



Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Technologia programowania

Testowanie oprogramowania

Anna Lauks-Dutka

Politechnika Wroclawska

16 grudnia 2024



Czym jest testowanie?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Testowanie

Jeden z procesów zapewniania jakości oprogramowania – aktywność zmierzająca do **wykrycia usterek** w implementowanym programie (systemie)

Inne praktyki mające na celu zapewnienie jakości

- zarządzanie konfiguracją - kontrolowanie i zarządzanie zmianami w kodzie, dokumentacji, środowisku, aby zapewnić spójność (np. SVN, GIT)
- regularne przeglądy kodu (ang. *code review*), programowanie w parach
- dokumentowanie kodu
- ...



Czy testowanie jest istotne?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

- 1 Bez testów stworzenie działającej wersji programu jest **prawie niemożliwe!**¹
- 2 **Testowanie = oszczędność czasu i kosztów**
(pozwala zidentyfikować błędy na wcześniejszych etapach)
- 3 Mniejsza liczba błędów pozytywnie wpływa na **zadowolenie użytkowników i reputację firmy**

¹Teoretycznie możliwe, ale ryzykowne, jeżeli chcemy zapewnić jak najwyższą jakość swoich produktów



Co możemy testować?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

funkcjonalność, poprawność - zgodność faktycznego zachowania programu z oczekiwanym (w tym poprawność względem norm prawnych)

użyteczność - intuicyjność interfejsu aplikacji, czytelność projektu graficznego, zgodność z zasadami dostępności

zgodność - poprawność działania w różnych środowiskach (różne systemy operacyjne, urządzenia, przeglądarki)

wydajność - spełnienie wymagań czasowych

niezawodność - odporność na działanie pod wymaganym obciążeniem

bezpieczeństwo - odporność na ataki, wykrycie luk w zabezpieczeniach



Pojęcia

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Usterka (ang. *fault, bug*)

Błąd projektu bądź też błąd w kodzie będący przyczyną wejścia systemu w **błędny stan**

Błędny stan (ang. *erronous state, error*)

Manifestowanie się usterki w czasie wykonywania programu. Jego następstwem może być **awaria**

- Uwaga: nie każdy błędny stan musi doprowadzić do awarii np. drobne nieprawidłowości w interfejsie, działanie nieoptymalne

Awaria (ang. *failure*)

Sytuacja, kiedy oprogramowanie przestaje działać zgodnie z oczekiwaniami lub przestaje spełniać swoje zadania.



Plan testów

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Planowanie testów

Przydzielenie zasobów (ludzkich, sprzętowych) na potrzeby testowania i określenie harmonogramu testów

Plan testów

Dokument opisujący, w jaki sposób będą prowadzone testy, kto będzie je przeprowadzał, co będzie testowane, jak długo potrwa cały proces, jaki będzie zakres testów

- analiza
- cele
- strategia
- planowanie zasobów
- przypadki testowe
- priorytety
- harmonogram



Czy można przetestować system w pełni?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Należy pamiętać

- testowanie to „szukanie dziury w całym”
- dane testowe obejmują nie tylko dane poprawne, ale też wartości graniczne oraz dane niepoprawne potencjalnie niebezpieczne



Czy można przetestować system w pełni?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Należy pamiętać

- testowanie to „szukanie dziury w całym”
- dane testowe obejmują nie tylko dane poprawne, ale też wartości graniczne oraz dane niepoprawne potencjalnie niebezpieczne

Wyczerpujące (obejmujące wszystkie kombinacje danych wejściowych) przetestowanie niebanalnego systemu jest w praktyce zadaniem niewykonalnym!



Czy można przetestować system w pełni?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Należy pamiętać

- testowanie to „szukanie dziury w całym”
- dane testowe obejmują nie tylko dane poprawne, ale też wartości graniczne oraz dane niepoprawne potencjalnie niebezpieczne

Wyczerpujące (obejmujące wszystkie kombinacje danych wejściowych) przetestowanie niebanalnego systemu jest w praktyce zadaniem niewykonalnym!

Testujemy wybrane **przypadki testowe** (ang. *test cases*)



Przypadki testowe

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Przypadek testowy

Zestaw danych wejściowych dla systemu, instrukcji lub kroków oraz wyników oczekiwanych od systemu na podstawie tych danych

Typowy format

- nazwa, id - do identyfikacji
- zwięzły opis celu
- priorytet
- warunki wstępne
- dane wejściowe
- kroki testowe
- oczekiwane wyniki
- warunki końcowe



Standard dokumentowania testów

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

IEEE 829-2008 → ISO/IEC/IEEE 29119 (2013-2022)

ISO/IEC/IEEE 29119

- Część 1 (general concept) - definicje, koncepcja testowania oprogramowania i sposoby ich zastosowania
- Część 2 (test process) - ogólny model procesu testowego
- Część 3 (test documentation) - dotyczy dokumentacji testów oprogramowania, zawiera szablony
- Część 4 (test techniques) - definicje w kategoriach: czarnoskrzynkowe, białoskrzynkowe i oparte doświadczeniu (niezależne od części 2)
- Część 5 (keyword-driven testing) - specyfikacje testów (zwykle zautomatyzowanych) oparte na słowach kluczowych

<https://www.iso.org/standard/81291.html>



Kiedy rozpocząć testowanie?

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

- testowanie pozwala wykryć usterki we wczesnych stadiach rozwoju oprogramowania
- usterki często pojawiają się już na etapie projektowania
- im później zostanie wykryta usterka, tym bardziej rośnie koszt wytworzenia oprogramowania

Kluczowe momenty

- 1 Po zakończeniu fazy projektowania (przypadki testowe wynikające ze specyfikacji)
- 2 Początek implementacji (testy jednostkowe)
- 3 Po zakończeniu kodowania (testy integracyjne)
- 4 Przed wdrożeniem (testy akceptacyjne, użytkownika)

Testowanie jest cykliczne



Poziomy testowania

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

- **Testowanie jednostkowe** (ang. *unit testing*)
 - ukierunkowane na wyszukiwanie usterek w poszczególnych komponentach systemu



Poziomy testowania

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

- **Testowanie jednostkowe** (ang. *unit testing*)
 - ukierunkowane na wyszukiwanie usterek w poszczególnych komponentach systemu
- **Testowanie integracyjne** (ang. *integration testing*)
 - ukierunkowane na poszukiwanie usterek w interfejsach oraz interakcjach pomiędzy zintegrowanymi komponentami systemu
 - coraz większe grupy komponentów są integrowane i testowane pod kątem wzajemnej komunikacji



Poziomy testowania

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

■ Testowanie systemu (ang. *system testing*)

- testowanie działania kompletnie zintegrowanego systemu
- testowanie systemu z perspektywy użytkownika końcowego
- celem jest wykrycie usterek w kontekście wymagań ze specyfikacji oraz celów projektowych

(funkcjonalność, użyteczność, wydajność, stabilność, odporność na błędy, bezpieczeństwo)



Poziomy testowania

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

■ Testowanie systemu (ang. *system testing*)

- testowanie działania kompletnie zintegrowanego systemu
- testowanie systemu z perspektywy użytkownika końcowego
- celem jest wykrycie usterek w kontekście wymagań ze specyfikacji oraz celów projektowych

(funkcjonalność, użyteczność, wydajność, stabilność, odporność na błędy, bezpieczeństwo)

■ Testowanie akceptacyjne (ang. *user acceptance testing*)

- wykonywane po stronie klienta
- weryfikacja zgodności systemu z umową

(ocena przez użytkowników końcowych w rzeczywistych warunkach przed wdrożeniem)



Poprawki

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Naturalną konsekwencją znalezienia usterek jest ich poprawianie, poprzez dokonywanie **poprawek**

Poprawki mogą mieć różną skalę

- modyfikacja pojedynczej klasy
- przeprojektowanie całego podsystemu



Testy regresyjne

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

Prawa Murphy'ego

- *„Patch to poprawka do programu usuwająca stare błędy i dodająca nowe”*
- *„Błędy pojawiają się w jednej części działającego programu, kiedy zmienisz drugą zupełnie z nią niepowiązaną”*



Testy regresyjne

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Prawa Murphy'ego

- *„Patch to poprawka do programu usuwająca stare błędy i dodająca nowe”*
- *„Błędy pojawiają się w jednej części działającego programu, kiedy zmienisz drugą zupełnie z nią niepowiązaną”*

Testy regresyjne

- ponowne testowanie fragmentu programu (po modyfikacji)
- sprawdzają czy poprawka nie skutkuje nowymi usterkami
- jako czynność powtarzalna są dość uciążliwe



Black-box vs. white-box testing

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Testy czarnoskrzynkowe

- traktują oprogramowanie jak tzw. czarną skrzynkę
- testują funkcjonalność bez wnikania w wewnętrzną strukturę (**perspektywa zewnętrzna**)
- można je zastosować na każdym poziomie testowania

Stosowane metody

- testy równoważności – podział możliwych danych wejściowych na grupy, wybór reprezentantów
- testy przypadków granicznych
- tabele decyzyjne – kombinacje warunków wejściowych i odpowiadające im wyniki wyjściowe
- testowanie oparte na wymaganiach
- testowanie eksploracyjne – bez planu w celu spontanicznego wykrycia błędów



Black-box vs. white-box testing

Testy białoskrzynkowe (strukturalne)

- wnikają szczegółowo w wewnętrzną strukturę i zachowanie komponentu (**perspektywa wewnętrzna**)
- opracowanie przypadków testowych wymaga znajomości systemu i umiejętności programistycznych (sprawdzanie poprawności struktur danych i algorytmów)
- zazwyczaj stosowane na poziomie testowania jednostkowego, choć można je zastosować również przy testowaniu integracyjnym oraz testowaniu systemu (do testowania ścieżek pomiędzy komponentami, podsystemami)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja



Wybrane typy testów systemu

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

- **testy instalacyjne** (ang. *installation testing*)- czy system spełnia wszystkie wymagania w środowisku docelowym (u klienta)
- **testy kompatybilności** (ang. *compatibility testing*) - czy oprogramowanie kompatybilne z innymi aplikacjami, środowiskiem, systemami operacyjnymi
- **testy alfa i beta** (ang. *alpha, beta testing*) - swoiste testy akceptacyjne dla oprogramowania „z półki” (wewnętrzne i zewnętrzne)



Wybrane typy testów systemu

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

- **testy wydajności** (ang. *performance testing*) - czy system jest stabilny i odpowiada pod określonym obciążeniem, może sprawdzać też zużycie konkretnych zasobów, skalowalność,...
- **testy użyteczności** (ang. *useability testing*) - testowanie łatwości użytkowania i zrozumienia (interfejsu) systemu przez jego użytkowników
- **testy bezpieczeństwa** (ang. *security testing*) - zmierzają do wykrycia luk w zabezpieczeniach systemu (testy penetracyjne, analiza podatności, testy społecznej inżynierii)



Testowanie w cyklu życia oprogramowania

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

- 1 W klasycznym modelu **kaskadowym** testowanie wydzielone jako osobna faza - wykonane przez niezależną grupę testerów (przed oddaniem produktu klientowi)
- 2 **Metodyki zwinne** uwzględniają testowanie jako jedną z najważniejszych faz cyklu życia oprogramowania, testerzy - kluczowi członkowie drużyny. Testowanie jest integralną częścią procesu wytwarzania oprogramowania, które odbywa się równolegle z procesem rozwoju produktu



Testowanie w metodykach zwinnych

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

- użytkownik jest częścią zespołu przygotowującego oprogramowanie (odpowiada na pytania, rozprasza wątpliwości...)
- testerzy stanowią pomost pomiędzy użytkownikiem a programistami
- podstawę stanowią:
 - **programowanie sterowane testami** (ang. *Test Driven Development - TDD*)
 - **podejście przyrostowe (iteracyjne)**: częsta integracja, nowa funkcjonalność akceptowana po pełnych testach regresji, odbiorca akceptuje system po akceptacji wszystkich przyrostów



Model przyrostowy (iteracyjny)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

- programowanie i testowanie zaczyna się wcześniej (w przeciwieństwie do modelu kaskadowego przed zakończeniem zbierania i definiowania wymagań)
- programowanie i testowanie przebiega w krótkich powtarzających się cyklach – projekt dzieli się na pewną liczbę mini-projektów np. 3 tygodniowych nazywanych **iteracjami** (sprinty w Scrum)
- wynikiem każdej iteracji jest przetestowany, zintegrowany i uruchamialny system częściowy
- duże znaczenie mają wyniki poprzednich kroków oraz adaptacja do zmieniających się wymagań



Agile testing quadrants (Brian Marick)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Testy stawiające czoło biznesowi

Testy wspierające drużynę

- Testy funkcjonalne
- Testy użyteczności

Q4

- Testy akceptacyjne
- Testy alfa/beta

Q3

Q1

- Testy jednostkowe
- Testy komponentów

Q2

- Testy wydajnościowe
- Testy bezpieczeństwa

Testy krytykujące produkt

Testy stawiające czoło technologii



Cele testów

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

- 1 Testy wspierające drużynę** (Q1+Q4, funkcjonalne, użyteczności, jednostkowe, komponentów) - pomagają drużynie nauczyć się jak rozwijać to, co chce klient
- 2 Testy stawiające czoło biznesowi** (Q4+Q3, funkcjonalne, użyteczności, akceptacyjne, alfa/beta) - wspierają rozwój systemu poprzez feedback od użytkowników
- 3 Testy krytykujące produkt** (Q3+Q2, akceptacyjne, alfa/beta, wydajnościowe, bezpieczeństwa) - pomagają odkryć ulepszenia do implementacji
- 4 Testy stawiające czoło technologii** (Q2+Q1, wydajnościowe, bezpieczeństwa, jednostkowe, komponentów) - pomagają drużynie wybrać najodpowiedniejszy projekt



Różna perspektywa

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

Testy wspierające drużynę

(funkcjonalne, użyteczności, jednostkowe, komponentów)

- czy decyzje dotyczące modelu i architektury słuszne?
- czy system zachowuje się poprawnie?
- czy takiego interfejsu użytkownika oczekuje użytkownik?

Testy stawiające czoło biznesowi

(funkcjonalne, użyteczności, akceptacyjne, alfa/beta)

- czy to co powiedział nam klient stanowi jego rzeczywistą potrzebę?
- czy implementujemy to, czego potrzebuje klient?
- czy dostarczamy wartość biznesową wymaganą przez klienta?



Różna perspektywa

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Testy krytykujące produkt

(akceptacyjne, alfa/beta, wydajnościowe, bezpieczeństwa)

- czy można dokonać jakiś ulepszeń produktu?
- czy produkt jest lepszy niż konkurencji?

Testy stawiające czoło technologii

(wydajnościowe, bezpieczeństwa, jednostkowe, komponentów)

- czy system jest wystarczająco szybki i bezpieczny?
- czy system jest rozszerzalny?
- czy kod jest testowalny i napisany uniwersalnie (ang. *reusable*)



Testowanie w metodykach zwinnych

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Własności

- nie ma z góry ustalonej kolejności testów
 - Q1, Q2, Q3, Q4 formalnie nie odnosi się do kolejności, można np. zacząć od testów wydajnościowych, żeby sprawdzić szybko pewne rozwiązanie
- po przeprowadzonych testach można wprowadzić nowe wymagania (odkryć brakujące)
- rezultaty testów skutkują ulepszeniami architektury
- intensywne testowanie podczas cyklu wytwarzania oprogramowania daje lepsze projekty



Programowanie sterowane testami

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Czym jest TDD?

- jest to element tzw. programowania ekstremalnego (XP od ang. *eXtreme Programming*)
- wpada w ćwiartkę Q1 (testy jednostkowe, testy komponentów)
- testy – sformalizowany sposób zapisu wymagań funkcjonalnych
- najpierw test a później kod, który ma go przejść
- niezaliczenie testu = błędny kod (napraw lub dopisz)

Metodyki zwinne i TDD są komplementarne

- Agile koncentruje się na szybkim i elastycznym rozwoju projektów z bliską współpracą z klientem
- TDD dostarcza narzędzi do tworzenia czystego, dobrze przetestowanego kodu



Testy jednostkowe - testowanie w izolacji

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testów
Ciągła integracja

Testy jednostkowe

Def.: pozwalają na testowanie małych fragmentów kodu w izolacji od reszty systemu

Cykl TDD: „Red-Green-Refactor”

1 Red (napisz test)

```
def test_sum():  
    assert sum(2, 3) == 5
```

2 Green (napisz kod)

```
def sum(a, b):  
    return a + b
```

3 Refactor (ulepsz kod)



Testy jednostkowe - testowanie w izolacji

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Testowanie w izolacji nie jest łatwe – często testowany fragment **zależy** od innych (mniej lub bardziej złożonych, czasem niegotowych) komponentów zewnętrznych

Ratunek - dublerzy testowi (ang. *test doubles*)

Komponent, od którego zależy nasz test, zamieniany jest przez testowego dublera o tym samym interfejsie

Przykład: testując moduł wyświetlania, zamiast łączyć się z bazą danych przy każdym teście, tworzymy „podróbkę” połączenia



Rodzaje dublerów testowych

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

- 1 Dummy objects** – atrapa przekazana, ale nieużywana (testowana metoda wymaga parametru, ale w teście nie dbamy o wartość)
- 2 Stubs** – symuluje input, komponent od którego zależy test niegotowy a potrzebny pośredni inputu od niego (np. stub z zasymulowanym wynikiem z API)
- 3 Fakes** – uproszczenie rzeczywistego komponentu (np. Fake Database przechowująca dane w pamięci a nie rzeczywistej bazie danych)
- 4 Mocks** – obiekty, które śledzą interakcje, aktywni dublerzy stosowani, gdy komponent niegotowy, ale wywołania jego metod w pewnym ustalonym porządku są niezbędne do zweryfikowania zachowania naszego komponentu (możemy sprawdzić, czy metoda została wywołana i jak zareagowała)
- 5 Spies** – podobne do mocków, ale w odróżnieniu od nich, rzeczywisty kod metody jest wykonywany w spy i dodatkowo spy zbiera informacje o wywołaniach dla późniejszej weryfikacji (jakie metody, z jakimi argumentami, ile razy)



Dublerzy testowi

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Mockito (Java Framework) – skupia się głównie na tworzeniu mocków i weryfikacji zachowań, pozwala

- tworzyć obiekty zastępcze (uproszczone), symulujące działanie rzeczywistych
- ustawić, jakie wartości mają być zwracane przez metody Mocka
- ustawić, aby metoda mocka rzucała wyjątek
- rozwijać testy z użyciem Mocków, które zapamiętują wywołania swoich metod i mogą weryfikować różne warunki (np. kolejność wywołania procedur z konkretnymi argumentami)
- tworzyć Stubs (niejawnie, jako część mockowania) poprzez metody `when(..).thenReturn(..)` oraz `doReturn(..).when(..)`, które przy wywołaniu zwracają z góry ustalone wartości
- monitorować rzeczywiste obiekty, które wykonują swoje prawdziwe zadania (Spies)



Mockito – przykład

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

Klasa produkcyjna

```
public class UserService {  
    private UserRepository userRepository;  
  
    public UserService(UserRepository userRepository)  
    {  
        this.userRepository = userRepository;  
    }  
  
    public String getUsername(int userId) {  
        return userRepository.findNameById(userId);  
    }  
}
```

Aby ją przetestować potrzebujemy obiekt `UserRepository`



Mockito – przykład (test)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi
Ciągła integracja

```
public class UserServiceTest {
    @Test
    public void testGetUserName() {
        // Tworzymy mock obiektu UserRepository
        UserRepository mockRep = Mockito.mock(
            UserRepository.class);
        // Definiujemy zachowanie mocka
        when(mockRep.findNameById(1)).thenReturn("Jan K");
        // Tworzymy obiekt UserService, wstrzykując mocka
        UserService userService = new UserService(mockRep);
        // Wywołujemy testowaną metodę
        String userName = userService.getUserName(1);
        // Sprawdzamy, czy metoda działa poprawnie
        assert userName.equals("Jan K");
        // Czy metoda wywołana z argumentem 1?
        verify(mockRep).findNameById(1);
    }
}
```



Ciągła integracja (ang. *Continuous Integration*)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

Czym jest CI?

- praktyka wytwarzania oprogramowania, która polega na regularnym/częstym łączeniu kodu źródłowego
- podobnie jak TDD jest to element programowania ekstremalnego
- wymaga zapewnienia poprawności kompilacji kodu po wykonaniu integracji



Ciągła integracja (ang. *Continuous Integration*)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

Nie tylko budowa i kompilacja źródeł . . .

- nawet kod, który się kompiluje może zawierać dowolną ilość poważnych błędów – trzeba go przetestować
- testy zgłaszają błędy. . .
- czasem skompilowanie kodów wymaga ręcznego odpalania wielu skryptów i długiego oczekiwania
- czasem budowa źródeł musi odbywać się w specjalnych warunkach

Główny cel CI

Integracja zmian wprowadzanych przez programistów w sposób **ciągły i automatyczny** - szybkie wykrywanie ewentualnych konfliktów, błędów czy regresji w kodzie



Ciągła integracja (ang. *Continuous Integration*)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

Jak to działa?

- programista pracuje nad kodem lokalnie, zmiany przesyła do repozytorium
- system CI wykrywa zmiany w repozytorium
- rozpoczyna się proces kompilacji kodu
- system uruchamia zestaw testów (np. jednostkowe, integracyjne)
- jeżeli kompilacja i testy pomyślnie, to zmiany zaakceptowane i dołączone do głównej gałęzi
- w przypadku błędów system CI generuje raport



Ciągła integracja (ang. *Continuous Integration*)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

Ciągła integracja vs Ciągłe dostarczanie

Ciągła integracja (CI) : Skupia się na automatyzacji budowy i testowania kodu po każdej zmianie

Ciągłe dostarczanie (CD) : Rozszerza CI, automatyzując wdrażanie aplikacji na środowiska testowe lub produkcyjne



Narzędzia wspierające CI

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testów
Ciągła integracja

Jenkins - <https://www.jenkins.io/>

GitLab CI/CD - <https://docs.gitlab.com/ee/ci/>

GitHub Actions - <https://github.com/features/actions>

Travis CI - <https://www.travis-ci.com/>

CircleCI - <https://circleci.com/>

TeamCity - <https://www.jetbrains.com/teamcity/>

Bamboo - <https://www.atlassian.com/pl/software/bamboo>

...



Przykładowy plik konfiguracyjny (GitLab CI/CD)

Anna
Lauks-Dutka

Wstęp

Poziomy
testowania

Kategorie
metod
testowych

Typy testów
systemu

Proces
testowania

Testowanie w
metodykach
zwinnych

Testy jednostkowe -
testowanie w izolacji

Dublerzy testowi

Ciągła integracja

```
stages:  
  - build  
  - test  
  
build-job:  
  stage: build  
  script:  
    - echo "Building the project"  
    - ./gradlew build  
  
unit-tests:  
  stage: test  
  script:  
    - ./gradlew test --tests *UnitTest  
  
integration-tests:  
  stage: test  
  script:  
    - ./gradlew test --tests *IntegrationTest
```