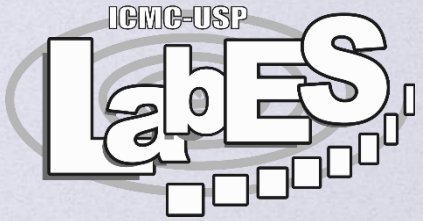


Métricas e Bad Smell

Allan Mori

allanmori@usp.br

Software Engineering Lab (LabES)
http://www.labes.icmc.usp.br/

O que vamos aprender hoje?

- Métricas
- Threshold
- Bad Smell
- Tipos de Bad Smell
- Estratégias de Identificação
- Exercício

Papel do Desenvolvedor

- Código
  - Escrever
  - Dar manutenção
  - Melhorar a Qualidade (Como?)
- Procurar por problemas
  - Arquitetura do sistema
  - Requisitos
  - Código

Métricas

Métricas

- Tamanho: Medidas que caracterizam o código
  - **Lines of Code (LOC)**
  - **Number of Attributes (NOA)**
  - **Number of Methods (NOM)**

- Complexidade: Medidas de decisões e acoplamento
  - Weighted Method per Class (WMC)
  - Lack of Cohesion in Methods (LCOM)

Métricas

- Herança: Medidas de aninhamento de classes
  - **Depth of Inheritance Tree (DIT)**
  - **Number of Children (NOC)**

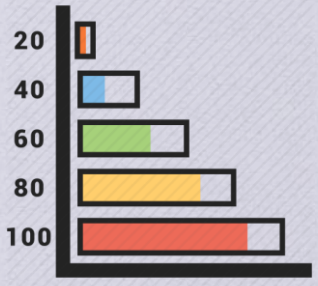
- Perguntas
  - O que é uma classe grande?
  - Uma classe de 100 linhas de código é grande?

Métricas

- Herança: Medidas de aninhamento de classes
  - Depth of Inheritance Tree (DIT)
  - Number of Children (NOC)

- Perguntas
  - O que é uma classe grande?
  - Uma classe de 100 linhas de código é grande?
  - **Que parâmetro utilizar?**

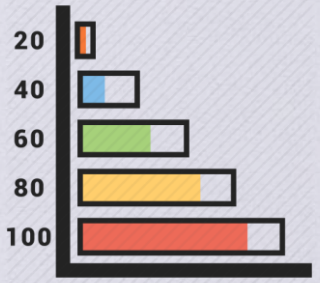
Threshold para Métricas



Threshold para Métricas

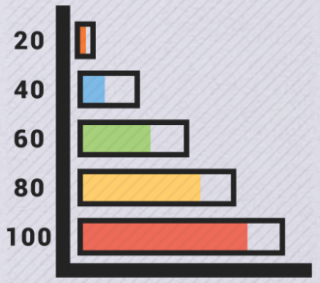
- O fator crucial em trabalhar com métricas é interpretar corretamente esses valores
  - É muito alto ou muito baixo?

- Thresholds separaram os valores em dois grupos
  - Acima do threshold
  - Abaixo do threshold



Fatos sobre Thresholds

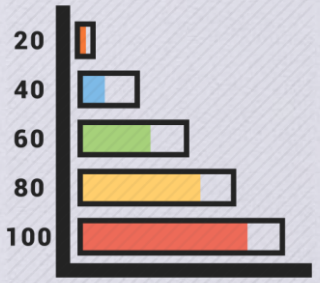
- Não existe um Threshold perfeito
- Podemos definir um Threshold explicável
  - Em outras palavras, os valores são escolhidos baseados em argumentos razoáveis
- Os Thresholds são uteis na prática
  - Identificar Bad Smell



Thresholds

- Um threshold para uma métrica de software tem que estar associado com um contexto para representar uma informação relevante

- Como pode ser definido um Threshold?
  - Experiência (Programador ou Engenheiro de Software)
  - Sistema de Referência (Qualidade)
  - Benchmark (Conjunto de sistemas)
  - Benchmark de Domínios de Software
    - E-Commerce, Games, Saúde...



Exemplo – Domínios de Software

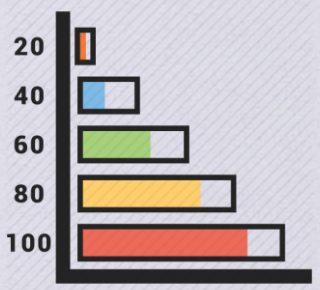
Domain	Description
Accounting	Record and process accounting related processes
Business	Implement validation, calculation, and law regulations
Communication	Manage connections between server and clients
Development	Support developers to implement projects
Dictionaries	Tools used to translate a variety of languages.
E-commerce	Support the transactions of products buying and selling
Education	Manage learning or administrate schools
Free Time	Entertainment systems
Games	Entertainment applications
Health	Systems that offer health-related services
Home	Systems that control many home devices and services.
Localization	Show local information normally based on GPS
Messaging	Allow the users exchange messages
Restaurant	Provide different services for food consumers
Science & Engineering	Designed to aid in several fields of science and engineering

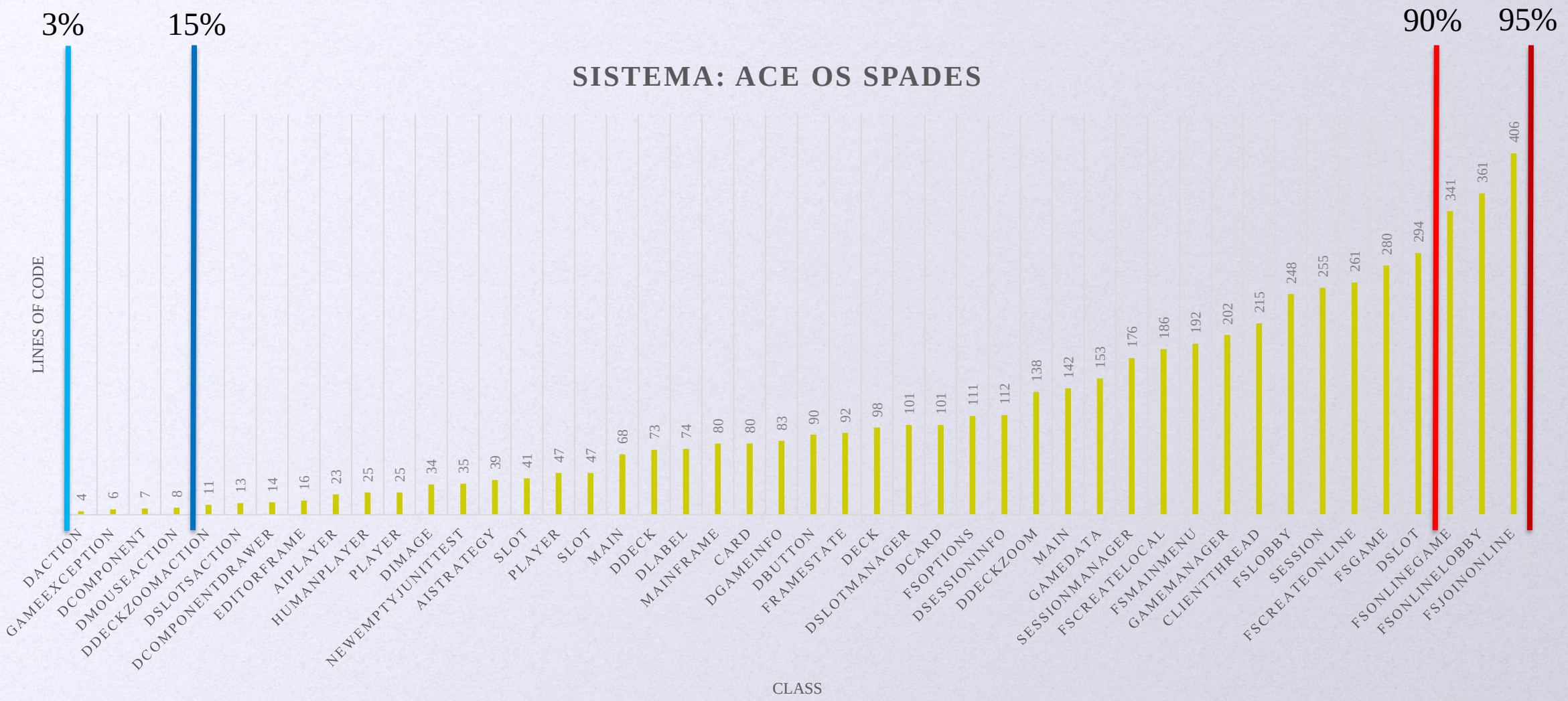
Threshold para Métricas

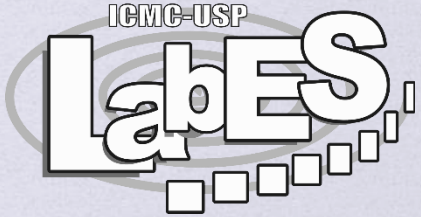
- No contexto da Engenharia de Software
  - O Thresholds é um número que categoriza os componentes de software

- O que é uma classe muito grande para um Sistema de Jogos?
  - Exemplo: Threshold para a métrica Linhas de Código

Rótulo	Classe	LOC Threshold
Muito-baixo	3%	1
Baixo	15%	8
Alto	90%	311
Muito-Alto	95%	491

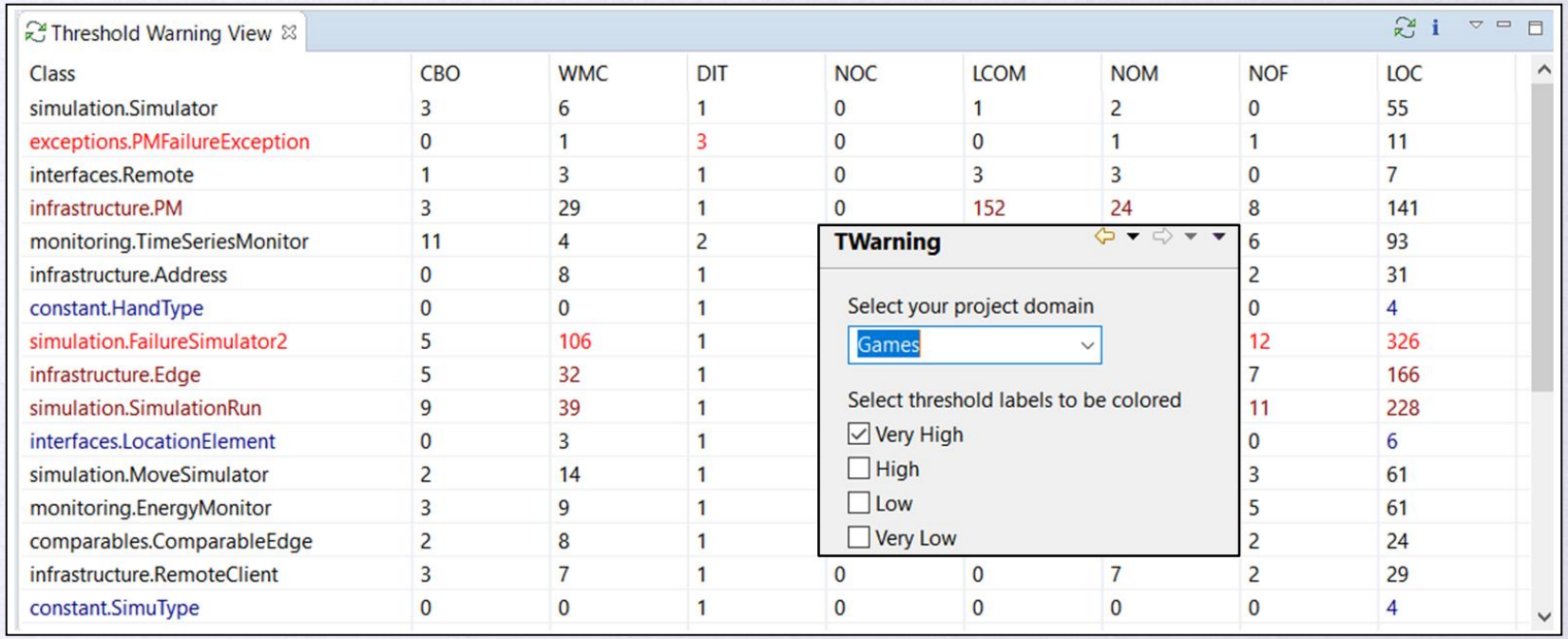



3%
15%
90%
95%
SISTEMA: ACE OS SPADES
LINES OF CODE
CLASS
4
6
7
8
11
13
14
16
23
25
25
34
35
39
41
47
47
68
73
74
80
80
83
90
92
98
101
101
111
112
138
142
153
176
186
192
202
215
248
255
261
280
294
341
361
406
DACTION
GAMEEXCEPTION
DCOMPONENT
DMOUSEACTION
DDECKZOOMACTION
DSLOTSACTION
DCOMPONENTDRAWER
EDITORFRAME
AIPLAYER
HUMANPLAYER
PLAYER
DIMAGE
NEWEMPTYJUNITTEST
AISTRATEGY
SLOT
PLAYER
SLOT
MAIN
DDECK
DLABEL
MAINFRAME
CARD
DGAMEINFO
DBUTTON
FRAMESTATE
DECK
DSLOTMANAGER
DCARD
FSOPTIONS
DSESSIONINFO
DDECKZOOM
MAIN
GAMEDATA
SESSIONMANAGER
FSCREATELOCAL
FSMAINMENU
GAMEMANAGER
CLIENTTHREAD
FSLOBBY
SESSION
FSCREATEONLINE
FSGAME
DSLOT
FSONLINEGAME
FSONLINELOBBY
FSJOINONLINE



TWarning

TWarning



- Plug-in do Eclipse
- Calcula 8 métricas e marca as classes que ultrapassam os thresholds
- Considera o domínios de software para marcar as classes

ICMC-USP LabES

Bad Smell

Bad Smell

- Sintomas no código que podem indicar um problema
- Determinar o que é ou não bad smell depende de:
  - Linguagem, programadores, metodologias de desenvolvimento
- Outros nomes: Code Smell, Anti-patterns
  - Technical Debts e Refatoraração

Lista de Bad Smells

- Definidos por Kent Beck e Martin Fowler
  - 21 Tipos

Tipos de Bad Smell		
Duplicated Code	Data Clumps	Message Chains
Long Method	Primitive Obsession	Middle Man
Large Class	Switch Statements	Inappropriate Intimacy
Long Parameter List	Parallel Inheritance Hierarchies	Classes with Different Interfaces
Divergent Change	Lazy Class	Incomplete Library Class
Shotgun Surgery	Speculative Generality	Data Class
Feature Envy	Temporary Field	Refused Bequest
Comment		

Lista de Bad Smells

- Definidos por Kent Beck e Martin Fowler
  - 21 Tipos

Tipos de Bad Smell		
Duplicated Code	Data Clumps	Message Chains
Long Method	Primitive Obsession	Middle Man
Large Class	Switch Statements	Inappropriate Intimacy
Long Parameter List	Parallel Inheritance Hierarchies	Classes with Different Interfaces
Divergent Change	**Lazy Class**	Incomplete Library Class
Shotgun Surgery	Speculative Generality	Data Class
Feature Envy	Temporary Field	Refused Bequest
Comment		

Duplicated Code / Code Clone

- A mesma estrutura de código aparece mais de uma vez
  - Seu programa pode ser melhor sem código duplicado

- Possíveis Refatorações
  - Extrair o Método: Criar um método com o código duplicado
  - Puxar o Método (Hierarquia): Mover o método genérico para uma superclasse

Long Method / God Method

- Método que centraliza o comportamento de uma classe
  - Quanto maior o método, mais complicado de entender e dar manutenção

- Possíveis Refatorações
  - Separar o Método: Separar o método em outros
  - Substituir o Método pelo Objeto do Método: Transforme o método em sua própria classe

Large Class / God Class

- Classe que está fazendo muitas coisas
  - O sintoma é ter muitos atributos e ter muito código

- Possíveis Refatorações
  - Extrair Classe: Quebrar a classe em duas (ou mais)
  - Extrair Subclasse: Crie uma subclasse

Divergent Change

- Ocorre quando uma classe é alterada de diferentes maneiras por diferentes razões
  - Uma classe deve mudar apenas por um tipo de modificação

- Possível Refatoração
  - Extrair Classe: Separe em duas classes (ou mais)

Shotgun Surgery

- É o oposto de Divergent Change
  - Toda vez que você modifica uma classe, é necessário fazer várias outras pequenas mudanças em outras

- Possíveis Refatorações
  - Mover Método e Mover Atributo: Aproxime os métodos e os atributos que mudam juntos
  - Inline Classe: Combine duas classes em uma classe

Feature Envy

- Um fragmento do código que faz mais acessos a outras classes do que a própria classe

- Possíveis Refatorações
  - Mover Método e Mover Atributos: Aproxime métodos e atributos que mudam juntos
  - Extrair Método: Se apenas uma parte do método sofre de Feature Envy

Lazy Class

- Cada classe custa dinheiro para receber manutenção e para ser entendida
  - A Lazy Class não está fazendo o suficiente para se pagar

- Possíveis Refatorações
  - Colapsar Hierarquia: Combine a superclasse com a subclasse
  - Classe Interna: Combinar duas classes

Comment

- Comentário é bad smell?

Comment

- Comentário não é bad smell
  - Comentários são usualmente usados como desodorantes
  - Comentários são escritos quando um código está ruim

- Possíveis Refatorações
  - Extrair Método: Criar método com bloco de comentários
  - Renomear Método/Atributos: Dar nomes mais significativos para métodos e atributos

Estratégias de Identificação

- Motivação: Uma métrica apenas não pode responder todas as questões sobre um sistema
- É preciso prover meios para o engenheiro de software entender e utilizar as métricas
- A ideia é utilizar uma abstração que combine as métricas

Metáfora Médica

- Métricas podem indicam os sintomas
  - Um único sintoma não é suficiente para entender de uma doença

- Um médico pode identificar uma doença baseado em vários sintomas
  - Um Engenheiro de Software pode identificar problemas de software baseado em várias métricas

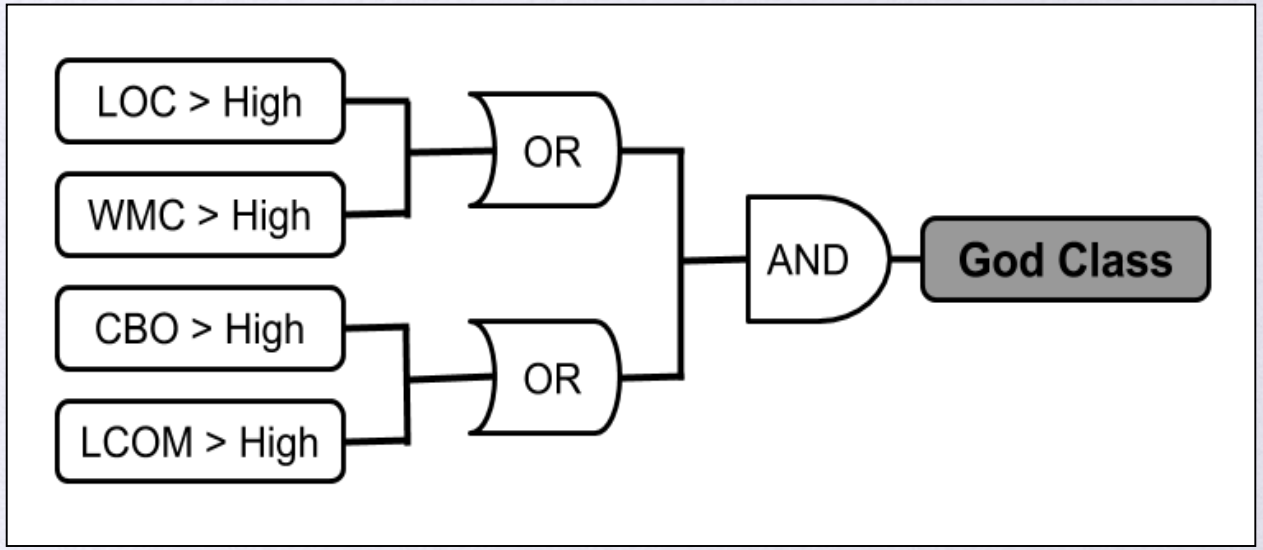
Estratégias de Identificação

- Para todos os bad smells apresentados podem ser criadas estratégias de identificação

- Definição de God Class:
  - Faz muito trabalho
  - É muito grande (LOC)
  - É muito complexa (WMC)
  - Acessa muitos dados de outras classes (CBO ou LCOM)

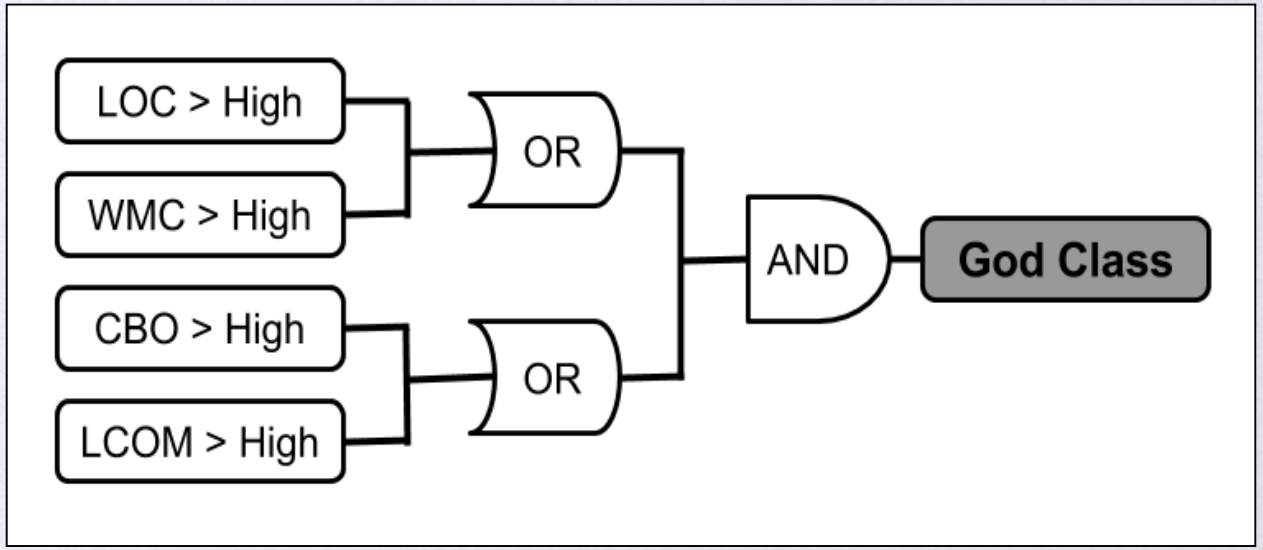
Estratégias de Identificação

- Em outras palavras, God Class é:
  - Muito grande ou muito complexa
  - Baixa coesão e alto acoplamento



Estratégias de Identificação

- Como utilizar na prática?



Estratégia para God Class

Passo 1 – Definir a classe e obter as métricas

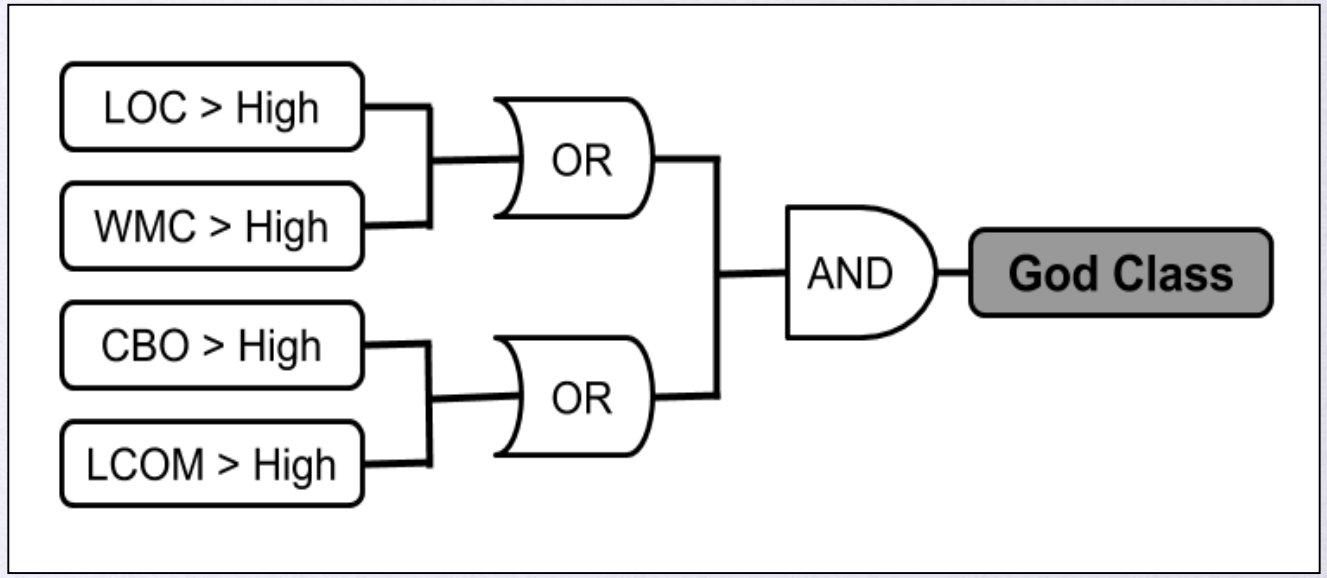
Class	cbo	wmc	dit	noc	lcom	nom	noa	loc
AcessGameEvent	6	154	1	0	178	37	24	482

Passo 2 – Definir os Thresholds

	cbo	wmc	dit	noc	lcom	nom	noa	loc
Very-Low	0	0	1	0	0	0	0	1
Low	0	0	1	0	0	0	0	8
High	13	59	4	0	120	20	11	311
Very-High	18	100	5	1	331	31	18	491

Estratégia para God Class

Passo 3 – Definir o Bad Smell e a Estratégia de Identificação



[(LOC > High) OR (WMC > High)] AND [(CBO > High) OR (LCOM > High)]

Estratégia para God Class

Passo 4 – Substituir os valores na Estratégia de Identificação

[(LOC > High) OR (WMC > High)] AND [(CBO > High) OR (LCOM > High)]

[(**482** > High) OR (**154** > High)] AND [(**6** > High) OR (**178** > High)]

Class	cbo	wmc	dit	noc	lcom	nom	noa	loc
AcessGameEvent	6	154	1	0	178	37	24	482

Estratégia para God Class

Passo 4 – Substituir os valores na Estratégia de Identificação

[(**482** > High) OR (**154** > High)] AND [(**6** > High) OR (**178** > High)]

[(**482** > 311) OR (**154** > 59)] AND [(**6** > 13) OR (**178** > 120)]

	cbo	wmc	dit	noc	lcom	nom	noa	loc
Very-Low	0	0	1	0	0	0	0	1
Low	0	0	1	0	0	0	0	8
High	13	59	4	0	120	20	11	311
Very-High	18	100	5	1	331	31	18	491

Estratégia para God Class

Passo 4 – Substituir os valores na Estratégia de Identificação

[(**482** > 311) OR (**154** > 59)] AND [(**6** > 13) OR (**178** > 120)]

[(**V** OR **V**) AND (**F** OR **V**)]

Verdadeiro

AcessGameEvent é God Class!

Ferramentas

Ferramentas

- Existem algumas ferramentas para encontrar bad smells (Jdeodorant, Jspirit, PMD, SonarQube)

- Geralmente possuem sua próprias métricas de classes
- Tem threshold pré-definido, algumas permitem configurar outro valor
- Utilizam diversas estratégias para encontrar Bad Smells

Exercício

Encontre os bad smells

- Baixe o exercício e aplique as estratégias de identificação de bad smell para as classe do sistema CardGame
  - Ex1 – God Class
  - Ex2 – Lazy Class
  - Ex3 – Feature Envy

Referências

Referências

- Martin Fowler. **Refactoring: improving the design of existing code**. Addison-Wesley Professional, 2018.
- Martin Fowler. **"Refactoring: Improving the design of existing code."** 11th European Conference. Jyväskylä, Finland. 1997.
- Allan Mori, Gustavo Vale, Markos Viggiato, Johnatan Oliveira, Eduardo Figueiredo, Elder Cirilo, Pooyan Jamshidi, and Christian Kastner. **Evaluating Domain-Specific Metric Thresholds: An Empirical Study**. In proceedings of the 1st International Conference on Technical Debt (TechDebt), Gothenburg, Sweden, 2018.

Referências

- Allan Mori, Elder Cirilo, Eduardo Figueiredo. **TWarning: A Warning Tool for Domain-Sensitive Thresholds**. In proceedings of the 9th CBSoft - Tool Session, São Carlos, Brasil, 2018.

- TWarning: https://github.com/Allan045/TWarning/blob/master/Tool/com.labsoft.twarning_1.0.0.jar