"Validation of ultra-high dependability…" – 20 years on

Bev Littlewood, Lorenzo Strigini

Centre for Software Reliability, City University, London EC1V 0HB

In 1990, we submitted a paper to the Communications of the Association for Computing Machinery, with the title "Validation of Ultra-High Dependability for Software-based Systems" [Littlewood, 1993]. The immediate trigger for the discussions that led to that paper were the requirements of failure probability of less than 10^{-9} per hour, or per cycle, for some safety-critical equipment in civil aircraft. We thought that the then-typical approach to this issue (codified in the DO-178B document) did not inspire confidence. We paraphrased (some people said caricatured) the position taken in DO-178B as "a very low failure probability is required but, since its achievement cannot be proven in practice, some other, insufficient method of certification will be adopted". We also predicted that both this kind of extreme requirements, and the inadequate justification of their satisfaction, would spread to many more systems and industrial sectors, as they have.

Back then, different people had different takes on the issue, but our concerns were widely shared. Two years later, for example, Ricky Butler and George Finelli, from NASA, submitted to the IEEE Transactions on Software Engineering a paper with the title "The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software" [Butler, 1993].

This anniversary of the SCSC falls about 20 years later, so it seems a good time to revisit briefly our article and see where the debate about these issues now stands.

Our paper's main points were:

- modern society depends on computers for a number of critical tasks in which failure can have very high costs
- thus, high levels of dependability (reliability, safety, etc.) are often required
- risk should be assessed quantitatively, so
    - these requirements must be stated in quantitative terms, and
    - a rigorous demonstration of their attainment is necessary
- for software-based systems used in the most critical roles, such demonstrations are not usually supplied
- most importantly, the requirements often lie near the limit of the current state of the art, and sometimes beyond, in terms
    - of the ability to satisfy them,
    - and also, and more often, of the ability to demonstrate that they are satisfied in the individual operational products.

This validation problem was the main theme of our paper. We discussed why such demonstrations could often not be provided before operation with the means available: reliability growth models, testing with stable reliability, structural dependability modelling exploiting redundancy and diversity, arguments based on good engineering practice. For each such form of argument in support of a dependability claim, we showed how it ran into limits as the requirements became more stringent. Combining disparate evidence from these different sources allowed stronger claims, but we concluded that these would fall short – often by several orders of magnitude – of what was needed in some real applications.

We said that "engineering practice must take into account [...] that no solution exists, at present, for the validation of ultra-high dependability in systems relying on complex software". That is, systems depending on such software could only be deployed with limited confidence in their safety requirements being satisfied; or not be deployed. Alternatively, less stringent requirements could be set for some systems, at least at the beginning of their operational life. In this case, the decision would be rightly cast in socio-political terms of acceptable risk, rather than depending on stretching the technical evidence beyond what it could prove.

Revisiting the paper now, we find this basic message is still valid, although technical progress has changed some details. There are still limits to the credible claims that can be made about any specific system before operational experience. And for some systems, the requirements are definitely beyond those limits. It is discouraging to find that in some applications, requirements are becoming even more onerous, without matching progress in the ability to validate systems against them: for example, the protection system of the proposed UK EPR requires a probability of failure on demand no worse than 10^{-9}, which is two orders of magnitude more stringent than the 10^{-7} *pfd* needed 20 years ago for the protection system of Sizewell B.

Of course, there *have* been changes over the years in the magnitude and the nature of the limits. For example, 20 years ago we gave examples of how a purely statistical approach, based on operationally realistic testing or real operation, required very long testing for it to contribute substantially to confidence, and the length of feasible testing determined the limits to the claims. Things have improved from that viewpoint: with much faster and cheaper computers it is feasible to simulate very extensive testing on emulators. However, sources of doubt different from the statistical power of the empirical test then become more important, e.g. whether the test harness and test oracle are completely trustworthy [Littlewood, 2007], and these limit the confidence that can be placed in claims.

There has been disappointingly little progress in some areas in the last 20 years. An important missed opportunity has been in documenting the results of these years of increasing use of software based systems and of methods for building and validating them. A common approach is still that of advising incrementally stringent "good practices" for building and validating software as a function of its criticality – see, for example, IEC 61508. This is a reasonable approach, in principle, to achieving good results. But having used good practice is not a guarantee that the resulting system will be ultra-reliable [1]. And in practice there is little hard evidence of the effectiveness of those

[1] It is astonishing – and a poor reflection on our technical community – that there is still no agreement in the community that depends on the IEC 61508 standard about what can be claimed about a system's achieved dependability from the fact of its having been built using the recommended practices appropriate to a particular SIL.

practices in improving the chances of success. The persistence of this situation is a special concern. For instance, formal methods and other means of static verification have improved - both in the tools available and the amount of collective experience in using them. And yet evidence of their effectiveness – how often, for instance, a property that has been "proved" to be true turns out to be false – is not collected.

There continues to be some controversy about the use of probabilistic measures of dependability. Some practitioners whom we respect are dead set against it: they think that it is infeasible for design faults, and thus demanding it from the purveyors of safety-critical systems is a waste of resources and a dangerous temptation for self-delusion. These experts tend to be dissatisfied with existing approaches and invoke the adoption of better practices for assurance, but without quantifying their results. At the same time, others have been citing arguments like ours to justify the status quo, by saying that since demonstrating the 10^{-9} claim probabilistically is infeasible, the DO-178B position on certification without such justification was correct.

We still believe that arguments about uncertainty are naturally stated in probabilistic terms (and that there is *inherent* uncertainty here that cannot be wished away). For instance, the differences between these two groups cannot be decided without an attempt to argue which sets of practices would give better assurance that a system that passes the advocated method for certification will exhibit a sufficiently low frequency of accidents.

Probabilistic reasoning is the natural way of debating such disagreements. In fact, we would now put much more emphasis on the notion of *confidence* in claims, and treat this probabilistically [Bloomfield, 2007]. It seems clear that a dependability claim – "this system has a *pfd* better than 10^{-x}" – is never known to be true with certainty. There will be doubts about assumptions made in the reasoning, about the validity of the evidence, and so on. Treating this "epistemic" uncertainty rigorously and formally seems necessary, and using probabilities brings the advantages of a unified treatment of the different sources of uncertainty. Such a probabilistic argument may then sometimes show that we have limited grounds for confidence in a system before deployment (e.g. confidence that *this* flight control system has a failure rate better than 10^{-9} per hour). This is a benefit, not a defect, of the probabilistic approach, if risk assessment practices are to be beneficial for the engineering profession and the public.

Explicit recognition of epistemic uncertainty has other implications. For instance, recommended practice focuses on avoiding, removing, and proving the absence of, bugs: it is not direct evidence about probability of software-caused failure, except insofar as such failures could be avoided altogether. It is evidence for *probability of perfection*, not for achievement of a specific non-zero bound on *pfd* or failure rate. Standards that link the practices with the latter implicitly mix issues of reliability bounds and of confidence in them. Acknowledging evidence of probability of perfection would bring definite advantages in various scenarios (long-lived systems [Bertolino, 1998]; "asymmetric" diverse systems [Littlewood, 2010]) and help to focus on collecting useful evidence. If we had to rewrite that paper now, greater emphasis on the role of confidence and epistemic uncertainty would probably be the main change.

Finally, we come to the question of "how long is a piece of string?". What *are* the limits to what can be assured? Many of the references to our earlier paper – in particular some by authors who are generally supportive of the position laid out there – imply that we suggested some hard numeric limits: figures of 10^{-4} or 10^{-5} *pfd* are often stated, for example. In fact we did not say anything like this. Our intention, instead, was to show how different kinds of argument and amounts of evidence would hit limits, and how these could be shifted. So, for example, in the case of statistical testing, we showed how much failure-free operation was needed to support a particular claim at a particular level of confidence, allowing the reader to judge whether it was feasible (i.e. they had sufficient funds) to do enough testing for a particular (claim, confidence) pair.

The limits to a feasible (claim, confidence) pair about a specific system depend on what the specific system is, what evidence can be collected about it, and the state of general knowledge about that category of systems and techniques applied. All these factors vary between systems, and shift as technology changes and experience accumulates. Claiming that the same limits apply to all systems would be absurd.[2] Acknowledging that limits exist should be a spur to engage with reasoning about specific evidence and its value, to privilege designs that support better evidence collection (e.g. having in mind both statistical testing and formal proof at the time of design), to favour collective effort in collecting general knowledge about methods and classes of systems, finding ways to counter market-driven incentives to secrecy, to identify routes for orderly transition to sounder practices of certification and licensing; not to retreat into compliance-based schemes in which little incentive exists for the learning that alone can deliver progress.

Acknowledgments

This work was performed in projects INDEED, "INterdisciplinary DEsign and Evaluation of Dependability" funded by the U.K. Engineering and Physical Sciences Research Council (grant EP/E001580/1,) and UnCoDe, "Uncertainty and confidence in safety arguments: effect on expert decision makers", funded by the Leverhulme Trust

References

[Bertolino, 1998] A. Bertolino and L. Strigini, "Assessing the risk due to software faults: estimates of failure rate vs evidence of perfection", *Software Testing, Verification and Reliability*, vol. 8, no. 3, 1998, pp. 155-166.

[Bloomfield, 2007] R. E. Bloomfield, B. Littlewood and D. Wright, "Confidence: its role in dependability cases for risk assessment", Proceedings International Conference on Dependable Systems and Networks, Edinburgh, pp. 338-346, 2007.

[Butler, 1993] R.W. Butler and G.B. Finelli, "The infeasibility of quantifying the reliability of life-critical real-time software", *IEEE Trans Software Engineering*, vol. 19, no. 1, 1993, pp. 3-12.

[Littlewood, 1993] B. Littlewood and L. Strigini, "Validation of Ultra-High Dependability for Software-based Systems", *Communications of the ACM*, vol. 36, no. 11, 1993, pp. 69-80.

[2] Software based systems with safety implications range nowadays from e.g. nuclear protection systems that can be in principle few lines of code with no operating system to the massive distributed, interactive, layered systems involved in air traffic control.

[Littlewood, 2007] B. Littlewood and D. Wright, "The Use of Multilegged Arguments to Increase Confidence in Safety Claims for Software-Based Systems: A Study Based on a BBN Analysis of an Idealized Example", *IEEE Transactions on Software Engineering*, vol. 33, no. 5, 2007, pp. 347-365. doi:10.1109/TSE.2007.1002

[Littlewood, 2010] SRI-CSL-09-02: B. Littlewood and J. Rushby. "Reasoning about the Reliability Of Diverse Two-Channel Systems In which One Channel is 'Possibly Perfect'", under final review for publication in *IEEE Transactions on Software Engineering*.