Systems Analysis and Design: Best Practices

By James Wetherbe and Nicholas Vitalari

REVIEWED BY

MOHAN TANNIRV
SYRACUSE UNIVERSITY

Ever since the introduction of computer technology for solving business problems, we always marvelled at the rapidity with which the hardware technology was getting smaller in size, powerful in its capabilities, and cheaper for acquisition. At the same time, we always struggled, often with very little success, to keep the software development in sync with both the user needs and hardware development. To bridge this gap, we seem to continuously come up with new techniques or practices, each working well for some organizations, but becoming ineffective for others. While we hope for a magic solution to solve the software development problems some day, the reality is that hardware development has a clear goal (make computers smaller, faster and cheaper), while software development has a moving target (solve business problems that are changing continuously and frequently to keep pace with the dynamics of the business climate). The best one can hope for is that we keep developing newer techniques to address newer challenges, while at the same time not abandoning outright older techniques that have served us well over time.

Systems analysis and design: best practices, a book by Wetherbe and Vitalari discusses many of the techniques that have been time tested and still useful in the systems development process, while at the same time, introduces the reader to many of the newer techniques that have been in the popular press lately. However, instead of discussing them simply as individual techniques, the authors put them in the context of "systems analysis and design" — still the basic two steps in the systems development process. This is useful as any technique, ultimately, can only be judged to have a value if it is effective either in the analysis of the business requirements or in the design of appropriate system solution. Also, the discussion of IS planning up front is useful as newer applications are becoming organization-wide in nature (e.g. BPR, TQM, etc.) and future effort of MIS units within firms will be expended more in the planning and analysis phases to extract requirements, and less in the design phase (given the amount of automatic code generation and maintenance outsourcing that is taking place nowadays). The authors also provide some conceptual foundations at the start to ensure that the system development process in not viewed in isolation of the organizational and technical infrastructure.

I have a few reservations, however. The book, due to the focus on the breadth of coverage of various techniques, fails to go into sufficient depth on some of these techniques. This can be a problem, especially if it is used in an undergraduate curriculum, unless the instructor/trainer can supplement the book with short cases or problems. Also, the end of each chapter has a lot of references, but no short cases or problems to work on. I find this often frustrating as many of the analysis and design concepts can only be learned by doing rather than by reading. The authors refer to a case supplement to go along with the book and I hope it can address these needs. Otherwise, educators and trainers may need to supplement the book with their own cases or problems to make the book effective.

Overall, the book provides a useful survey of many "state of the art" practices (I would refrain from the word "best" as such an assessment is often a function of the level of maturity of the organization adopting these practices) that are used in the analysis and design of systems, and makes a solid contribution, especially to graduate curriculum and/or to the training practioners.