

Wprowadzenie do testowania oprogramowania

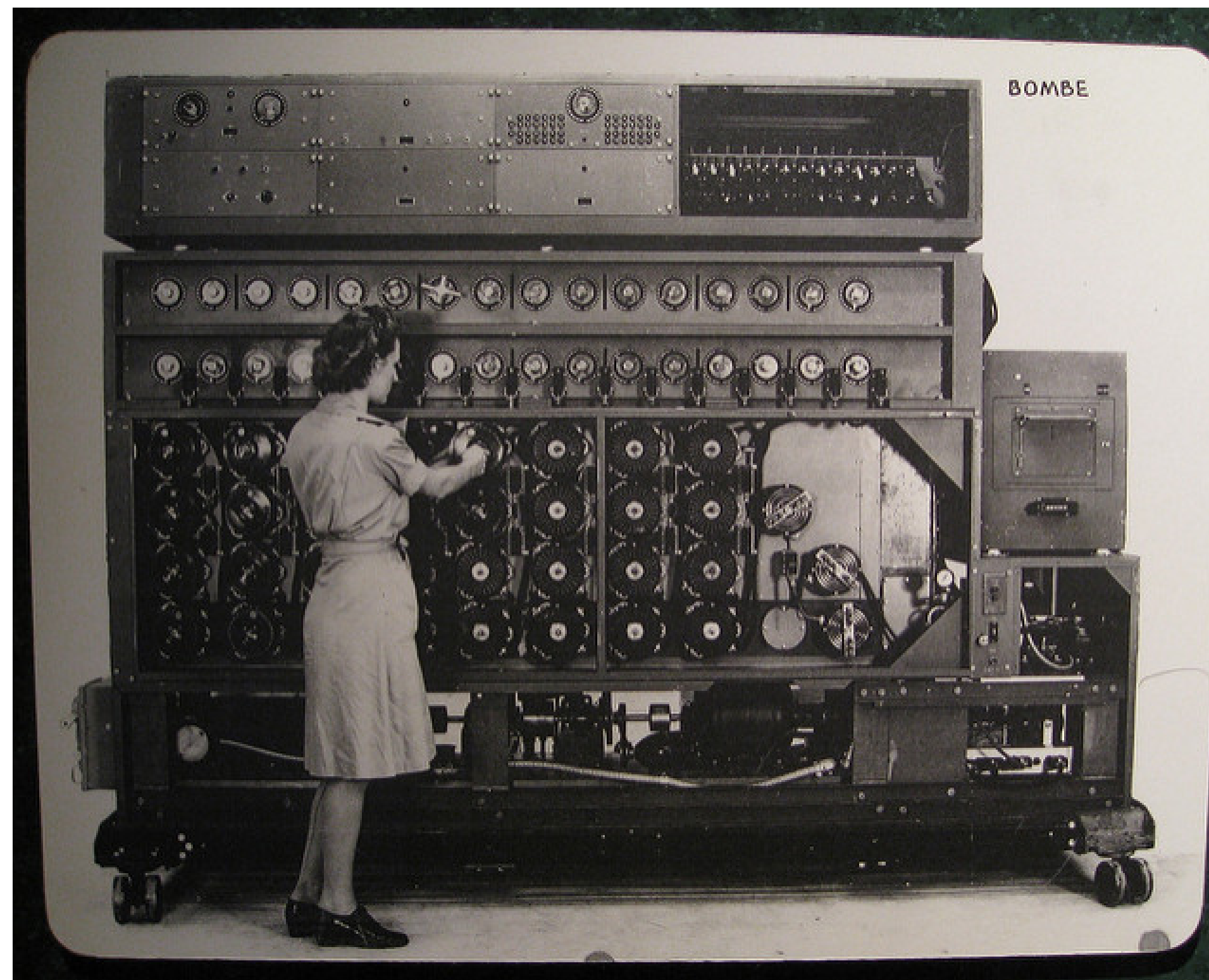


Przypadki Testowe

WOJCIECH BARCZYŃSKI

- Tech Lead
- Software Developer -> System Engineer
- wojciech.barczynski@wsb.wroclaw.pl

Ćwiczenia



Ćwiczenie

- Problem
- Mierzymy się z zadaniem samemu [timebox]
- Pytamy

Można przyjść z swoim laptopem*

Założenia

- Programowania w Pythonie
- Używanie wirtualnego środowiska:

```
python3 -m venv venv
```



Własne narzędzie pracy

Opcjonalnie:

- Ubuntu / *BSD (może być w Virtualbox)
- PyCharm lub vscode
- Python3

Materiały dodatkowe

James Bach

Dość wyraziste opinie, inspirujące:

- Open lecture on Testing
- Test Cases are Not Testing
- On testing in an agile software development team
- Introduction to Rapid Software Testing Course
- On Future of testing

Niekoniecznie ISTQB

James Bach

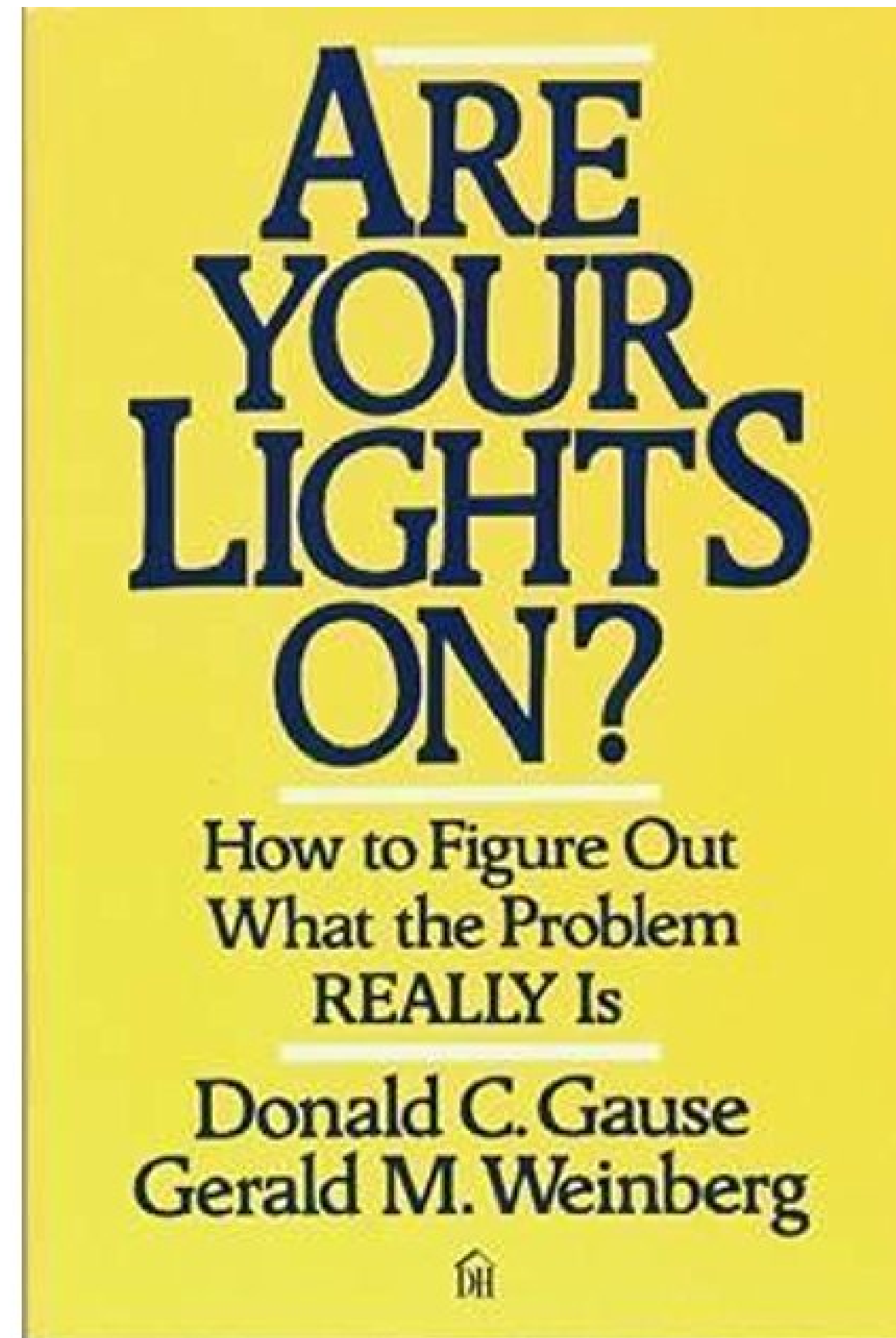
- [Blog](#)
- [Rapid Software Testing](#)

Warto poszukać kogoś do wzorowania się.

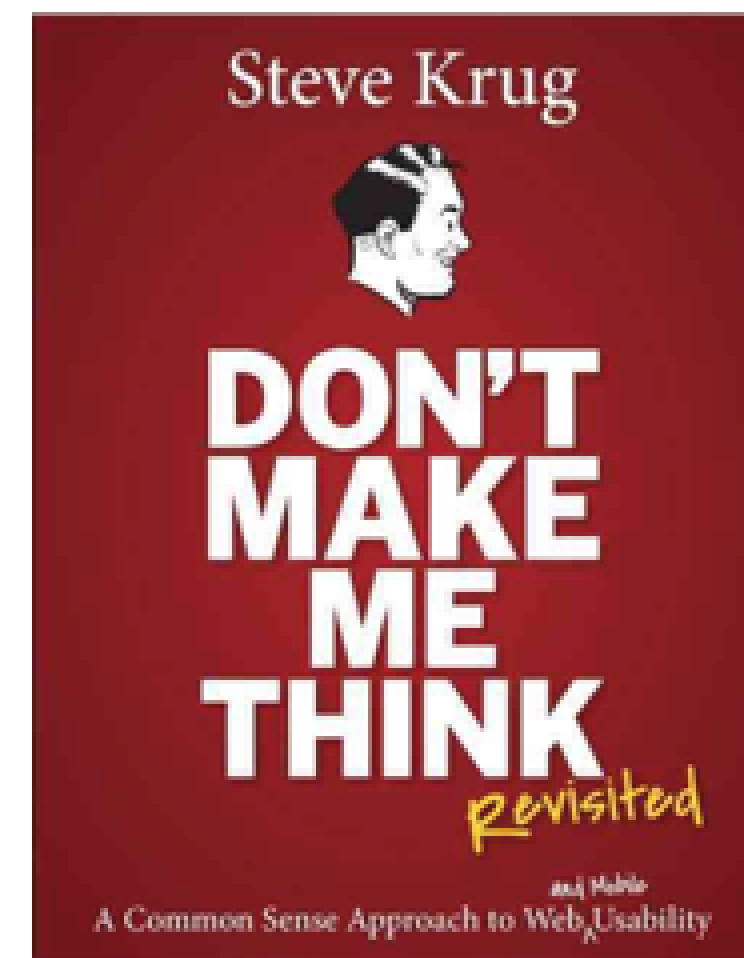
Na kim się wzorować

- Robert C. Martin (Uncle Bob): [arch](#), [scrum](#), [craftsmanship](#)
- Martin Fowler: [arch](#)
- Martin Thomson: [perf](#)
- Brendan Greg: [perf](#)
- Golang: [Peter Bourgon](#)
- C++ (Scott Meyers), Java, ..., microservices, startups,..., coding sessions

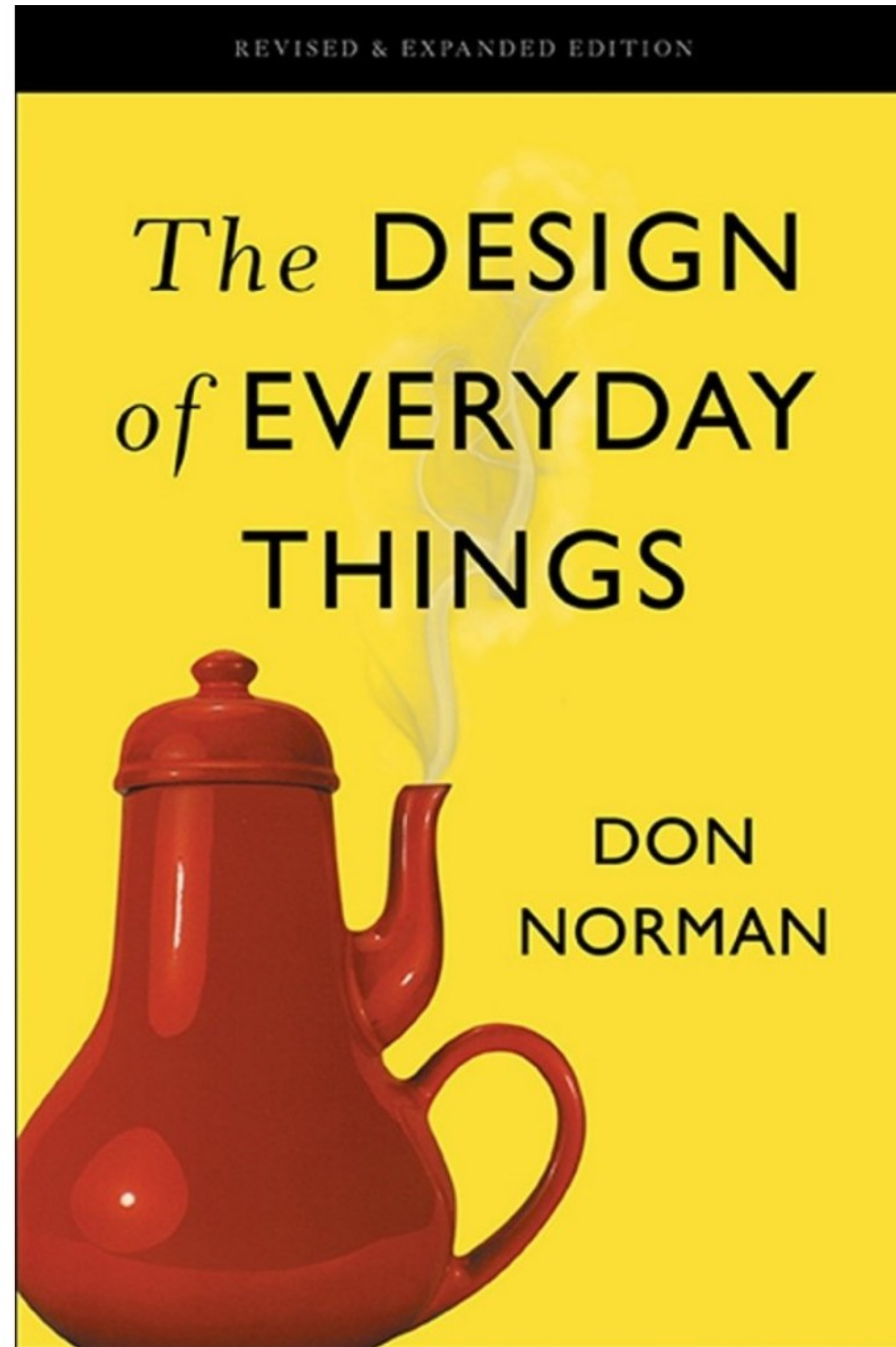
Are Your Lights On?:
How to Figure Out What the Problem Really Is?



Don't Make Me Think: A Common Sense Approach to Web Usability



The Design of Everyday Things



Python



<https://automatetheboringstuff.com/>

Lekka Lektura o najlepszych praktykach
dla programistów

**Andrew Hunt, David Thomas, "Pragmatic
Programmer". Addison-Wesley Professional**

AGENDA

- Kto to jest tester?
- Co to jest testowanie?
- Przypadki testowe

Tester oprogramowania



Moja perspektywa: Developer, System Engineer, Startup



Kto to jest tester
oprogramowania?

- Reflektor, każdego projektu
- Obserwator na dziobie okrętu

Oprogramowanie jest wszędzie

- Rozruszniki serca
- Systemy hamulcowe
- ...
- Zakupy online

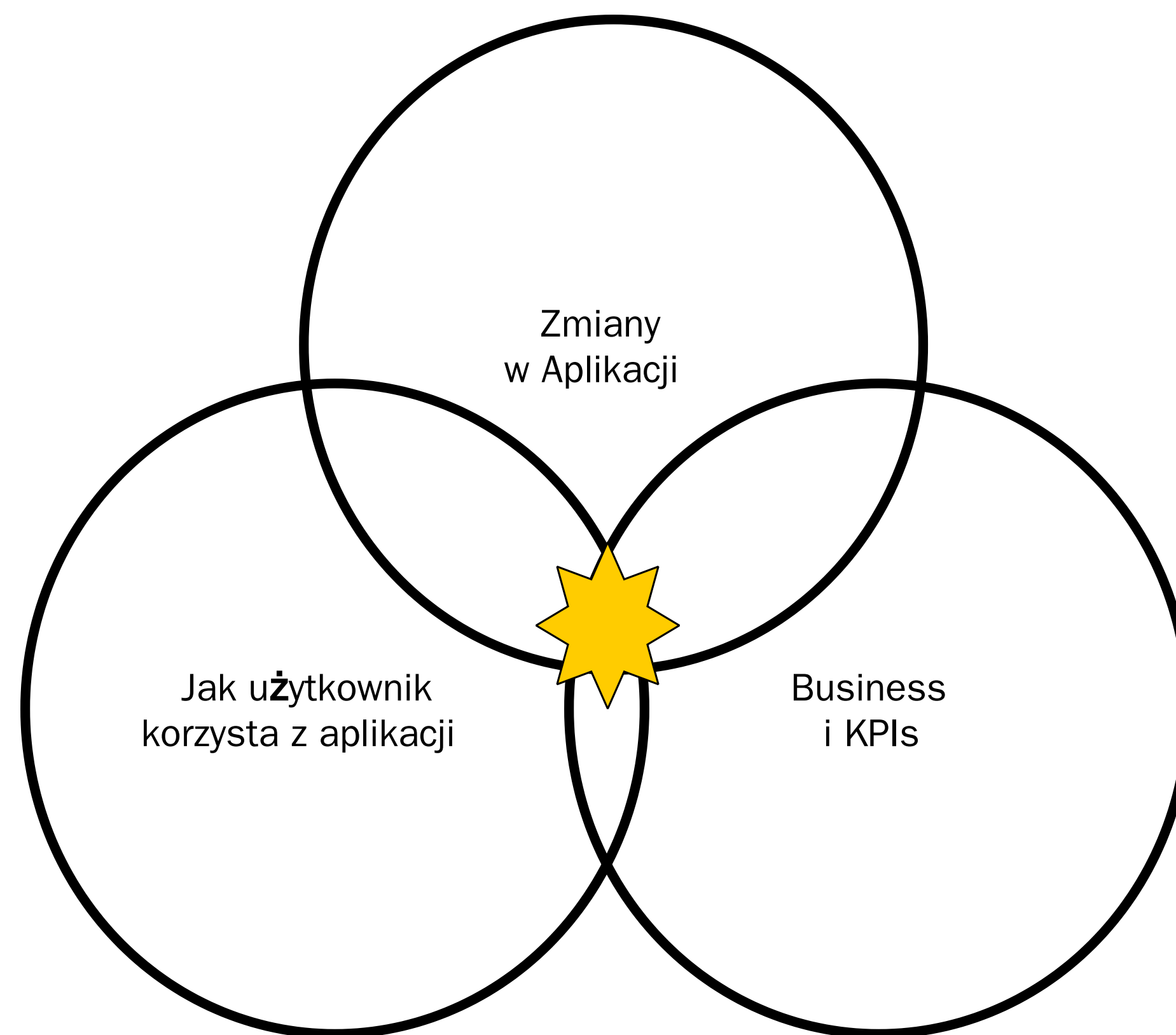
Oprogramowanie jest wszędzie

- Wpływ na każdy aspekt naszego życia
- Czas vs Jakość vs Funkcjonalność

Dlaczego

- Budowanie oprogramowania jest trudne
- Produkt jest bardzo skomplikowany
- Deweloperzy nie są w stanie być jednocześnie użytkownikami
- Deweloperzy skupiają się na jednym aspekcie

Kontekst



Tester oprogramowania

- Zna i interesuje się produktem (często również konkurencji)
- Chętnie eksperymentuje z aplikacją, eksplorującw poszukiwaniu błędów
- Myśli krytycznie i identyfikuje fałszywe założenia
- Obserwuje dynamikę projektu

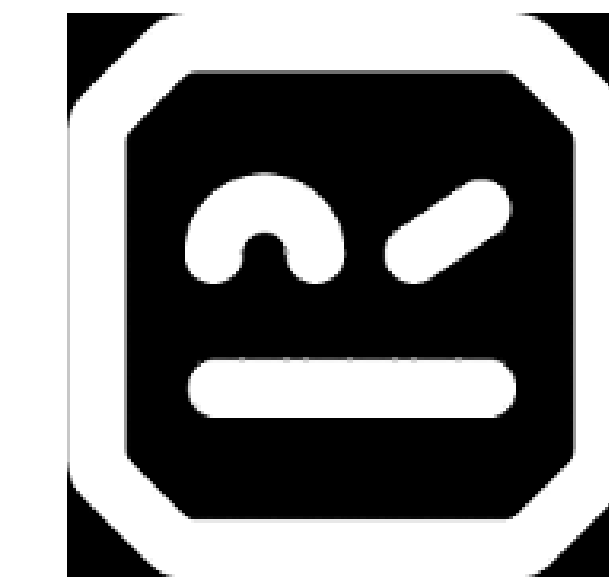
Tester Inżynier

- Potrafi zautomatyzować testy
- Rozumie narzędzia oraz metryki z produkcji
- Nie boi się zanurzyć w kodzie ramie
w ramie z deweloperem

Tester Inżynier

- Walczy z complexity i rigidity
- Zapobiega negatywnym wpływom testów na cykl rozwoju oprogramowanie

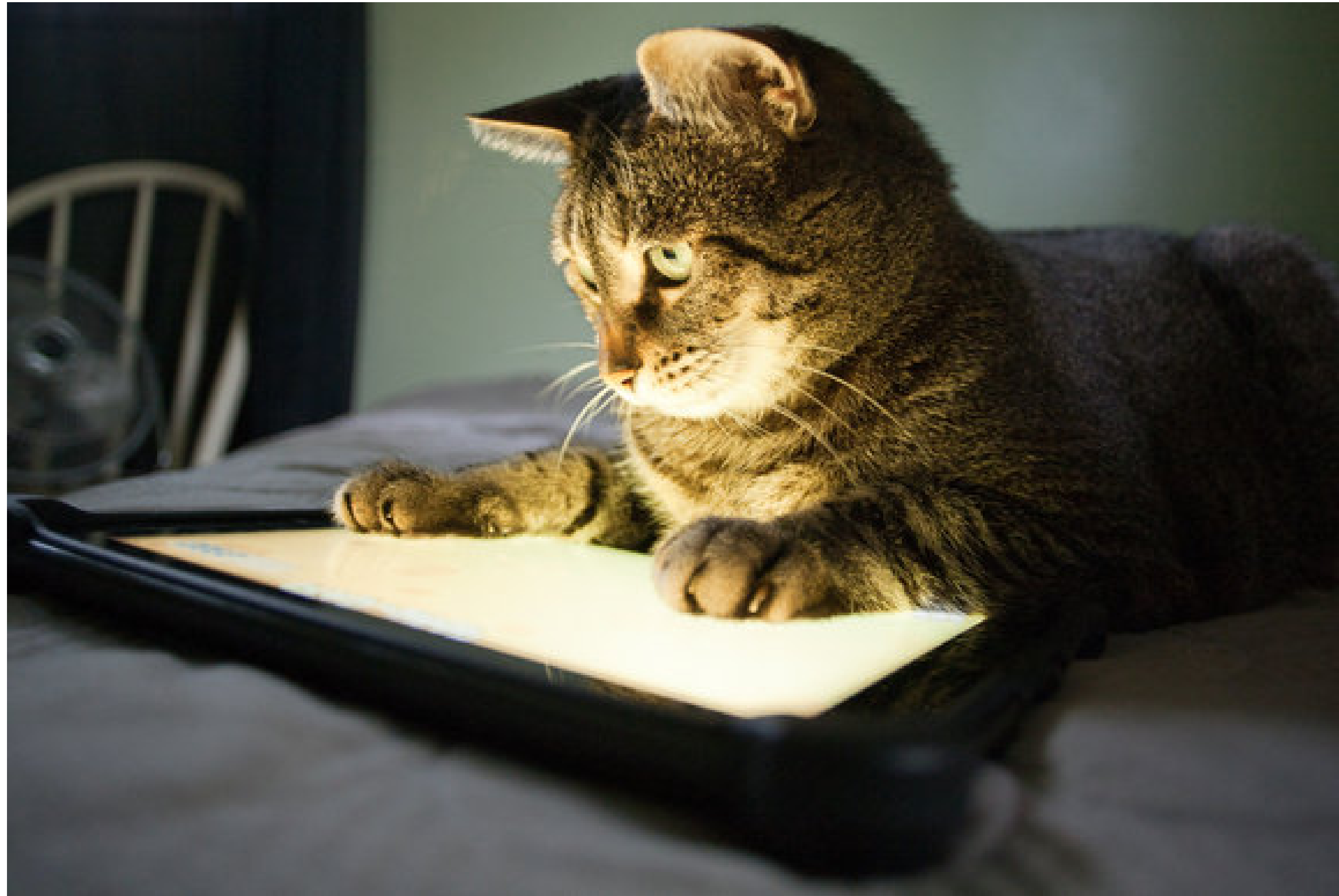
Technologie



RobotFramework



Testy



Testy

- Testy - Performance [1]/Eksploracyjne (Bach)
- Testy - Weryfikacja Automatyczna

[1] tutaj w znaczeniu jak w teatrze

Testy - Performance

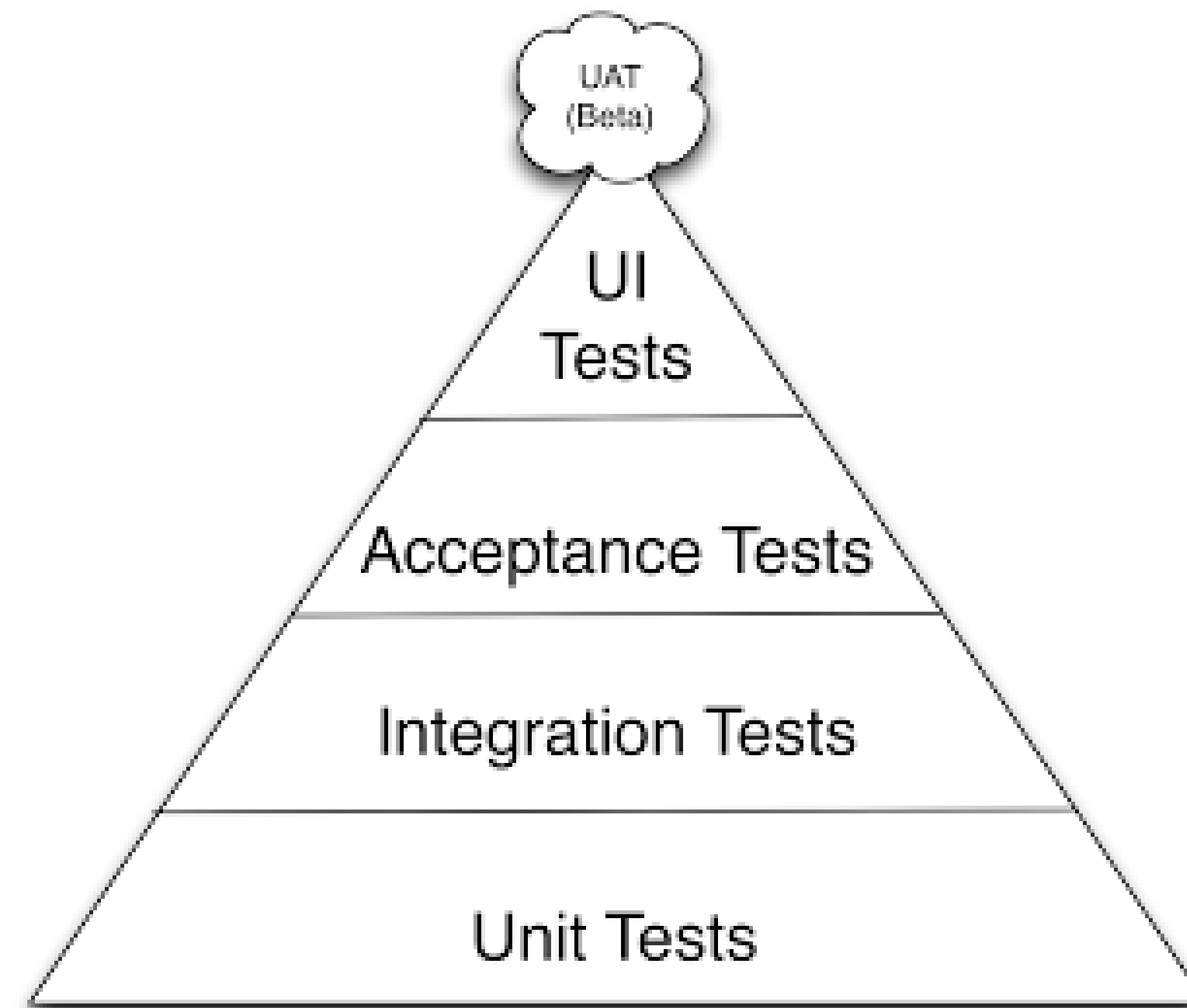
- Manualny
- Może być wspierany przez narzędzia
(curl, postman, selenium, tcpdump, consola dev)

Patrz : [Jak można zarobić 15k na Bug Bounty](#)

Testy - Automatyczne

- Pasy bezpieczeństwa
- Weryfikacja czy czegoś nie zepsuliśmy
- Fuzzing

Testy



Przypadki Testowe

Przypadki Testowe

- Wstępne warunki wykonania / Stan Początkowy
- Zbiór danych wejściowych / Kroki
- Oczekiwane rezultaty

Obiektywne spojrzenie

- Upewnienie się, że cały zespół tak samo rozumie scenariusz
- Zweryfikowanie założeń, każdego członka zespołu
- Znalezienie różnic w zrozumieniu w zespole
- Pomoc w jednakowym rozumieniu wymagań

Przypadki Testowe

- Znalezienie błędów
- Wsparcie w podejmowaniu decyzji
- Ocena zgodności ze specyfikacją
- Zapewnienie jakości
- Dokumentacja

Automatyzacja testów

1. Zbudowanie zbioru przypadków testowych
2. Wybór scenariusza, który daje największe korzyści
3. Budowa autoamtycznego testu

Kluczowe elementy

1. Unikalny numer
2. Jednoznacznie brzmiący tytuł
3. Warunki wstępne
4. Kroki
5. Spodziewane rezultaty

Przypadki Testowe

- Dane testowe
- Środowisko, Wersja
- Powiązanie z wymaganiem (pokrycie wymagań)
- Kategoria, Tagi, Autor

Wykorzystanie

Techniki oparte:

- na wymaganiach - black box
- na strukturze - white box
- na doświadczeniu

Przypadki Testowe

- Pamiętaj, że inne osoby będą korzystały z Twoich przypadków testowych
- Pisz szczegółowo
- Jedna konwencja
- Przypadki pozytywne i negatywne
- Zaplanuj przypadki testowe

Pamiętaj!

- Jak najbliżej kodu i boardu
- Łatwo dostępne

Narzędzia

- Google Spreadsheet
- Jira (XRay, Confluence, Zephyr)
- TestLink
- Pliki tekstowe - Markdown / RestructuredText
- Excel
- Twoje własne notatki, np., [zim](#)

Stopień sformalizowania

- Kultury inżynierii oprogramowania w firmie
- Ograniczeń czasowych
- Wymagania prawne czy regulacje
- ...

Przykład

ID	1234
Tytuł	Logowanie użytkownika do aplikacji webowej
Tagi	Logowanie, użytkownik
Warunki początkowe	Użytkownik - klient banku, posiada rachunek bankowy i dostęp do eBanku
Środowisko	Wszystkie
Wersja PT	0.1
Autor	Emil Bakalarz
Kroki	Rezultaty
Otwórz stronę www.ebank.pl	Widoczna strona, przycisk zaloguj widoczny w prawym górnym rogu
Kliknij w przycisk "Zaloguj"	Popup "Logowanie" widoczny, pole login i hasło widoczne. Przycisk zaloguj widoczny.
Wpisz login	Login widoczny w polu Login
Wpisz hasło	Hasło niewidoczne, wpisane w polu hasło
Kliknij przycisk Zaloguj	Użytkownik jest zalogowany, strona główna eBanku jest widoczna.
Warunki końcowe	Zalogowany użytkownik, przycisk Wyloguj widoczny w prawym górnym rogu ekranu.

Zauważ

Kroki niekoniecznie muszą być manualne, mogą być częściowo wykonane skryptem.

Twoja kolej

1. Wybierz stronę internetową lub aplikację na swoim telefonie
2. Przeglądamy jej funkcjonalność:
 - Co by mogłoby nie zadziałać?
 - Co jest krytyczną funkcjonalnością?
3. Napisz 3-4 przypadki testowe
4. Wyślij na wojciech.barczynski@wsb.wroclaw.pl

Kilka pomysłów

1. <https://www.phptravels.net>
2. mediamarkt
3. <https://www.ryanair.com/pl/pl>

DZIĘKUJĘ ZA UWAGĘ

Wojciech Barczyński

wojciech.barczynski@wsb.wroclaw.pl

BACKUP SLIDES