

# Teoria testów

# Robert Kaszubowski

Dawniej Java Developer  
Tester automatyczny & manualny  
Test lead  
Test team leader i test manager  
People manager – mentor i coach  
Project coordinator  
Global head of testing automation CC in Epam Poland  
Trener ☺

<http://robertkaszubowski.com/>  
robertkaszubowski147@gmail.com

Hobby: sporty walki, trening siłowy, gry wideo, dobrej jakości sprzęt audio



# Agenda

Podział testów  
Przypadek testowy  
Techniki projektowania testów  
Zgłoszenie błędu  
Piramida testowa  
Waterfall vs Scrum  
Techniki estymacji  
Raportowanie wyników testów  
Przydatne narzędzia wspomagające testowanie  
Metryki  
Ryzyka i zarządzanie nimi  
Częste pytania rekrutacyjne i jak na nie odpowiadać





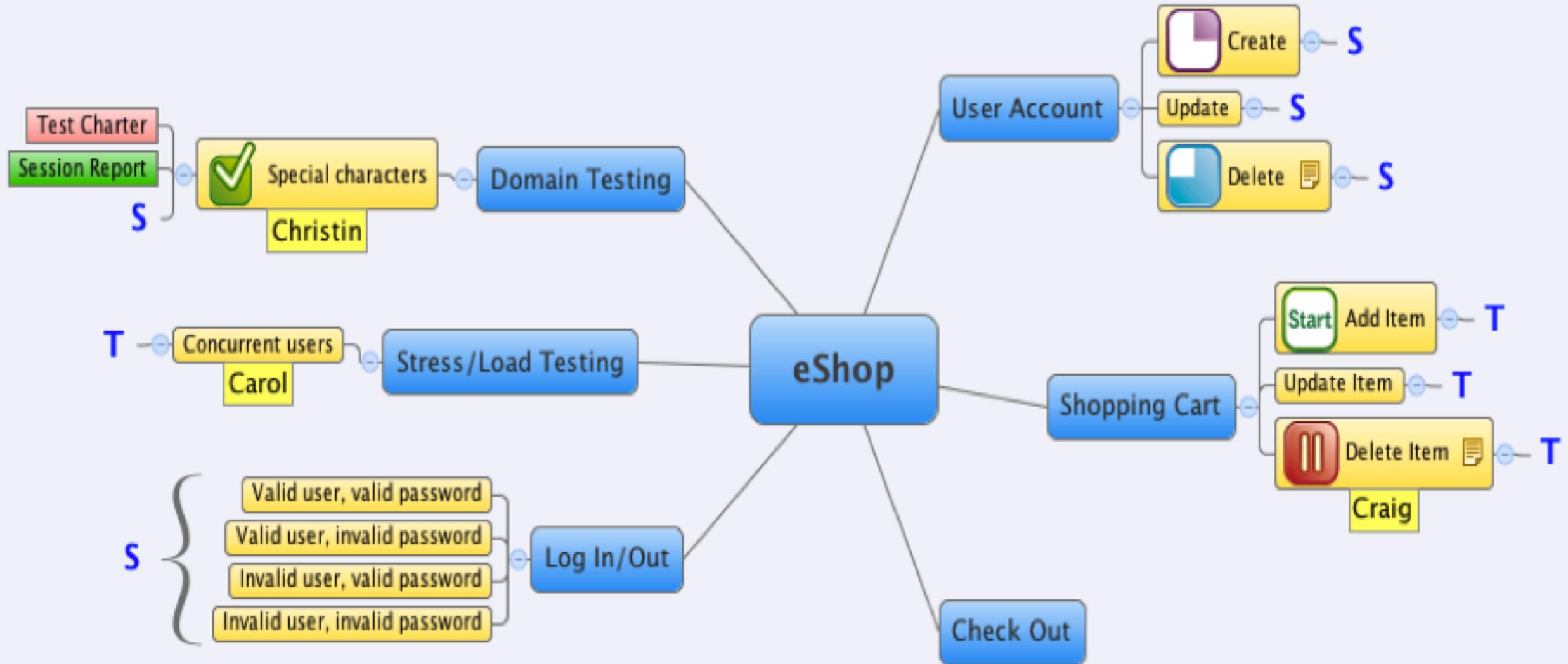
Co to jest testowanie?



Dlaczego teoria jest ważna?



# Dokumentowanie testów eksploracyjnych

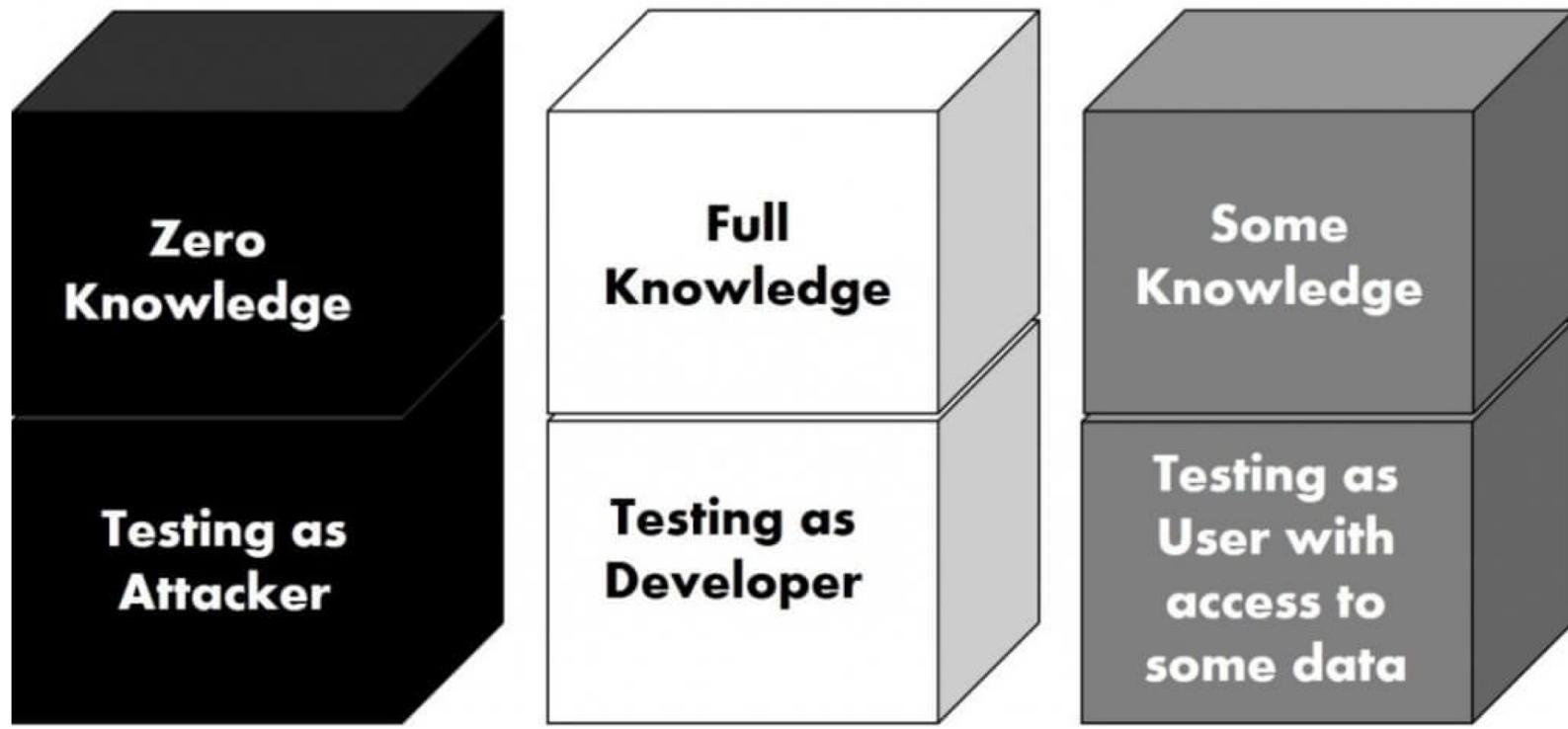


## Podział testów #1

Ze względu na widzialność kodu:

- białośkrzynkowe
- szarośkrzynkowe
- czarnoskrzynkowe

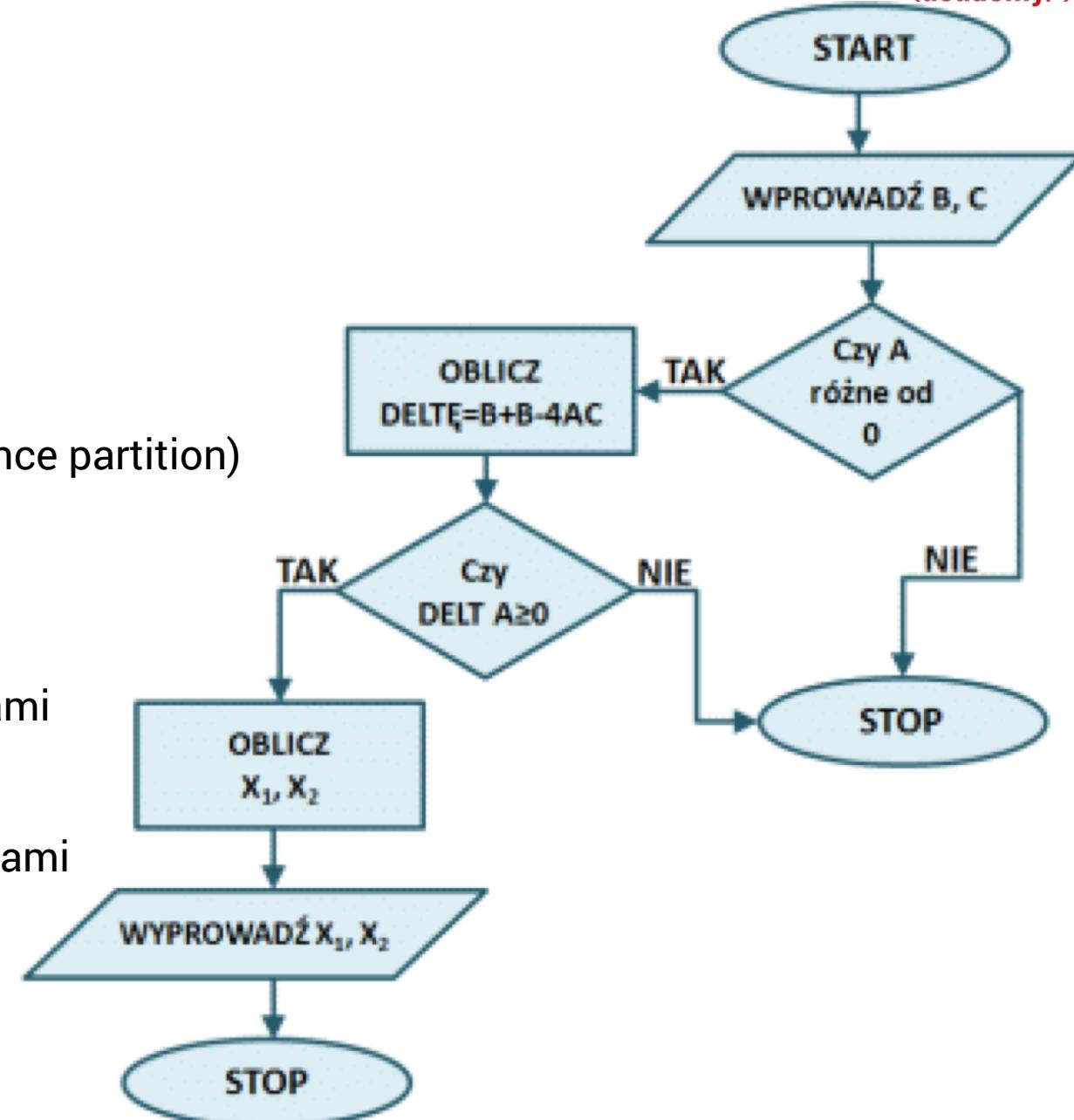
### Differences between Types of Penetration Testing



## Podział testów #2

Ze względu na poziom testowania:

- **testy modułowe (unit/component testing)**
  - analiza ścieżek (path analysis)
  - użycie klas równoważności (equivalence partition)
  - testowanie wartości brzegowych
  - tablice decyzyjne
- **testy integracyjne (integration testing)**
  - a) testy integracyjne pomiędzy modułami
    - funkcjonalne
    - wydajnościowe
  - b) testy integracyjne pomiędzy systemami
    - funkcjonalne
    - wydajnościowe
    - regresywne



## Decision Table

<b>Conditions</b>	TC1	TC2	TC3	TC4
Request login	0	1	1	1
Valid user name entered	X	0	1	1
Valid password entered	X	X	0	1
<b>Actions</b>				
Offer recovery credentials	0	1	1	0
Activate entrybox user name	0	1	1	0
Activate entrybox password	0	0	1	0
Enter privileged area	0	0	0	1

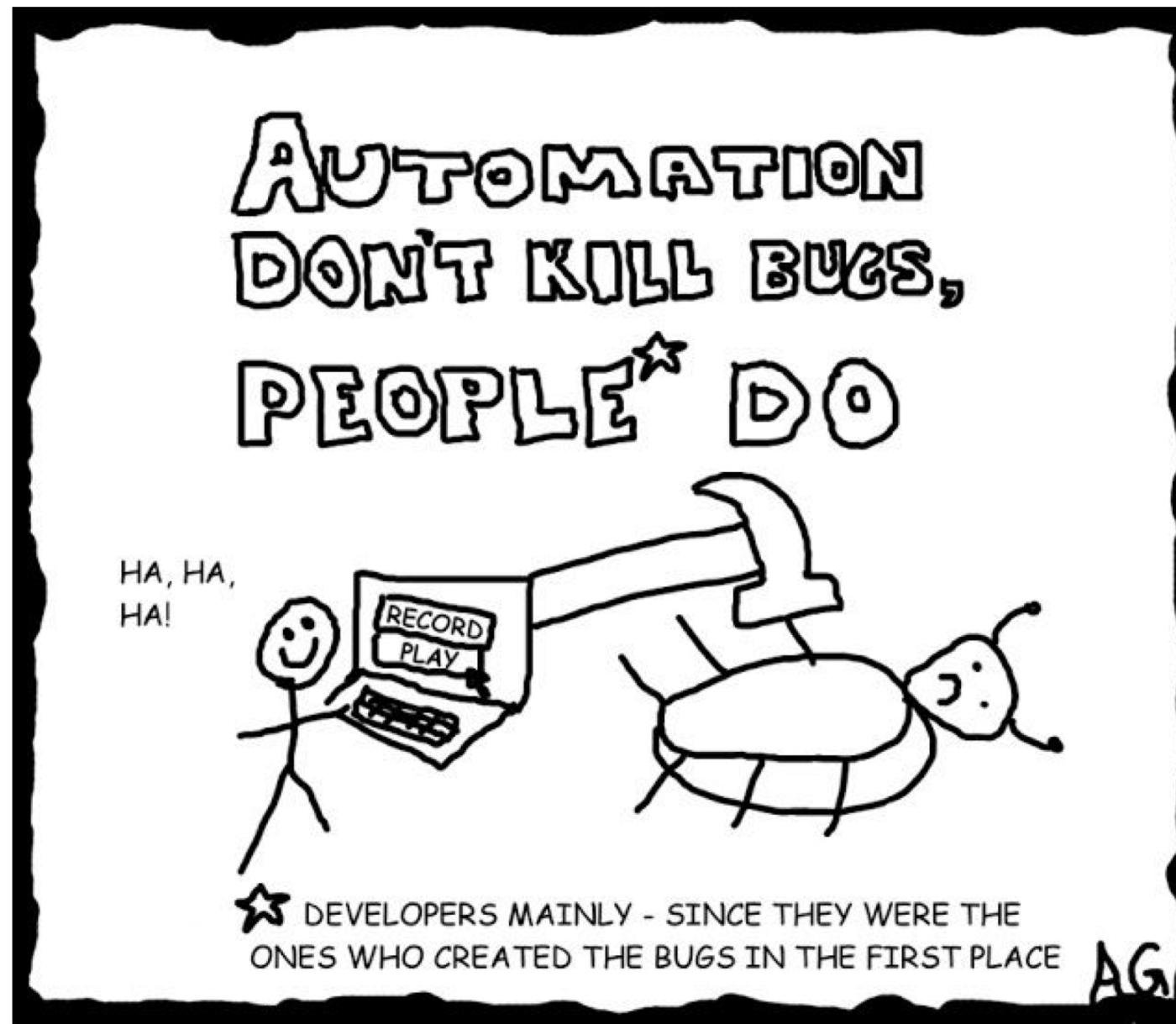
## Podział testów #3

- **testy systemowe (system testing)**
  - instalacyjne
  - funkcjonalne
  - interfejsu (użyteczności)
  - wydajnościowe
  - regresywne
  - bezpieczeństwa
- **testy akceptacyjne (acceptance testing)**
  - funkcjonalne
  - wydajnościowe
  - bezpieczeństwa

## Podział testów #4

Ze względu na sposób wykonywania:

- manualne
- automatyczne



Czy możemy się pozbyć całkowicie testów manualnych?

# Podział testów #5

- **Testy funkcjonalne**

- testy jednostkowe
- testy dymne (smoke testing) / testy kondycji
- testowanie integracyjne
- testy systemowe
- testy regresji
- testy akceptacyjne użytkownika
- testowanie metodami białej skrzynki oraz metodą czarnej skrzynki

## Testy niefunkcjonalne

- testy bezpieczeństwa
- testy niezawodności
- testy użyteczności
- testy łatwości konserwacji

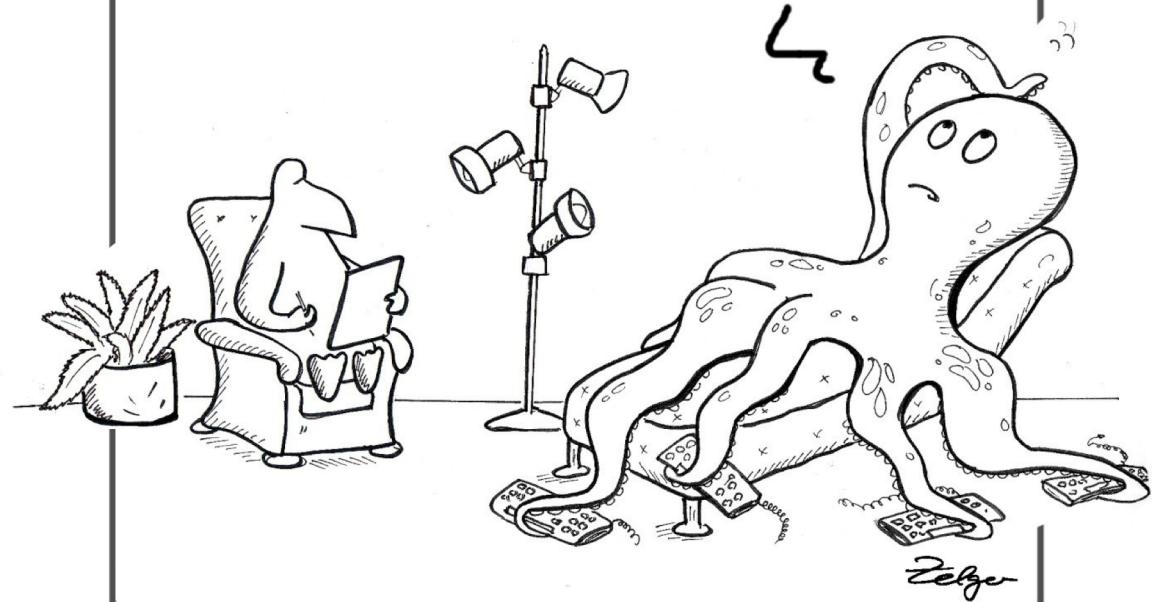
## Podział testów #6

- **Testy wydajnościowe (rodzaj testów niefunkcjonalnych)**

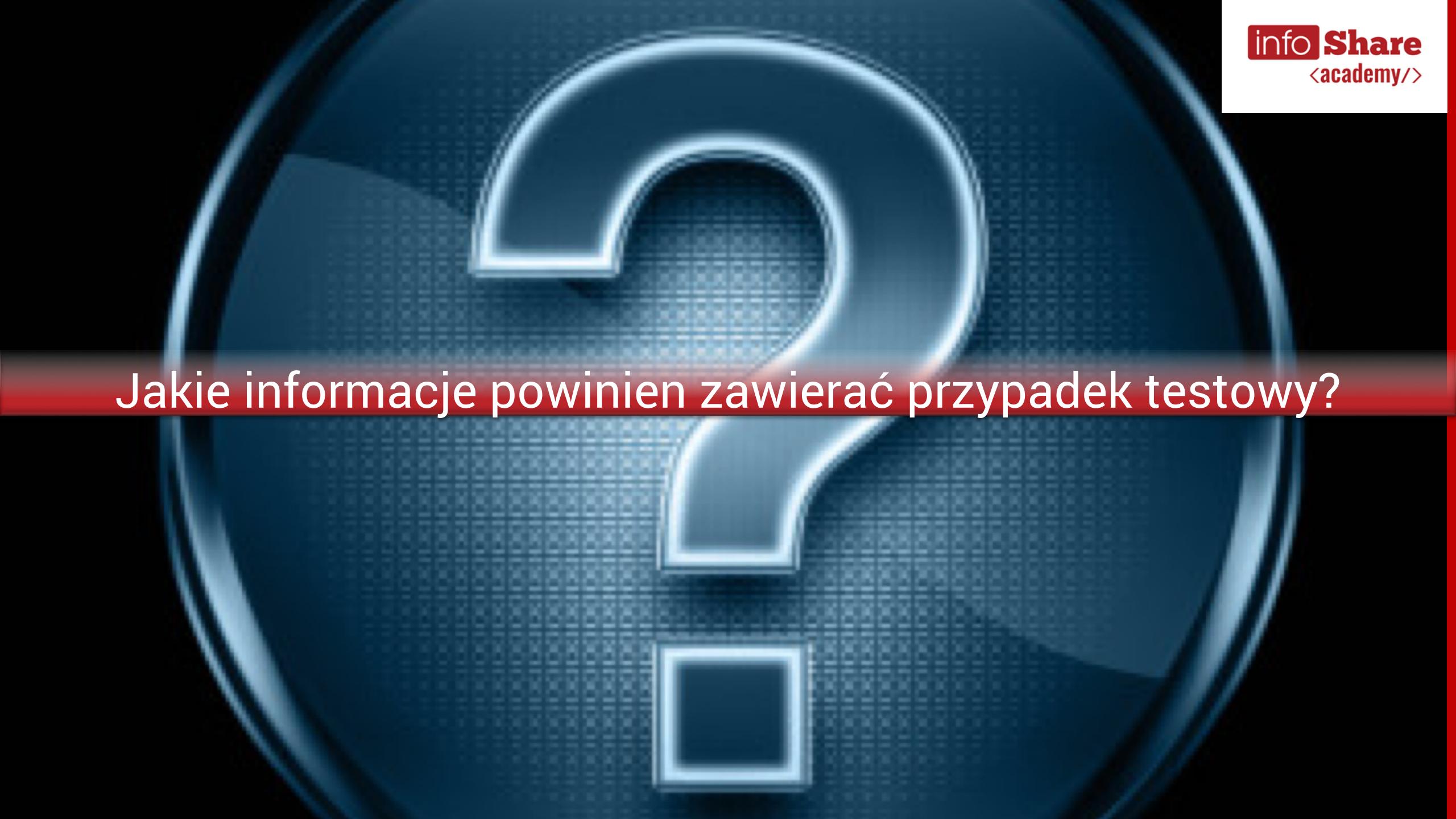
- testy wydajności
- testy przeciążeniowe
- testy obciążeniowe

[www.zeiger.org](http://www.zeiger.org)

"I'M JUST WONDERING WHY THEY  
ALWAYS CHOOSE ME FOR EXECUTING  
THEIR PERFORMANCE TESTS"



The multi-threaded octopus was looking for a change



Jakie informacje powinien zawierać przypadek testowy?

# Scenariusz testowy (Test Case)

<i>Identyfikator testu</i>	<i>Nazwa testu</i>
<b>Dotyczy wymagania</b>	
<b>Cel testu</b>	
<b>Warunek wstępny</b>	
<b>Scenariusz (kroki testowe)</b>	
<b>Akcje użytkownika</b>	<b>Odpowiedź systemu</b>
<b>Komentarze</b>	
<b>Ocena testu</b>	

Można dodać także priorytet, który może być wykorzystany przy podejściu Risk Based Testing oraz dane testowe

# Wskazówki jak pisać dobre przypadki testowe

Używaj opisowych tytułów

Dołącz opis

Dołącz warunki wstępne (preconditions) i założenia

Zachowaj kroki przejrzystymi i o podobnym poziomie abstrakcji

Dołącz opis oczekiwanej zachowania

Przypadki pisz tak by były możliwa reużyta po raz kolejny

# Przykładowy przypadek testowy

TC02 - Login Page - Authenticate Successfully on gmail.com

Last updated on: 29th Nov 2015, Last saved by: Jake Bartlett, [View stats](#)

A registered user should be able to successfully login at gmail.com.

Precondition: the user must already be registered with an email address and password.

Assumption: a supported browser is being used.

## Test steps

1. Navigate to gmail.com
2. In the 'email' field, enter the email of the registered user.
3. Click the 'Next' button.
4. Enter the password of the registered user
5. Click 'Sign In'

## Expected result

A page displaying the gmail user's inbox should load, showing any new message at the top of the page.

# Techniki projektowania testów

Techniki oparte na specyfikacji, czarnoskrzynkowe:

Podział na klasy równoważności

Analiza wartości brzegowych

Testowanie w oparciu o tablicę decyzyjną

Testowanie przejść między stanami

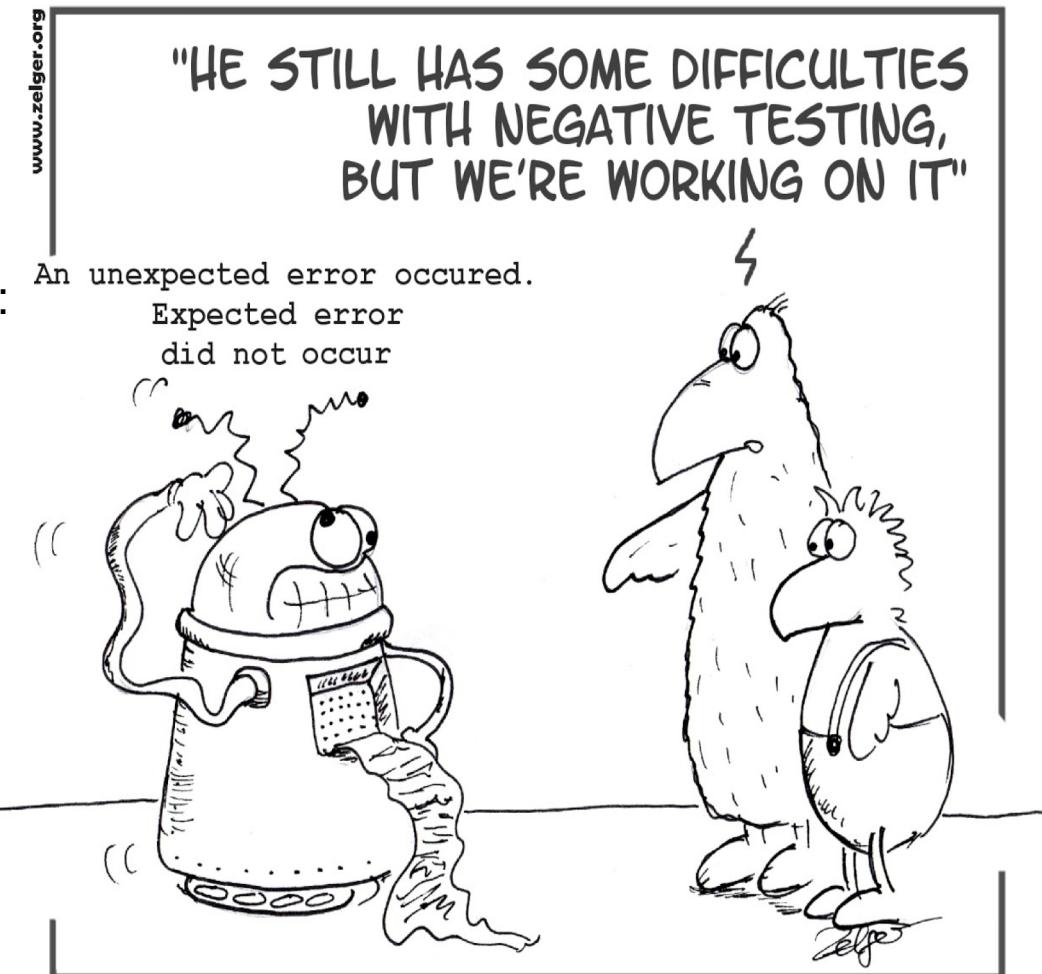
Testowanie w oparciu o przypadki testowe

Techniki oparte na strukturze, białośkrzynkowe:

Testowanie i pokrycie instrukcji

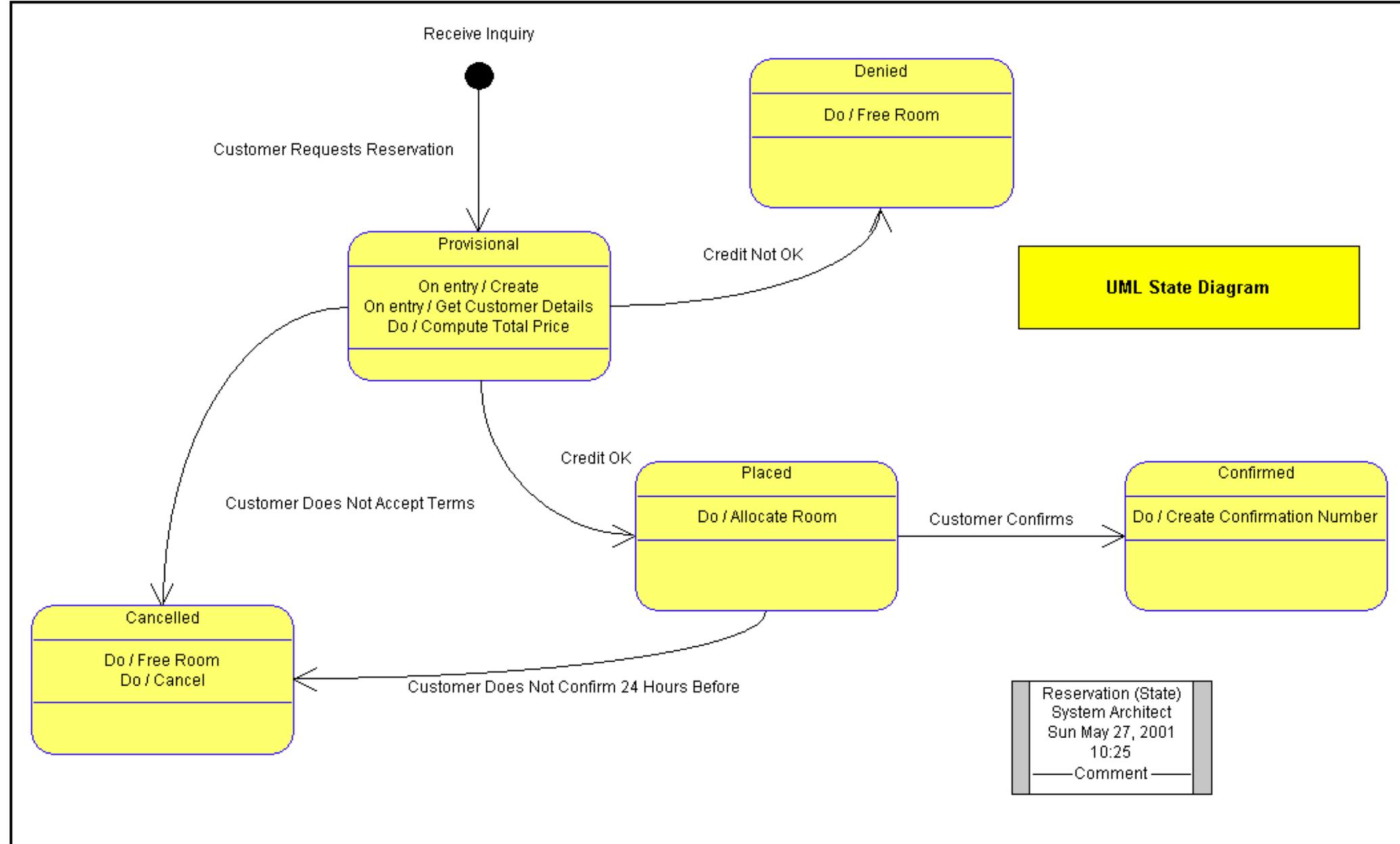
Testowanie i pokrycie decyzji

Inne techniki oparte na strukturze



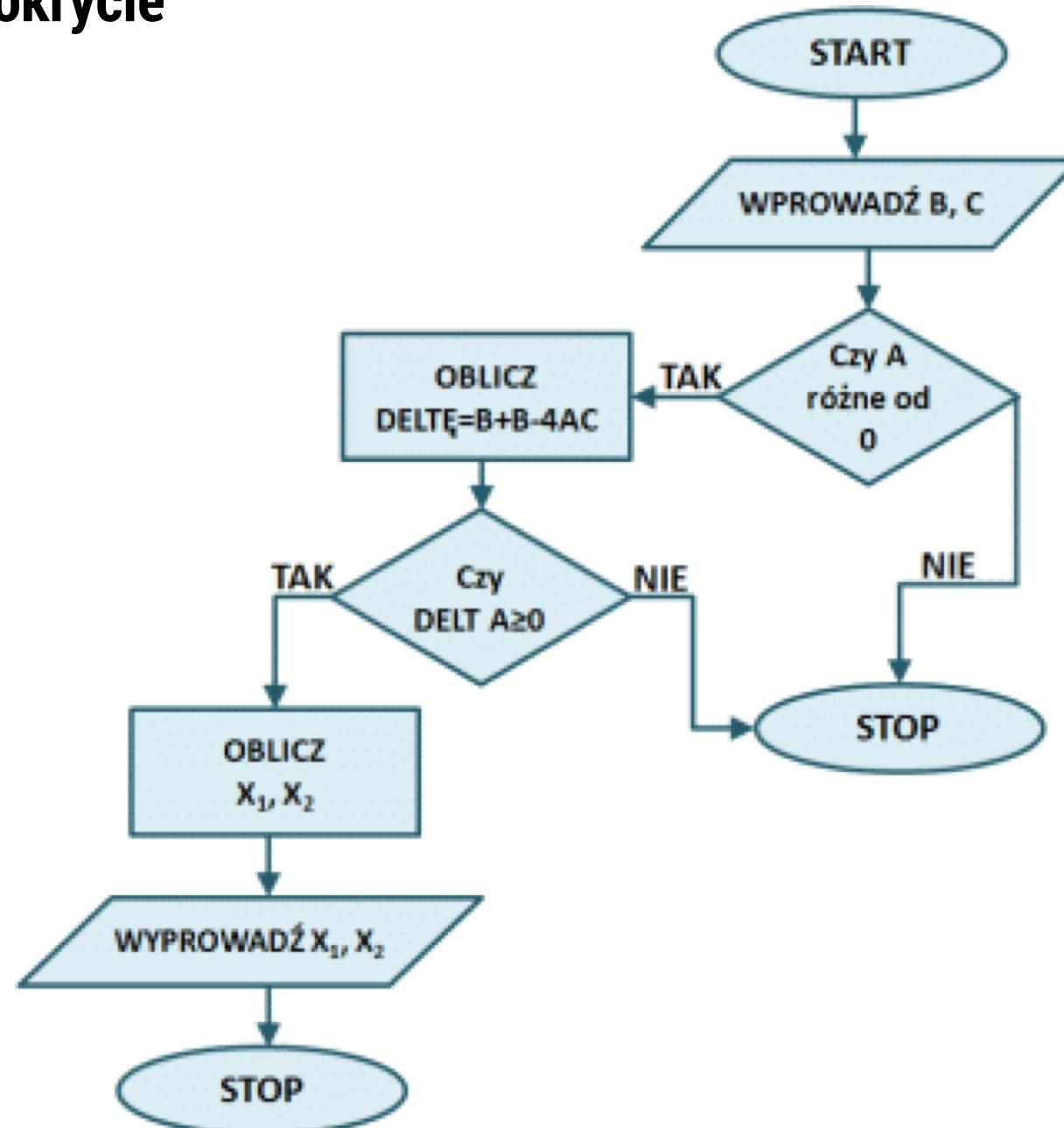
Ciekawy blog - <http://jakzostacnerdem.blogspot.com/2016/10/istqb-part-iii-techniki-projektowania.html>

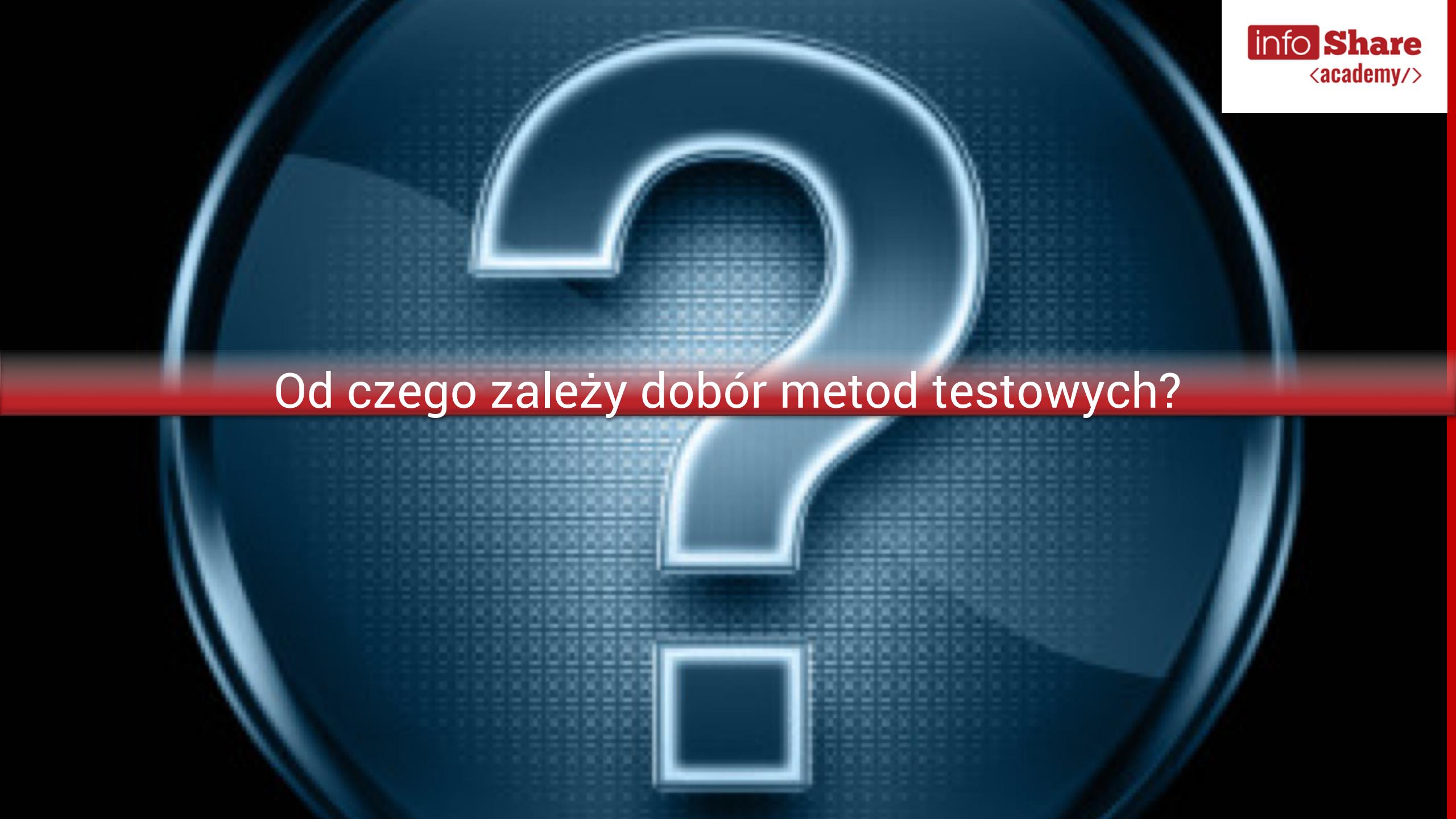
# Przykładowy diagram stanu



# Pokrycie instrukcji vs pokrycie decyzji

**100% pokrycia decyzji gwarantuje  
100% pokrycia instrukcji ale nie  
odwrotnie!**

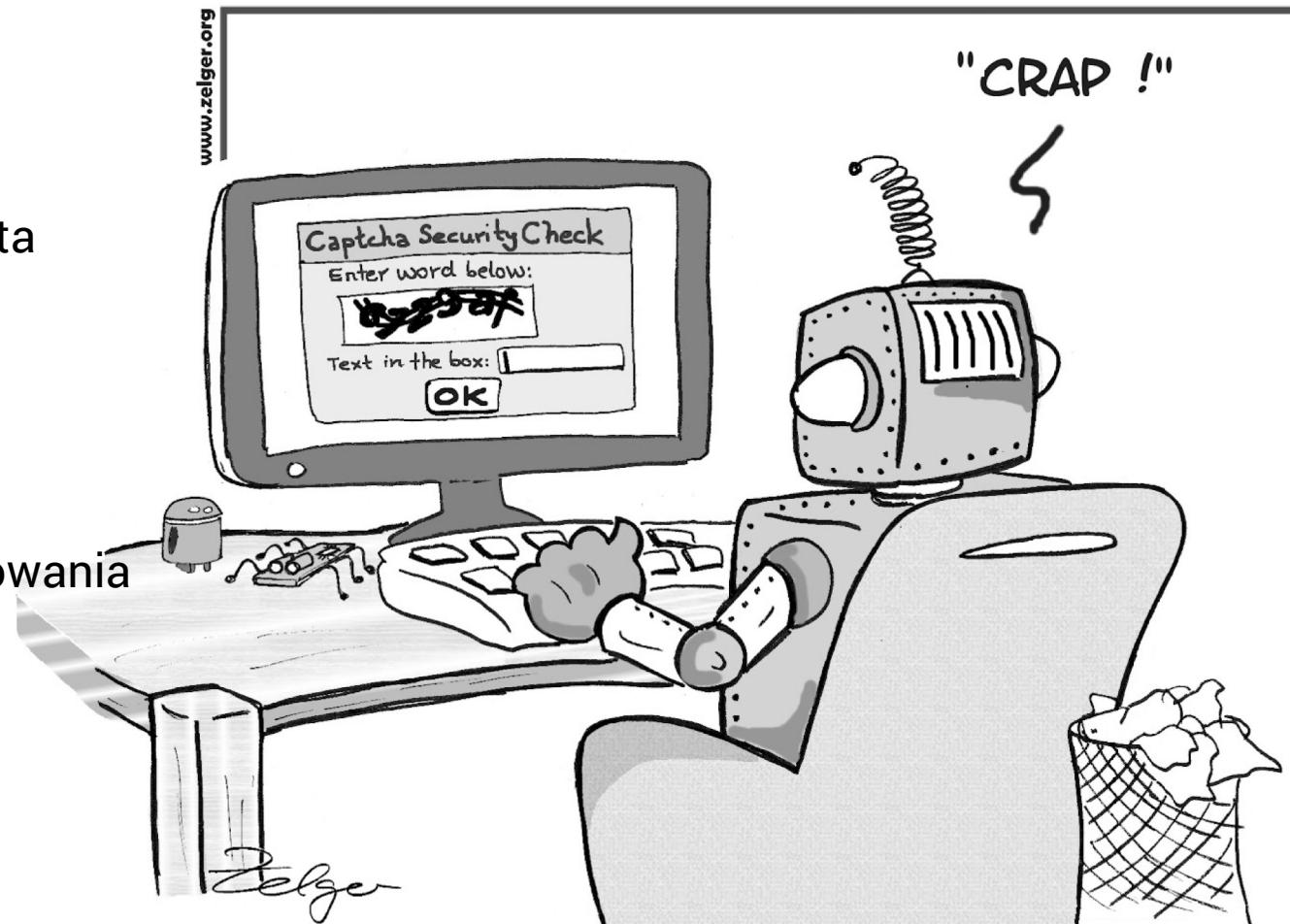




Od czego zależy dobór metod testowych?

## Dobór metod zależy od:

- Typ systemu
- Standardy dla systemu
- Wymagania kontraktu lub klienta
- Poziom ryzyka
- Cele testów
- Dostępna dokumentacja
- Wiedza testerów
- Czas i budżet
- Typ cyklu tworzenia oprogramowania
- Model przypadków użycia
- Poprzednie doświadczenia
- Typy znalezionych defektów.



And this is where Rusty the robot,  
all at once, ended up in a blind alley.

## Zadanie #1

Każdy z Was tworzy mind mapę a następnie przypadki testowe dla aplikacji:

<http://www.phptravels.net/supplier>

Email [supplier@phptravels.com](mailto:supplier@phptravels.com)

Password demosupplier

Przypadki testowe powinny znaleźć się w Jirze w waszym projekcie.

Czas: 2.5h