Why should they believe us?
Determinism, non-determinism and evidence

David Budgen

In Software Engineering, as in Computing Science, the topics that we teach to our students can be considered as falling into two broad categories: the deterministic, and the non-deterministic. Deterministic topics are those where a specific scenario or operation leads to outcomes that can be assessed in terms of true/false values, and so this classification encompasses large elements of computer architecture, databases, metrics and testing. However, much of the Software Engineering body of knowledge is really concerned with much more non-deterministic processes such as requirements elicitation, design, construction, maintenance etc. These are activities in which humans play a central role, making value judgements that result in outcomes that are more appropriately assessed by using some form of better/worse ranking than through a true/false categorisation.

How much we recognise the existence of this distinction in our teaching is a moot point. Many of our students, educated in the classical science paradigm, will be familiar with the type of reasoning that leads to the outcomes for the deterministic elements. In my presentation I will examine some of the reasons why this experience may not be adequate when they encounter the non-deterministic elements of our subject, and hence why we may need to inculcate some degree of understanding of the evidence-based paradigm in order to support both our teaching and also their learning. I will discuss the nature of this paradigm, present some experiences of how it may be adapted for use in Software Engineering, and review some of the questions that it raises.

David Budgen is a Professor of Software Engineering at Durham University, UK. He received BSc and PhD degrees in theoretical physics from the Durham University in 1969 and 1973. After a period working on naval command and control systems, during which he developed an interest in software design issues, he was appointed to a post at the University of Stirling, moving to be Professor of Software Engineering at Keele University in 1991 and then moving to his current post at Durham in 2005. His current research interests include software design, component based and service based technologies, software development environments, and empirical software engineering, with a particular emphasis upon the accumulation and formalisation of evidence.