jak ograniczyć proces testowania i udoskonalania aplikacji do elementów kluczowych. Po drugie zapoznasz się z realistycznym podejściem, umożliwiającym przeprowadzanie wczesnego i częstego testowania oraz wykrywania braków w funkcjonalności. Po trzecie nauczysz się skutecznie usuwać problemy. Po czwarte szanujemy Twój czas, dlatego pokażemy Ci, jak wszystko to zrobić szybko, prosto i możliwie bezboleśnie.

- Niedoceniona sztuka samodzielnego testowania funkcjonalności
- Testowanie starej witryny, innych witryn, makiet, projektów stron, prototypów i dalsze badania
- Poszukiwanie i wybór odpowiednich uczestników testów oraz przydzielanie im zadań
- Zasady stosowania najmniej pracochłonnych i najskuteczniejszych metod
- Najważniejsze problemy, na które możesz natrafić, i sposoby ich usuwania
- Testowanie zdalne: szybkie, tanie, choć czasem poza kontrolą
- Przykładowy program testu i formularz zgody na nagrywanie
- Lista lektur dla wszystkich zainteresowanych pogłębieniem wiedzy

Książce towarzyszy witryna www.rocketurgerymadeeasy.com z plikami do ściągnięcia (na przykład wideo demonstrujące test) oraz wszystkimi skryptami i formularzami, które pojawiają się w tekście. Pliki są dostępne dla każdego, kto ma ochotę samodzielnie wykonać testy.

Spis treści

SŁOWO WSTĘPNE	Nazwij mnie niewiernym	6
	<i>Jak powstała ta książka, kilka ostrzeżeń, określenie zakresu tematyki</i>	
<hr/>		
WYSZUKIWANIE PROBLEMÓW Z FUNKCJONALNOŚCIĄ		
<hr/>		
ROZDZIAŁ 1	Nie widzisz tu nigdzie słoni, prawda?	16
	<i>Czym jest samodzielne testowanie funkcjonalności, dlaczego zawsze działa i dlaczego tak rzadko się je przeprowadza</i>	
ROZDZIAŁ 2	Przetnę teraz moją uroczą asystentkę tą oto piłą	24
	<i>Jak wygląda samodzielny test funkcjonalności</i>	
ROZDZIAŁ 3	Jedno przedpołudnie w miesiącu. To wszystko	26
	<i>Plan, który dasz radę zrealizować</i>	
ROZDZIAŁ 4	Co testować i kiedy?	34
	<i>Dlaczego najtrudniej jest wystarczająco wcześnie zacząć</i>	
ROZDZIAŁ 5	Ocena uczestników jest ważniejsza niż ich początkowy wybór	42
	<i>Kogo zapraszać do testów i jak go znaleźć</i>	
ROZDZIAŁ 6	Znajdź im jakieś zajęcie	54
	<i>Wybieranie zadań dla uczestników i pisanie scenariuszy</i>	
ROZDZIAŁ 7	Nudne listy kontrolne	60
	<i>I dlaczego powinieneś ich używać, nawet jeżeli, podobnie jak ja, nie przepadasz za listami kontrolnymi</i>	
ROZDZIAŁ 8	Czytanie w myślach dla każdego	66
	<i>Prowadzenie testu</i>	
ROZDZIAŁ 9	Niech to będą widowiskowe	92
	<i>Jak przyciągnąć i przygotować obserwatorów</i>	

WYSZUKIWANIE PROBLEMÓW Z FUNKCJONALNOŚCIĄ

ROZDZIAŁ 10	Spotkanie podsumowujące <i>Porównywanie notatek i wybieranie problemów do usunięcia</i>	104
ROZDZIAŁ 11	Jak najmniej pracy™ <i>Dlaczego metody najmniej pracochłonne są często najlepsze</i>	112
ROZDZIAŁ 12	Główni podejrzani <i>Kilka problemów, na które możesz natrafić, i sposoby ich usuwania</i>	120
ROZDZIAŁ 13	By rzeczy zmierzały ku lepszemu <i>Sztuka pracy z ludźmi</i>	128

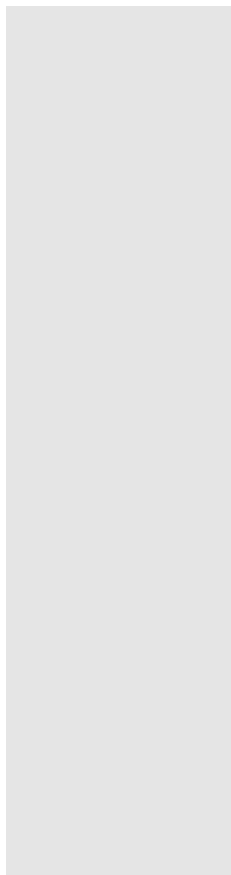
BADANIE FUNKCJONALNOŚCI DLA ZAAWANSOWANYCH

ROZDZIAŁ 14	Teleportacja dla każdego <i>Testowanie zdalne: szybkie, tanie, choć czasem poza kontrolą</i>	134
ROZDZIAŁ 15	Tylko dla zainteresowanych <i>Lista lektur</i>	140
ROZDZIAŁ 16	Szczęśliwej drogi <i>Lista maksym i kilka słów zachęty</i>	144
	Przykładowy program testu i formularz zgody na nagrywanie	146
	Podziękowania	154
	Skorowidz	157

3

Jedno przedpołudnie w miesiącu. To wszystko

PLAN, KTÓRY DASZ RADĘ ZREALIZOWAĆ



Jak wspomniałem w rozdziale 1., nie brakuje dobrych powodów, aby pominąć testy funkcjonalności. Jednak główną przyczyną, dla której są one ignorowane, jest to, że zazwyczaj myśli się o nich jak o wielkim przedsięwzięciu, na miarę co najmniej próby generalnej w teatrze.

Prowadząc warsztaty, wypracowałem pewien schemat, plan działania, na którego realizację może pozwolić sobie każdy, czy to w wielkim przedsiębiorstwie, czy w jednoosobowej firmie. Umożliwi on testowanie dowolnego projektu nawet wiele razy przed jego ukończeniem.

Plan jest wykonalny i prowadzi do celu. Pozwala on wykryć tyle problemów, ile faktycznie jesteśmy w stanie naprawić. Sprzyja też koncentracji na tych najistotniejszych.

Lubię podsumowywać ten „wielki plan” następująco:



Jedno przedpołudnie w miesiącu.
To wszystko.

Plan sprowadza się zasadniczo do jednej rundy testów miesięcznie, z trzema użytkownikami.

W wyznaczonym dniu wykonujemy trzy testy rano i omawiamy je w czasie lunchu. W efekcie koniec lunchu to koniec pracy z badaniem funkcjonalności na najbliższy miesiąc. Jednocześnie dysponujesz informacjami o tym, co powinno zostać naprawione przed kolejnym takim dniem¹.

Czerwiec						
N	P	W	Ś	C	P	Ś
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Lipiec						
N	P	W	Ś	C	P	Ś
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Sierpień						
N	P	W	Ś	C	P	Ś
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Czwartek, 24 czerwca	
10:00	Pierwsza sesja
11:00	Druża sesja
12:00	Tercia sesja
13:00	Spotkanie podsumowujące

¹ Jeśli projekt jest realizowany na zasadach Agile Development, nie panikuj. Patrz strona 30.

Kluczowe są dwa słowa:

- > **Rano.** Ograniczenie testów do połowy dnia — skąd wynika ograniczenie do trzech użytkowników — upraszcza rekrutację uczestników i umożliwia przyjscie tym, którzy chcą lub powinni obejrzeć przebieg badania.
- > **Miesiąc.** „Co miesiąc” okazuje się dość dobrym wyborem częstości testowania. Na więcej i tak zazwyczaj nie można sobie pozwolić, a uzyskujemy wystarczająco dużo informacji o problemach, aby mieć co robić do następnego razu.

Jeżeli ogłosisz, że w trzeci czwartek każdego miesiąca będziesz zapraszał osoby z zewnątrz, to wytworzysz pewną atmosferę oczekiwania, która sprawi, że pojawią się osoby zainteresowane obejrzeniem przebiegu testów, a programiści dostarczą coś gotowego do poddania weryfikacji.

Wprowadzenie takiej rutyny i stałego dnia uwalnia od kwestii podejmowania decyzji o tym, *kiedy* testować. Po prostu testujesz to, co jest w danym dniu gotowe. Pozostawienie kwestii terminu otwartej, podlegającej decyzji, którą dopiero trzeba podjąć, może mieć tylko jeden skutek — zmniejszenie częstości testów.

Samodzielny test a Wielkie Testowanie

Zasada „jedno przedpołudnie w miesiącu” to nie tylko kwestia harmonogramu. Takie określenie podejścia do zagadnienia przypominać ma także o zachowaniu prostoty — prostoty, która pozwala wykonywać badanie naprawdę często i regularnie.

Samodzielne testowanie nie daje wszystkich korzyści, jakie można by uzyskać w wyniku Wielkiego Testu Funkcjonalności, ale pozwala uzyskać potrzebne wyniki za przystępną cenę. Oto przegląd różnic (wszystkie elementy „od – do” zostaną omówione szerzej w dalszych rozdziałach):

	WIELKIE TESTOWANIE FUNKcjONALNOŚCI	SAMODZIELNY TEST
CZAS TRWANIA RUNDY TESTÓW	1 – 2 dni testów, tydzień na przygotowanie spotkania podsumowującego, a potem pewien proces wyboru elementów do naprawy.	Jedno przedpołudnie w miesiącu obejmuje testowanie, omówienie wyników i podjęcie decyzji o elementach do naprawy. Wczesnym popołudniem masz testowanie funkcjonalności na miesiąc z głową.

	WIELKIE TESTOWANIE FUNKCJONALNOŚCI	SAMODZIELNY TEST
KIEDY TESTOWANIE JEST PRZEPROWADZANE?	Gdy witryna jest prawie skończona.	Regularnie, w trakcie pracy nad witryną.
LICZBA RUND TESTÓW	Zazwyczaj jedna lub dwie w całym projekcie, zależnie od dostępnego czasu i budżetu.	Po jednej na każdy miesiąc pracy.
LICZBA UCZESTNIKÓW W KAŻDEJ RUNDZIE	5 – 8, czasem 10, żeby przekonać sceptycznego menedżera.	Trzy.
KTO JEST UCZESTNIKIEM?	Precyzyjnie dobrane osoby reprezentujące grupę docelową.	Swobodnie wybrane osoby. Regularność testowania jest ważniejsza niż dobór „prawdziwych” użytkowników.
MIEJSCE TESTÓW	Poza miejscem pracy, w wynajętym studiu z pokojem obserwacyjnym i lustrem weneckim.	W miejscu pracy, z obserwatorami w dowolnym pokoju konferencyjnym korzystającymi z oprogramowania do udostępniania zawartości ekranu.
KTO OBSERWUJE?	2 – 3 dni poza miejscem pracy oznaczają, że niewiele osób uzyskuje informacje „z pierwszej ręki”.	Pół dnia w miejscu pracy oznacza, że więcej osób może zobaczyć testowanie „na żywo”.
RAPORTY	Pewna osoba musi mieć co najmniej tydzień na przygotowanie podsumowania.	Jedno-, dwustronicowy e-mail podsumowuje decyzje podjęte w trakcie spotkania.
KTO IDENTYFIKUJE PROBLEMY?	Osoba prowadząca testy zazwyczaj analizuje wyniki i zaleca zmiany.	Cały zespół programistów i projektantów oraz wszelkie inne zainteresowane osoby porównują notatki i tego samego dnia w porze lunchu decydują o tym, co zmienić.
GLÓWNY CEL	Lista obejmująca wiele (czasem setki) problemów podzielonych na kategorie i uporządkowanych według ważności.	Krótką listą najistotniejszych problemów i postanowienie usunięcia ich przed kolejną rundą testów.
ZAPIS WIDEO TWARZY UCZESTNIKA W TRAKCIE TESTÓW	Tak. Obserwatorzy muszą widzieć reakcje uczestnika na różne elementy (zwłaszcza momenty zniechęcenia).	Nie. Obserwacja czynności uczestnika na ekranie i słuchanie, co mówi, jest wszystkim, czego potrzebujemy.
KOSZT	Od 5 000 do 15 000 dolarów za rundę przy wynajmowaniu specjalisty z zewnątrz.	Kilkaset dolarów na rundę.

FAQ

Czy naprawdę można to zrobić w jedno przedpołudnie w miesiącu?

Nie całkiem. Nie wszystko. W pół dnia można zrobić *testy* i *spotkanie*. Dla większości członków zespołu na tym się kończy.

Jednak jako osoba prowadząca przed każdą rundą testów musisz się przygotować: wybrać, co ma być testowane, określić zadania, napisać scenariusze, wybrać uczestników i zadbać o to, aby wszystkie osoby, których obecność jest pożądana, były faktycznie obecne.

Gdy będziesz to robił po raz pierwszy, przygotowania zajmą Ci dwa albo trzy pełne dni. Jednak w kolejnych rundach powinieneś zmieścić się w dwóch dniach, a nawet w jednym.

Czy mogę to robić częściej niż raz w miesiącu?

Zdecydowanie tak. Jedno przedpołudnie w miesiącu to minimum. Cokolwiek projektujesz, skorzysta na każdej ilości testów, jaką jesteś w stanie przeprowadzić.

Ważną rzeczą jest to, aby nie testować *rzadziej* niż raz w miesiącu. Gdy tylko zaprzestasz testów w trzeci czwartek każdego miesiąca, powrócisz do problemu podjęcia *decyzji* o tym, kiedy wykonać testowanie funkcjonalności, a to oznacza, że drastycznie spadnie prawdopodobieństwo tego, że zostanie ono faktycznie przeprowadzone.

Stosujemy Agile Development. Jedno przedpołudnie w miesiącu? Ha!

A, tak. Agile². Przy krótkich cyklach, z jakimi mamy do czynienia w środowisku Agile, czekanie przez miesiąc brzmi jak czekanie na zupełnie inny produkt. Może w tym przypadku maksyma powinna raczej brzmieć: „Jedno przedpołudnie co sprint, nic więcej”.

Można powiedzieć, że samodzielne testowanie funkcjonalności doskonale wpasowuje się w metodologię Agile, opartą na szybkim wytwarzaniu działających części produktu i przedstawianiu ich użytkownikom. Jedynym problemem jest to, że w wielu przypadkach ci „użytkownicy” to w rzeczywistości członkowie zespołu, a nie osoby z zewnątrz (to jeszcze skorygujesz).

² Dla uproszczenia używam terminów „Agile” i „Agile Development”, ale warto podkreślić, że mam przy tym na myśli dowolne z wielu popularnych metodologii wytwarzania oprogramowania, które charakteryzują się częstymi, krótkimi cyklami produkcyjnymi oraz tym, że iteracyjność i adaptacyjność ma wyższy priorytet niż długofalowe planowanie.

Ponieważ będziesz testował częściej niż raz w miesiącu, możesz proces skrócić jeszcze bardziej (na przykład wybierając dwóch uczestników zamiast trzech), a część rund wykonywać zdalnie (patrz rozdział 14.), co pozwoli zaoszczędzić wiele czasu. Jednak poza tym sam proces nie ulega znaczącym zmianom.

Największym wyzwaniem w trakcie testowania funkcjonalności w środowisku Agile jest chyba to, że zmusza ono do wyprzedzania szybko poruszających się programistów, którzy nie mają czasu na przygotowywanie prototypów (wszystko, co piszą, jest z założenia działającym kodem).

Oznacza to najprawdopodobniej, że będziesz poświęcał część swojego czasu na tworzenie prototypów tego, co będzie realizowane dopiero w następnym sprincie. W efekcie w każdej rundzie będziesz testował to, co powstało w poprzednim sprincie, i stworzony na papierze prototyp tego, co powstanie w następnym.

Czy to musi być przedpołudnie?

Nie ma żadnej szczególnej magii w tym, że wybrałem przedpołudnie, a nie popołudnie. Niektórzy uczestnicy mogą nie mieć czasu na udział w testach w godzinach pracy, więc mogą one odbywać się o 6, 7 czy 8 wieczorem (frekwencję obserwatorów można podwyższyć wtedy ofertą gastronomiczną). Spotkanie podsumowujące można odłożyć wtedy na porę śniadania lub lunchu następnego dnia.

Ważną rzeczą jest to, aby starać się wykonać wszystkie testy w ciągu kilku godzin. Pozwala to zgromadzić największą liczbę obserwatorów. Spotkanie podsumowujące też nie powinno być odkładane, bo istotne jest to, aby dysponować w trakcie jego trwania jak największą liczbą świeżo zapamiętanych szczegółów.

Co odpowiedzieć na stwierdzenie: „Ale jeśli testujesz tylko po trzy osoby, to badanie nie ma wartości statystycznej. Niczego w ten sposób nie dowiedziesz?”

Odpowiedz tak:

„Masz całkowitą rację. Testowanie z tak małą liczbą uczestników nie ma wartości statystycznej. Próbki są tak małe, że nie warto nawet próbować podsumowań. Jednak celem takiego badania nie jest dowiedzenie czegokolwiek. Celem jest zidentyfikowanie najpoważniejszych problemów i ulepszenie produktu przez ich usunięcie. To się po prostu sprawdza, bo większość problemów jest tak oczywista, że nie ma potrzeby przeprowadzania żadnego dowodu”.

Postaraj się powiedzieć to z dużą dozą przekonania i przyjaznym uśmiechem na twarzy.

Ile to będzie kosztować?

Oto przeciętny budżet na rok samodzielnych testów funkcjonalności (nie uwzględnia Twojego czasu):

	Koszt jednostkowy	Koszt roczny
Mikrofon	25 zł	25 zł
Głośniki	35 zł	35 zł
Rejestrowanie zawartości ekranu	Camtasia: 1063 zł	1063 zł
Udostępnianie zawartości ekranu	GoToMeeting: 150 zł/miesiąc	1800 zł
Lunch/poczęstunek dla obserwatorów	100 zł/miesiąc	1200 zł
„Zachęty” dla uczestników	50 – 100 zł/osobę x 36 uczestników	1800 – 3600 zł
	SUMA (w przybliżeniu)	5923 – 7723 zł

A oto wersja „bez wodotrysków”, przydatna w sytuacjach, gdy *nie ma* żadnego budżetu:

	Koszt jednostkowy	Koszt roczny
Mikrofon	25 zł	25 zł
Głośniki	35 zł	35 zł
Rejestrowanie zawartości ekranu	CamStudio (Open Source)	0 zł
Udostępnianie zawartości ekranu	NetMeeting (bezpłatny)	0 zł
Lunch/poczęstunek dla obserwatorów	100 zł/miesiąc	1200 zł
„Zachęty” dla uczestników	Kubki, koszulki lub bony podarunkowe po 25 zł x 36 uczestników	0 – 900 zł
	SUMA (w przybliżeniu)	1260 – 2160 zł