

# DŁUG TECHNICZNY

## Narzędzie profesjonalisty

Adrian Mularczyk

PGS Software

# Krzysztof Kędzierski

"Dług techniczny - Narzędzie Profesjonalisty"

konferencja Boiling Frogs 2018

<https://www.youtube.com/watch?v=PxcQISUIpjQ>

# Agenda

- 1 Wyjaśnienie problemu
- 2 Dług techniczny
- 3 Spłacalny dług techniczny

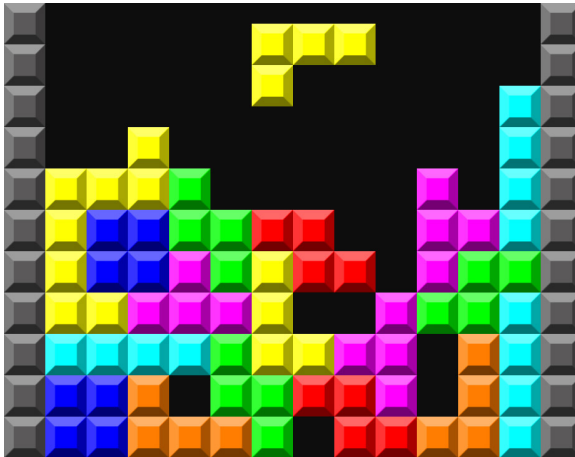
# Wyjaśnienie problemu

$\text{kod} + \text{nowe wymagania} = \text{dług techniczny}$

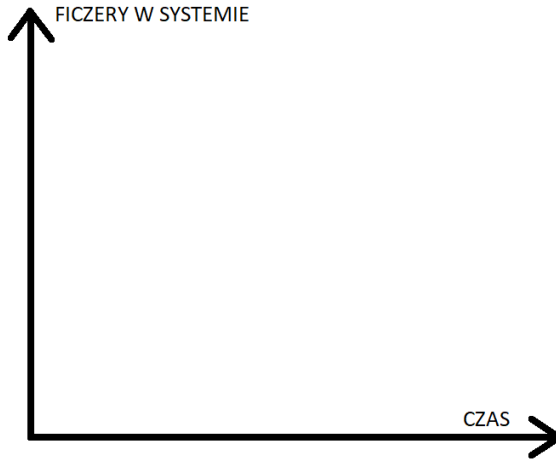
# Dług techniczny

# Dług techniczny

# Tetris

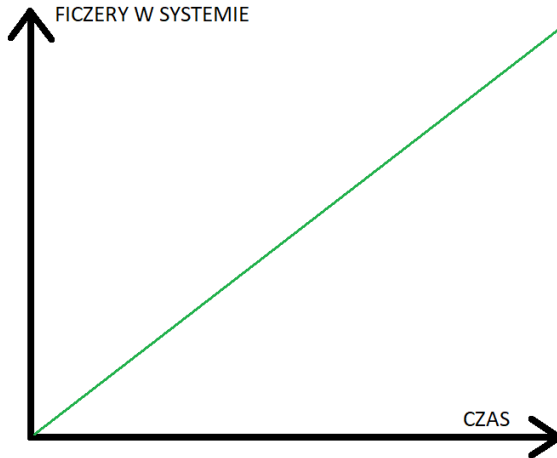


# *Design Stamina Hypothesis*

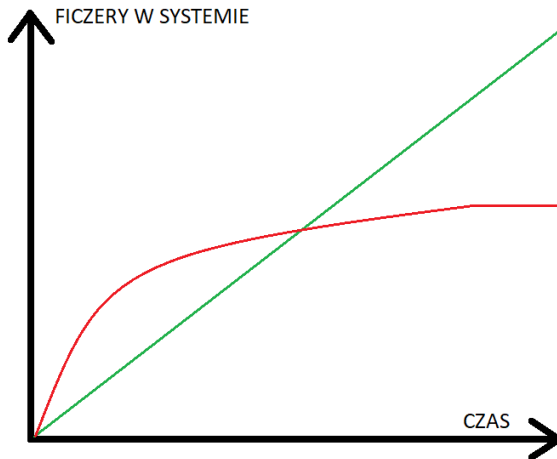




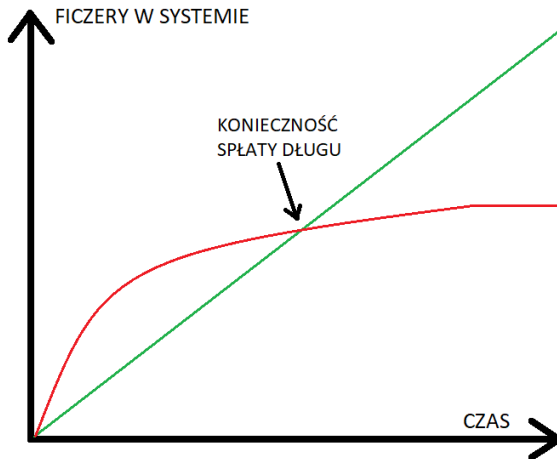
# *Design Stamina Hypothesis*



# *Design Stamina Hypothesis*



# Design Stamina Hypothesis



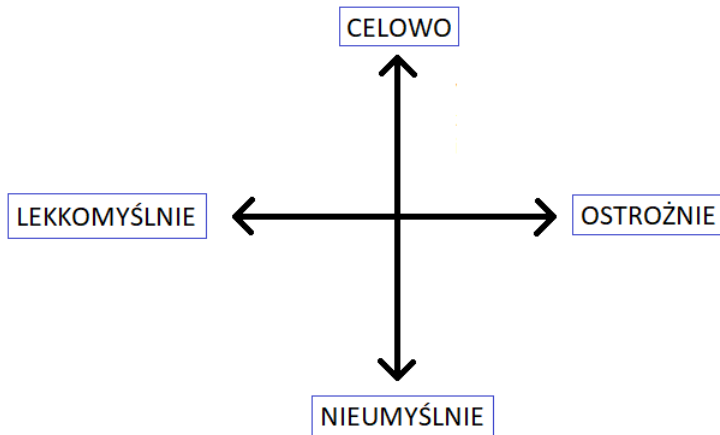
# Spłacalny dług techniczny

# *Technical debt quadrant*

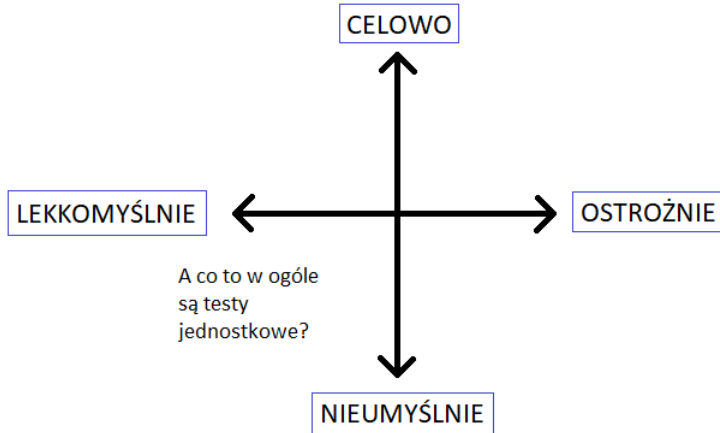
# *Technical debt quadrant*



## *Technical debt quadrant*

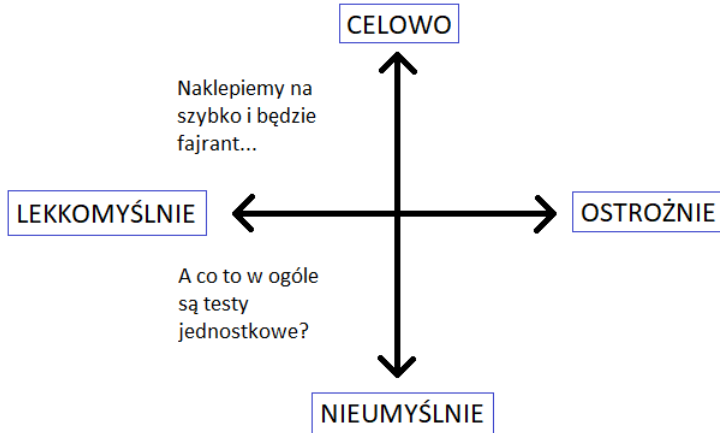


# Technical debt quadrant

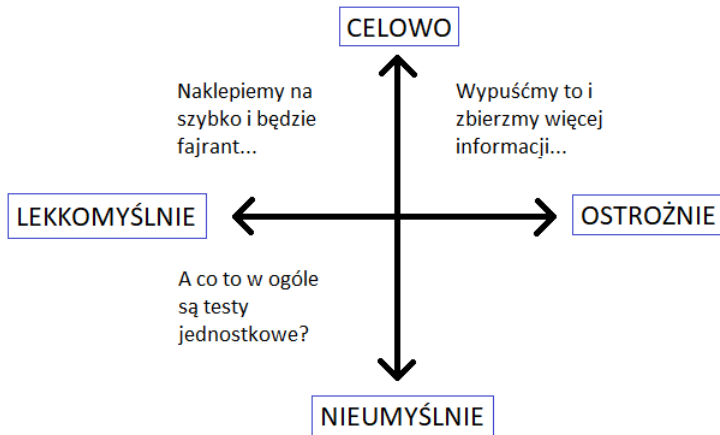




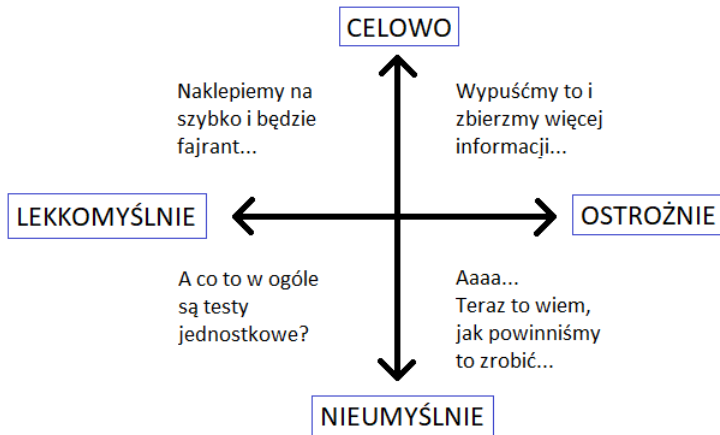
## Technical debt quadrant



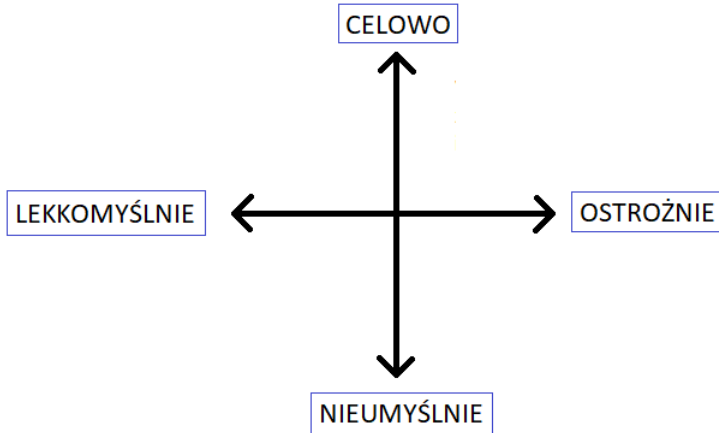
## Technical debt quadrant



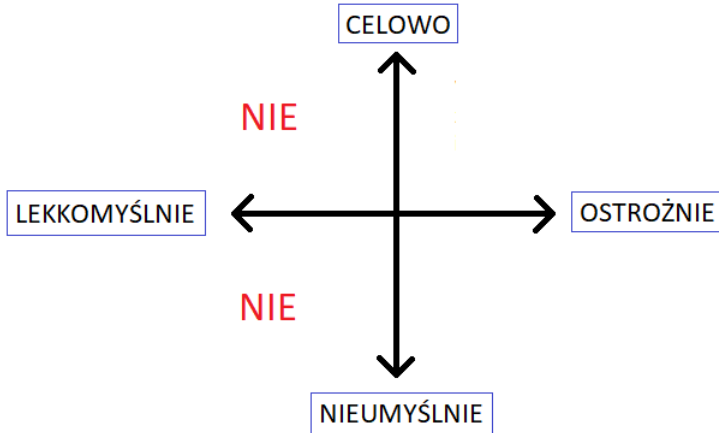
# Technical debt quadrant



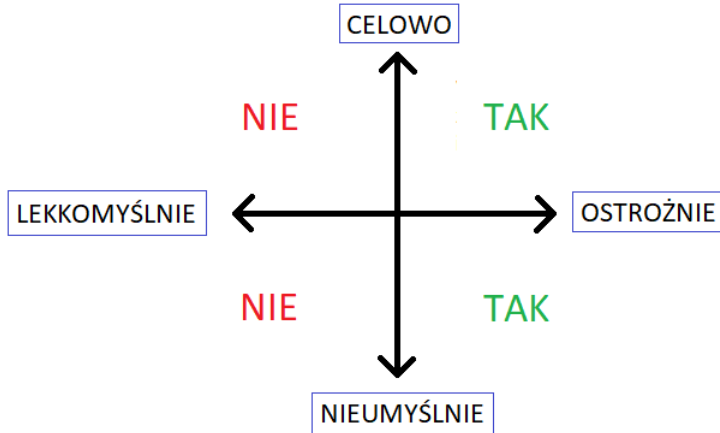
# Czy możliwa jest spłata długu?



# Czy możliwa jest spłata długu?



# Czy możliwa jest spłata długu?



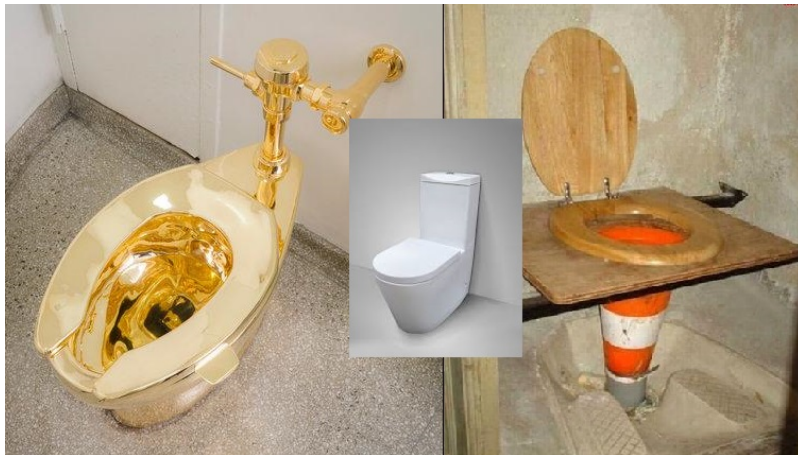
# Spłacalny dług techniczny

- Piszmy czysty kod









- Piszmy czysty kod
- Piszmy prosty kod
- 
- 
-

# Zasada Pareto



# Zasada Pareto

**X%** KODU ODPOWIADA  
ZA **Y%** ZŁOŻONOŚCI/PROBLEMÓW

# Zasada Pareto

5% KODU ODPOWIADA  
ZA 95% ZŁOŻONOŚCI/PROBLEMÓW

# git log --pretty=format:'%ad %aN %s' --numstat

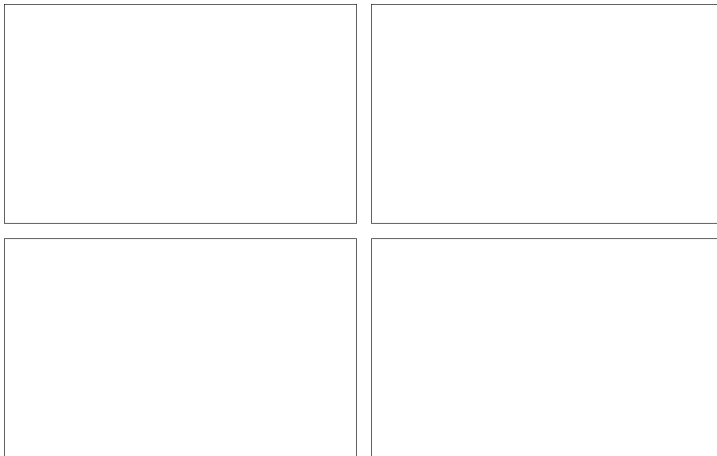
KIEDY

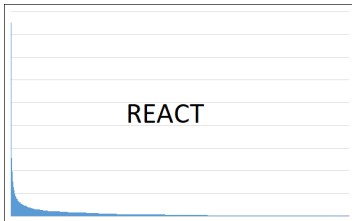
KTO

```
MINGW64: c:/work/employer-back
Tue Sep 4 11:02:45 2018 +0200 [Adrian] [EM-2175] Changing Person Mapper to replace empty JobTitle to null. (#463)
31      0      src/EmployerPortal.Test.Unit.Controllers.UserControllerTests.cs
3       1      src/EmployerPortal/Mappers/PersonMapper.cs

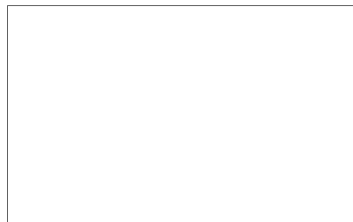
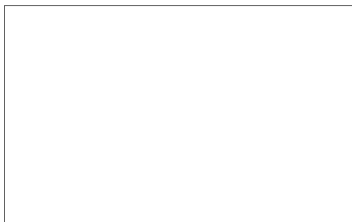
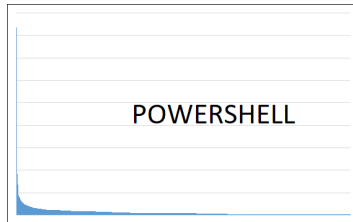
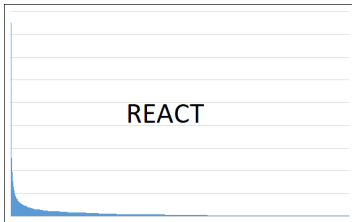
Fri Aug 31 10:18:42 2018 +0200 v-maire [EM-2163] [Company Information] Add Signatory Information panel to the ... (#460)
1       0      src/EmployerPortal.Domain/Models/PersonDto/PersonDto.cs
117     0      src/EmployerPortal.Test.Integration.Controllers.SignatorAccountControllerTests.cs
46      0      src/EmployerPortal/Controllers/SignatorAccountController.cs
30      0      src/EmployerPortal/Validators/DigitalAccount/CreateSignatorAccountValidator.cs
3       0      src/EmployerPortal/Validators/User/UserViewModelValidator.cs
16      0      src/EmployerPortal/ViewModels/DigitalAccount/CreateSignatorAccountRequest.cs
2       0      src/EmployerPortal/ViewModels/User/UserViewModel.cs
|
```

CO



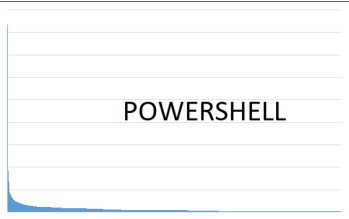




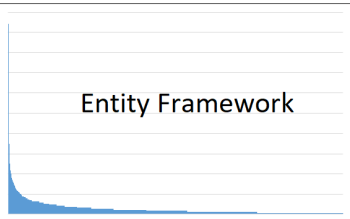




REACT



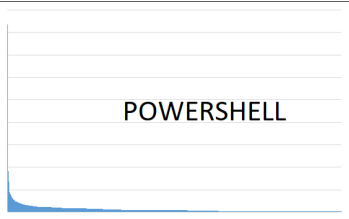
POWERSHELL



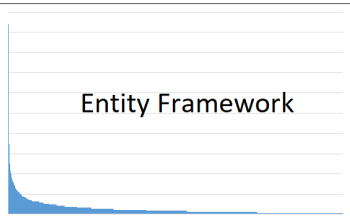
Entity Framework



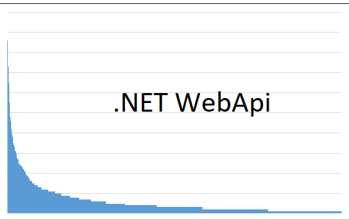
REACT



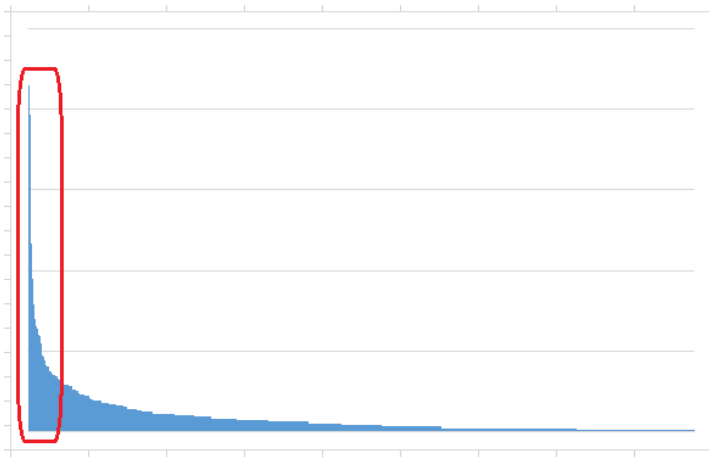
POWERSHELL

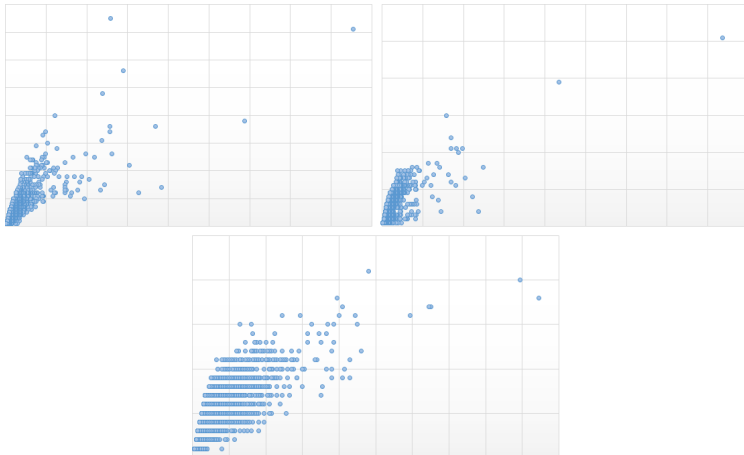


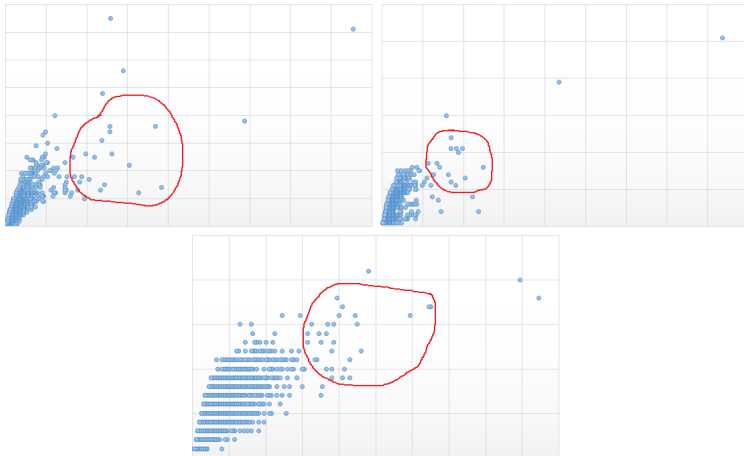
Entity Framework



.NET WebApi





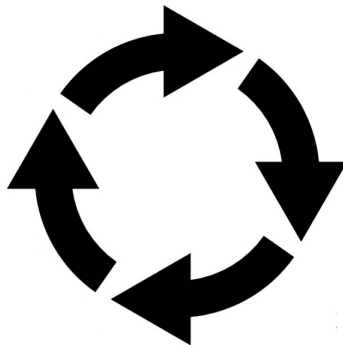


- Piszmy czysty kod
- Piszmy prosty kod
- Monitorujemy gdzie narastają odsetki
- 
-

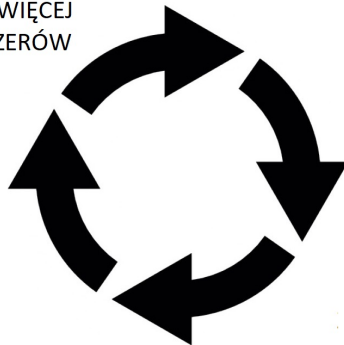




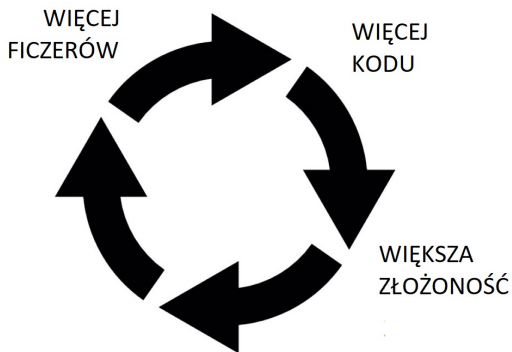
# PGF

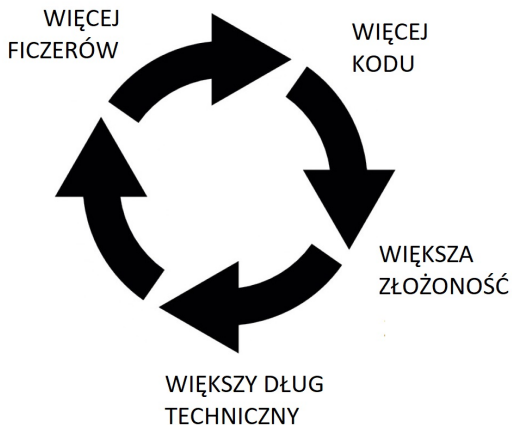


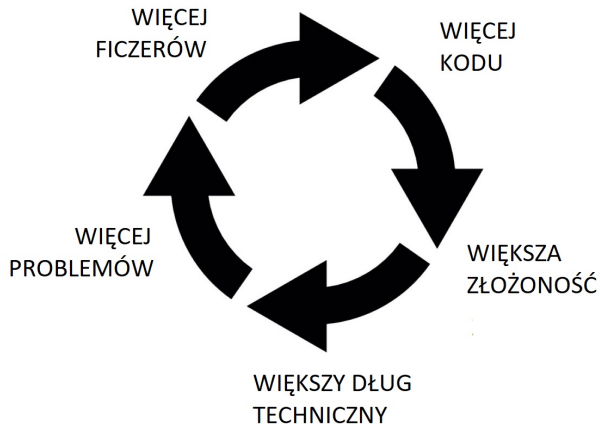
WIĘCEJ  
FICZERÓW

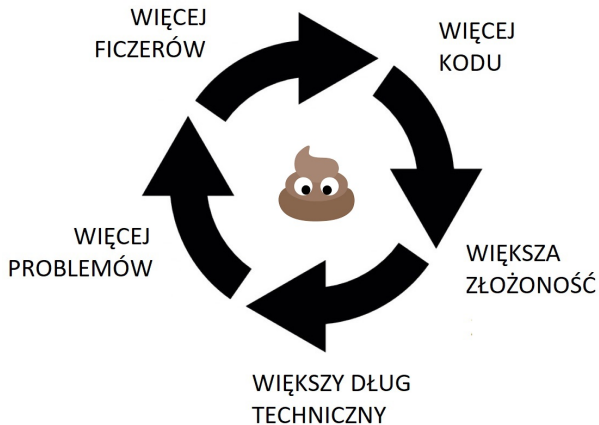






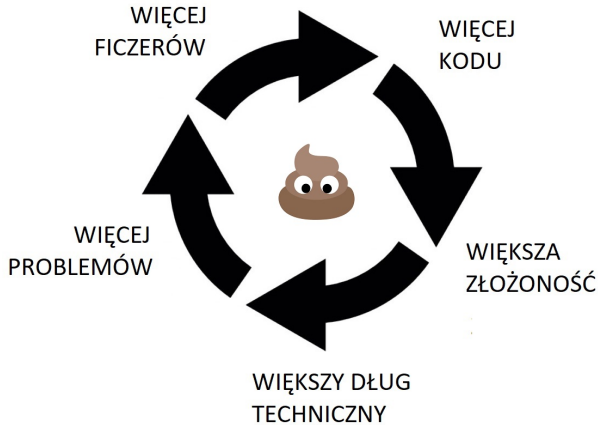






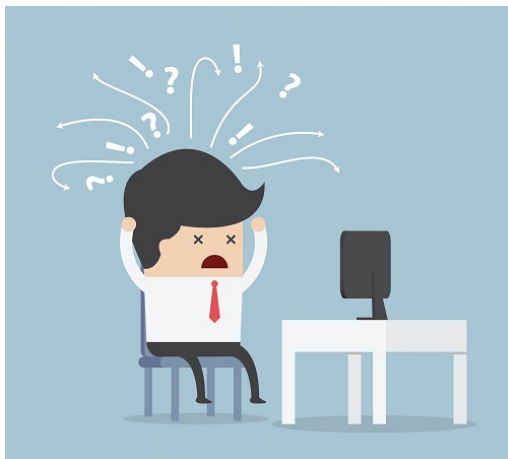


# Pętla gnijących ficzerów

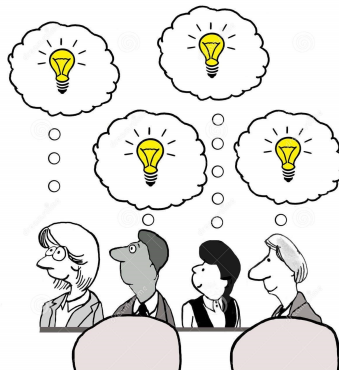


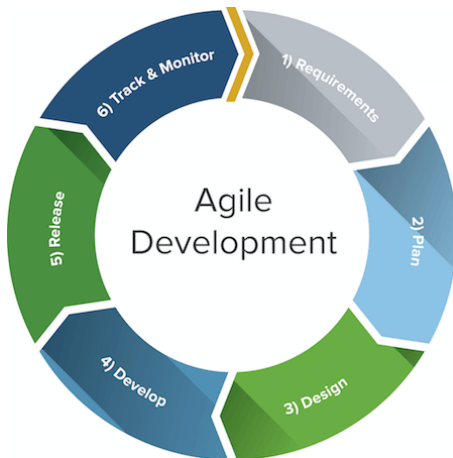
- Piszmy czysty kod
- Piszmy prosty kod
- Monitorujemy gdzie narastają odsetki
- Usuwajmy nieużywane funkcjonalności
-

# Co jest problemem?



# Wymagania





We cannot be agile if our code sucks...

- Piszmy czysty kod
- Piszmy prosty kod
- Monitorujemy gdzie narastają odsetki
- Usuwajmy nieużywane funkcjonalności
- Traktujmy siebie jako profesjonalistów

# Pytania?

# Dziękuję!

e-mail: [amularczyk@pgs-soft.com](mailto:amularczyk@pgs-soft.com)