Software reliability: basic concepts and assessment methods

Bev Littlewood **Lorenzo Strigini**
Centre for Software Reliability
City University
Northampton Square,
London EC1V OHB, U.K.
+44 20 7477 8420 +44 20 7477 8245
b.littlewood@csr.city.ac.uk l.strigini@csr.city.ac.uk

Software reliability is important for many sectors of the software industry. Besides knowing how to achieve it, it is important to know the actual reliability achieved in a specific software product. Assessing the reliability of software-based systems is increasingly necessary: more users bet larger amount of money, the survival of companies and at times the lives and limbs of people on the service they expect from the software. Sound decision-making requires some understanding of the uncertainties thus incurred. Meanwhile, software complexity increases and progress in development tools enables more poorly-trained people to build software-based systems. The short-term economic incentive to use off-the-shelf software, even in sensitive applications, imposes new requirements to evaluate the risk thus assumed. The pressure on vendors to guarantee some level of quality of service will thus also increase, extending from bespoke software to off-the-shelf software and from mission-critical to productivity-critical software.

The only sound methods now available rely on direct evidence from observing software behaviour in realistic test situations.

A difficulty is that many methods are recommended for software reliability assessment, but this variety hides the common underlying principles and makes it difficult to decide which method to use.

Given that different methods often produce different estimates, practitioners may well conclude that software reliability prediction is about producing precise numbers with no definite meaning. It is significant that many companies accept, as a substitute for software reliability predictions, inappropriate measures like the number of defects detected during development.

This tutorial is designed for practitioners who need to deal with reliability claims (in procurement of software, shipping decisions, contractual matters), with choosing ways of estimating software reliability and with interpreting the resulting estimates.

It covers:

- the need for probabilistic measures of software dependability;
- the principles of realistic testing for reliability estimation, and methods for defining and reproducing input profiles, i.e., criteria for defining a test case and for choosing a sequence of test cases;
- the precise meanings of the various questions that can be asked of reliability prediction, and of the predictions produced by different methods. We clarify the two scenarios of reliability evaluation: stable reliability and reliability growth. We also explain the practical differences between the predictions produced by "classical" and "Bayesian" methods;
- how to decide among various reliability growth projections using the techniques developed at the Centre for Software Reliability. Many alternative models have been produced that purported to be able to compute reliability predictions from failure data, but none of these could be shown to be universally accurate, so that it could be chosen with confidence by practitioners. The selection techniques allow a prediction method to be chosen, during a specific software project, by a procedure based on the method's performance to date during the same project;
- the limits in the levels of reliability that can confidently be predicted, how these are affected by the specific information available, and how they can be somewhat improved by more sophisticated approaches.

The tutorial is meant for an audience with a software engineering background, with a basic understanding of probability and statistics.

To appear in Proc. ICSE 2000, the 22nd International Conference on Software Engineering