About the SCSC

The Safety-Critical Systems Club (SCSC), **thescsc.org**, established in 1991 by the IET, BCS and DTI, is a professional network for sharing information about safety-critical systems. These systems on which life and property depend are typically complex and often software and data-intensive. The SCSC therefore focuses on current and emerging practices in software and systems engineering. It produces guidance on systems assurance, security, data, safety culture, services and other areas. It holds an annual 3-day technical conference, 1-day seminars, hosts a web site and publishes a journal and newsletter. Many large organisations hold corporate membership of the SCSC. The SCSC has published its position on Post Office Horizon (see Appendix A).

1) The current common law (rebuttable) presumption is that computers producing evidence were operating correctly at the material time.

(a) Is this presumption fit for purpose in modern criminal prosecutions?

This presumption is not fit for purpose given the number of computer failures that we all experience as part of our daily lives. Of course, the majority of failures are not serious, but some failures can lead to actual harm.

Indeed, the current presumption places more confidence in computer evidence than on other kinds of evidence, which is not justifiable. Furthermore, it is increasingly easy to modify or fake computer evidence, especially using Artificial Intelligence (AI) techniques.

The existence of many hours of successful operations or millions of transactions is not proof of absence of failure. Standards such as BS EN 61508-7 Annex D show the difficulty of making a ‘proven in use’ claim that is statistically sound for software.

As an example, the Boeing 737 MAX aircraft operated globally, amassing many hours of operation, prior to the crashes that killed hundreds. The hours flown without incident were not an indicator of no problems with the aircraft, but indicated that the conditions required to demonstrate the dangerous flaw in the aircraft had not been reached during those flights.

(i) Please specify why you gave this answer

The failure of the Post Office Horizon system is the most recent high profile example but there have been many other less prominent failures across multiple disciplines and industries including systems for control, navigation, administration, banking, and medical. Some recent examples include:

1. A faulty software update from cybersecurity vendor CrowdStrike on 19 July 2024 caused about 8.5 million computers running Microsoft Windows to crash, affecting airports, supermarkets and banks.
2. On 28 August2023, a software defect at NATS (the UK air navigation service provider) led to more than 700,000 air passengers impacted during a peak travel period.
3. On 31 January 2025, customers of Barclays Bank were left unable to access app and online banking services. The IT outage lasted three days.
4. In May 2024, the BBC News reported that:
    a. IT system failures have been linked to the deaths of three patients and more than 100 instances of serious harm at NHS hospital trusts in England.
    b. A Freedom of Information request found 200,000 medical letters had gone unsent due to widespread problems with NHS computer systems.
    c. Nearly half of hospital trusts with electronic patient systems reported issues that could affect patients.

Systems and software defects can be latent. The system or software can appear to be working correctly for years, yet be producing incorrect outputs. Post Office Horizon is just one example where computer evidence has been found to be unsound. There are many others which indicate that the computer systems we rely upon are not always as robust or resilient as they should be, and that evidence from their operation may not be relied upon.

(b) How easy or difficult do you believe it is at present for this presumption to be effectively rebutted?

It is very difficult to rebut the common law presumption. It places the onus on the defendant rather than on the prosecution. (In the case of Post Office Horizon, the legal system was heavily loaded against the defendants: Post Office Limited (POL) carried out the original investigation and withheld evidence from the defendants. It was also able to spend far more on legal fees than the defendants.)

(c) What barriers do you see in effectively rebutting this presumption?

For defendants to rebut the common law presumption, they would need to understand and have access to the computer evidence, and know how to locate suitable representation for them in respect of this evidence. Most legal representatives would not have the expertise to offer this support and third-party experts would be needed. All of this limits defendants' ability to rebut the presumption.

In more detail there are several barriers that might prevent effective rebuttal.

(i) Assessment - it is difficult to assess the potential for the computer system to cause harm as intended usage may be different to that assumed, and it may change over time.

(ii) Development - it is difficult to establish what processes were used to address risks and how the system was developed and verified.

(iii) Expertise - assessment requires experts, who are difficult to find and can be expensive.

(iv) Vendor - the cooperation of the system or software supplier(s) is essential; however they may not be based in the UK. Usually only the supplier has the data, tools and skills to interpret the evidence base. There may also be commercial issues such as Intellectual Property Rights (IPR) that protect the supplier's business and restrict access.

In principle, these obstacles may be surmountable but in practice they make challenging the presumption impracticable.

Note SCSC SG Horizon statement para 6: *"Where there are disputes involving computer-based systems there must be fair treatment, i.e. where relevant, there must be access for both sides to technical experts who in turn must be given access to appropriate software and data."*

(i) Please give examples where possible.

In the case of Post Office Horizon, POL made a civil claim against Lee Castleton to recover £23,000 in shortfalls. Lee Castleton represented himself in court. The judge found in favour of POL and awarded £321,000 in costs against Lee Castleton.

The same financial imbalance is true of any individual or Small/Medium Business (SME) facing a corporation or public body in court.

Another example is the Toyota unintended acceleration issue, which was eventually settled out of court.

2) Are you able to provide examples from other jurisdictions or situations where the reliability of software must be certified?

There are many application domains in which very high reliance is placed on computer systems and software. This includes safety-critical sectors such as civil aviation, defence, rail and nuclear power generation.

a) As examples of good practice?

These domains must demonstrate good engineering processes according to defined management systems and standards. They must also show they have reduced the risks of the systems to ALARP (As Low As Reasonably Practicable). Software is typically developed to standards and industry guidance such as RTCA DO-178C/EUROCAE ED-12C and BS EN 61508-3 resulting in compliance claims. Other aspects of systems development are available on data safety and service assurance, e.g. https://scsc.uk/r127J:1 and https://scsc.uk/scsc-156D.

Many of these standards and guidelines use a ‘levels’ or ‘grading’ scheme where the amount of effort employed in activities such as design, testing and analysis is proportional to the perceived risk the system or software presents. We suggest that a similar scheme may be useful in this context, where the nature, type and detail of the evidence is proportional to risk.

Many standards, e.g. UK DEF STAN 00-056, require the development of an assurance case. Furthermore, it is not sufficient to just follow these standards or guidelines, it is also necessary to have a strong safety culture and a safety management system in place. See, for example John Rushby: New Challenges In Certification For Aircraft Software (https://www.csl.sri.com/~rushby/papers/emsoft11.html).

b) As examples of things to be aware of?

We recommend that a holistic systems view is taken. It is not just software: the whole system must be considered including hardware, other connected systems

and importantly the data consumed and produced by the system and the interaction with users.

Note that most commercial software development follows a very different business model to safety-critical software development, focusing on functionality and time to market rather than reliability.

Even so, this does not mean that software written to even the most stringent standards is defect-free. Studies, e.g. https://www.adacore.com/tokeneer and https://apothecaryshed.com/wp-content/uploads/2018/11/secure_software_process1.pdf, show that *even the best software development processes result in about 1 defect per 1,000 lines of code*. It follows that a software program consisting of 1 million lines of code could easily contain at least 1,000 defects even if it were developed to the state of the art - many more defects if it were not.

The situation is getting worse rather than improving. The size and complexity of modern systems involving software is now such that it is impracticable to comprehensively test or analyse them. Complexity causes unexpected failures.

Note that most software does not come with any meaningful warranty. If the vendor is unwilling to warrant the software, it is surely reasonable that a court would consider the resulting computer evidence untrustworthy.

3) If the position were to be amended, what in your opinion would be the most appropriate and practicable solution given our aims and objectives set out above? It would be helpful if your answer could address as many of the below as possible:

Each computer system, including its software, should be assessed on its merits, and the context in which it is used. The assessment needs to consider the potential for the computer system to cause harm, and what processes and techniques are used to address the risks.

Harm can occur in many ways, including: financial losses; mental health impact including depression, stress or anxiety; loss of employment; loss of future prospects; physical health impact, and, as in the case of Horizon, suicide. All routes to possible harm should be considered.

Note SCSC SG Horizon statement para 5: *“We propose that organisations relying on computer-based system evidence in court should, where challenged, be required to justify that the system, including aspects such as hardware, software, data and service*

delivery, is reliable. Furthermore, the evidence should be shown to be trustworthy. The justification should also show appropriate confidence in use, including in the way that reported problems are managed. Courts should not accept evidence relating to the computer-based system without this justification."

Evidence presented in court relating to systems or software must surely be "credible" in some sense, which means it should have been produced in a methodical, documented and ideally reproducible way. There are three primary possible strands for evidence of credibility:

1. The system and software was produced with appropriate care and diligence throughout its development. This includes good design and assessing the risks of the system.

2. Appropriate verifications (including tests) were performed throughout its life-cycle.

3. The system and software was operated and monitored with care and proper controls in place, including through all evolution and maintenance. Faults have been identified and fixed appropriately.

There should be solid evidence from all three strands bolstered by backing evidence about how it was produced, and would, ideally, include independent evidence from a third party, e.g. tests, audit or review reports.

This evidence should be structured and documented in the form of an assurance case or other argued justification so there are a set of conclusions that are logically drawn from the evidence. The SCSC has published guidance on good practice for assurance cases https://scsc.uk/scsc-159.

One major issue is that all such evidence will have to be analysed and interpreted for meaning: often this can only be done by the supplier of the system or software. Third-party interpretation is possible with experts, but very difficult to achieve in practice.

If backing evidence is missing, or information about dates, processes used to produce it or doubts about possible modifications or provenance, then the court must take this into account

a) What procedural safeguards need to be in place to ensure your proposed solution is effective?

There should be a check that the evidence related to the computer system is:

(i) Complete as far as relates to the case

(ii) Unaltered

(iii) Reproducible if required

(iv) Has been produced in a methodical and documented manner, and by competent personnel

(iv) Is interpreted in a fair manner, i.e. available to both sides experts who are free to draw their own conclusions

(v) Is independently checked by a third-party expert where possible

(vi) In addition the environment and configuration of the system when it was produced should be available for examination

b) How might we ensure that any proposed solution is, as far as is reasonable possible, future-proofed?

The steps outlined above do not rely on any particular technology or tools. However they do require expert skills which can become out of date or lost. There is not much that can be done about this: if the computer system was developed a long time ago, there will be few people around today who understand its design rationale and details of its implementation.

The use of AI to create credible but false evidence is a concern.

c) How might we ensure that any proposed solution is operationally practical?

It may be possible to make good progress towards better systems and software reliability by aiming for greater consensus regarding good practice. Government is in a good position to appeal to various interested parties regarding their responsibilities to whoever they are representing or providing products or services to. There has been discussion following the events at the Horizon inquiry of a need for greater candour, perhaps using a code of practice. That would help build a greater level of trust between organisations.

Inevitably, there will be some organisations who will be reticent, but there may be opportunities for independent oversight to assure products and services. That

assumes the overseer has the necessary expertise which might be confined to the design organisation.

Increased oversight will mean increased cost in time and resource, to mitigate this it is recommended a risk based approach is adopted where the level of scrutiny is proportional to the perceived risk. See response to question 1. c).

Encouragement of good practice by introducing accepted ratings of systems and software (e.g. A to G), where these are assigned by an independent assessment body. The availability of such ratings would have clear implications for the validity of evidence generated from such systems.

Systems and software engineering qualifications and formal competency management could become mandatory for those involved with dependable computer systems.

There is clearly a role of professional societies, e.g. the BCS to set minimum competency levels, and possibly to assess systems.

d) If your proposed solution requires the use of expert witnesses (either jointly or singly instructed), what expertise and qualifications would that person require? To your knowledge are there sufficient such people at present?

Generally only the system or software developer has the detailed knowledge of the software and data. Additionally, the type of software likely to appear in court will typically be poorly produced, poorly documented and poorly tested and therefore very difficult to understand. This makes it difficult to use expert witnesses who are third parties, likely unfamiliar with the detailed workings. Given sufficient time and resources experts can examine and draw conclusions from evidence but it is expensive and time-consuming. Experts will need to demonstrate expertise with the technologies used and with the domain.

4) In your opinion, how should ‘computer evidence’ for these purposes be best defined?

We propose that ‘computer evidence’ be defined as ‘evidence that has been generated by a computer system or software’. This may or may not be trustworthy, depending on the correctness of the software itself, but also on the data on which it relies, on the algorithms used, the surrounding software such as virtualisation platforms, operating

systems, third-party libraries, and the target hardware. It may also have been modified or falsified, intentionally or otherwise.

a) Do you agree that evidence generated by software, as set out above, should be in scope, and that evidence which is merely captured / recorded by a device should be out of scope? Please provide a rationale for your answer.

This appears to be a very artificial distinction, with which we do not agree. All evidence that has been generated by a computer system or software could potentially be in scope. It will have to be decided on a case-by-case basis what can be excluded without materially changing the overall evidence provision.

For complex systems, justification evidence has proved to be difficult and costly and should be appropriate to the risk to the users of the system. Consequently, most of these systems operate with a residual risk of an unwanted/undesirable outcome and acceptability needs to be justified.

i) Can you provide specific examples of the type of evidence you believe should be in scope?

All outputs from the system including evidence of the processes applied to produce the software should be in scope as evidence. This could include displays on a screen, updates to a database, messages exchanged via networks and audit reports. It is also necessary to retain all associated data so that the environment of the system can be analysed.

Human interaction with the system is a major source of computer failure due to a mixture of deficient design, user error and malicious intent.

Furthermore, the issues raised by this discussion of ‘computer evidence’ apply to evidence generated by any system that is subject to ‘systematic errors’, not just software programs. ‘Systematic errors’ are consistent mistakes in outcomes caused by faulty equipment, incorrect methods, biased observations, etc.

ii) Can you provide specific examples of the type of evidence you believe should be out of scope?

It is very difficult to exclude any evidence as it could all have a material impact on the outputs. It might be possible to create a ‘priority list’ for a

specific application, but even the smallest bit of evidence could be important, e.g. a single message exchanged or a setting of a configuration parameter.

5) Are there any other factors which you believe are important for us to consider?

It should be stressed that the main reason that the Post Office Horizon system caused harm to subpostmasters was not just that the software was defective, but the way the organisation managed that IT system and its defects, including how defects were handled and the way in which subpostmasters were held solely accountable for consequent losses. Many other large IT systems have similar problems to Horizon when first deployed, but because the problems were handled differently no (or little) harm followed.

Management and Governance of the system needs to be appropriate to the risks posed by the system. Whilst there are good examples of contracting out the management and maintenance of IT systems, the responsibility for its governance must be retained by the organisation owning and/or using the system. That requires an appropriate level of understanding of how the system is supposed to function.

6) Conclusions

It is clear that the current common law presumption that computers producing evidence were operating correctly at the material time is unsafe. There is an ever-growing body of evidence to show that computer systems can and do fail and produce unexpected outcomes, including harm to people and property.

The mechanism for rebutting computer evidence needs to be reviewed and more clearly defined. To achieve this, thresholds need to be defined.

Modern computer systems are often complex, may be distributed across national boundaries and their design and operation may be hidden from scrutiny by vendors seeking to protect their products. All of these factors present difficulties when assessing computer systems and when developing rebuttals.

However, there are established methods and techniques for assuring software and computing systems, used in industries which already recognise the need to have reliable systems, with an associated cost. Nevertheless, the SCSC has drawn upon its experience to suggest means by which the current situation could be considerably improved. At the very least, there should be a justification produced for a system which explains why the evidence can be relied upon.

The adoption of a grading system based upon one or more desired attributes could provide a proportionate approach to providing dependable computer systems. It would require industry to adopt good practice and be backed up by a means of oversight. This could be achieved through the existing professional bodies (who already have a role in ensuring suitable qualifications and competency are in place), but there are alternatives. The key is to obtain as wide a consensus as possible which is where the government can provide direction, influence and if necessary, legislate.

In conclusion, the presumption of fault free operation is not sound and the burden should be on the system or software provider to demonstrate trustworthiness of computer evidence.

Appendix A SCSC Post Office Horizon Position

1. The Post Office Horizon system would not normally be regarded as a safety system, yet it is a computer-based system that has indirectly led to widespread harm.
2. The ongoing public enquiry has raised important legal, ethical and technical concerns. Problems highlighted include: sub-postmasters could not see what was going on (for example, figures changed remotely without sub-postmaster knowledge), poor quality coding and lack of both audit and fault logging.
3. The SCSC fully supports the public enquiry and other investigations, and we agree with professional computer bodies (e.g. the British Computer Society) that there should be a review of how computer-based system evidence is treated by the courts.
4. We will look to adopt relevant recommendations and encourage our members to do the same once the enquiry and investigations conclude.
5. We propose that organisations relying on computer-based system evidence in court should, where challenged, be required to justify that the system, including aspects such as hardware, software, data and service delivery, is reliable. Furthermore the evidence should be shown to be trustworthy. The justification should also show appropriate confidence in use, including in the way that reported problems are managed. Courts should not accept evidence relating to the computer-based system without this justification.
6. Where there are disputes involving computer-based systems there must be fair treatment; i.e. where relevant, there must be access for both sides to technical experts who in turn must be given access to appropriate software and data.
7. Post Office Horizon is an example of how systems, organisations, agreements, people and processes came together within a delivered service to result in indirect but severe harm. We suggest that the SCSC Service Assurance Guidance could be useful in such situations to reduce risks.

8. We will extend the remit of the SCSC to cover any computer-based systems and services which could cause harm. This will include harm in the wider context of the system including all stakeholders, the environment and consequential harms, not just harm caused directly by the system or service itself.

Note: Here “computer-based system” includes aspects such as hardware, software, data and service delivery.