

5 rzeczy o których usłyszycie dopiero w pracy

Jakub Marchwicki · YDP SA · @kubem

Agenda

Agenda

... o kodzie nowym

Agenda

... o kodzie nowym
... o kodzie który już istnieje

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach
- ... o zespołach

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach
- ... o zespołach

Robert C. Martin

CLEAN CODE
CLEAN CODE
CZYSTY KÓD

PODRĘCZNIK
DOBREGO PROGRAMISTY



Poznaj najlepsze metody tworzenia doskonałego kodu

Jak piisać dobry kod, a zły przekształcać w dobry?

Jak formatować kod, aby osiągnąć maksymalną czytelność?

Jak implementować pełną obsługę błędów bez zasłaniając logiki kodu?

Walentynka

Podstawowe zasady Wuja Boba

Podstawowe zasady Wuja Boba

Nazwy

Podstawowe zasady Wuja Boba

Nazwy
Funkcje

Podstawowe zasady Wuja Boba

Nazwy

Funkcje

Komentarze

Podstawowe zasady Wuja Boba

Nazwy

Funkcje

Komentarze

Formowanie kodu

Podstawowe zasady Wuja Boba

Nazwy

Funkcje

Komentarze

Formowanie kodu

Obiekty i struktury danych

Podstawowe zasady Wuja Boba

Nazwy

Funkcje

Komentarze

Formowanie kodu

Obiekty i struktury danych

Obsługa błędów

Nazwy

BadCode.java

```
for (int i=0; i<10; i++){
    k += ((l[i]*1.5) / 3 );
}
```

GoodCode.java

```
float milleageRate;
const int NUMER_OF_EMPLOYEE = 3;
float sum = 0;

for ( int i=0; i<numberOfTrips; i++ ){
    float totalCompensation = tripLength[i] * milleageRate;
    float deduction = totalCompensation / NUMER_OF_EMPLOYEE;
    sum += deduction;
}
```

Funkcje

Funkcje

ZASADA PIERWSZA

Funkcje

ZASADA PIERWSZA
funkcje powinny być małe

Funkcje

ZASADA PIERWSZA
funkcje powinny być małe

ZASADA DRUGA

Funkcje

ZASADA PIERWSZA
funkcje powinny być małe

ZASADA DRUGA
funkcje powinny być jeszcze mniejsze

Functions should **do**
one thing.
Should do it well and should do
it only

```
try {  
  
    Context currContext = context.clone();  
    currContext.parseElement(element);  
  
    if (nodeName.equals("mrow")){  
        int nodesCount = XmlUtils.getChildElementNodesCount(element);  
        Vector<Token> tokens = new Vector<Token>();  
        for (int n = 0 ; n < nodesCount ; n ++){  
            Element childElement = XmlUtils.getChildElementNodeAtIndex(n, element);  
            Token parsedToken = parseElement( childElement , currContext );  
            tokens.add( parsedToken );  
        }  
        return new MRow(tokens);  
    } else if (nodeName.equals("mtable")){  
        Vector<Vector<Token>> tokens = new Vector<Vector<Token>>();  
        NodeList trNodes = element.getChildNodes();  
  
        for (int row = 0 ; row < trNodes.getLength() ; row ++){  
            if (trNodes.item(row).getNodeType() == Node.ELEMENT_NODE && "mtr".equals(trNodes.item(row).getNodeName())){  
                NodeList tdNodes = element.getChildNodes().item(row).getChildNodes();  
                tokens.add(new Vector<Token>());  
                for (int col = 0 ; col < tdNodes.getLength() ; col ++ ){  
                    if (tdNodes.item(col).getNodeType() == Node.ELEMENT_NODE && "mtd".equals(tdNodes.item(col).getNodeName())){  
                        tokens.get(tokens.size()-1).add(parseElement( XmlUtils.getChildElementNodeAtIndex(0, (Element) tdNodes.item(col)) , currContext ));  
                    }  
                }  
            }  
        }  
        MTable mTable = new MTable(tokens);  
        mTable.setColumnAlign(XmlUtils.getAttribute(element, "columnalign"));  
        return mTable;  
    } else if (nodeName.equals("mfrac")){
```

Why? Oh why?

Komentarze

don't

“ If you decide to write a comment, then spend the time necessary to make sure it is the best comment you can write

– Robert C. Martin

Obiekty

Zasada
niewiedzy

albo minimalnej wiedzy

moduł nie powinien wiedzieć o wnętrzu obiektów,
którymi manipuluje

Prawo Demeter o zabawie

Prawo Demeter o zabawie z **samym sobą**

Prawo Demeter o zabawie
z **samym sobą**
własnymi zabawkami (ale bez
rozbierania!)

Prawo Demeter o zabawie
z **samym sobą**
własnymi zabawkami (ale bez
rozbierania!)
wszystkim co dostałeś

Prawo Demeter o zabawie
z **samym sobą**
własnymi zabawkami (ale bez
rozbierania!)
wszystkim co dostałeś
jak zrobiłeś samodzielnie - to
Twoje!

**THIS IS NOT 'NAM, THIS IS
PROGRAMMING**

THERE ARE RULES

SOLIDny programista

SOLIDny programista

Single Responsibility Principle

klasa powinna mieć tylko jeden powód do zmiany

SOLIDny programista

Single Responsibility Principle

klasa powinna mieć tylko jeden powód do zmiany

Open Closed Principle

klasę można łatwo rozszerzać, nie modyfikując jej

SOLIDny programista

Single Responsibility Principle

klasa powinna mieć tylko jeden powód do zmiany

Open Closed Principle

klasę można łatwo rozszerzać, nie modyfikując jej

Liskov Substitution Principle

klasy pochodne muszą być przeźroczystymi

zamiennikami klasy nadzędnej

SOLIDny programista

SOLIDny programista

Interface Segregation Principle

dla różnych klientów twórz osobne interfejsy

SOLIDny programista

Interface Segregation Principle

dla różnych klientów twórz osobne interfejsy

Dependency Inversion Principle

bądź zależny od abstrakcji a nie od konkretnych
implementacji

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch



Wzorce projektowe to
nie tylko
singleton

Po co zawracać sobie tym głowę?

Jakość to nie
tylko znaczy że dzia-
ła

Więc czym jest jakość?

Więc czym jest jakość?

low coupling

Więc czym jest jakość?

**low coupling
high cohesion**

Więc czym jest jakość?

low coupling
high cohesion
well defined API

Więc czym jest jakość?

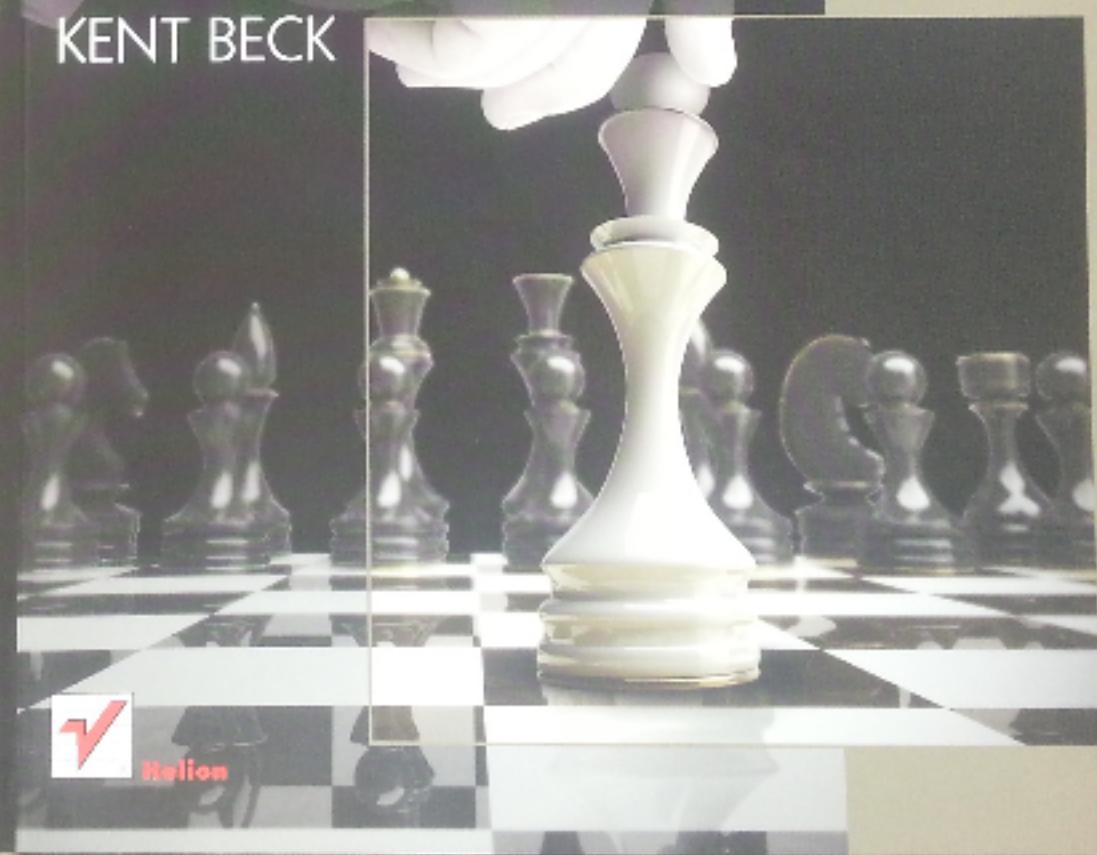
low coupling
high cohesion
well defined API
low cyclomatic complexity

Sprawdzone rozwiązania
Twoich problemów!



WZORCE IMPLEMENTACYJNE

KENT BECK



Helion

Wartości

Wartości

Kod jest podstawowym
medium komunikacji w
projektie

Wartości

Kod jest podstawowym
medium komunikacji w
projektie

Jako zespół jesteśmy
jednością

Wartości

Wartości

Programy są częściej czytane
niż pisane

Wartości

Programy są częściej czytane niż pisane

Więcej czasu poświęcamy na modyfikację istniejącego kodu niż na tworzenie nowego

Implementation patterns

Implementation patterns

KOMUNIKACJA

Implementation patterns

KOMUNIKACJA

kod źródłowy powinno się czytać jak książkę

Implementation patterns

KOMUNIKACJA

kod źródłowy powinno się czytać jak książkę

PROSTOTA

Implementation patterns

KOMUNIKACJA

kod źródłowy powinno się czytać jak książkę

PROSTOTA

wprowadzaj złożoność tylko kiedy jest to konieczne

Implementation patterns

KOMUNIKACJA

kod źródłowy powinno się czytać jak książkę

PROSTOTA

wprowadzaj złożoność tylko kiedy jest to konieczne

ELASTYCZNOŚĆ

Implementation patterns

KOMUNIKACJA

kod źródłowy powinno się czytać jak książkę

PROSTOTA

wprowadzaj złożoność tylko kiedy jest to konieczne

ELASTYCZNOŚĆ

to dodatkowa złożoność, więc wprowadzaj ją tylko tam gdzie to konieczne

Implementation patterns

Implementation patterns

LOKALNE KONSEKWENCJE

Implementation patterns

LOKALNE KONSEKWENCJE

zmiana w jednym miejscu nie powoduje zmian
w innych

Implementation patterns

LOKALNE KONSEKWENCJE

zmiana w jednym miejscu nie powoduje zmian
w innych

DANE I LOGIKA RAZEM

Implementation patterns

LOKALNE KONSEKWENCJE

zmiana w jednym miejscu nie powoduje zmian
w innych

DANE I LOGIKA RAZEM

ponieważ zmieniają się w tym samym czasie

Implementation patterns

LOKALNE KONSEKWENCJE

zmiana w jednym miejscu nie powoduje zmian
w innych

DANE I LOGIKA RAZEM

ponieważ zmieniają się w tym samym czasie

SYMETRIA

Implementation patterns

LOKALNE KONSEKWENCJE

zmiana w jednym miejscu nie powoduje zmian
w innych

DANE I LOGIKA RAZEM

ponieważ zmieniają się w tym samym czasie

SYMETRIA

utrzymuj podobny poziom abstrakcji

“Czysty kod jest prosty i bezpośredni. Czysty kod czyta się jak dobrze napisaną prozę. Czysty kod nigdy nie zacieśnia zamiarów projektanta; jest pełen trafnych abstrakcji i prostych ścieżek sterowania

– Grady Booch - Software Archeologist - IBM

Agenda

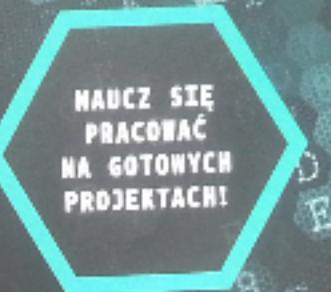
- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach
- ... o zespołach

<Michael_Feathers>

PRACA

Z ZASTANYM KODEM

NAJLEPSZE TECHNIKI



NAUCZ SIĘ
PRACOWAĆ
NA GOTOWYCH
PROJEKTACH!

Helion 

W istniejącym kodzie czai się ZŁO

W istniejącym kodzie czai się ZŁO

requirements change

W istniejącym kodzie czai się ZŁO

**requirements change
you never start from scratch**

W istniejącym kodzie czai się ZŁO

**requirements change
you never start from scratch
spaghetti monster hides
everywhere**

take opaque, convoluted
system and slowly, gradually

reverse rot

piece by piece turn it into well
designed system

Jak przeżyć?

Jak przeżyć?

małe, inkrementalne zmiany

Jak przeżyć?

małe, inkrementalne zmiany
brak branchy

Jak przeżyć?

małe, inkrementalne zmiany
brak branchy
nieustanna refaktoryzacja

Jak przeżyć?

małe, inkrementalne zmiany
brak branchy
nieustanna refaktoryzacja
wytyczanie granic i fasad
istniejących systemów

Jak przeżyć?

małe, inkrementalne zmiany
brak branchy
nieustanna refaktoryzacja
wytyczanie granic i fasad
istniejących systemów
ciągła integracja

Jak przeżyć?

małe, inkrementalne zmiany
brak branchy
nieustanna refaktoryzacja
wytyczanie granic i fasad
istniejących systemów
ciągła integracja
testy, testy, testy

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach
- ... o zespołach

Kent Beck

TDD.

Sztuka tworzenia dobrego kodu



Twórz niezawodny kod!

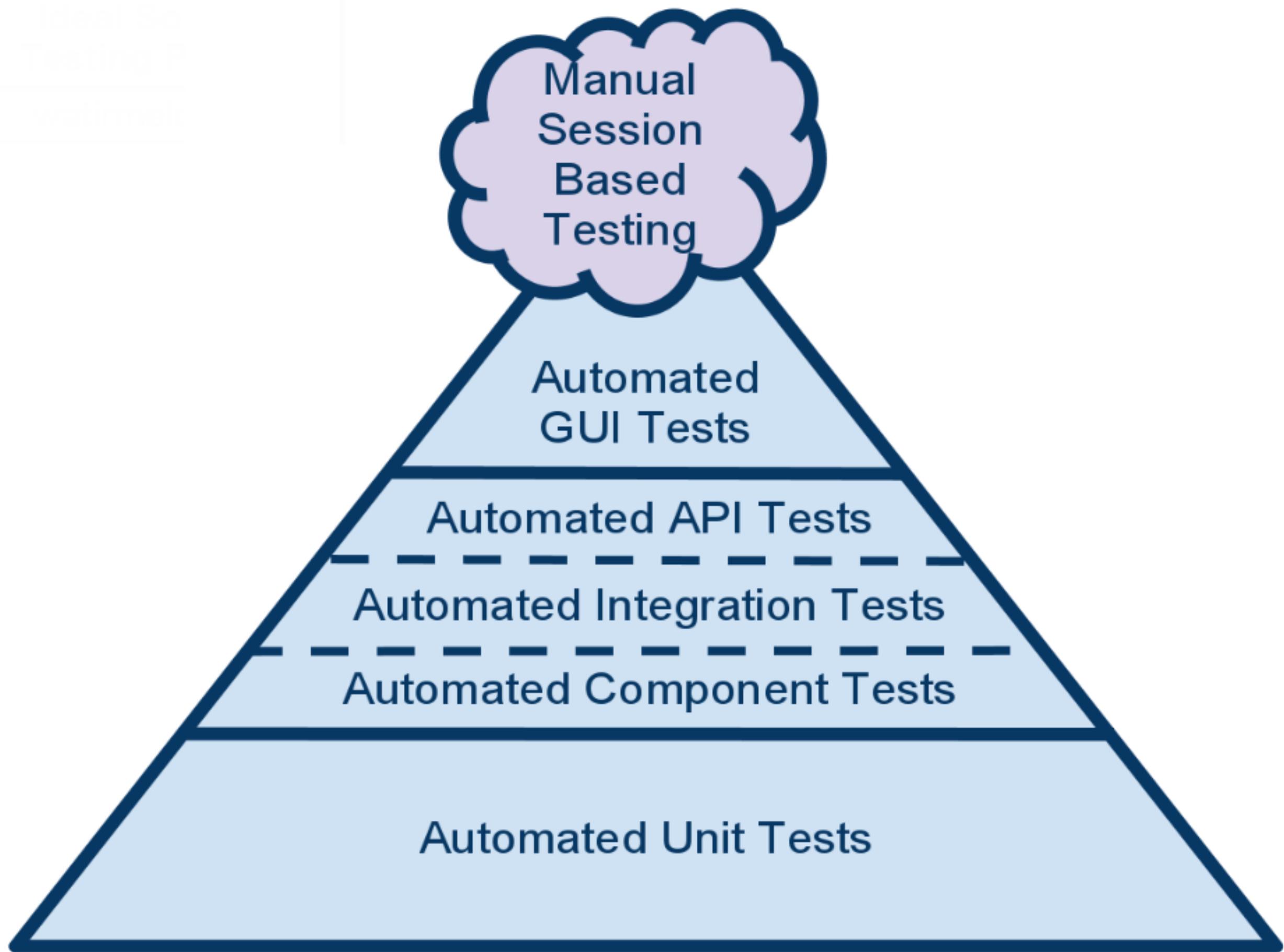


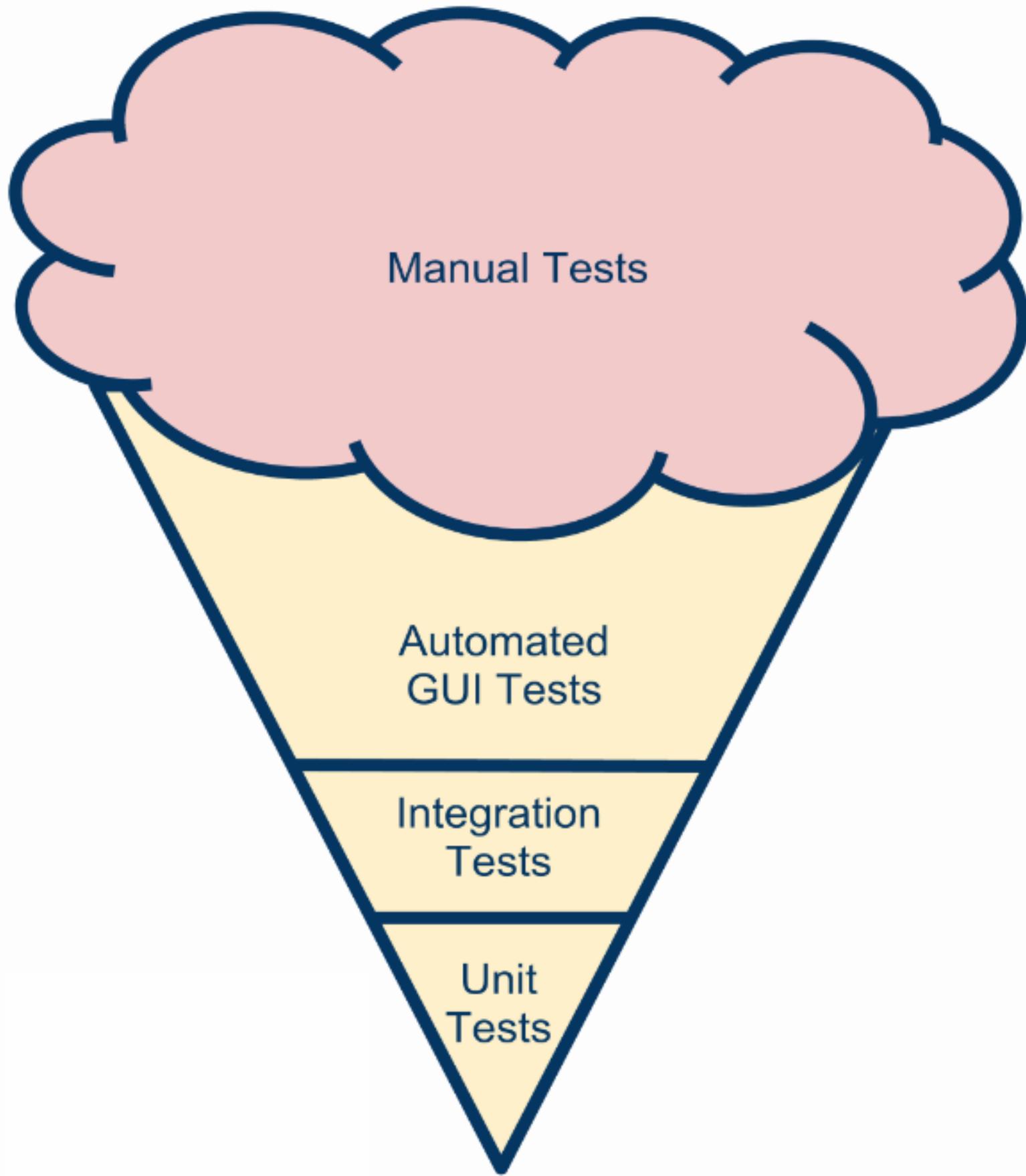
określenie

"projekt
studencki"

nie jest synonimem dobrej
jakości

testerzy **nie** są od
testowania





“There have been lots of controversial blogs and articles written about TDD over the years and there still are(...) however, they are just rants. The bottom line is that TDD works, and everybody needs to get over it

– Robert C. Martin - The Clean Coder

Po co TDD?

Po co TDD?

Pewność, że działa i będzie działać

Po co TDD?

Pewność, że działa i będzie działać
Szybki feedback dla programisty

Po co TDD?

Pewność, że działa i będzie działać

Szybki feedback dla programisty

Redukcja liczby błędów (10x)

Po co TDD?

Pewność, że działa i będzie działać

Szybki feedback dla programisty

Redukcja liczby błędów (10x)

Testy to najlepsza dokumentacja

Po co TDD?

Pewność, że działa i będzie działać

Szybki feedback dla programisty

Redukcja liczby błędów (10x)

Testy to najlepsza dokumentacja

TDD pozwala na projektowanie przejrzystych

Po co TDD?

Pewność, że działa i będzie działać

Szybki feedback dla programisty

Redukcja liczby błędów (10x)

Testy to najlepsza dokumentacja

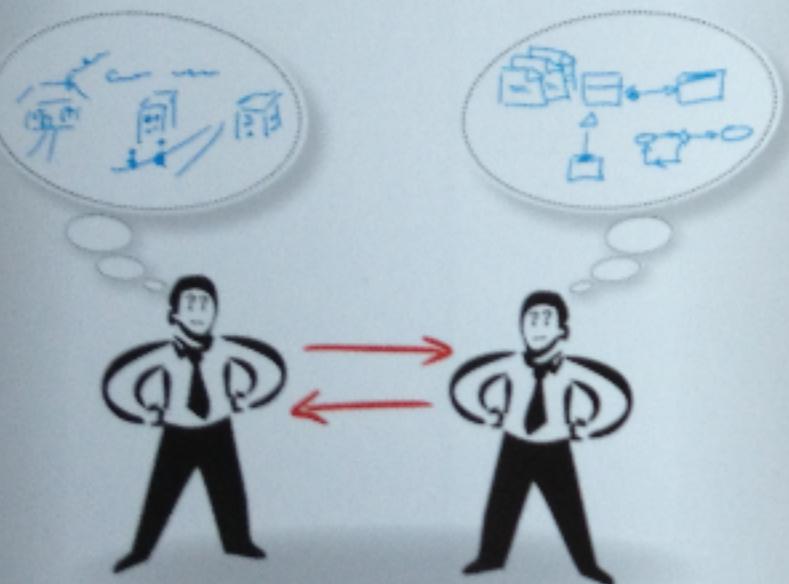
TDD pozwala na projektowanie przejrzystych

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach
- ... o zespołach

Michał Bartyzel

Oprogramowanie szyte na miarę



Jak rozmawiać z klientem, który nie wie, czego chce

Jeśli zapytasz stu ludzi, jak wybierają sobie piłkę, każdy powie Ci coś innego.
Nie osiąb, tyle różnych spojrzeń na każdy temat. Dlaczego więc zakładasz, że dokładnie wiesz,
czego potrzebuje Twój klient? Dlaczego zakładasz, że Twój klient wie, co masz na myśli?
Gdy zaproponujesz mu więc nowego systemu informatycznego?



one
partner

Ujęcie klasyczne

Ujęcie klasyczne

specyfikacja wymagań

Ujęcie klasyczne

specyfikacja wymagań
zarządzanie wymaganiami

Ujęcie klasyczne

specyfikacja wymagań
zarządzanie wymaganiami
klasyfikacja wymagań

Ujęcie klasyczne

specyfikacja wymagań
zarządzanie wymaganiami
klasyfikacja wymagań
diagramy UML

Ujęcie klasyczne

specyfikacja wymagań
zarządzanie wymaganiami
klasyfikacja wymagań
diagramy UML
modele COCOMO

Ujęcie klasyczne

specyfikacja wymagań
zarządzanie wymaganiami
klasyfikacja wymagań
diagramy UML
modele COCOMO
zarządzanie zmianą

Czy to wszystko? Czy
wystarczy?

Oprogramowanie **ma**
służyć
użytkownikom

Oprogramowanie ma
rozwiązać
problemy

Klient wie **czego
potrzebuje**

Agenda

- ... o kodzie nowym
- ... o kodzie który już istnieje
- ... o testach
- ... o użytkownikach
- ... o zespołach

The Addison-Wesley Signature Series



A KENT BECK
Signature Book
Autobiography

IMPLEMENTING LEAN SOFTWARE DEVELOPMENT

FROM CONCEPT TO CASH

MARY AND TOM
POPPENDIECK



Forewords by Jeff Sutherland and Kent Beck

Agile Principles

Agile Principles

Individuals and interactions

over processes and tools

Agile Principles

Individuals and interactions

over processes and tools

Working software

over comprehensive documentation

Agile Principles

Individuals and interactions

over processes and tools

Working software

over comprehensive documentation

Customer collaboration

over contract negotiation

Agile Principles

Individuals and interactions

over processes and tools

Working software

over comprehensive documentation

Customer collaboration

over contract negotiation

Responding to change

over following a plan

To podstawa...

To podstawa...

Agile to nie tylko Scrum

To podstawa...

Agile to nie tylko Scrum
Kanban, Lean

To podstawa...

**Agile to nie tylko Scrum
Kanban, Lean
Agile to filozofia
to sposób pracy**

To podstawa...

Agile to nie tylko Scrum

Kanban, Lean

**Agile to filozofia
to sposób pracy**

Inne traktowanie problemów

To podstawa...

Agile to nie tylko Scrum

Kanban, Lean

**Agile to filozofia
to sposób pracy**

Inne traktowanie problemów

Inne wartości

czy to już

WSZYSTKO?

Liczą się rezultaty!

Liczą się rezultaty!

Projekty opensource nie tylko kod

Liczą się rezultaty!

Projekty opensource nie tylko kod
Dokumentacja, testy,
przykłady

Liczą się rezultaty!

Projekty opensource nie tylko kod
Dokumentacja, testy,
przykłady
JUGi i inne spotkania takie jak to

Liczą się rezultaty!

Projekty opensource nie tylko kod
Dokumentacja, testy,
przykłady
JUGi i inne spotkania takie jak to
Java, Ruby, PHP, JavaScript

Liczą się rezultaty!

Projekty opensource nie tylko kod
Dokumentacja, testy,

przykłady

JUGi i inne spotkania takie jak to

Java, Ruby, PHP, JavaScript

When one teaches, two learn!



Jak żyć?

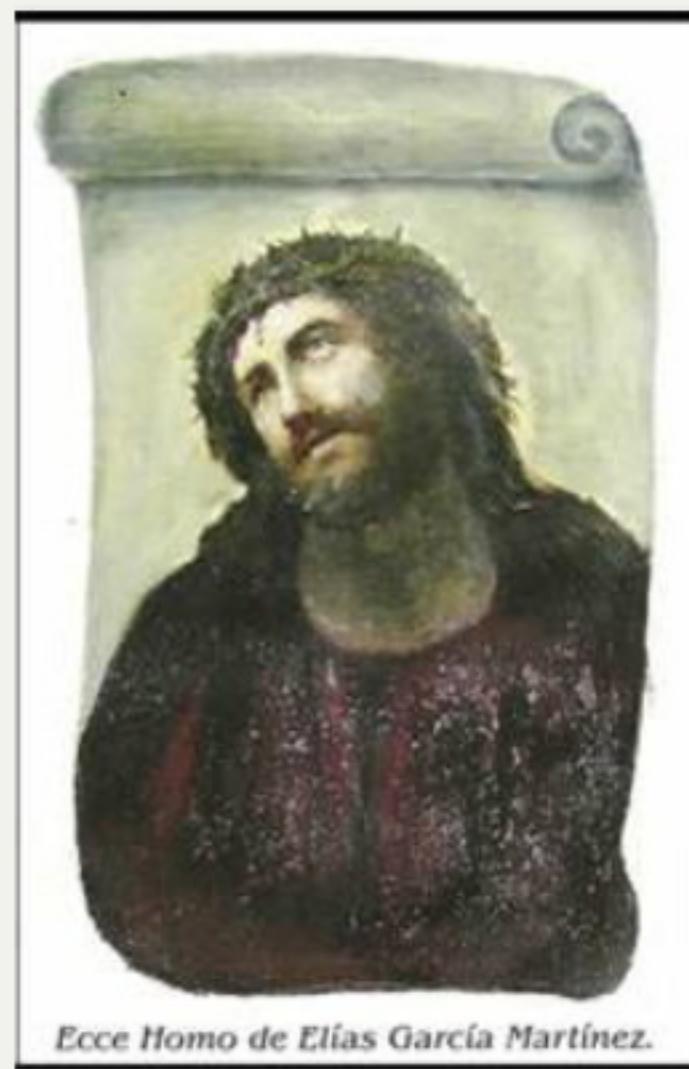
Bo w sytuacjach trudnych...



... brniemy nie wiadomo gdzie ...



... mimo że cel był wyraźny



Ecce Homo de Elías García Martínez.



Questions?

Jakub Marchwicki

 @kubem