Seeing beyond the Post Office Horizon

Harold Thimbleby

Swansea University

Abstract *The Post Office Horizon case is the largest miscarriage of justice in the UK. While voiding convictions and providing adequate compensation is an immediate priority, the technical complexity of the case and the on-going Post Office Horizon Inquiry have paralysed decision making, and distracted from the strategic urgency of addressing the poor IT culture that led to and fed the problems. Horizon is a symptom of deeply-entrenched cultural problems with IT, including poor programming causing errors, and undermining the reliability of computer evidence for use in court. Failure to understand IT led to the misleading common law presumption that computer evidence is reliable, which undermines disclosure requirements in courts and further reduces scrutiny of computer evidence. Legal reasoning on the reliability of computers in court is flawed. Throughout the Horizon scandal, the inability to distinguish naïve and dishonest IT optimism from rigorous scientific thinking and evidence ensured that incompetence knew no limits. In short, what started (put charitably) as incompetence transformed into a scandalous "delay and deny" cover-up.*

IT problems have a wide impact in many areas far beyond the Post Office Horizon scandal. As AI gains wider use it will create worse problems, particularly for legal evidence. Raising, debating and taking steps to manage these generic and besetting IT problems are of fundamental importance in the digital age to achieve a safe and just society.

1 Introduction

Lee Castleton opened his Bridlington Post Office in 2003, but soon found discrepancies in his financial accounts run by the Post Office system called Horizon. Castleton made 91 calls to the Horizon helpline. By March 2004, his unexplained losses had grown to £25,000. He was suspended by the Post Office, and taken to court. Summing his case up, the judge said *"The losses must have been caused by his own error or that of his assistants,"* and *"it is inescapable that the Horizon system was working properly in all material respects."* Castleton was left with costs of £321,000. He went bankrupt.

© Harold Thimbleby 2024. Cite as H Thimbleby, "Seeing beyond the Post Office Horizon," *Safe AI Systems*, Proceedings of the 32nd Safety-Critical Systems Symposium, Mike Parsons, ed, **SCSC-188**, pp243–252, Safety-Critical Systems Club, 2024.
Published by the Safety-Critical Systems Club. All Rights Reserved

2 Background

Over a 16-year period starting around 1999, the Post Office prosecuted over 900 postmasters for shortfalls the Post Office claimed were theft, false accounting, and fraud.

We now know these shortfalls were due to errors in the Post Office's Horizon and other software as well as unauthorised remote access to accounts. Although some postmasters have had their convictions overturned by the Court of Appeal, as of January 2024, most of those wrongly convicted are still waiting to have their convictions overturned. None have had adequate compensation. The Court of Appeal Judge, Peter Fraser said the Post Office's malicious prosecutions and failures were so egregious as to make their prosecutions an affront to the conscience of the court.

A public inquiry[1] is ongoing, and is revealing more problems and mendacious behaviour daily. The police are now investigating both the Post Office and Fujitsu for perjury, fraud, and other offences. (Horizon was originally implemented by ICL, a British company that was later taken over by Fujitsu.) Ironically, the Inquiry itself has become a standard excuse not to say or do anything: Fujitsu, for example, has repeatedly said they cannot comment while the Inquiry is ongoing.

At root, the horrendous Post Office failures can be traced back to poor software and poor IT culture and poor management, plus a culture of denial that spread across the Post Office.

Initially this might be explained as a toxic mix of naïve optimism of people with little technical knowledge, perhaps being motivated by first needing to present Horizon as a wise investment, then needing to keep Horizon looking like a successful political initiative, and, finally, deliberately avoiding upsetting wider UK-Japan relations (Fujitsu is Japanese).

3 Prosecutions

When cases were prosecuted, one might have assumed that the courts would be critical and objective, but there is a common law presumption that assumes computers are correct, so court scrutiny was minimised (Marshall *et al*, 2021). Post Office prosecutors never needed to face serious cross-examination on the Post Office claims about the reliability of Horizon. Under the presumption, defendants have no way of finding out what documents or computer records might show relevant computer errors to support their case. Defendants trying to find such evidence would be accused of fishing.

Problems were covered up by management apparently more interested in Horizon maintaining a public image as a successful project and maintaining the public

[1] https://www.postofficehorizoninquiry.org.uk

image of the Post Office as a successful company. Some of the cover up and inertia against doing anything may have been sustained by believing the successful court prosecutions exonerated Horizon — to this way of thinking, one successful prosecution after another reinforces the apparent reliability of Horizon. To another way of thinking, the extraordinary numbers of prosecutions and the consistent complaints by postmasters on the unreliability of Horizon suggests a systemic failing, and certainly a profound lack of curiosity.

One argument presented in court was that if a system like Horizon correctly processes thousands of transactions every day, as no doubt it did, then it must be reliable, and therefore any postmaster's claims that Horizon is unreliable cannot be taken seriously. This reasoning is false, and is an example of the Prosecutor's Fallacy[2].

The common law presumption might perhaps be justified because a typical court cannot be expected to understand technical arguments about computer programming or bugs one way or the other. If computer evidence wasn't assumed to be reliable, perhaps we would all be trying to argue against speeding fines, unpaid parking tickets, bank fees, and much more. It is plausible that the Law Commission who created the presumption were as technically incompetent and desperate to cover their ignorance as the Post Office. Indeed, the Post Office itself gave evidence to the Law Commission to encourage creating the presumption. The presumption significantly reduces the cost of prosecutions relying on digital evidence regardless of whether that evidence is valid.

The common law presumption papers over large cracks. In reality, we have no idea whether any computer evidence is reliable, because the people who built the computer systems that produce the evidence do not even have to be competent. Their programs could easily make a mess of any evidence. A court simply cannot tell if the software they are relying on for evidence is wonderful or terrible. The common law presumption side-steps this worry by saying the questions don't even need to be raised.

Inside and outside of courts, then, we are unable to distinguish between safe and unsafe computer systems. We cannot tell whether the systems providing evidence

[2] Consider a simple analogy between accounting faults with mechanical faults in cars. From publicly available UK figures, the chance a random car is under repair on any given day is about 1 in 10,000, but this low figure does not mean that you would most likely be lying if you claimed your car was faulty. Your car (if faulty) is not a random car: your car has not been selected at random from all cars in the UK, most of which are not faulty, as it is one of the few that have problems. Whether your car is faulty should be established by examining *your* car and by identifying one or more faults, not by abstractly comparing it to the thousands of *other* cars that do not have faults. Unfortunately the common law presumption removes pressure on the courts to examine relevant evidence, and then it becomes easy for prosecutors to fallaciously claim the average evidence is relevant — why should they disclose faults when the law presumes there are none? Indeed, the presumption means the defendant has to prove their car is faulty when the prosecution does not even need to reveal the specific fault it may have. The defendant may know their car does not work, but the prosecution need not disclose the specific reason it is faulty — so the prosecution can argue, if the defence is unable to identify a specific fault, the claim that there is a fault must be a lie.

are reliable (and managed reliably), and we certainly cannot tell whether the expert witnesses are competent to provide reliable professional advice to the court.

4 Delusions

However the seeds were sown, soon claiming the correctness of Horizon became an article of faith, which was reinforced by silencing critics, like Second Sight. There was a self-fulfilling cycle of not disclosing evidence of problems, and believing there was no evidence of Horizon problems because nobody had disclosed any. And, of course, successful convictions were taken to confirm that Horizon was blameless, which would in turn drive even more prosecutions. The Post Office was soon prosecuting one postmaster a week.

Then there was the delaying and denying compensation to postmasters, and lying to Parliament.

All of it can only be understood as a shared delusional world, paralysing action, paralysing all critical thinking. To do anything, like compensating a few postmasters fairly, would expose decades of staggering incompetence and open flood gates of conscience for all the many people who have done nothing.

Nobody wants that cognitive dissonance (Tavris & Aronson, 2015), as it is *far* easier and more comfortable to stay in denial, especially if no colleagues understand computers. Indeed, the Post Office is a textbook case of the Dunning-Kruger Effect (1999): inaccurate self-assessment leads people to make bad decisions, and inhibits them from noticing or addressing their shortcomings. They are then unable and unwilling to improve themselves. *Unconscious incompetence*.

News reports, like many in the persistent *Computer Weekly* and *Private Eye*, were trying to cover complex cases against the powerful public persona of the Post Office. They never gained traction. Nick Wallis's powerful book, *The Great Post Office Scandal* (Wallis, 2015), was detailed and thick, but perhaps overwhelming in the wrong way. Even the Post Office Inquiry was as turgid as it was damning.

And then the postmaster, Alan Bates, was portrayed as the protagonist in an extraordinarily powerful 2024 ITV drama *Mr. Bates vs The Post Office*, playing in our own homes very human stories of injustices and suffering. The drama turned the rumbling scandal into front page news, and quickly into a political priority. Suddenly, one-time Post Office CEO, Paula Vennells handed back her CBE. Suddenly, Prime Minister Rishi Sunak promised legislation to quash convictions wholesale – dramatically challenging British traditions about separation of powers.

Bad software is nothing new, but the constricting cultural spiral of:

digital incompetence → ignorance → denial → corruption → institutionalised incompetence

drove the Post Office Horizon scandal. The Post Office has the power, has the law on its side, has the legal and financial resources, has managers and politicians

needing it to stay on brand; and the Post Office incentivised its prosecutors. The Post Office only has one shareholder, the Government's Department for Business, Energy & Industrial Strategy, and presumably avoided all the scrutiny that is routine in normal diverse shareholder meetings.

The Post Office never needed to reflect why it was not just claiming to be victim, but was the prosecutor as well. We can only speculate on its lack of diversity and lack of technical skills, but the Horizon scandal is the culmination of four centuries of, literally, just paper-pushing and failing to prepare itself to understand digital technology, getting out of its depth, maybe at first unwittingly but soon ending up in a whirlpool of intentional cover-up and denial.

4 Beyond the Horizon

Apart from sheer scale, there is nothing unique in the Post Office's drift into failure. There are other serious failures and miscarriages of justice over faulty software, as terrible to the individual victims concerned but harder to acknowledge because there isn't a popular drama to highlight the human tragedy in a way the public cannot ignore.

The new booklet, *Patient Safety — Stories for a digital world* (Thimbleby & Thimbleby, 2024) gives many examples paralleling Horizon taken from across the NHS and international healthcare — situations caused by misunderstanding digital systems, leading to unnecessary patient harms and staff convictions. In some ways the NHS mirrors the Post Office — centuries of routine, very human, work suddenly computerized and made incomprehensible to everyone in the organisations.

One example concerns over 70 nurses who were disciplined. Some of the 70 nurses were prosecuted for alleged criminal negligence, on the basis of missing patient data that should have been recorded by the nurses. The court found that the data been deleted by an engineer. How could the hospital have been so blind to their own IT failings?

Or consider that when an anaesthetist presses a button to put a patient to sleep, by law they have to be competent and must have up to date qualifications backed by substantial training. Yet what happens when they press that button is anyone's guess, because a computer will do it, and we have no idea whether the computer is reliable: there are no regulations governing the qualifications, supervision, oversight, or insurance of people programming systems, whether that is for accounting (as in Horizon) or delivering anaesthetics (as in hospitals). It is ironic and dangerous that programmers need have no insight or supervision into what they are doing. Nobody has to take responsibility or sign off that their code is safe.

Many developers don't even realise they are incompetent. They know so little about good programming, they don't realise they are not good programmers. It is called, as hinted above, *unconscious incompetence*. While our laws are written and voted on by people with negligible computer knowledge or expert advice, our laws will hold back and undermine both safety and justice.

There are ways to fix this sorry mess.

Centuries ago, quack doctors were a danger to society, and so the government responded by passing the Medical Act of 1858 because, in the Act's opening words, *"it is expedient that Persons requiring Medical Aid should be enabled to distinguish qualified from unqualified Practitioners."* We now think the idea of registering educated and qualified doctors is self-evidently sensible. Indeed, one of the solutions Dunning and Kruger proposed in their original 1999 paper was education, as it is only with education that we are reliably calibrated to see (and for our employers and clients to see) our skills and limitations.

Not all education is equal. In the UK, there is a long traditional of professionalising computer education, leading to professional recognition such as becoming an incorporated or chartered engineer, but these qualifications do not in and of themselves assure an engineer is a competent developer (or auditor, or tester, etc) able to develop systems that generate electronic evidence of probative value.

In comparison, to be a qualified electrical engineer who could, for instance, legally install electrical wiring, very specific technical qualifications are required (e.g., AM2E) followed with regular continuous professional development (CPD) to ensure continued competence. So, instead of certifying existing, generic computer science education qualifications, we need to develop new educational standards that are designed to ensure more reliable code and more reliable forensic evidence generated by that code. Again, we take regulation of electrical engineers for granted — yet we do not worry that high voltage electrical installations may be controlled by unregulated computer systems, implemented by unregulated programmers. (The unsafe, poor programming of a regulated electrical safety test tool is provided in Thimbleby, 2022.)

Just as there was resistance to the Medical Act (and the Pharmacy Act, regulating pharmacists, that soon followed it), there will be strong resistance to analogous legislation for computers. But resistance does not mean the idea is wrong; arguably, resistance is confirmation that there is an unsafe culture that needs addressing by such an Act.

It is time, then, that the Government legislated so that everyone can avoid being unwitting victims of quack computer systems. A computer Act based on the uncontroversial Medical Act would require programmers to be registered, have a decent education (to be defined), and have respectable, relevant qualifications, and, as in healthcare, the professionals around them would also need to be suitably qualified.

Of course, this wouldn't apply to hobbyists and children programming. But if you wanted to build a serious system then you would have to be registered as competent.

Working out how to crack the chicken-or-egg problem will not be trivial: nobody will want qualifications until they are required, and it is unreasonable to require qualifications until there are enough people with them and enough appropriate qualifications available. Some incremental process, perhaps implemented over a decade, will be required. Perfection is the enemy of the good, and here perfection may be set up as a strawman; on the contrary, there is a lot that can be done now that is

worth doing, even though less than immediate perfection. It is a big job — but very worthwhile.

The point of this article is to provoke and stimulate debate, showing that there is models in other critcal areas, and a plausible start to a solution. Regulations requiring competence, even as taken for granted in other industries, may not be sufficient to solve all the types of problems covered in the story of Horizon and outlined in this article. There may be other possible solutions that are better (more are discussed in Marshall *et al*, 2021), or that can be leveraged on better qualifications or clearer definitions of competence. If there are better ideas, or ways to improve the ideas suggested here, we urgently need to find them — and meanwhile we should start moving in the right direction.

Peter Fraser, a judge who heard appeals arising out of the Horizon cases, said that the Post Office's stubborn pursuit of convictions over Horizon was equivalent to maintaining that the earth is flat. It is time, then, that we started looking beyond the Post Office Horizon to find broader more general solutions, not just to avoid another Horizon, not just for postmasters, but to build a more rounder, safer, world for everybody using and affected by computers.

APPENDIX: Example poor code from the Horizon Inquiry

The code examples in this appendix are taken from section 7.3, *Report on the EPOSS PinICL Task Force, Post Office Inquiry document FUJ00080690.*

The examples of poor code below do not in themselves prove that the programs do not work, or even that the programs this code has been sampled from are buggy. What the examples do show is that the code is unprofessional and unreliable. The code is so bad that it would be pointless to debug it. Its confused and confusing style suggests there were no adequate requirements. Without clear requirements there is no standard to even define correctness.

Compare this situation with other industries, where poor quality can be illegal without having to prove a product is actually dangerous. Electrical installations, for example, are required to be wired to the relevant safety standards, and checked as compliant to those standards. Sloppy wiring is (in the appropriate circumstances) a criminal offence, even if the specific details of the sloppy wiring may have no direct causal link to any particular problems.

Just as electricians are expected to use appropriate standards and test equipment, there are plenty of standards and test tools available for programmers. The code examples below, taken verbatim from Inquiry evidence, imply that basic standards and software tools were not used (or, if used, their results were misunderstood or ignored).

This evidence shows extraordinary incompetence and lack of awareness of basic IT skills. Failing to notice or fix the problems shows extraordinary managerial incompetence.

Example 1

```
Public Function ReverseSign(d)
    If d < o Then
        d = Abs(d)
    Else
        d = d -(d*2)
    End If
    ReverseSign = d
End Function
```

This pointlessly obscure and inefficient function `ReverseSign` returns the negative value of its parameter.

Note that the variable `o` [sic] was copied from the report to the Inquiry. It is probably supposed to be zero, not the letter o. It is possible that `o` is a global variable so this code would compile and run, but it is unlikely to do what was intended (however it will work correctly provided the value of `o` ≤ 1).

On a typical computer, `Abs(d)` will require a function call and a further test. Since the if statement guard ensures `d < 0`, it would be faster to write `d = -d` instead of `Abs(d)`, as this is what `Abs` will do after its own internal repeat test.

Since compilers will typically optimise `d = d-(d*2)` to `d = -d` this expression has no advantage over the clearer `d = -d`. In fact, the subexpression `d*2` can overflow and cause incorrect results that `-d` does not generate. Many optimisers do not implement checks for overflow, so in this case optimisation may make the code more robust. (It would seem bizarre for a programmer to *want* undetected incorrect results under overflow conditions, and unprofessional not to comment the code to warn of its peculiar properties.)

A function like `ReverseSign` could be used for tracing or assertions, such as checking there is no overflow (e.g., on a twos complement machine, `-MinInt = MinInt`). There are no signs of tracing or assertions in any code exhibited to the Inquiry.

Example 2

```
If lstockrootnode = 3013 Or lstockrootnode = 3016 Then
    bremedprods = False
    intbalancerootlevel = 5
    lbalancerootenode = 3017
    If lstockrootnode = 2493 Then
        bremedprods = False
        intbalancerootlevel = 3
        lbalancerootenode = 3006
    End If
Else
    bremedprods = True
    intbalancerootlevel = 5
    lbalancerootenode = 3017
End If
```

Node numbers here are hard-coded and not documented. Typos in the ID values (`3013, 3017, 2493`, etc.) would be very hard to detect. The purpose of the conditions are not commented.

The shaded code above is unreachable, as after testing `lstockrootnode` is `3013` or is `3016` it cannot possibly become `2493` with the code shown. If the inner test `lstockrootnode = 2493` is not incorrect, the code is pointless. Indeed, the line of code `bremedprods = False` repeats an assignment a few lines earlier and is pointless — or perhaps the programmer meant to assign to a different variable or assign a different value to the variable?

Furthermore, the variables `intbalancerootlevel` is set to `5` and `lbalancerootenode` set to `3017` *regardless* of the main test conditions — the assignments are duplicated.

Overall, the code implies the programmer did not understand (and/or did not check) what was being written. Standard static analysis tools would have detected the unreachable code and the duplicated code; and standard testing practice would have been to ensure coverage (i.e., that all code was tested), and that cannot have happened with this code.

Example 3

```
If s<>”” Then
    Do
        If s<>”” Then
            // Significant code removed to save space
            Exit Do
        End If
    Loop
    End If
    Next
End If
```

At face value (i.e., the report comments out any details to the contrary), the loop is only executed at most once and is equivalent to the following clearer code:

```
If s<>”” Then
      // Significant code removed to save space
End If
```

It is possible the commented-out large body of code (the comment is in the Inquiry evidence) contains repeat loop commands, but if so the report writer who commented out the code missed them, and it should have been simplified (for instance) with function calls.

We also note the code in the report has incorrect indentation, a dangling `End If`, and the loop control `Next` appears to be outside any loop (at least any loop in the code as presented).

Example 4

```
Select Case Val(ObjAtributeValue(SCAMapping, "Data.Leaf.N")
    Case
```

```
            99995026, 99995027, 99995028, 99995029,
            99995030, 99995031, 99995032, 99995033,
            99995046, 99995056, 99995057, 99995058,
            99995059, 99995060, 99995061, 99995062,
            99995063, 99995064, 99995065, 99995066
               stxn = stxn &
            ObjMake("SuspenseContainer","S")
               Exit Do
      Case Else
  End Select
```

The object value IDs here are hard-coded and have no documentation. Typos would be very hard to detect.

The fact that the cases cover all consecutive IDs `99995026-33` and `99995056-66` inclusive, plus an isolated value `99995046`, is not clear. Anyone reviewing this code needs to know what the ID values mean, and needs to know how they are related: how else can a reviewer check whether, say, `99995035` has been missed out or not?

The line `stxn = stxn & ...` appears to be bit fiddling, with no documentation on the meanings of the bits affected; this sort of code is very obscure (e.g., what it does is not commented); it should have been written with explicitly-named methods (that would also hide the low-level bit implementation).

Acknowledgments Many thanks to Martyn Thomas who made many very perceptive and helpful comments.

Disclaimers The author was not funded for this article, and declares no conflicts of interest.

References

P Marshall, J Christie, PB Ladkin, B Littlewood, S Mason, M Newby, J Rogers, H Thimbleby & M Thomas, "Recommendations for the probity of computer evidence," *Digital Evidence and Electronic Signature Law Review,* 2021. DOI 10.14296/deeslr.v18i0.5240

H Thimbleby, "Cowboy digital undermines safety-critical systems," in *Safer Systems: The Next 30 Years, Proceedings of the 30th Safety-Critical Systems Symposium*, M Parsons & M Nicholson eds., SCSC-161:203–226, Safety-Critical Systems Club, 2022.

H Thimbleby & P Thimbleby, *Patient Safety — Stories for a digital world*, https://www.harold.thimbleby.net/booklet, 2024.

N Wallis, *The Great Post Office Scandal*, Bath Publishing, 2021.

C Tavris & E Aronson, *Mistakes Were Made, but Not by Me: Why We Justify Foolish Beliefs, Bad Decisions, and Hurtful Acts*, Pinter Martin Ltd, 3rd ed, 2015.

J Kruger & D Dunning, "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments," *Journal of Personality and Social Psychology*, **77**(6):1121–1134, 1999. DOI: 10.1037/0022-3514.77.6.1121