

GOTO Copenhagen 2021

#GOTOcph

#FALL

@KEVLINHENNEY



@tackline



Follow

Arriving in Bologna, I saw a [@KevlinHenney](#) screen. Whilst queueing to leave Ancona another appeared as I waited.

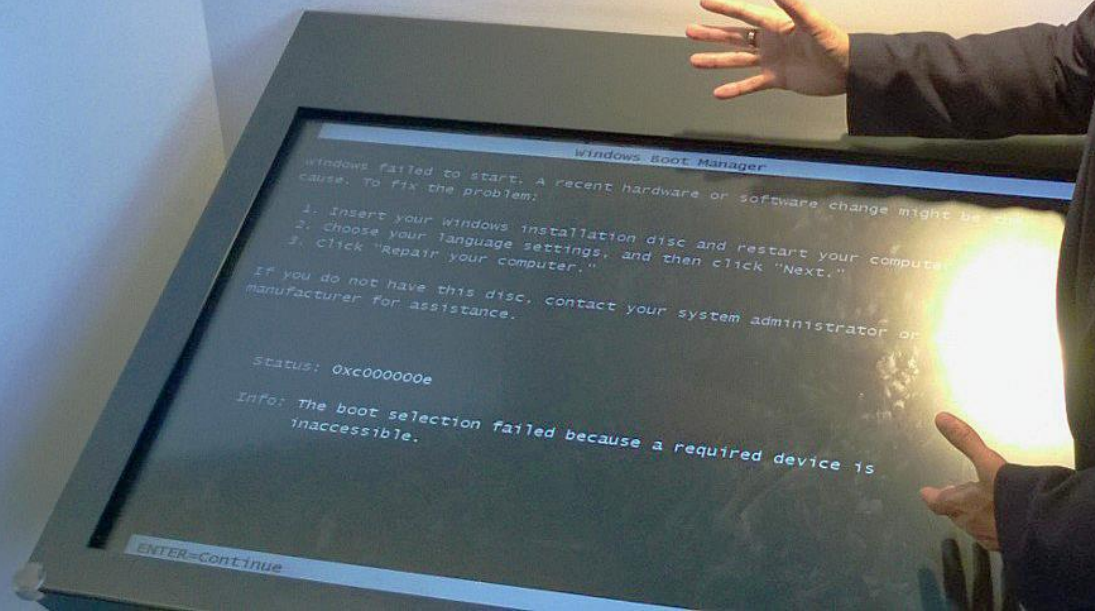
2:05 PM - 25 Jul 2016

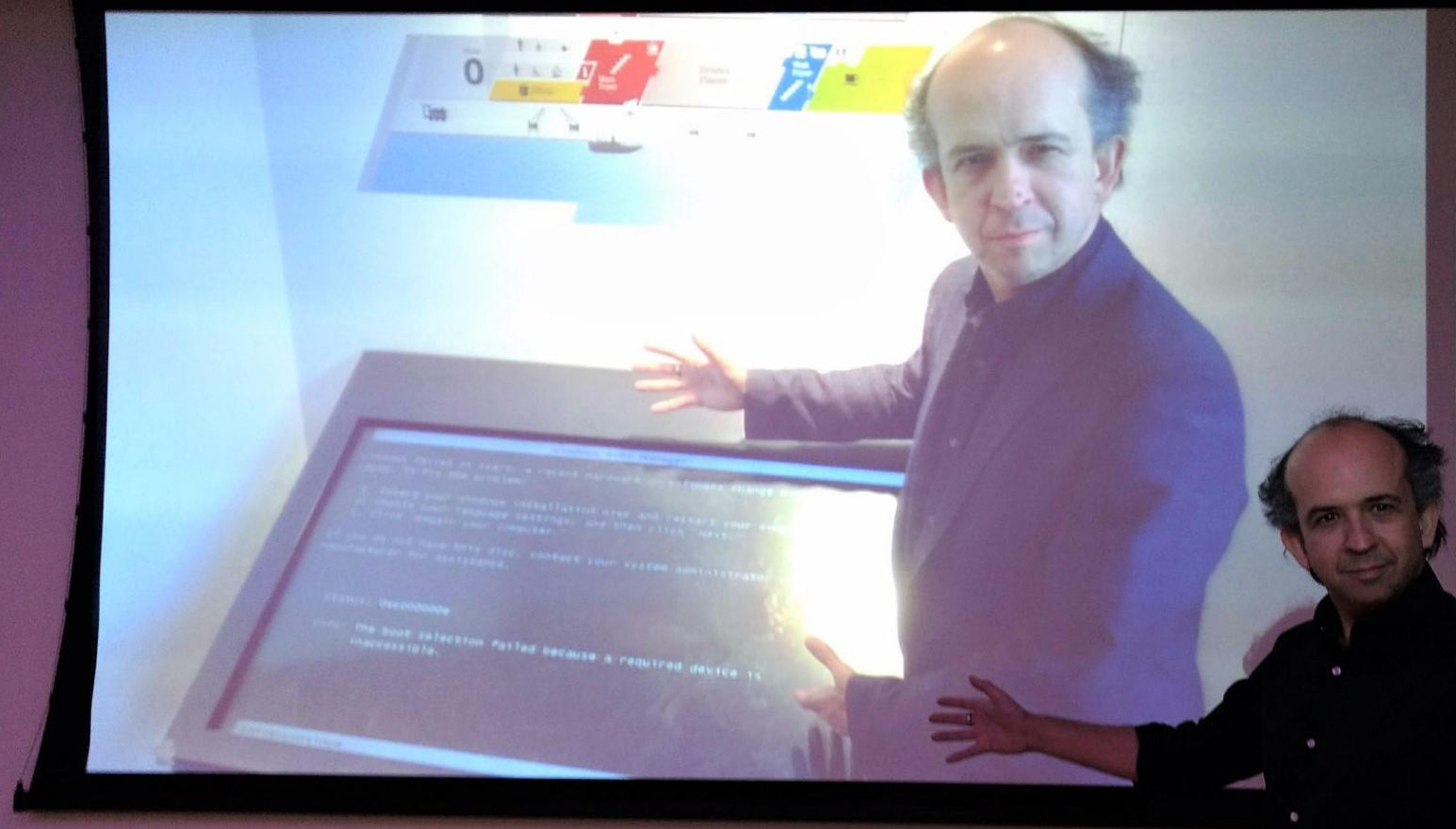


3



2





Twitter

Follow us
@agilecitybrs
Tag your Tweets
#agilecitybrs

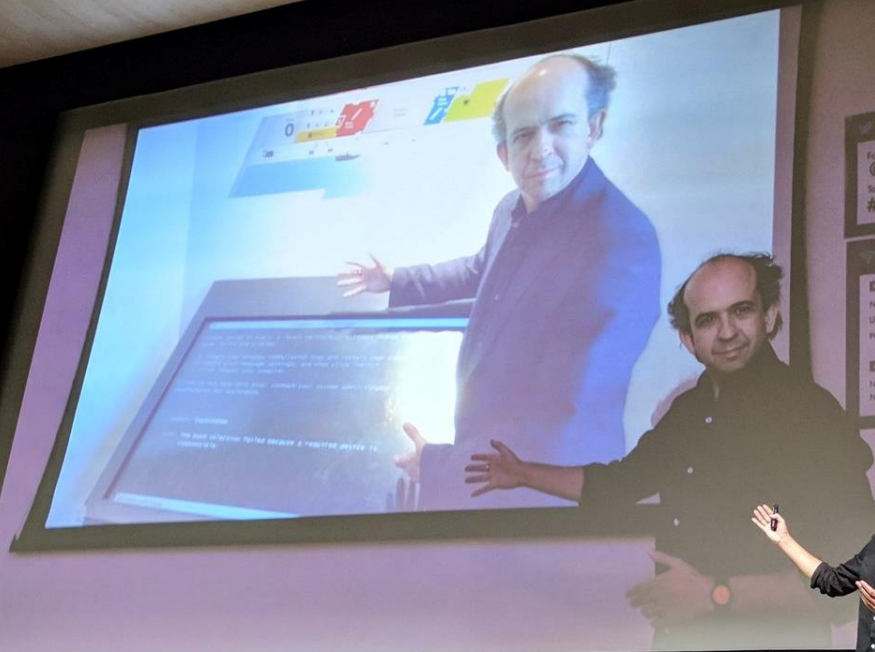
Wifi Access

Floor 2 only

Network: **BGuest**
Username: **guest**
Password: **bristol**

All of the building

Network: **Mshed**
No password required



Twitter
Follow us
@agilecitybrs
Tag your Tweets
#agilecitybrs

Wifi Access

Enter 2 lines

Networks BGuest

Username guest

Password bristol

All of the buttons

Networks Mashed

No password required

DevTernit V

Bootnotes

¹ My sample is a Tom Lehrer cover of a Noël Coward song, no less. Be impressed.

² A @KevlinHenney is a humiliating, public software failure, photographed and tweeted to the eponymous account. @KevlinHenneys are frequently seen at ATM machines and supermarket checkouts, but the best ones usually occur at transport hubs, where they enjoy the full benefit of giant displays.

TOP DEFINITION

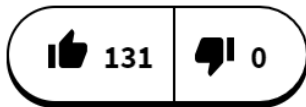


KevlinHenney 🔊

A software failure that happens in some [public space](#), for instance an airport flight information screen that has crashed, or an [ATM](#) that displays a [reboot](#) message.

Oh look, [the bus monitor](#) just [pulled](#) a KevlinHenney

by [srda56](#) December 06, 2018



Get a **KevlinHenney** mug





Michel Schudel
@MichelSchudel



Does this count as a [@KevlinHenney](#) ?

12:43 AM - May 26, 2018



62



19 people are talking about this



00004200021076035600

EXPEDITED PARCEL COLIS ACCÉLÉRÉS

2

CANADA POST / POSTES CANADA

From / Exp.:

\$retAdd.getFirstName().toUpperCase()

\$retAdd.getAddressLine1().toUpperCase()

\$retAdd.getCity().toUpperCase() \$retAdd.getState().toUpperCase() \$retAdd.g

\$retAdd.getDayPhone()

Payer / Facturé à:

7307904

Method of Payment /

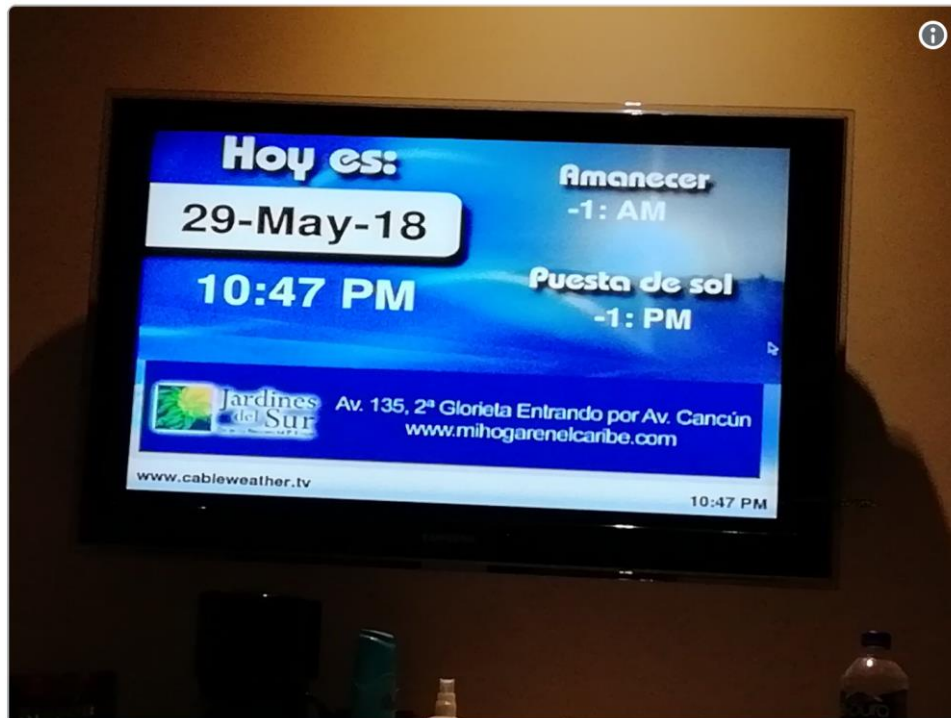
Mode de paiement:

To / Dest.:



C:\Users\Mappedin\AppData\Local\Programs\Python\Python36\python.exe

```
2018-07-02 23:25:16.135952: I T:\src\github\tensorflow\tensor
rts instructions that this TensorFlow binary was not compile
Model filename: C:\videoAnalytics\FaceEncoder\../models/Face
C:\videoAnalytics\FaceAnalyzer\../models/FaceAnalyzer/gender
C:\videoAnalytics\FaceAnalyzer\../models/FaceAnalyzer/age/in
C:\videoAnalytics\FaceAnalyzer\../models/FaceAnalyzer/age/in
```

Mr Anderson

@AndezFernandez



Not too sure if I am time traveling but pretty sure there is no
-1am or pm @KevlinHenney #mexicofail

5:54 AM - May 30, 2018



5



See Mr Anderson's other Tweets





Our Reply

31 December 1969

Your feedback will be used to improve Facebook. Thanks for taking the time to make a report.



Our Reply

31 December 1969

Your feedback will be used to improve Facebook. Thanks for taking the time to make a report.

The time function shall
return the value of time in
seconds since the Epoch.

Payment method

Cash

Card

PayPal

PayPalTM



You cannot pay by PayPal for
orders over £0.00. Please select
another payment method.

Optional updates

Choose the updates you want and then select Download and install.

✓ Driver updates

If you have a specific problem, one of these drivers might help.

☐ INTEL - System - 10/3/2016 12:00:00 AM - 10.1.1.38

☐ INTEL - System - 1/1/1970 12:00:00 AM - 10.1.1.42

☐ Intel - System - 4/12/2017 12:00:00 AM - 14.28.47.630



Anil Dash ✓

@anildash

The natural enemy of the programmer is the timezone.

5:32 AM · Dec 16, 2019



1.4K



335

twitter.com/anildash/status/1206447093872812033



Our Reply

31 December 1969

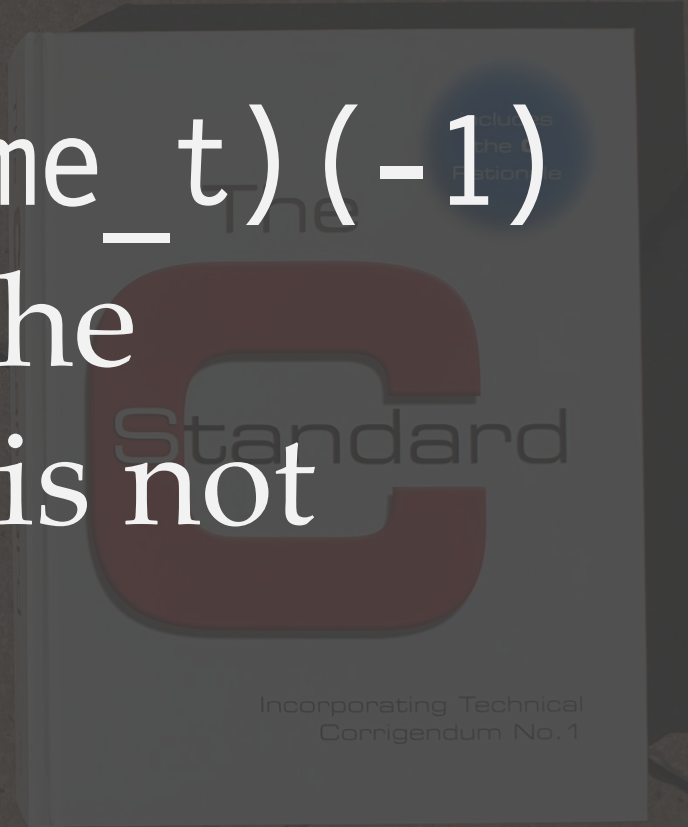
Your feedback will be used to improve Facebook. Thanks for taking the time to make a report.

includes
the **C**
Rationale

The **C** Standard

Incorporating Technical
Corrigendum No. 1

The value $(\text{time_t})(-1)$
is returned if the
calendar time is not
available.





The page at book.lufthansa.com says:



try to parse : NaN but it is not a number

OK





Kevlin Henney

@KevlinHenney

Replying to @brayniverse @ignotus_ph and @Spotify

Perhaps that's what we should call these: Batman bugs 🤔

10:52 AM · Mar 6, 2021



1



Share this Tweet

Driverless racecar drives straight into a wall

So during this initialization lap something happened which apparently caused the steering control signal to go to NaN and subsequently the steering locked to the maximum value to the right.

reddit.com/r/formula1/comments/jk9jrg/ot_roborace_driverless_racecar_drives_straight

Simple Testing Can Prevent Most Critical Failures

An Analysis of Production Failures in Distributed Data-Intensive Systems

Ding Yuan, Yu Luo, Xin Zhuang, Guilherme Renna Rodrigues,
Xu Zhao, Yongle Zhang, Pranay U Jain & Michael Stumm

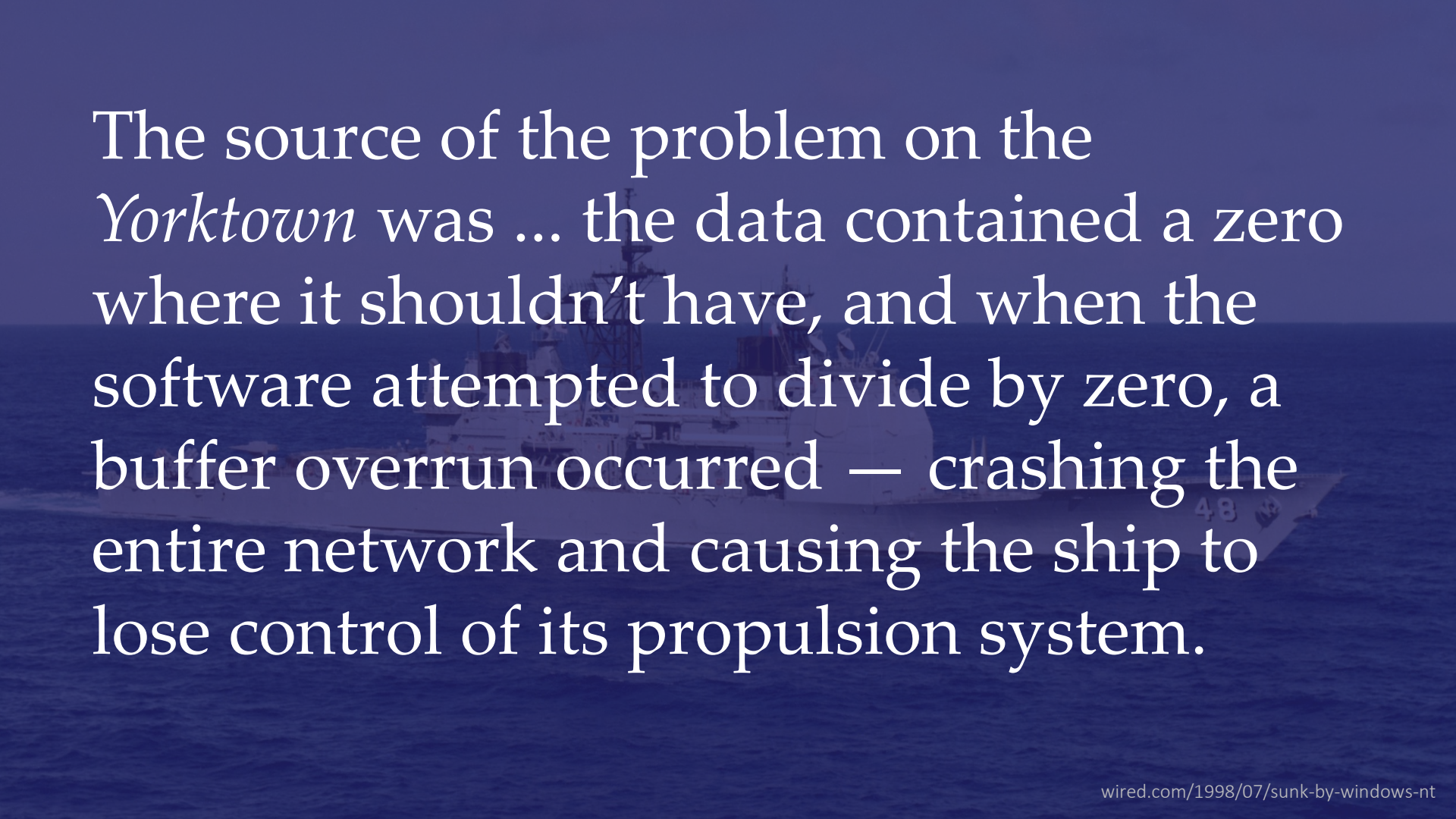
usenix.org/system/files/conference/osdi14/osdi14-paper-yuan.pdf

A majority of the production failures (77%) can be reproduced by a unit test.

Ding Yuan, Yu Luo, Xin Zhuang, Guilherme Renna Rodrigues,
Xu Zhao, Yongle Zhang, Pranay U Jain & Michael Stumm

usenix.org/system/files/conference/osdi14/osdi14-paper-yuan.pdf



The background of the slide is a photograph of the USS Yorktown (LHA-26), a Ticonderoga-class amphibious transport dock ship, sailing on the ocean. The ship is white with a dark hull and has the number '48' visible on its bow. The text is overlaid on this image in a white serif font.

The source of the problem on the *Yorktown* was ... the data contained a zero where it shouldn't have, and when the software attempted to divide by zero, a buffer overrun occurred — crashing the entire network and causing the ship to lose control of its propulsion system.

programming pearls

By Jon Bentley

WRITING CORRECT PROGRAMS

In the late 1960s people were talking about the promise of programs that verify the correctness of other programs. Unfortunately, it is now the middle of the 1980s, and, with precious few exceptions, there is still little more than talk about automated verification systems. Despite unrealized expectations, however, the research on program verification has given us something far more valuable than a black box that gobbles programs and flashes “good” or “bad”—we now have a fundamental understanding of computer programming.

The purpose of this column is to show how that fundamental understanding can help programmers write correct programs. But before we get to the subject itself, we must keep it in perspective. Coding skill is just one small part of writing correct programs. The majority of the task is the subject of the three previous columns: problem definition, algorithm design, and data structure selection. If you perform those tasks well, then writing correct code is usually easy.

The Challenge of Binary Search

I’ve given this problem as an in-class assignment in courses at Bell Labs and IBM. The professional programmers had one hour (sometimes more) to convert the above description into a program in the language of their choice; a high-level pseudo-code was fine. At the end of the specified time, almost all the programmers reported that they had correct code for the task. We would then take 30 minutes to examine their code, which the programmers did with test cases. In many different classes and with over a hundred programmers, the results varied little: 90 percent of the programmers found bugs in their code (and I wasn’t always convinced of the correctness of the code in which no bugs were found).

I found this amazing: only about 10 percent of professional programmers were able to get this small program right. But they aren’t the only ones to find this task difficult. In the history in Section 6.2.1 of his *Sorting and Searching*, Knuth points out that while the first binary search was published in 1946, the first published binary search without bugs did not appear until 1962.

```
L:=1; U:=N
loop
  { MustBe(L,U) }
  if L>U then
    P:=0; break
  M := (L+U) div 2
  case
    X[M] < T:  L:=M+1
    X[M] = T:  P:=M; break
    X[M] > T:  U:=M-1
endloop
```


One of the major benefits of program verification is that it gives programmers a language in which they can express that understanding.

```

{ MustBe(1,N) }
L := 1; U := N
{ MustBe(L,U) }
loop
  { MustBe(L,U) }
  if L>U then
    { L>U and MustBe(L,U) }
    { T is nowhere in the array }
    P := 0; break
  { MustBe(L,U) and L<=U }
  M := (L+U) div 2
  { MustBe(L,U) and L<=M<=U }
  case
    X[M] < T:
      { MustBe(L,U) and CantBe(1,M) }
      { MustBe(M+1,U) }
      L := M+1
      { MustBe(L,U) }
    X[M] = T:
      { X[M] = T }
      P := M; break
    X[M] > T:
      { MustBe(L,U) and CantBe(M,N) }
      { MustBe(L,M-1) }
      U := M-1
      { MustBe(L,U) }
  { MustBe(L,U) }
endloop

```

One of the major benefits of program verification is that it gives programmers a language in which they can express that understanding.

These techniques are only a small part of writing correct programs; keeping the code simple is usually the key to correctness.

On the other hand, several professional programmers familiar with these techniques have related to me an experience that is too common in my own programming: when they construct a program, the “hard” parts work the first time, while the bugs are in the “easy” parts.

On the other hand, several professional programmers familiar with these techniques have related to me an experience that is too common in my own programming: when they construct a program, the “hard” parts work the first time, while the bugs are in the “easy” parts.

#foreshadowing

```
public static int binarySearch(int[] a, int key) {  
    int low = 0;  
    int high = a.length - 1;  
  
    while (low <= high) {  
        int mid = (low + high) / 2;  
        int midVal = a[mid];  
  
        if (midVal < key)  
            low = mid + 1  
        else if (midVal > key)  
            high = mid - 1;  
        else  
            return mid; // key found  
    }  
    return -(low + 1); // key not found.  
}
```



```
public static int binarySearch(int[] a, int key) {  
    int low = 0;  
    int high = a.length - 1;  
  
    while (low <= high) {  
        int mid = (low + high) / 2;  
        int midVal = a[mid];  
  
        if (midVal < key)  
            low = mid + 1  
        else if (midVal > key)  
            high = mid - 1;  
        else  
            return mid; // key found  
    }  
    return -(low + 1); // key not found.  
}
```



```
L:=1; U:=N
loop
  { MustBe(L,U) }
  if L>U then
    L:=U; break
    M := (L+U) div 2
  else
    X[M] < T: L:=M+1
    X[M] = T: P:=M; break
    X[M] > T: U:=M-1
  endloop
```

Muphry's law, *noun*

- If you write anything correcting or criticising the quality of someone else's editing, proofing, spelling, grammar, etc., there will be some kind of editorial error in what you have written.
- Muphry's law is a specific application of Murphy's law, that anything that can go wrong will go wrong.

```
public static int binarySearch(int[] a, int key) {  
    int low = 0;  
    int high = a.length - 1;  
  
    while (low <= high) {  
        int mid = (low + high) / 2;  
        int midVal = a[mid];  
  
        if (midVal < key)  
            low = mid + 1  
        else if (midVal > key)  
            high = mid - 1;  
        else  
            return mid; // key found  
    }  
    return -(low + 1); // key not found.  
}
```

```
public static int binarySearch(int[] a, int key) {  
    int low = 0;  
    int high = a.length - 1;  
  
    while (low <= high) {  
        int mid = low + ((high - low) / 2);  
        int midVal = a[mid];  
  
        if (midVal < key)  
            low = mid + 1  
        else if (midVal > key)  
            high = mid - 1;  
        else  
            return mid; // key found  
    }  
    return -(low + 1); // key not found.  
}
```



```
public static int binarySearch(int[] a, int key) {  
    int low = 0;  
    int high = a.length - 1;  
  
    while (low <= high) {  
        int mid = (low + high) >>> 1;  
        int midVal = a[mid];  
  
        if (midVal < key)  
            low = mid + 1  
        else if (midVal > key)  
            high = mid - 1;  
        else  
            return mid; // key found  
    }  
    return -(low + 1); // key not found.  
}
```

```
public static int binarySearch(int[] a, int key) {  
    int low = 0;  
    int high = a.length - 1;
```

Probably faster, and arguably as clear

```
    while (low <= high) {  
        int mid = (low + high) >> 1;  
        int midVal = a[mid];  
  
        if (midVal < key)  
            low = mid + 1;  
        else if (midVal > key)  
            high = mid - 1;  
        else  
            return mid; // key found  
    }  
    return -(low + 1); // key not found.  
}
```



/THEORY/IN/PRACTICE

Beautiful Code

Leading Programmers Explain How They Think

O'REILLY

Edited by Andy Oram & Greg Wilson

Probably faster but may
be obscure to most Java
developers (including me)

Beautiful Code
Leading Programmers Explain How They Think

Edited by Andy Oram & Greg Wilson

Alberto Savoia

One of the major benefits of program verification is that it gives programmers a language in which they can express that understanding.

These techniques are only a small part of writing correct programs; keeping the code simple is usually the key to correctness.

More Programming Pearls

Confessions of a Coder

Jon Bentley





More Programming Pearls

Confessions of a Coder

Jon Bentley

If the programmer can simulate
a construct faster than the
compiler can implement the
construct itself, then the compiler
writer has blown it badly.

Guy L Steele, Jr

The general lesson that I take away from this bug is humility: It is hard to write even the smallest piece of code correctly, and our whole world runs on big, complex pieces of code.

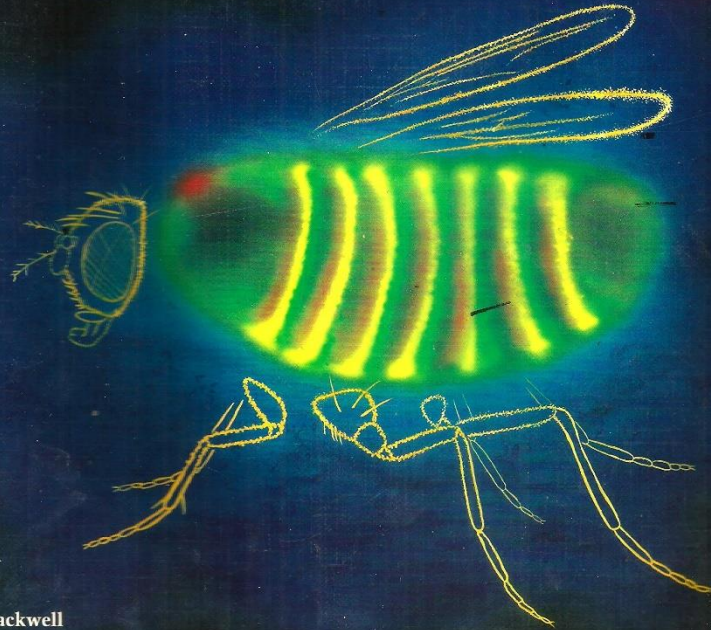
Joshua Bloch

ai.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html

PETER A. LAWRENCE

The Making of a Fly

THE GENETICS OF ANIMAL DESIGN



Blackwell
Scientific
Publications



The Making of a Fly: The Genetics of Animal Design (Paperback)

by Peter A. Lawrence

[Return to product information](#)

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about [Safe Online Shopping](#) and our [safe buying guarantee](#).

Price at a Glance

List
Price: ~~\$70.00~~

Used: from **\$35.54**

New: from
\$1,730,045.91

Have one to sell? [Sell yours here](#)

All

New (2 from \$1,730,045.91)

Used (15 from \$35.54)

Show ☒ New ☐ Prime offers only (0)

Sorted by Price + Shipping

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information	Buying Options
\$1,730,045.91 + \$3.99 shipping	New	<p>Seller: profnath</p> <p>Seller Rating: ★★★★★ 93% positive over the past 12 months. (8,193 total ratings)</p> <p>In Stock. Ships from NJ, United States. Domestic shipping rates and return policy.</p> <p>Brand new, Perfect condition, Satisfaction Guaranteed.</p>	<p>Add to Cart</p> <p>or</p> <p>Sign in to turn on 1-Click ordering.</p>
\$2,198,177.95 + \$3.99 shipping	New	<p>Seller: bordeebook</p> <p>Seller Rating: ★★★★★ 93% positive over the past 12 months. (125,891 total ratings)</p> <p>In Stock. Ships from United States. Domestic shipping rates and return policy.</p> <p>New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!</p>	<p>Add to Cart</p> <p>or</p> <p>Sign in to turn on 1-Click ordering.</p>



The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

[Return to product information](#)

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about [Safe Online Shopping](#) and our [safe buying guarantees](#).

Price at a Glance

List: \$70.00
Price:
Used: from **\$35.54**
New: from
\$1,730,045.91

Have one to sell? [Sell yours here](#)

\$1,730,045.91

All **New** (2 from \$1,730,045.91) **Used** (15 from \$35.54)

Showing 1-2 of 2 offers

Price	Shipping	Condition	Seller Information	Shipping Options
\$1,730,045.91	Free shipping	New	prerath Seller Rating: ★★★★★ 93% positive over the past 12 months. (125,891 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy . Brand new, Perfect condition, Satisfaction Guaranteed.	Add to Cart or Sign in to turn on 1-Click ordering.
\$2,198,177.95	+ \$3.99 shipping	New	bordeebook Seller Rating: ★★★★★ 93% positive over the past 12 months. (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy . New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!	Add to Cart or Sign in to turn on 1-Click ordering.



The Making of a Fly: The Genetics of Animal Design (Paperback) by Peter A. Lawrence

[Return to product information](#)

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about [Safe Online Shopping](#) and our [safe buying guarantees](#).

Price at a Glance

List: \$70.00
Price:

Used: from **\$35.54**

New: from
\$1,730,045.91

Have one to sell? [Sell yours here](#)

\$2,198,177.95

All

New (2 from \$1,730,045.91)

Used (15 from \$35.54)

New 3 of 2 offers

Price Shipping

\$1,730,045.91

Condition

New

Seller information

bordeebook

Seller Rating: **93%** positive over the past 12 months

(125,891 total ratings)

In Stock. Ships from NJ, United States.

Domestic shipping rates and return policy.

Brand new, Perfect condition, Satisfaction Guaranteed.

\$2,198,177.95

New

+ \$3.99 shipping

Seller: **bordeebook**

Seller Rating: **93%** positive over the past 12 months.

(125,891 total ratings)

In Stock. Ships from United States.

Domestic shipping rates and return policy.

New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!

Price + Shipping

Buying Options

Add to Cart

Sign in to turn on 1-Click ordering.

Add to Cart

or

Sign in to turn on 1-Click ordering.



The Making of a Fly: The Genetics of Animal Design (Paperback) by Peter A. Lawrence

[Return to product information](#)

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about [Safe Online Shopping](#) and our [safe buying guarantees](#).

Price at a Glance

List
Price: **\$70.00**

Used: from **\$35.54**

New: from
\$1,730,045.91

Have one to sell? [Sell yours here](#)

All

New (2 from \$1,730,045.91)

Used (15 from \$35.54)

Show ☒ New ☐ Prime offers only (0)

Price + Shipping

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information	Shipping Options
\$1,730,045.91 + \$3.99 shipping	New	Seller: profnath Seller Rating: ★★★★★ (8,193 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy . Brand new, Perfect condition, Satisfaction Guaranteed.	Add to Cart or Sign in to turn on 1-Click ordering.
\$2,198,177.95 + \$3.99 shipping	New	Seller: bordeebook Seller Rating: ★★★★★ 93% positive over the past 12 months. (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy . New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!	Add to Cart or Sign in to turn on 1-Click ordering.

\$35.54

	profnath	bordeebook	profnath over previous bordeebook	bordeebook over profnath
8-Apr	\$1,730,045.91	\$2,198,177.95		1.27059
9-Apr	\$2,194,443.04	\$2,788,233.00	0.99830	1.27059
10-Apr	\$2,783,493.00	\$3,536,675.57	0.99830	1.27059
11-Apr	\$3,530,663.65	\$4,486,021.69	0.99830	1.27059
12-Apr	\$4,478,395.76	\$5,690,199.43	0.99830	1.27059
13-Apr	\$5,680,526.66	\$7,217,612.38	0.99830	1.27059

0.99830

	profnath	bordeebook	profnath over previous	bordeebook over previous
8-Apr	\$1,730,040.91	\$2,098,177.55	0.99830	1.27059
9-Apr	\$2,194,440.04	\$2,788,753.00	0.99830	1.27059
10-Apr	\$2,783,493.00	\$3,530,675.57	0.99830	1.27059
11-Apr	\$3,530,663.65	\$4,486,021.69	0.99830	1.27059
12-Apr	\$4,478,395.76	\$5,690,199.43	0.99830	1.27059
13-Apr	\$5,680,526.66	\$7,217,612.38	0.99830	1.27059

	profnath	bordeebook	profnath over previous bordeebook	bordeebook over profnath
8-Apr	\$1,730,045.9	\$2,198,777.95	1.27059	
9-Apr	\$2,194,443.0	\$2,782,233.00	0.99830	1.27059
10-Apr	\$2,783,493.5	\$3,530,663.65	0.99830	1.27059
11-Apr	\$3,530,663.65	\$4,478,395.76	0.99830	1.27059
12-Apr	\$4,478,395.76	\$5,680,526.66	0.99830	1.27059
13-Apr	\$5,680,526.66	\$7,217,612.38	0.99830	1.27059

1.27509



The Making of a Fly: The Genetics of Animal Design (Paperback)

by Peter A. Lawrence

[Return to product information](#)

Always pay through Amazon.com's Shopping Cart or 1-Click.

Learn more about [Safe Online Shopping](#) and our [safe buying guarantee](#).

Price at a Glance

List Price: \$70.00

Used: from **\$42.56**

New: from **\$18,651,718.08**

Have one to sell? [Sell yours here](#)

All **New** (2 from \$18,651,718.08) **Used** (11 from \$42.56)

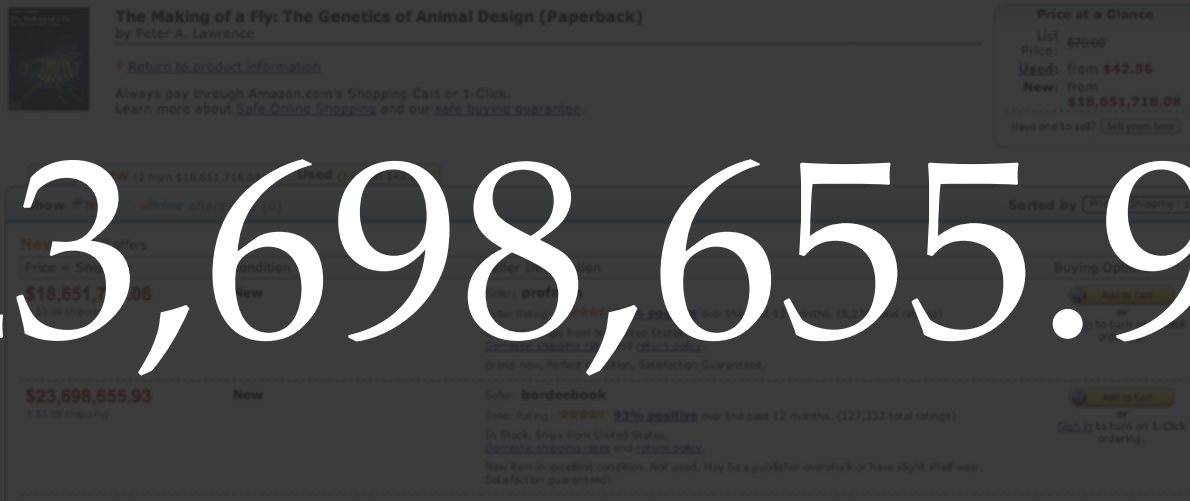
Show **New** ☐ Prime offers only (0)

Sorted by Price + Shipping 1

New 1-2 of 2 offers

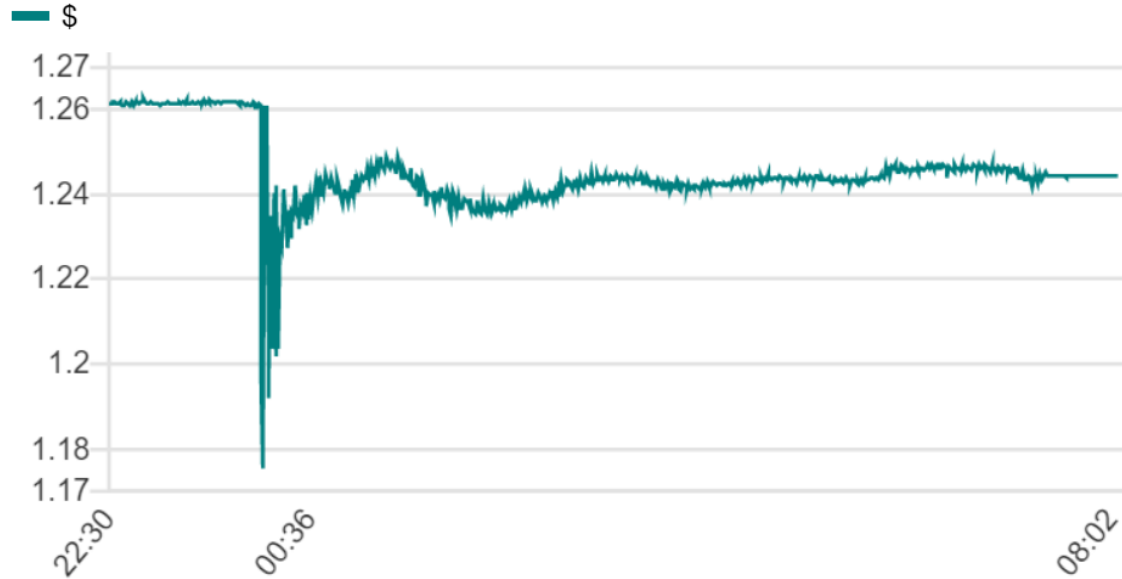
Price + Shipping	Condition	Seller Information	Buying Options
\$18,651,718.08 + \$3.99 shipping	New	Seller: profnath Seller Rating: ★★★★★ 93% positive over the past 12 months. (8,278 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy . Brand new, Perfect condition, Satisfaction Guaranteed.	Add to Cart or Sign in to turn on 1-Click ordering.
\$23,698,655.93 + \$3.99 shipping	New	Seller: bordeebok Seller Rating: ★★★★★ 93% positive over the past 12 months. (127,332 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy . New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!	Add to Cart or Sign in to turn on 1-Click ordering.

\$23,698,655.93



Sterling flash crash

£/\$, 6-7 October



Source: Bloomberg

BBC

The pound has dived on Asian markets with automated trading being blamed for the volatility.

Digital devices tune
out small errors while
creating opportunities
for large errors.

Earl Wiener



**101 Things I Learned
in Architecture School**

Matthew Frederick

Always design a thing by
considering it in its next
larger context.









1596

```

G_M_INFO_DERIVE(T_ALG.E_DCN)
if L_M_DON_32 > 32767 then
    P_M_DERIVE(T_ALG.E_DON) := 16#7FFF#;
elsif L_M_DON_32 < -32768 then
    P_M_DERIVE(T_ALG.E_DON) := 16#8000#;
else
    P_M_DERIVE(T_ALG.E_DON) := UC_16S_EN_16NS(
        TDB.T_ENTIER_16S(L_M_DON_32));
end if;

P_M_DERIVE(T_ALG.E_DOE) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
    ((1.0/C_M_LSB_DOE) *
    G_M_INFO_DERIVE(T_ALG.E_DOE))

L_M_BV_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
    G_M_INFO_DERIVE(T_ALG.E_BV));

if L_M_BV_32 > 32767 then
    P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
    P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
    P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M
end if;

P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
    ((1.0/C_M_LSB_BH) *
    G_M_INFO_DERIVE(T_ALG.E_BH)))

```

501

```

end LIRE_DERIVE;
--$finprocedure

```

The connections between modules are the assumptions which the modules make about each other.

David Parnas

“Information Distribution Aspects of Design Methodology”

Deleting dead code is not a technical problem; it is a problem of mindset and culture.

Kevlin Henney

“Dead Code Must Be Removed”
infoq.com/news/2017/02/dead-code



Soyuz fails to deliver 19 satellites from Vostochny

russianspaceweb.com/meteor-m2-1.html

A photograph of a rocket launch from a snowy field, with a blue overlay. The rocket is ascending vertically, leaving a large plume of white smoke and fire. The launch pad is visible in the foreground, and the surrounding landscape is covered in snow. The text is overlaid on the image in a white, sans-serif font.

All the hardware aboard the Fregat upper stage performed as planned.

But, almost unbelievably, the flight control system on the Fregat did not have the correct settings for the mission originating from the new launch site in Vostochny, as opposed to routine launches from Baikonur and Plesetsk.

Early Detection of Configuration Errors to Reduce Failure Damage

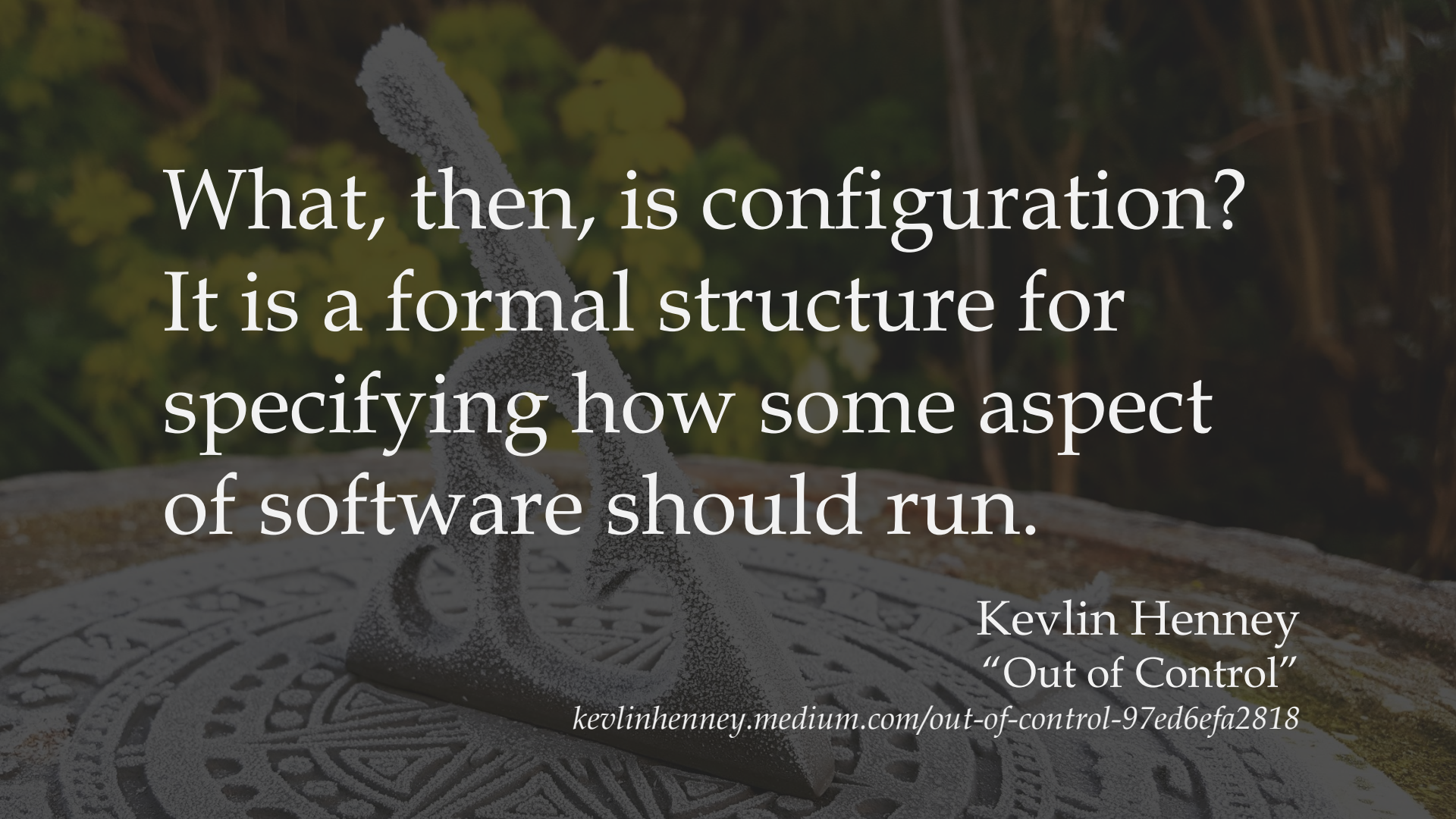
Tianyin Xu, Xinxin Jin, Peng Huang, Yuanyuan Zhou,
Shan Lu, Long Jin & Shankar Pasupathy

usenix.org/system/files/conference/osdi16/osdi16-xu.pdf

Our study shows that many of today's mature, widely used software systems are subject to latent configuration errors in their critically important configurations.

Tianyin Xu, Xinxin Jin, Peng Huang, Yuanyuan Zhou,
Shan Lu, Long Jin & Shankar Pasupathy

usenix.org/system/files/conference/osdi16/osdi16-xu.pdf



What, then, is configuration?
It is a formal structure for
specifying how some aspect
of software should run.

Kevlin Henney
“Out of Control”

kevinhenney.medium.com/out-of-control-97ed6efa2818

A background image of a sandcastle on a beach. The sandcastle is in the foreground, with a sunset sky in the background. The text "Configuration is code." is overlaid on the image.

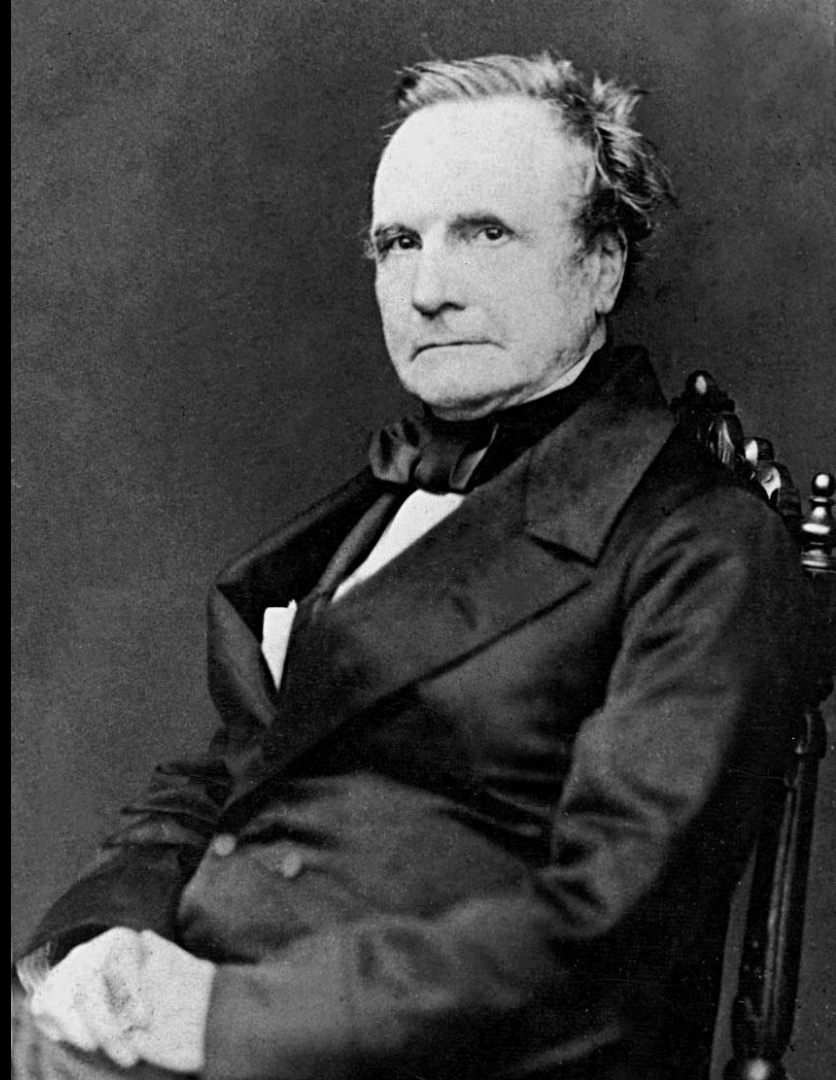
Configuration is code.

Kevlin Henney
"Out of Control"

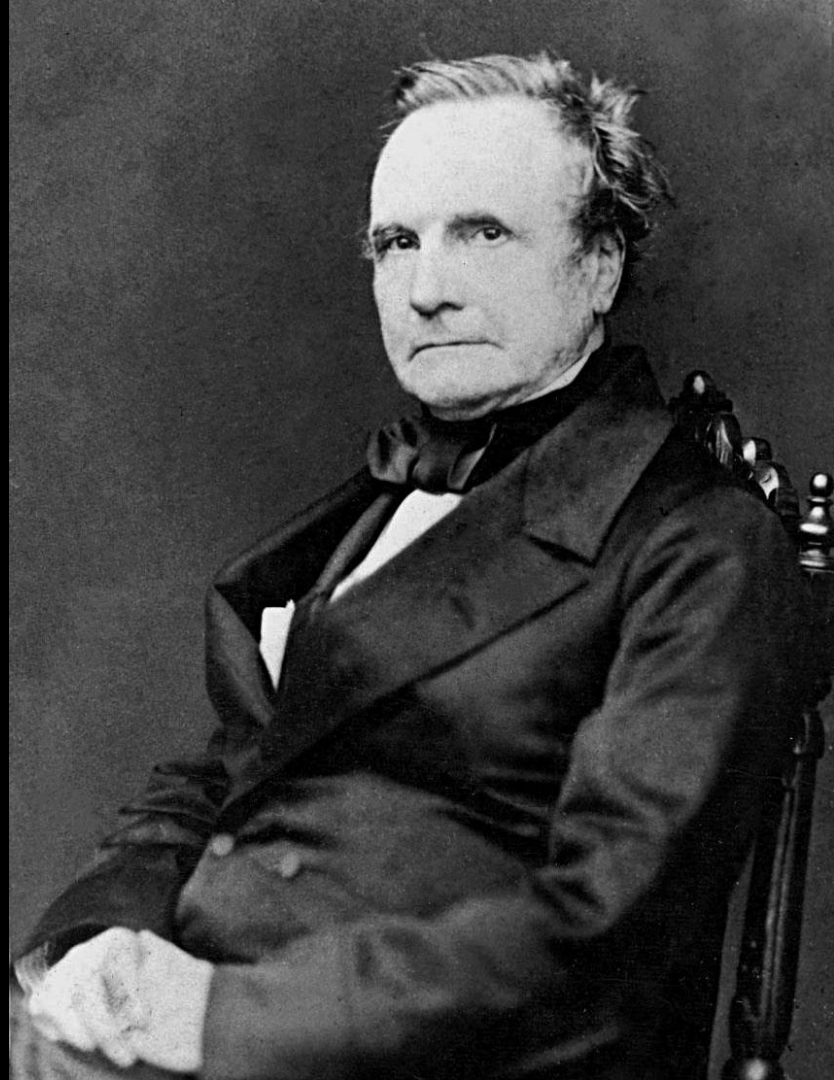
kevinhenney.medium.com/out-of-control-97ed6efa2818

GIGO

On two occasions I have been asked, — “Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?”



I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.



In August 1984, the Wall Street Journal reported that a Texas-based oil and gas company had fired several executives after the firm lost millions of dollars in an acquisition deal because of “errors traced to a faulty financial analysis spread sheet model.”

Steven Levy

backchannel.com/a-spreadsheet-way-of-knowledge-8de60af7146e

Excel is the world's
most popular
functional language

Simon Peyton-Jones

In December 2020, we announced LAMBDA, which allows users to define new functions written in Excel's own formula language. With LAMBDA, Excel has become Turing-complete.

LAMBDA: The ultimate Excel worksheet function
microsoft.com/en-us/research/blog/lambda-the-ultimate-excel-worksheet-function

Gene name errors are widespread in the scientific literature

Mark Ziemann, Yotam Eren & Assam El-Osta

genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1044-7

	gene names	internal date format	default date format		gene names	internal date format	default date format		gene names	internal date format	default date format
	A	B	C	D	E	F	G	H	I	J	K
1	APR-1	35885	1-Apr		OCT-1	36068	1-Oct		SEP2	36039	2-Sep
2	APR-2	35886	2-Apr		OCT-2	36069	2-Oct		SEP3	36040	3-Sep
3	APR-3	35887	3-Apr		OCT-3	36070	3-Oct		SEP4	36041	4-Sep
4	APR-4	35888	4-Apr		OCT-4	36071	4-Oct		SEP5	36042	5-Sep
5	APR-5	35889	5-Apr		OCT-6	36073	6-Oct		SEP6	36043	6-Sep
6	DEC-1	36129	1-Dec		OCT1	36068	1-Oct		SEPT1	36038	1-Sep
7	DEC-2	36130	2-Dec		OCT11	36078	11-Oct		SEPT2	36039	2-Sep
8	DEC1	36129	1-Dec		OCT2	36069	2-Oct		SEPT3	36040	3-Sep
9	DEC2	36130	2-Dec		OCT3	36070	3-Oct		SEPT4	36041	4-Sep
10	MAR1	35854	1-Mar		OCT4	36071	4-Oct		SEPT5	36042	5-Sep
11	MAR2	35855	2-Mar		OCT6	36073	6-Oct		SEPT6	36043	6-Sep
12	MAR3	35856	3-Mar		OCT7	36074	7-Oct		SEPT7	36044	7-Sep
13	NOV1	36099	1-Nov		SEP-1	36038	1-Sep		SEPT8	36045	8-Sep
14	NOV2	36100	2-Nov		SEP-2	36039	2-Sep		SEPT9	36046	9-Sep
15					SEP1	36038	1-Sep				
16											

excel.gene2date.xls

Sheet1 Sheet2

Ready Sum=0 SCRL CAPS NUM

A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

Mark Ziemann, Yotam Eren & Assam El-Osta

genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1044-7

Comment | [Published: 03 August 2020](#)

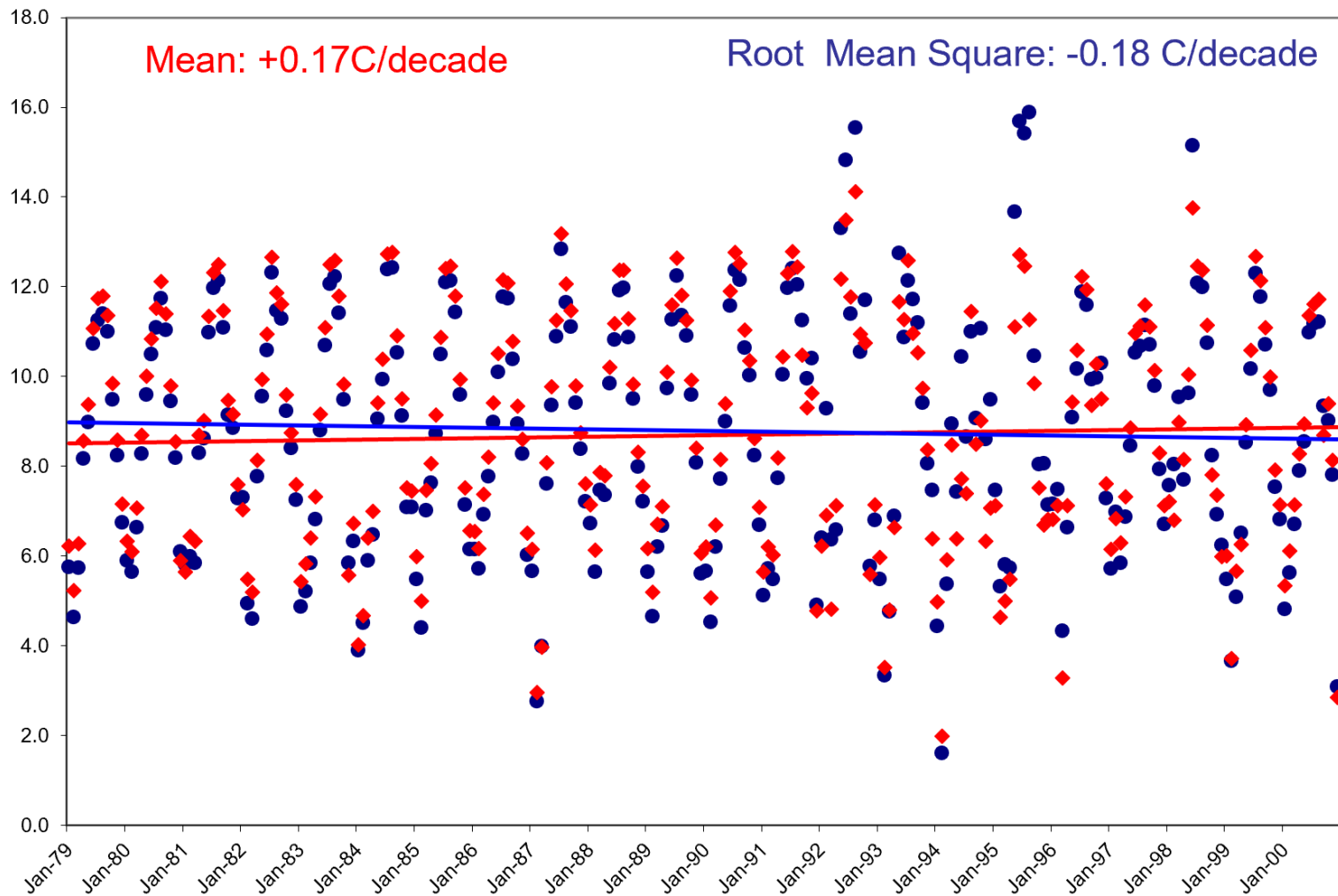
Guidelines for human gene nomenclature

Elspeth A. Bruford , Bryony Braschi, Paul Denny, Tamsin E. M. Jones, Ruth L. Seal & Susan Tweedie

Nature Genetics **52**, 754–758(2020) | [Cite this article](#)

Standardized gene naming is crucial for effective communication about genes, and as genomics becomes increasingly important in health care, the need for a consistent language to refer to human genes becomes ever more essential. Here, we present the current HUGO Gene Nomenclature Committee (HGNC) guidelines for naming not only protein-coding genes but also RNA genes and pseudogenes, and we outline the changes in approach and ethos that have resulted from the discoveries of the past few decades.





Corrections to the McKittrick (2002) Global Average Temperature Series



By [tlambert](#) on May 20, 2004.

scienceblogs.com/deltoid/2004/05/20/mckitrick3

Last week I wrote about Paul Georgia's review of Essex and McKittrick's *Taken by Storm*. Based on their book, Georgia made multiple incorrect statements about the physics of temperature. Of course, it might have just been that Georgia misunderstood their book. Fortunately Essex and McKittrick have a [briefing](#) on their book, and while Georgia mangles the physics even worse than them, they do indeed claim that there is no physical basis to average temperature. They present two graphs of temperature trends that purport to show that you can get either a cooling trend or a warming trend depending on how you compute the average. McKittrick recently was in the news for publishing a controversial paper that claimed that an audit of the commonly accepted reconstruction of temperatures over the past 1000 years was incorrect, so it only seems fair to audit Essex and McKittrick's graphs. As we will see, *both* of their graphs are wrong, and their results go away when the errors are corrected.

=AVERAGE(B28:K28)

$$x_{\text{rms}} = \sqrt{\frac{1}{n} (x_1^2 + x_2^2 + \cdots + x_n^2)}$$

$$\begin{aligned}
& 26)^2 + (273.15 + G26)^2 + (273.15 + F26)^2 + (273.15 + E26)^2 + (273.15 + D26)^2 + (273.15 + C26)^2 + (273.15 + B26)^2))^0.5 - 273.15 \\
& = (0.1 * ((273.15 + K27)^2 + (273.15 + J27)^2 + (273.15 + I27)^2 + (273.15 + H27)^2 + (273.15 + G27)^2 + (273.15 + F27)^2 + (273.15 + E27)^2 + (273.15 + D27)^2 + (273.15 + C27)^2 + (273.15 + B27)^2))^0.5 - 273.15 \\
& = (0.1 * ((273.15 + K28)^2 + (273.15 + J28)^2 + (273.15 + I28)^2 + (273.15 + H28)^2 + (273.15 + G28)^2 + (273.15 + F28)^2 + (273.15 + E28)^2 + (273.15 + D28)^2 + (273.15 + C28)^2 + (273.15 + B28)^2))^0.5 - 273.15 \\
& = (0.1 * ((273.15 + K29)^2 + (273.15 + J29)^2 + (273.15 + I29)^2 + (273.15 + H29)^2 + (273.15 + G29)^2 + (273.15 + F29)^2 + (273.15 + E29)^2 + (273.15 + D29)^2 + (273.15 + C29)^2 + (273.15 + B29)^2))^0.5 - 273.15 \\
& = (0.1 * ((273.15 + K30)^2 + (273.15 + J30)^2 + (273.15 + I30)^2 + (273.15 + H30)^2 + (273.15 + G30)^2 + (273.15 + F30)^2 + (273.15 + E30)^2 + (273.15 + D30)^2 + (273.15 + C30)^2 + (273.15 + B30)^2))^0.5 - 273.15 \\
& = (0.1 * ((273.15 + K31)^2 + (273.15 + J31)^2 + (273.15 + I31)^2 + (273.15 + H31)^2 + (273.15 + G31)^2 + (273.15 + F31)^2 + (273.15 + E31)^2 + (273.15 + D31)^2 + (273.15 + C31)^2 + (273.15 + B31)^2))^0.5 - 273.15
\end{aligned}$$

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        square = (cell.Value + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0

    For Each cell In values.Cells
        square = (cell.Value + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next

    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        square = (cell.Value + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```



```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        square = (cell.Value + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

	Toronto, Canada	Honolulu, Hawaii	Bogota, Columbia	Egedesminde, Greenland	Dublin, Ireland	Salehard, Russia	Chiang Mai, Thaila	Ceduna, Australia	Jan Smuts, South	Halley, Antarctica			
lat	43.7	21.4	4.7	68.7	53.4	66.5	18.8	-32.1	-26.1	-75.5			
lon	-79.6	-157.9	-74.2	-52.8	-6.2	66.7	99	133.7	28.2	-26.6	Average	RMS	Year
Mar-94	-0.9	22.9	14.1	-22.4	7.6	-11	27	20.3	17.7	-21.6	5.4	5.9	1994
Apr-94	7.2	24.4	14.4	-9.4	7.8		29.2	17.1	15.4	-25.6	8.9	8.5	1994
May-94	11.8	26.3		0.2	9.4	-3.7	28.2	15.6		-28.3	7.4	6.4	1994
Jun-94	19.1	27.2	14.2	2.4			27.6		8.7	-26.1	10.4	7.7	1994
Jul-94	21.7	28.1	13.5	5.8	14.9			12.5	8	-35.3	8.7	7.4	1994
Aug-94	18.7	29.2	13.4	5.1	13.9	12.1	26.4	11.9	11.7	-32.5	11.0	11.4	1994
Sep-94	15.9		13.7	1.8	11.9	5.4	27.2	14.5	16.3	-25	9.1	8.5	1994
Oct-94	10		13.2	-2.6	10.1	-1.6	25.7	18.1	15.7		11.1	9.0	1994
Nov-94		27.3	13.2	-6.4	9.4	-19.1		18	17.8		8.6	6.3	1994
Dec-94		24.7	12.9	-12.5		-22.8	22.7	22.6	18.8		9.5	7.1	1994
Jan-95	-3.1	23.6	12.8	-21.1	4.6	-13.2	22.8	21.1	19.7		7.5	7.1	1995
Feb-95	-7.2	22.8	13.1	-22.1	6	-13		22.7	20.2		5.3	4.6	1995
Mar-95	1.9	24.2	13.7	-22.9	5.3	-11.9		18.3	17.9		5.8	5.0	1995
Apr-95	4.6	24.8	14	-7.2	8.6	1		16.2	15.1	-25.5	5.7	5.5	1995
May-95		25.9	13.8	1.3	10.3	3.1	29.1	14.1	11.8		13.7	11.1	1995
Jun-95	20	27.5	14	4.9	13.1	7.3	29.2		9.5		15.7	12.7	1995
Jul-95	21.9		13.3	7.2	16.3	14.8	27.8	11.9	10.1		15.4	12.5	1995
Aug-95	21.8	28.6	12.8	4.9	17.4	13.1			12.6		15.9	11.3	1995
Sep-95		28.4	13.7	3.6	12.7	6.5	27.5	14.3	17	-29.6	10.5	9.8	1995
Oct-95	11		13.4	-3.7	12.4	-3.4	26.1	17.8	17.4	-18.6	8.0	7.5	1995
Nov-95	1		13.2	-2.5	7.7	-16.3	24.5	18.9	18		8.1	6.7	1995
Dec-95	-5.1	26.2	12.5	-7.9	4.6	-24	20.4	20.3	17.2		7.1	6.8	1995
Jan-96	-6.8	24.5	12	-12.6	6	-18.5		19.7	19.1		7.2	6.8	1996
Feb-96	-5.8	23.3	13.1	-15.5	5.6	-16.4	22.3	22.5	18.3		7.5	7.1	1996
Mar-96	-2.8	23.6	13.2	-16.4	4.8	-9.8			17.7		4.3	3.3	1996
Apr-96	4.4	26.4	12.8	-9.1	8.3	-10.2	29.1	15.8	14.7	-25.8	6.6	7.1	1996
May-96	11.8	26.2	13	-1.8	8.5	1.8	27.7	13.6	12.9	-22.8	9.1	9.4	1996
Jun-96	18.6	27.5	13.3	1.3	13.1	8.5	26.8	12.6	10.6	-30.7	10.2	10.6	1996
Jul-96	19.6	27.8	12.9	4.8	14.9	15.2	26.8	12.6	8.3	-24.1	11.9	12.2	1996
Aug-96	20.8	28.4	12.7	4.3	14.3	8.7	26.6	11.9	11.4	-23.1	11.6	11.9	1996
Sep-96	16.5	27.2		1.5	12.9	1.9	26.9	14.6	15.9	-28	9.9	9.4	1996
Oct-96	9.2	27.5	13.1	-3.1	10.9	-1	25.5	16.7	18.6	-17.7	10.0	10.3	1996
Nov-96	0.9	25	12.9	-6.3	5.7	-6.6	25	18	18		10.3	9.5	1996
Dec-96	-0.4	22.8	12.5	-6.1	4.2	-18.1	22.3	20.1	19.3	-3.7	7.3	7.6	1996
Jan-97	-6.5	22.2	13	-8.6	4.2	-26.2	20.8	22.4	19.7	-3.9	5.7	6.2	1997
Feb-97	-3.2	23.6	13.5	-18	6.2	-27.8	22.6	25.1	20.8		7.0	6.8	1997
Mar-97	-1.4	24.2	13.2	-16.9	7.7	-13	26.4	18	17.7	-17.5	5.8	6.3	1997
Apr-97	5.7	24.8	13.3	-11	8.8	-2.7	27.1	17.3	13.6	-28.2	6.9	7.3	1997
May-97	9.7	24.5	13.8	-0.9	9.9	1.6	29.7	13.7	10.8	-28.3	8.5	8.9	1997
Jun-97	19.7	27.2	13.6	5.3	12.4	8.2	28.7	11.2	10.1	-31.1	10.5	11.0	1997
Jul-97	20.9	27.5	13.9	4.2	15	9.8	28.2	9	9.8	-31.6	10.7	11.1	1997
Aug-97	19	28.1	13.9	4.5	16	9.8	27.6	11.3	13.6	-32.4	11.1	11.6	1997
Sep-97	15.3	28.1	13.3	1.9	13.1	7.4	26.9	14.3	15.3	-28.5	10.7	11.1	1997
Oct-97	9.5	27	13.9	-1.5	10.5	1.6	26.1	16.5	16.7	-22.4	9.8	10.1	1997
Nov-97	2.1	24.5	13.4	-5.3	8.3	-16.5	24.8	19.9	18	-9.8	7.9	8.3	1997
Dec-97	-0.9	23.3	13.3	-6.6	6.4	-25.9	23.5	19.9	19.6	-5.6	6.7	7.1	1997
Jan-98	-2.2	22.5	13.8	-8.9	5.7	-24.1	21.7	20.7	19		7.6	7.2	1998
Feb-98	-0.3	22.8		-19.4	7.9		23.5	20.8	19.6	-10.5	8.1	6.8	1998

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        square = (cell.Value + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        square = (          + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        square = (0 + 273.15) ^ 2
        total = total + square
        count = count + 1
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```



```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        If Not IsEmpty(cell) Then
            square = (cell.Value + 273.15) ^ 2
            total = total + square
            count = count + 1
        End If
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

```
Public Function RMS(values As range)
    Dim square, total, count
    total = 0
    count = 0
    For Each cell In values.Cells
        If Not IsEmpty(cell) Then
            square = (cell.Value + 273.15) ^ 2
            total = total + square
            count = count + 1
        End If
    Next
    RMS = (total / count) ^ 0.5 - 273.15
End Function
```

Growth in a Time of Debt

By CARMEN M. REINHART AND KENNETH S. ROGOFF✉

In this paper, we exploit a new multi-country historical dataset on public (government) debt to search for a systemic relationship between high public debt levels, growth and inflation.¹ Our main result is that whereas the link between growth and debt seems relatively weak at “normal” debt levels, median growth rates for countries with public debt over roughly 90 percent of GDP are about one percent lower than otherwise; average (mean) growth rates are several percent lower. Surprisingly, the relationship between public debt and growth is remarkably similar across emerging markets and advanced

especially against the backdrop of graying populations and rising social insurance costs? Are sharply elevated public debts ultimately a manageable policy challenge?

Our approach here is decidedly empirical, taking advantage of a broad new historical dataset on public debt (in particular, central government debt) first presented in Carmen M. Reinhart and Kenneth S. Rogoff (2008, 2009b). Prior to this dataset, it was exceedingly difficult to get more than two or three decades of public debt data even for many rich countries, and virtually impossible for most emerging markets.

Does High Public Debt Consistently Stifle Economic Growth? A Critique of Reinhart and Rogoff

Thomas Herndon* Michael Ash Robert Pollin

April 15, 2013

JEL CODES: E60, E62, E65

Abstract

We replicate Reinhart and Rogoff (2010a and 2010b) and find that coding errors, selective exclusion of available data, and unconventional weighting of summary statistics lead to serious errors that inaccurately represent the relationship between public debt and GDP growth among 20 advanced economies in the post-war period. Our finding is that when properly calculated, the average real GDP growth rate for countries carrying a public-debt-to-GDP ratio of over 90 percent is actually 2.2 percent, not -0.1 percent as published in Reinhart and Rogoff. That is, contrary to RR, average GDP growth at public debt/GDP ratios over 90 percent is not dramatically different than when debt/GDP ratios are lower.

... the average real GDP growth rate for countries carrying a public-debt-to-GDP ratio of over 90 percent is actually 2.2 percent, not -0.1 percent...

Abstract

(We replicate Reinhart and Rogoff (2010a and 2010b) and find that coding errors, selective exclusion of available data, and conventional weighting of summary statistics lead to serious errors that inaccurately represent the relationship between public debt and GDP growth among 20 advanced economies in the post-war period. Our finding is that when properly calculated, the average real GDP growth rate for countries carrying a public-debt-to-GDP ratio of over 90 percent is actually 2.2 percent, not -0.1 percent as published in Reinhart and Rogoff. That is, contrary to RR, average GDP growth at public debt/GDP ratios over 90 percent is not dramatically different than when debt/GDP ratios are lower.

Institute for Public Policy Research



ENDING THE BLAME GAME

THE CASE FOR A NEW
APPROACH TO PUBLIC
HEALTH AND PREVENTION

A decade of austerity has resulted in cuts to public health, prevention and mental health budgets in the NHS, and wider national and local government services which help drive better health.

ENDING THE
BLAME GAME
THE CASE FOR A NEW
APPROACH TO PUBLIC
HEALTH AND PREVENTION

Had the trend from 1990 and 2012 continued, we estimate that there could have been 130,000 deaths averted between 2012 and 2017.

ENDING THE
BLAME GAME

THE CASE FOR A NEW
APPROACH TO PUBLIC
HEALTH AND PREVENTION

Institute for Public Policy Research

Had the trend from 1990 and 2012 continued, we estimate that there could have been 130,000 deaths averted between 2012 and 2017.

ENDING THE
BLAME GAME

THE CASE FOR A NEW
APPROACH TO PUBLIC
HEALTH AND PREVENTION

#forshadowing

Those who cannot
remember the past are
condemned to repeat it.

George Santayana

What experience and history teach is this — that people and governments never have learned anything from history, or acted on principles deduced from it.

Georg Hegel