Book Review

Package evaluation

Richard Sharland, Avebury Technical, Aldershot, 1991.
ISBN 1856288196. 232 p. £32.50

One of the effects of IT becoming a tool of very many organizations is the proliferation of software packages on the market. Often these packages are not aimed at the traditional computer specialists, but at others whose assessment of packages might be based on little knowledge either of the technology or the functionality.

Richard Sharland's book helps to guide amateurs through the process of selecting an appropriate package for a task. Almost half of the book is taken up with a detailed set of forms and templates, which can be used to provide a structure in evaluating packages. The rest takes the reader through the evaluation process, mostly by referring to a fictitious example of a sales ledger, produced by 'Honest John's Software'. If the example looks a little trivial (its research and development was funded by William Hill and two of its rivals fell at the first hurdle), it is sufficient to illustrate the point.

The first step in evaluating a software package (beyond establishing its basic feasibility) is usually some sort of analysis of strengths and weaknesses. So in the same vein, it seems appropriate to look at the strengths and weaknesses of Sharland's book.

First the strengths. The book offers a rigorous framework for carrying out what has, too often before, been a purely heuristic process based on gut feeling and responses to sales material. The techniques suggested have a solid grounding; they are widely used, proven, and have the approval of procurement bodies, notably the CCTA. There are some particularly interesting hints, notably on how best to handle salesmen from prospective suppliers, and on possible sources of background information.

Now the weaknesses. While the example, of a sales ledger, is accessible and illustrates Sharland's point well,

it remains a package to perform a recognized, clearly defined, task. There is no discussion of the case where acquisition of a package has further ramifications for an organization, where the functionality offers new possibilities beyond the automation of what happens at present. So much of the book is devoted to examples of forms and structures, that the narrative seems to slightly lack substance, especially as the presentation is quite spread out.

Who, then, is the book likely to help? Like most models in the field of IT estimating, Sharland's framework does not remove the guesswork – it simply provides a structure within which it might be easier to identify what needs to be guessed. In areas like usability and future stability of suppliers, intuition, judgement, and (for usability) prototyping will always count for a lot, though Sharland offers some useful guidance to those choosing between a package with the best functionality, and one whose supplier is most likely still to be in business a few years hence.

Those who need a source of reference, a structure, checklists of factors to use during their evaluation will find this book worthwhile. If much of what it covers can be categorized as applied common sense, then it is nevertheless valuable for it to be documented in one place. If the book looks expensive, its price is still trivial compared with the possible costs of a bad decision about packaged software. But those looking for major new insights, or for new bridges between software evaluation and financial management, will be disappointed. There are no radical ideas, and Sharland's framework introduces no fresh perspectives on the use of packages. Overall, it is a sound work, but not one that offers any real advance in knowledge.

Martin Rich
City University Business School