

Aplikacje WWW



Wykład 9 – Testowanie oprogramowania, xUnit

dr hab. inż. Marcin Zalasinski, prof. PCz

2024

▶ Wprowadzenie

- ▶ Zmiany wprowadzone w oprogramowaniu o dowolnej złożoności mogą spowodować wystąpienie nieoczekiwanych błędów.
- ▶ Dlatego testowanie aplikacji po wykonaniu modyfikacji jej kodu jest wymagane praktycznie zawsze.
- ▶ Wyróżniamy dwa rodzaje testów:
 - ▶ Manualne
 - ▶ Automatyczne

► Testy manualne

Pros

- ▶ Dokładne informacje zwrotne
- ▶ Brak konieczności zakupu oprogramowania do testów
- ▶ Doświadczony tester manualny może być pomocny w przypadku złożonych problemów
- ▶ Testy manualne mogą być szybko wdrożone

Cons

- ▶ Test wykonywany przez człowieka jest podatny na błędy
- ▶ Nie mogą być automatycznie zapisywane w celu ich późniejszego dokładnego odtworzenia lub przeanalizowania
- ▶ W przypadku niektórych zadań uzyskanie wyniku jest czasochłonne

Testy automatyczne

Pros

- ▶ Można szybko otrzymywać dokładne wyniki dla złożonych scenariuszy testowych
- ▶ Proces ten może być łatwo monitorowany, można go później przeanalizować i odtworzyć
- ▶ Eliminują problemy związane np. ze zmęczeniem pracownika
- ▶ Zakres testowanych czynników może być duży – program nie zapomni o sprawdzeniu nawet najmniejszych elementów

Cons

- ▶ Bez elementu ludzkiego trudno jest uzyskać odpowiednie wnioski dotyczące wizualnych aspektów (np. interfejsu użytkownika itp.)
- ▶ Narzędzia do przeprowadzania testów są często drogie
- ▶ Narzędzia do automatyzacji testów nie są niezawodne – każde z nich ma swoje ograniczenia, a ich skrypty też są podatne na błędy

▶ Testy bez testera manualnego?

- ▶ Oprogramowanie testujące potrzebuje testera manualnego, a tester manualny potrzebuje automatycznych testów.
- ▶ Warto pozostawić niektóre zadania skryptowi, zwłaszcza jeśli są stosunkowo proste, jednak wymagają dużej liczby powtórzeń.
- ▶ Z drugiej strony nic nie zastąpi doświadczenia i analizy wykonywanej przez branżowego specjalistę.
- ▶ Tester manualny jest w stanie przeanalizować złożone problemy i przygotować skrypty testujące specjalnie pod dany scenariusz.
- ▶ W praktyce warto stosować różne rodzaje testów przy sprawdzaniu jakości i funkcjonalności, gdyż wzajemnie się uzupełniają.

▶ Testy jednostkowe

- ▶ Testują pojedynczą część logiki aplikacji
- ▶ Działa całkowicie w pamięci, nie komunikuje się z systemem plików lub bazą danych
- ▶ Testują jednostkę kodu, bez zewnętrznych zależności – w związku z tym powinny działać niezwykle szybko
- ▶ Bazują na podaniu pewnych danych wejściowych i sprawdzeniu czy wartości danych wyjściowych są zgodne z oczekiwaniem
- ▶ Powinny być uruchamiane najlepiej przed każdym przesłaniem danych do repozytorium z kontrolą źródła

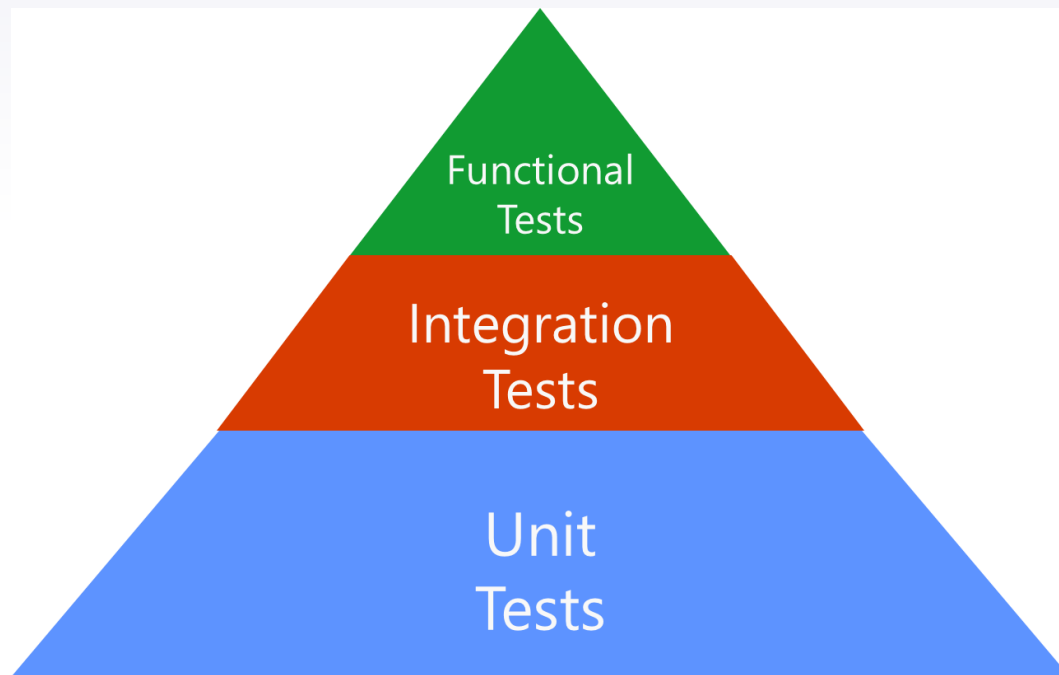
▶ Testy integracyjne

- ▶ Dotyczą intergracji poszczególnych składowych systemu, np. związanych dostępem do danych czy systemu plików
- ▶ Mogą dotyczyć współpracy pomiędzy poszczególnymi warstwami oprogramowania
- ▶ Są zwykle wolniejsze i trudniejsze do skonfigurowania niż testy jednostkowe
- ▶ Na przykład test integracji, który jest zgodny z rzeczywistą bazą danych, będzie wymagał sposobu na przywrócenie bazy danych do znanego stanu przed każdym uruchomieniem testu
- ▶ W miarę dodawania nowych testów i ewolucji schematu produkcyjnej bazy danych, skrypty testowe będą miały tendencję do zwiększania rozmiaru i złożoności

▶ Testy funkcjonalne

- ▶ Testy funkcjonalne są pisane z perspektywy użytkownika i weryfikują poprawność systemu na podstawie jego wymagań
- ▶ Testy funkcjonalne działają na poziomie systemu, mogą więc wymagać pewnego stopnia automatyzacji interfejsu użytkownika
- ▶ Podobnie jak testy integracyjne, zwykle działają również z pewnego rodzaju infrastrukturą testową

► Piramida testowa



Źródło: [2]

xUnit.net

- ▶ Jest bezpłatnym narzędziem open-source do przeprowadzania testów jednostkowych w środowisku .NET
- ▶ Jest napisany przez twórcę narzędzia NUnit v2
- ▶ Jest najnowszym narzędziem do wykonywania testów jednostkowych dla języków C#, F#, VB.NET
- ▶ Jest udostępniany na licencji Apache 2

► Przykład



► WebApplicationFactory

- Do poczytania:

<https://www.codeproject.com/Articles/5377616/WebApplicationFactory-in-ASP-NET-Core-Practical-Ti>

► Literatura

- ▶ [1] <https://docs.microsoft.com/>
- ▶ [2] <https://smartbees.pl/blog/testy-manualne-vs-automatyczne-poznaj-kluczowe-roznice>