

Model oceny jakości implementacji wzorców projektowych

mgr inż. Rafał Wojszczyk

Promotor: prof. dr hab. inż. Volodymyr Khadzhynov

Promotor pomocniczy: dr inż. Piotr Ratuszniak

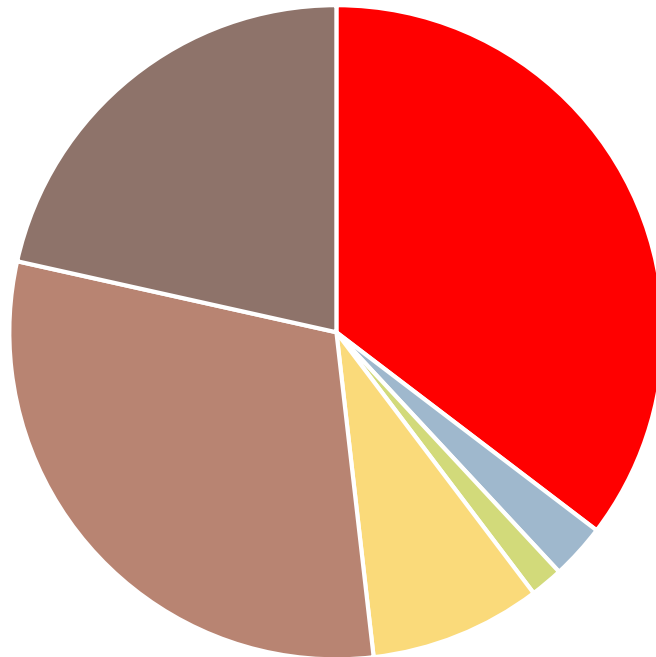
Politechnika Koszalińska

Agenda

1. Geneza i problem badawczy
2. Metoda oceny jakości implementacji wzorców projektowych
3. Przykład użycia w/w metody
4. Badania weryfikujące i podsumowanie

Geneza – zespoły wytwórcze

Popularność



■ Agile

■ RUP

■ Spiralny

■ Kaskadowy

■ Brak

■ Pozostałe

- ▶ małe zespoły,
- ▶ znikoma ilość dokumentacji projektowej (lub brak).
- ▶ Wytwarzają:
 - ▶ autorskie oprogramowanie bazodanowe w nowoczesnym, obiektowym języku programowania,
 - ▶ stale rozwijane, aby zapewnić konkurencyjność.
- ▶ Jak obniżyć koszty konserwacji i rozwoju oprogramowania?

Geneza – wzorce projektowe

- ▶ Szablony rozwiązań dla powtarzających się problemów w programowaniu obiektowym.
- ▶ Wymaga zrozumienia i własnego wkładu od osoby implementującej.

Korzyści:	Trudności:
• unikanie wybranych błędów, • łatwiejsze poprawianie błędów, • większa czytelność kodu, • tańsza rozbudowa.	• brak „wzorcowych” wzorców, • brak automatycznej implementacji, • różnorodność implementacji, • zbyt „luźne” podejście w zwinnych zespołach, • nie testuje się wzorców, • pozorna implementacja.

Geneza – przykład Singleton

► Cel: jeden egzemplarz klasy

Jakość implementacji w kryterium rozbudowy i integracji

Poziom jakości	1 (niski)	2 (średni)	3 (wysoki)
Przykład implementacji wzorca	<pre>public class Singleton1 { public Singleton1 instance = new Singleton1(); private Singleton1() { } }</pre>	<pre>public class Singleton2 { static Singleton2 instance = new Singleton2(); private Singleton2() {} }</pre>	<pre>public class Singleton3 { public static Singleton3 instance = new Singleton(); private Singleton3() {} }</pre>
Konsekwencje	Błędy w działaniu aplikacji, wymaga poprawy błędów i ich skutków (np. uszkodzenie danych)	Brak możliwości integracji przez zewnętrzne aplikacje, wymaga ponownej implementacji	Umożliwia integrację przez zewnętrzne aplikacje, łatwiejsze wykorzystanie mechanizmu refleksji

Geneza – poziomy jakości

Oczekiwany poziom jakości

Alternatywne metody:
Spine, Lepus2, D³,
Simplicity, Pinot

4. Występuje wzorzec i zapewnia korzyść

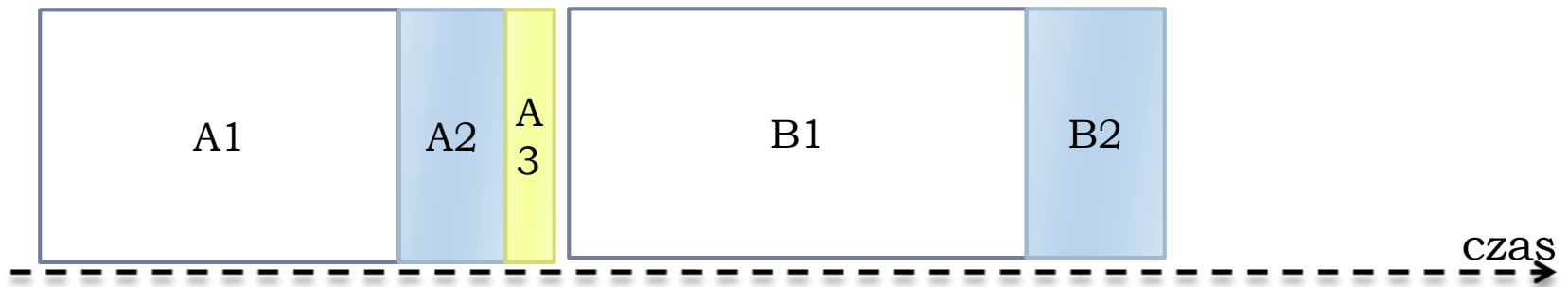
3. Występuje wzorzec i zapewnia korzyści, ale występują drobne usterki

2. Występuje wzorzec (spełnia cel) ale jego zastosowanie nie zapewnia korzyści

1. Brak wzorca, ale fragmenty implementacji przypominają wzorzec

0. Brak wzorca

Geneza – proces wytwórczy



A1 – iteracja z implementacją wzorca,

B1 – rozbudowa oprogramowania,

A2, B2 – czas zarezerwowany na przegląd i ogólną poprawę oprogramowania, tzw. code-review, w tym wykorzystanie metod statycznej analizy oprogramowania,

A3 – ocena jakości implementacji wzorców projektowych i poprawa implementacji wzorców.

Problem badawczy

Zwinny zespół wytwórczy stale rozwija swój produkt programowy (aplikacja bazodanowa napisana w obiektowym języku programowania), w którym implementuje wzorce projektowe, aby obniżyć koszty rozbudowy i konserwacji.

Poszukiwana jest **metoda oceny jakości implementacji wzorców projektowych**, którą można wykorzystać w trakcie tzw. code-review.

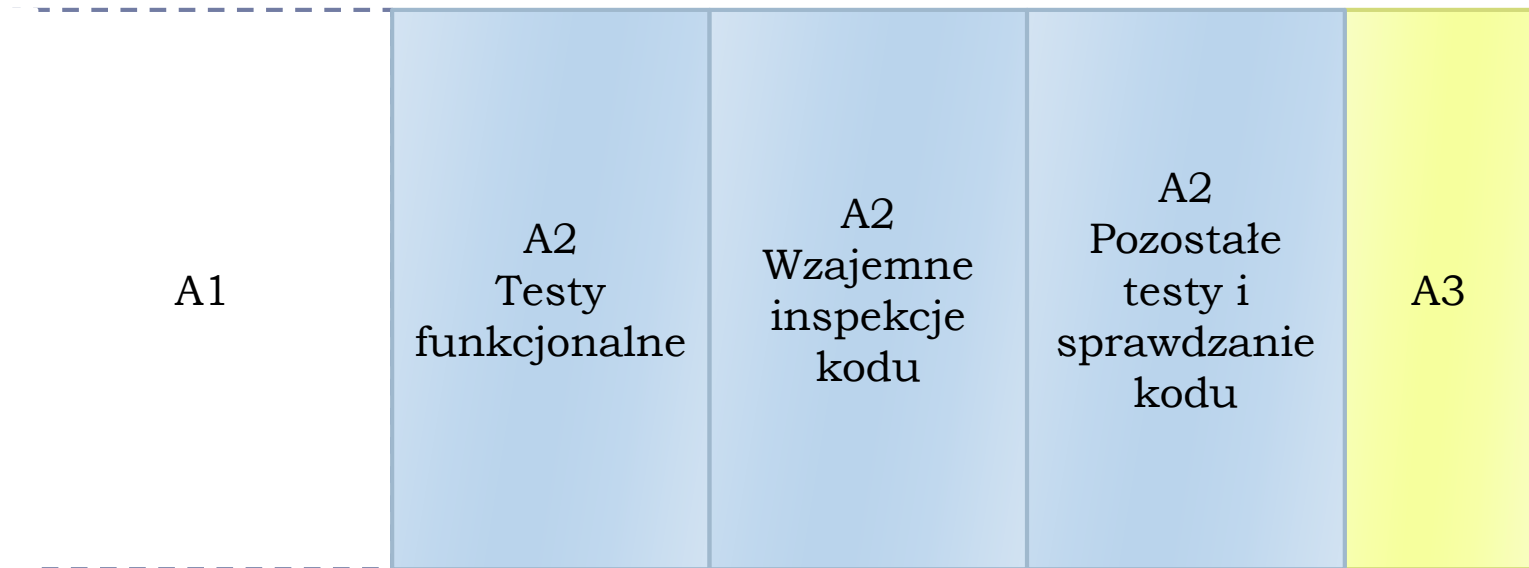


Teza

Struktury danych oparte na założeniach paradygmatu programowania obiektowego oraz wykorzystanie przynajmniej dwóch rodzajów metryk¹, umożliwia budowę metody oceny jakości implementacji wzorców projektowych, która pozwala uzyskiwać wyniki oceny z lepszą dokładnością i nie większym kosztem niż jest to możliwe przy pomocy alternatywnych metod statycznej analizy oprogramowania.

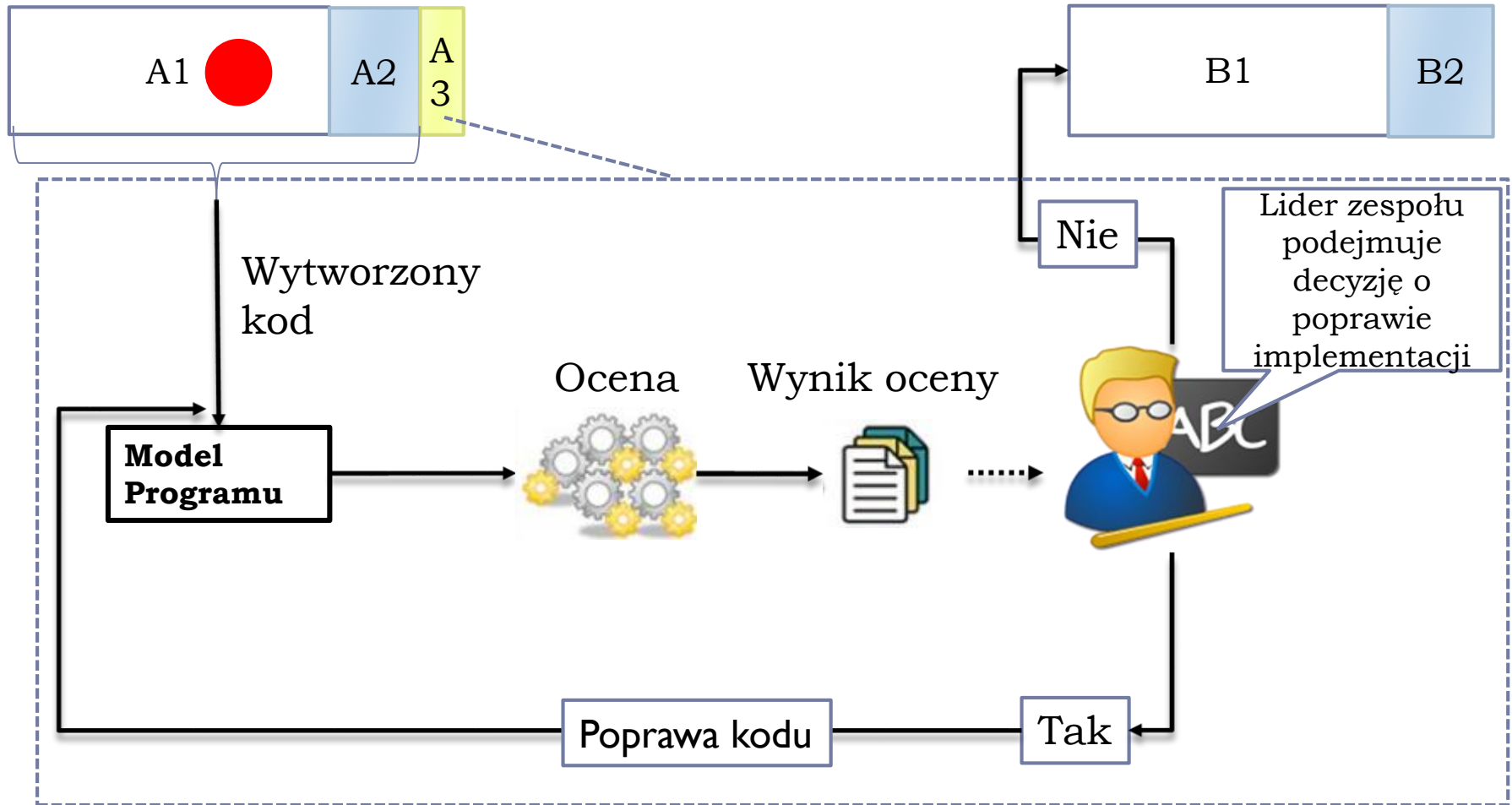
¹ Opisują szablony wzorców

Rozwiązanie – wykorzystanie w procesie wytwórczym



- ▶ Proponowana metoda to uzupełnienie metod statycznej analizy oprogramowania, które są wykorzystywane w trakcie code-review (dobór metod zależy od konkretnego zespołu wytwórczego)

Rozwiązanie - Metoda oceny implementacji wzorców projektowych



Rozwiązanie – podejmowanie decyzji

- ▶ zawężony obszar kodu źródłowego do sprawdzenia – wyłącznie fragmenty poniżej oczekiwanego poziomu,
- ▶ znana jest liczba i miejsca występowania usterek, podpowiedzi poprawy.

Wynik oceny:



2 – poniżej oczekiwań

3 – zgodne z oczekiwaniami

Element w kodzie źródł.	Plik	Wiersz	Poziom jakości
Klasa1	Class1	12	3
Pole1	Class1	45	2
Właściwość2	Class2	70	2
Liczba modyfikacji: 80		% kodu poniżej 3 poziomu: 30	



Jeżeli:

*implementacja poniżej 3 poziomu
i program będzie rozbudowywany
i liczba modyfikacji mieści się w A3,*

to: popraw implementację, jeżeli nie to: przejdź do kolejnej iteracji.

Przykład wykorzystania – budowa modelu wzorca i metryk

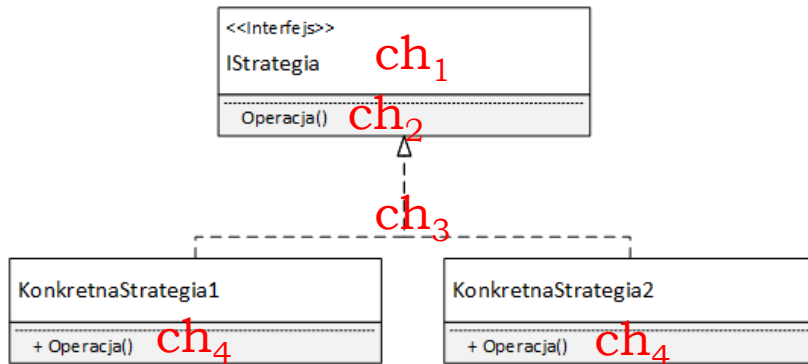


Diagram klas wzorca Strategia

Model Wzorca (charakterystyki):

- ch_1 = (deklaracja interfejsu, M_1)
- ch_2 = (deklaracja operacji, M_2)
- ch_3 = (realizacja interfejsu, M_3)
- ch_4 = (implementacja Operacji, M_4)
- ch_5 = (kontekst implementacji, M_5)
- ch_6 = (inicjalizacja i wybór, M_6)

Model Metryk dla charakterystyka 1 (ch_1):

- $m_{1,1} = \begin{cases} 3 \text{ jeżeli } (MET, brak, public, \dots Strategy, interface) \\ \dots \\ 2 \text{ jeżeli } (MET, abstract, public, \dots Strategy, class) \\ \dots \\ 1 \text{ jeżeli } (MET, brak, public, \dots Strategy, class) \end{cases}$
- $m_{1,2} = \begin{cases} 4 \text{ jeżeli } |MET| = 1 \\ 2 \text{ jeżeli } 2 \leq |MET| \leq 10 \\ 1 \text{ jeżeli } |MET| > 11 \\ 0 \text{ jeżeli } |MET| = 0 \end{cases}$

Gdzie: MET – zbiór metod w danym typie

Przykład wykorzystania - ocena

```
public abstract class SortStrategy
```

$$t_1 = (\emptyset, \emptyset, MET_1, \text{abstract}, \text{public}, \text{SortStrategy}, \text{class})$$

Model programu

$$m_{1,1} = \begin{cases} 3, (MET, \text{brak}, \text{public}, \dots \text{Strategy}, \text{interface}) \\ 2, (MET, \text{abstract}, \text{public}, \dots \text{Strategy}, \text{class}) \\ 1, (MET, \text{brak}, \text{public}, \dots \text{Strategy}, \text{class}) \end{cases}$$

$$m_{1,2} = \begin{cases} 4, |MET| = 1 \\ 2, 2 \leq |MET| \leq 10 \\ 1, |MET| > 11 \\ 0, |MET| = 0 \end{cases}$$

Model metryk

Ocena



Wynik oceny



Wynik metryk: $m_{1,1} = 2$, $m_{1,2} = 4$
 Wynik dla $ch_1 = \text{Min}(m_{1,1}, m_{1,2}) = 2$
 Poniżej oczekiwanego poziomu

Przykład wykorzystania – interpretacja wyniku

- ▶ Wnioskowanie na podstawie wszystkich charakterystyk w kryterium rozbudowy:
 - ▶ 42% kodu wzorca nie spełnia oczekiwanego poziomu jakości,
 - ▶ 42% to 200 leksemów (pojedynczych słów w kodzie) do zmodyfikowania – każda modyfikacja to dodatkowy koszt dla zespołu.



Przykład wykorzystania – ocena po poprawie

public interface SortStrategy

$t_1 = (\emptyset, \emptyset, MET_1, brak, public, SortStrategy, interface)$

Model programu

$$m_{1,1} = \begin{cases} 3, (MET, brak, public, \dots Strategy, interface) \\ 2, (MET, abstract, public, \dots Strategy, class) \\ 1, (MET, brak, public, \dots Strategy, class) \end{cases}$$

$$m_{1,2} = \begin{cases} 4, |MET| = 1 \\ 2, 2 \leq |MET| \leq 10 \\ 1, |MET| > 11 \\ 0, |MET| = 0 \end{cases}$$

Model metryk

Ocena



Wynik oceny

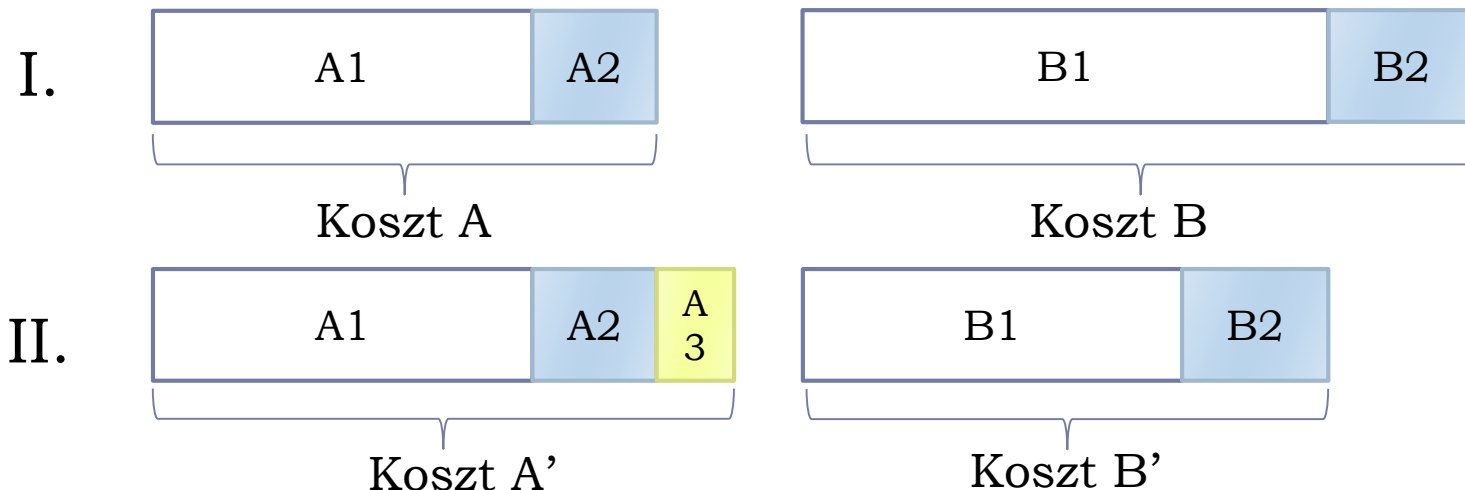


Wynik metryk: $m_{1,1} = 3$, $m_{1,2} = 4$
 Wynik dla $ch_1 = \text{Min}(m_{1,1}, m_{1,2}) = 3$
 Spełnia oczekiwany poziom

Wstępny eksperyment weryfikacyjny

- ▶ symulacja bez oraz z użyciem metody,
- ▶ dodanie nowej operacji do wzorca Strategia,
- ▶ wykazanie nierówności:

$$\text{Koszt A} + \text{B} > \text{Koszt A}' + \text{B}'$$



Wynik eksperymentu

- ▶ każdy z uczestników dodał nową operację oraz poprawił błąd w jednej z istniejących operacji,
- ▶ program składał się z 24 typów (klas, interfejsów itp.).

Uczestnik	Sumaryczny koszt I przebiegu [rhb]	Sumaryczny koszt II przebiegu [rbh]	Zysk czasowy po zastosowaniu proponowanej metody
1	3,5	2	43%
2	5,5	4	28%
3	7	6	14%

Średnio 28%

Plan weryfikacji praktycznej

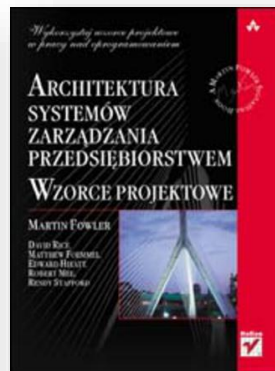
- ▶ Przy współpracy z Quick-Solution.
- ▶ Przebieg:
 - ▶ rozbudowa modelu wzorca i metryk wspólnie z zespołem,
 - ▶ ocena implementacji wzorców po wykonaniu iteracji.
- ▶ Pytania do zespołu wytwórczego:
 - ▶ Ile usterek zostało słusznie znalezionych przez metodę? Ile to fałszywe alarmy? Ile usterek zostało pominięte przez metodę?
- ▶ Budowa narzędzia (programu komputerowego) wspierającego metodę.

Plan weryfikacji dokładności i kosztu użycia

- ▶ analiza porównawcza opracowanej metody oraz Spine, Lepus2,
- ▶ na przykładzie wybranych wzorców (Singleton, Strategia, Fabryka, Stan),
- ▶ oczekiwane rezultaty:
 - ▶ wykazanie większej dokładności proponowanej metody od alternatywnych,
 - ▶ wyznaczenie minimalnego zestawu mierzonych własności (metryk), które są niezbędne, aby odróżnić 2 od 3 poziomu jakości,
 - ▶ wykazanie niższego sumarycznego kosztu użycia proponowanej metody.

Kierunki dalszych badań

- ▶ ocena implementacji wzorców projektowych pod względem innych kryteriów, np. odporność na nieuprawnioną ingerencję w kod, wymiennność modułów oprogramowania,
- ▶ zastosowanie do innych wzorców i dobrych praktyk, np. wzorce architektury (MVC, MVP), wzorce Martina Fowlera, zasady GRASP i SOLID.



Podsumowanie

- ▶ zdefiniowano problem związany z implementacją wzorców projektowych przez zwinne zespoły,
- ▶ zdefiniowano modele:
 - ▶ programu komputerowego,
 - ▶ wzorca projektowego,
 - ▶ metryk,
- ▶ opracowano nową metodę oceny jakości implementacji wzorców,
- ▶ przeprowadzono wstępny eksperyment weryfikujący w/w metodę,
- ▶ opracowano plany dalszej weryfikacji.

Wykaz publikacji 1 / 4

- ▶ Rafał Wojszczyk, *Quality Assessment of Implementation of Strategy Design Pattern*, w: *Advances in Intelligent Systems and Computing*, Springer International Publishing, Switzerland 2017 – przyjęte do druku.
- ▶ Rafał Wojszczyk, Piotr Ratuszniak, *Trudności w implementacji wzorców projektowych w małych zespołach programistycznych*, *Research on Enterprise in Modern Economy – theory and practice*, Gdańsk 2017 – przyjęte do druku.
- ▶ Rafał Wojszczyk, Włodzimierz Khadzhynov, *The Process of Verifying the Implementation of Design Patterns - Used Data Models*, w: *Advances in Intelligent Systems and Computing*, Vol. 521, strony od 103 do 116, Springer International Publishing, Switzerland 2017 - **WoS**.

Wykaz publikacji 2/4

- ▶ Daniel Czyczyn-Egird, Rafał Wojszczyk, *Determining the popularity of design patterns used by programmers based on the analysis of questions and answers on stackoverflow.com Social Network*, w: Communications in Computer and Information Science, vol. 608, strony od 421 do 433, Springer International Publishing, Switzerland 2016 – **WoS**.
- ▶ Rafał Wojszczyk, Robert Wójcik, *The Model of Quality Assessment of Implementation of Design Patterns*, w: Advances in Intelligent Systems and Computing, vol. 474, strony od 515 do 524, Springer International Publishing, Switzerland 2016 – **WoS**.
- ▶ Rafał Wojszczyk, Włodzimierz Khadzhynov, *Wykorzystanie modeli danych do weryfikacji implementacji wzorców projektowych*, w: Zeszyty Naukowe Wydziału Elektroniki i Informatyki nr 10, strony od 193 do 209, Wydawnictwo Uczelniane Politechniki Koszalińskiej, ISSN 1897-7421, Koszalin 2016.
- ▶ Daniel Czyczyn-Egird, Rafał Wojszczyk, *Zastosowanie technik eksploracji danych na przykładzie badania popularności wzorców projektowych w serwisie społecznościowym Stackoverflow.com*, w: Zeszyty Naukowe Wydziału Elektroniki i Informatyki nr 10, strony od 81 do 94, Wydawnictwo Uczelniane Politechniki Koszalińskiej, ISSN 1897-7421, Koszalin 2016.

Wykaz publikacji 3/4

- ▶ Rafał Wojszczyk, Włodzimierz Khadzhynov, *Data Models in the Verification of the Singleton Pattern*, w: Journal of Theoretical and Applied Computer Science, vol. 9, no. 3/2015, Szczecin 2016.
- ▶ Rafał Wojszczyk, *Model oceny jakości implementacji wzorców projektowych w oprogramowaniu*, w: Innowacje w zarządzaniu i inżynierii produkcji, Tom II, strony od 300 do 309, Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole 2016.
- ▶ Rafał Wojszczyk, *The model and function of quality assessment of implementation of design patterns*, w: Applied Computer Science, Vol. 11, No. 3, strony od 45 do 56, Politechnika Lubelska, ISSN 1895-3735, Lublin 2015.
- ▶ Rafał Wojszczyk, *Weryfikacja poprawności implementacji struktury wzorców projektowych w oparciu o model referencyjny*, w: Od procesów do oprogramowania: badania i praktyka, strony od 73 do 83, Zeszyty Rady Naukowej Polskiego Towarzystwa Informatycznego, ISBN 978-83-60810-73-6, Warszawa 2015.

Wykaz publikacji 4/4

- ▶ Rafał Wojszczyk, *Koncepcja hybrydowej metody do oceny jakości zaimplementowanych wzorców projektowych*, w: Zeszyty Naukowe Wydziału Elektroniki i Informatyki nr 7, strony od 17 do 26, Wydawnictwo Uczelniane Politechniki Koszalińskiej, ISSN 1897-7421, Koszalin 2015.
- ▶ Rafał Wojszczyk, *Pozyskiwanie struktury obiektowej z kodu zarządzanego przy wykorzystaniu metod inżynierii odwrotnej*, w: Inżynieria oprogramowania: badania i praktyka, strony od 199 do 213, Zeszyty Rady Naukowej Polskiego Towarzystwa Informatycznego, ISBN 978-83-63919-15-3, Warszawa 2014.
- ▶ Rafał Wojszczyk, *Porównanie sposobów reprezentacji wzorców projektowych*, w: Modele inżynierii teleinformatyki 9, strony od 133 do 145, Wydawnictwo Uczelniane Politechniki Koszalińskiej, ISSN 2353-6535, Koszalin 2014.
- ▶ Rafał Wojszczyk, *Zestawienie metryk oprogramowania obiektowego opartych na statycznej analizie kodu źródłowego*, w: Zarządzanie projektami i modelowanie procesów, strony od 95 do 107, Zeszyty Rady Naukowej Polskiego Towarzystwa Informatycznego, ISBN 978-83-7518-599-7, Warszawa 2013.

Model oceny jakości implementacji wzorców projektowych

mgr inż. Rafał Wojszczyk

dziękuję za uwagę