Characterization Testing : regaining control of your *Legacy Code*

FÉLIX-ANTOINE BOURBONNAIS
B.ING., M.SC, PSM

&

PASCAL ROY
ING., CSM, PSM, PMP

Version: Mai 2017 (DEV)

Think how *Legacy code* affects your work in everyday life…

Imagine a tool that helps you explore, understand and refactor it without fear.

… a nice dream, isn't?

ELAPSE Technologies

welcome

Félix-Antoine Bourbonnais

B.ing., PSM, M.Sc.

Pascal Roy

Ing., PSM, CSM, PMP

ELAPSE technologies

Training

Mentoring

Diagnostics

Talks

> We are

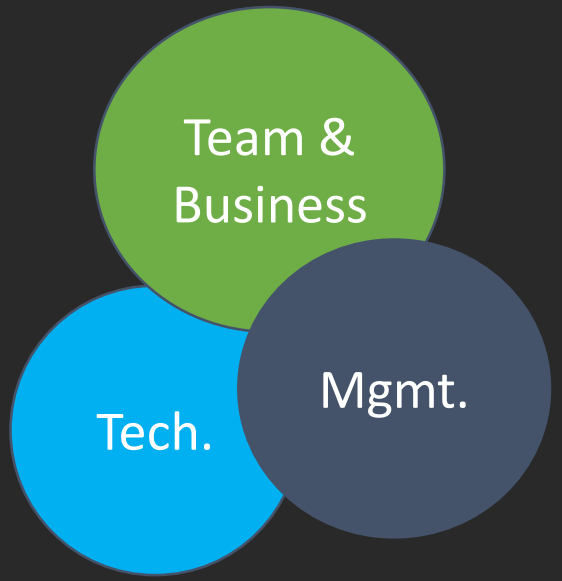
Speakers

Trainers

Coaches & Mentors

> What we do

- Scrum
- Agile QA
- BDD
- Automated tests
- TDD
- Emergent architecture
- DDD
- ...



- Strategic advice
- Project management
- Agility

3WELAPSE technologies

Legacy Code

Am I the only one to have *Legacy Code* ?

ELAPSE Technologies

What is *Legacy Code* ?

Code that is difficult to modify and evolve.

Regardless how old it is or why it got this way.

ELAPSE Technologies

Other popular definitions …

- Code written by others
- Code nobody wants to touch anymore
- Code no longer supported by the people who wrote it
- Code that could be rewritten using better coding practices, tools or languages
- …

Code without tests

Michael Feathers,
Working Effectively with Legacy Code

The negative impacts of *Legacy Code*

What to do with my *Legacy Code* ?

Two basic strategies…

ELAPSE Technologies

Fear : software development’s worst enemy

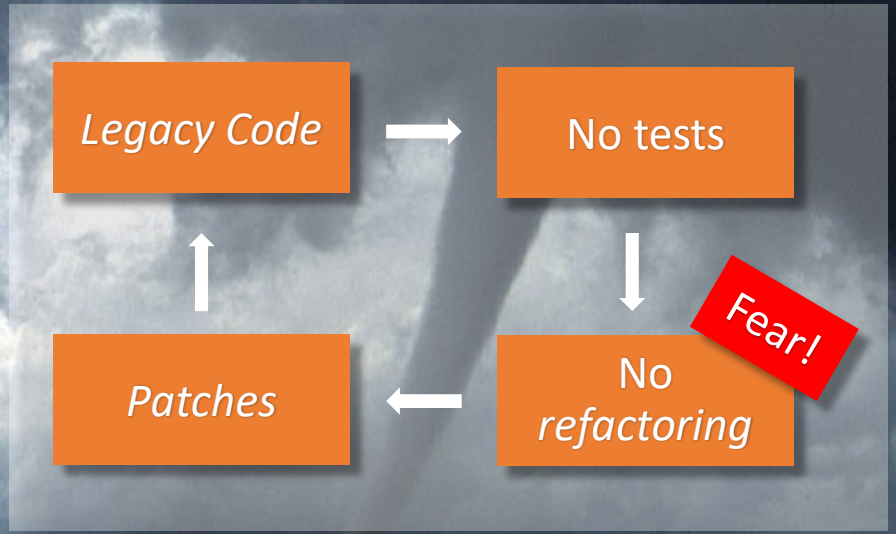
ELAPSE Technologies

Tired of the **continuous stress** of delivery, of ***endless debugging sessions***, of the fear of breaking something?

ELAPSE Technologies


Please, give me a
new project
!@/$%!/%

The cycle of death…



A strategy :

Renovating

Why not

rejuvenate

your code?

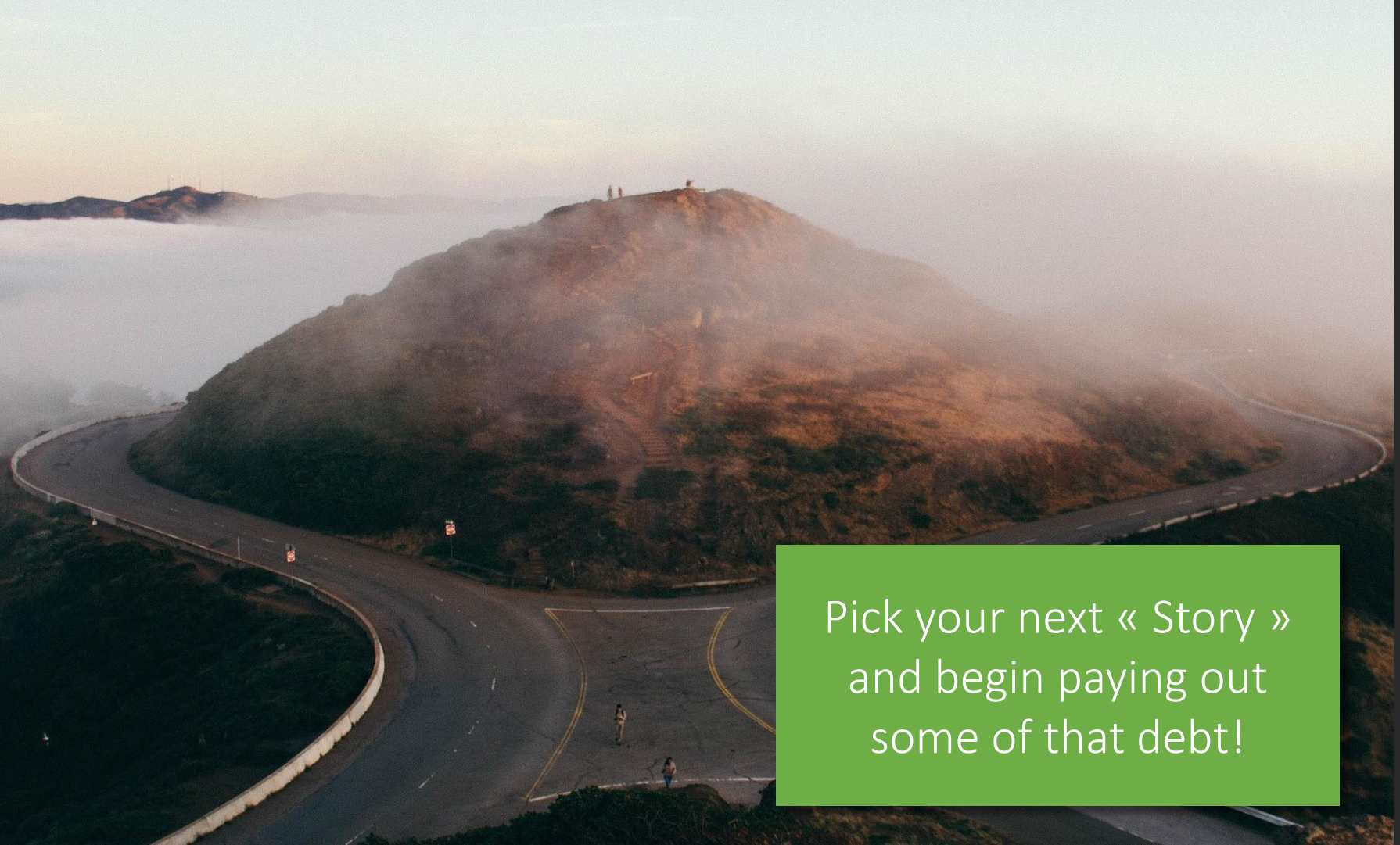
ELAPSE Technologies

Correctly equipped,

yes you can !

ELAPSE Technologies

Gradually,
while still producing
business value



Pick your next « Story » and begin paying out some of that debt!

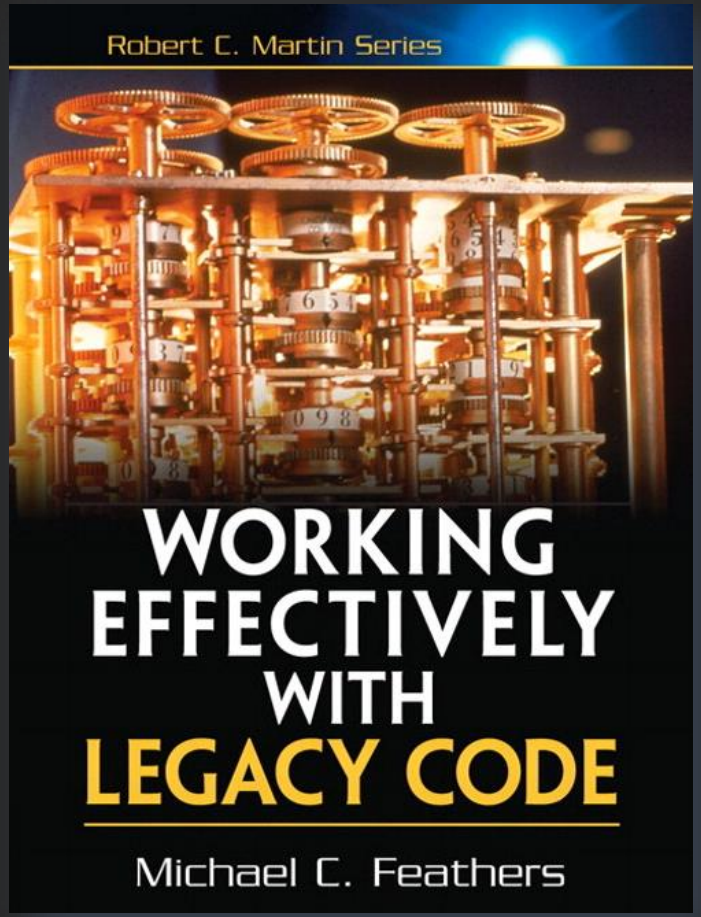
ELAPSE Technologies

A tool:

The characterization test

Characterization test

ELAPSE Technologies


Robert C. Martin Series
WORKING EFFECTIVELY WITH LEGACY CODE
Michael C. Feathers

“

A characterization test describes the actual behavior of an existing piece of code.

- Michael Feathers

”

To…

Overcome *fear*!

Understand and document what the code does

+

Protect against unintended changes when refactoring

ELAPSE Technologies

Writing a characterization test

5 steps for writing a characterization test

1. Find and isolate a piece of code that you need to analyze or modify
2. Write a test that uses that code, use an assertion you know will fail
3. Execute the test and let it tell you the actual behavior
4. Change the test assertion and the naming to match the actual behavior
5. Repeat…

An example of Legacy Code?

```
public class SalesUtil {
  double BQ = 1000.0;
  double BCR = 0.20;
  double OQM1 = 1.5;
  double OQM2 = OQM1 * 2;

  public double calculate(double tSales) {
    if (tSales <= BQ) {
      return tSales * BCR;
    } else if (tSales <= BQ * 2) {
      return (BQ) * BCR + (tSales - BQ) * BCR * OQM1;
    } else {
      return (BQ) * BCR +
            (tSales - BQ) * BCR * OQM1 +
            (tSales - BQ * 2) * BCR * OQM2;
    }
  }
}
```

WTF?

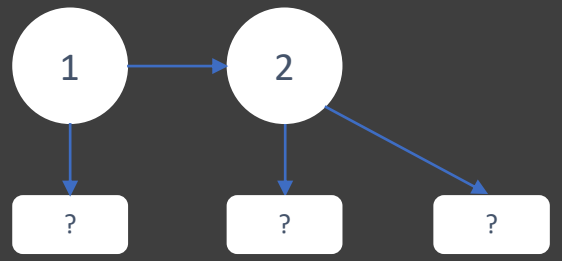
WTF?

WTF?

http://s3.amazonaws.com/giles/demons_010609/wtfm.jpg

Step 1: identify a piece of code to characterize

```
@Test
public void test... {
   assert(...)
}
```



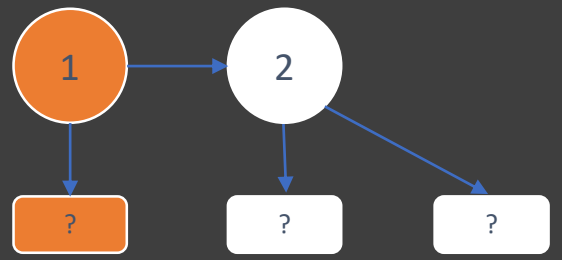

```
public class SalesUtil {
  double BQ = 1000.0;
  double BCR = 0.20;
  double OQM1 = 1.5;
  double OQM2 = OQM1 * 2;

  public double calculate(double tSales){
    if (tSales <= BQ) {
      return tSales * BCR;
    } else if (tSales <= BQ * 2) {
      return (BQ) * BCR +
          (tSales - BQ) * BCR * OQM1;
    } else {
      return (BQ) * BCR +
        (tSales - BQ) * BCR * OQM1 +
        (tSales - BQ * 2) * BCR * OQM2;
    }
  }
}
```

1

2

Step 2: write a failing assertion

```
@Test
public void testCalculate() {
  assertEquals(
    0.0,
    SalesUtil.calculate(1000.0)
  );
}
```



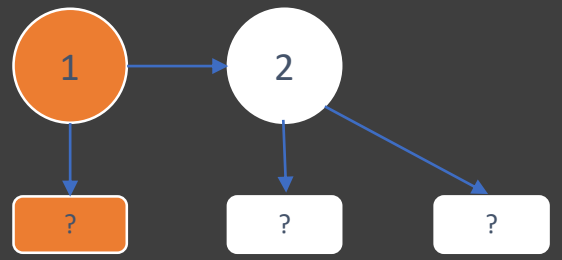

```
public class SalesUtil {
  double BQ = 1000.0;
  double BCR = 0.20;
  double OQM1 = 1.5;
  double OQM2 = OQM1 * 2;

  double calculate(double tSales) {
    if (tSales <= BQ) {
      return tSales * BCR;
    } else if (tSales <= BQ * 2) {
      return (BQ) * BCR +
        (tSales - BQ) * BCR * OQM1;
    } else {
      return (BQ) * BCR +
        (tSales - BQ) * BCR * OQM1+
        (tSales - BQ*2)*BCR * OQM2;
    }
  }
}
```

1

2

Step 3: execute the test
+ find out the actual behavior is

```
@Test
public void testCalculate() {
  assertEquals(
    0.0,
    SalesUtil.calculate(1000.0)
  );
}
```



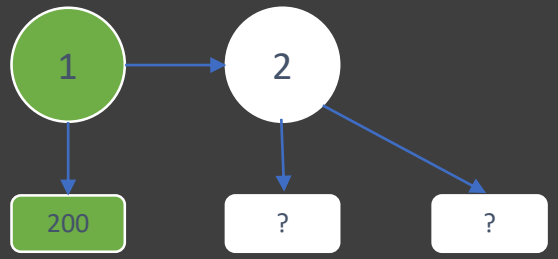

```
public class SalesUtil {
  double BQ = 1000.0;
  double BCR = 0.20;
  double OQM1 = 1.5;
  double OQM2 = OQM1 * 2;

  double calculate(double tSales) {
    if (tSales <= BQ) {
      return tSales * BCR;
```

> junit.framework.AssertionFailedError:
expected:<0.0> **but was:<200.0>**

```
    }
  }
}
```

Step 4: Replace the assertion by the actual behavior

```
@Test
public void
lessThanBaseQuota_useBaseCommissionRate() {
  assertEquals(
    200.0,
    SalesUtil.calculate(100
  );
}
```



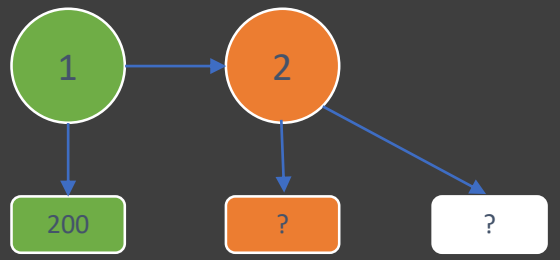

```
public class SalesUtil {
  double BQ = 1000.0;
  double BCR = 0.20;
  double OQM1 = 1.5;
  double OQM2 = OQM1 * 2;

  double calculate(double tSales) {
    if (tSales <= BQ) {
      return tSales * BCR;
    } else if (tSales <= BQ * 2) {
      return (BQ) * BCR +
        (tSales - BQ) * BCR * OQM1;
    } else {
      return (BQ) * BCR +
        (tSales - BQ) * BCR * OQM1+
        (tSales – BQ*2)*BCR * OQM2;
    }
  }
}
```

1

2

Step 5: Repeat

```
...

@Test
public void testCalculate () {
  assertEquals(
    0.0,
    SalesUtil.calculate(2000.0)
  );
}
```




```
public class SalesUtil {
  double BQ = 1000.0;
  double BCR = 0.20;
  double OQM1 = 1.5;
  double OQM2 = OQM1 * 2;

  double calculate(double tSales) {
    if (tSales <= BQ) {
      return tSales * BCR;
    } else if (tSales <= BQ * 2) {
      return (BQ) * BCR +
        (tSales - BQ) * BCR * OQM1;
    } else {
      return (BQ) * BCR +
        (tSales - BQ) * BCR*OQM1+
        (tSales – BQ*2) * BCR*OQM2;
      }
    }
  }
```

1

2

Watch out for :

« While we're at it, we might as well… » !

Work only on what you need to modify now.

Demo

We really don’t have the time for this ?!?

How long does it take to understand a piece of Legacy code before you can change it (reasonably safely)?

Characteristics of a characterization test

How is a CT different?

Unit test	Characterization test
• Specifies **required** behavior	• Specifies **actual** behavior
• **Known** behavior and **new** code	• Legacy code, **Lost or forgotten** behavior
• **Permanent**	• **Temporary**

Will end-to-end tests around my application help me characterize my code?

Rewrite or refactoring?

Refactoring is <u>not</u> a rewrite on a small scale!

ELAPSE Technologies

Why are managers/customers/project managers so afraid of refactoring?

Watch out for *Big Bang* refactoring !

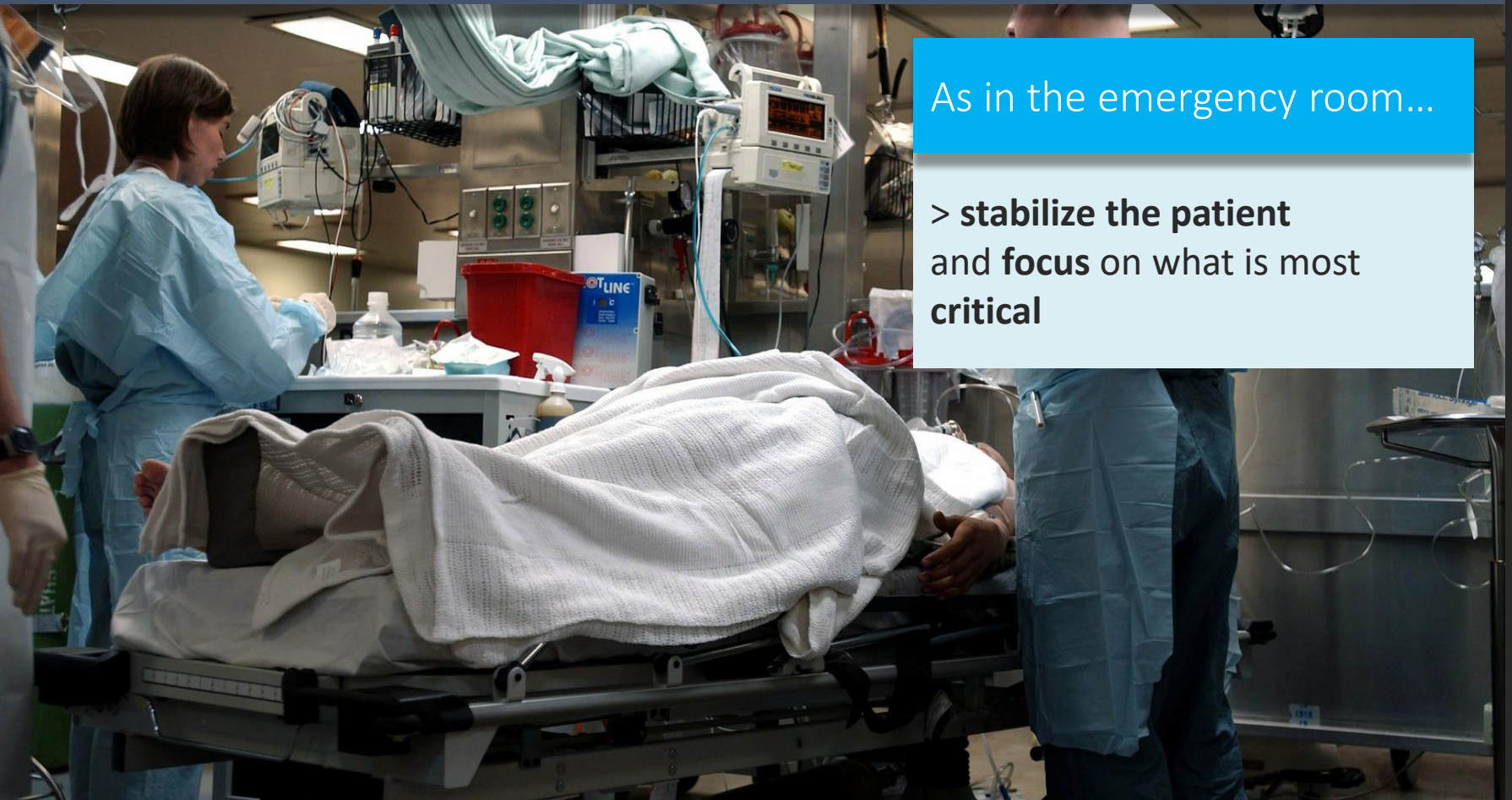
Rewrite or refactoring ?!?

Rewrite	Renovate / Rejuvenate
A big expense	Regular payments (debt)
Big Bang	Step by Step
Very high risk	Lower risk
No new business value	Still produce value

Sometimes necessary!

ELAPSE Technologies

A strategy



As in the emergency room…

> **stabilize the patient**
and **focus** on what is most **critical**

ELAPSE Technologies

Let what you are currently working on guide your efforts

3ELAPSE Technologies

Conclusion

Next… Refactoring techniques

This talk was about **understanding** and **securing** your *Legacy Code*….

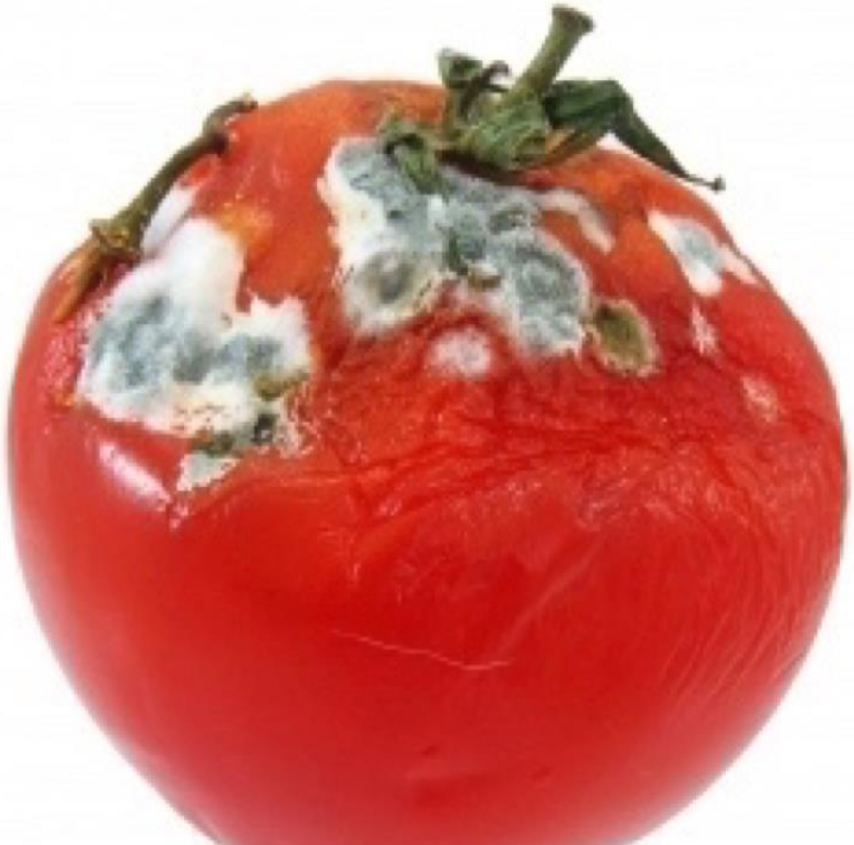
Now, you can learn **how to mercilessly renovate** your *Legacy Code*:

- Sprout Methods/Classes
- Instance Delegator
- Extract to Method
- …

ELAPSE Technologies

Maintainability: a great challenge for the future

ELAPSE Technologies



Code rot is avoidable

ELAPSE Technologies

Legacy code can kill the flame!

Although our first joy of programming may have been intense, the misery of dealing with legacy code is often sufficient to extinguish that flame.

Michael Feathers,
Working Effectively with Legacy Code

Characterization testing…

Another tool to add to your toolbox!

Image de http://beinweb.fr/wp-content/uploads/2014/04/boite-a-outils-entrepreneurs.jpg

3ELAPSE Technologies

« Legacy code » can be treated !

ELAPSE Technologies

Thank you.

All our conferences (french)

conferences.elapsetech.com

Slides and references (french)

conferences.elapsetech.com
/legacy-tests-caracterisation

Félix-Antoine Bourbonnais

Pascal Roy

Site
elapsetech.com

Twitter
@fbourbonnais

Email
fbourbonnais@elapsetech.com
pascalroy@elapsetech.com

LinkedIn
linkedin.com/in/fbourbonnais
ca.linkedin.com/in/roypa