



# Naturalny Porządek Refaktoryzacji ®

Java Developer Days 2012





**Dlaczego refaktoryzacja jest jak seks?**

Naturalny Porządek Refaktoryzacji ®

Kończysz dopiero wtedy, gdy już jesteś kompletnie wyczerpany!

Jedna pomyłka i jesteś umoczonej do końca życia!

Więcej się o tym mówi niż robi.

Można to robić dla przyjemności lub dla pieniędzy.

Przydaje się druga osoba, żeby uzyskać pełną satysfakcję.

Początkujący robią wokół tego dużo szumu.

Niektórzy w tym obszarze są szczególnie utalentowani,

.. ale niektórzy są w tym bardzo kiepscy i obrzucą Cię błotem, jeśli im to wytkniesz.

Słabo idzie po pijanemu, ale wtedy masz na to największą ochotę.

Będzie Ci tego brakować, jeśli będziesz miał dłuższą przerwę.



## Jak jest (w Polsce)?

Naturalny Porządek Refaktoryzacji ®

Dawniej (>5 lat temu)	Dziś
Mało kto o tym mówił	Wszędzie się o tym mówi
Mało kto to robił	Mało kto to robi „na poważnie”
Nie było zrozumienia u menedżmentu	Często menedżment (IT) rozpoznaje to hasło, ale ciągle nie ma przyzwolenia
Nie było wiadomo jak to robić z sensem	Robi się to nieregularnie. Nie wiadomo jak to robić w dużych odziedziczonych projektach



Obecnie refaktoryzują w sposób regularny i konsekwentny tylko jednostki – Fascynaci!

**CZAS TO ZMIENIĆ!**



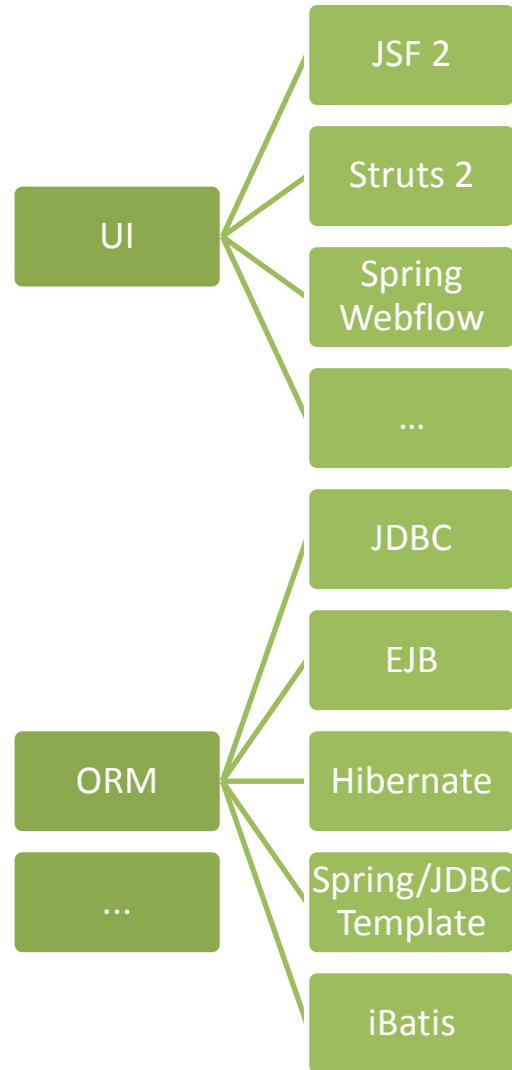
## Mistrzostwo to Ćwiczenie podstaw

Naturalny Porządek Refaktoryzacji ®

wpROWadZAnle.dozMian!A,!aczkoLwieK  
!aNazywaNyCH,opROgrAmOWaniA ,iczesco  
!i!!prOgreSywnyMizwyKLE.poniEkAd,!PieRWotn  
a.StrUkturEnarusza.a.!PRoGramU.kUMulaCja,Z  
mian a.TEN,I !prOcEs!,,i.Tylko,liCzba,nasiLA,  
,AcZkoLwiek,powiaZaniIntErakCji  
poniEkAD!!roZnymi!ponlekad!,I!pomieDzYMod  
uLamiw,sYStemiEZWiEkSZasie!.a, co!  
,a,utruDniAzRoZUmienie!!A,!acZKolwiek!!..go,t  
akzE!jEgO..modYfikacje..dalsze,

Wprowadzanie zmian do oprogramowania (nazywanych często progresywnymi) zwykle narusza pierwotną strukturę programu, a kumulacja zmian tylko ten proces nasila. Liczba powiązań i interakcji pomiędzy różnymi modułami w systemie zwiększa się, co utrudnia zrozumienie go, a także jego dalsze modyfikacje.

**Problem leży w pierwszej kolejności w  
czytelności kodu, a nie w kompetencjach  
programistów!**





**Rozwój technologiczny nie daje spełnienia!**

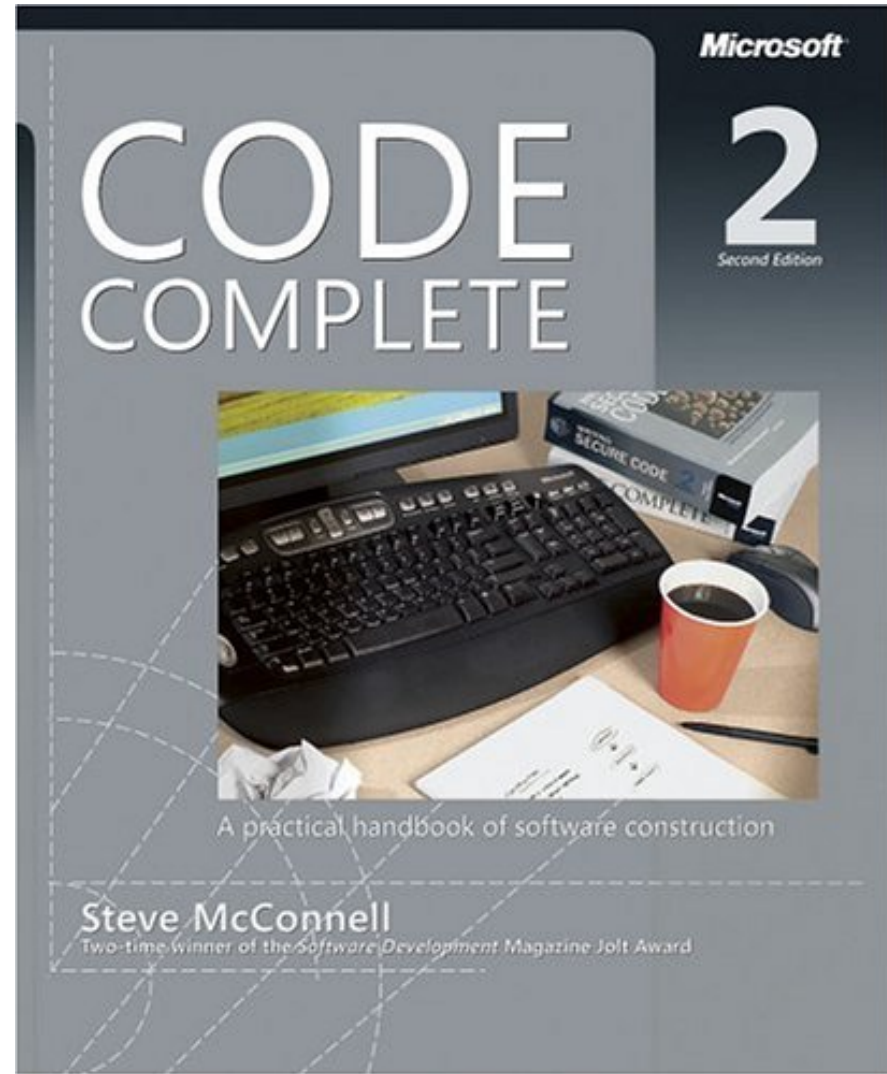
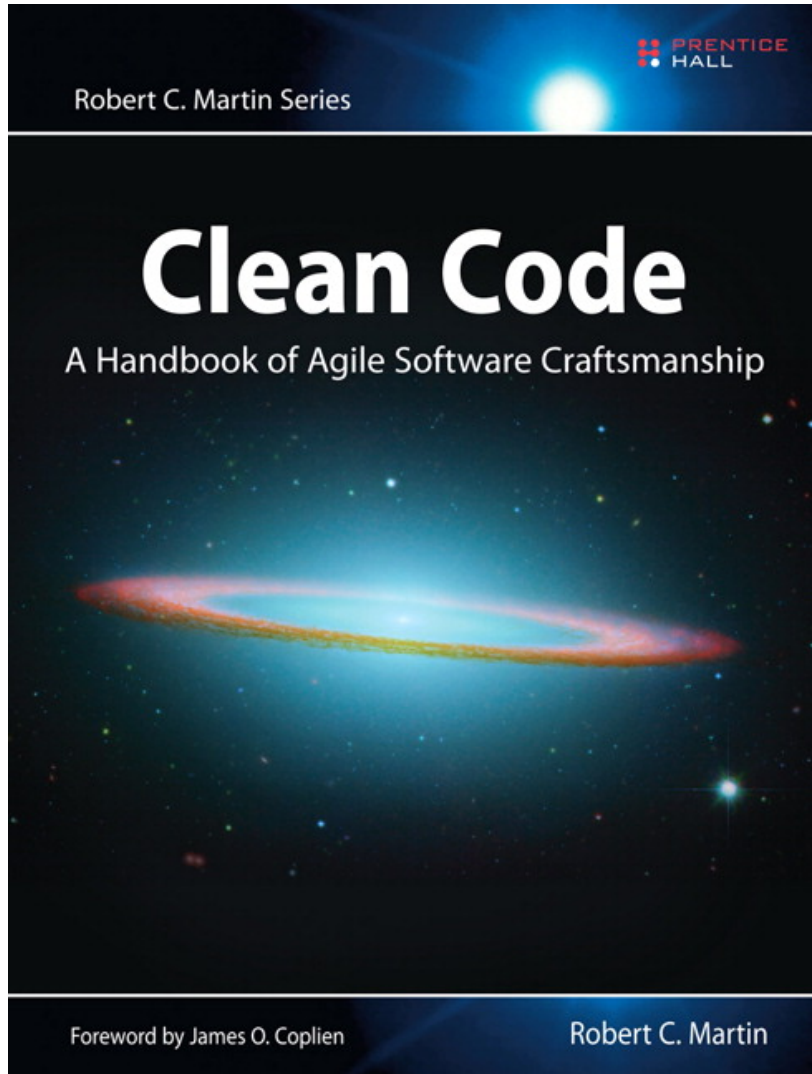
**Rozwój w obszarze narzędzi mentalnych daje  
pełną satysfakcję!**



**Mistrzostwo osiąga się poprzez ciągłe  
doskonalenie podstawowych technik.**



- # Nazwy
- # Nazwy
- # Nazwy
- # Komponowanie metod
- # Pojedyncza odpowiedzialność i kompozycja
- # Wzorce projektowe
- # \*Orientacja we wzorcach architektonicznych





To dlaczego refaktoryzacja się nie udaje?

- # Bo nie ma na to czasu
- # Bo biznes się nie chce zgodzić na refaktoryzacje
- # U nas jest taki burdel, że to nie ma sensu

- # Musi stać się ważne
- # Musi to być wysiłek zespołowy
- # Trzeba mieć odwagę
- # Musi być lider zmiany

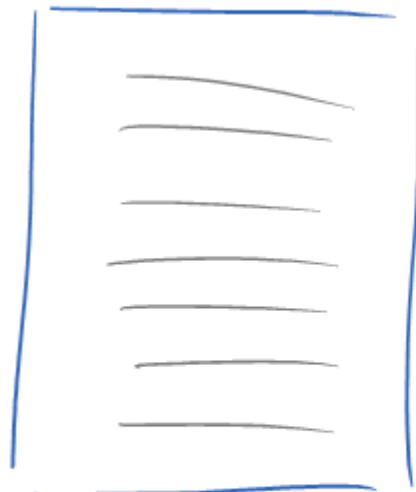
Będzie trudno ... na szczęście głównie tylko na początku

**Refaktoryzacja musi być nawykiem, żeby się działa.**



Naturalny Porządek Refaktoryzacji ®





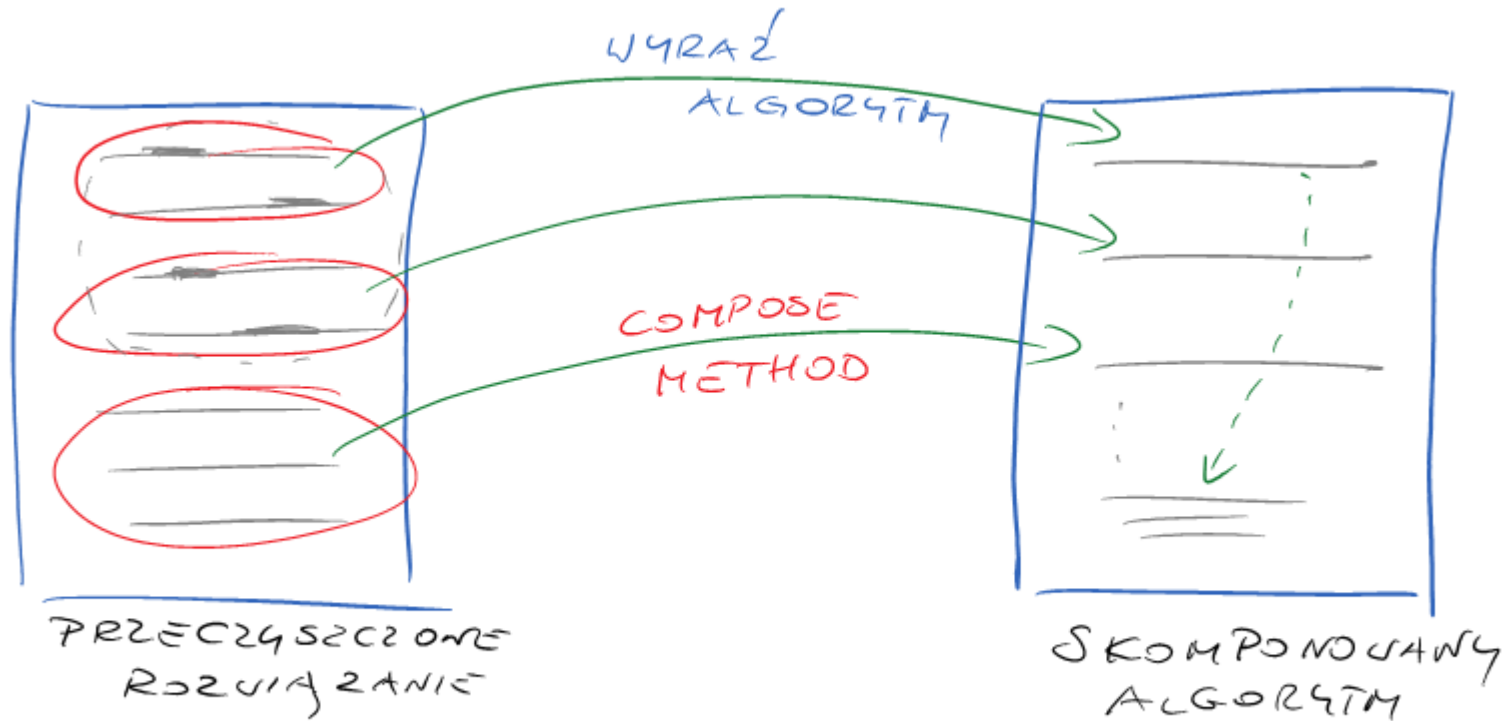
PIERUSZE  
ROZWIĄZANIE

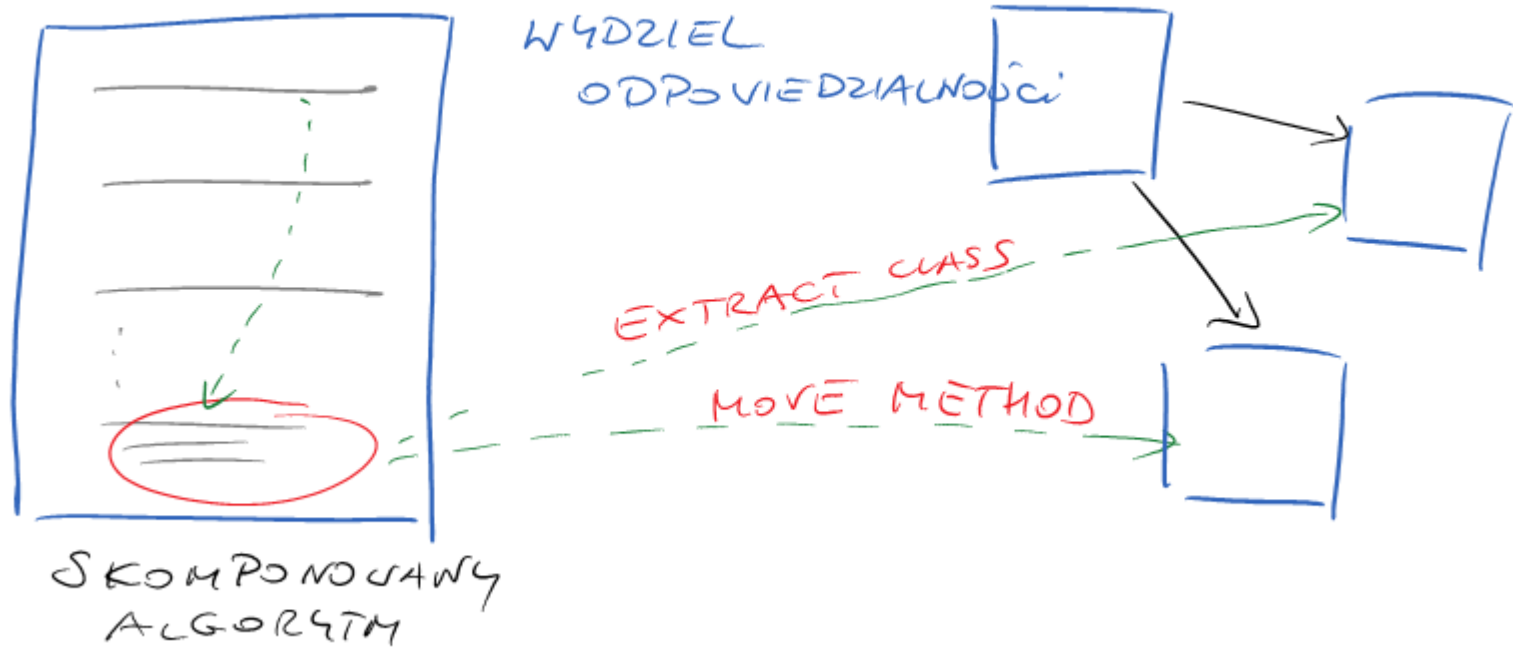
ZROZUM

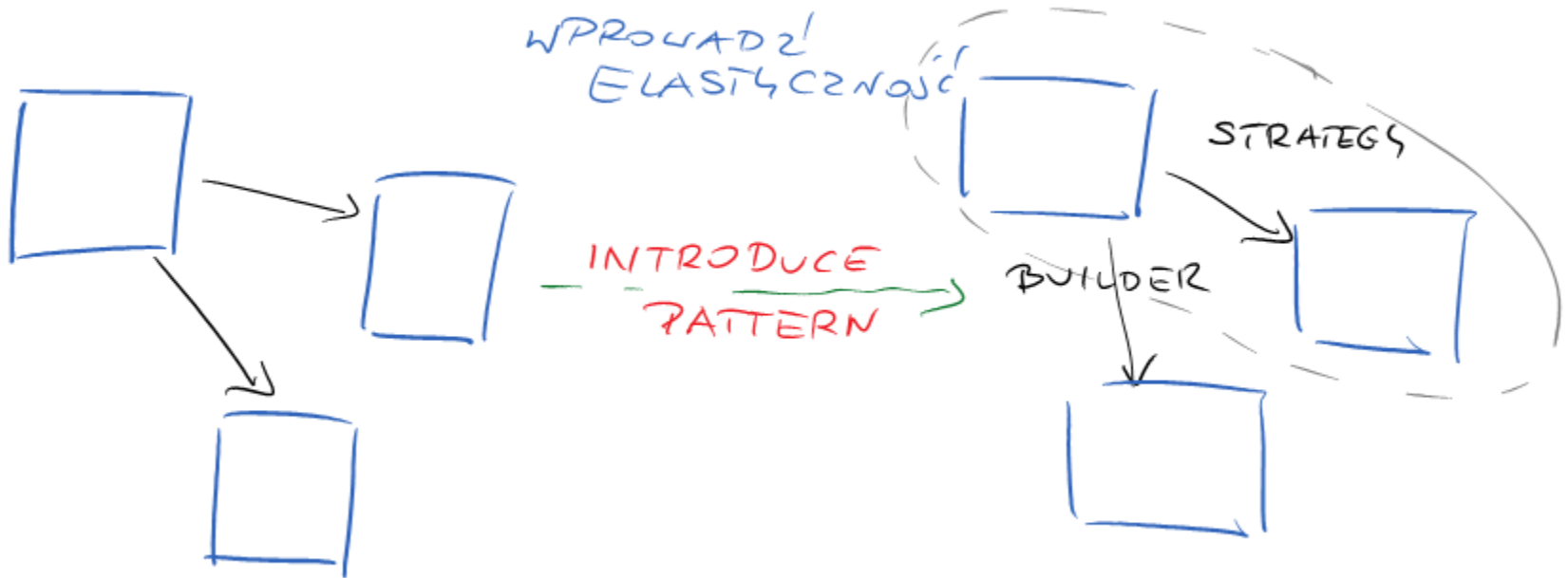
NAZELNICIWO  
UPRASZCZANIE  
WARUNKO'U

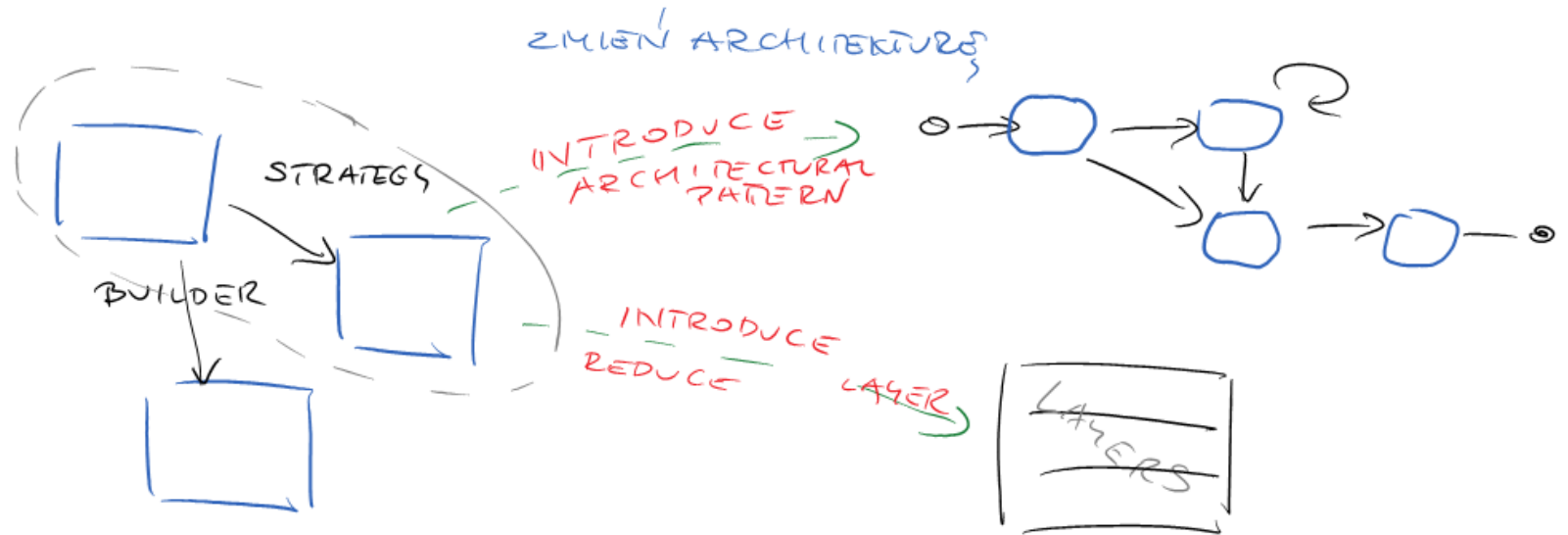


PRZECZYSCZONE  
ROZWIĄZANIE









- # To jest model
- # Duże klasy i duże metody mają kod na różnym poziomie wg NOR
- # Bardzo rzadko kod jest na poziomie Skomponowanej metody
- # Kolejne kroki procesu dzieją się odpowiednio rzadziej niż poprzednie

- # 85 % twoich refaktoryzacji powinna dotyczyć zmian nazewnictwa, komponowania metod i upraszczania warunków
- # W większości mogą być one automatyczne
- # Czasami kroki można zamienić

# PODSTAWY



1. Przekonaj siebie, że to jest ważne
  2. Przekonaj zespół, że to jest ważne
  3. Przekonaj kierownika, że to jest ważne
  4. Określ plan wdrożenia procesu
  5. Znajdź przynajmniej 3 sposoby, jakie zapewnią, że inicjatywa przeżyje trudności
- lub przekonaj kierownika, żeby zatrudnił nas ;-)

