

## **The Art of Systems Architecting, Second Edition by Mark W. Maier and Eberhardt Rechtin**

*A Book Review by Ian Jay*

Projects come in different forms but this text provides a means of classifying them into five groups each of which has its own key characteristics. Defining which group a project fits depends on the nature of the project product and its environment; these two dimensions determine the appropriate project approach to the solution design. The five groups are; builder designed, manufacturing, social, software and IT, and collaborative.

The success or failure of many projects is due to the architectural approach. This book aims to provide a picture of what approaches best suit specific situations. Failed projects have often adopted inappropriate approaches to a problem. In the text, examples show how clear purpose and understanding gives rise to successful project outcomes. This clarity of purpose arises from the process of solution design.

To address the risk of project failure, develop an architectural vision of the project products. This vision takes the form of a set of heuristics, these are guidelines, statements or lessons learned that help steer the project. The book contains many of these heuristics and presents them in the applicable project context.

There are four approaches to developing solution architectures. They are Normative (solution based), Rational (method based), Participative (stakeholder based) and Heuristic (Lessons learned). The book deals with all of them however, its focus is on heuristics.

To understand what a heuristic is, contrast it with the engineering approach. Engineering relies on precise tools, calculus, and scientific methods. Heuristics are applicable where formal methods will not work. A heuristic is a pragmatic statement of lessons learned, taken to a level of abstraction and used as a guideline. An example of a heuristic is `if you can't analyze it, don't build it`. Another example is `build in and maintain options`. The latter advice, used in car and software design, provides for later add on features as the product matures.

Part two of the book explains the five domains. The first is the builder-architected project, such as aircraft, cars and computers. The form of these systems arises from the designer's idea of what the market requires the use made of them lies with the user. The second form are manufacturing systems, examples include lean production and ultra quality systems. These require that the system match the product. The third group are social systems that require extensive client (not user) involvement. An example of the third type of system is a civil project such as roads and bridges. The fourth domain is that of software systems. Finally, the fifth group are collaborative systems of which the Internet is an example. Discussion of each domain clarifies the considerations that affect the project choices. These choices affect the project design and strategy, which in turn determine the level of risk.

In its third part, the book becomes more concerned with the technicalities of the systems architects task. It describes a variety of modelling tools and approaches. The authors begin by explaining the need for six different views of the system. These range from purpose and objective statements (what the client wants) through to managerial views (the process by which the system is constructed and managed). Of these views, the latter relates directly to project control and reporting, whilst the other five are important elements determining project scope.

Regarding design itself, the book explains that in many cases key choices were never obvious decisions at the time. Something captured in the heuristic `before the flight it was opinion, after the flight it was obvious`. The aim of the design process is to move from an ill structured and chaotic view that can only be managed using heuristics, to a structured engineering view. The architecture process thus proceeds from abstract and general models to more domain specific views.

The fourth and final part of the book consists of two essays. The first essay discusses the political aspects of architecture design and the second to applicable standards. Architectural standards are important because they provide a common framework for technical communication across disciplinary interfaces. These standards are relevant across the entire project life cycle, one of the heuristics explored here is `to be tested, a system must be designed to be tested`.

The book ends with an appendix of heuristics grouped by area of use, such as multi task, modelling, and prioritising. An example of a prioritising heuristic for resource-limited situations is `the true value of a given service is determined by what one is prepared to give up obtaining it`. There are many more like this.

This text has a strong systems engineering flavour, it is useful for understanding how to deal with situations where different technologies meet. Its target audience is the systems architect or engineer but much of what it has to say is of direct relevance to the project manager.

Mark Maier, Ph.D is a senior engineering specialist at The Aerospace Corporation. His speciality is systems architecture in which he teaches and consults. He is responsible for pioneering work in the fields of socio-technical systems, collaborative systems, modelling, and application of distributed systems of systems.

Eberhardt Rechtin Ph.D. is a retired professor of the University of Southern California where he specialised in systems architecture in the fields of Industrial and Systems Engineering as well as Aerospace Engineering.

**`The Art of Systems Architecting second edition. Mark W Maier and Eberhardt Rechtin; 2002 CRC Press. ISBN 0-8493-0440-7**

Published in the July / August 2005 Edition of ProjectNet Magazine.

*©This Book Review cannot be reproduced in any form without the written permission of the relevant authority at the original publisher, ProjectNet Magazine. Contact the Editor at [info@projectnet.co.za](mailto:info@projectnet.co.za).*