“A short potter through 21 years of software er, engineering"

Les Hatton

Professor of Forensic Software Engineering
CISM, University of Kingston
L.Hatton@kingston.ac.uk

Version 1.2: 01/Apr/2005

Overview

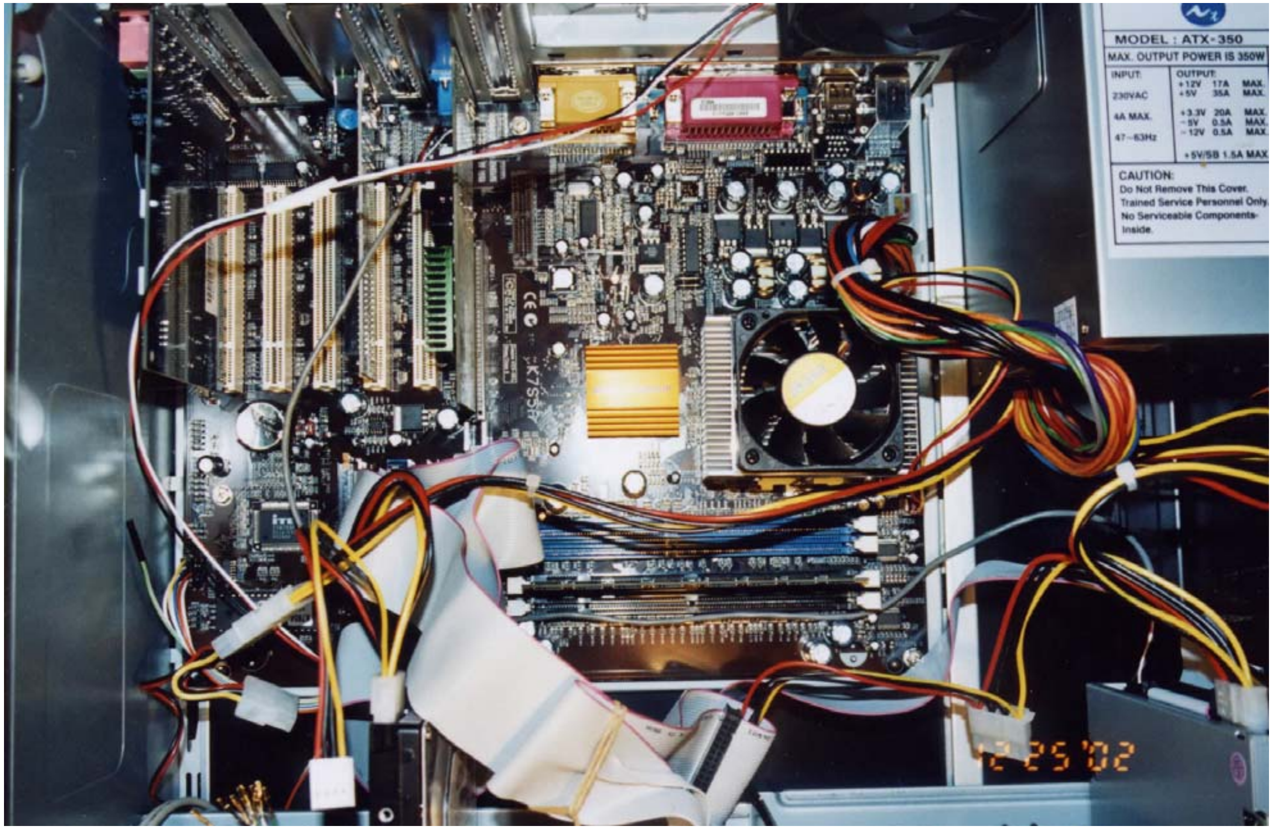
- Nostalgia ain’t what it used to be
- Things I’m not very happy about
- Things I’m quite excited about

1984

Mine 8-)

Its big brother :-(

2005



Build your own .. http://www.leshatton.org/lxf37_pc.html

Old and new

1984

- Cray X MP, 0.5 sec.

2005

- Hand-built 200 quid PC with bits from a computer fair, 0.044 sec.

If you have a Fortran compiler, feel free to download the benchmark from:-

http://www.leshatton.org/FB_885.html

1984

Technology

- Mostly Fortran 77, some C. Embedded systems, assembler.
- Command line and Problem Oriented Languages. Interfaces relatively unambitious although sometimes legendarily terse.
- Seismic data processing package ~ 500,000 LOC was pretty big then.
- Defect density ~0.9 per KLOC, MTBF 4 weeks at Cray volumes (both asymptotic)

2005

Technology

- Mostly languages based on C, (C++, PHP, Perl, Java, Javascript, …)
- Networked systems
- Highly complex graphical interfaces
- Windows XP is around 35 Million LOC
- Defect density ~2.0-4.0 per KLOC.

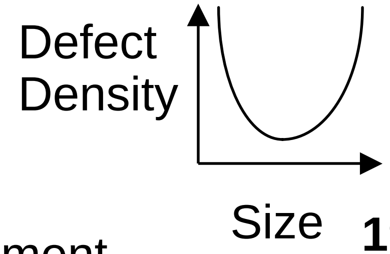
1984-2005 a personal view

1984-1988:
Porting same F77 package
gave 4 sig.fig. agreement
on different platforms.

1990-92:
T1: ~10 static faults/KLOC
in F77, C. (C++ worse)

1990-1993:
T2: 9-version dynamic experiment.
Only 1 sig. fig. agreement left at end.

1989-1995



1995-1999:
Win'95 1 defect every 42 mins.
Mac - 1 defect every 188 mins.
Linux - Almost never.

1995-7:
Formal methods better ? – Sometimes.
Static fault highly correlated
to dynamic failure.

1996:
O-O/C++ has 2x defect
correction cost.

1997:
Compression and accuracy

1997:
N-version might be better
than we thought

1998-1999:
Why do we have so much
repetitive failure in software ?

1999-:
Necessary and unnecessary
complexity

2000-2003:
Formalisation of safer
Subsets and definition of
signal to noise.

2001--:
Thermodynamic properties of
software failure

2002-:
Embedded system
paranoia to test arithmetic

2003-:
Gives up and joins blues band

Overview

- Nostalgia ain’t what it used to be
- Things I’m not very happy about
- Things I’m quite excited about

Things I’m not very happy about

- Reliability
- Increasing obfuscation. Is parsimony now a sin ?
- Language decay

Reliability

1984

- VAX/VMS MTBF of > 4 years.
- Unix systems, MTBF > 2-4 years

Reliability

2005

- VAX/VMS MTBF of > 4 years.
- Linux systems, MTBF > 5-10 years

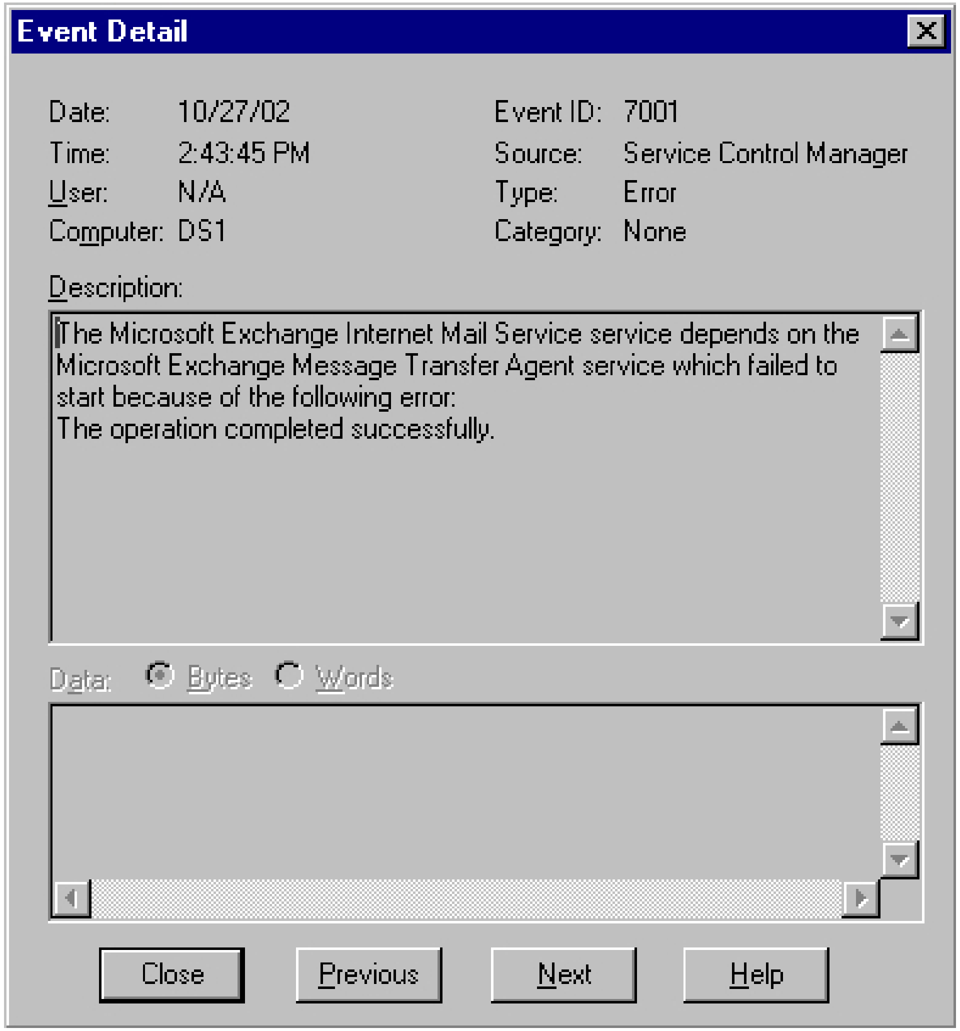
So what do we all use …

WINDOZE !

06/01/2005

- Blue screen of Death crashes Gates at CES

WINDOZE !



Event Detail

Date: 10/27/02
Time: 2:43:45 PM
User: N/A
Computer: DS1

Event ID: 7001
Source: Service Control Manager
Type: Error
Category: None

Description:

The Microsoft Exchange Internet Mail Service service depends on the Microsoft Exchange Message Transfer Agent service which failed to start because of the following error:
The operation completed successfully.

Fortunately nobody would be stupid enough to put this in a critical system.

WINDOZE !

06/09/2004

- Royal Navy to run warships on Windows 2000.

  This follows on from the deployment of Windows NT on the USS Enterprise in 1997 which then had to be rebooted frequently and occasionally towed to port.

22/09/2004

- Total air-traffic failure at Los Angeles after Unix system replacement Windows 2000 server hung because they forgot to reboot it frequently enough. The Unix systems had never failed

Security

- Recovering a tainted Windows XP machine …
  - Even disc scan failed. To save disc, mount in firewire caddy and back up under Unix. (3 hrs)
  - Reformat hard disc (1 hr)
  - Contact supplier to find XP Home CD is 70 quid *$&%^** !
  - Reload from own disc and restore (4 hrs)
  - Install ZoneAlarm, download upgrades, service packs, security fixes, (10 hrs)
  - Norton anti-virus now fights it out with XP SP2 for privilege of protecting us, and *switches off messages to avoid duplicates*
  - ZoneAlarm informs us that it has blocked 54 intrusion attempts whilst we were downloading upgrades.

Conclusion

We live in an era when software reliability is considered a joke.

This is not an appropriate culture within which to develop critical systems.

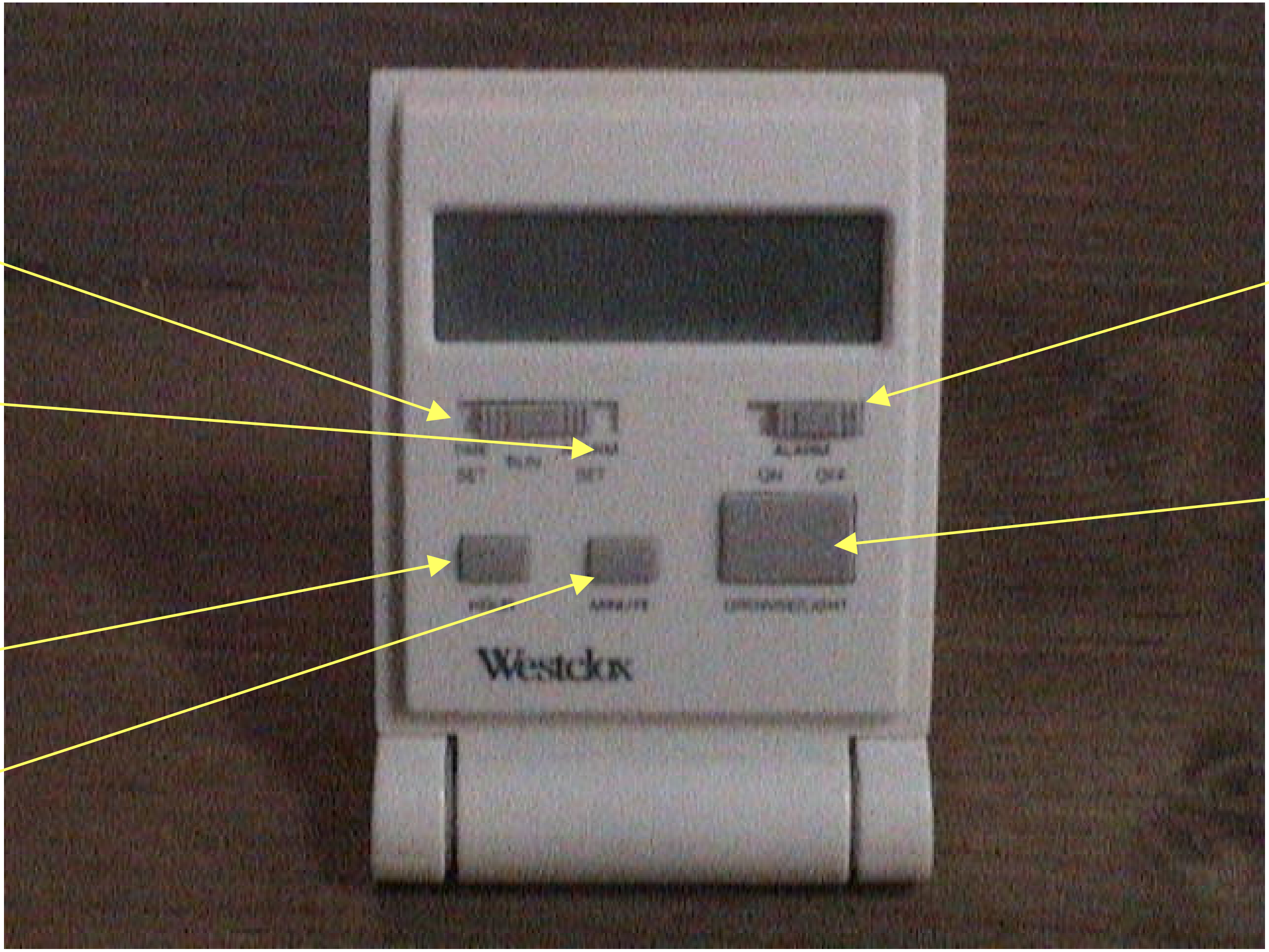
Things I’m not very happy about

- Reliability
- Increasing obfuscation. Is parsimony now a sin ?
- Language decay

A tale of two alarm clocks

- Alarm clock 1
  - Purchased 5 years ago and faultless ever since.
  - Staggeringly simple and intuitively obvious interface which has never required the instructions to be consulted.

A tale of two alarm clocks – the sublime



Time set

Alarm set

Hour

Minute

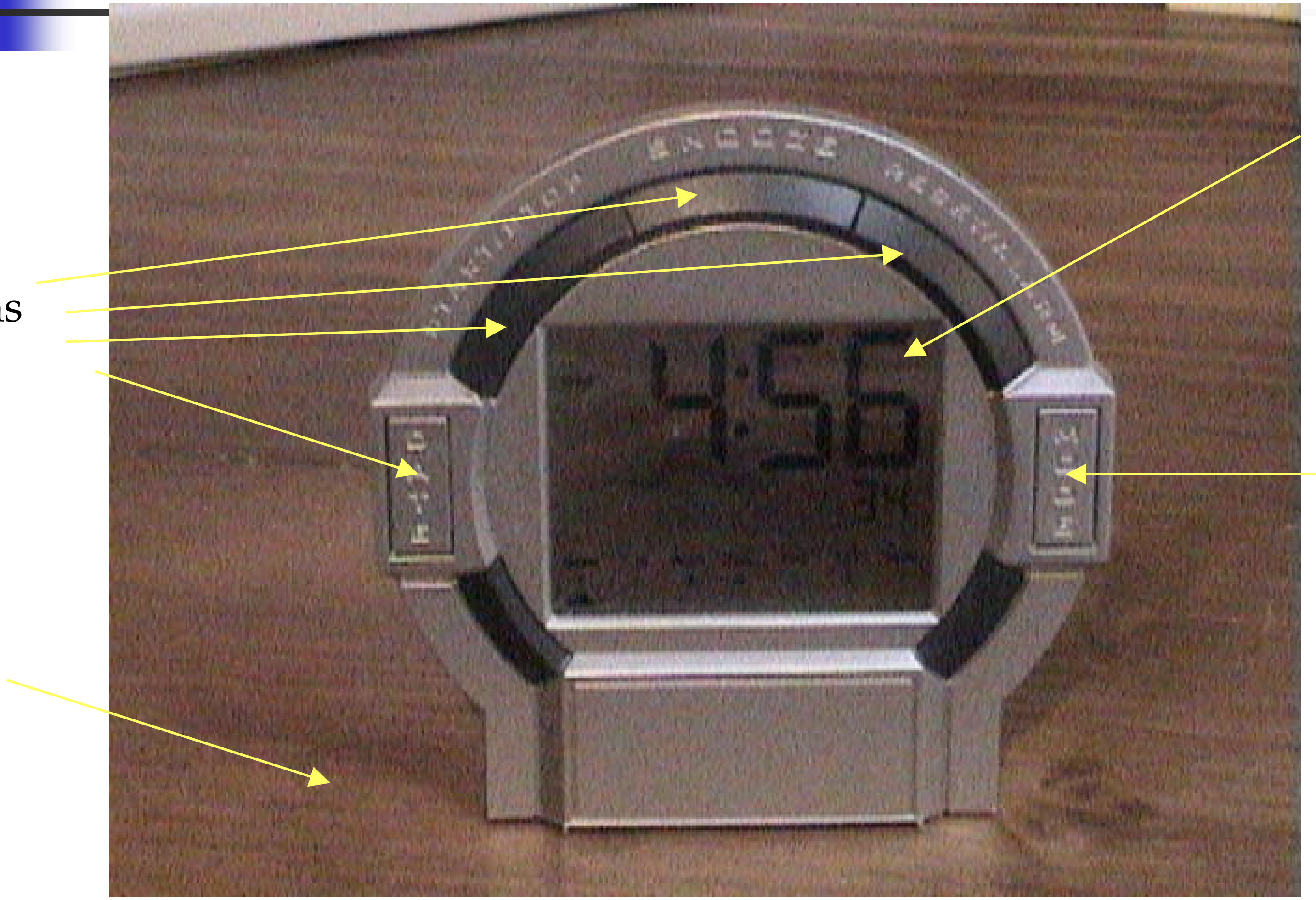
Alarm On/off

Drowse

A tale of two alarm clocks

- Alarm clock 2
  - Purchased 1 year ago and frequently resets itself
  - Staggeringly complex and intuitively non-obvious interface.

A tale of two alarm clocks – the ridiculous



Random numbers

More buttons

Mode - ARGH !

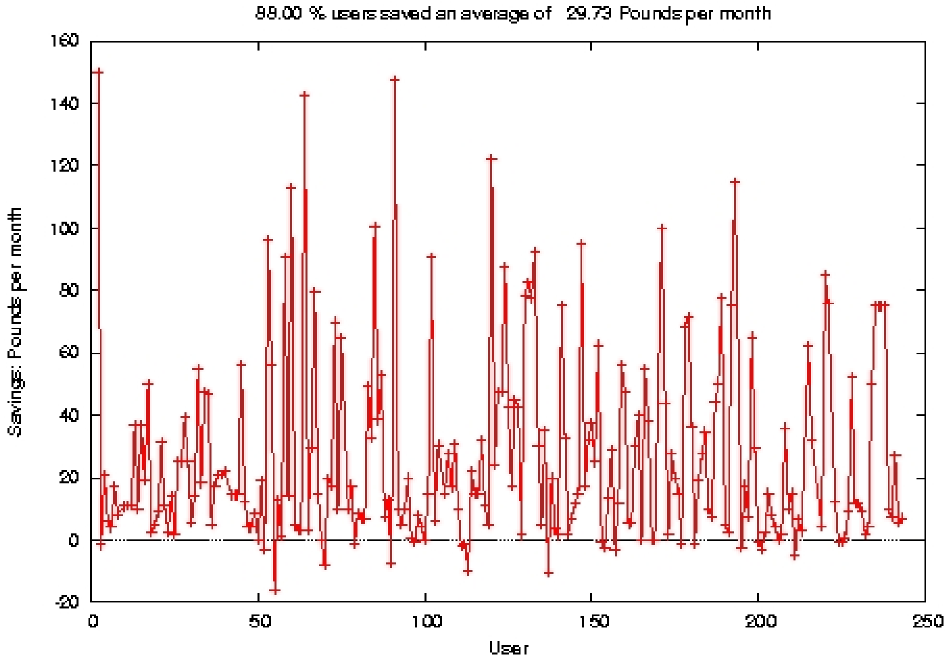
Desk

A tale of two alarm clocks

- Alarm clock 2 – some examples
  - Time setting
    - Press MODE button THREE times. Press SELECT and SET repeatedly.
  - Alarm setting
    - Press MODE button TWO times. Then follow time setting.
  - Hourly chime on/off
    - Press SELECT and MODE button *simultaneously*
  - Alarm on
    - Press DATE and SELECT button *simultaneously*
  - Stop watch / lap counter
    - Press MODE button ONCE and then beg for mercy as clock gibbers with entertaining series of random beeps.
  - Turning alarm off
    - Hurl out of window after standing on clock screaming.

Obfuscation in mobile phone charges UK Results 2Q04

Illustrates the natural growth of complexity *when technology allows*



Fuzzy global optimisation web server finding minimum phone charges

Average saving £29.73 per month

http://www.betterdeal.co.uk/

Obfuscated interfaces …

Automobile industry:

- 06/02/2005. Whole string of problems, shaking Mercedes, Ford that bakes back seat passengers …

  http://www.nytimes.com/2005/02/06/automobiles/06AUTO.html

- 26/10/2004. BMW disables dynamic stability control and ABS. Two police drivers vindicated after investigation.

  http://www.daserste.de/plusminus/beitrag.asp?iid=254

Things I’m not very happy about

- Reliability
- Increasing obfuscation. Is parsimony now a sin ?
- Language decay

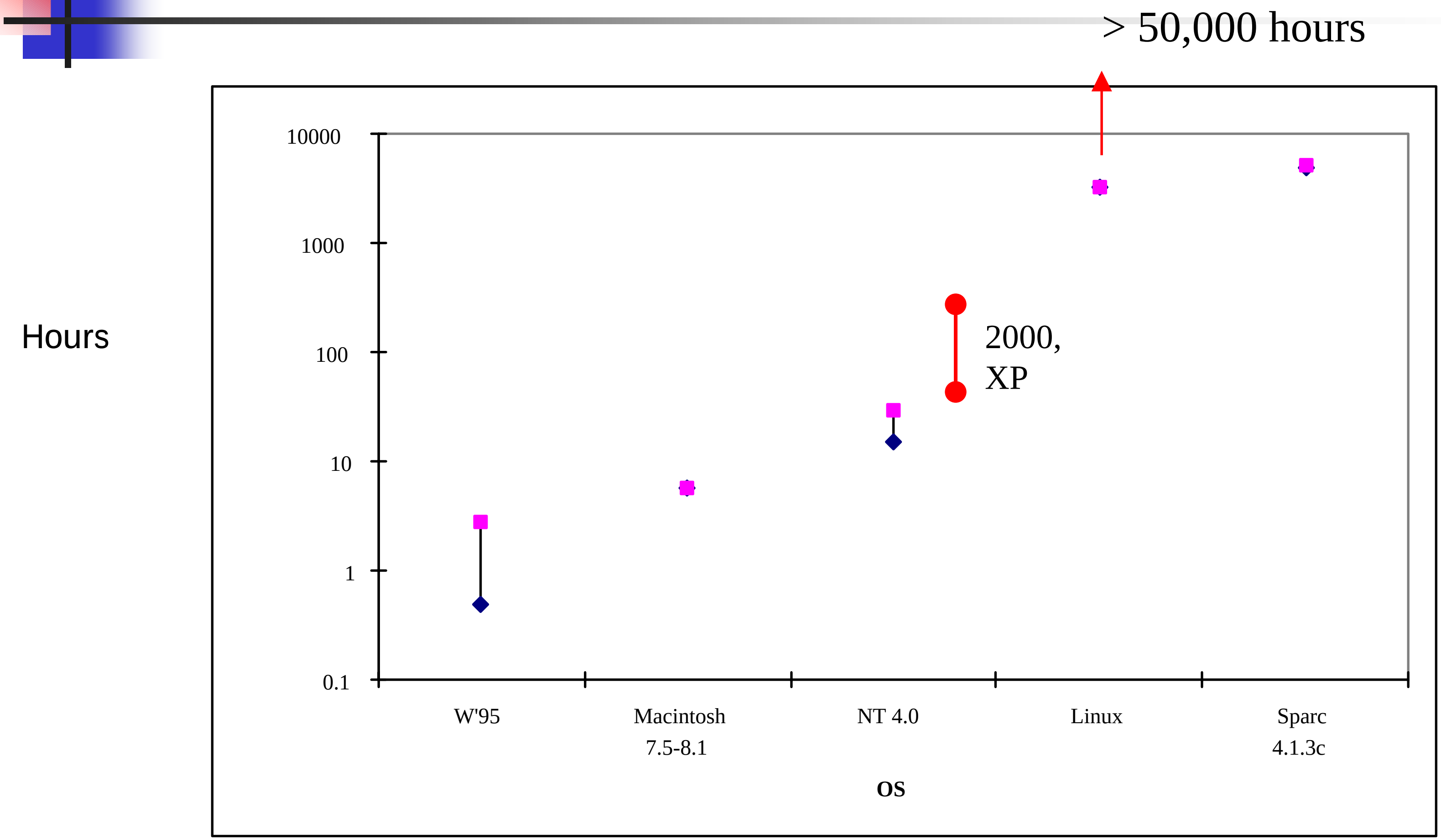
# of Babel The thrives Tower still

- Note the following:-
  - In the 1980s and 1990s we kept pace with the language committees in subsetting to exclude unsafe features through measurement.
  - We’ve now lost the race. ISO C99 is appalling, C++99 is arguably worse and I have recently seen my first critical system in Perl.
  - Dat’s why I sing da blues.

Overview

- Nostalgia ain’t what it used to be
- Things I’m not very happy about
- Things I’m quite excited about

OS Reliability and Open Source



Mean Time Between Failures of various operating systems

Other good signs …

- Note the following:-
  - The CSR is still here after 21 years
  - Much greater awareness of the responsibilities
  - Much better formalisation although pointless bureaucracy remains a continuing threat
  - The MSc at York and other academic initiatives

(However, beware of the legal environment …)

Reference site

For more information, downloadable papers and software, see:-

http://www.leshatton.org/

l.hatton@kingston.ac.uk