



# Cykl wytwarzania (życia) oprogramowania c.d.

*Sztuka Wytwarzania Oprogramowania, w. 7 cd*

Konrad Grochowski

Instytut Informatyki, Politechnika Warszawska, 2024 ©





# Podział testów

- › Ze względu na badane aspekty:
  - Testy funkcjonalne (functional testing)
  - Testy wydajnościowe (performance testing)
  - Testy przeciążeniowe (stress testing)
  - Testy bezpieczeństwa (security testing)
  - Testy bezpieczeństwa (safety testing)
  - Testy niezawodności (dependability testing)
  - Testy odporności (recovery testing)
  - Testy zgodności (compatibility testing)
  - Testy dokumentacji (documentation testing)
  - ...



## Inne przykłady rodzajów testów

- › Testy dymne (smoke tests)
- › Testy regresji (regression testing)
- › Testy destrukcyjne (destructive testing)
- › Testy „interakcji z człowiekiem”
  - Testy tłumaczeń i regionalizacji (localization and internationalization)
  - Testy używalności (usability)
  - Testy dostępności (accessibility)



## Dobry test

- › Jednoznacznie ustalony cel testu
- › Czytelny scenariusz testu
- › Określony warunek końca testu
- › Określony warunek oceny testu (zaakceptowania)
- › Jeśli możliwe – odwołanie do wymagań, z których test wynika.
- › *Idealnie:*
  - Automatyczny
  - Deterministyczny i powtarzalny
  - Jednoznacznie i łatwo oceniany
  - Szybki



# Testowanie jednostkowe

- › „Najwyższa stopa zwrotu w jakości”
  - Krótka ścieżka pomiędzy inwestycją/czasem a wynikiem i wpływem na kod
- › „Najlepszy przyjaciel programisty”
  - Tylko test mówi, co kod naprawdę robi, więc test jednostkowy pozwala wiedzieć co jest zrobione „na koniec dnia”
- › Wymusza lepszą architekturę (decoupling)
- › Brak formalnej definicji jednostki
- › Nie wszystko da się unit-testować
- › Nie zawsze odpowiada bezpośrednio wymaganiom
  - Usunie drobne błędy, ale nie poważne naruszenia funkcjonalności



# Najlepsze praktyki tworzenia testu (jednostkowego)

- › Automatyczny
- › Powtarzalny i deterministyczny
- › Szybki
- › Jednoznaczny
- › Skoncentrowany na pojedynczym aspekcie
- › Czytelny (testy to najlepsza dokumentacja kodu)
- › Niezależny
  
- › *Nieidealny test zazwyczaj jest lepszy niż brak testu*
- › *Zły test obniża morale i często przeszkadza*

Dziękuję za uwagę

[Konrad.Grochowski@pw.edu.pl](mailto:Konrad.Grochowski@pw.edu.pl)

