

Automatyzacja testowania

Trener: Piotr Krzosa

Poznajmy się

- Mam na imię ...
- Testuję od ... lat
- Programuję/ nie programuję
- Automatyzuję/ nie automatyzuję
- Moim celem jest ...



Zadanie:

1. W oparciu o zdobytą wiedzę o testowaniu oprogramowania przygotuj opis „Automatyzacji testów”

Co to jest?

Jak z niej korzystać?

Kiedy automatyzować?



1. Wprowadzenie do automatyzacji



Automaty

- Czym jest automatyzacja?



Definicja

- **automatyzacja:** to znaczne ograniczenie lub zastąpienie (proces zastępowania) ludzkiej pracy fizycznej i umysłowej przez pracę maszyn działających na zasadzie samoregulacji i wykonujących określone czynności bez udziału człowieka. Również zastosowanie maszyn do pracy niemożliwej do wykonania w inny sposób.

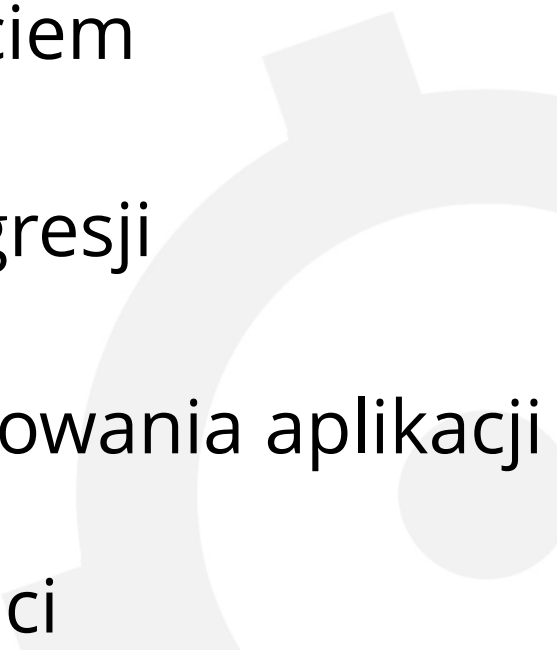
[Wikipedia]

Definicja

- **automatyzacja testowania:** Użycie oprogramowania do wykonania lub wspierania czynności testowych, np. zarządzania testami, projektowania testów, wykonania testów i sprawdzania wyników.

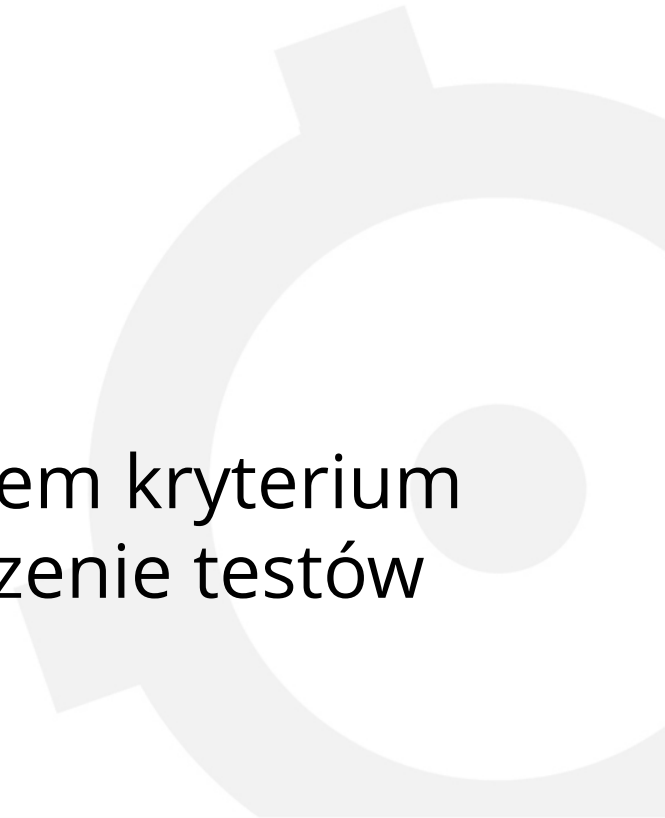
*[Słownik wyrażeń związanych z
testowaniem Wersja 2.0]*

Dlaczego potrzebujemy automatyzować?

- Coraz bardziej złożone systemy
 - Krótszy czas na testowanie
 - Coraz większa zależność między oprogramowaniem i ludzkim życiem
 - Kryzys na rynku (?)
 - Konieczność częstych testów regresji
 - Zmienne wymagania
 - Zmienność środowiska funkcjonowania aplikacji (bezpieczeństwo, wydajność)
 - Coraz bardziej wymagający klienci
- 

Automatyzacja w procesie testowym

- Proces
 - Planowanie testów
 - Projektowanie testów
 - Wykonanie testów
 - Rejestracja wyników testów
 - Raportowanie testów
 - Sprawdzenie testów pod kątem kryterium zakończenia testów i zakończenie testów



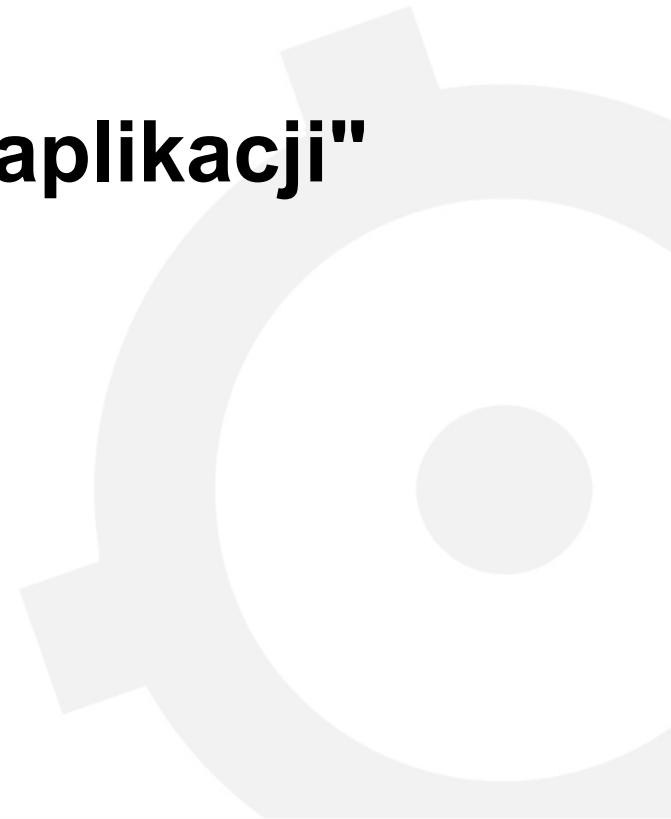
Automatyzacja w cyklu życia produktu informatycznego

- Testy jednostkowe
- Testy integracyjne
- Testy systemowe
- Testy akceptacyjne

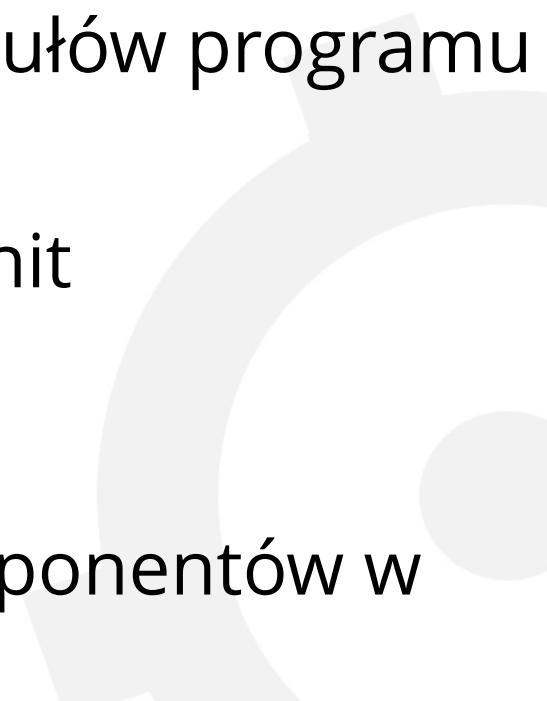


Automatyzacja w testach jednostkowych

jakosc_kodu != "Jakość aplikacji"



Automatyzacja w testach jednostkowych

- Analiza statyczna kodu oraz pokrycie kodu
 - Testowanie pojedynczych modułów programu
 - Testy w oparciu o aplikacje XUnit
 - JUnit, CUnit, NUnit...
 - Testowanie pojedynczych komponentów w separacji - Zaślepki (Stubs)
- 

Automatyzacja w testach integracyjnych

- Testowanie komunikacji między modułami programu przy pomocy samodzielnie pisanego kodu
- Testy w oparciu o aplikacje XUnit i środowisko testów integracyjnych

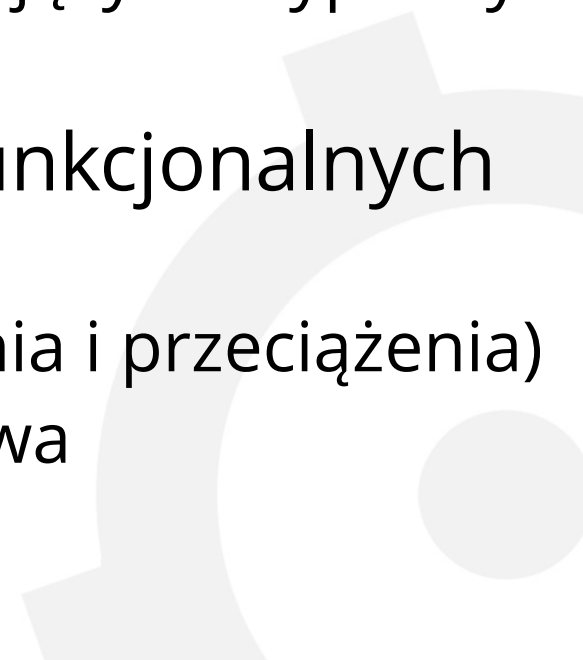


Mock up dla web serwisów

- Dlaczego
 - Testowanie przed produkcją
 - Opóźnienie
 - Testowanie offline



Automatyzacja w testach systemowych

- Testowanie funkcji systemu w oparciu o wymagania użytkownika lub wzorce rynkowe
 - Skrypty automatyczne oparte o język skryptowy
 - Narzędzia Capture / Re-play
 - Testowanie parametrów niefunkcjonalnych systemu:
 - Narzędzia wydajności (obciążenia i przeciążenia)
 - Narzędzia testów bezpieczeństwa
 - Narzędzia użyteczności
 - Walidatory
- 

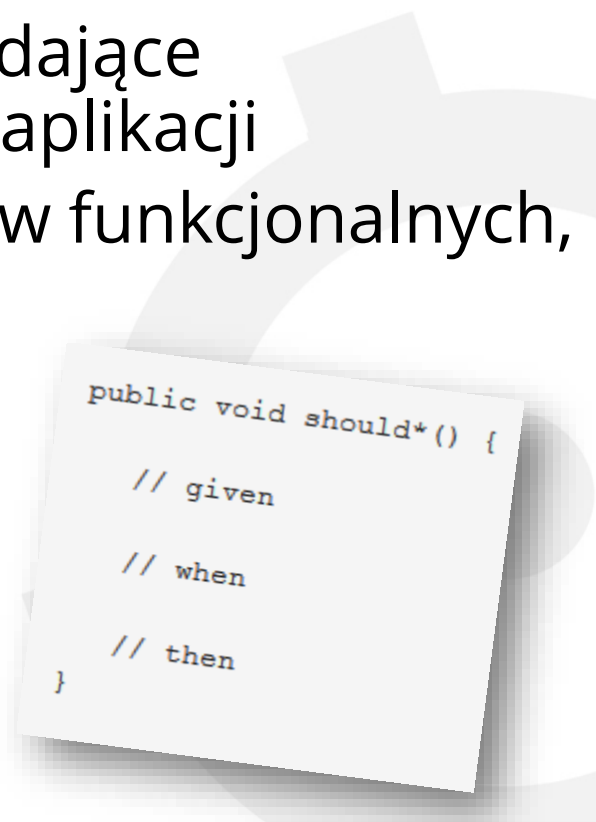
Automatyzacja w testach akceptacyjnych

- Elementy

- wykonuje się wszystkie skrypty przygotowane w ramach rozwoju oprogramowania
- uruchamia się skrypty odpowiadające najważniejszemu przepływowi w aplikacji
- testowanie zarówno parametrów funkcjonalnych, jak i нефunkcjonalnych

- ATDD oraz BDD

- Cucumber
- JBehave



```
public void should*() {  
  
    // given  
  
    // when  
  
    // then  
  
}
```


Ocena gotowości do automatyzacji

- Czy jesteśmy gotowi do wdrożenia automatyzacji?

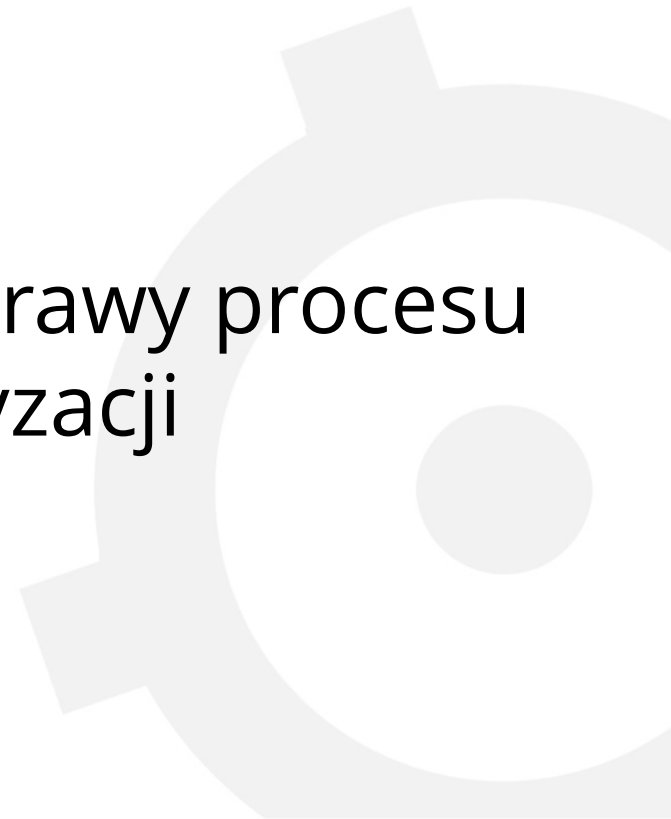


Analiza SWOT

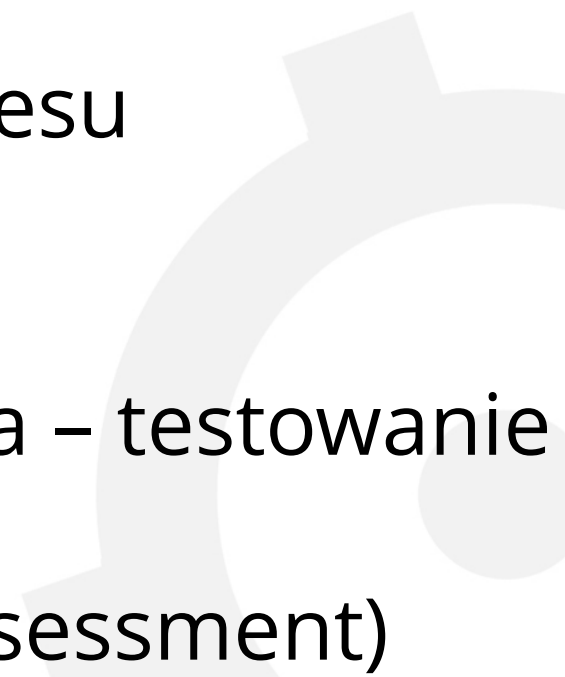


Zadanie: Przeprowadź analizę SWOT

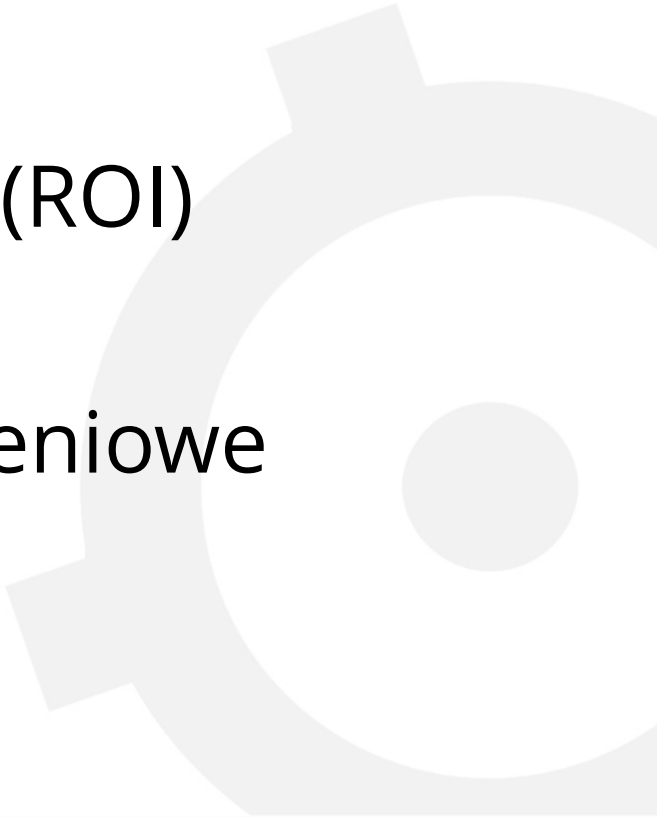
- W zespołach przeanalizujcie wdrożenie automatyzacji
- Zidentyfikuj szanse do poprawy procesu dzięki wdrożeniu automatyzacji



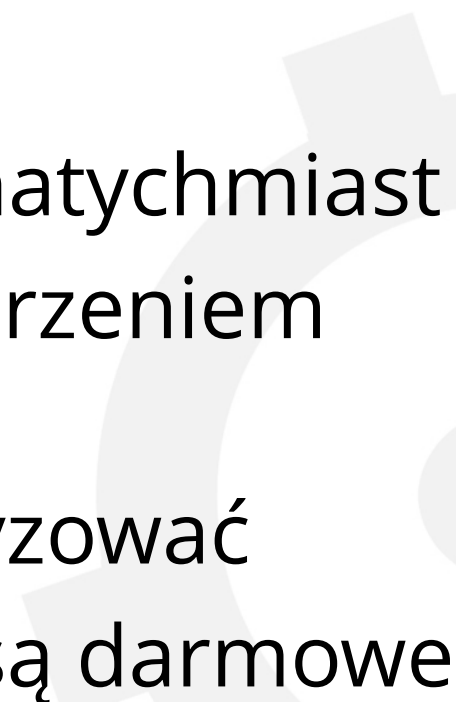
Metodologia wdrażania automatyzacji

- Faza 1: Decyzja o automatyzacji
 - Faza 2: Wybór, sprawdzenie i zakup testowego narzędzia (POC)
 - Faza 3: Wprowadzenie procesu automatyzacji
 - Faza 4: Szkolenia
 - Faza 5: Wdrożenie narzędzia – testowanie automatyczne
 - Faza 6: Przegląd i ocena (assessment)
- 

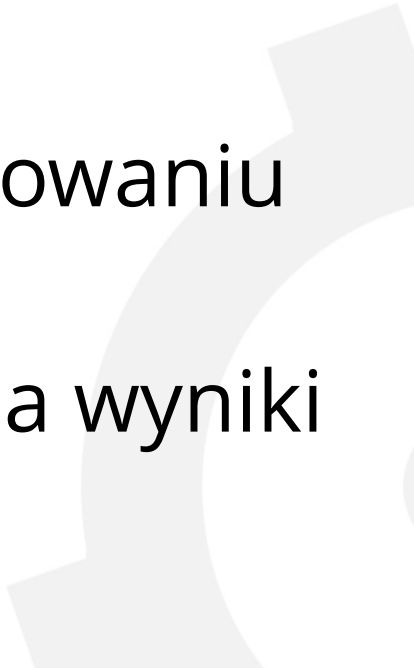
Zadanie: Wdrożenie

- Określ wymagania na narzędzie
 - Określ koszty / określ zyski (ROI)
 - Zidentyfikuj potrzeby szkoleniowe
- 

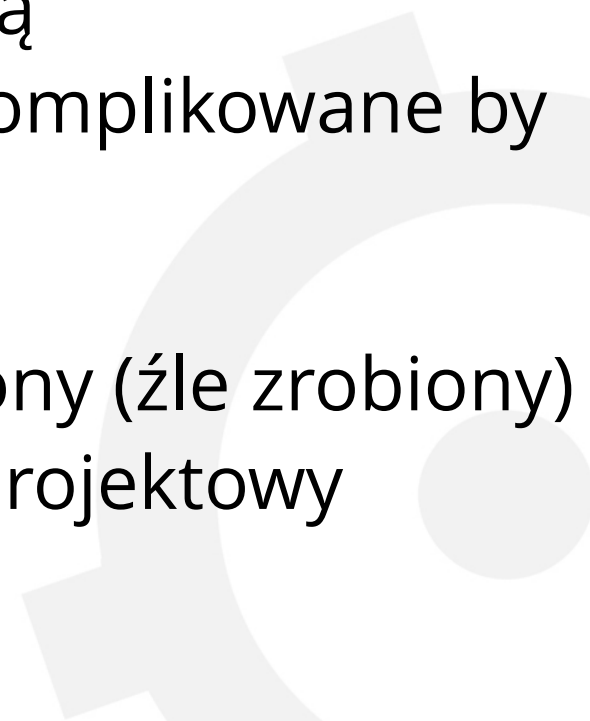
Mity automatyzacji

- Zastąpi testy manualne
 - Capture / Re-play – lekarstwo na wszystko
 - Automatyzacja zwróci się natychmiast
 - Automatyzacja nie jest tworzeniem oprogramowania
 - Wszystko da się zautomatyzować
 - Rozwiązania open source są darmowe
- 

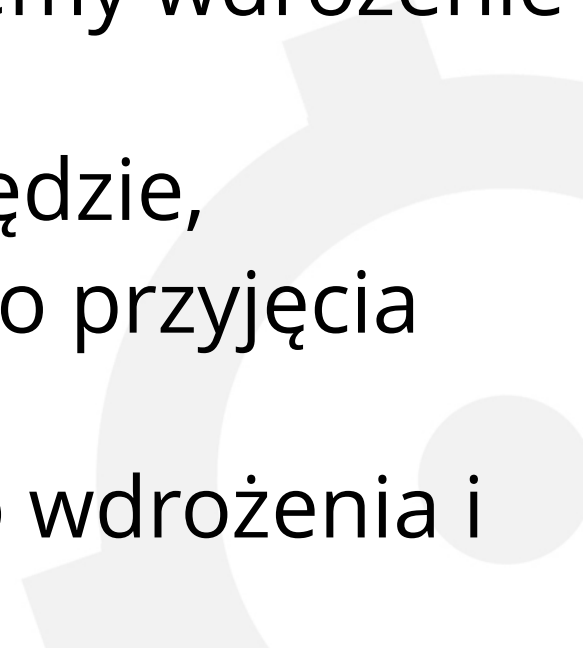
Wady automatyzacji

- Niemożność stosowania w zmiennym środowisku
 - Ślepe i bezmyślne testy
 - Możliwe błędy w oprogramowaniu szukającym błędów
 - Narzędzia mogą wpływać na wyniki testów
 - Kosztowna (?)
- 

Dlaczego automatyzacja kończy się porażką?

- Złe przygotowanie do automatyzacji!
 - Prawda, którą dostrzegamy zbyt późno:
 - Narzędzia testowe nie działają
 - Narzędzia testowe są zbyt skomplikowane by ich używać
 - Narzędzia są zbyt drogie
 - Obiekt testowy jest zbyt złożony (źle zrobiony)
 - Zbyt napięty harmonogram projektowy
 - Brak wsparcia MGMT
 - Zbyt duży opór testerów
- 

Automatyzacja jest prosta, gdy:

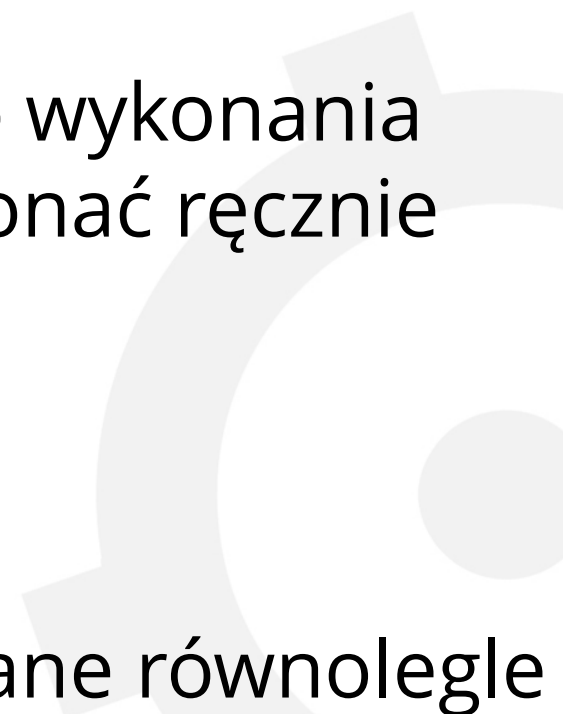
- ... wiemy, co chcemy zautomatyzować,
 - ... znamy ograniczenia narzędzi,
 - ... profesjonalnie przygotujemy wdrożenie narzędzia lub metody,
 - ... wdrożymy właściwe narzędzie,
 - ... organizacja jest gotowa do przyjęcia narzędzia,
 - ... powołamy organizację do wdrożenia i utrzymania narzędzia.
- 

Zalety dobrej automatyzacji

- Szybsze testowanie
- Praca 365 dni w roku 24 godziny na dobę
- Więcej negatywnych testów
- Utrzymywanie stałej jakości
- Większe pokrycie testowe
- Większa niezawodność
- Precyzyjniejsze testowanie
- Taniej (?)

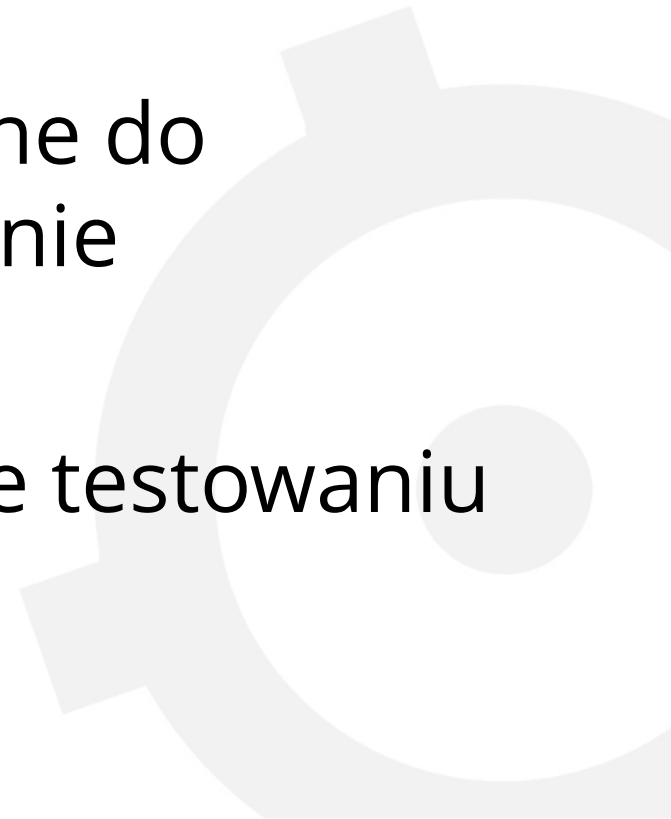


Co automatyzować?

- Czynności, które są wykonywane często
 - Czynności, które są trudne do wykonania ręcznie lub nie da się ich wykonać ręcznie
 - Czynności, które trwają długo
 - Czynności, które są wykonywane równolegle
- 

Czego nie automatyzować?

- Czynności, które wykonywane są rzadko
- Kiedy wyniki testów są trudne do zweryfikowania automatycznie
- Kiedy programy podlegające testowaniu ulegają częstym zmianom



Dobór podejścia do automatyzacji testów

- Capture/playback
- Linear scripting
- Structured scripting
- Data-driven testing
- Keyword-driven testing
- Process-driven scripting
- Model-based testing



Dyskusja: Odpowiednie podejście

- Jakie wybrać podejście do automatyzacji testów?



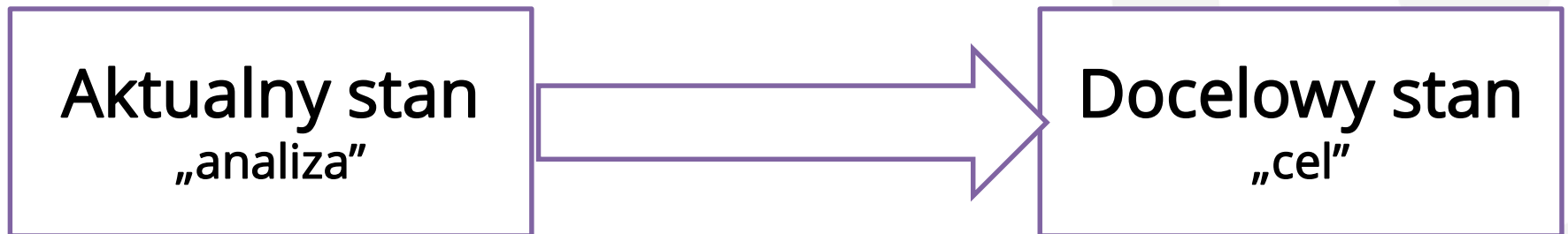
Czynności w ramach automatyzacji

1. Cel, strategia i planowanie
2. Projektowanie
3. Monitorowanie i kontrola
4. Utrzymanie

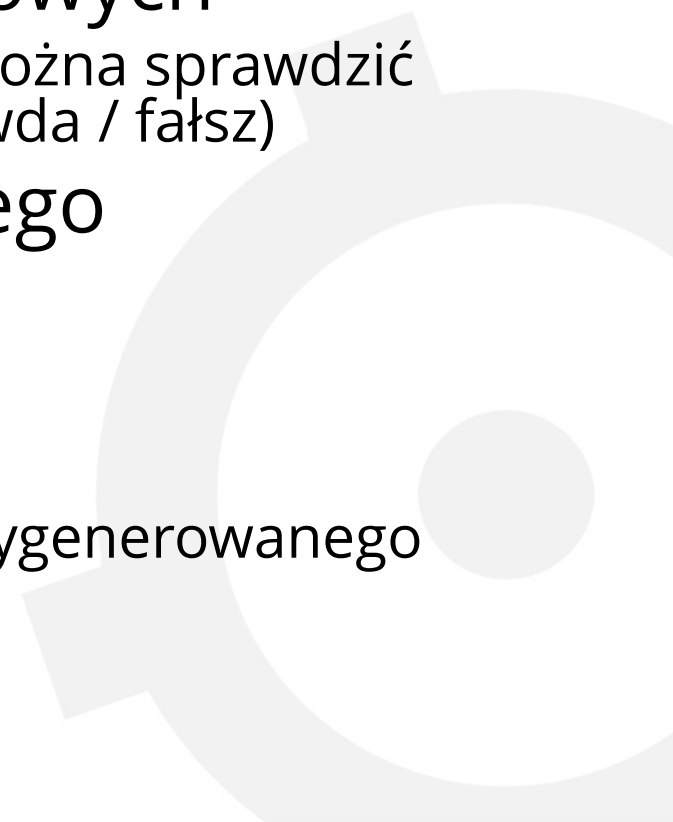


Strategia

- Cel:
 - Co chcemy osiągnąć?
- Plan
 - Zrozumienie kontekstu
 - Zdefiniowanie zakresu
- Środowisko / Zasoby / Koszty

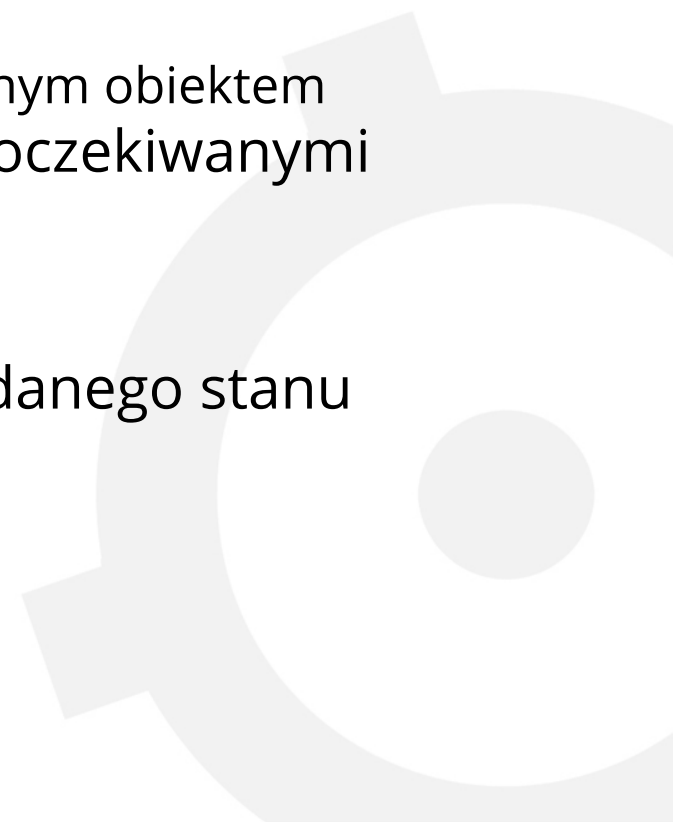


Projektowanie (1/2)

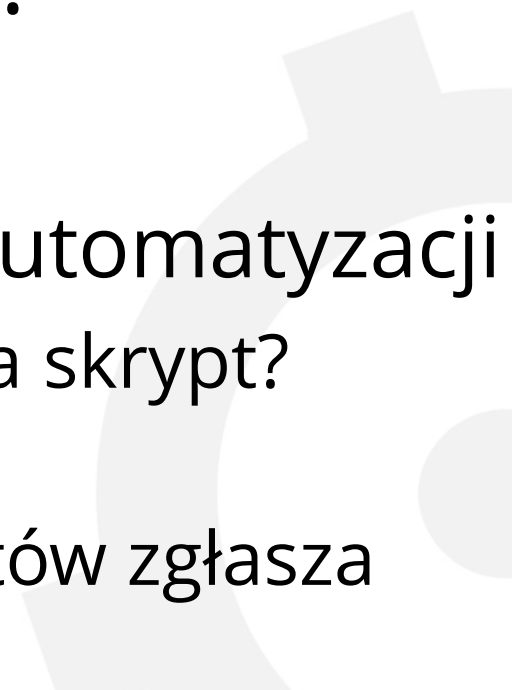
- Przygotowanie środowiska
 - Ustawienie pożądanego stanu systemu
 - Identyfikacja warunków testowych
 - Warunki testu to zdarzenia, które można sprawdzić (funkcja zaimplementowana -> prawda / fałsz)
 - Zbudowanie skryptu testowego
 - Weryfikatory czy asercje?
 - Jak wiele?
 - Margines błędu?
 - Czas odpowiedzi, poprawność wygenerowanego pliku etc.
- 

Projektowanie (2/2)

- Uruchomienie skryptu
 - Test na test:
 - uruchomienie skryptu bez obiektu testowego
 - uruchomienie skryptu z obiektem
 - uruchomienie skryptu z zmodyfikowanym obiektem
 - Porównanie uzyskanych wyników z oczekiwanymi
- Posprzątanie środowiska
 - Ustawienie początkowego lub pożądanego stanu systemu
 - Usuwanie zbędnych danych
 - Plików tymczasowych
 - Czyszczenie pól formularzy, itp.



Monitorowanie i kontrola

- Czy skrypt ciągle działa?
 - Określanie skuteczności automatyzacji
 - Jak wiele defektów wykrywa skrypt?
 - Jak wiele nieprawdziwych (niepotwierdzonych) defektów zgłasza skrypt
- 

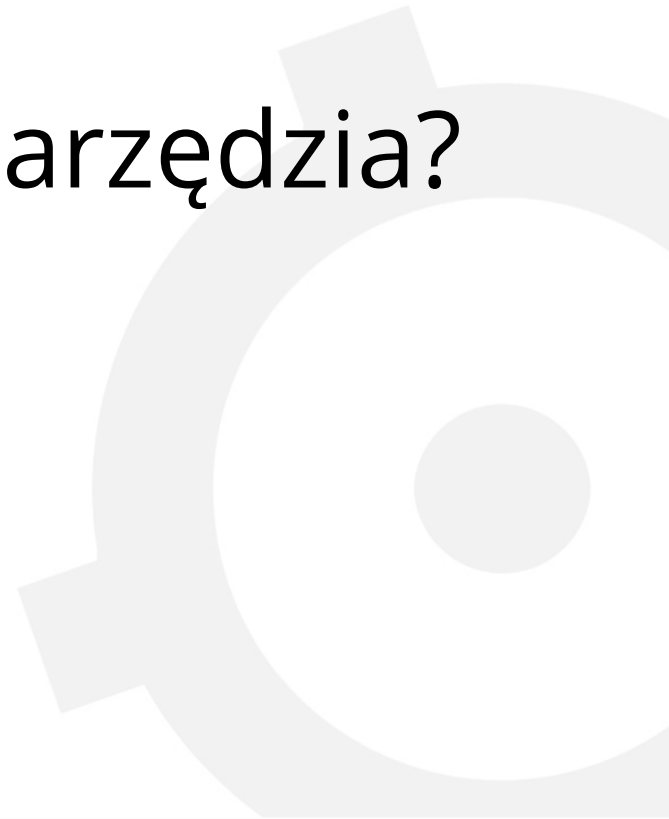
Utrzymanie

- Co oznacza utrzymanie?
 - Jak wspomóc utrzymanie?
 - *Durability*
- Koszty utrzymania



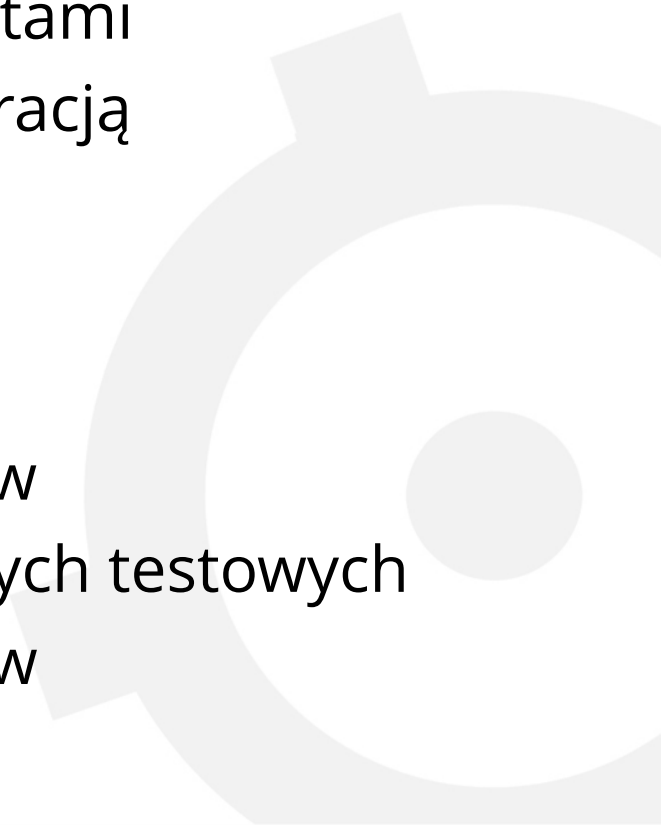
Dyskusja: Klasyfikacja narzędzi

- Jak można podzielić narzędzia?

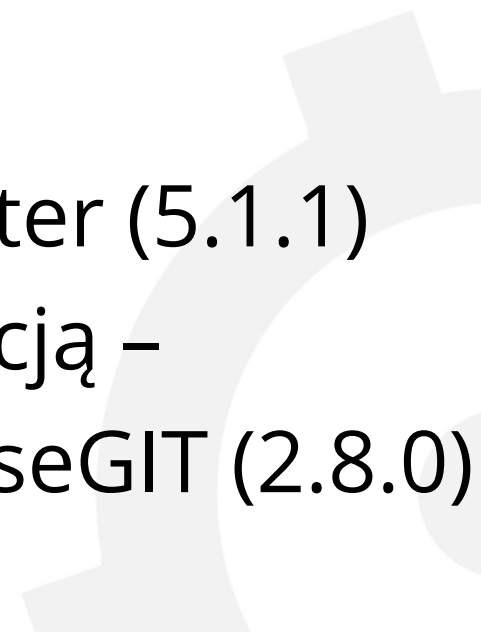


Klasyfikacja narzędzi w zależności od przeznaczenia

- Narzędzia w zarządzaniu testami
- Narzędzia w zarządzaniu wymaganiami
- Narzędzia w zarządzaniu incydentami
- Narzędzia w zarządzaniu konfiguracją
- Narzędzia modelowania
- Narzędzia analizy statycznej
- Narzędzia analizy dynamicznej
- Narzędzia w projektowaniu testów
- Narzędzia do przygotowania danych testowych
- Narzędzia do wykonywania testów



Praca domowa

- System - Powershell
 - REST - Postman (7.3.5)
 - SOAP - SoapUI (5.5.0)
 - Testy wydajności - JMeter (5.1.1)
 - Zarządzanie konfiguracją –
GIT (2.22.0) + TortoiseGIT (2.8.0)
 - Bazy danych - SQL
- 

Kontakt

- piotr.krzosa@testerzy.pl

