

Readability

Michel Grootjans – SmallSteps



michel.grootjans@gmail.com



#michelgrootjans

An example

The librarian

Ook

A programming language designed for
orang-utans

Ook: design principles

A programming language should be writable and readable by orang-utans.

To this end, the syntax should be simple, easy to remember, and not mention the word "monkey".

Bananas are good.

Ook: Syntax elements

Ook.

Ook!

Ook?

Ook: Commands

Command	Symbol	Meaning
Ook. Ook?	>	increment the pointer
Ook? Ook.	<	decrement the pointer
Ook. Ook.	+	increment the integer at the pointer
Ook! Ook!	-	decrement the integer at the pointer
Ook! Ook.	.	output the ASCII character from the integer at the pointer
Ook. Ook!	,	input to the integer at the pointer (ASCII).
Ook! Ook?	[jump forward to the statement after the corresponding Ook? Ook! if the byte at the pointer is zero
Ook? Ook!]	jump back to the statement after the corresponding Ook! Ook? if the byte at the pointer is nonzero

Ook: Comments

Since the word "ook" can convey entire ideas, emotions, and abstract thoughts depending on the nuances of inflection, Ook! has no need of comments. The code itself serves perfectly well to describe in detail what it does and how it does it.

Provided you are an orang-utan.

Let's see some code

“Any monkey can write code that a computer can understand.

Good apes write code that an orang-utang can understand.”

-- The Librarian

Readability

Why?

Why?

It's not about **bits**, **bytes** and **protocols**,
but **profits**, **losses** and **margins**.

-- Lou Gerstner

Why?

$$\text{cost}_{\text{total}} = \text{cost}_{\text{develop}} + \text{cost}_{\text{maintain}}$$

$$\text{cost}_{\text{maintain}} = \text{cost}_{\text{understand}} + \text{cost}_{\text{change}} + \text{cost}_{\text{test}} + \text{cost}_{\text{deploy}}$$



Why?

Easier to understand

Easier to change

Easier to test

== Cheaper to maintain



In short

“There are two ways of constructing a software design.

One way is to make it so simple that there are **obviously no deficiencies.**

And the other way is to make it so complicated that there are **no obvious deficiencies.**”

-- *C.A.R. Hoare*

Readability

For whom?

Know your audience



Who are you writing code for?

The computer



Yourself

//Dear future me. Please forgive me.

//I can't even begin to express how sorry I am.

“Any code of your own that you haven’t looked at for six or more months might as well have been written by someone else.”

-- *Eagleson's Law*

Other programmers

```
// no comments for you  
// it was hard to write  
// so it should be hard to read
```

Other programmers

// I am not responsible of this code.

// They made me write it, against my will.

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.”

-- Martin Golding



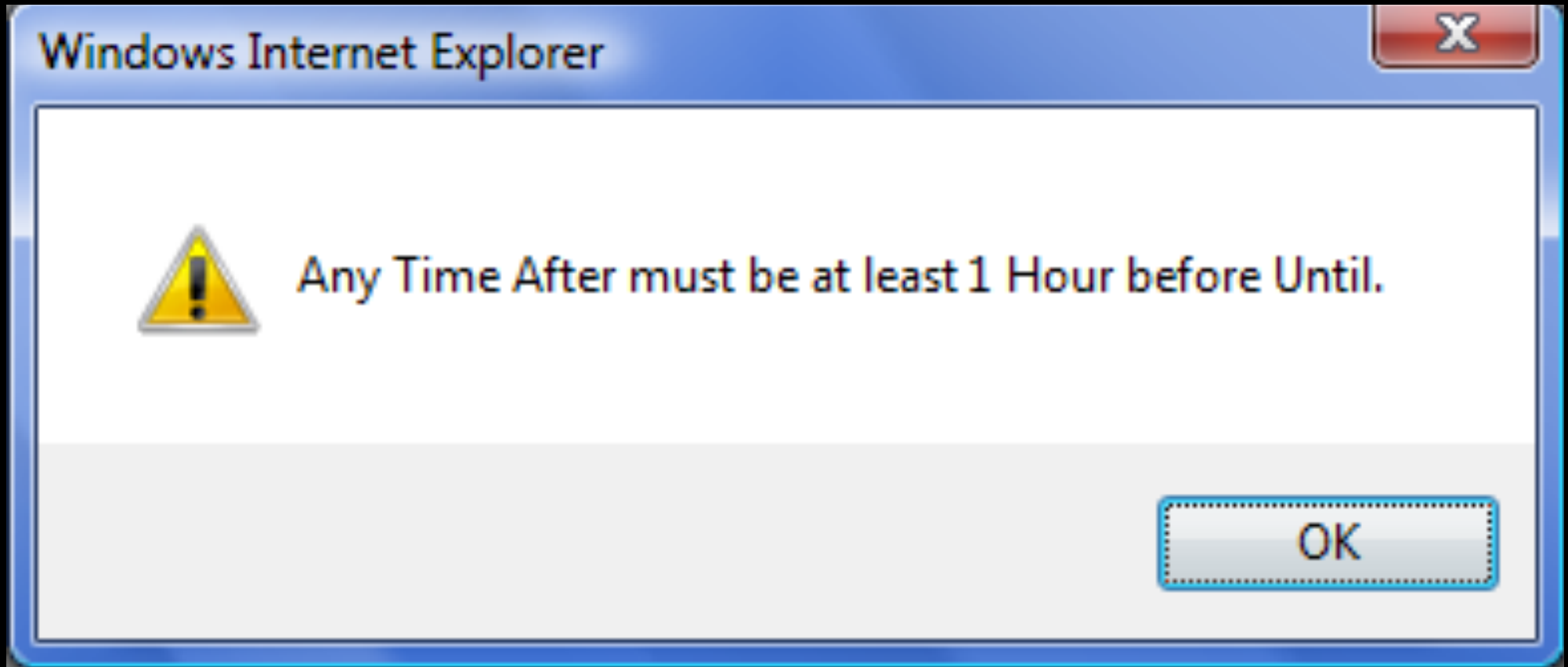
Other programmers

```
//  
// Dear maintainer:  
//  
// Once you are done trying to 'optimize' this routine,  
// and have realized what a terrible mistake that was,  
// please increment the following counter as a warning  
// to the next guy:  
//  
// total_hours_wasted_here = 16  
//
```

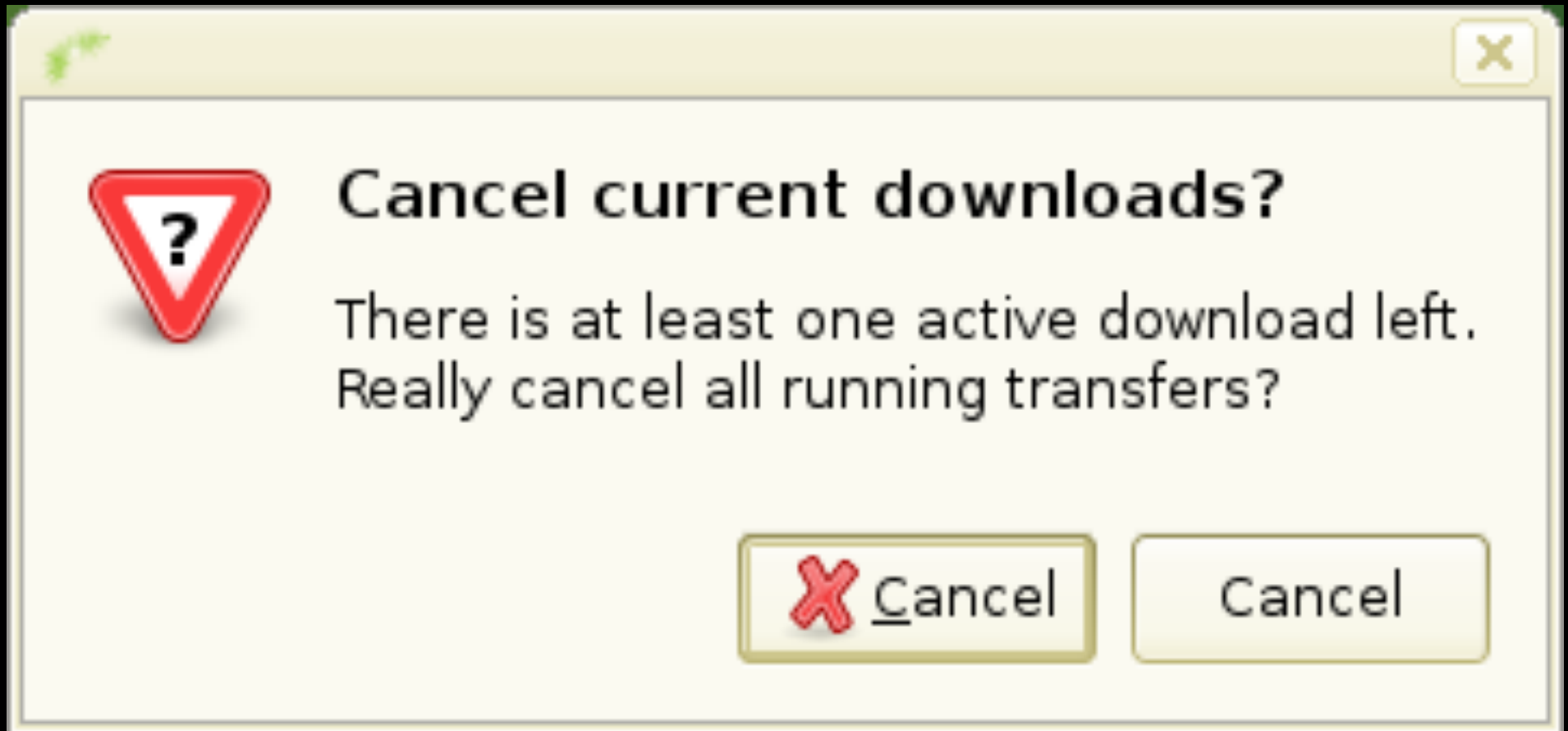
Your next boss



Your users




Your users



Your users

Mobile Master

 **Do you use a Nokia phone?**

Yes, Nokia phone

Your installers

Installing Installation Manager with the Installation Manager installer

Use the IBM Installation Manager installer to install both IBM Installation Manager and Rational products.

About this task

If Installation Manager is not installed, you can use the Installation Manager installer to silently install Installation Manager with a single command. The Installation Manager installer is included in the installation files for Installation Manager.

The Installation Manager installer uses the initialization, ini, file: `silent-install.ini`. This file located in the `InstallerImage_platform` directory. The `silent-install.ini` file includes default values for command-line arguments. For information about the command line arguments, see [IBM Installation Manager command-line arguments](#).

After Installation Manager is installed, you can use it to silently install other products packages.

Err...

//When I wrote this,

//only God and I understood what I was doing

//Now, God only knows

Readability

What not?

Bad code

Bad Code -- Bob Martin

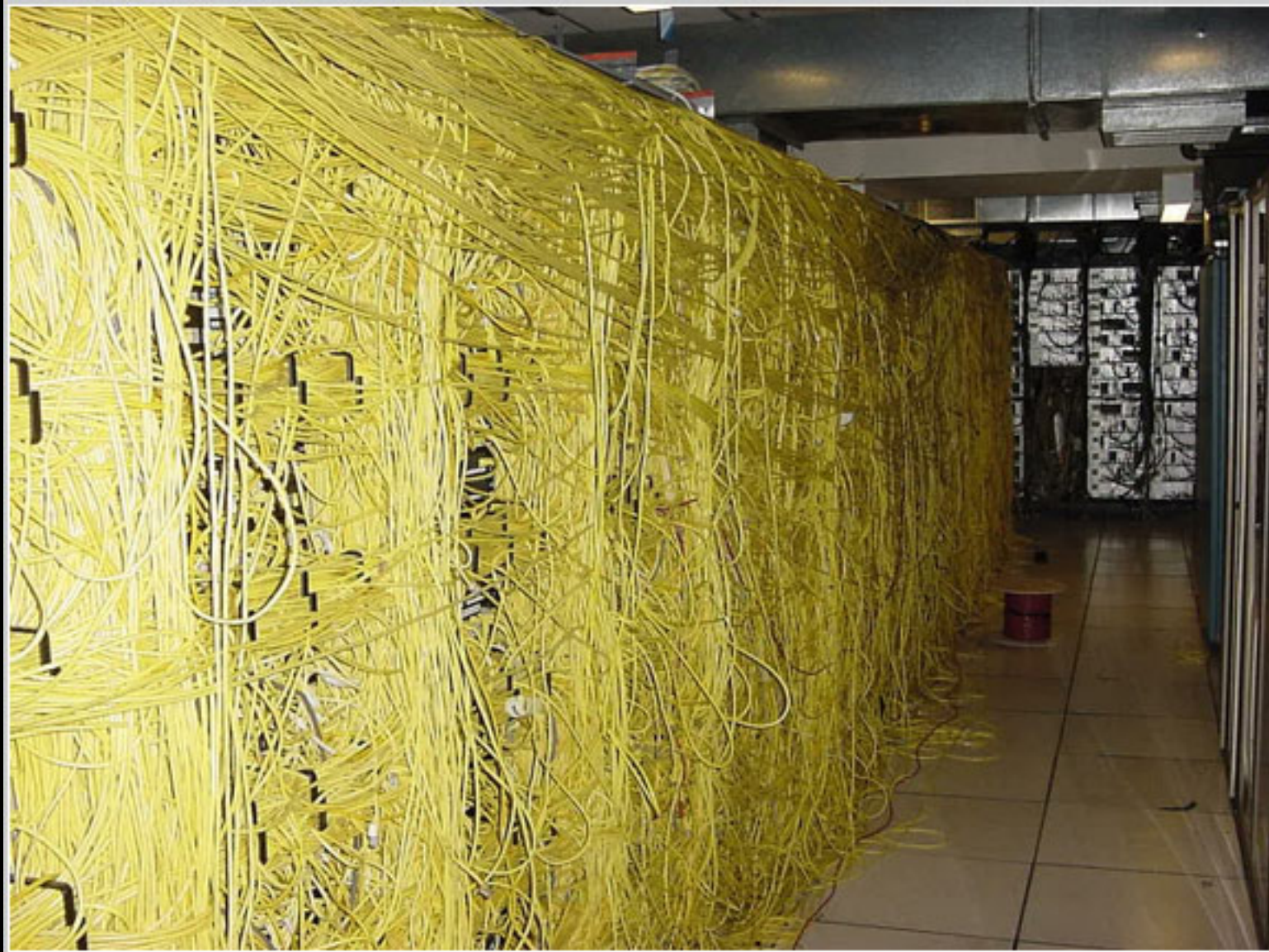
What goes wrong with software?

Code ROTS

The first release



After 3 years of maintenance



Software today is very much like an **Egyptian pyramid**

Software today is very much like an Egyptian pyramid with millions of bricks **pled** on top of each other,

Software today is very much like an Egyptian pyramid
with millions of bricks piled on top of each other,
with **no** structural **integrity**,

Software today is very much like an Egyptian pyramid
with millions of bricks piled on top of each other,
with no structural integrity,
but just done by **brute force**

Software today is very much like an Egyptian pyramid
with millions of bricks piled on top of each other,
with no structural integrity,
but just done by brute force
and thousands of **slaves**

Software today is very much like an Egyptian pyramid
with millions of bricks piled on top of each other,
with no structural integrity,
but just done by brute force
and thousands of slaves

-- Alan Kay


```
public string rom2num(string r)
{
    if (r == "I") return "1";
    if (r == "II") return "2";
    if (r == "III") return "3";
    if (r == "IV") return "4";
    if (r == "V") return "5";
    if (r == "VI") return "6";
    if (r == "VII") return "7";
    if (r == "VIII") return "8";
    if (r == "IX") return "9";
    //
    // Snipped LOTS of "code" here
    //
    if (r == "MMVIII") return "2008";
    if (r == "MMIX") return "2009";
    if (r == "MMX") return "2010";
    if (r == "MMXI") return "2011";
    return "E";
}
```

```
$times = array(
    "" => "",
    "12:00:00" => "12:00 pm",
    "12:30:00" => "12:30 pm",
    "13:00:00" => "1:00 pm",
    "13:30:00" => "1:30 pm",
    "14:00:00" => "2:00 pm",
    "14:30:00" => "2:30 pm",
    "15:00:00" => "3:00 pm",
    "15:30:00" => "3:30 pm",
    "16:00:00" => "4:00 pm",
    "16:30:00" => "4:30 pm",
    "17:00:00" => "5:00 pm",
    "17:30:00" => "5:30 pm",
    "18:00:00" => "6:00 pm",
    "18:30:00" => "6:30 pm",
    "WY" => "Wyoming");
```

```
function doseq()  
    dim a, b, c, d  
    b = #11/4/2007 7:04:21 PM#  
    a = datediff("s", b, now)  
  
    a = a + 1194231874  
    doseq = a  
end function
```

```
def meets_criteria num
  criteria = {}
  (2..num).each do |i|
    criteria[i] = true
  end
  (2..(num/2)).each do |i|
    j = i + i
    while (j <= num) do
      criteria[j] = false
      j = j + i
    end
  end
  criteria.find_all{|c| c[1]}.map{|c| c[0]}
end
```

```
resources.reverse.drop(1).reverse.each { |resource| args[1] << resource }
```

Readability

What?

Legibility

کتابتِ اسلامیہ کے بارے میں
میں نے کئی کتابیں پڑھی ہیں
اور ان میں سے بہت سے
نکات یاد رکھنے کے قابل ہیں
جو ہمیں اپنی زندگی میں
بہتر بنانے کے لیے
استعمال کرنے چاہئیں۔

Understandability

You need to know the technology

```
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
```


Understandability

You need to know the language

Synchronized interactive conglomeration

Right-sized static pricing structure

Cross-group intangible process improvement

Synergized flow catchments

Understandability

The words need to make sense



Clean



You have to learn what clean means



Readability

How?

Documentation

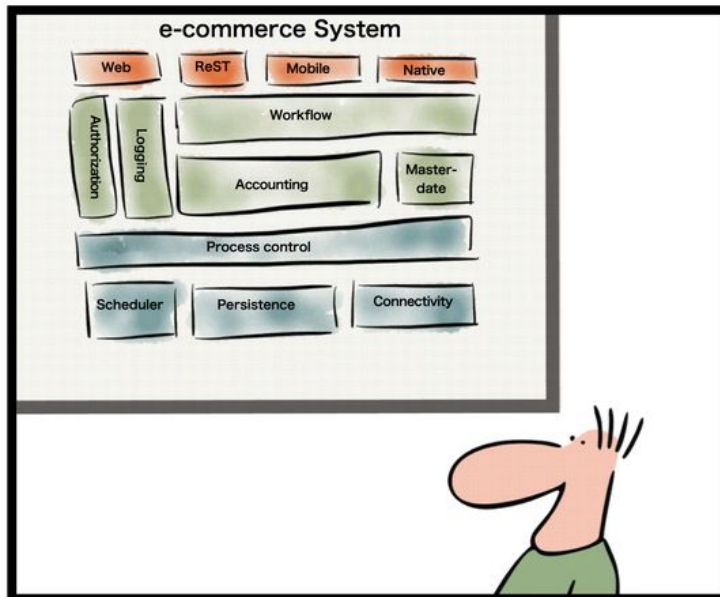
The palest ink is better than the best memory
-- Chinese proverb

**If you want them to
RTFM, make a better FM**



Documentation

Describe the environment



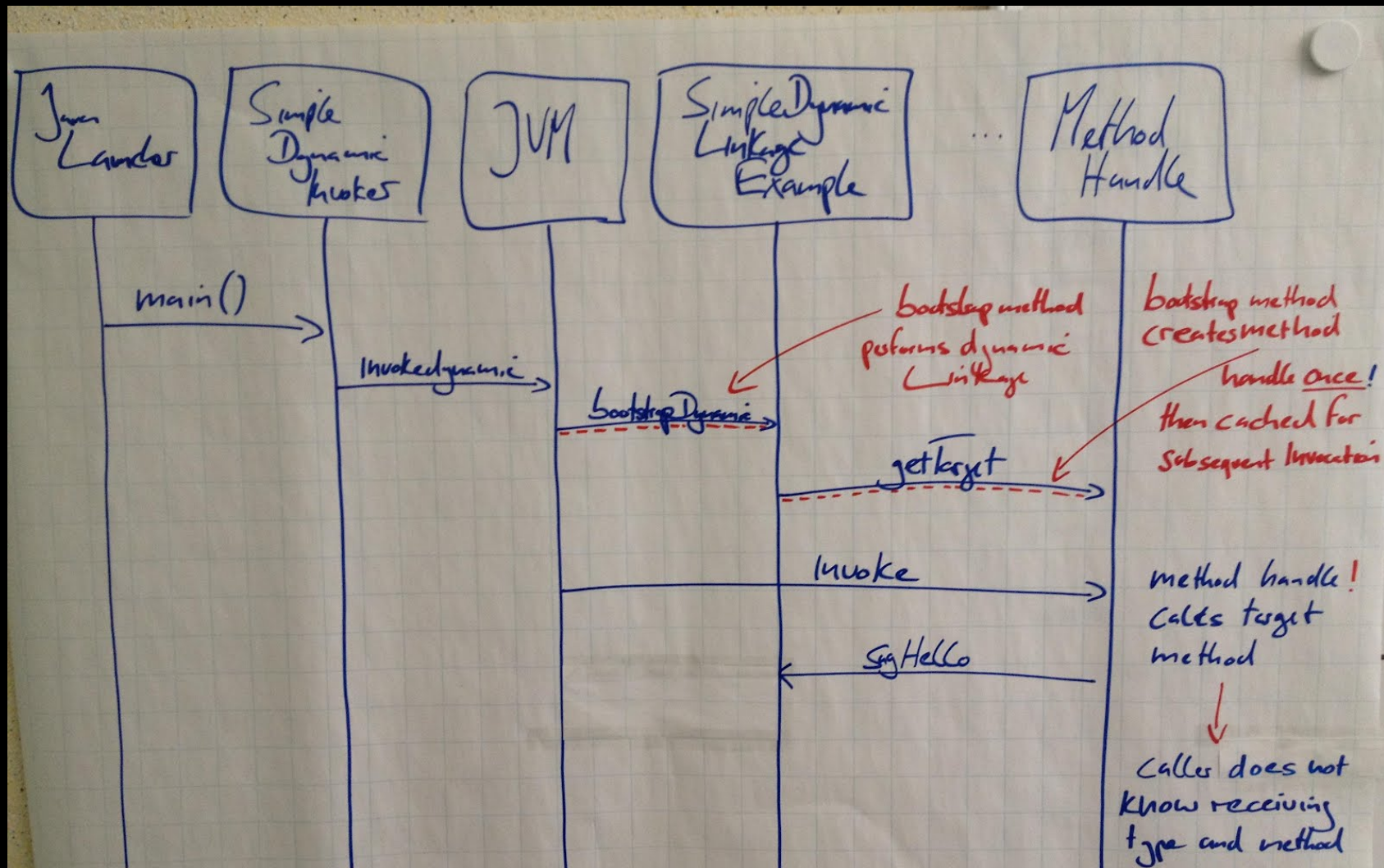
RULE 1: JUST MAKE IT NICE



RULE 2: AND NOT REALISTIC!!!

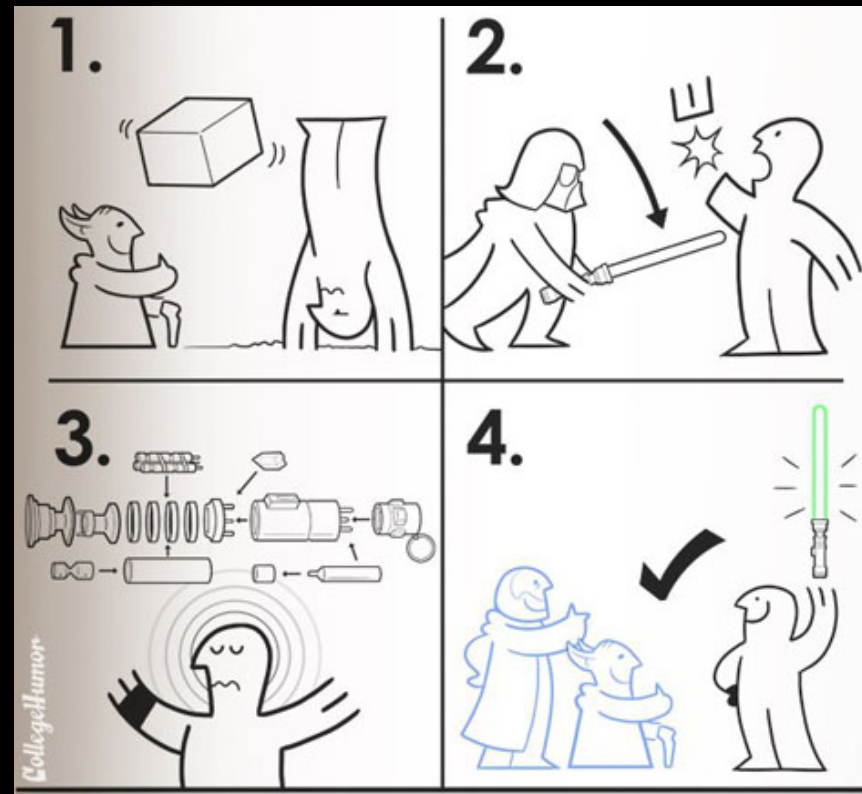
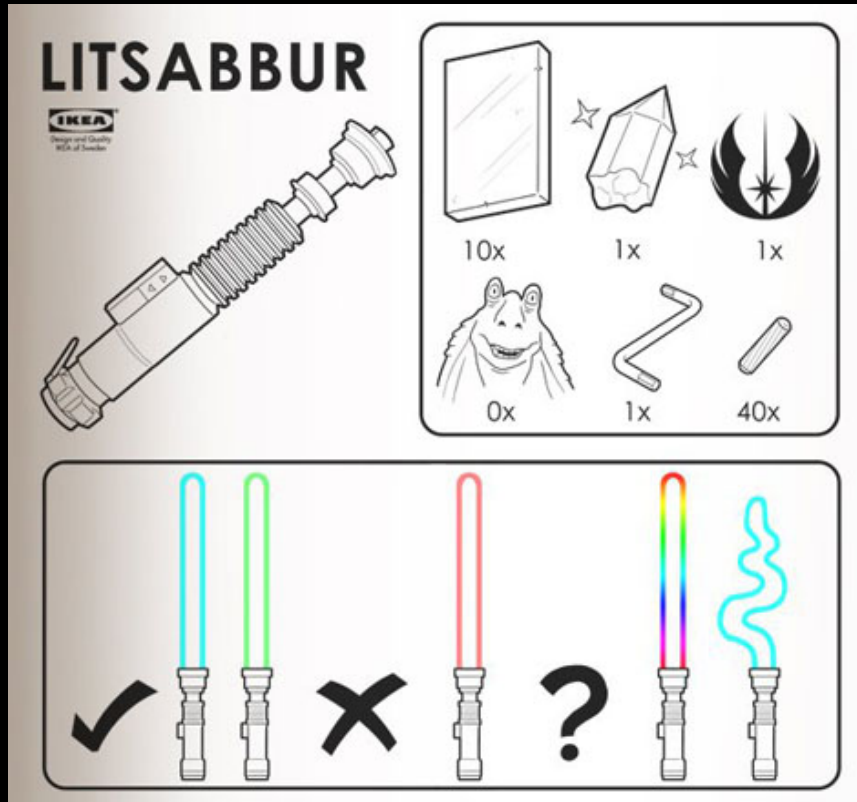
Documentation

Describe the design



Documentation

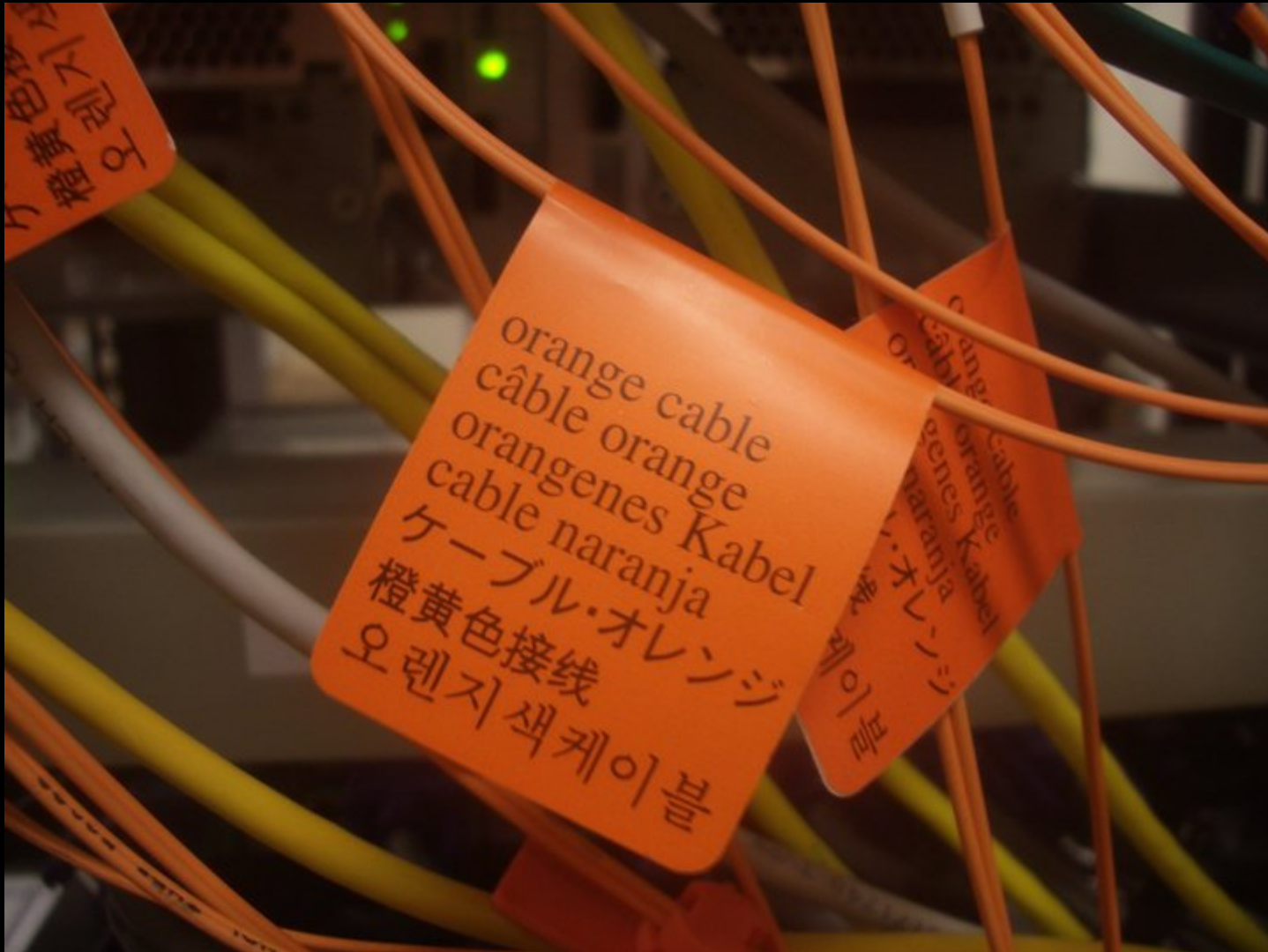
Describe the steps needed to install



Comments

Comments

Commenting the obvious



Comment

Too many comments

```
/* Name: CopyString
```

```
    Purpose:      This routine copies a string from the source  
                  string (source) to the target string (target)
```

```
    Algorithm:    It gets the length of "source" and then copies each character, one at a time, into "target". It uses  
                  the loop index as an array index into both "source" and "target" and increments the loop/array index  
                  after each character is copied.
```

```
    Inputs:       input  The string to be copied
```

```
    Outputs:      output The string to receive the copy of "input"
```

```
    Interface assumptions: none
```

```
    Modification history: none
```

```
    Author:       Dwight K. Coder
```

```
    Date Created: 10/01/04
```

```
    Phone:        (555) 222-2255
```

```
    SSN:          111-22-3333
```

```
    Eye color:    green
```

```
    Maiden name:  none
```

```
    Blood type:   AB-
```

```
    Mother's Maiden Name: Bartlett
```

```
    Favorite car: Pontiac Aztec
```

```
    License Plate: "Tek-ie"
```

```
*/
```

```
public void copyString(String s1, String s2){
```

```
...
```

Comment

Too many comments



Repeating the code

There's even a tool for that

```
public class Class1
{
    private bool m_bEnabled=true;

    /// <summary>
    /// Gets or sets a value indicating whether this <see cref="Class1"/> instance is enabled
    /// </summary>
    public bool Enabled
    {
        get { return m_bEnabled; }
        set { m_bEnabled=value; }
    }

    /// <summary>
    /// Gets a value indicating whether [is this cool]
    /// </summary>
    public bool IsThisCool
    {
        get { return true; }
    }

    /// <summary>
    /// Shows the cool features.
    /// </summary>
    /// <param name="aParameter">a parameter</param>
    /// <param name="anotherParameter">another parameter</param>
    public void ShowCoolFeatures(string aParameter, int anotherParameter)
    {
    }
}
```

Readability

Practices

Code conventions

Use the business language

TDD

Wishful programming

Your tests have to be readable too

Refactoring

Make it correct

Make it clear

Make it concise

Make it fast

Have reviews

Formal

Informal

Pairing



Write pseudocode

Let's make a sandwich

Convert comments to code

Let's improve our sandwich

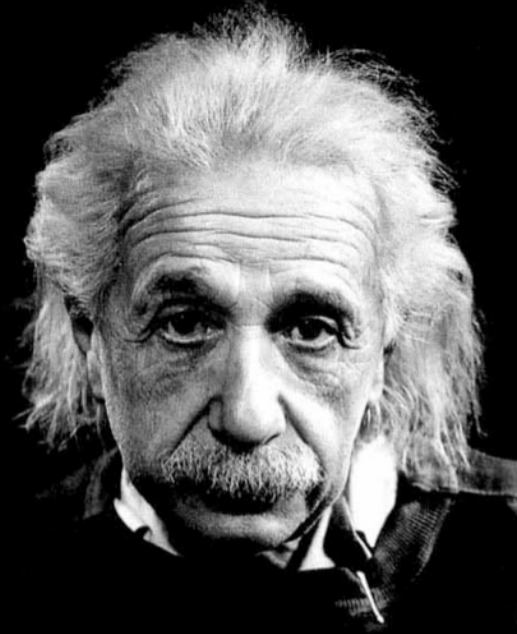
TDD pseudocode

Try it out with our sandwich

Simplicity

*"Everything should be made
as simple as possible,
but not simpler."*

Albert Einstein



Rules of simple design

All tests pass

Intent revealing

No duplication

Small

The boy scout principle



Don't live with broken windows



Readability

Techniques

Typing is not the bottleneck

Writing code is the easy part

Code is read 10 times more than it is written

Blank lines

Add blank lines to separate paragraphs of code

Indentation

Indent logically

Try one indentation per method

Choose good names

SomethingManager

SomethingService

SomethingComponent

Demeter

Just 1 dot per line

```
HttpContext.Current.User.Identity.Name.ToString();
```

Focus on the outcome

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(CreateStrategischeDoelstellingCommand command)
{
    return this.Execute(command)
        .ThenRedirectTo(c => c.Index())
        .EvenWithException();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Update(UpdateStrategischeDoelstellingCommand command)
{
    return this.Execute(command)
        .ThenRedirectTo(c => c.Index())
        .UnlessException(c => c.Edit(command.Id));
}

[HttpPost]
public ActionResult Delete(DeleteStrategischeDoelstellingCommand command)
{
    return this.Execute(command)
        .ThenRedirectTo(c => c.Index())
        .EvenWithException();
}
```

Command-Query separation

Let's code

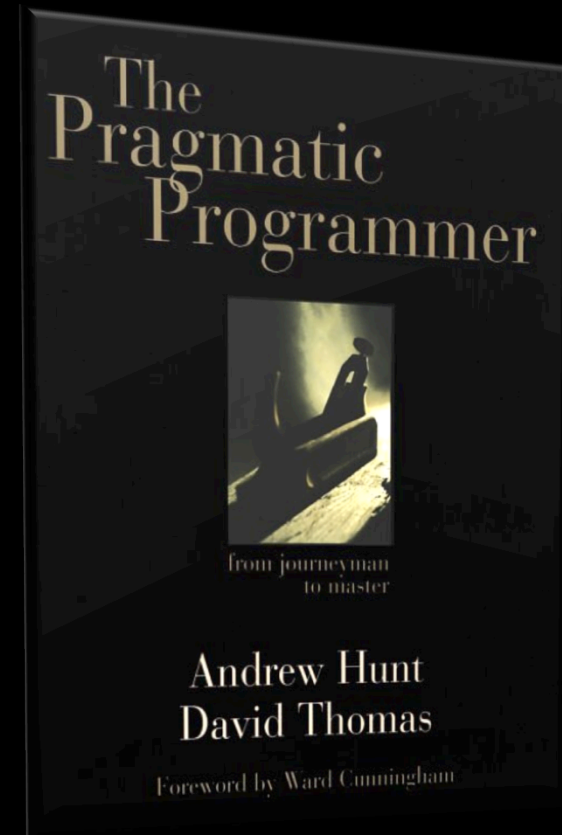
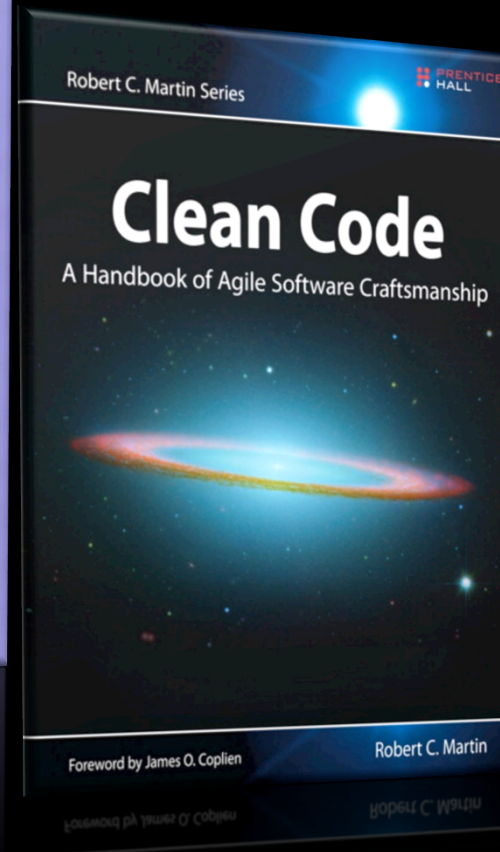
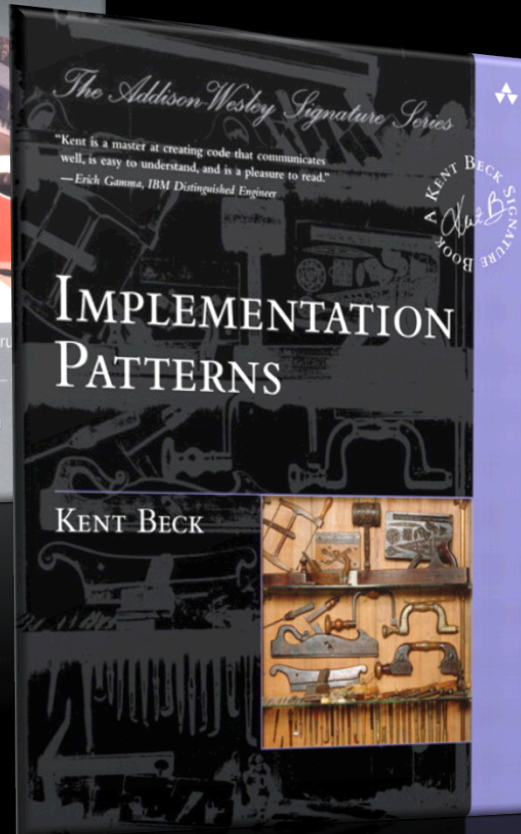
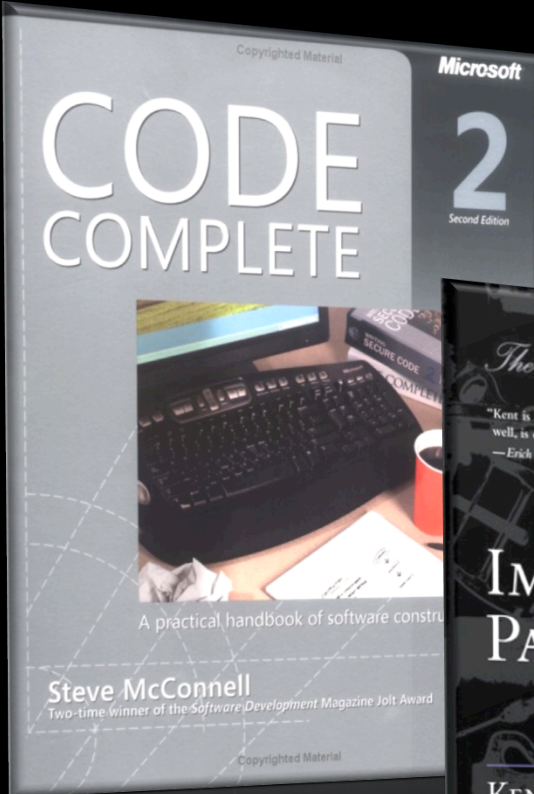
If conditions

Breaking down expressions

The Sieve of Eratosthenes

	2	3	4	5	6	7	8	9	10	Prime numbers
11	12	13	14	15	16	17	18	19	20	
21	22	23	24	25	26	27	28	29	30	
31	32	33	34	35	36	37	38	39	40	
41	42	43	44	45	46	47	48	49	50	
51	52	53	54	55	56	57	58	59	60	
61	62	63	64	65	66	67	68	69	70	
71	72	73	74	75	76	77	78	79	80	
81	82	83	84	85	86	87	88	89	90	
91	92	93	94	95	96	97	98	99	100	
101	102	103	104	105	106	107	108	109	110	
111	112	113	114	115	116	117	118	119	120	

References



Thank you



Readability

Michel Grootjans – SmallSteps



michel.grootjans@gmail.com



#michelgrootjans

