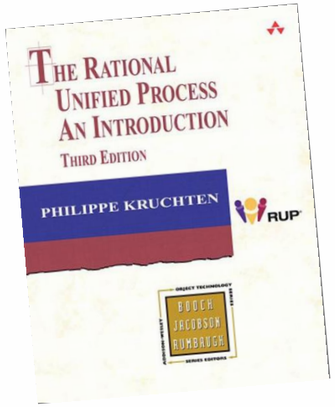
CAMC and NHD

Yegor Bugayenko

Lecture #9 out of 24
80 minutes

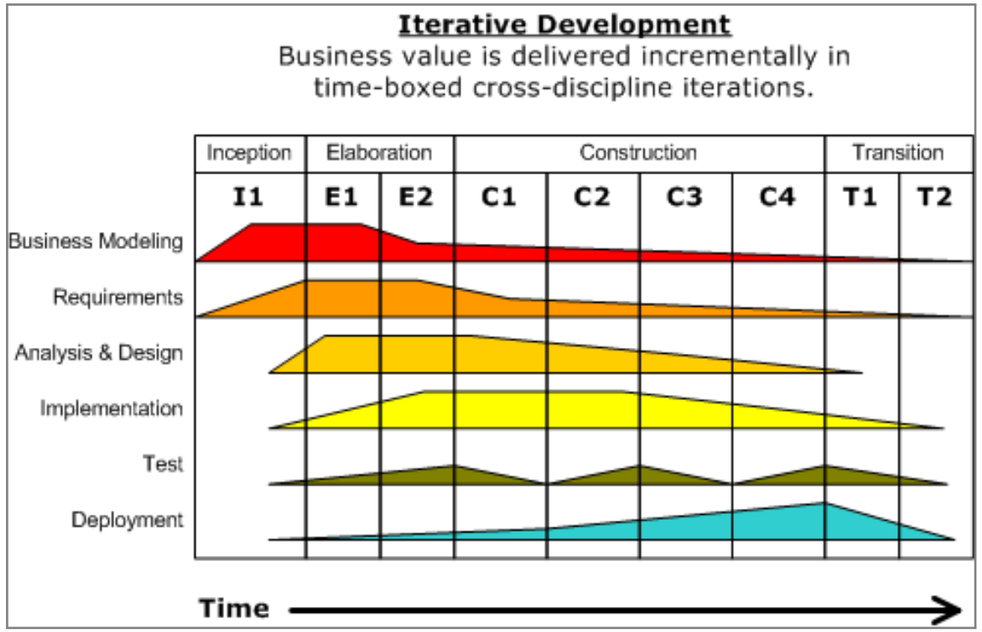
The slidedeck was presented by the author in this YouTube Video



"Without a well-defined process, your development team will develop in an *ad hoc* manner, with success relying on the heroic efforts of a few dedicated individual contributors. This is not a sustainable condition."

— Philippe Kruchten. *The Rational Unified Process: An Introduction.* Addison-Wesley Professional, 2004. doi:10.5555/518604

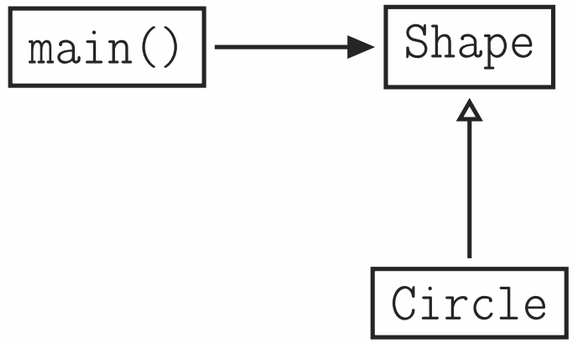
Rational Unified Process (RUP)



Source: Ahmad K. Shuja and Jochen Krebs. IBM Rational Unified Process (RUP) Reference and Certification Guide: Solution Designer, 2007

Decoupling via Interfaces (in Java)

```
1  interface Shape
2    double area();
3
4  class Circle implements Shape
5    int r;
6    @Override
7    double area()
8      return r * r * 3.14d;
9
10 void main(Shape s)
11   double a = s.area();
```



..

Jagdish Bansiya

"**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."

— Jagdish Bansiya. Class Cohesion Metric for Object Oriented Designs. *Journal of Object-Oriented Programming*, 11(8):47–52, 1999

```
Alert(AlertType, byte, *text = 0, Bitmap *bm = 0);
~Alert();
VObject *DoCreateDialog();
int Show(char *fmt);
int ShowV(char *fmt, va_list ap);
class Menu *GetMenu();
void InspectorId(char *buf, int sz);
```

k — total number of methods

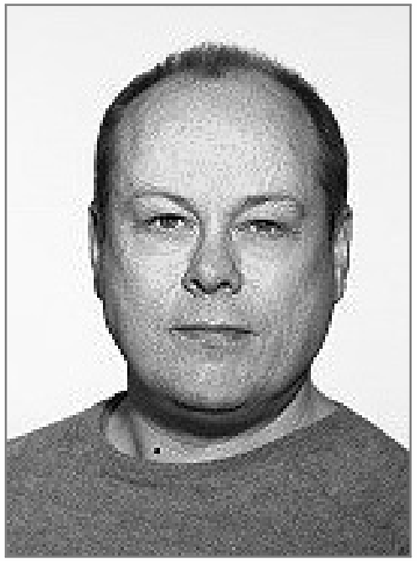
l — total number of types

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

O	AlertType	byte	Bitmap	char	va_list	int
Alert	1	1	1	0	0	0
~Alert	0	0	0	0	0	0
DoCreateDialog	0	0	0	0	0	0
Show	0	0	0	1	0	0
ShowV	0	0	0	1	1	0
GetMenu	0	0	0	0	0	0
InspectorId	0	0	0	1	0	1

(b) The parameter occurrence matrix.

Source: Jagdish Bansiya. Class Cohesion Metric for Object Oriented Designs. *Journal of Object-Oriented Programming*, 11(8):47–52, 1999



Steve Counsell

"**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."

— Steve Counsell, Stephen Swift, and Jason Crampton. The Interpretation and Utility of Three Cohesion Metrics for Object-Oriented Design. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 15(2):123–149, 2006. doi:10.1145/1131421.1131422

```
Alert(AlertType, byte, *text = 0, Bitmap *bm = 0);
~Alert();
VObject *DoCreateDialog();
int Show(char *fmt);
int ShowV(char *fmt, va_list ap);
class Menu *GetMenu();
void InspectorId(char *buf, int sz);
```

k — total number of methods

l — total number of types

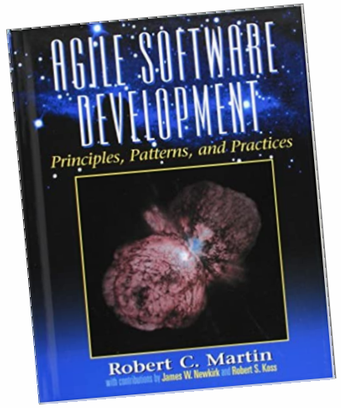
$$\text{NHD} = \frac{2}{l \times k \times (k-1)} \times \sum_{j=1}^{k-1} \sum_{i=j+1}^{k} a_{ij}$$

O	AlertType	byte	Bitmap	char	va_list	int
Alert	1	1	1	0	0	0
~Alert	0	0	0	0	0	0
DoCreateDialog	0	0	0	0	0	0
Show	0	0	0	1	0	0
ShowV	0	0	0	1	1	0
GetMenu	0	0	0	0	0	0
InspectorId	0	0	0	1	0	1

(b) The parameter occurrence matrix.

A	Alert	~Alert	DoCreateDialog	Show	ShowV	GetMenu
~Alert	3					
DoCreateDialog	3	6				
Show	2	5	5			
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.



Robert C. Martin

"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."

— Robert C. Martin. *Agile Software Development, Principles, Patterns, and Practices.* ACM, 2002. doi:10.5555/515230

InputStream in Java 1.0

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: Yegor Bugayenko. Why InputStream Design Is Wrong. https://www.yegor256.com/160426.html, apr 2016. [Online; accessed 22-09-2024]

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

Source: Yegor Bugayenko. Fat vs. Skinny Design. https://www.yegor256.com/200219.html, feb 2020. [Online; accessed 26-10-2024]

References

Jagdish Bansiya. Class Cohesion Metric for Object Oriented Designs. *Journal of Object-Oriented Programming*, 11(8):47–52, 1999.

Yegor Bugayenko. Why InputStream Design Is Wrong. `https://www.yegor256.com/160426.html`, apr 2016. [Online; accessed 22-09-2024].

Yegor Bugayenko. Fat vs. Skinny Design. `https://www.yegor256.com/200219.html`, feb 2020. [Online; accessed 26-10-2024].

Steve Counsell, Stephen Swift, and Jason Crampton. The Interpretation and Utility of Three Cohesion Metrics for Object-Oriented Design. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 15(2):123–149, 2006. doi:10.1145/1131421.1131422.

Philippe Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley Professional, 2004. doi:10.5555/518604.

Robert C. Martin. *Agile Software Development, Principles, Patterns, and Practices*. ACM, 2002. doi:10.5555/515230.

Ahmad K. Shuja and Jochen Krebs. IBM Rational Unified Process (RUP) Reference and Certification Guide: Solution Designer, 2007.