

Zapewnienie jakości oprogramowania

wykład 3

Techniki projektowania testów

dr inż. Luiza Fabisiak



1. Podstawy testowania
2. Testowanie w cyklu życia oprogramowania
- 3. Techniki projektowania testów**
4. Zarządzanie testowaniem
5. Testowanie wspierane narzędziami

Zadanie 3



Optymalizacja zarządzaniem zmianami i śledzenia błędów - 30 minut – 30 min

- Przez 15 minut, pracujemy w grupach, wypracowujemy – każda grupa własny-projekt wymarzonego systemu do zarządzania zgłoszeniami błędów i innych zmian,
- Stopień, w jaki uda nam się zbliżyć do ideału, ograniczony jest niestety bezwzględnością narzuconych ram czasowych.
- Ostatecznie uzgodnioną wersję projektu zapisujemy – w formie uproszczonego schematu przepływu stanów na papierze
- Przez kolejne 15 minut każda grupa prezentuje swoją pracę.



-
1. Przeglądy i proces testowy
 2. Proces przeglądu
 3. Analiza statyczna

3.1 Przeglądy i proces testowy



Podstawowe definicje:

- Testowanie statyczne:

Testowanie TP (testowany produkt) :

- Techniki ręczne (przeglądy);
- Techniki zautomatyzowane (analiza statyczna)

Najczęściej spotykaną formą testowania statycznego jest **przegląd** (review)

3.1 Przeglądy i proces testowy



- Przeglądy są sposobem testowania produktów pracy projektowej (także kodu).
- Mogą być wykonywane przed stworzeniem oprogramowania.
- Defekty wykryte podczas przeglądów na wczesnym etapie cyklu życia oprogramowania są dużo tańsze do usunięcia niż defekty wykryte podczas wykonywania późniejszych testów:

Zwłaszcza defekty znalezione w wymaganiach

3.1 Przeglądy i proces testowy



- Przegląd – to czynność „**ręczna**”
 - ale istnieje wsparcie narzędziowe
- Można przejrzeć **każdy** produkt pracy projektowej:
 - Specyfikacja wymagań,
 - Specyfikacja projektu,
 - Kod,
 - Plan testów,
 - Specyfikacja testów,
 - Przypadki testowe,
 - Skrypty testowe,
 - Instrukcja użytkownika,
 - Strony webowe,
 -

3.1 Przeglądy i proces testowy



ZALETY PRZEGLĄDÓW

- Wczesne wykrycie i naprawa defektu;
- Poprawa produktywności oprogramowania;
- Skrócony czas tworzenia oprogramowania;
- Skrócony czas i koszt testowania;
- Zmniejszenie liczby defektów;
- Lepsza komunikacja;
- Wykrycie niejasności i podpuszczeń (np. w wymaganiach)

3.1 Przeglądy i proces testowy



- Przeglądy, analiza statyczna i testowanie dynamiczne uzupełniają się
- Przeglądy znajdują defekty, nie awarie.
- Typowe defekty znajdowane podczas przeglądów:
 - odchylenie od standardów;
 - błędy i nieścisłości w wymaganiach;
 - błędy projektu;
 - niewystarczająca zdolność do konsekwencji (insufficient maintainability)
 - nieprawidłowe specyfikacje interfejsów

3.1 Przeglądy i proces testowy



- Przeglądy, analiza statyczna i testowanie dynamiczne

„Do typowych usterek, które łatwiej wykryć w testach statycznych niż dynamicznych należą:

- Odchylenie od standardów,
 - Usterki w wymaganiach,
 - Usterki w projekcie,
 - Niedostateczna pielęgnowalność oraz nieprawidłowe specyfikacje interfejsów
- niewystarczająca maintainability)
 - nieprawidłowe specyfikacje interfejsów



-
1. Przeglądy i proces testowy
 2. Proces przeglądu
 3. Analiza statyczna

3.2 Proces przeglądu



Przegląd – Proces lub spotkanie, w czasie którego produkt lub kilka produktów, jest przedstawiany zespołowi wytwórczemu, kierownictwu, użytkownikom lub innym interesariuszom w celu uzyskania uwag lub akceptacji

- Przeglądy słabo ustrukturalizowane
- Przeglądy ustrukturalizowane

3.2 Proces przeglądu



Przeglądy słabo ustrukturalizowane	Przeglądy ustrukturalizowane
<ul style="list-style-type: none">• Brak jednoznacznego określonego procesu przeglądu• Uczestnicy nie przygotowują się na spotkania• Brak podziału uczestników na role• Wyniki przeglądu nie są dokumentowane• Wynik prac po przeglądzie nie jest kontrolowany	<ul style="list-style-type: none">• Czynności przeglądowe uwzględnione są w harmonogramie projektu, przegląd przebiega wg. Zdefiniowanego procesu• Uczestnicy wcześniej dostają materiały i przygotowują się do spotkania• Występuje podział na role• Zarówno przebieg, jak i wynik przeglądu, jest dokumentowany• Po przeglądzie następuje kontrola prac, które po nim zostały wykonane.

3.2 Proces przeglądu



Zasady przeprowadzania przeglądów:

- **Integracja** przeglądów z całym procesem wytwarzania.
- **Zdefiniowany** przebieg przeglądu
- **Wczesne** planowanie i zagwarantowanie potrzebnych zasobów.
- Procesowi przeglądu może podlegać **każdy** dokument
- Przeprowadzenie przeglądów w **początkowych** fazach projektu
- Cykliczne przeprowadzenie przeglądów;
- Wyznaczenie warunków wstępnych i końcowych;
- **Przemyślany dobór** uczestników przeglądu
- Zapewnienie **przyjaznej** atmosfery podczas spotkania;
- Korekta nieprawidłowości następująca **bezpośrednio** po przeglądzie;
- Zakończenie projektu **przeglądem powdrożeniowym**

3.2 Proces przeglądu



Główne fazy formalnego przeglądu:

PLANOWANIE: wybór osób, przypisanie ról, określenie kryteriów wejściowych i warunków wyjścia (dla bardziej formalnych typów przeglądu np. inspekcji); wybór fragmentów dokumentu do przejrzania.

ROZPOCZĘCIE (kick-off): dystrybucja dokumentów; wyjaśnienie uczestnikom celów procesu i dokumentów: sprawdzenie kryteriów wejściowych (dla bardziej formalnych typów przeglądów).

INDYWIDUALNE PRZYGOTOWANIE: praca wykonywana samodzielnie przez każdego z uczestników przed spotkaniem przeglądownym, przejrzanie dokumentów, znotowanie potencjalnych błędów pytań i komentarzy.

3.2 Proces przeglądu



Główne fazy formalnego przeglądu:

SPOTKANIE PRZEGLĄDOWE (kontrola/ocena/zapisanie wyników) dyskusja lub zapis z udokumentowanymi wynikami lub protokołami (dla bardziej formalnych typów przeglądu). Uczestnicy spotkania mogą po prostu zanotować defekty, przedstawić zlecenia odnośnie obsługi defektów lub poczynić decyzje w sprawie defektów.

OBRÓBKA (poprawki) (rework): naprawianie znalezionych defektów- zazwyczaj wykonywane przez autora

ZAKOŃCZENIE (dalsza część/sprawdzenie) (follow-up) sprawdzenie, czyli defekty zostały zaadresowane/obsłużone (czy przypisano im osobę odpowiedzialną za naprawę); zebranie metryk i sprawdzenie warunków wyjścia (dla bardziej formalnych typów przeglądów)

3.2 Proces przeglądu



Role i odpowiedzialność:

Typowy formalny przegląd może zawierać następujące role:

KIEROWNIK: decyduje o wykonywaniu przeglądów, przydziela czas w harmonogramie projektów i określa, czy zostały uwzględnione cele przeglądu;

MODERATOR: osoba, która prowadzi przegląd dokumentu lub zestawu dokumentów, wliczając w to planowanie przeglądu, przebieg spotkania i dalszą część po spotkaniu. Jeżeli konieczne, moderator może prowadzić mediacje pomiędzy różnymi punktami widzenia i często jest osobą na której spoczywają sukcesy przeglądu.

AUTOR: autor lub osoba odpowiedzialna za przeglądany dokument lub dokumenty.

PRZEGLĄDAJĄCY: osoba ze specyficznym technicznym lub biznesowym przygotowaniem (zwane też kontrolerami lub inspektorami), które po niezbędnym przygotowaniu, identyfikują i opisują wyniki badań (np. błędy) przeglądanego produktu. Przeglądający powinni być wybrani w taki sposób, aby reprezentować różne perspektywy i role. Uczestniczą we wszelkich spotkaniach przeglądowych.

PROTOKÓLANT (lub skryba): dokumentuje wszystkie kwestie, problemy i otwarte punkty, które zidentyfikowano podczas spotkania.

3.2 Proces przeglądu



Podstawowe rodzaje przeglądów:

- **Przegląd nieformalny** (przegląd ad hoc)(informal review)
- **Przejrzenie** (przejście)(walkthrough)
- **Przegląd techniczny** (technical review)
- **Inspekcja** (inspection)

Nieformalny



Formalny

3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

Przegląd nieformalny:

- Brak formalnego procesu;
- Może być zastosowany przy programowaniu parami lub do przeglądów projektu i kodu przez kierownika zespołu;
- Opcjonalnie może być udokumentowany;
- Może różnić się w przydatności w zależności od przeglądającego;
- Główny cel: niedrogi sposób uzyskania niewielkiej korzyści

3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

Przejrzenie:

- Spotkanie prowadzone przez autora;
- Grupa koleżeńska, otwarte dyskusje;
- Przegląd dot. Wymagań, kodu, scenariuszy, projektów;
- Opcjonalne: przygotowanie przeglądających przed spotkaniem, raport z przeglądu;
- Lista wykrytych rzeczy i protokolant, który nie jest autorem;
- Może w praktyce być zróżnicowane od zupełnie nieformalnego do bardzo formalnego;
- Zalecane dla dokumentów mniejszej wagi;
- Zalecenie dla małych grup projektowych (dla 5 osób)

3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

Przegląd techniczny:

- Zdefiniowany proces wykrywania defektów m.in. Przez kolegów i ekspertów technicznych; opcjonalnie z udziałem kierownictwa;
- W idealnej sytuacji poprowadzony przez przeszkolonego moderatora (nie autora)
- Wstępne przygotowanie przed spotkaniem;
- Opcjonalnie: użycie list kontrolnych;
- Tworzony raport przeglądu, listy uwag oraz rekomendacji, jeśli konieczne;
- W praktyce może być zróżnicowany od zupełnie nieformalnego do bardzo formalnego;
- Główne cele: przedyskutować, podjąć decyzje, ocenić alternatywy, znaleźć defekty, rozwiązać problemy techniczne oraz sprawdzić zgodność ze specyfikacjami i standardami.

3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

Przegląd techniczny:

- Zdefiniowany proces wykrywania defektów m.in. Przez kolegów i ekspertów technicznych; opcjonalnie z udziałem kierownictwa;
- W idealnej sytuacji poprowadzony przez przeszkolonego moderatora (nie autora)
- Wstępne przygotowanie przed spotkaniem;
- Opcjonalnie: użycie list kontrolnych;
- Tworzony raport przeglądu, listy uwag oraz rekomendacji, jeśli konieczne;
- W praktyce może być zróżnicowany od zupełnie nieformalnego do bardzo formalnego;
- Główne cele: przedyskutować, podjąć decyzje, ocenić alternatywy, znaleźć defekty, rozwiązać problemy techniczne oraz sprawdzić zgodność ze specyfikacjami i standardami.

3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

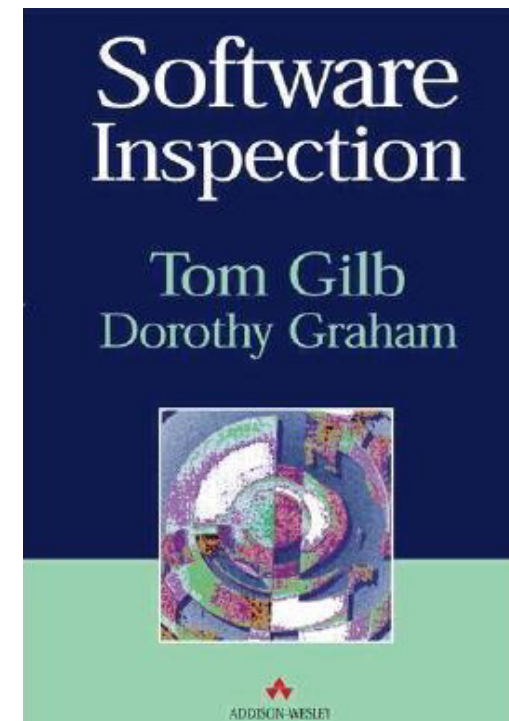
Inspekcja:

- *Inspekcja:*

Sformalizowany proces poprawy jakości polegającej na grupowym przykładzie materiałów pisemnych. Posiada on dwa aspekty: **ulepszania produktu** oraz **ulepszania procesów** produkcji i kontroli jakości.

Stosowana jest:

Dla bardzo ważnej dokumentacji oraz w sytuacjach, gdy zachodzi konieczność udowodnienia sposobu przeprowadzania i wyniku przeglądu



3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

Inspekcja:

- przeprowadzona przez wyszkolonego moderatora (który nie jest autorem);
- Wszystkie działania są dokładnie zaplanowane, podporządkowane harmonogramowi;
- Uczestnicy mają określone role;
- Zawiera miary (metryki);
- Formalny proces oparty jest na regułach oraz listach kontrolnych z kryteriami wejściowymi i warunkami wyjścia;
- Wstępne przygotowanie przed spotkaniem;
- Wyniki raportowane Tworzona jest lista wykrytych problemów;
- Formalny proces kontroli po wykonaniu napraw;
- Cel: znalezienie defektów. Poprawa jakości produktu, procesu i poprawa samej kontroli jakości.

3.2 Proces przeglądu



Rodzaje przeglądów i ich cechy:

Inspekcja:

- przeprowadzona przez wyszkolonego moderatora (który nie jest autorem);
- Wszystkie działania są dokładnie zaplanowane, podporządkowane harmonogramowi;
- Uczestnicy mają określone role;
- Zawiera miary (metryki);
- Formalny proces oparty jest na regułach oraz listach kontrolnych z kryteriami wejściowymi i warunkami wyjścia;
- Wstępne przygotowanie przed spotkaniem;
- Wyniki raportowane Tworzona jest lista wykrytych problemów;
- Formalny proces kontroli po wykonaniu napraw;
- Cel: znalezienie defektów. Poprawa jakości produktu, procesu i poprawa samej kontroli jakości.

3.2 Proces przeglądu



TECHNIKA	Używany do	Obejmuje	Prowadzący	Formalizacja
Przegląd nieformalny	Tanie zidentyfikowanie usterek	Koledzy	-	Nieformalnie
Przejrzenie	Edukacja	Grupa koleżeńska	Autor	Formalne/ nieformalne
Przegląd techniczny	Znajdowanie usterek technicznych	Grupa koleżeńska, eksperci techniczni	Moderator	Formalne/ nieformalne
Inspekcja	Znajdowanie usterek	Zdefiniowane role	Moderator	Formalne



-
1. Przeglądy i proces testowy
 2. Proces przeglądu
 3. Analiza statyczna

3.3 Analiza statyczna



- **Cel:** wykrycie defektów w kodzie źródłowym oprogramowania lub w modelach oprogramowania
- Przeprowadzana jest bez wykonywania TP
- Wykrywa defekty nie awarie
- Narzędzia analizy statycznej są w stanie analizować kod programu, jak również generowanie przez niego dane, np. w HTML, XML itp

3.3 Analiza statyczna



Zalety analizy statycznej:

- Wczesne wykrycie błędów – przed wykonaniem testów dynamicznych;
- Dzięki obliczeniu miar wczesne ostrzeganie o podejrzanych aspektach kodu lub projektu;
- Identyfikacja defektów, które nie dają się łatwo wykryć podczas testowania dynamicznego;
- Wykrycie zależności i spójności w modelach oprogramowania takich jak linki;
- Lepsza pielęgnowalność kodu i projektu;
- Zapobieganie błędom w dalszym procesie wytwórczym w oparciu o informacje z analizy statycznej;

3.3 Analiza statyczna



Typowe defekty wykrywane przez analizę statyczną:

- Odwołanie się do niezainicjalizowanej zmiennej;
- Niespójny interfejs pomiędzy modułami;
- Zmienne niewykorzystywane;
- Nieosiągalny (martwy);
- Naruszenie standardów programowania;
- Słabe punkty zabezpieczeń;
- Błędy składni kodu i modeli oprogramowania;
- Brakująca albo błędna logika (pętle potencjalnie nieskończone);
- Zbyt skomplikowane konstrukcje.

3.3 Analiza statyczna



Narzędzia do analizy statycznej:

- Używane są zwykle przez:
 - **Programistów**
Testy modułowe;
Integracyjne.
 - **Projektantów:**
Modelowanie oprogramowania.
 - Mogą wytworzyć dużą liczbę komunikatów ostrzegawczych;
 - Wymaga dobrego zarządzania;



- Identyfikacja warunków testowych i projektowanie przypadków testowych
- Kategorie technik projektowania testów
- Techniki testowania w oparciu o specyfikacje czyli czarnoskrzynkowe
- Techniki testowania w oparciu o strukturę lub białoskrzynkowe
- Techniki oparte na doświadczeniu
- Wybór technik testowych



Kroki procesu identyfikacji warunków testowych i projektowania testów:

- Projektowanie testów przez identyfikowanie warunków testowych
- Specyfikowanie przypadków testowych
- Specyfikowanie procedur testowych

„Warunek testowy definiuje się jako element lub zdarzenie, które może zostać sprawdzone przez jeden lub więcej przypadków testowych (np. Funkcja, transakcja, atrybut jakościowy lub element struktury)”



Proces projektowania przypadków testowych:

- **Nieformalny** z małą ilością dokumentacji lub bez niej

Albo

- **Bardzo formalny**

w zależności od kontekstu testowania:

- Organizacji;
- Dojrzałości procesów testowania i programowania;
- Ograniczeń czasowych;
- Zasobów ludzkich.



Analiza dokumentacji podstawy testów:

- **CEL:**

Określenie, **co** testować (czyli identyfikacja warunków testowych)



Budowa warunków testowych:

- Identyfikacja elementu lub zdarzenie, które powinno być zweryfikowane przez jeden lub więcej przypadków;
- Śledzenie powiązań pomiędzy warunkami testowymi, a specyfikacją lub wymaganiami
 - Analiza wpływu w momencie zmiany wymagań
 - Analiza pokrycia wymagań określonym zestawem testów.

Podczas projektowania testów szczegółowe podejście testowe jest budowane na podstawie zidentyfikowanego **ryzyka**



Specyfikacja przypadków testowych:

- Następuje budowa i szczegółowe opisanie
 - Danych wejściowych;
 - Przypadków testowych (akcji do wykonania);
 - Wyników oczekiwanych, które:
 - Powinny zawierać:
 - Wartości wyjścia
 - Zmiany danych i stanów;
 - Inne wyniki wykonania testów
 - Powinny być zawsze określone przed wykonaniem testów
- Przypadek testowy (test case) składa się z zestawu wartości wejściowych, warunków początkowych, oczekiwanych wyników i wyników końcowych, zaprojektowanych w celu pokrycia pewnych celów lub warunków testowych (IEEE 829)



Procedura testowa:

- Przygotowane przypadki testowe ustawione są w kolejności wykonywania;
- Dla testów automatycznych kolejność działań określana jest w skrypcie testowych (zautomatyzowanej procedurze testowej).

Harmonogram wykonania testu:

- Kolejność wykonania: procedur testowych; zautomatyzowanych skryptów testowych;
- Terminarz;
- Osoba odpowiedzialna;
- Priorytet
- Zależności techniczne i logiczne i inne – konieczność wykonania testów regresyjnych

4.2 Kategoria technik projektowania testów



Podstawowe definicje:

Testy czarnoskrzynkowe (funkcjonalne): Technika projektowania przypadków testowych, w której przypadki te są projektowane bez zaglądania w wewnętrzne mechanizmy działania modułu:

- Przypadki testowe na podstawie analizy dokumentacji jako podstawowy testów zarówno funkcjonalnej jak i нефunkcjonalnej bez odwołania się do jego wewnętrznej struktury;
- Przypadki oparte na doświadczenia

Testy białoskrzynkowe (strukturalne, oparte na strukturze): Technika projektowania przypadków testowych w oparciu o analizę struktury modułu



Podstawowe cechy technik opartych na specyfikacji:

- Modele formalne lub nieformalne są wykorzystywane do specyfikowania rozwiązywanych problemów, systemu lub modułów;
- Przypadki testowe mogą być wytwarzane w sposób usystematyzowany na podstawie modeli

4.2 Kategoria technik projektowania testów



Podstawowe cechy technik opartych na doświadczeniu:

Do tworzenia przypadków testowych wykorzystuje się:

☐ Wiedzę:

- Na temat TP;
- O wykorzystaniu TP;
- Środowisku produkcyjnym;
- O możliwych defektach i ich występowaniu

☐ Doświadczenie:

- Testerów
- Programistów
- Użytkowników

Wykład opracowano na podstawie:



1. L. Rosenfeld, P. Morville, *Architektura informacji w serwisach internetowych*, Helion, Gliwice 2003.
2. J. Kalbach, *Projektowanie nawigacji strony WWW. Optymalizacja funkcjonalności witryny*, Helion, Gliwice 2008.
3. M. Pear, *Funkcjonalność serwisów internetowych*, PWN 2013.
4. L. Flordi, *The Blackwell Guide to the Philosophy of Computing and Information*, Blackwell 2002.
5. K. Visocky O'Grady, J. Visocky O'Grady, *The Information Design Handbook*, F+W Media 2008.
6. <http://www.merixstudio.pl/blog/7-zlotych-regul-testowania-stron-i-aplikacji-internetowych/> (dostęp 9.10.16)
7. <http://www.istqb.org/downloads/glossary-current.pdf>, słownik terminów, ISTQB Standard Glossary of Terms used in Software Testing V.2.0, Dec, 2nd 2007 (dostęp 1.10.15)
8. http://www.dmoz.org/Computers/Programming/Software_Testing/Products_and_Tools/, katalog narzędzi i produktów związanych z testowaniem oprogramowania (dostęp 1.10.15)
9. <http://www.nist.gov/director/prog-ofc/report02-3.pdf>, The Economic Impacts of Inadequate Infrastructure for Software Testing, NIST (dostęp 1.10.15)
10. <http://sunnyday.mit.edu/papers/therac.pdf>, artykuł o Therac 25 (błędy oprogramowania spowodowały śmierć pacjentów), Medical Devices: The Therac 25 – Nancy Leveson (dostęp 1.10.15)
11. <http://www.bcs.org/iseb>, <http://www.istqb.org>, strony organizacji certyfikujących w zakresie kompetencji związanych z testowaniem oprogramowania. (dostęp 1.10.15)

Koniec

