

Michał Bartyzel

X-DRIVEN DESIGN, Y-DRIVEN DEVELOPMENT NICZEGO NIE ZMIENIĄ



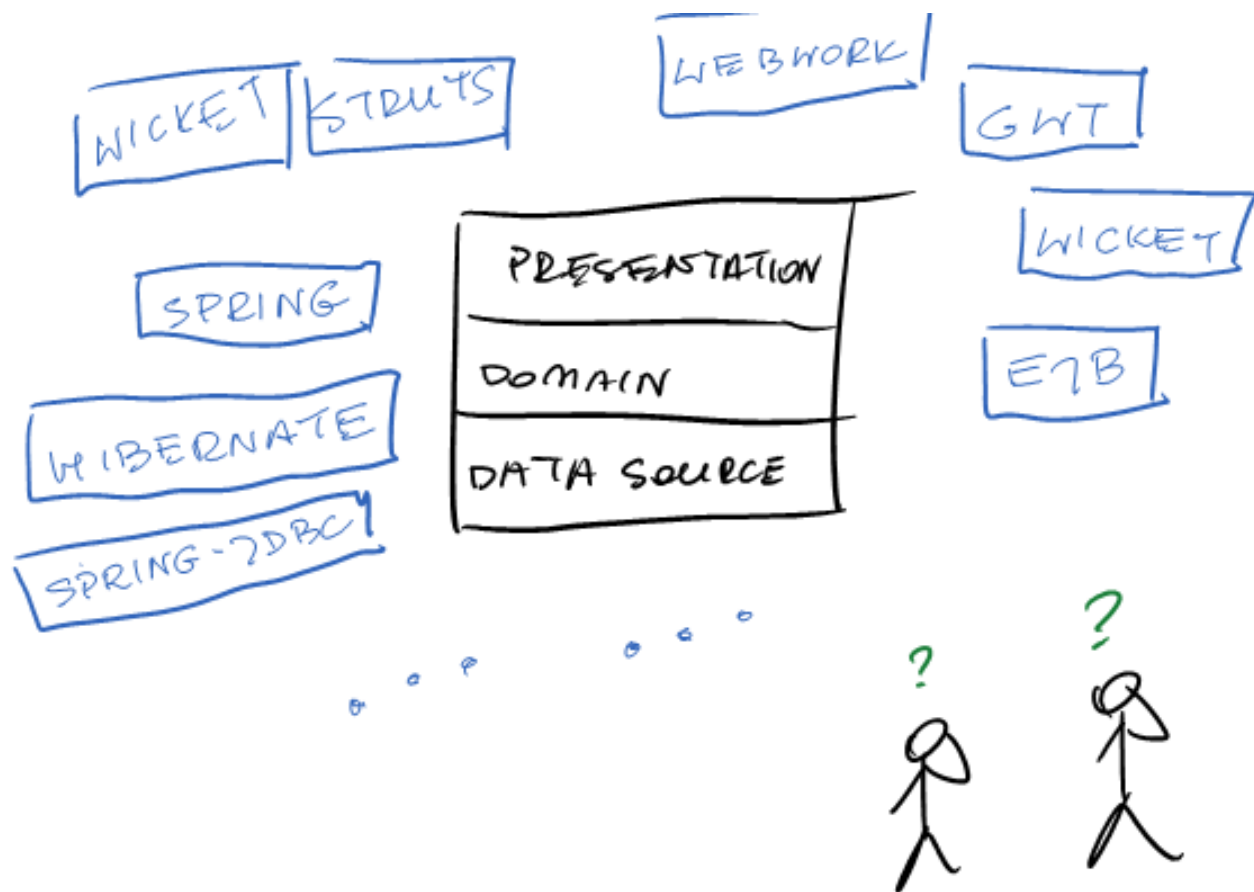
mbartyzel.blogspot.com
@MichalBartyzel



33rd Degree Conference
Conference for Java Masters

13-15 March 2013
Warsaw, Poland

„Lepszy” framework



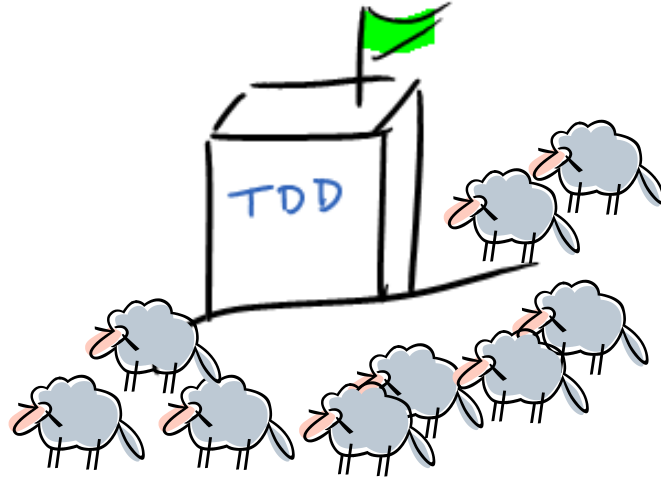
- A. Rozwiązuje problemy poprzedników
- B. Rozwiązuje problemy znane jego twórcom
 - Tych nieznanych nie rozwiązuje
- C. Wprowadza nowe „zasady gry”
 - Generuje nowe problemy
- D. Nowe problemy są wyzwaniem dla kolejnego frameworka
- E. GOTO A

Model-Driven Development
 Architecture-Driven Development
 Behavior-Driven Development
 Test-Driven Development
 Use Case-Driven Design
 Data-Driven Development
 Use Ca
 Data-Dr
 * -Driven *
 Design
 lopment
 Use Case-Driven Development
 Data-Driven Design
 Design Use Case-Driven
 Data-Driven Development
 Architecture-Driven Design
 Architecture-Driven Development
 Domain-Driven Design
 Use Case-Driven Design

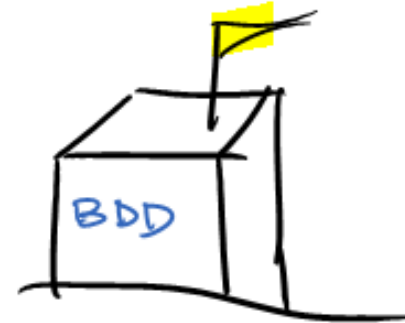
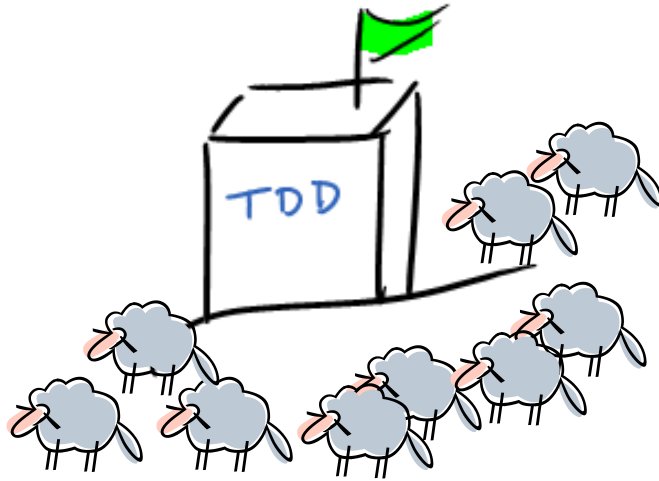
*-Driven * wzmagają zjawiska społeczne



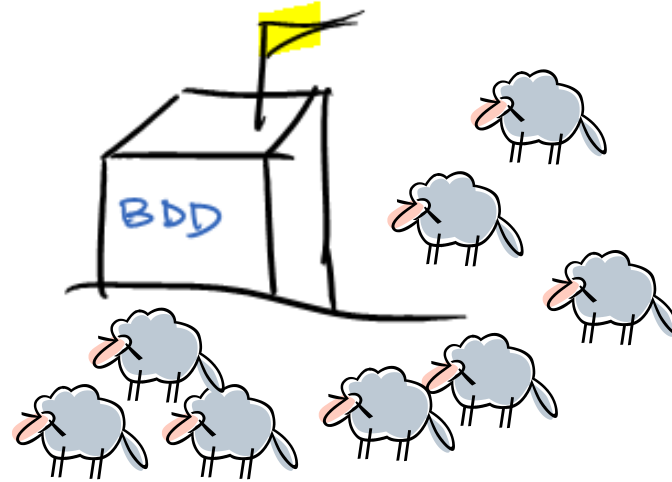
*-Driven * wzmagają zjawiska społeczne



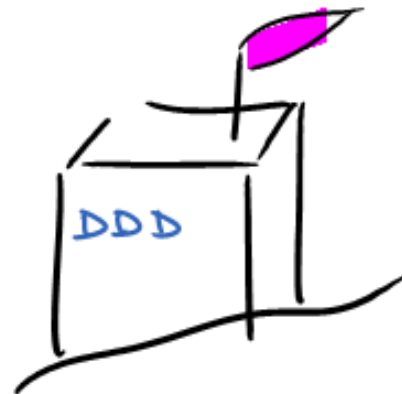
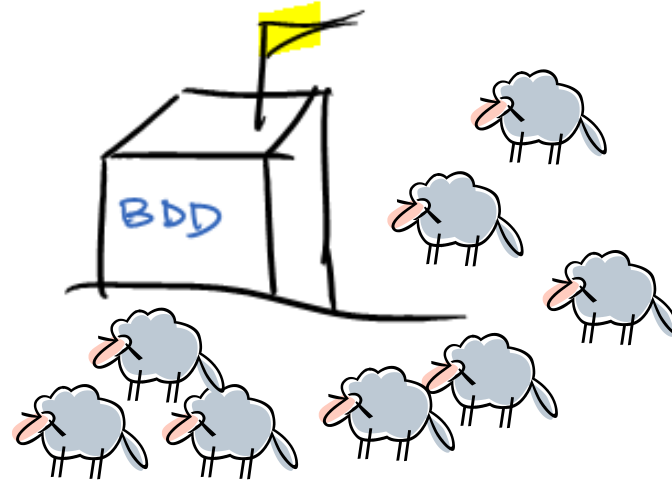
*-Driven * wzmagają zjawiska społeczne



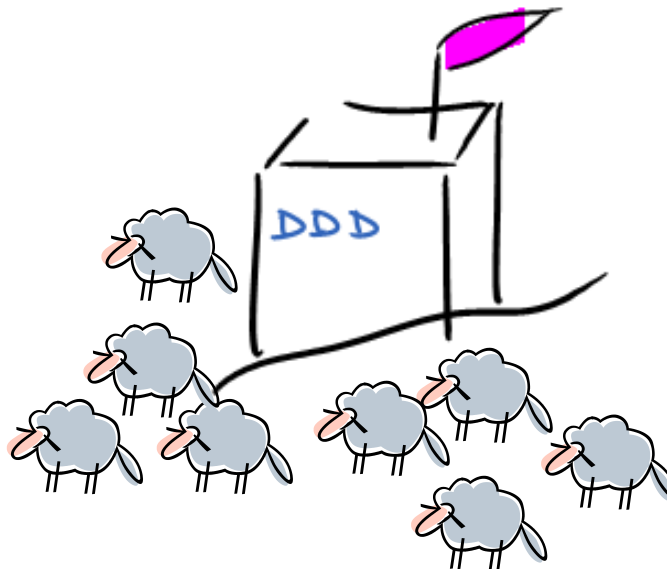
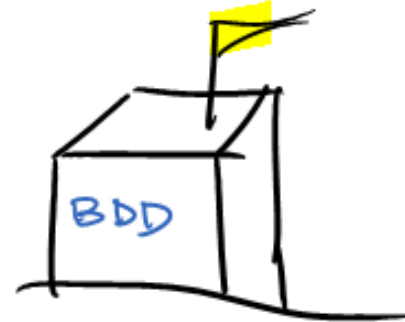
*-Driven * wzmagają zjawiska społeczne



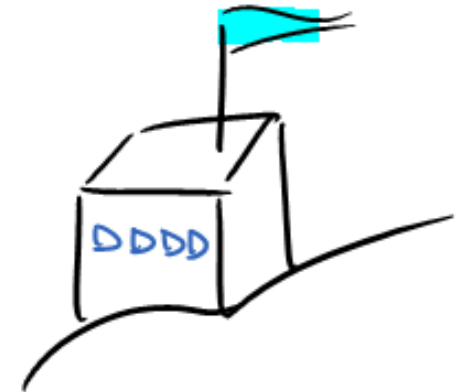
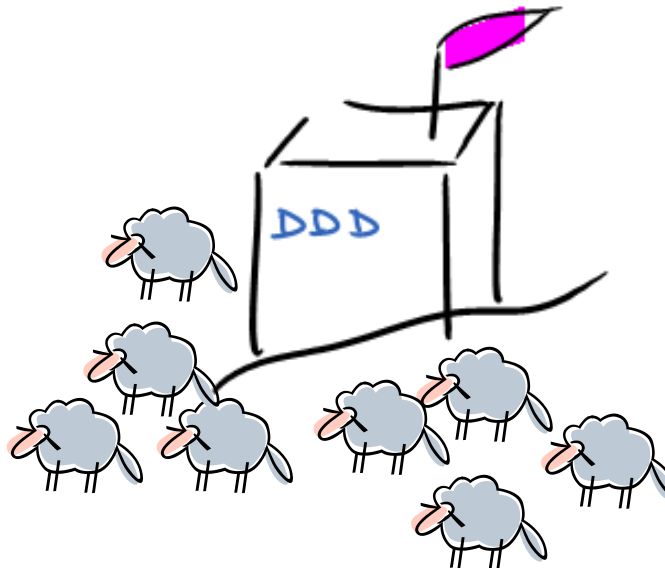
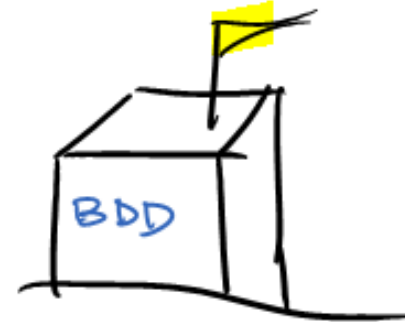
*-Driven * wzmagają zjawiska społeczne



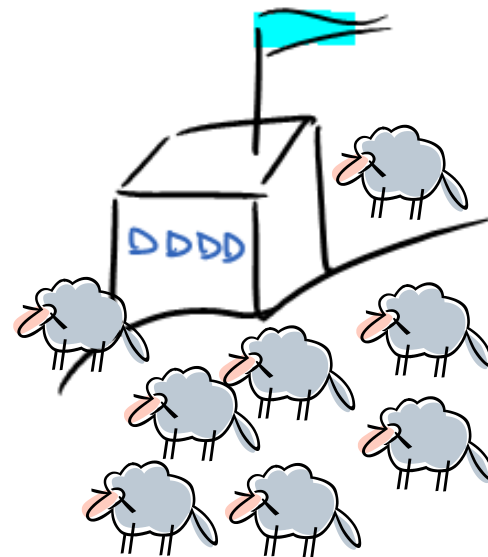
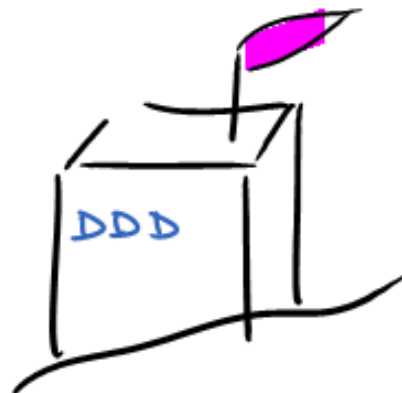
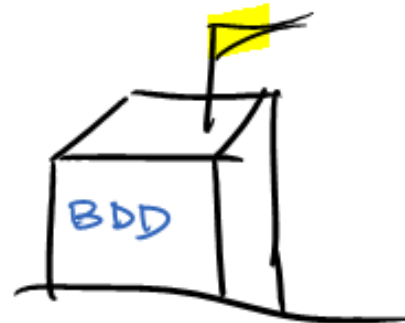
*-Driven * wzmagają zjawiska społeczne



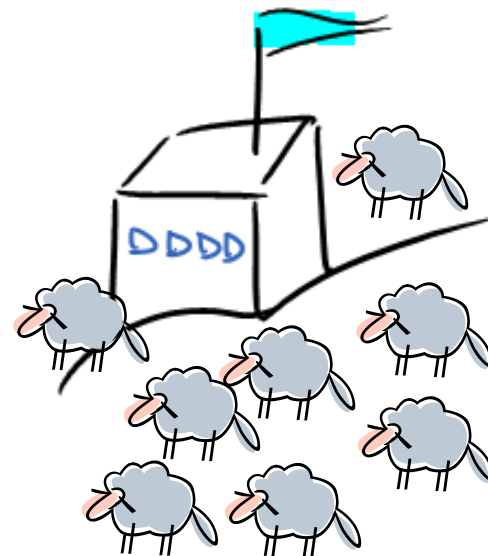
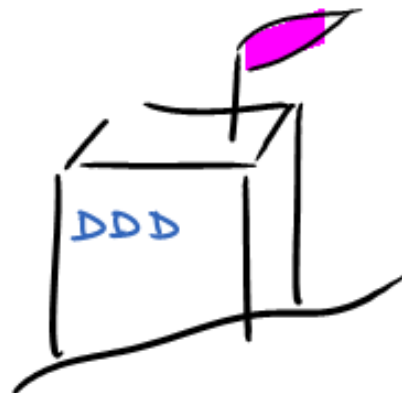
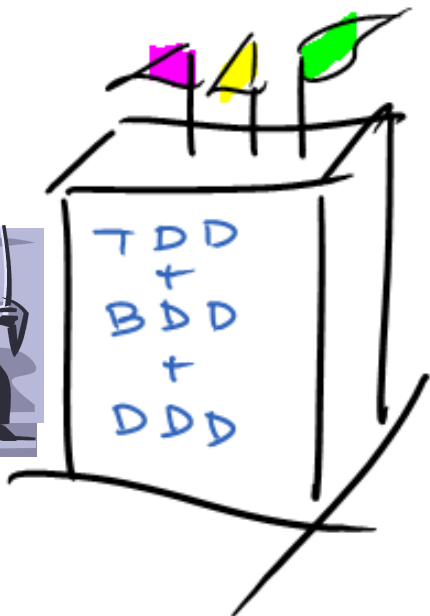
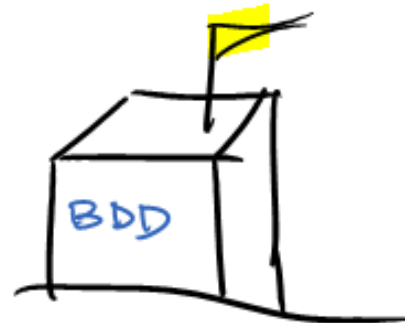
*-Driven * wzmagają zjawiska społeczne



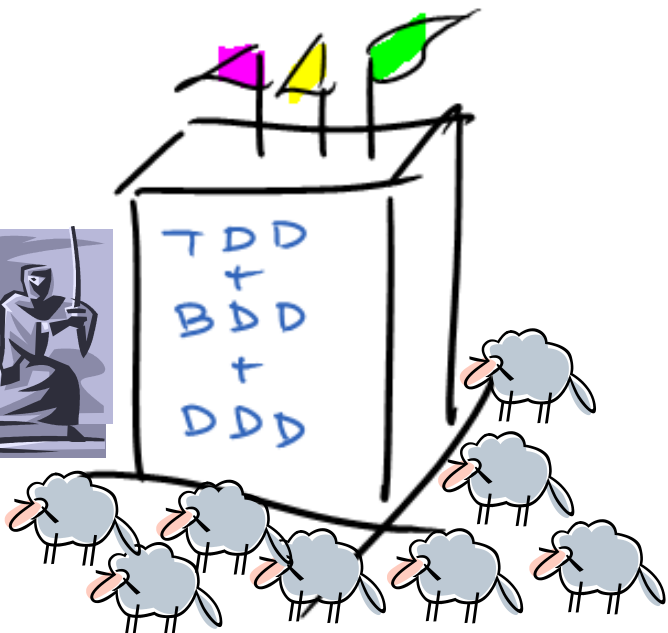
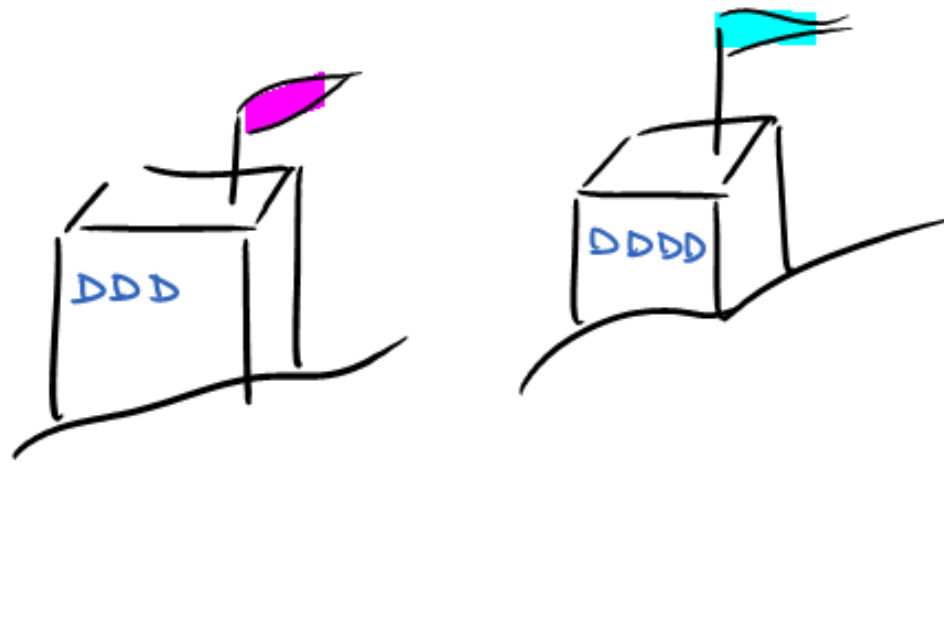
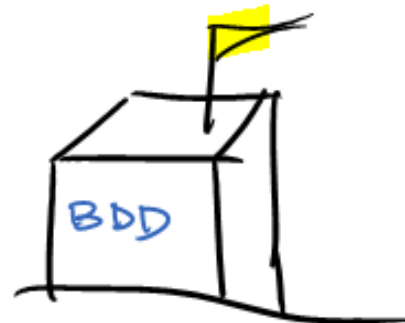
*-Driven * wzmagają zjawiska społeczne



*-Driven * wzmagają zjawiska społeczne



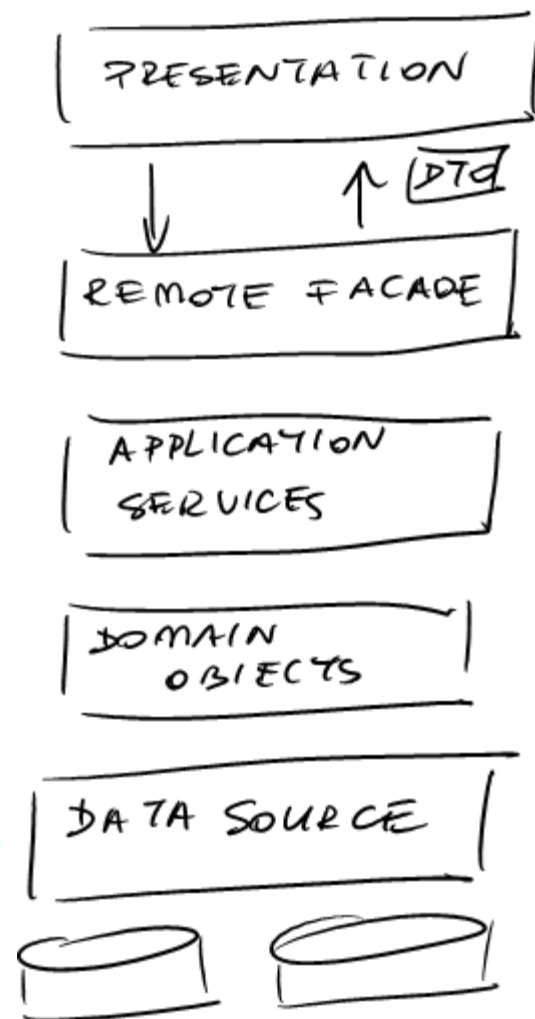
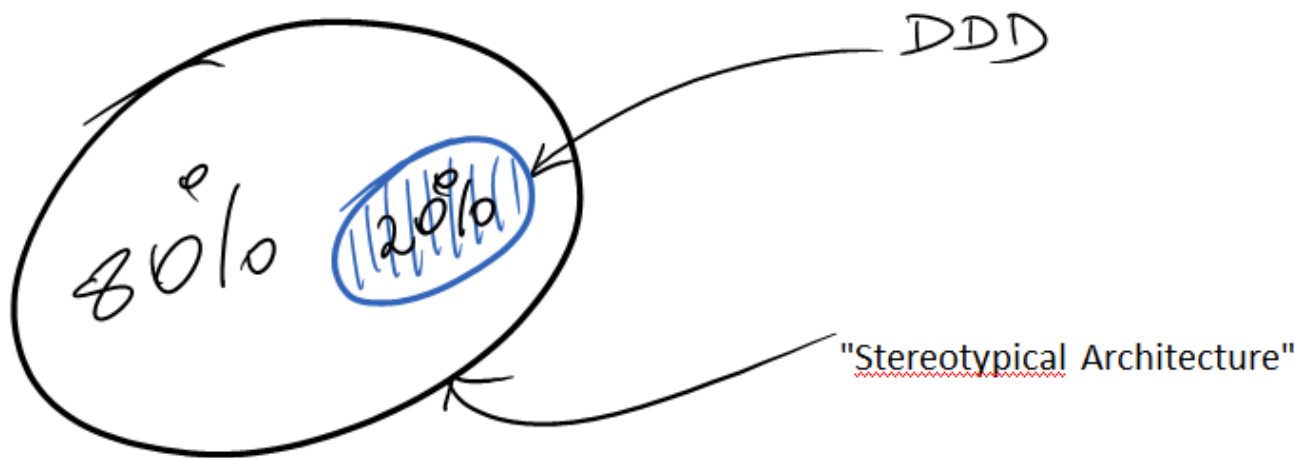
*-Driven * wzmagają zjawiska społeczne



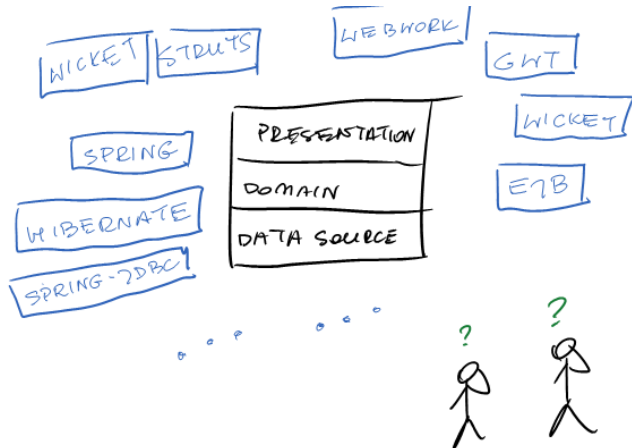
Przykład: DDDD by Greg Young

- <http://abdullin.com/storage/uploads/2010/04/2010-04-16-DDDD-Drafts-by-Greg-Young.pdf>

*„ (...) this architecture will be “good enough”
for 80% of the projects that they run into”*

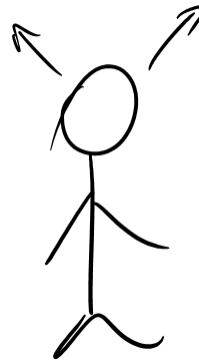


Programiści lubią frameworki



Model-Driven Development
 Architecture-Driven Development
 Behavior-Driven Development
 Test-Driven Development
 Use Case-Driven Design
 Data-Driven Development
 Use Case-Driven Design
 Data-Driven Development
 Use Case-Driven Design
 Data-Driven Development
 Use Case-Driven Design
 Data-Driven Development
 Architecture-Driven Design
 Architecture-Driven Development
 Domain-Driven Design
 Use Case-Driven Design

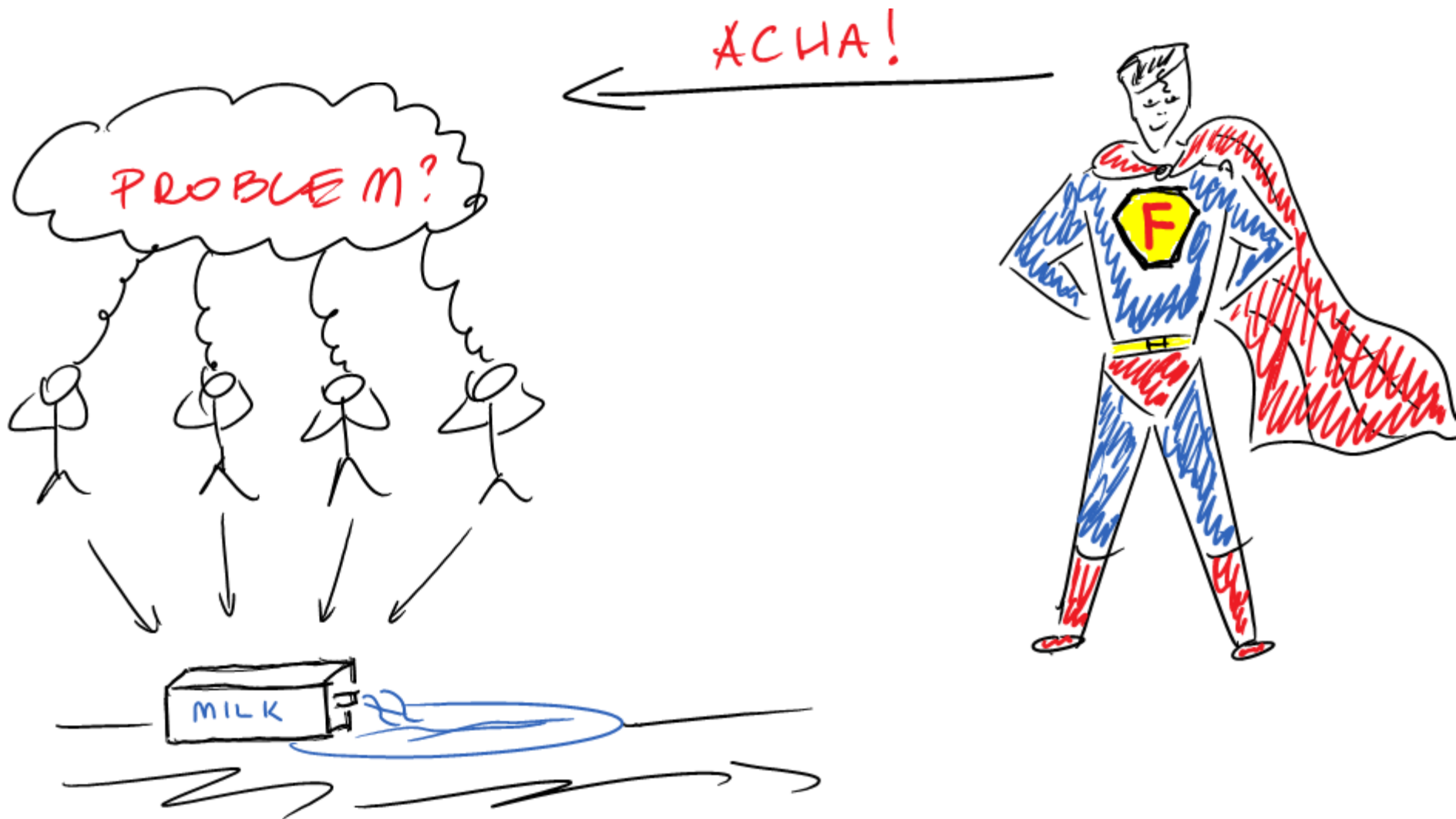
***-Driven ***



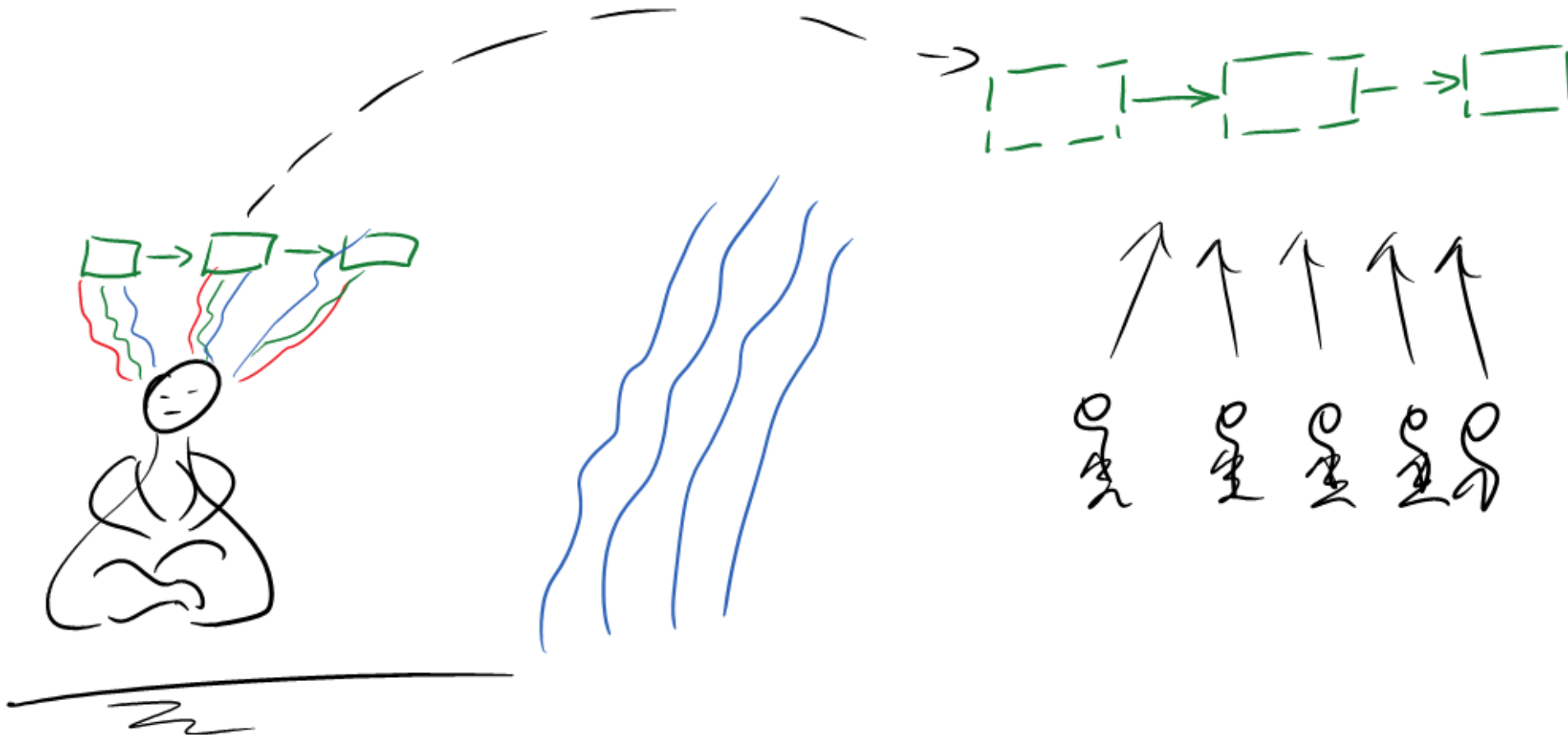


- # Strategie myślenia i postępowania
- # Uporządkowany sposób działania
- # Sposób na ...

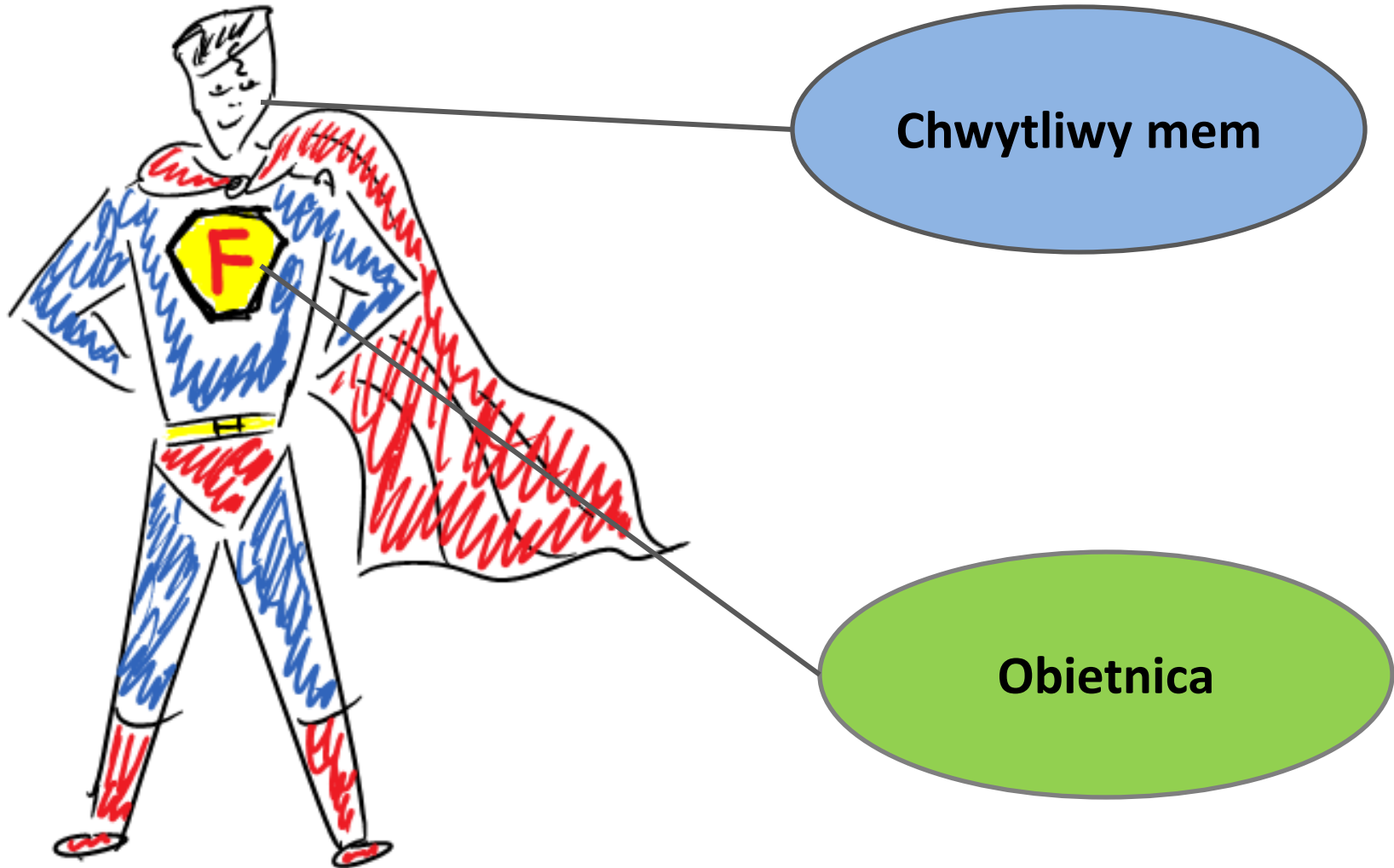
Dlaczego frameworki mentalne są popularne?



Framework odrywa się od intencji jego twórcy



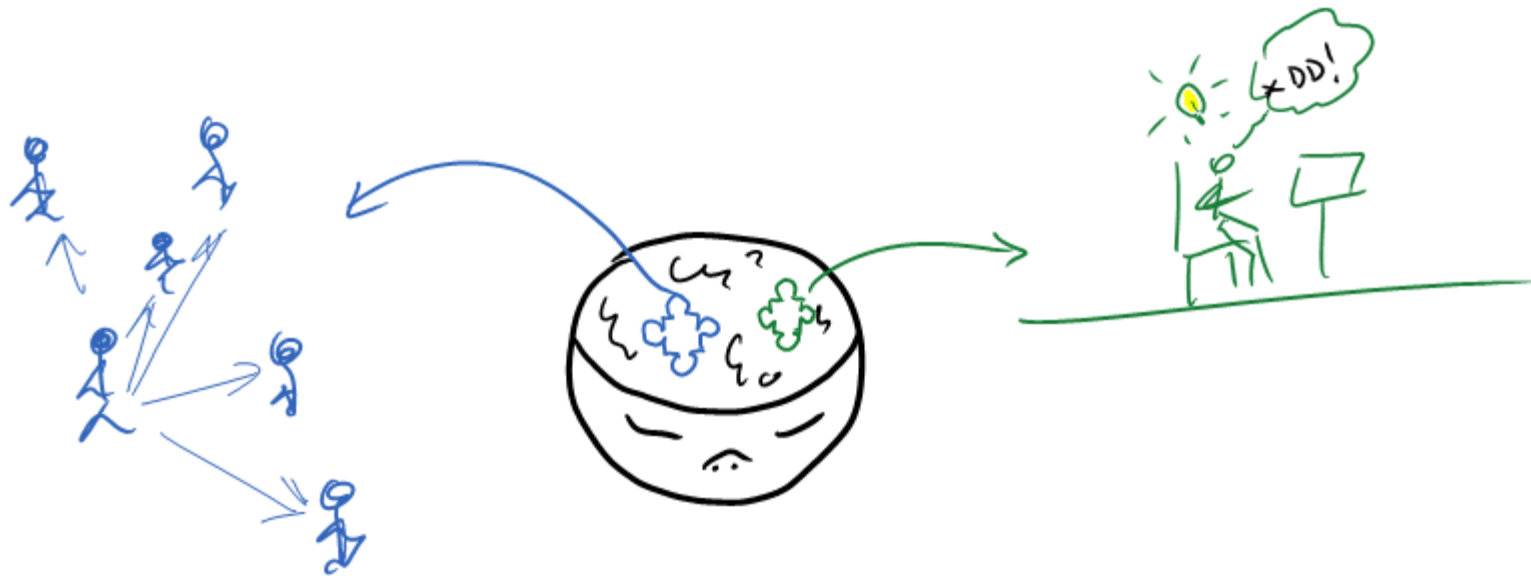
Klucz do powodzenia frameworka



Klucz do powodzenia frameworka

Chwytliwy mem

Obietnica



Obiecuje Ci, że:

- błędy wykryjesz na wczesnym etapie
- stworzysz tylko tyle kodu ile to konieczne
- dobry design
- prawie zapomnisz o debuggerze

Chwytliwe memy:

- Red-Green-Refactor
- Unit testing

Obiecuje Ci, że:

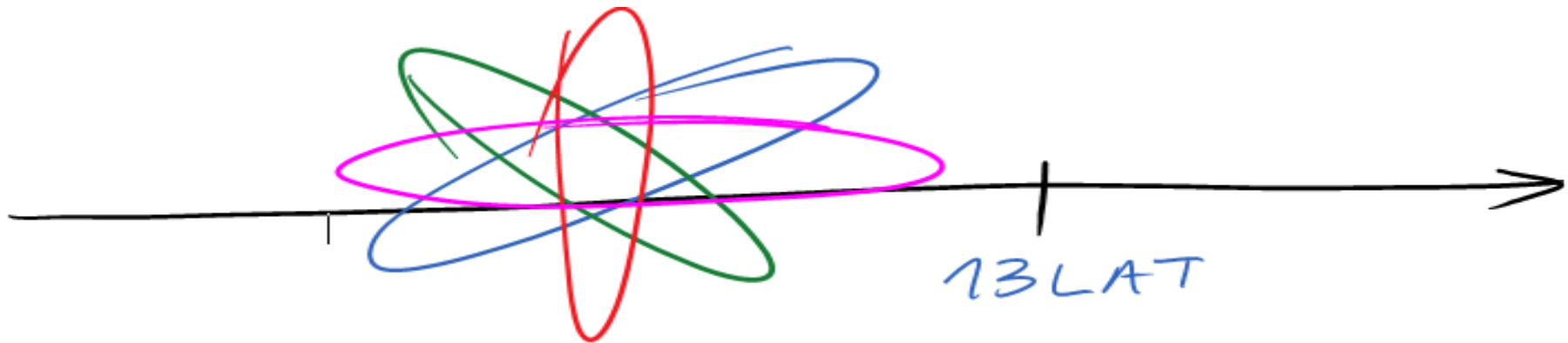
- we **właściwy** sposób zastosujesz programowanie obiektowe
- elastyczną architekturę
- odwzorowanie rzeczywistości => przygotowanie na kolejne wymagania

Chwytliwe memy:

- Building Blocks
- Unambiguous Language
- Strategic Design

Kto wymyśla *-Driven *?

Kto wymyśla *-Driven *?

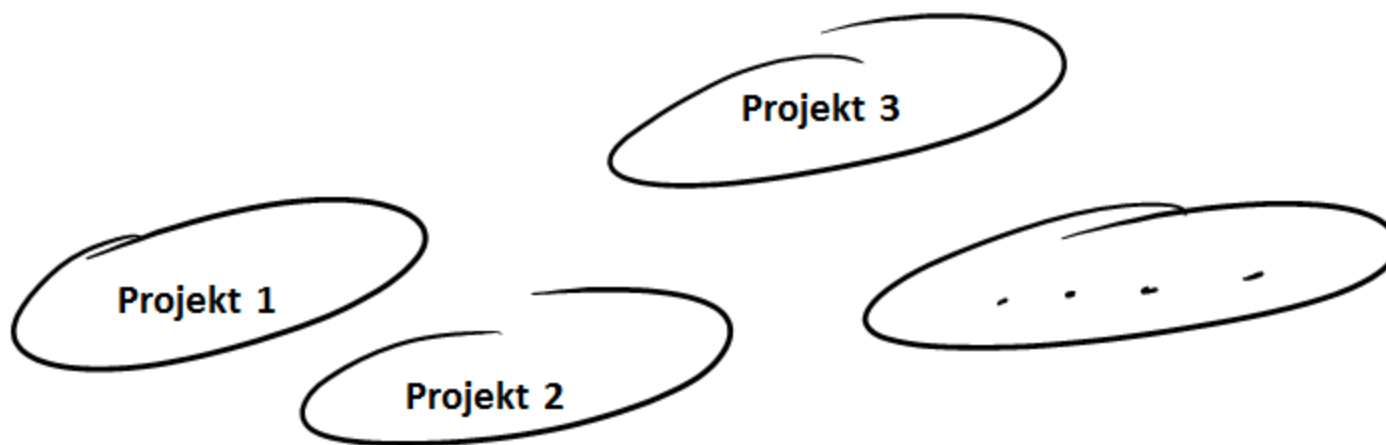


<http://www.domainlanguage.com>

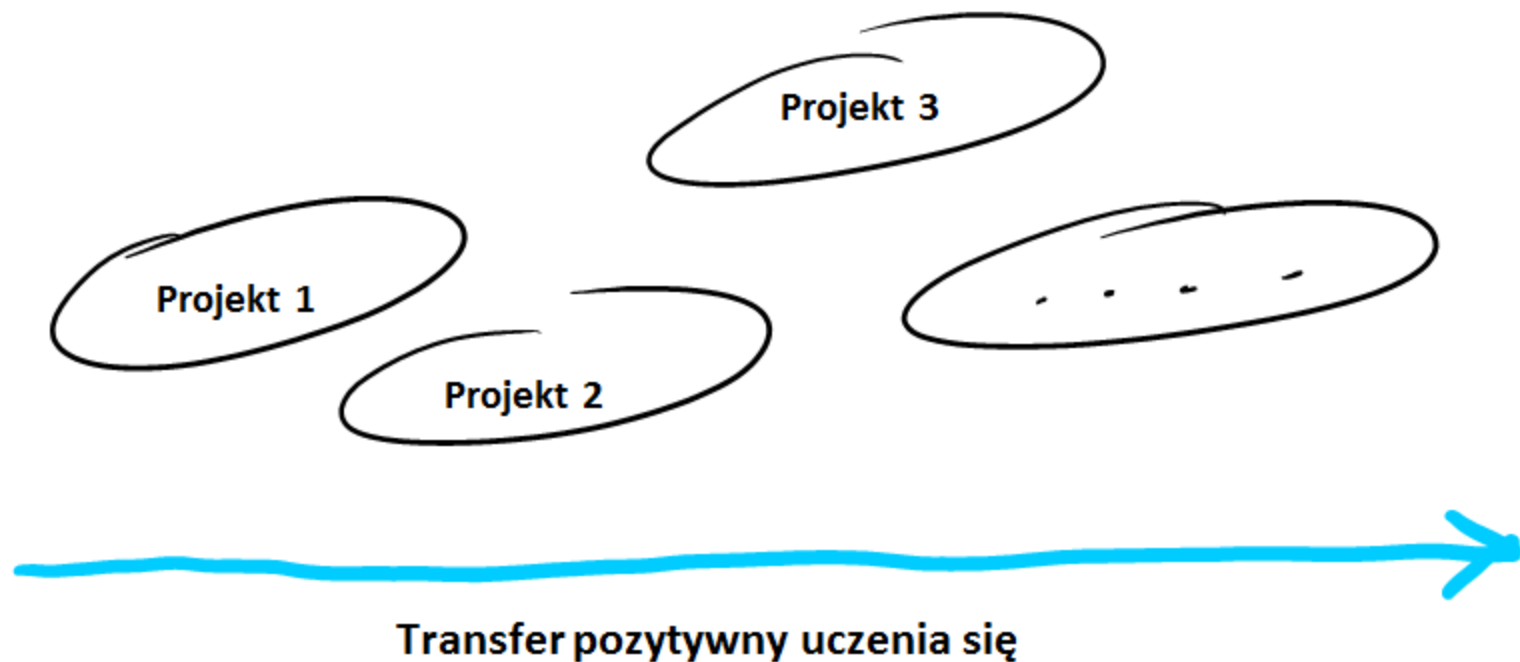
<http://programmer.97things.oreilly.com/wiki/>

http://pl.wikipedia.org/wiki/Kent_Beck

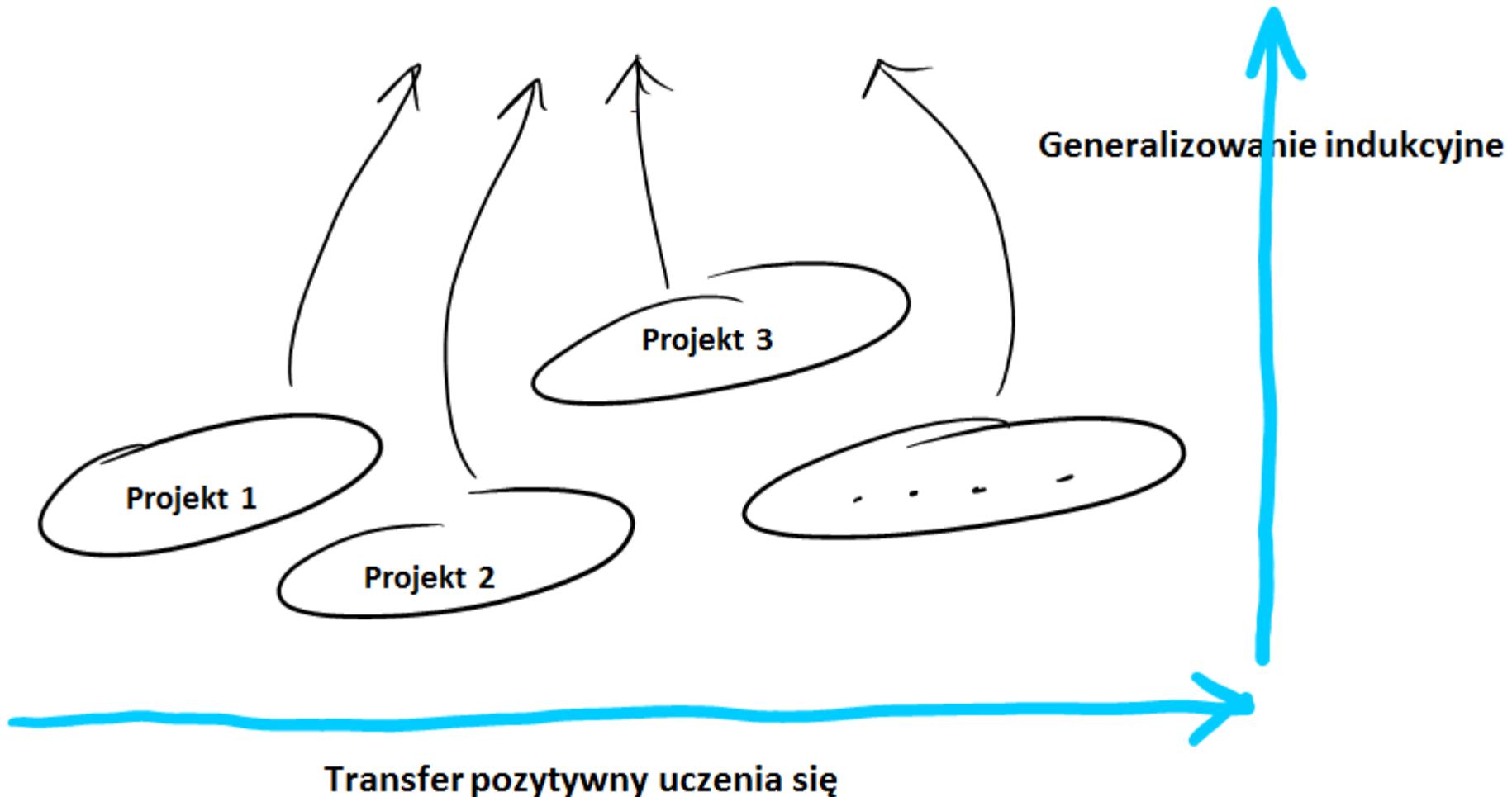
Jak powstaje framework mentalny?



Jak powstaje framework mentalny?

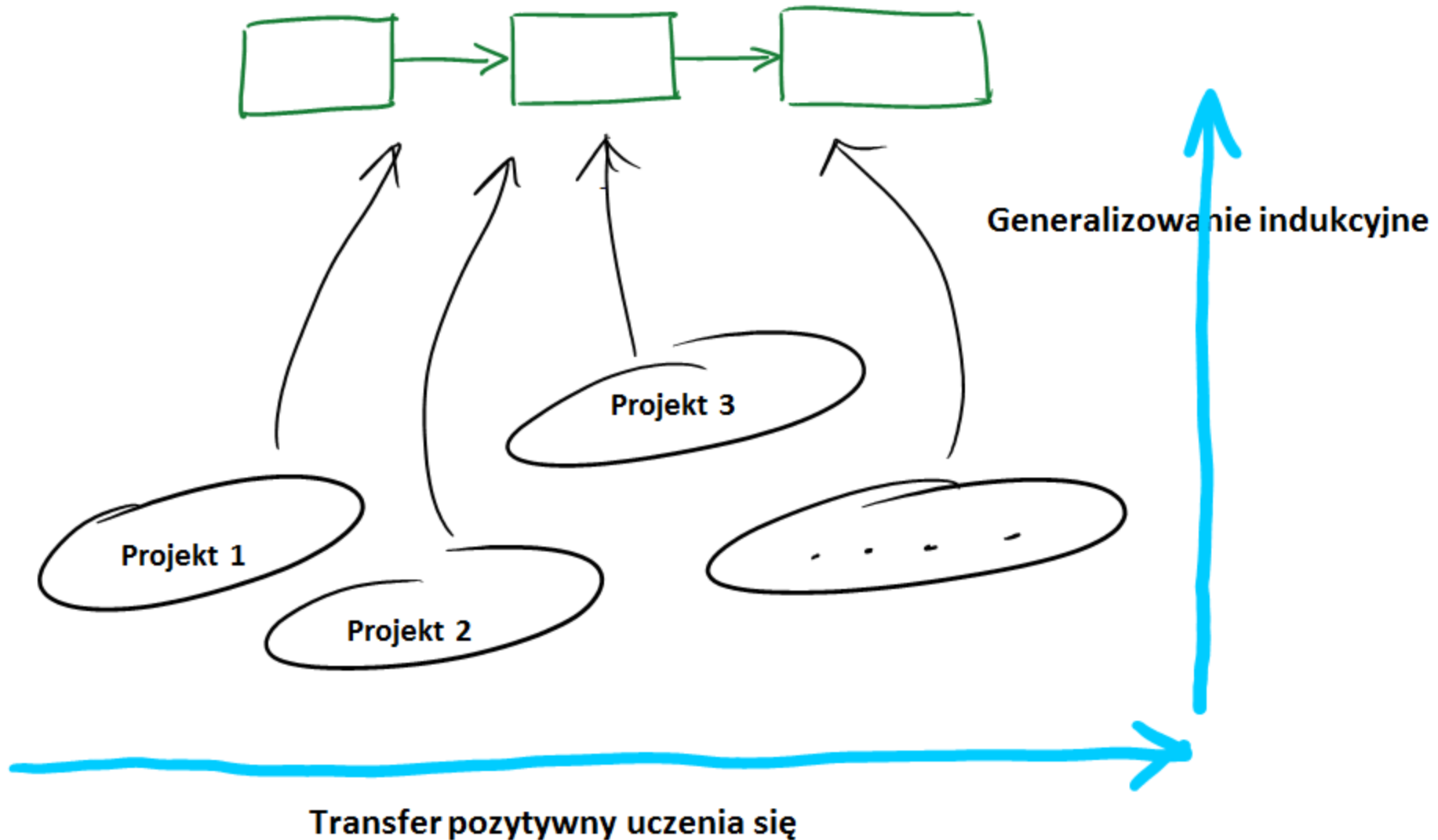


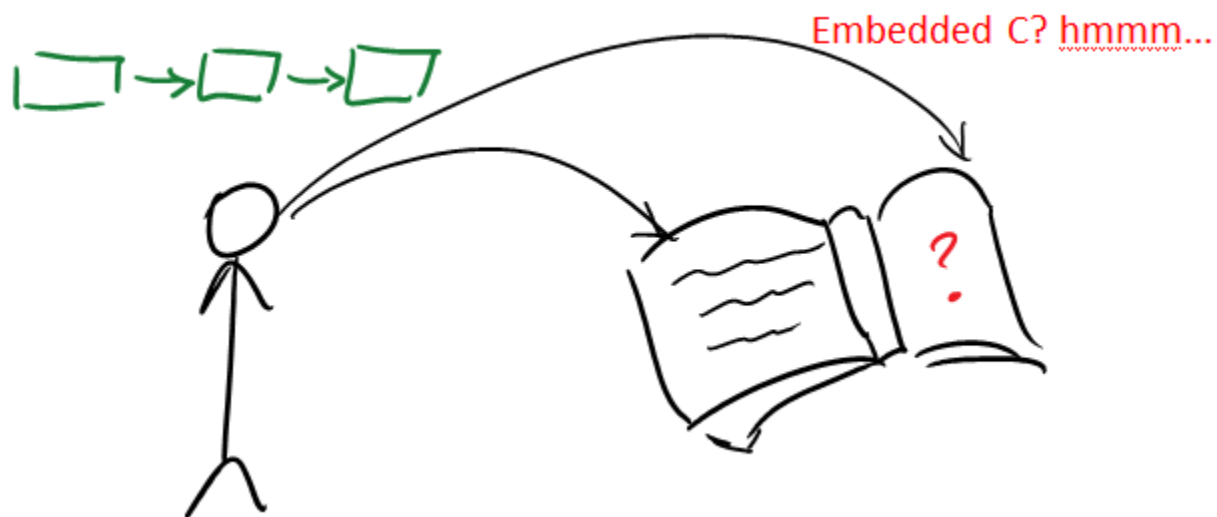
Jak powstaje framework mentalny?

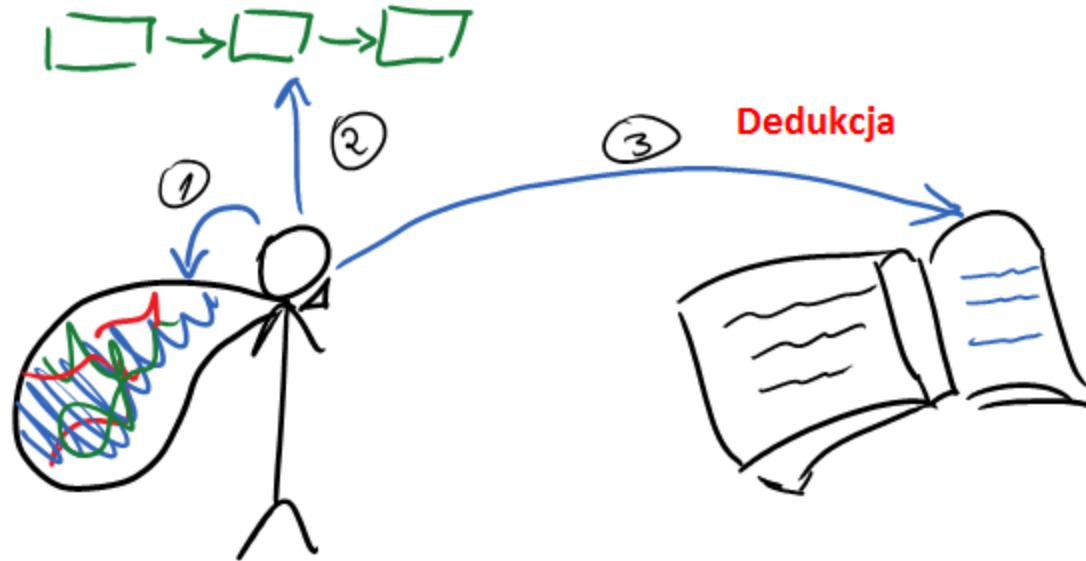


Jak powstaje framework mentalny?

Framework mentalny





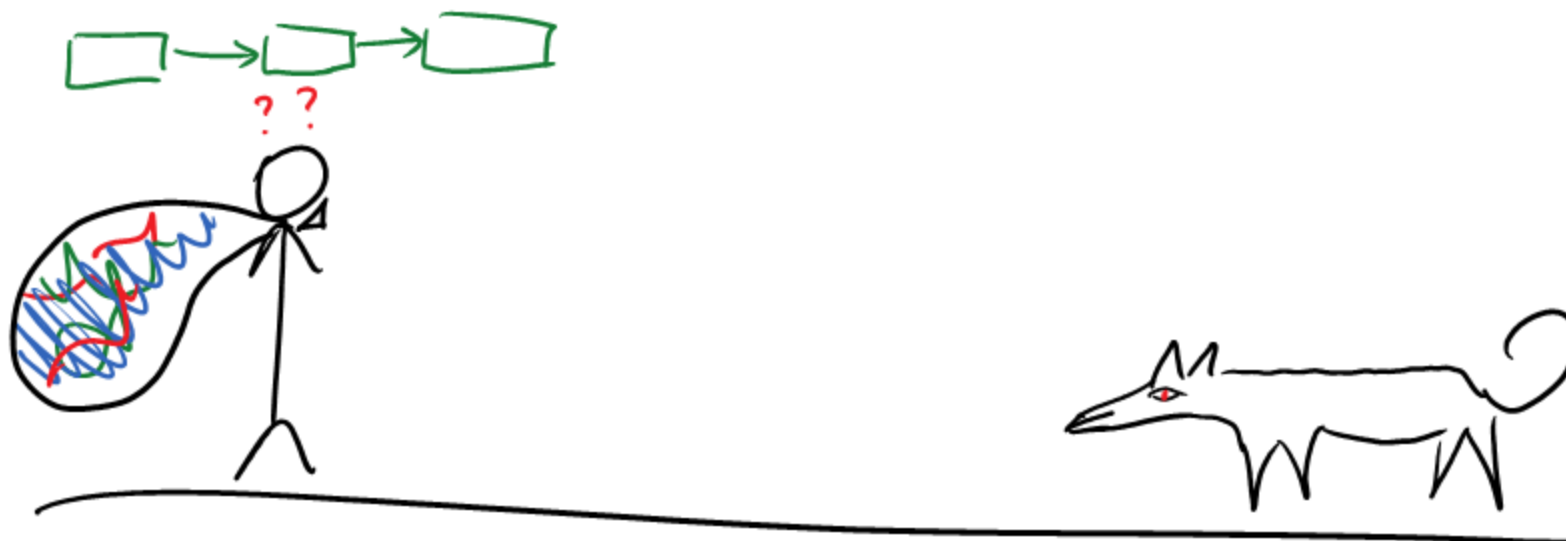


- # Wyprowadzanie logicznych wniosków
- # Niekoniecznie sprawdzonych w rzeczywistości
- # Wnioski można weryfikować, ale...czas

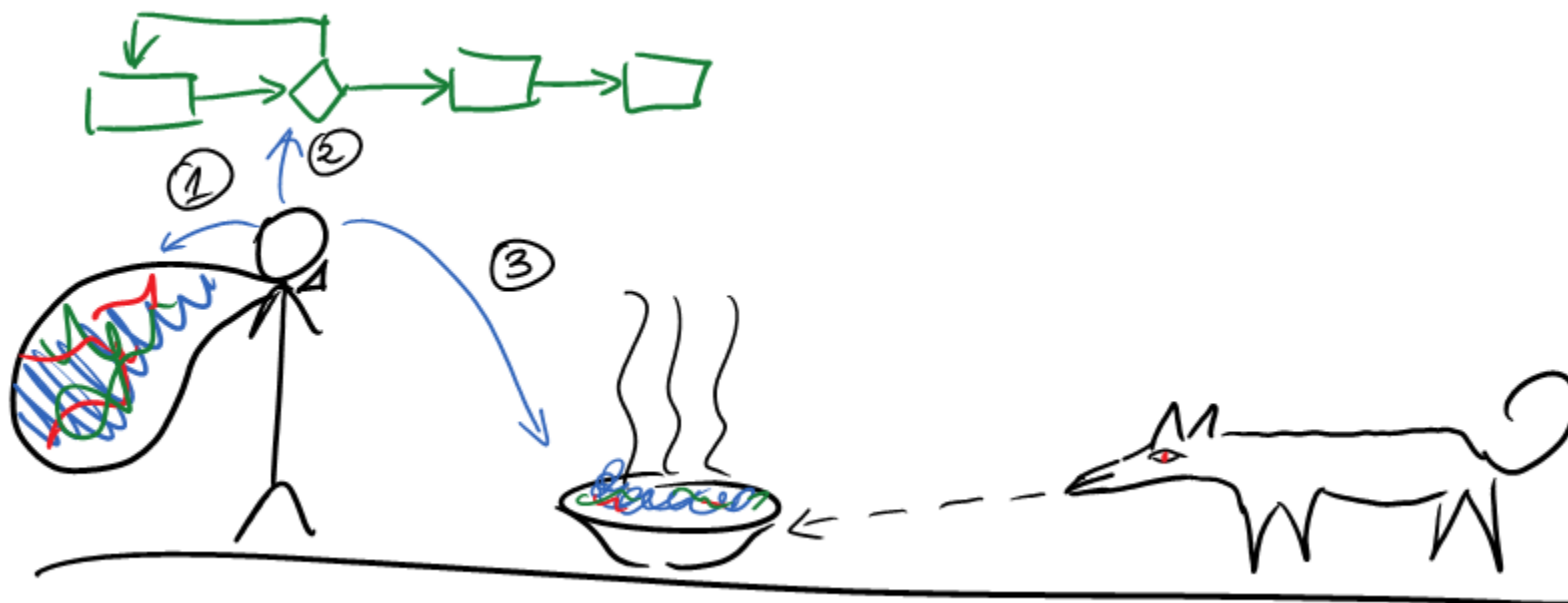
What I've learnt about DDD since the book

- www.infoq.com/presentations/ddd-eric-evans
- Podkreślenie roli Bounded Context
- Doprecyzowanie znaczenia Domain Event
- Doprecyzowanie relacji między Bounded Contexts
- Krytyka Context Map
- Proces Model Exploration Whirpool

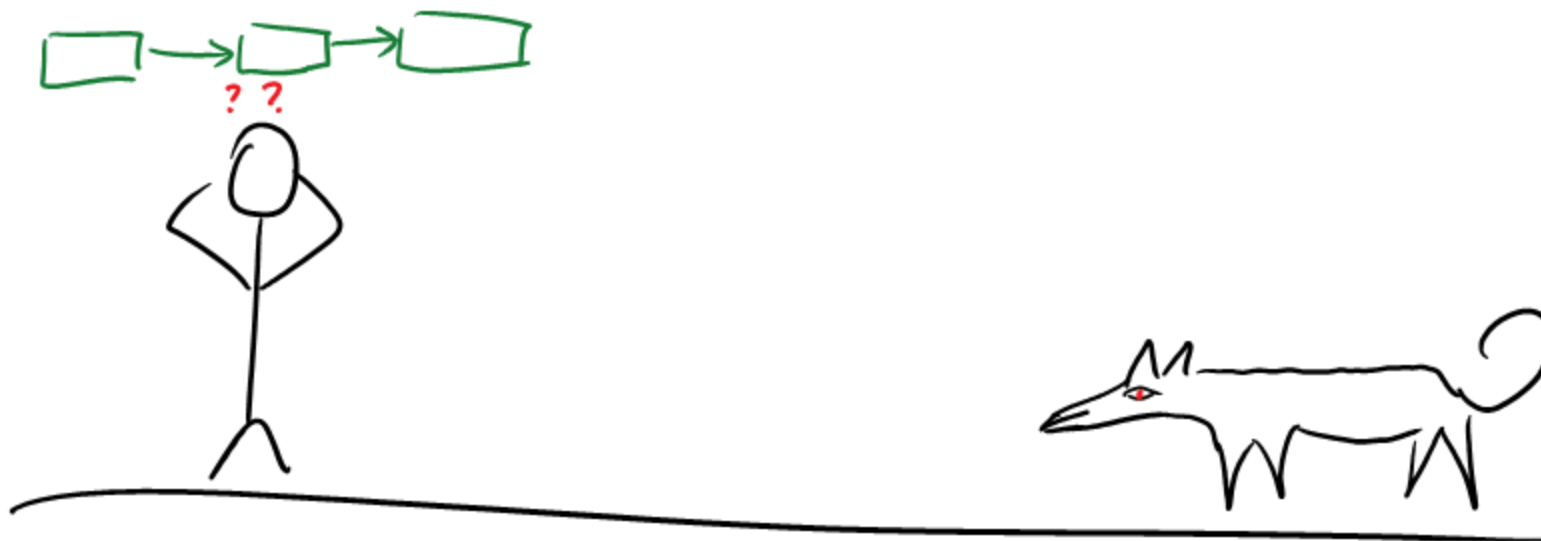
Co robi Twórca, gdy framework nie działa?



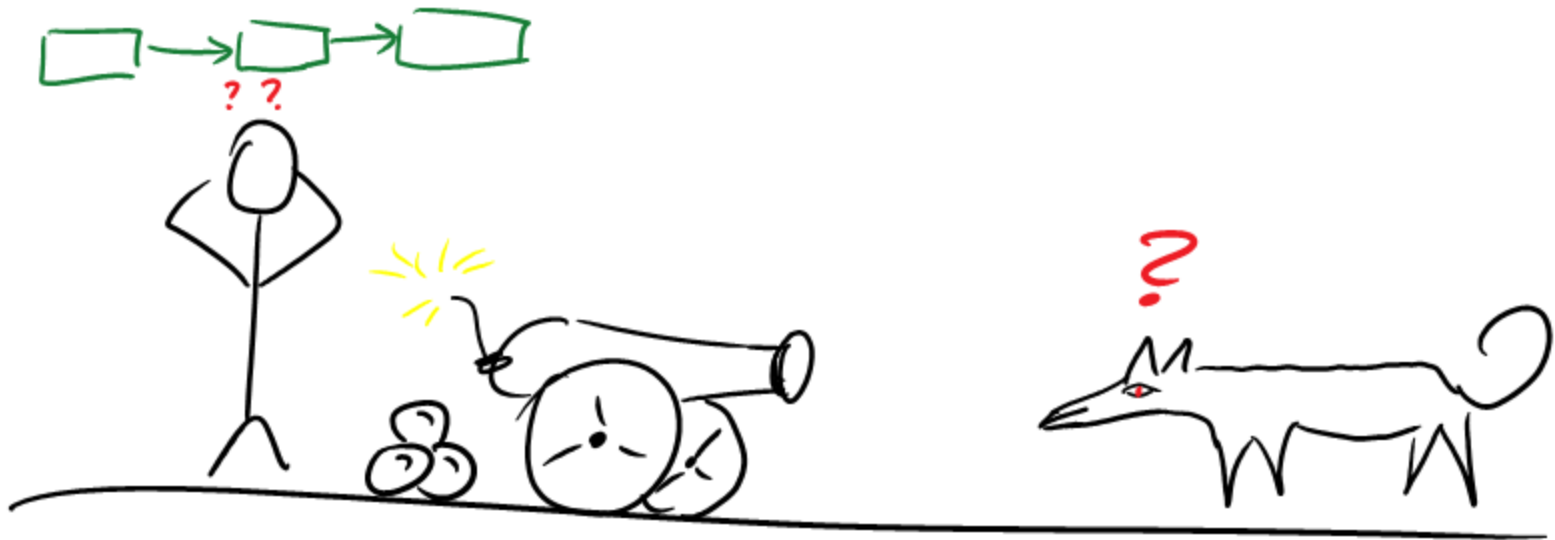
Co robi Twórca, gdy framework nie działa?



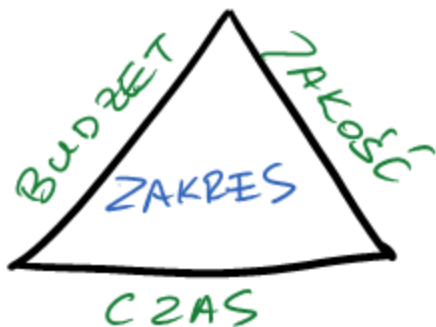
Co zrobi programista z małym doświadczeniem?



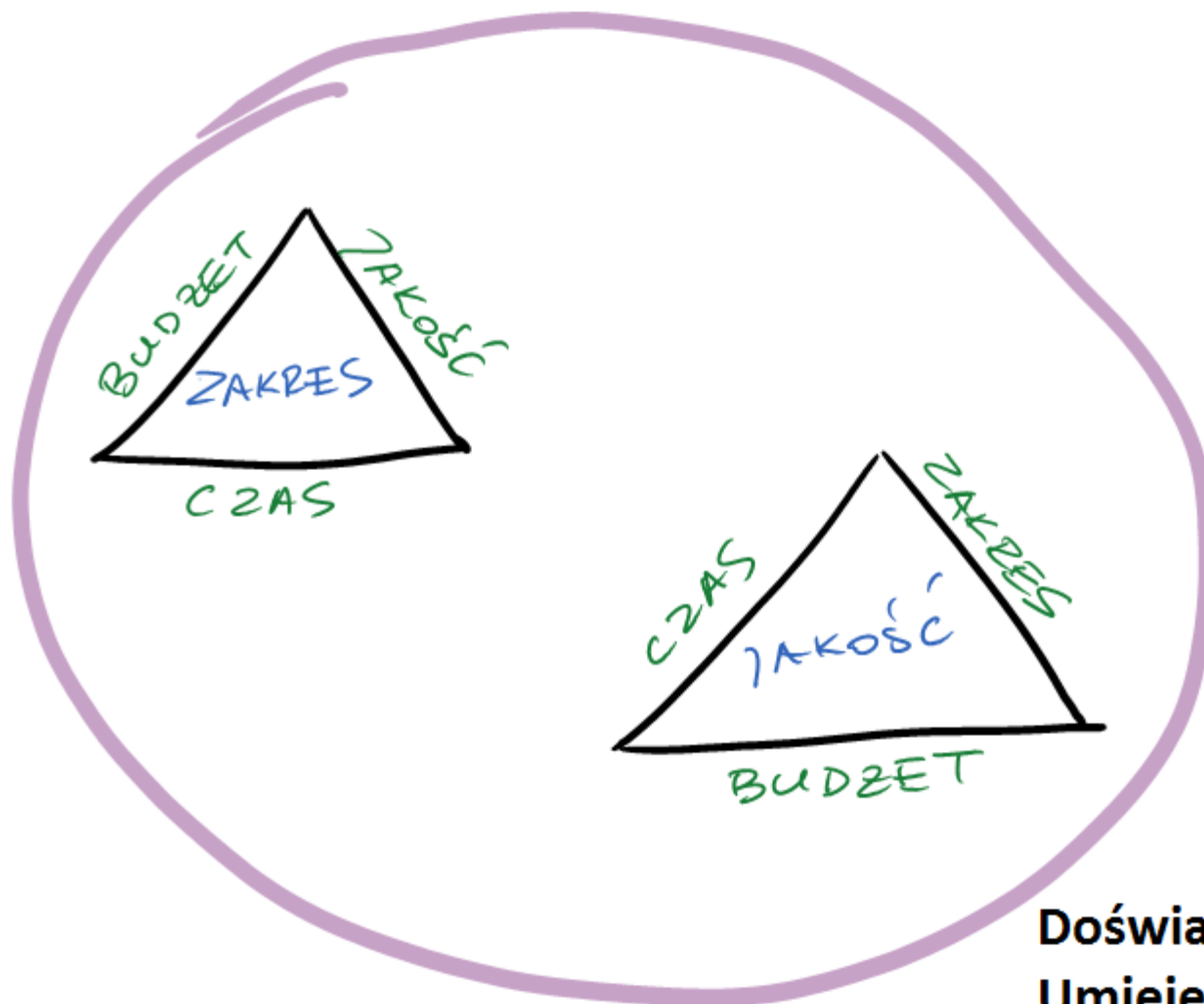
Co robi programista z małym doświadczeniem?



Ograniczenia w zespołach



Ograniczenia w zespołach



**Doświadczenie
Umiejętności**

Pytania, które zadajemy zbyt rzadko?

- # *Czym mam wystarczające doświadczenie i umiejętności, aby zastosować *-Driven?*
- # *Czy zastosowanie *-Driven * jest dla mnie zbyt trudne?*
- # *Ile będzie kosztować moja nauka?*

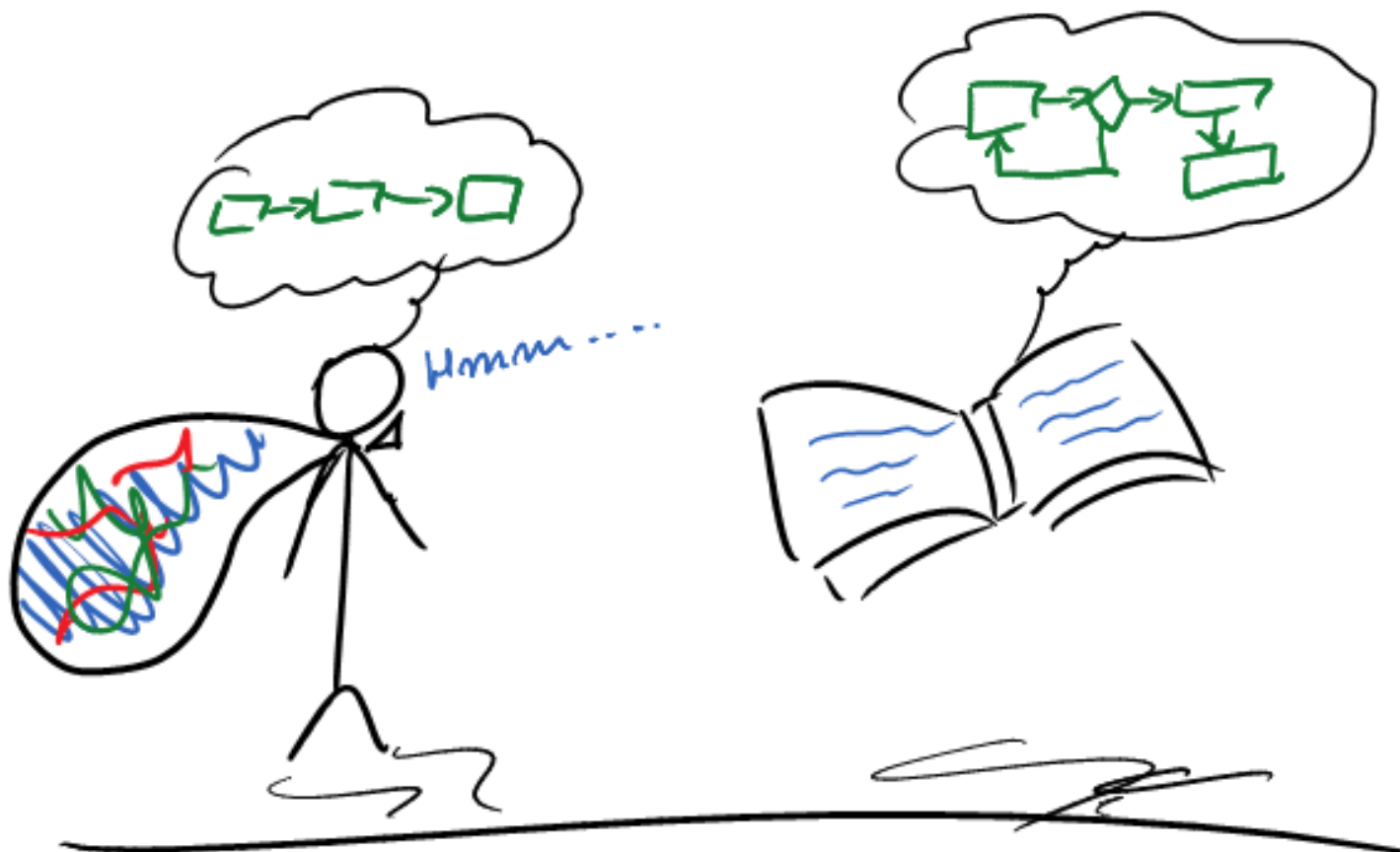
**A może oceniasz się zbyt
optymistycznie?**

Na czym wykładamy się najczęściej?



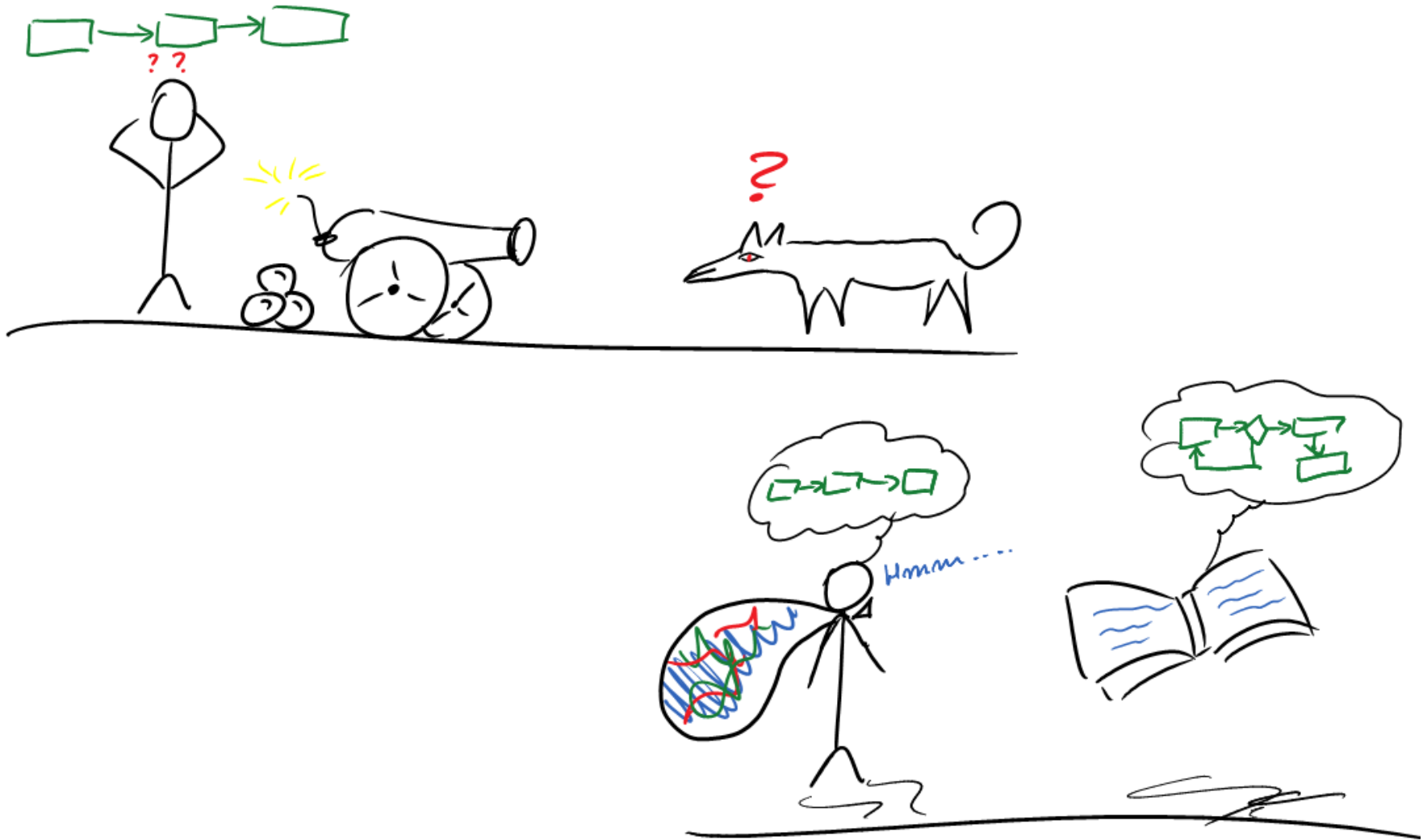
KONTEKST

Co robi doświadczony programista?



*Implementing *-Driven **

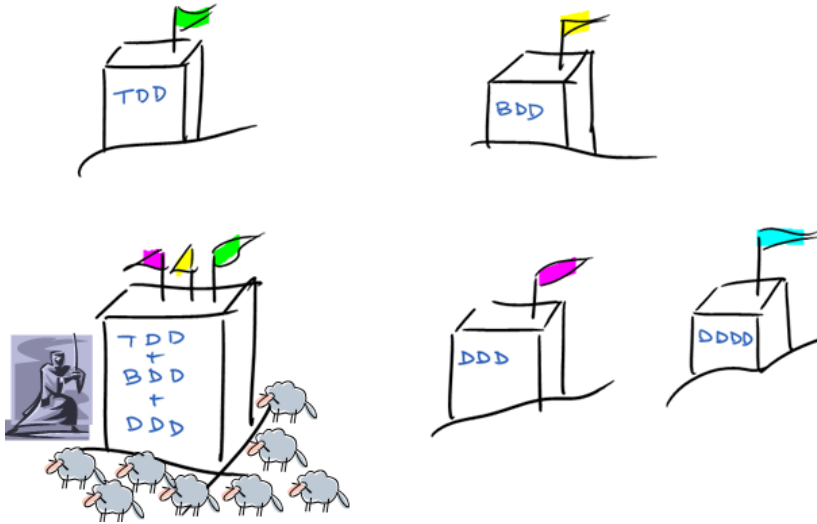
Dlaczego *-Driven * niczego nie zmienia?



**Czy to oznacza, że zniechęcam
do
TDD, BDD, DDD, DDDDD, etc. ?**

Czy to oznacza, że zniechęcam
do
TDD, BDD, DDD, DDDDD, etc. ?

- 1. Buduj swoje doświadczenie**
- 2. Ułatwiaj sobie pracę za pomocą frameworków mentalnych**



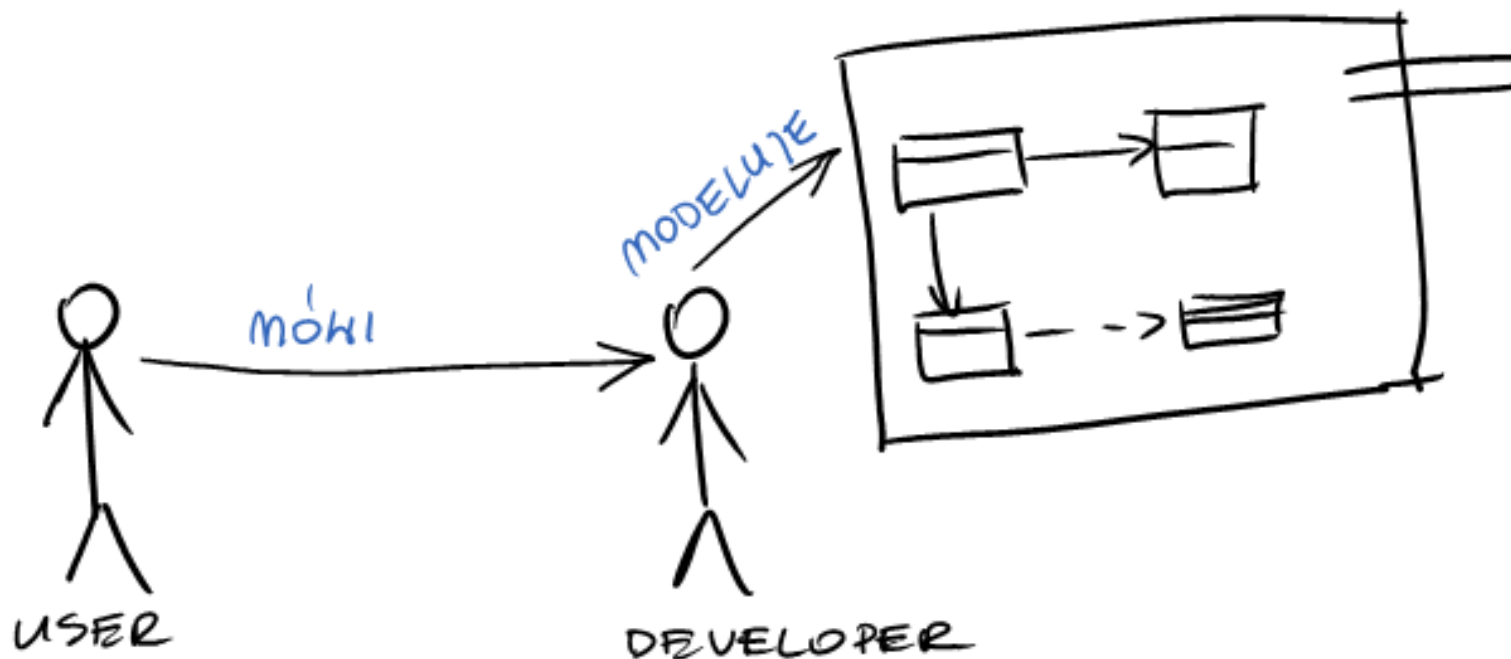
1. Frameworki mentalne
2. Doświadczenie



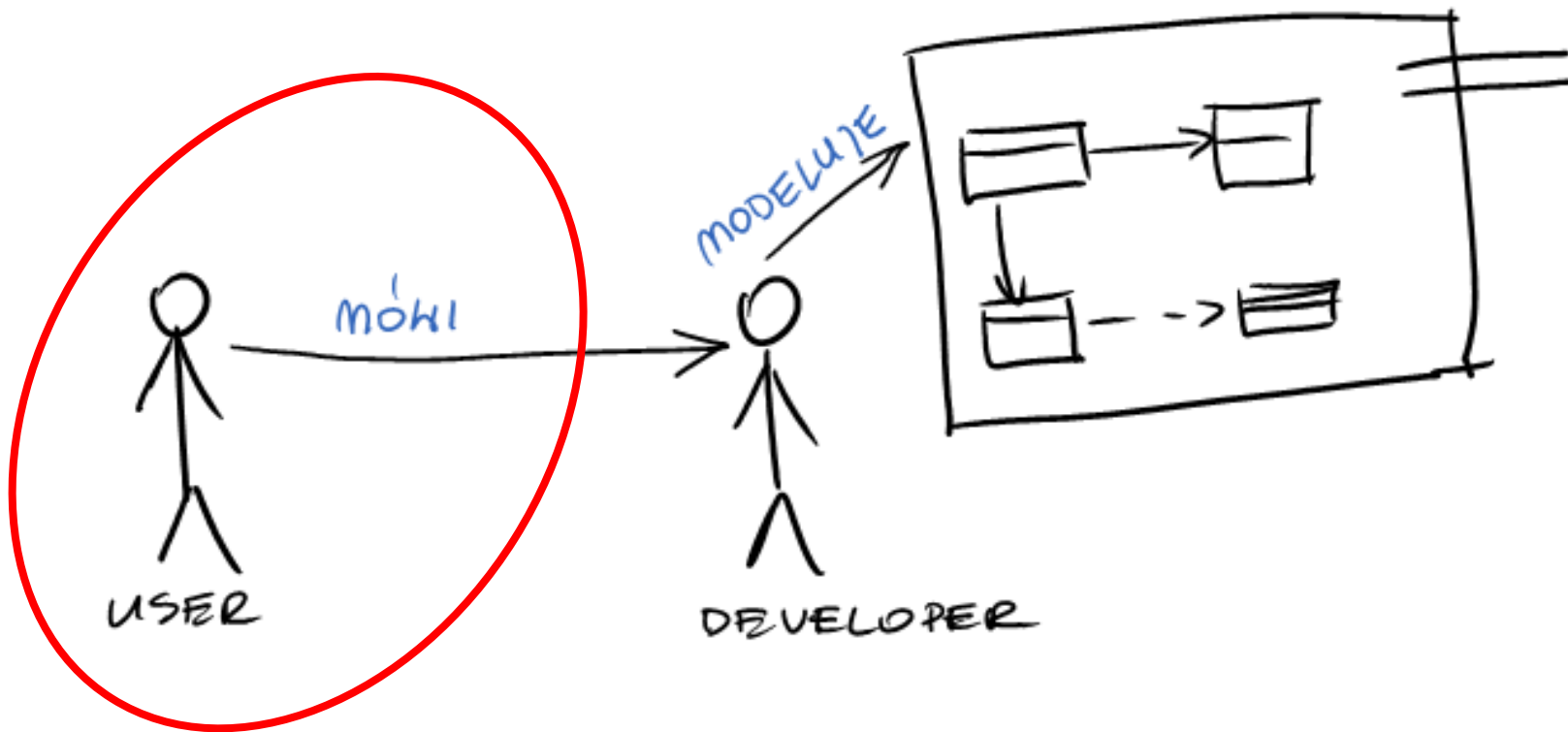
1. Doświadczenie
2. Frameworki mentalne

Na czym się skoncentrować i jak budować doświadczenie?

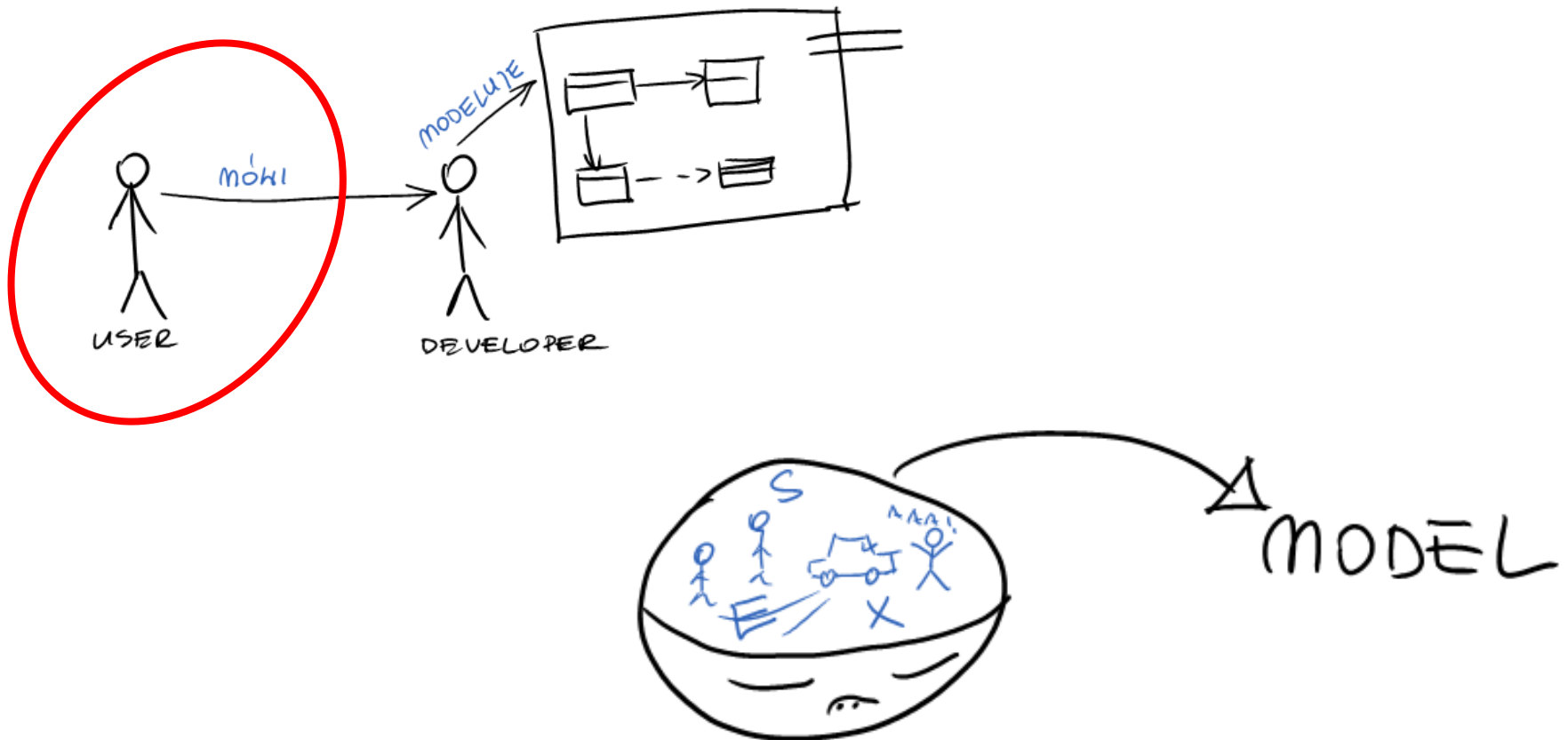
Przykład: Modelowanie „rzeczywistości”



Przykład: Modelowanie „rzeczywistości”



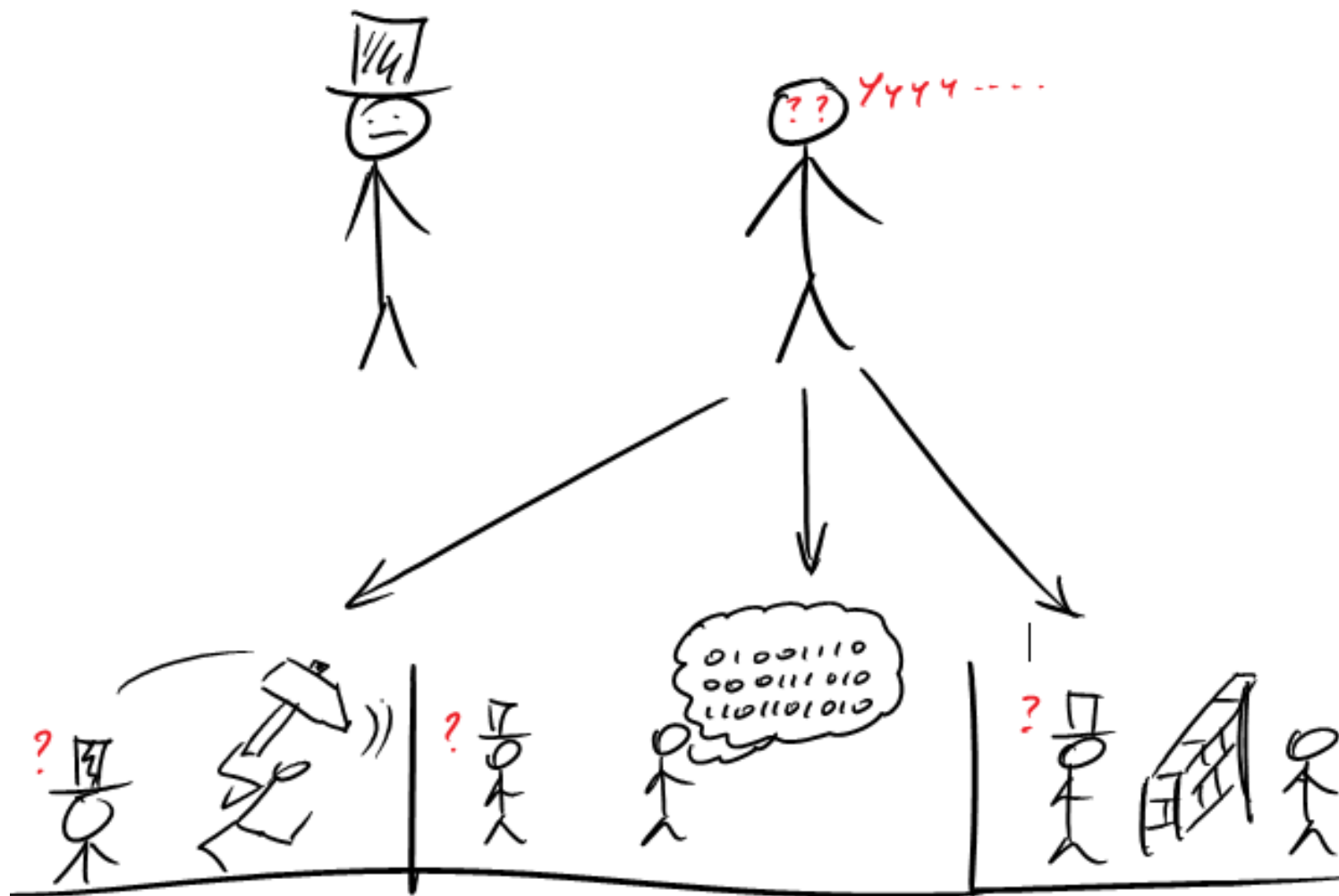
Przykład: Modelowanie „rzeczywistości”



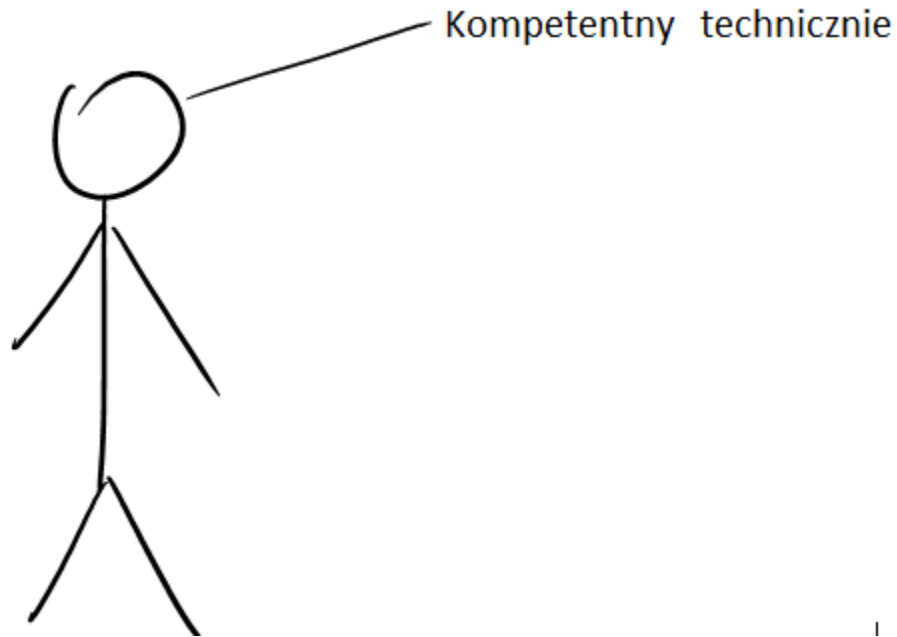
Wie, czy nie wie, czego chce?



Fight, Flight, Freeze

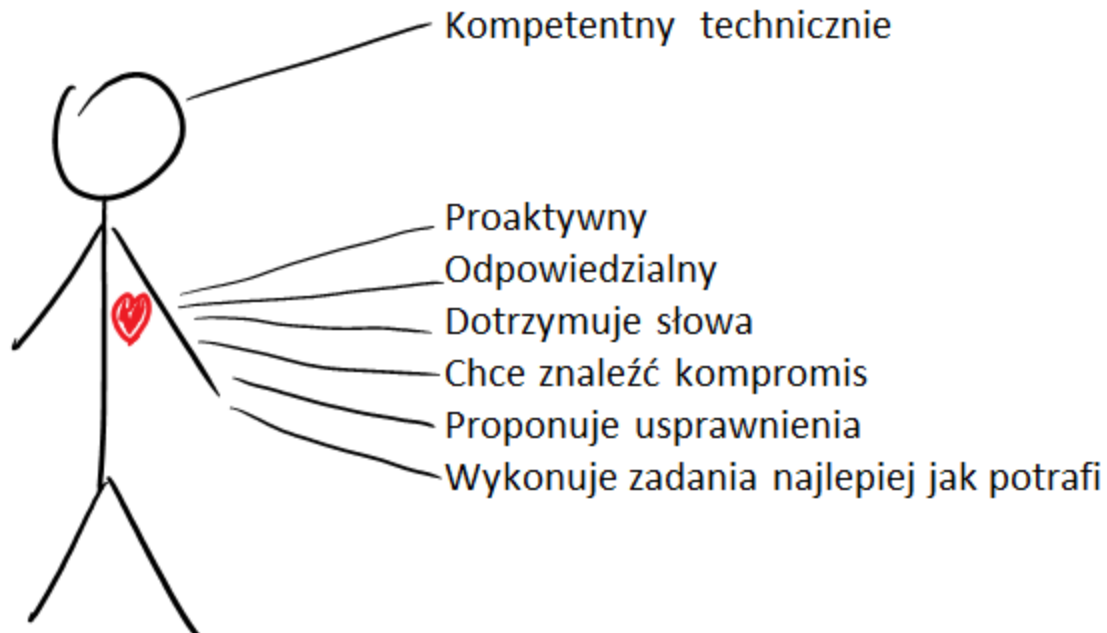


Z jakimi programistami chcą pracować klienci? Jakich programistów cenią liderzy?



I

Z jakimi programistami chcą pracować klienci? Jakich programistów cenią liderzy?



Z jakimi programistami chcą pracować klienci? Jakich programistów cenią liderzy?



Zaangażowany := robi wszystko co konieczne,
aby doprowadzić sprawy do końca

Czy zatrudnisz programistę?

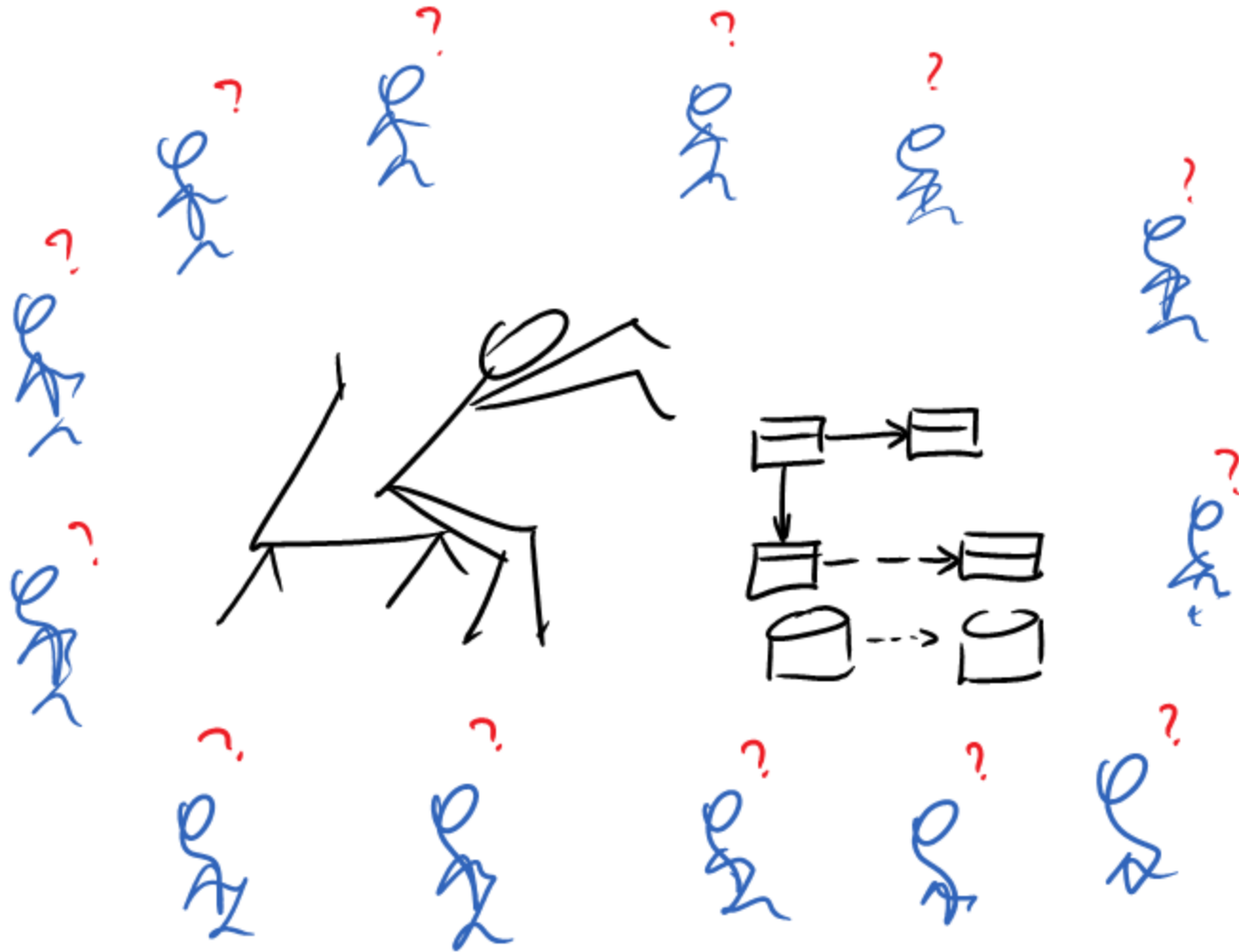


- i niewielkie umiejętności techniczne
- Raczej tak

duże umiejętności techniczne bez



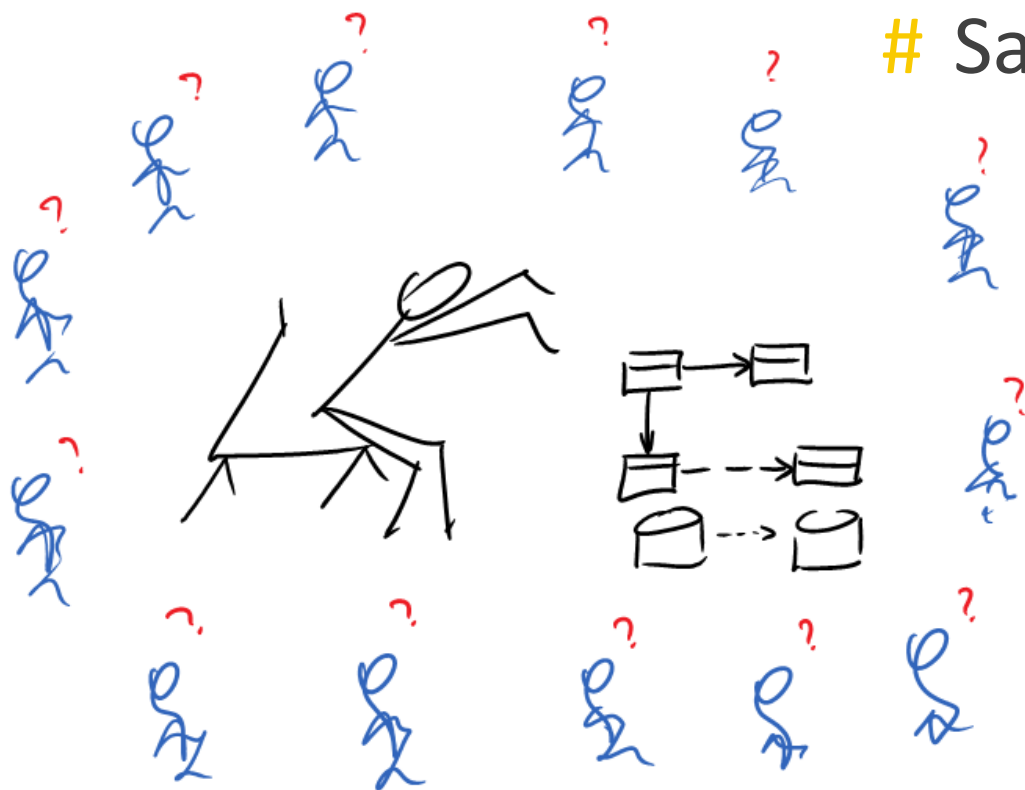
- Raczej nie



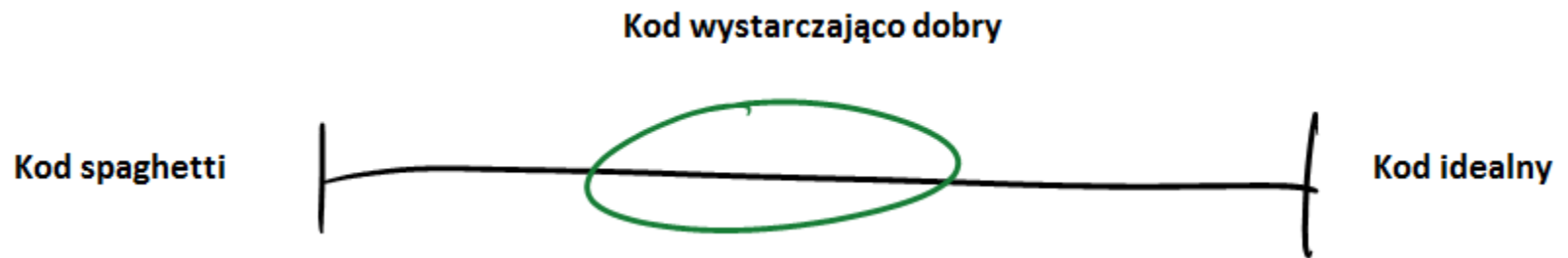
Czytelność dla innych

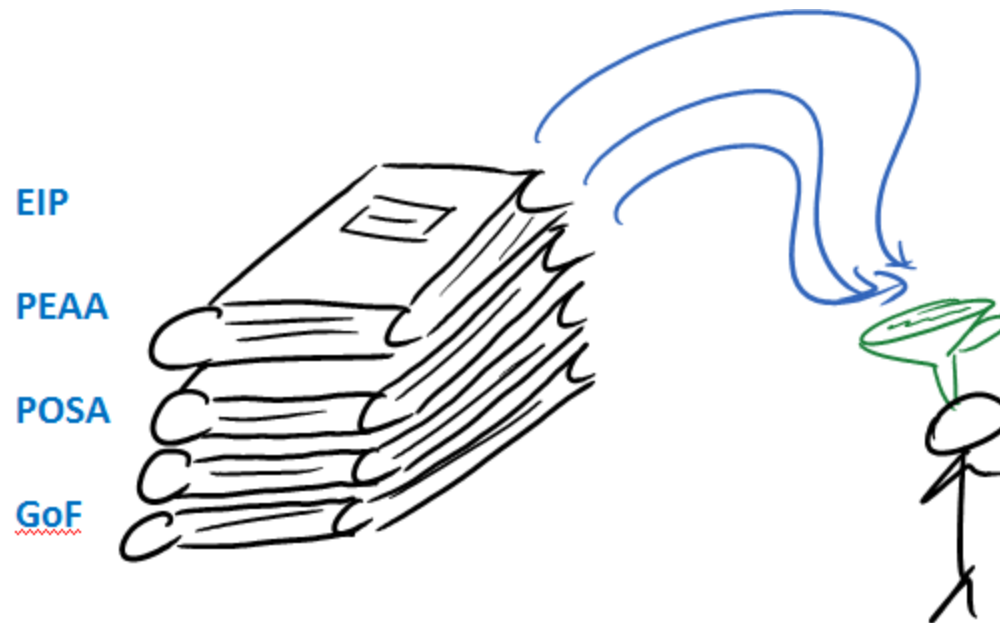
Prostota

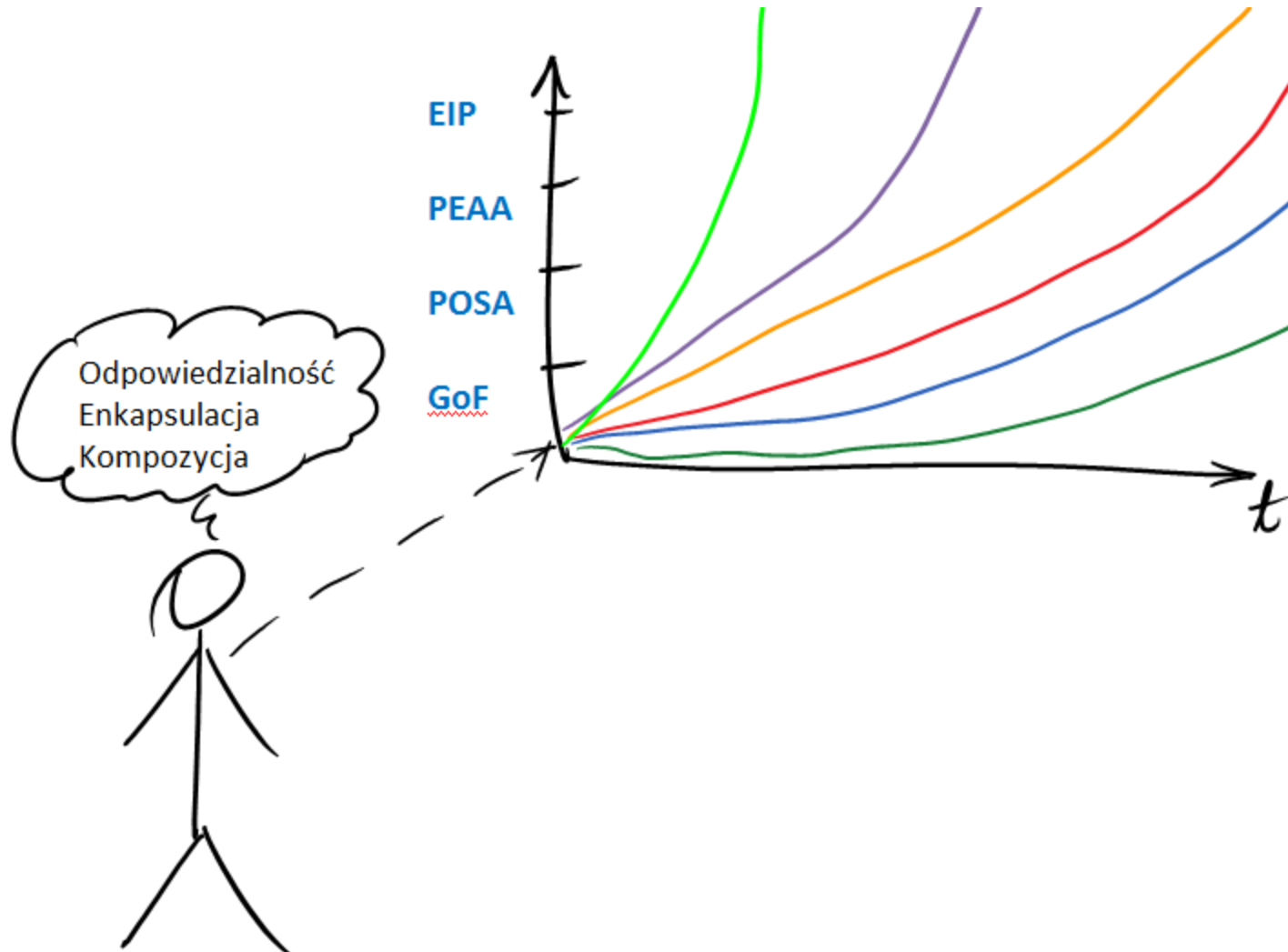
Samoopisywanie się



Kod jest dla programisty, a nie programista dla kodu







Jak żyć, Panie Prelegencie, jak żyć?!

W czym się mierzy doświadczenie?

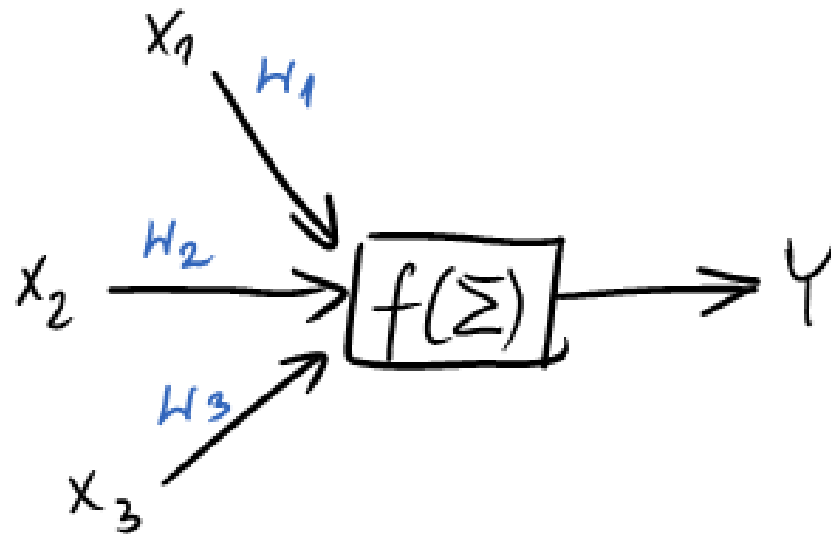
W latach?

W liczbie „wyłożonych” projektów?

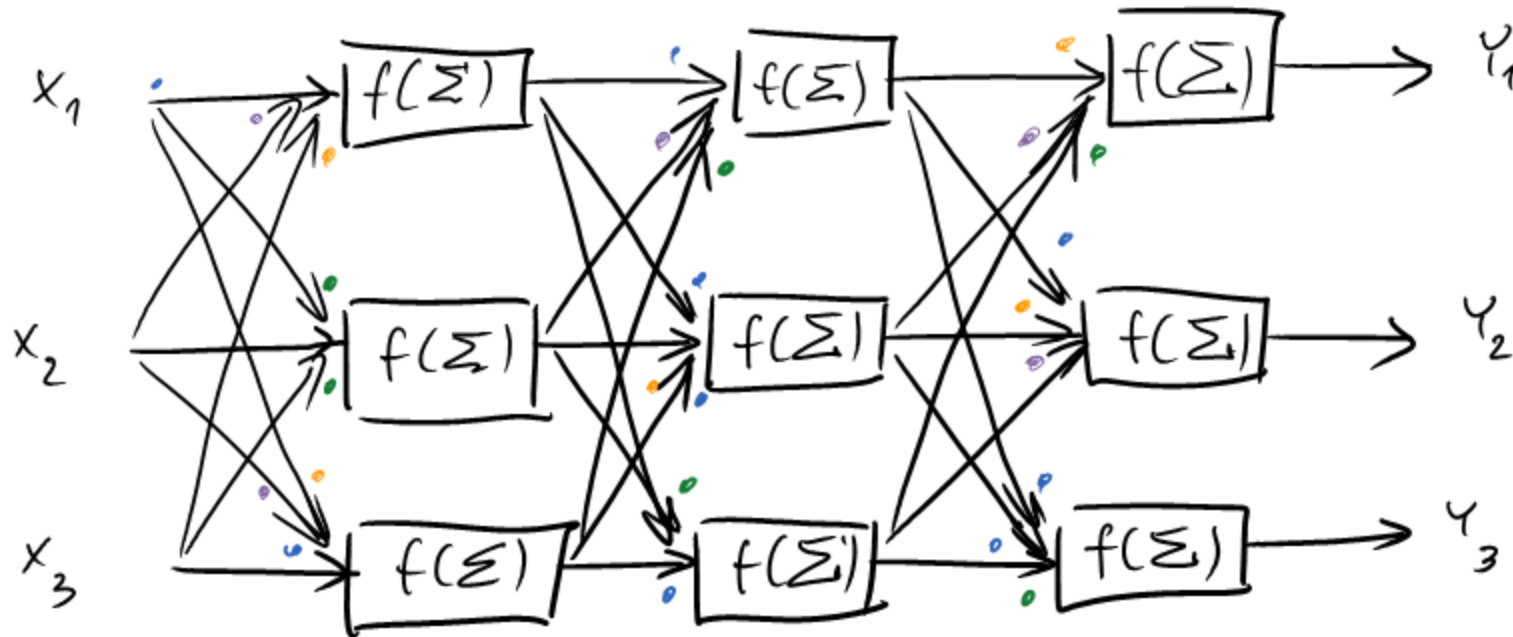
W zmienności:

- Projektów
- Branż
- Współpracowników
- Technologii

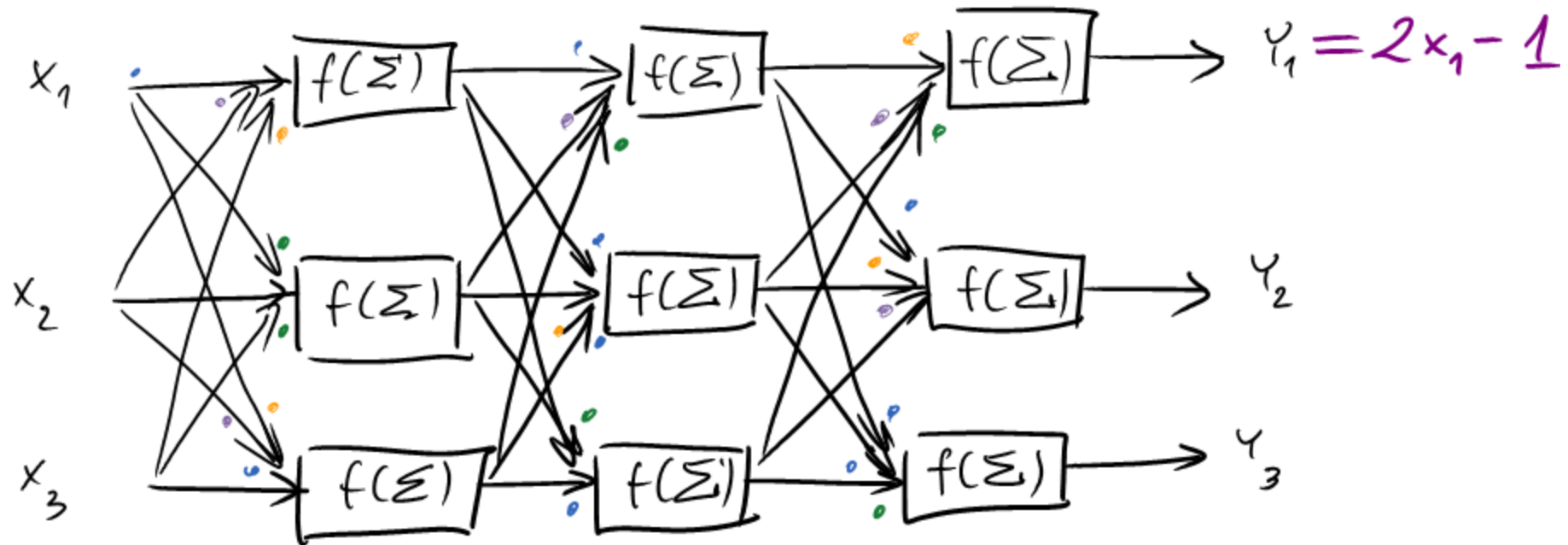
Jak zachodzi uczenie się?



Gdzie są Twoje umiejętności?

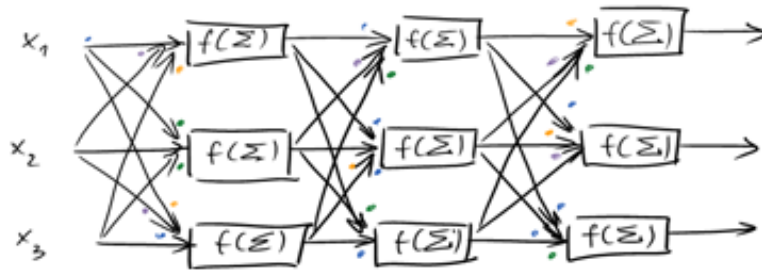


Gdzie są Twoje umiejętności?



Propagacja błędów

Zdobywanie doświadczenia



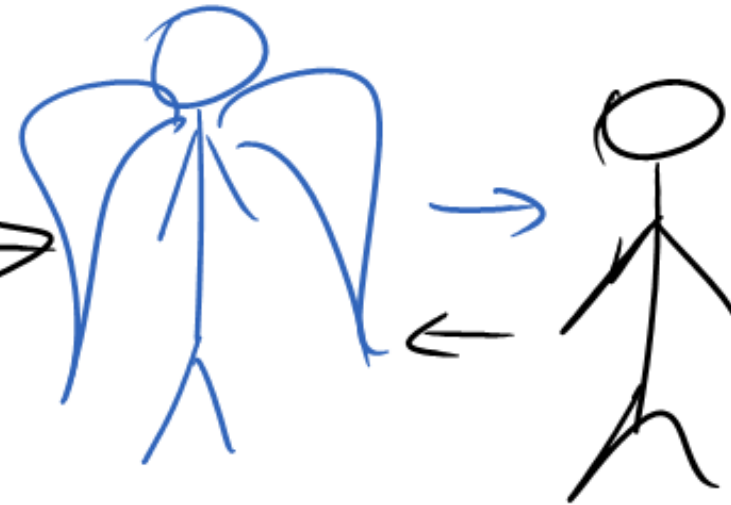
Dobry design

Prosty kod

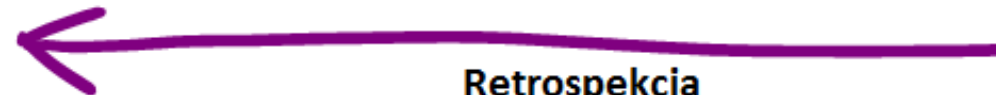
Wystarczająco
dobrze

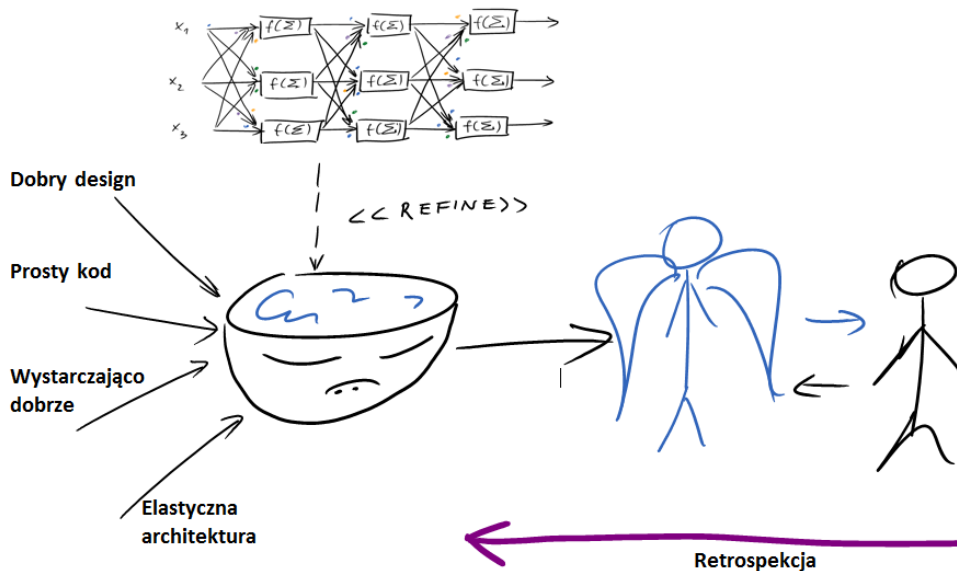
Elastyczna
architektura

<< REFINE >>



Retrospekcja





- # Pracuj z lepszymi od siebie
- # Szukaj różnorodności
- # Nie zwalaj winy za niepowodzenia na czynniki zewnętrzne
- # Nie przywiązuj się zbyt mocno do swojego zdania

- # Szukaj mentorów
- # Szukaj różnorodności
- # Bądź wytrwały
- # Przeprowadzaj retrospekcje
- # Praktykuj pokorę

1. Buduj doświadczenie

2. Ułatwiaj sobie życie za pomocą *-Driven *

Michał Bartyzel

DZIĘKUJĘ ZA WASZ CZAS!



mbartyzel.blogspot.com

@MichalBartyzel



33rd Degree Conference
Conference for Java Masters

13-15 March 2013
Warsaw, Poland