Acronis

Industrial Programming

L3: Coding style. Refactoring

Dual headquarters
in Switzerland and Singapore

Agenda

- Why low level is important (C\C++ examples)
- More about management
- Code style
- Obfuscation

Why low level is important

What about python?

Problems with usual C++ programs

Multi-component program configurations

The preprocessor performs textual inclusion of the contents of <iostream> before real compilation.

It adds 50.000+ lines of code!

```
#include <iostream>

int main()
{
  std::cout << "Hello, world!" << std::endl;
  return 0;
}
```

You didn't pay for that code

...And the most of the code from <iostream> is actually not needed for our "Hello World" program!

7/15

Why low level is important

Problems with usual C++ programs

Multi-component program configurations

Flexibility and control!

The preprocessor performs textual inclusion of the contents of <iostream> before real compilation.

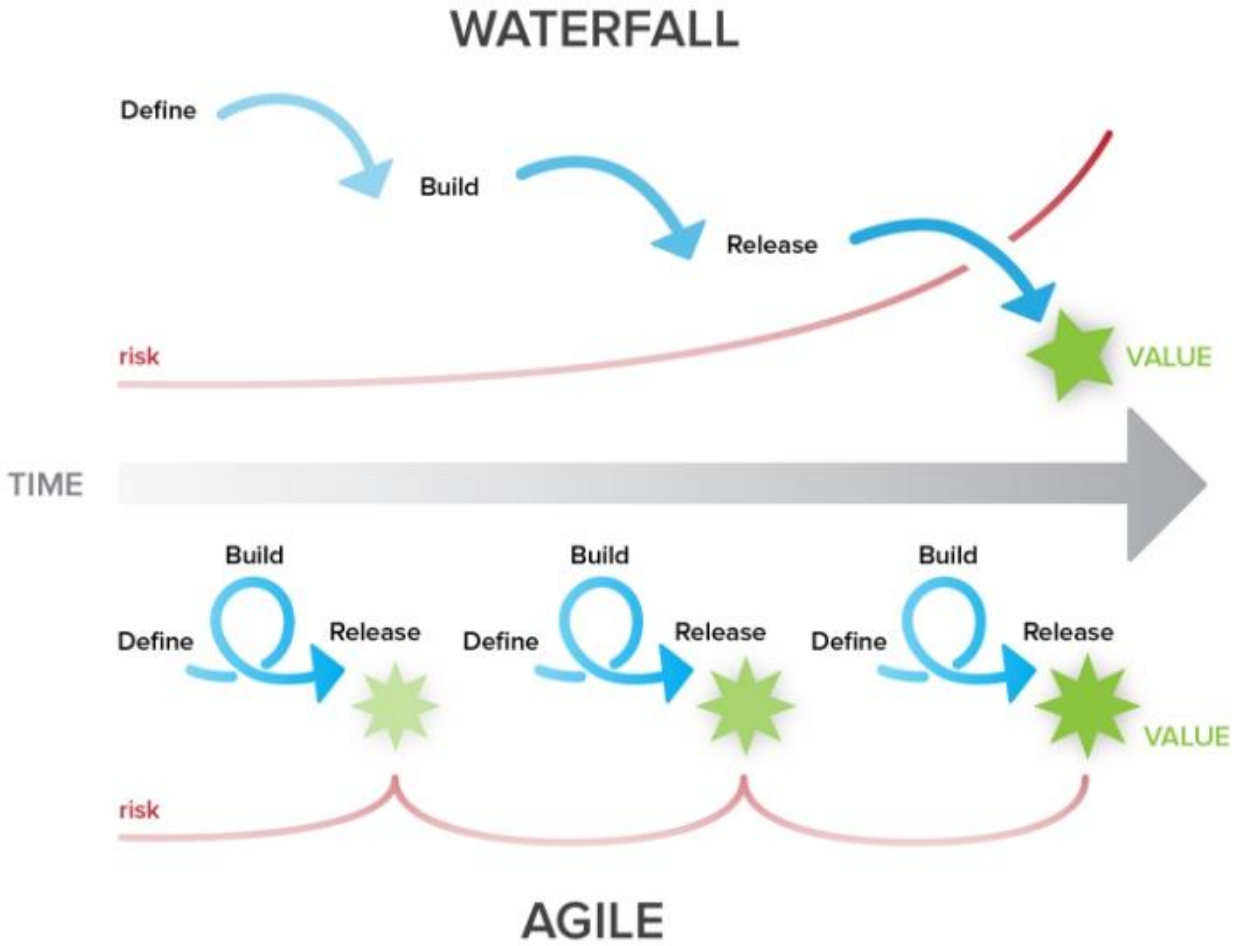
It adds 50.000+ lines of code!

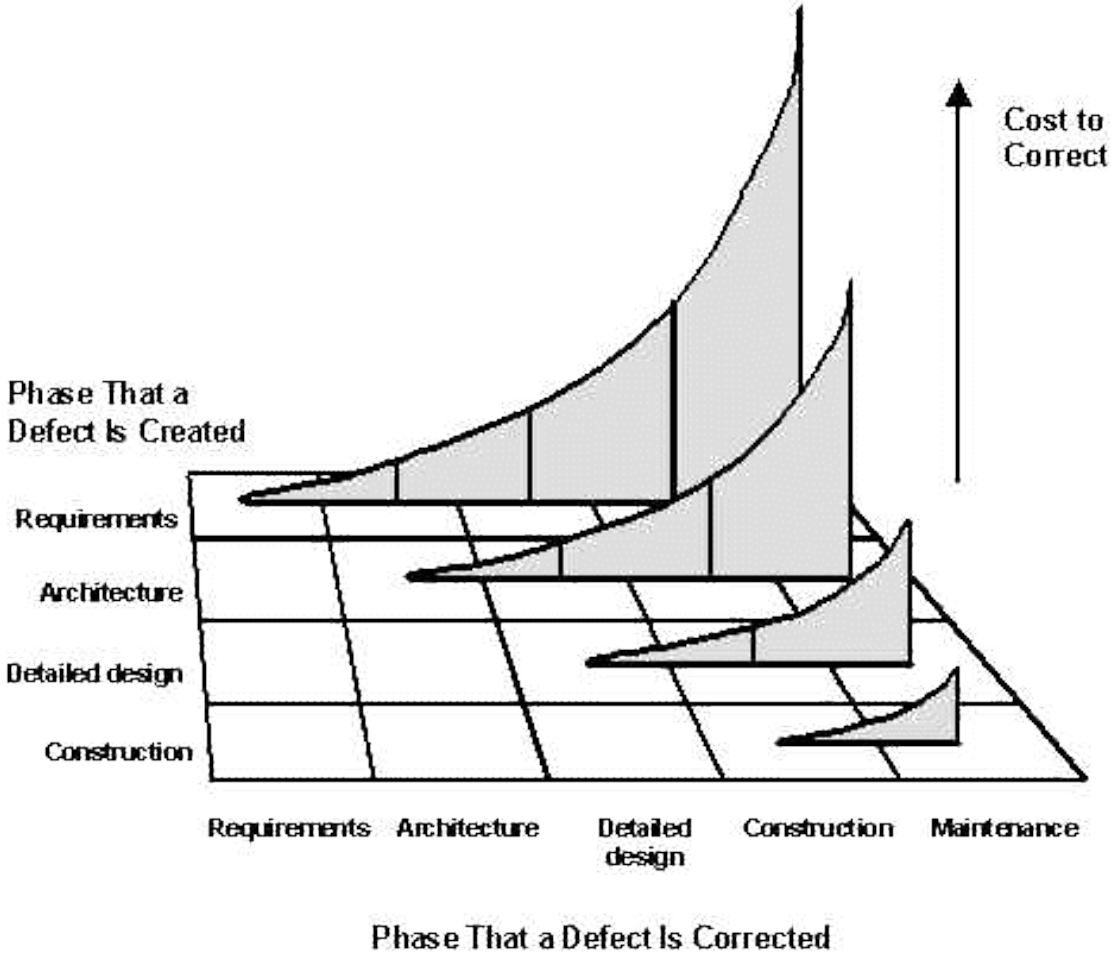
```
#include <iostream>

int main()
{
  std::cout << "Hello, world!" << std::endl;
  return 0;
}
```

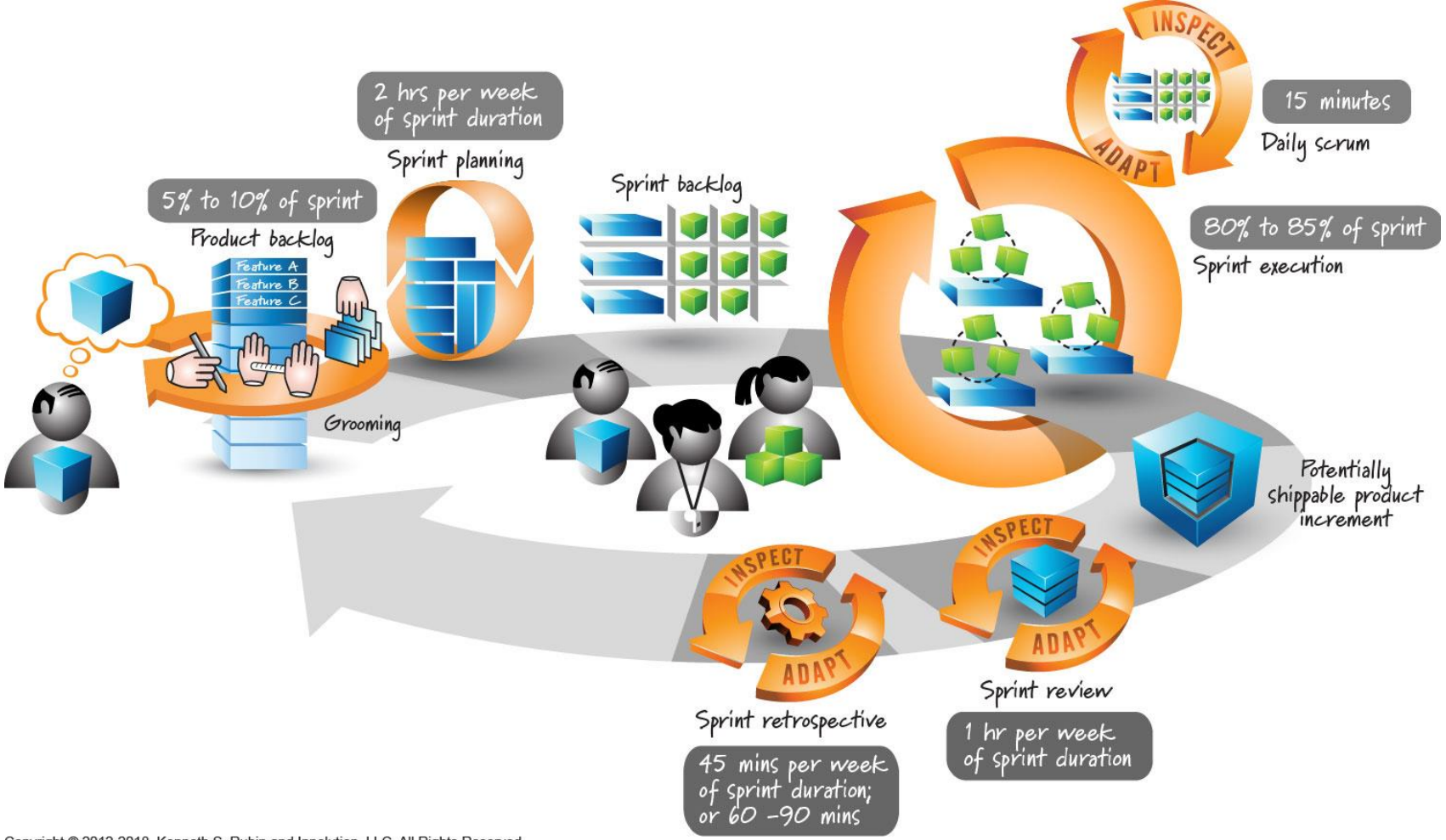
You didn't pay for that code

...And the most of the code from <iostream> is actually not needed for our "Hello World" program!

7/15


WATERFALL
Define
Build
Release
risk
VALUE
TIME
Build
Define
Release
Build
Define
Release
Build
Define
Release
VALUE
risk
AGILE


Cost to
Correct
Phase That a
Defect Is Created
Requirements
Architecture
Detailed design
Construction
Requirements
Architecture
Detailed
design
Construction
Maintenance
Phase That a Defect Is Corrected

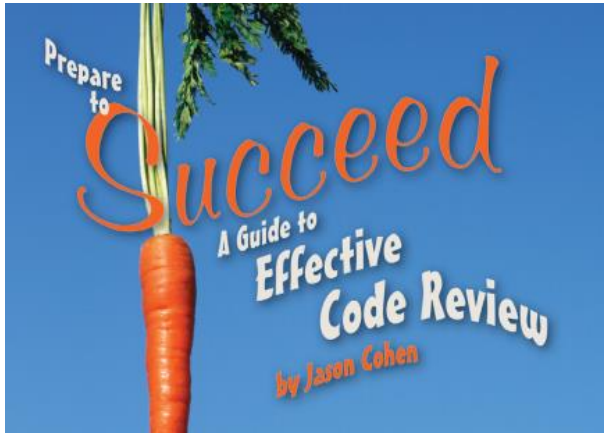


Example of code style and comments

- https://paste.ubuntu.com/p/R8vkBjkbGK/

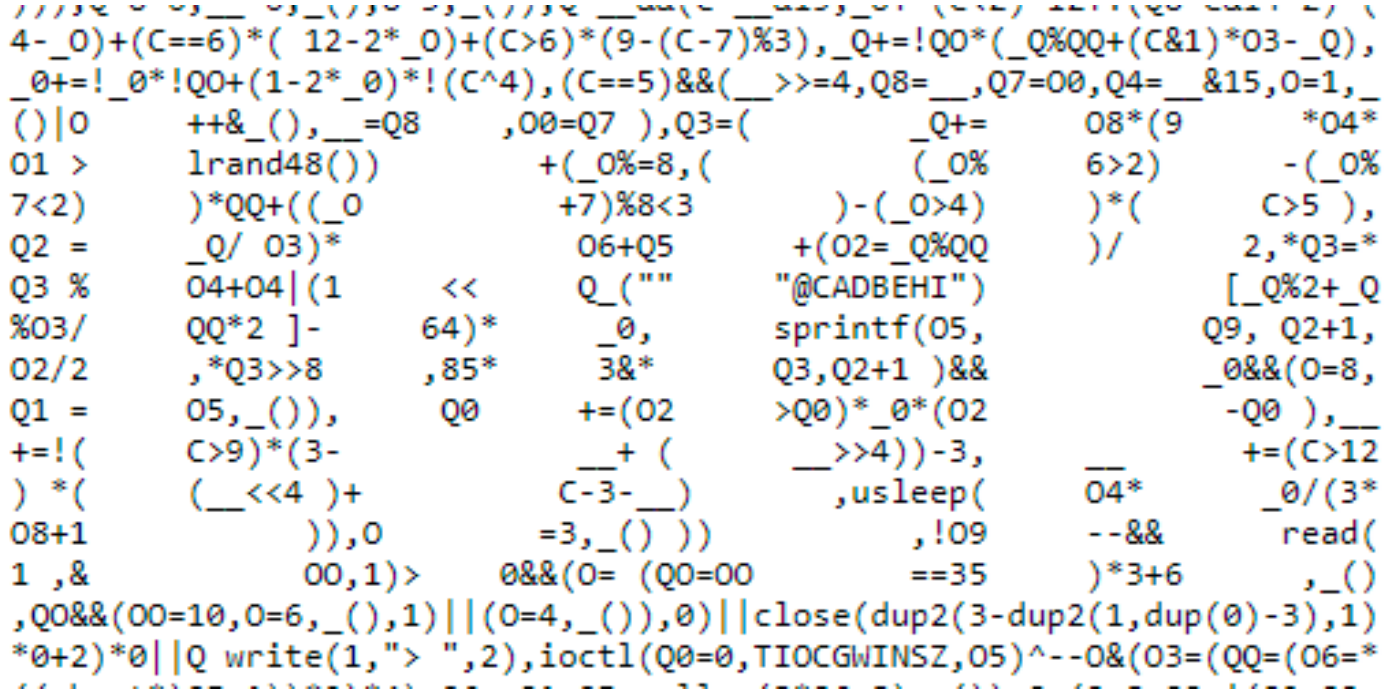
Coding style

1. Names
2. Spaces
3. Parentheses
4. Comments



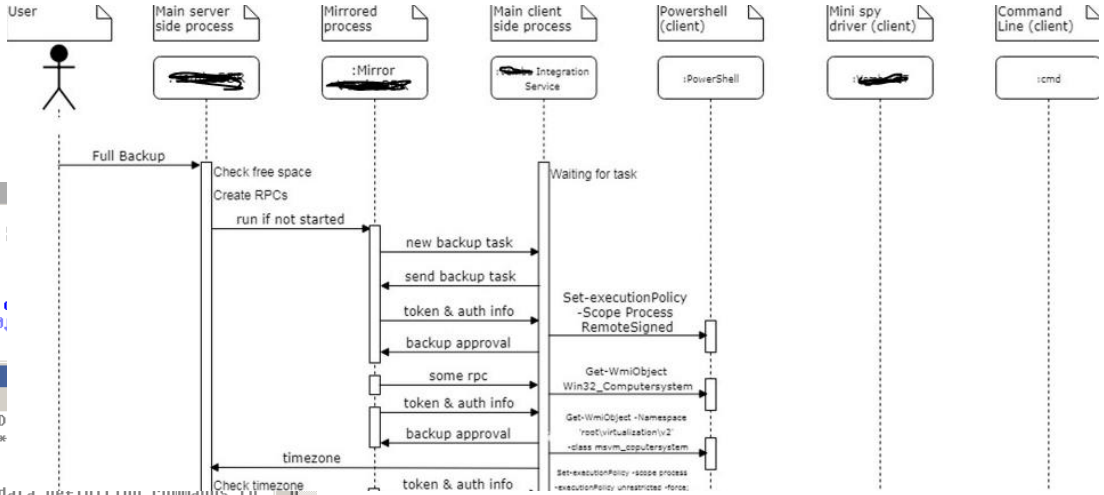

```
// compute the square root of Num using the Newton-Raphson
approximation
r = num / 2;
while ( abs( r - (num/r) ) > TOLERANCE ) {
    r = 0.5 * ( r + (num/r) );
}
System.out.println( "r = " + r );
```

Obfuscation



http://en.wikipedia.org/wiki/International_Obfuscated_C_Code_Contest

http://www.ioccc.org/1988/phillipps.c



IDA View-A

```
.text:004011A1
.text:004011A1 ; ┌──────────────── S U B R O U T I N E ──────────────────
.text:004011A1
.text:004011A1 ; Attributes: bp-based frame
.text:004011A1
.text:004011A1 ; int __cdecl main(int argc,const char **argv,const
.text:004011A1 _main   proc near         ; DATA XREF: .data:0040A0D0
.text:004011A1
.text:004011A1 array   = byte ptr -1Ch
.text:004011A1 size    = dword ptr -8
.text:004011A1 ptr     = dword ptr -4
.text:004011A1 argc    = dword ptr  8
.text:004011A1 argv    = dword ptr  0Ch
.text:004011A1 envp    = dword ptr  10h
.text:004011A1
.text:004011A1         push    ebp
.text:004011A2         mov     ebp, esp
.text:004011A4         add     esp, 0FFFFFFE4h
.text:004011A7         push    esi
.text:004011A8         push    edi
.text:004011A9         mov     eax, init_val
.text:004011AF         mov     [ebp+size], eax
.text:004011B2         mov     al, byte_40A12C
.text:004011B8         mov     byte ptr [ebp+ptr],
.text:004011BB         mov     esi, offset default
.text:004011C0         lea     edi, [ebp+array]
.text:004011C3         mov     ecx, 5
.text:004011C8         rep movsd
.text:004011CA         push    5
.text:004011CC         lea     eax, [ebp+size]
.text:004011CF         push    eax
.text:004011D0         call    adjust_array
.text:004011D5         add     esp, 8
.text:004011D8         movsx   edx, al
.text:004011DB         push    edx
.text:004011DC         push    offset format ; format
.text:004011E1         call    _printf
```

Stack of _main

Edit Jump Search

```
-0000001C ; Ins/D
-0000001C ; D/A/*
-0000001C ; N
-0000001C ; U
-0000001C ; Use data definition commands to
-0000001C ; Two special fields " r" and " s"
-0000001C ; Frame size: 1C; Saved regs: 4; P
-0000001C ;
-0000001C
-0000001C array   db 20 dup(?)      ; char
-00000008 size    dd ?              ; base 10
-00000004 ptr     dd ?              ; offset
+00000000  s      db 4 dup(?)
+00000004  r      db 4 dup(?)
+00000008 argc    dd ?
+0000000C argv    dd ?              ; offset
+00000010 envp    dd ?              ; offset
+00000014
+00000014 ; end of stack variables
```

SP++00000000

Example

- https://paste.ubuntu.com/p/wmbBykxBgN/

Task

1. Choose algorithm and implement obfuscated version	20 min	+ 6 points
2. Share with colleague		
3. Try to guess initial algorithm	10 min	

If your colleague failed to determine what algorithm you`ve implemented	+3 points
If you succeeded to find out what algorithm your colleague have obfuscated	+1 point

Bonus! During the week we will choose the best versions

Rare obfuscated algorithm	+ 2 points
EPIC obfuscated algorithm	+ 5 points
LEGENDARY obfuscated algorithm	+ 10 points

Feedback

https://goo.gl/forms/4xERbxc1BR2hdFJg2

Linkage Phase & Code Bloat Problem

Template instantiation may cause **problems**.

Example: ~~separate~~ independent compilation

T.h

```
template<typename T>
T Max ( T a, T b )
{
   return a>b?a:b;
}
```

File1.cpp

```
#include "T.h"
 . . .
res = Max(x-1,y+2.5);
 . . .
```

File1.obj

Object code with Max_{double}

App.exe

Executable with two copies of Max_{double}

File2.cpp

```
#include "T.h"
 . . .
res = Max(1.0,res);
 . . .
```

File2.obj

Object code with Max_{double}

- Both compilations produce **the same** function-by-template
- Executable contains **two copies** of Max_{double} ("**code bloat**")

8/15

From the Compiler Construction course