Welcome to Mission Critical Systems

EECS 4315

www.eecs.yorku.ca/course/4315/

Instructor

- Name: Franck van Breugel
- Email: franck@eecs.yorku.ca
- Office: Lassonde Building, room 3046 and 1012U
- Office hours: Lassonde Building, room 3046, Wednesdays 10:30-11:30 or by appointment

Evaluation

- Bi-weekly quizzes (5% each)
- Project (40%)
- Final exam (30%)

Labs

- Lassonde Building, room 1006C
- Mondays, 11:00-12:00
- The first lab will be held this week.
- Some of the quizzes will be held during the labs.

Quizzes

- The quizzes will take place during labs or lectures and will be announced in advance.
- Students with a documented reason for missing a quiz, such as illness, compassionate grounds, etc., will have the weight of the missed quiz(zes) shifted to the final exam. This final exam will cover all the material covered in the course.
- During quizzes, students are expected to do their own work. Looking at someone else's work during a quiz, talking during a quiz, using aids not permitted (such as a phone) during a quiz, and impersonation are all examples of academically dishonest behaviour.

Drop Deadline

March 10

Until this date you can drop the course without getting a grade for it.

https://registrar.yorku.ca/enrol/dates/fw16
contains important dates.

Academic Honesty

"If you put your name on something, then it is your work, unless you explicitly say that it is not."

Examples of academic dishonesty include

- copying code,
- looking at someone else's work during a quiz,
- talking during a quiz,
- using aids not permitted (such as a phone) during a quiz,
- impersonation.

Read http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/ for more details.

Academic Honesty

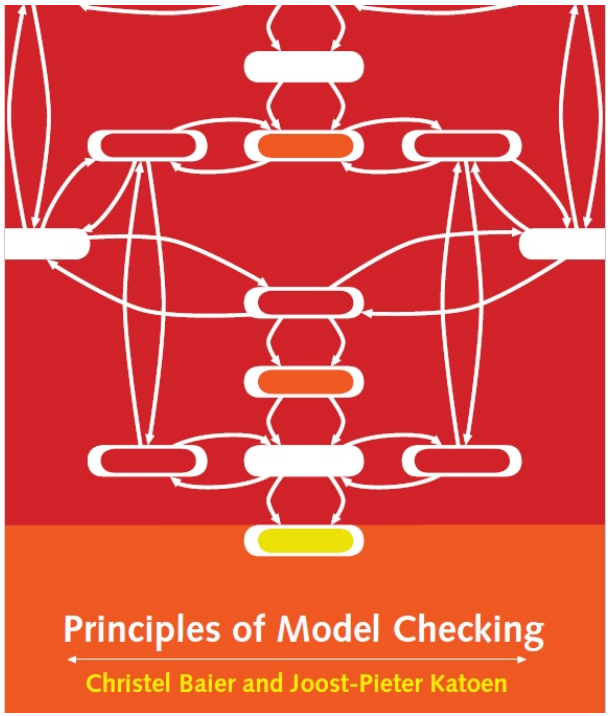
Academic honest behaviour of students increases the value of your degree.

- The instructors will do their best to design quizzes and policies that promote honest behaviour.
- The students are expected to behave honestly.

Textbook

Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

Other reading material will be provided.



Study the Textbook

The textbook is required for this course.

Studying only the slides and your lecture notes is not sufficient. There will be questions on quizzes and final exam about material that is not covered in class. Therefore, you should study the textbook.

Although you need to memorize some material, most of the material you have to understand.

Students are expected to . . .

- attend the lectures (3 hours per week)
- attend the lab (1 hour per week)
- prepare for the lab (2 hours per week)
- study the textbook and other reading material (3 hours per week)

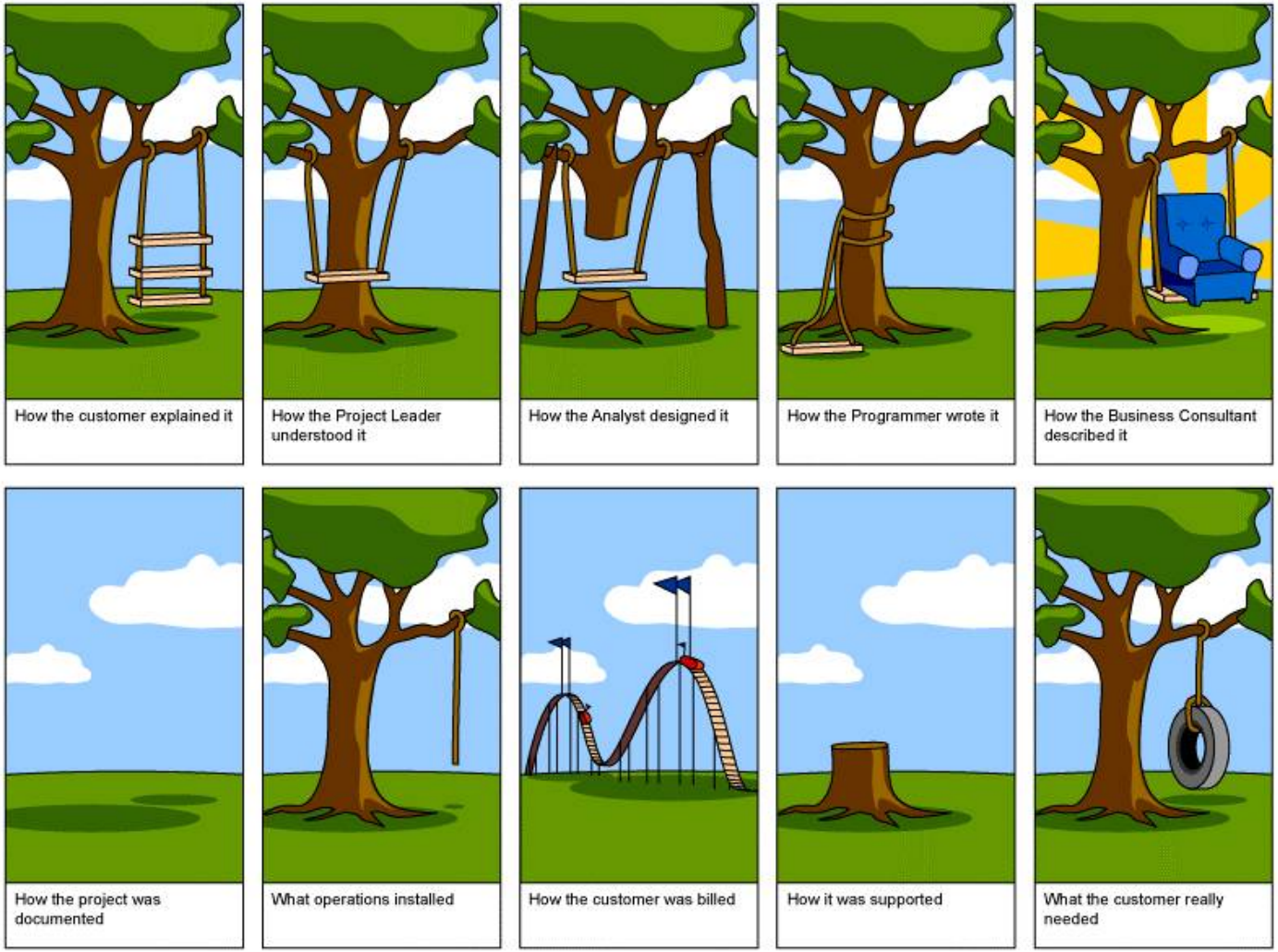
Bugs are Everywhere

EECS 4315

www.eecs.yorku.ca/course/4315

What is Verification?

"Have you made what you were trying to make?"



Source: Paragon Innovations

What is Verification?

"Have you made what you were trying to make?"

Does the hardware/software system satisfy (all the properties of) its specification?

In contrast to . . .

"Have you made the right thing?"

Is the specification of the system correct?

which is also known as *validation*.

Why do we Verify?

Bugs are everywhere.

Source: Bruce Campbell

Classic Bug

1968 Brazilian Beetle

Source: Dan Palatnik

Classic Bug

"A clear example of the risks of poor programming and verification techniques is the tragic story of the Therac-25—one in a series of radiation therapy machines developed and sold over a number of years by Atomic Energy Canada Limited (AECL). As a direct result of inadequate programming techniques and verification techniques, at least six patients received massive radiation overdoses which caused great pain and suffering and from which three died."

Source: unknown

Peter H. Roosen-Runge. Software Verification Tools. 2000.

Classic Bug

"A computer malfunction at Bank of New York brought the Treasury bond market's deliveries and payments systems to a near standstill for almost 28 hours . . . it seems that the primary error occurred in a messaging system which buffered messages going in and out of the bank. The actual error was an overflow in a counter which was only 16 bits wide, instead of the usual 32. This caused a message database to become corrupted. The programmers and operators, working under tremendous pressure to solve the problem quickly, accidentally copied the corrupt copy of the database over the backup, instead of the other way around."

Wall Street Journal, November 25, 1985

Source: unknown

Classic Bug

"To correct an anomaly that caused inaccurate results on some high-precision calculations, Intel Corp. last week confirmed that it had updated the floating-point unit (FPU) in the Pentium microprocessor. The company said that the glitch was discovered midyear and was fixed with a mask change in recent silicon. "This was a very rare condition that happened once every 9 to 10 billion operand pairs," said Steve Smith, a Pentium engineering manager at Intel."

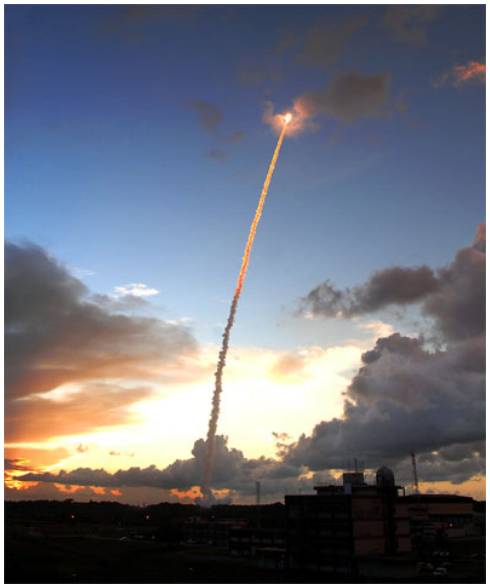
EE Times, November 7, 1994

Source: Konstantin Lanzet

Classic Bug

"On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 meters, the launcher veered off its flight path, broke up and exploded. . . . The reason why the active SRI 2 did not send correct attitude data was that the unit had declared a failure due to a software exception. . . . The data conversion instructions (in Ada code) were not protected from causing an operand error, although other conversions of comparable variables in the same place in the code were protected."

Report of the Ariane Inquiry Board



Source: unknown

Bug of the 21st Century

2012 Beetle

Source: unknown

Bug of the 21st Century

The Toronto Skyline

Source: unknown

Bug of the 21st Century

The Toronto Skyline on August 14, 2003

Source: unknown

Bug of the 21st Century

"The first known death caused by a self-driving car was disclosed by Tesla Motors on Thursday, a development that is sure to cause consumers to second-guess the trust they put in the booming autonomous vehicle industry. . . . Against a bright spring sky, the car's sensors system failed to distinguish a large white 18-wheel truck and trailer crossing the highway, Tesla said. The car attempted to drive full speed under the trailer, with the bottom of the trailer impacting the windshield of the Model S, Tesla said."

Danny Yadron and Dan Tynan, The Guardian, July 1, 2016

Source: Daily Mail

Why do we Verify?

Ask Jessie J!

Source: unknown

It's all about the money money money.

What's the Price Tag?

Bank of New York bug: $ 5 million
Pentium bug: $ 475 million
Ariane bug: $ 500 million
Blackout bug: $ 6 billion
Denver bug: $ 300 million
Knight bug: $ 440 million

"The cost of software bugs to the U.S. economy is estimated at $ 60 billion per year."

National Institute of Standards and Technology, 2002

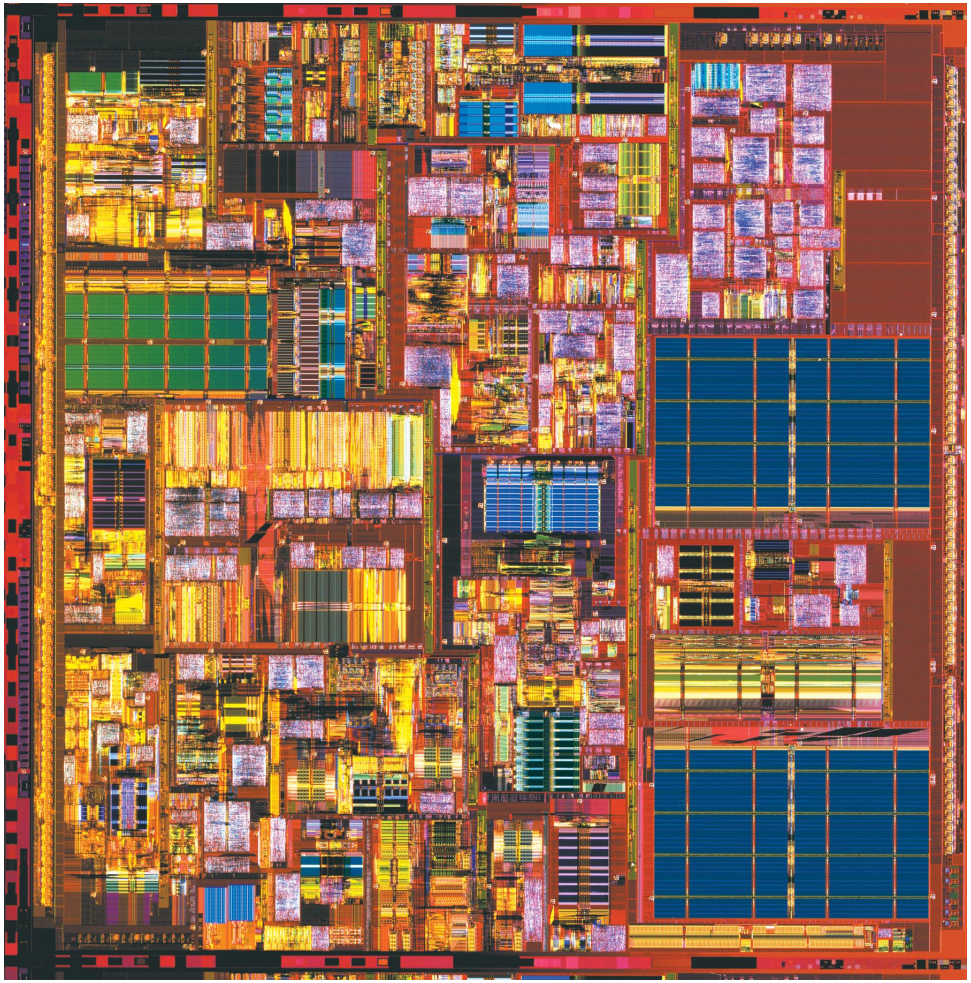
"Wages-only estimated cost of debugging: US $312 billion per year."

Tom Britton, Lisa Jeng, Graham Carver, Paul Cheak and Tomer Katzenellenbogen, 2013

Why are Bugs introduced?

Hardware and software systems are among the most complex artifacts ever produced by humans.

Pentium 4 Microprocessor



Source: unknown

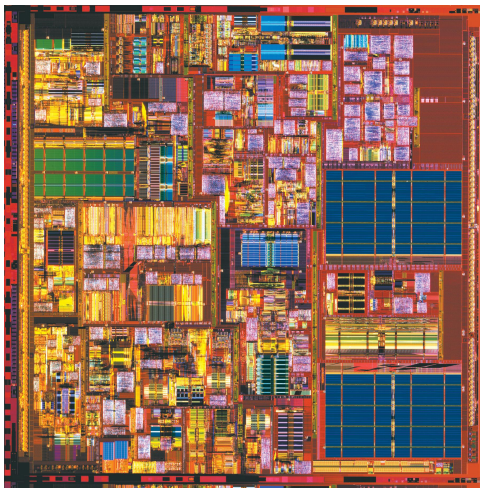
- transistors:
  55 million
- area:
  146 mm^2
- clock rate:
  3.2×10^9 Hz

If . . .

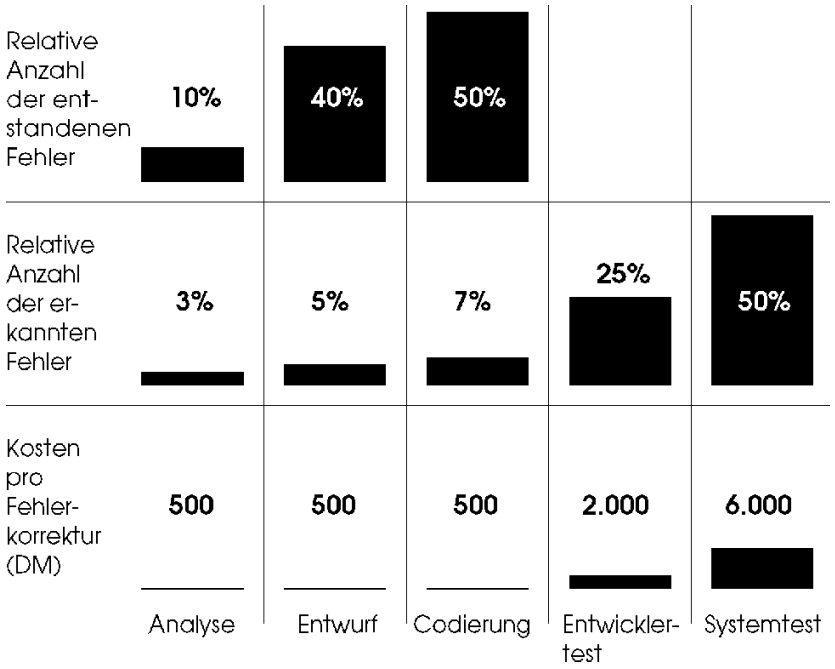
. . . the connections on a microprocessor were roads in Scotland, . . .

Area of microprocessor: 146 mm^2
Area of Scotland: 80,234 km^2
Scale: 12 mm / 283 km $\approx$ 1 / 14,150,000

. . . then, since each connection is 0.13μm wide, the roads in Scotland would be 1.84 m wide, 1.84 m apart and eight layers deep!

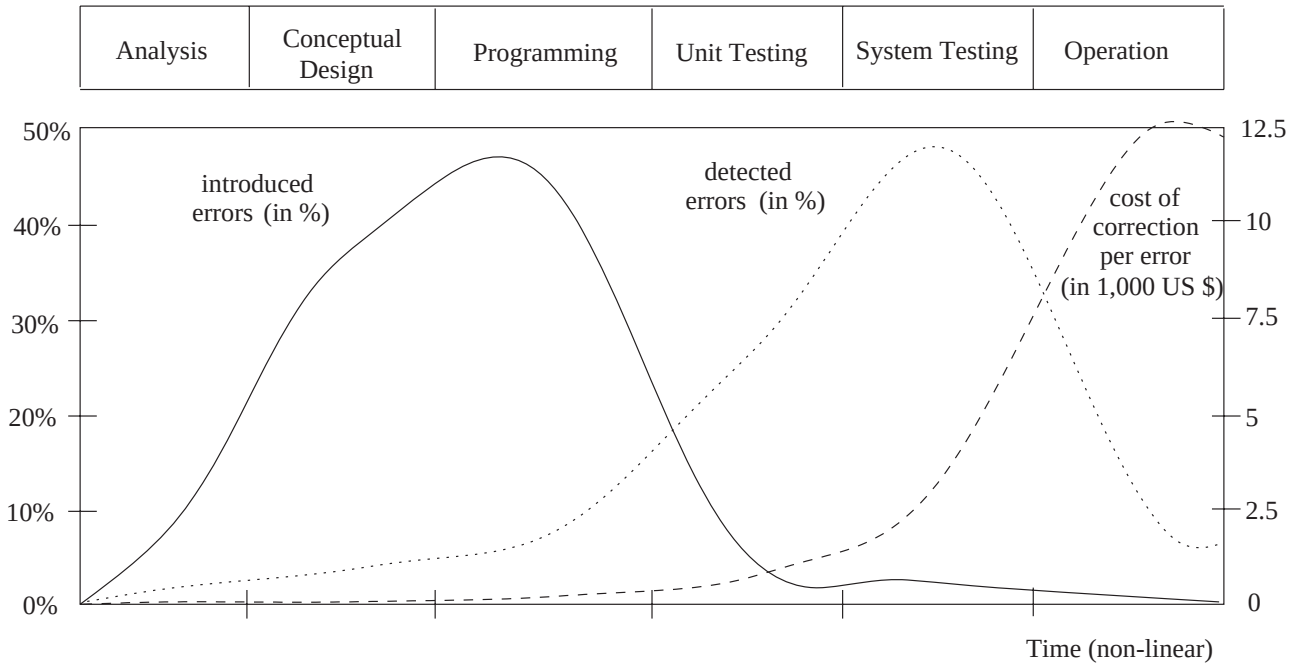


When are Bugs introduced and detected?



Peter Liggesmeyer, Martin Rothfelder, Michael Rettelbach, and Thomas Ackermann. Qualitätssicherung Software-basierter technischer Systeme – Problembereiche und Lösungsansätze. *Informatik-Spektrum*, 21(5):249–258, October 1998.

When are Bugs introduced and detected?



Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press. 2008.

How are Bugs detected?

- Peer review
- Simulation
- Testing
- Verification

Limitations of Peer Review

- Catches on average only 60% of the bugs.
- Is labour intensive (250 lines per hour).

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

How many can we check in one second?

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

How many can we check in one second?
3×10^{9}

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

How many can we check in one second?
3×10^{9}

How many seconds does it take?

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

How many can we check in one second?
3×10^{9}

How many seconds does it take?
$1.2 \times 10^{77} / 3 \times 10^{9} = 4 \times 10^{67}$

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

How many can we check in one second?
3×10^{9}

How many seconds does it take?
$1.2 \times 10^{77} / 3 \times 10^{9} = 4 \times 10^{67}$

How many years is that?

Limitations of Simulation

Provide your answer at
`http://pollev.com/franckvanbre226`.

How long does it take to simulate a 128-bit multiplier on a 3 GHz machine?

How many cases need to be checked?
$2^{2\times 128} \approx 1.2 \times 10^{77}$

How many can we check in one second?
3×10^{9}

How many seconds does it take?
$1.2 \times 10^{77} / 3 \times 10^{9} = 4 \times 10^{67}$

How many years is that?
2×10^{59}

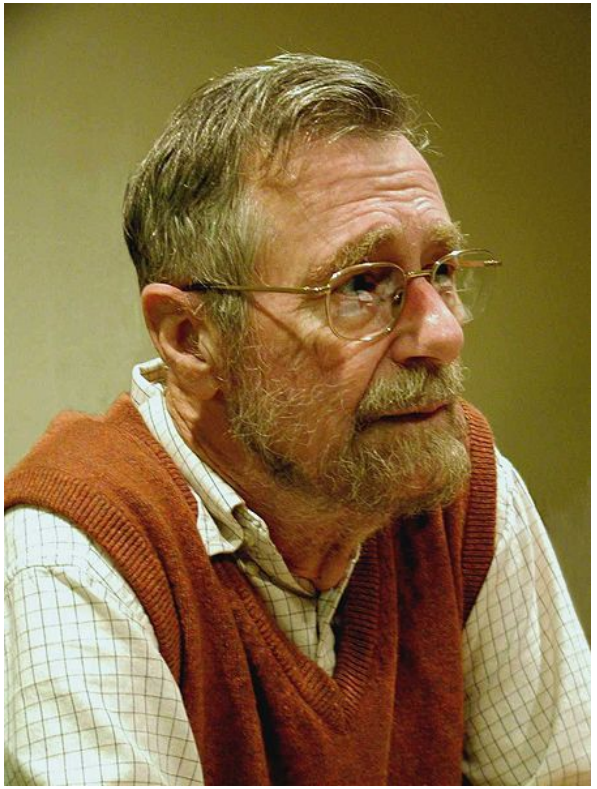
Limitations of Testing

"Program testing can be used to show the presence of bugs, but never to show their absence!"

Edsger W. Dijkstra. Notes on structured programming. Report 70-WSK-03, Technological University Eindhoven, April 1970.

Edsger Wybe Dijkstra (1930–2002)

- Member of the Royal Netherlands Academy of Arts and Sciences (1971)
- Distinguished Fellow of the British Computer Society (1971)
- Recipient of the Turing Award (1972)
- Foreign Honorary Member of the American Academy of Arts and Sciences (1975)



Source: Hamilton Richards