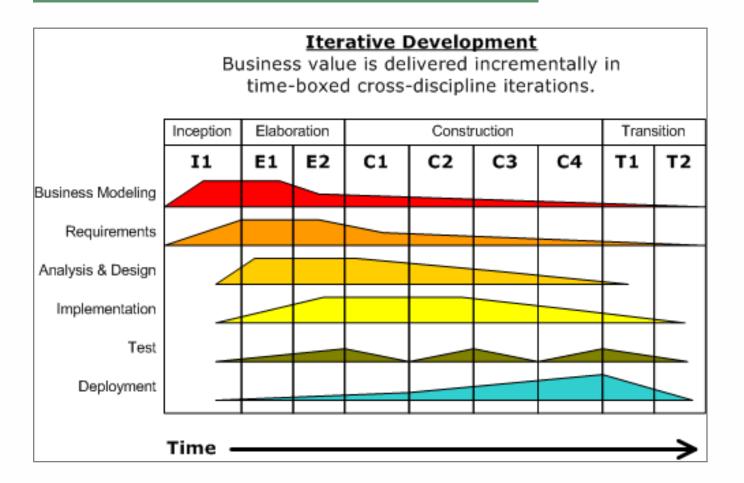
# CAMC and NHD

YEGOR BUGAYENKO

Lecture #9 out of 24 80 minutes

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as website. Copyright belongs to their respected authors.

# Rational Unified Process (RUP)

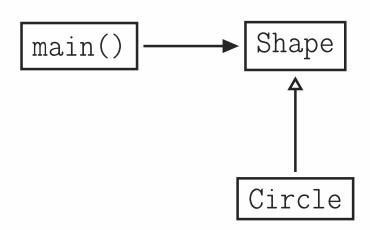


# Decoupling via Interfaces (in Java)

```
interface Shape
  double area();

class Circle implements Shape
  int r;
  @Override
  double area()
  return r * r * 3.14d;

void main(Shape s)
  double a = s.area();
```



### CAMC



"Cohesion Among Methods of Classes (CAMC) evaluates the relatedness of methods in the interface of a class using the parameter lists defined for the methods. It can be applied earlier in the development than can traditional cohesiveness metrics because it relies only on method prototypes declared in a class."

A Class Cohesion Metric for Object-Oriented
 Designs, Jagdish Bansiya, Letha Etzkorn, Carl Davis
 and Wei Li, Journal of Object-Oriented
 Programming, vol. 11, no. 8, 1999

k — total number of methods

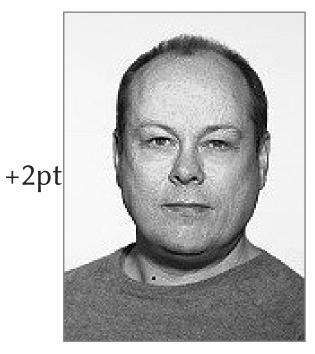
l — total number of types

$$\mathtt{CAMC} = \tfrac{1}{k \times l} \times \sum_{i=1}^k \sum_{j=1}^l o_{ij}$$

O	AlertType	byte	Bitmap	char	va_list	int
Alert	1	1	1	0	0	0
$^{\sim}$ Alert	0	0	0	0	0	0
${\tt DoCreateDialog}$	0	0	0	0	0	0
Show	0	0	0	1	0	0
${\tt ShowV}$	0	0	0	1	1	0
GetMenu	0	0	0	0	0	0
${\tt InspectorId}$	0	0	0	1	0	1

(b) The parameter occurrence matrix.

## NHD



"The hamming distance (HD) provides a measure of disagreement between rows in a binary matrix. The Normalised Hamming Distance (NHD) metric measures agreement between rows in a binary matrix."

— The Interpretation and utility of three cohesion metrics for object-oriented design, Steve Counsell, Stephen Swift, Jason Crampton, ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 15, no. 2, 2006

k — total number of methods

l — total number of types

$$NHD = \frac{2}{l \times k \times (k-1)} \times \sum_{j=1}^{k-1} \sum_{i=j+1}^{k} a_{ij}$$

0	AlertType	byte	Bitmap	char	va_list	int
Alert	1	1	1	0	0	0
$^{\sim}$ Alert	0	0	0	0	0	0
${\tt DoCreateDialog}$	0	0	0	0	0	0
Show	0	0	0	1	0	0
${\tt ShowV}$	0	0	0	1	1	0
GetMenu	0	0	0	0	0	0
${\tt InspectorId}$	0	0	0	1	0	1

(b) The parameter occurrence matrix.

A	Alert	$^{\sim}$ Alert	DoCreateDialog	Show	ShowV	GetMenu
$^\sim$ Alert	3					
${\tt DoCreateDialog}$	3	6				
Show	2	5	5			
${\tt ShowV}$	1	4	4	5		
GetMenu	3	6	6	5	4	
InspectorId	1	4	4	5	4	4
	13	25	19	15	8	4

(c) The parameter agreement matrix.

# Interface Segregation Principle



"Classes that have 'fat' interfaces are classes whose interfaces are not cohesive. In other words, the interfaces of the class can be broken up into groups of methods."

 Agile Software Development, Principles, Patterns, and Practices, Robert C. Martin, 2002

### InputStream in Java

#### Bad:

```
abstract class InputStream
int read();
int read(byte[] b);
int read(byte[] b, int o, int l);

class FileInputStream
implements InputStream
native int read();
native int read(byte[] b, int o, int l);
int read(byte[] b)
return read(b, 0, b.length);
```

#### Better (but slower!):

```
interface InputStream {
  int read(byte[] b, int o, int l);

class FileInputStream
  implements InputStream
  native int read(byte[] b, int o, int l);

class OneByteStream
  InputStream s;
  int read()
  byte[] b = new byte[1];
  s.read(b, 0, 1);
  return (int) b[0];
```

Source: Why InputStream Design Is Wrong (2016)

# Also Known As...

- "interface" in Java
- "protocol" in Objective-C
- •"interface" in C#
- "abstract class" in C++

- absent in Python
- absent in JavaScript
- "interface" in Go
- •"trait" in Rust

#### Read this:

A Class Cohesion Metric for Object-Oriented Designs, Jagdish Bansiya, Letha Etzkorn, Carl Davis and Wei Li, Journal of Object-Oriented Programming, vol. 11, no. 8, 1999

*The Interpretation and utility of three cohesion metrics for object-oriented design*, Steve Counsell, Stephen Swift, Jason Crampton ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 15, no. 2, 2006

Agile Software Development, Principles, Patterns, and Practices, Robert C. Martin, 2002

Why InputStream Design Is Wrong (2016)

Fat vs. Skinny Design (2020)

Fear of Decoupling (2018)

SRP is a Hoax (2017)