



# Wstęp do testowania

Presented by: **Szymon Ramczykowski**  
Date: **29.04.2014**

**Kainos**

k a i n  s<sup>®</sup>

---

k a i n  s<sup>®</sup>

INNOVATION BY KAINOS

# Gdzie jesteśmy

k a i n o s<sup>®</sup>



## Branże



## Technologie



- **Ponad 100 osób zatrudnionych**
- **Wiele ciekawych projektów**
- **Rozbudowany dział testów automatycznych**
- **Incjatywy dla studentów:**
  - AppCamp
  - Test Automation Academy
  - Prelekcje / wykłady / warsztaty
  - Projekty grupowe

- **Absolwent PG i PJWSTK**
- **Od 5 lat zawodowo związany z testowaniem w zróżnicowanych projektach**
  - Sektor Bankowy
  - Bazy danych
  - Aplikacje sieciowe
- **Od września 2012 roku Senior Test Engineer w firmie Kainos**
  - Aplikacja EDRM dla szpitali na terenie UK
  - Szeroki zakres obowiązków 😊

- **Wprowadzenie do Testowania**
  - Po co testujemy?
  - Testowanie, weryfikacja i walidacja
  - Rodzaje testów
  - Błędy i ich raportowanie
- **Automatyzacja Testów**
  - Słownik podstawowych pojęć
  - Dlaczego testy automatyczne?
  - Koszty automatyzacji

- **Automatyzacja Testów c.d.**
  - Kiedy i jakie testy automatyzować?
  - Rodzaje testów automatycznych
  - Narzędzia testowe







## Błędy w oprogramowaniu mogą doprowadzić do:

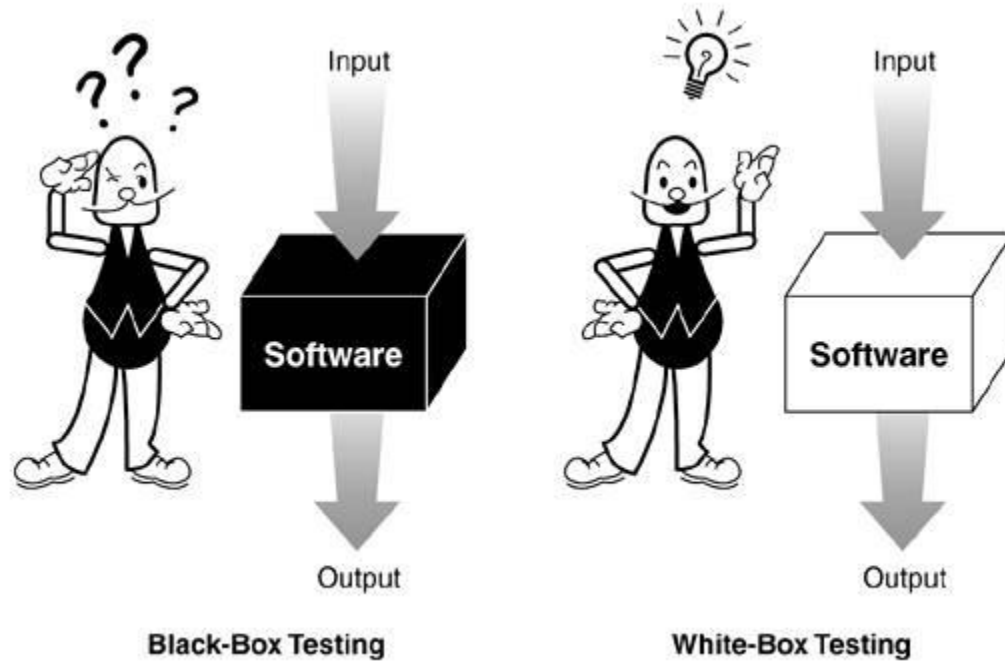
- Strat finansowych
- Straty czasu
- Start wizerunkowych (utrata zaufania)
- Śmierci lub kalectwa!

Software testing is the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item

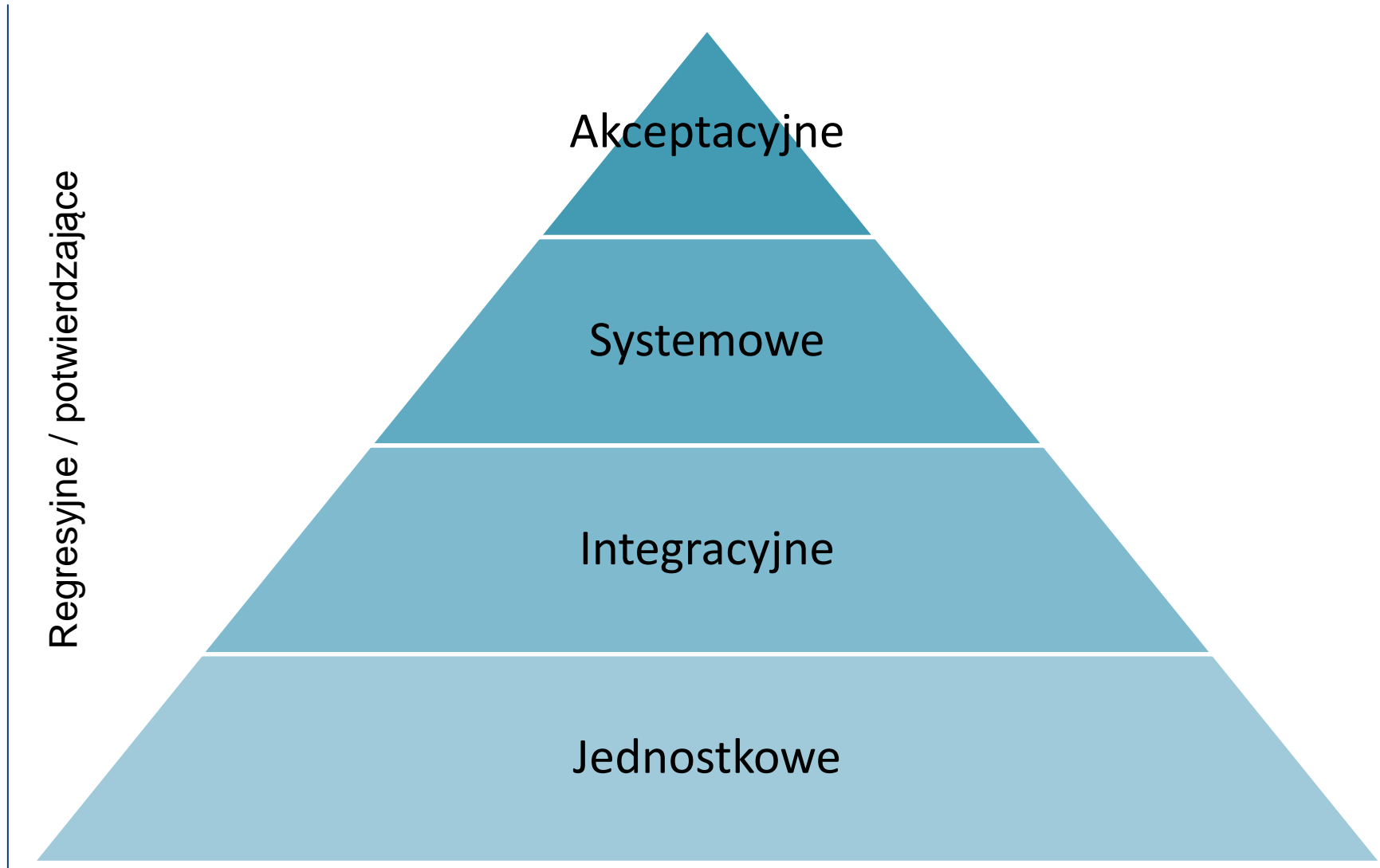
The Institute of Electrical and Electronics Engineers (IEEE)

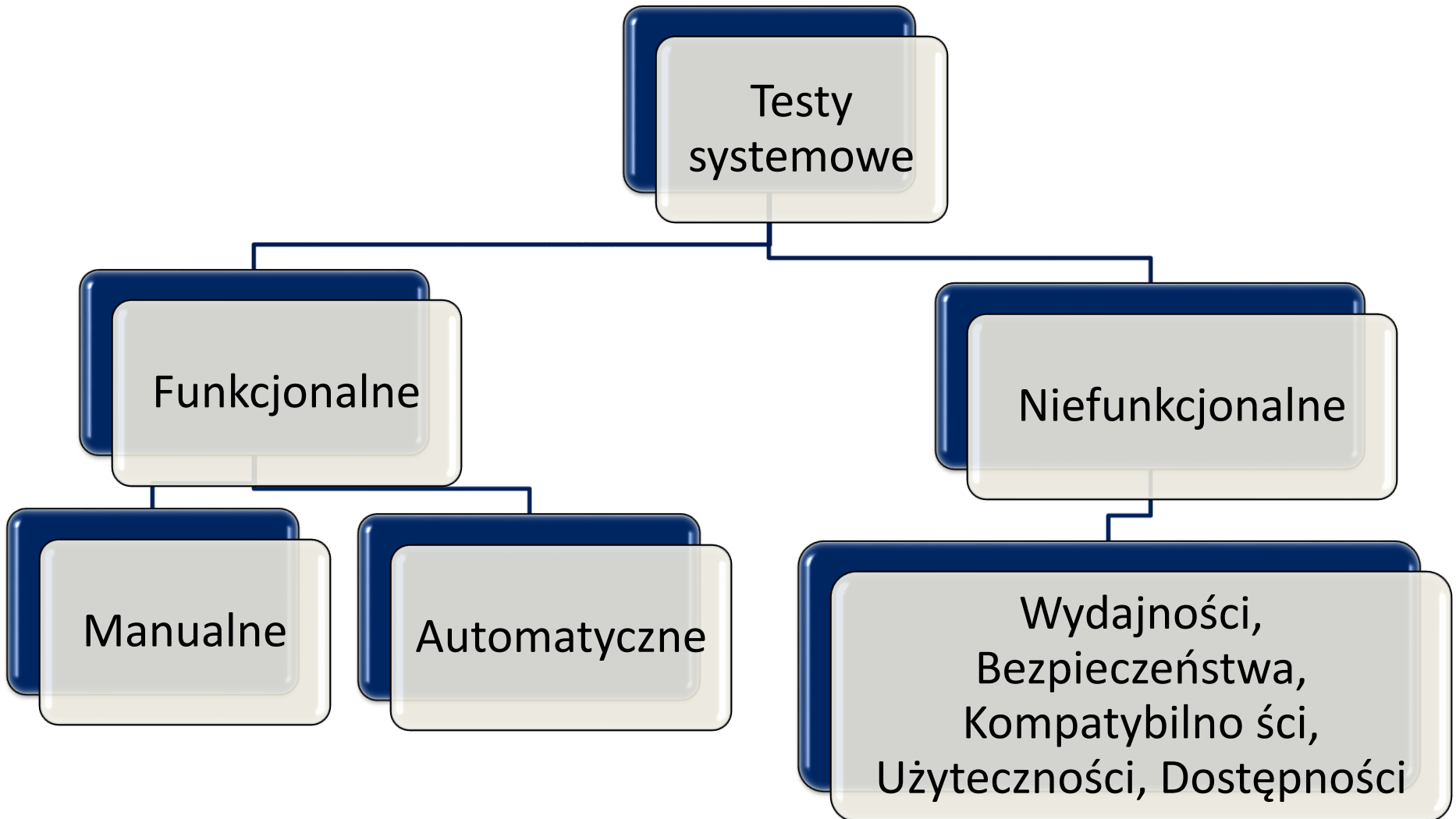
Verification Are we building the product <u>right</u> ?	Validation Are we building the <u>right</u> product?
 <p>"I landed on "Go" but didn't get my \$200!"</p>	 <p>"I know this game has money and players and "Go" – but this is not the game I wanted."</p>

**Verification vs. Validation**

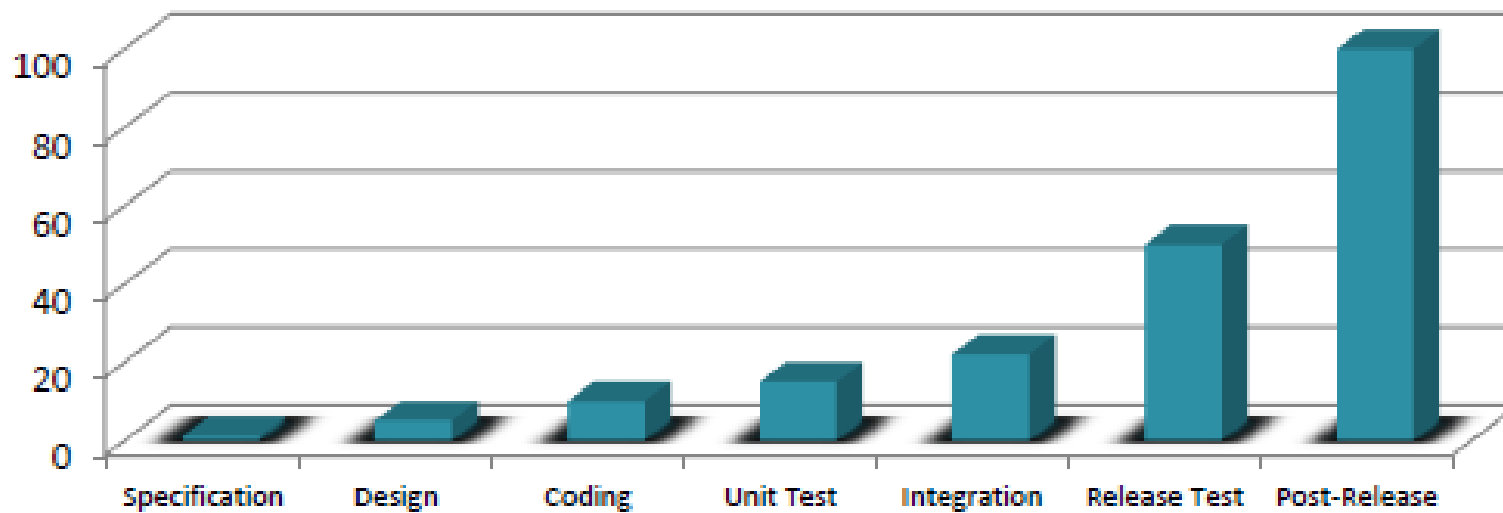


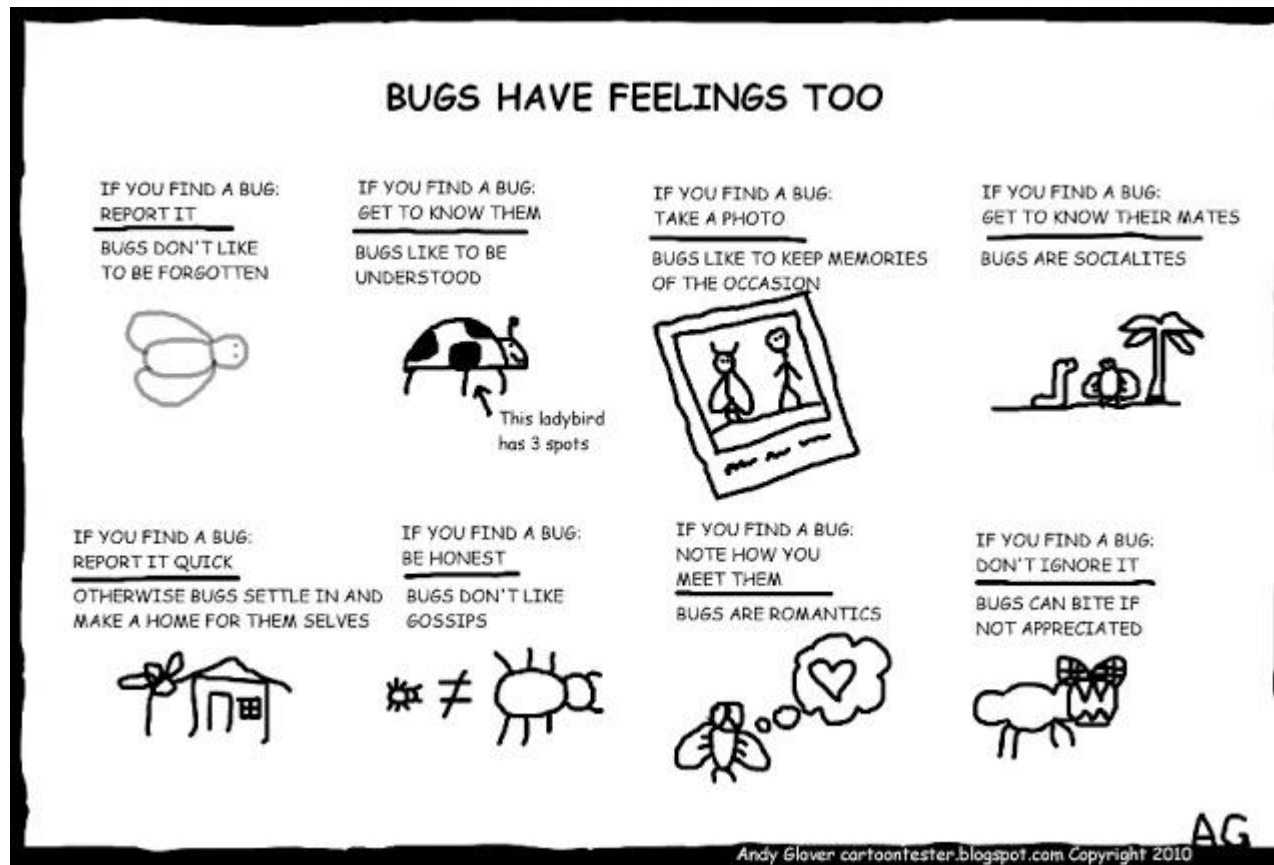
# Rodzaje testów (2)





*(Relative cost of defect, by time of discovery)*







# Raportowanie błędów (2)

Priority	Low
Resolution	Code Updated
Status	Fixed
Platform	
OS	
OS Version	
Fixed in Build	
Target Build	
Description	See Luke
Steps To Reproduce	See Luke
Build Number	
Clinical Risk	Medium
Environment	Development



- **Test case:** przypadek testowy, scenariusz testu, najczęściej w formie wypunktowanej listy akcji i oczekiwanych rezultatów oraz warunków początkowych
- **Test suite:** zbiór przypadków testowych, najczęściej dotyczący jednej funkcjonalności
- **Wynik testu:** rezultat wykonania przypadku testowego (passed/failed/blocked/not run)
- **Defekt:** błąd oprogramowania

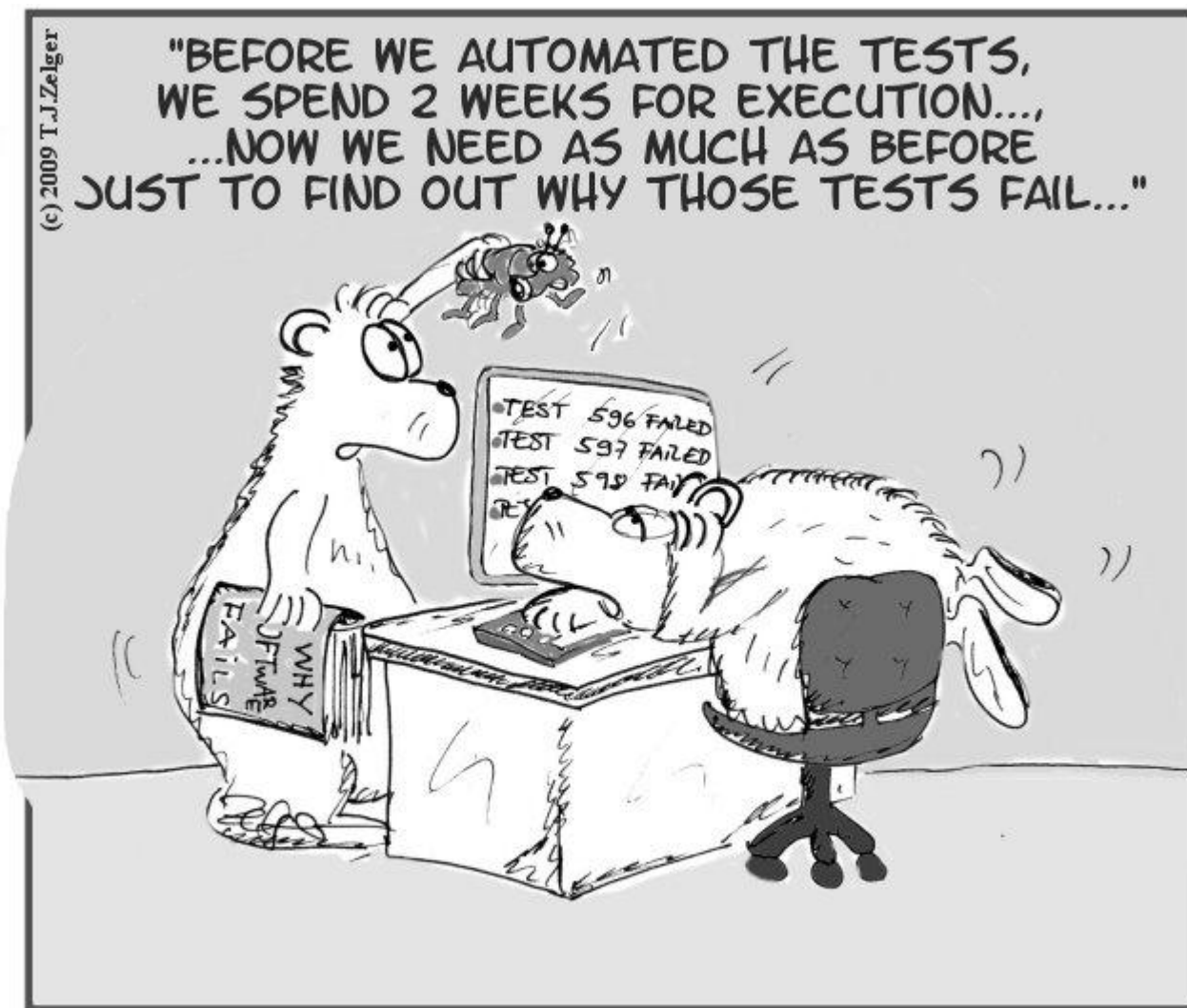
## Tab. Przykładowy przypadek testowy

Name:	Login page test	
Summary:	This test verifies if login functionality works properly	
Preconditions:	User has access to application	
Step:	Actions:	Expected results:
1	Open application	Application is opened and login screen is displayed
2	Enter a valid username into username input field	Username is properly entered
3	Enter a valid password into password input field	Password is properly entered
4	Click on 'Login' button	User is logged into application

# Dlaczego testy automatyczne?

- uzupełniają (nie zastępują całkowicie!) testy manualne
- są powtarzalne
- mogą być wykonywane systematycznie (np. codziennie)
- oszczędzają czas testerów
- dają aktualny stan oprogramowania – szybka informacja, że coś przestało działać
- redukują błędy związane z czynnikiem ludzkim

źródło: [www.simply-the-test.blogspot.com](http://www.simply-the-test.blogspot.com)



**Test Automation side effect**

- **koszt implementacji**
  - skrypty testowe
  - dokumentacja
- **koszt eksploatacji**
  - obsługa środowiska testowego
  - analiza wyników
- **koszt utrzymania**
  - wprowadzanie zmian do testów wraz ze zmianami produktu

**Kiedy warto automatyzować testy?**

**Gdy nam się to opłaca!**

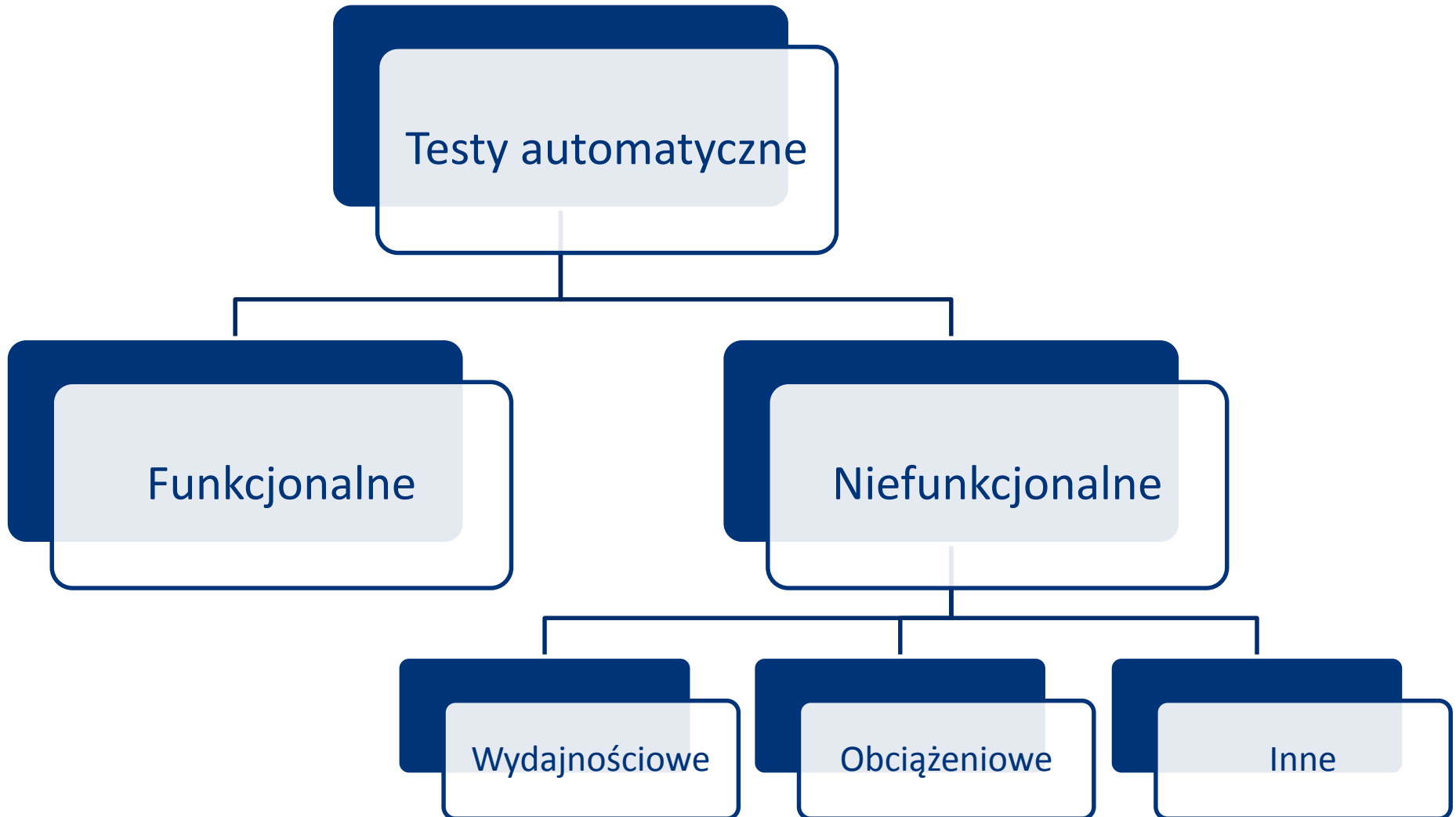


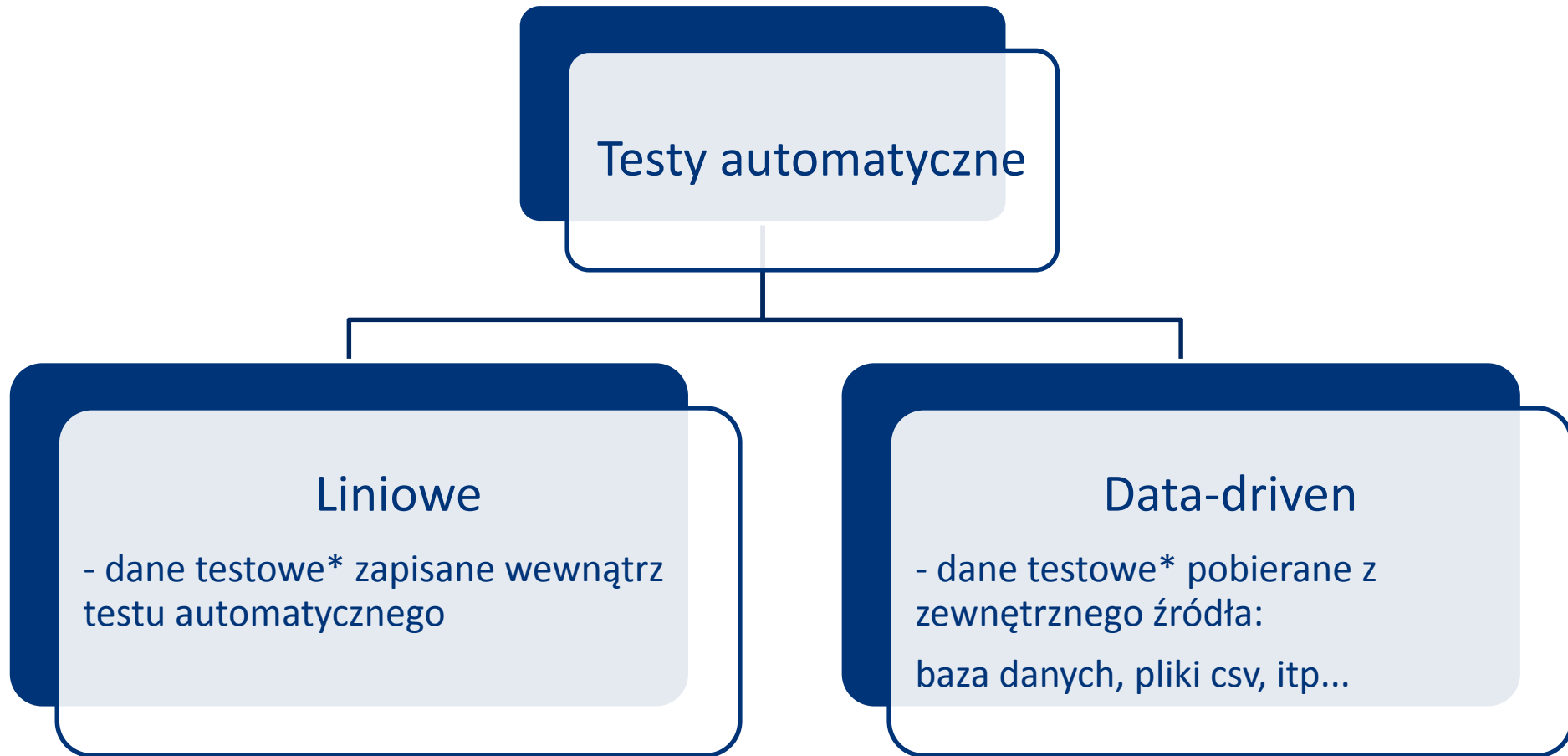
- **testy wykonywane wiele razy**
  - im więcej, tym większy zysk z automatyzacji
- **testy dotyczące krytycznych części oprogramowania**
  - wiążą się z dużym kosztem regresji
- **testy wrażliwe na błędy związane z czynnikiem ludzkim**
- **testy wykonywane dla wielu danych testowych lub konfiguracji**

# Kiedy i jakie testy automatyzować? (2) k a i n s®

---

- **testy dotyczące rzadko zmieniających się obszarów oprogramowania**
- **testy monotonne lub pochłaniające dużo czasu, gdy są wykonywane ręcznie**
  - np. wymagające wprowadzania dużej ilości danych
- **testy niemożliwe do wykonania w sposób manualny**
- **gdy wynik testu może być stwierdzony w sposób obiektywny**
  - działa/nie działa, a nie ładny/brzydki





\* dane testowe - np. nazwa użytkownika i hasło z przykładu

Tab. Przykładowy przypadek testowy

Name:	Login page test	
Summary:	This test verifies if login functionality works properly	
Preconditions:	User has access to application	
Step:	Actions:	Expected results:
1	Open application	Application is opened and login screen is displayed
2	Enter a valid <b>username</b> into username input field	Username is properly entered
3	Enter a valid <b>password</b> into password input field	Password is properly entered
4	Click on 'Login' button	User is logged into application

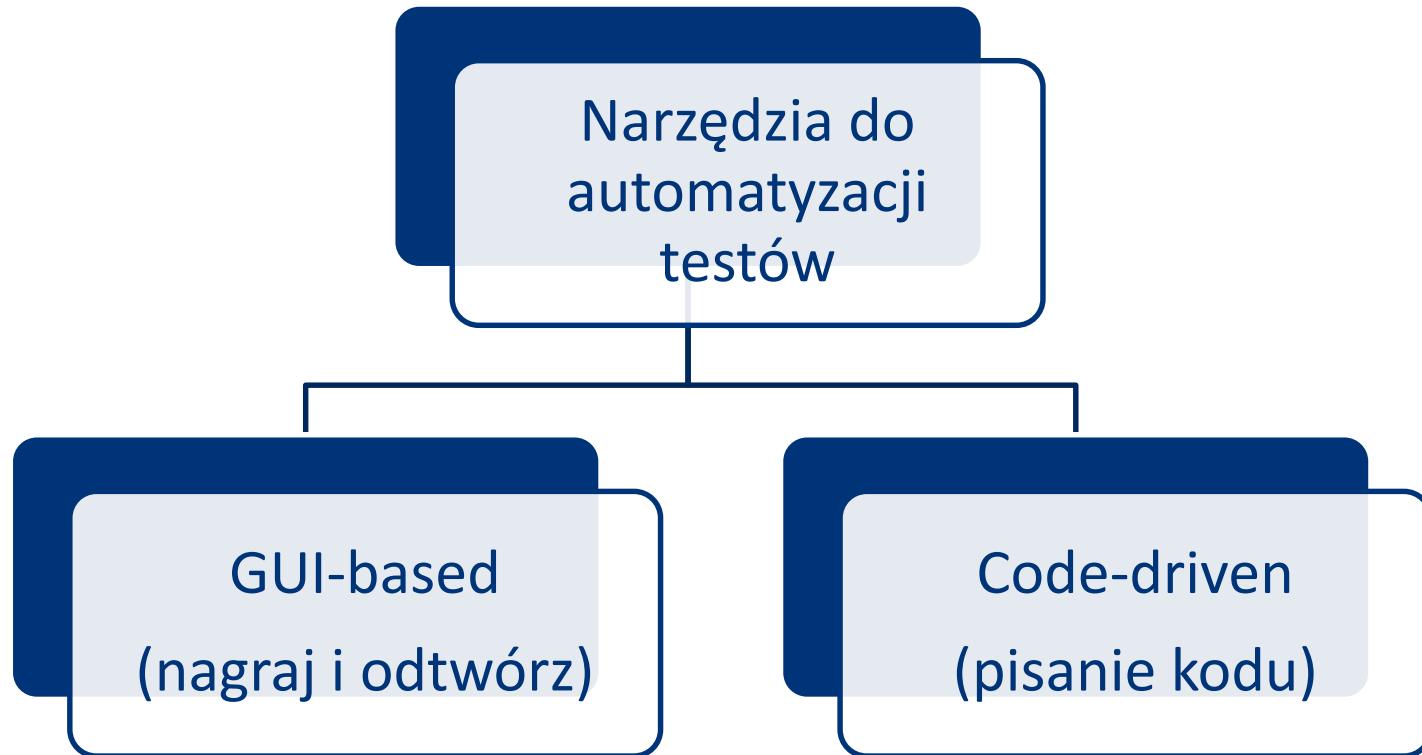
- **Liniowe:**

- + prostota
- + szybkość implementacji
- + niezależność testu
- nadmiarowość
- niska konfigurowalność
- niska elastyczność
- w dużej mierze zależne od środowiska testowego
- trudne do zarządzania

- **Data-driven:**

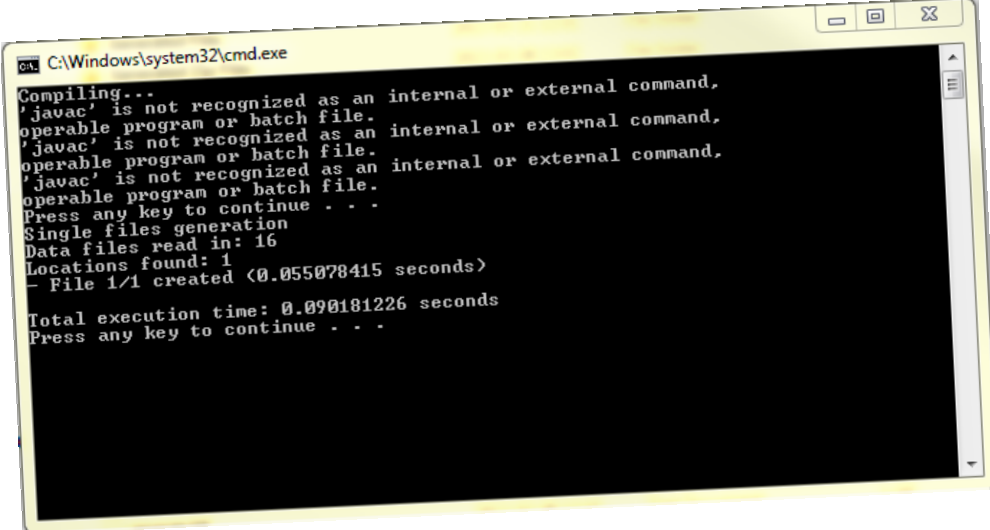
- + łatwe do ponownego użycia w różnych środowiskach (*reusability!*)
- + mniejsza nadmiarowość
- + konfigurowalność
- + elastyczność
- bardziej rozbudowana struktura
- zależność od źródła danych testowych







- Pliki wsadowe
- Makra
- Języki skryptowe:
  - Bash
  - Python
  - Pearl
  - ...



```
ca: C:\Windows\system32\cmd.exe
Compiling...
'javac' is not recognized as an internal or external command,
operable program or batch file.
'javac' is not recognized as an internal or external command,
operable program or batch file.
'javac' is not recognized as an internal or external command,
operable program or batch file.
Press any key to continue . . .
Single files generation
Data files read in: 16
Locations found: 1
- File 1/1 created (0.055078415 seconds)
Total execution time: 0.090181226 seconds
Press any key to continue . . .
```

- **HP-QuickTest Professional**

- rozbudowane narzędzie do przeprowadzania automatycznych testów funkcjonalnych oprogramowania



- **Selenium**

- darmowe narzędzie do automatycznych testów funkcjonalnych aplikacji webowych
- wsparcie dla wielu przeglądarek
- wsparcie dla języków programowania:
  - Java
  - C#
  - Perl
  - PHP
  - Python
  - Ruby



<http://kainos.pl/blog/>

- **Monkey Talk**

- darmowe narzędzie do automatyzacji testów aplikacji mobilnych
- wsparcie dla Android i iOS
- wsparcie dla HTML5 i Flex



- **NeoLoad**

- narzędzie do wykonywania automatycznych testów obciążeniowych aplikacji webowych
- nagrywanie scenariuszy (*request-response*) i ich odtwarzanie dla dużej ilości użytkowników
- rozbudowane statystyki obciążenia serwerów



# Pytania?

k a i n o s®

