# “A legal framework for understanding software systems behaviour"

Professor Les Hatton

The Computing Laboratory, University of Kent
L.Hatton@ukc.ac.uk, lesh@oakcomp.co.uk

Version 1.1: 09/Feb/2004

# Overview

*Introduction*

Some unpleasant truths about software

The nature of legal liability

Conclusions

# An opening question …

*If a software engineer fully 'qualified' and experienced produces a really 'good' system which nevertheless fails and causes damage, is he or she liable and if so, for how much ?*

# Overview

Introduction

*Some unpleasant truths about software*

The nature of legal liability

Conclusions

# Some unpleasant truths …

**Size matters:-**

- Typical legal contract

  Perhaps 20 pages and 100-500 ‘logical relationships’

- Onboard software of Airbus A340, AT&T main switching software …

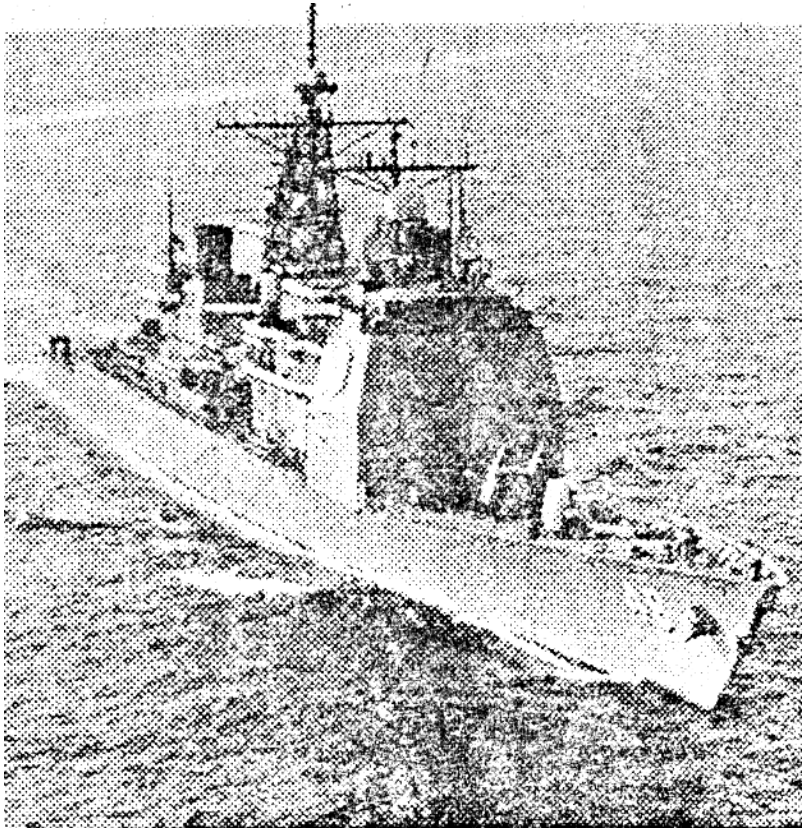
  Perhaps 100,000 pages and unknown but **very** large number of ‘logical relationships’.

# Some unpleasant truths …

- Software fails frequently. When it does it is sometimes impossible to fix
- Many software failures are entirely avoidable
- Software failure is highly unpredictable
- Software development is immature and little progress has been made in reliability in the last 25-30 years
- Many software failures can take an astonishingly long time to appear for the first time
- New bespoke projects have a very low success rate
- It is debatable as to whether software is tangible or not
- Expert opinion is likely to differ very considerably
- The cost of failure is limited only by the imagination. Insurance ?
- Arguably the world's most reliable major software product, (the Linux kernel), fails most of the tests for process quality, (CMM level 1)

# Not a good decade:
# The USS Yorktown in "Please wait ..." mode

"What does 'You have performed an illegal act' mean sir ?"

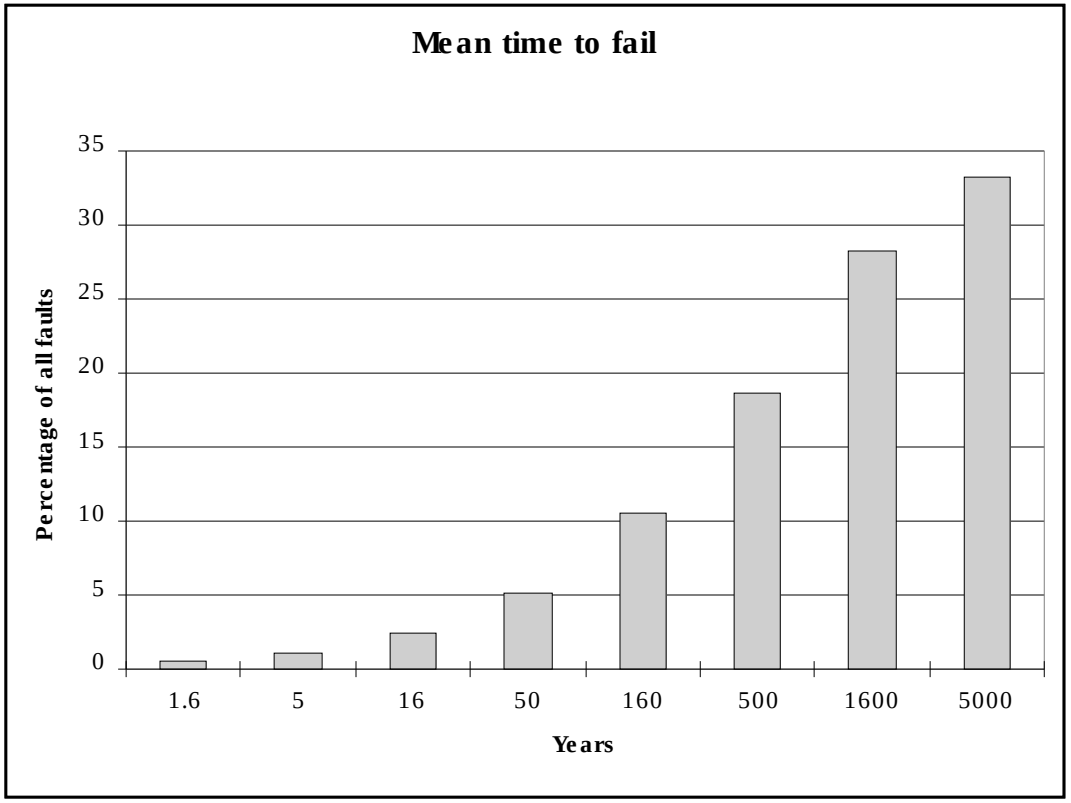


# Not a good decade:
# An Airbus having a bad day

A Tarom airlines Airbus which performed an uncontrolled dive, climb, roll and spin near Orly in 1995 due to ‘a fault in the automatic pilot’. The plane landed safely, a tribute to the pilots’ skill.

# Not a good decade:
# Ariane 5: What goes up ...

# Mean time to fail in Adams (1984)



# Software Process – the layman's guide to the CMM

- A five level model developed on behalf of the US DoD at Carnegie-Mellon in the 80s and 90s
  - Level 1 (Initial – used to be called chaos)
  - Level 2 (Repeatable)
  - Level 3 (Defined)
  - Level 4 (Managed)
  - Level 5 (Optimised or Godlike)
- There are around 50 groups at level 5 in the world, around half of them in India, (who take software development a lot more seriously than we do). BUT … what about Linux ?

# The CMM levels

| | | |
|---|---|---|
| 5 | Optimised | ^<br>? |
| 4 | Managed | ^<br>Full statistical process control; metrics used for defect prevention |
| 3 | Defined | ^<br>Process database, process metrics collected and analysed; risks managed |
| 2 | Repeatable | ^<br>Process focus, software engineering process group; training program throughout |
| 1 | Chaotic | ^<br>Project planning, tracking; software quality assurance, configuration management |

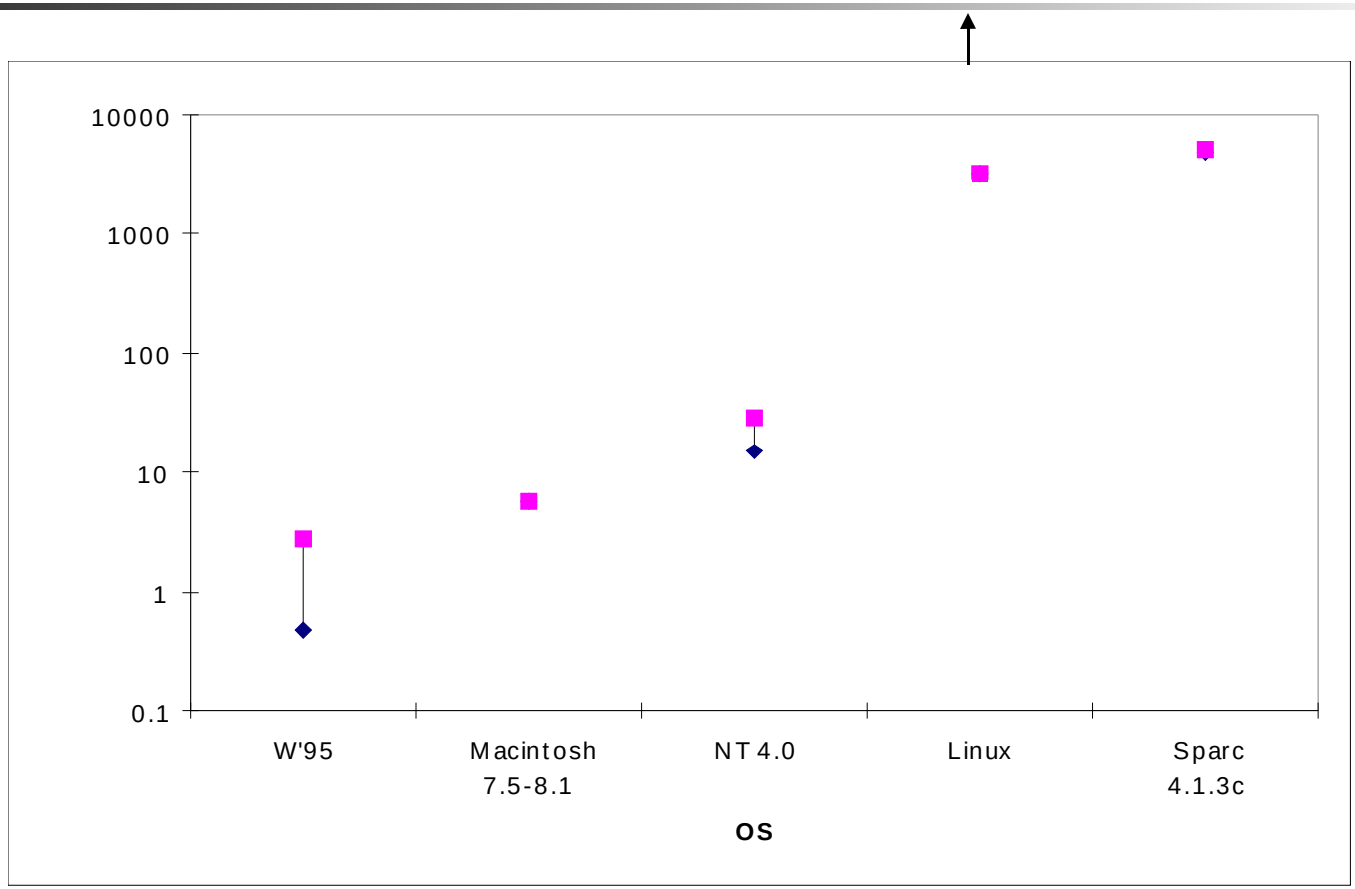
# CMM statistics

Who is where ?

- About 70% of all companies believed to be at level 1
- About 20% believed to be at level 2
- About 9% believed to be at level 3
- Around 50 organisations in the world are at level 5 of which about half are in India
- It takes around 18 months to move between levels realistically
- There is some evidence that software development costs are reduced considerably as higher levels are achieved.

# Do software quality initiatives help ... ?

Hours




Mean Time Between Failures of various operating systems

# Overview

Introduction

Some unpleasant truths about software

*The nature of legal liability*

Conclusions

# The nature of legal liability

- Contract
  - Saphena Computing v. Allied Collection Agencies (1985) (Court of Appeal)
  - St. Alban's City Council v. ICL (1996) (Court of Appeal)
- Tort of negligence
  - ?
- Product liability, (CPA87)
  - ?

## Contract: Goods v. Services

- Goods are covered by SGA 1979, SSGA 94
  - Implied terms for satisfactory quality and fitness for purpose
- Services are covered by SGSA 82
  - "Reasonable skill and care". This is rather less onerous for the developer.
  - In its present form, the relationship of the law to software is uneasy.

# The nature of legal liability

## Contract (some notes)

Bypassing the Goods v. Services debate, the Court of Appeal in St. Albans v. ICL unanimously agreed to infer the existence of an implied term for quality and fitness for purpose in a contract for the supply of software howsoever transmitted. (This follows Lord Pearson's decision in Trollope and Colls Ltd. V. North West Metropolitan Regional Hospital Board).

In Saphena v. Allied Collection Agencies, Mr. Recorder Havery QC also implied a fitness for purpose term in a contract essentially for services and also stated:-

*"Further, even programs that are reasonably fit for their purpose may contain bugs."*

Staughton LJ further added:-

*"… software is not a commodity which is delivered once, only once and once and for all, but one which will necessarily be accompanied by a degree of testing and modification …".*

# The nature of legal liability

## Tort of negligence

- No cases so far (although analogous cases in maritime law)
- Negligence must be proved by plaintiff (non-strict liability)
- Standard tests
  - Duty of care, ('neighbour in law' but more restrictive in European law generally)
  - Standard of care
    - Qualifications attract a higher standard of care as does increased risk. IEC 61508 also correlates standard of care with risk(SIL level). They may also lead to increased liability for economic loss.
    - Failure to observe a reasonable standard of care resolved by **expert witnesses** in UK.
    - Departures from a standard code of practice may not necessarily constitute negligence, (Kelly v. Mears and Partners (1983)).
  - Defect and damage caused
  - Foreseeability
- In the opinion of some experts, software poses no new questions for the tort of negligence.

# Difficulties with expert witnesses

## Expert witnesses in IT can differ by ‘unusual amounts’

In Saphena v Allied Collection Agencies, the court specifically commented on the relative quality of the testimony of the two expert witnesses

In Missing Link Software v. Magee (1989), the expert witnesses disagreements prompted the judge to comment on one witness’s report using phrases such as ‘fundamental errors’, ‘no person of reasonable common sense’, ‘an error of which even a schoolboy would be shamed’, ‘attempting to mislead the court’, and then he really let himself go.

# Difficulties of foreseeability

**An example from real life, Airbus A320 AF319, 25/8/88, (Mellor (1994)):-**

- MAN PITCH TRIM ONLY, followed in quick succession by ...
- Fault in right main landing gear
- Fault in electrical flight control system computer 2
- Fault in alternate ground spoilers 1-2-3-5
- Fault in left pitch control green hydraulic circuit
- Loss of attitude protection
- Fault in Air Data System 2
- Autopilot 2 shown as engaged when it was disengaged
- LAVATORY SMOKE

# Difficulties with insurance

Costs – limited only by the imagination. Try telling this to an insurer.

This was a continuing bone of contention throughout Salvage Association v. CAP Financial Services Ltd. (1990) (covered delay but not complete failure to deliver a system).

# The nature of legal liability

## Product liability (CPA87)

No cases so far

Liability without fault, (strict liability)

Not obvious it even applies to software, (*defines a product as goods or* ***electricity*** *whatever this means*)

Provides a number of defences including the controversial ‘development risks defence’, although in the opinion of a number of experts, e.g. Lloyd (1997), these give scant comfort to software developers.

# Conclusions

## Legal liability

Software should have a legal definition and regime, probably *sui generis*

In their present form, both CPA87 and the law of tort seem peculiarly divorced from software, (still no cases)

The lack of relationship between 'best practice' processes and product reliability does not help with certification initiatives

Dramatic variation between expert witnesses is probably a symptom of IT being more of a fashion industry than having anything to do with engineering. This is something for the IT industry to address as a matter of urgency

If litigation plays a part for example by enforcing insurance, software procurement costs will rise dramatically.

# References

**Adams, Ed (1984)** “Optimising preventive service of software products”, IBM J.Res.Dev, 28(1)

**Hatton, L. (1999)** “Towards a consistent legal framework for understanding software systems behaviour”, LL.M. thesis, University of Strathclyde Law School

www.leshatton.org

**Kaner, C., Pels D**. (1998) “Bad Software”, Wiley ISBN 0-471-31826-4

**Lloyd, I.J.** (1997) “Information Technology Law”, Butterworth, ISBN 0-406-89515-5

**Mellor, P**. (1994) “CAD: Computer aided disaster”, High Integrity Systems Journal, 1(2)

**Reed, C.** (ed) (1996) “Computer Law”, Blackstone, ISBN 1-85431-448-3