



Advice for investigators

RICHARD MATEOSIAN

xrm@pacbell.net

Advice for a Young Investigator,

Santiago Ramón y Cajal, translated by Neely Swanson and Larry W. Swanson (MIT Press, 2004, 172 pp., US\$19.95, ISBN 0-262-68150-1, <http://mitpress.mit.edu>)

..... Santiago Ramón y Cajal (1852-1934) shared the Nobel Prize in Physiology or Medicine in 1906, just over a century ago. He established the structure of the brain and nervous system as networks of neurons connected by synapses.

In 1897, on the occasion of his induction into a Spanish scientific academy, Ramón y Cajal gave a speech in which he advised young investigators to focus their work in ways that he felt would help them avoid many of the mistakes he had made as he began his own scientific career 25 years earlier. The speech eventually evolved into this book, with editions in 1898, 1912, 1916, and 1923.

Ramón y Cajal lived in a time when Spain was not in the mainstream of scientific work. Much of his advice concerns overcoming this obstacle—learning languages, studying abroad, and so forth. His advice in this area is fascinating to look back on in the light of the shifts of power and influence that occurred in the 20th century. He pronounced German the most important language for scientists, followed by English and French. He considered Italian to be important too, but so much like Spanish that any Spaniard can read it without difficulty.

The most thoroughly obsolete part of this book concerns how a young investigator should choose and relate to his wife. One of his milder precepts in this area is as follows:

As a general rule, we advise the man inclined toward science to seek in the one whom his heart has chosen a compatible psychological profile rather than beauty and wealth. In other words, he should seek feelings, tastes, and tendencies that are to a certain extent complementary to his own. He will not simply choose a woman, but a woman who belongs to him, whose best dowry will be a sensitive compliance with his wishes and a warm and full-hearted acceptance of her husband's view of life.

Some of Ramón y Cajal's advice is as apt today as it was a century ago. Some of the examples he gives are tied to old technologies, but the principles are still correct. For example, he warns young investigators against undue admiration of authority. Ultimately, they must find their own way and dethrone established wisdom, just as Galileo refuted Aristotle's view of gravity and Copernicus displaced Ptolemy's view of the universe.

Ramón y Cajal also warns against the fallacy that everything important has been discovered. The 20th century is

filled with counterexamples to that nonsense.

In his recitation of desirable intellectual qualities, Ramón y Cajal concurs with the 21st century view in some areas, but perhaps not entirely in others. For example, he advocates independent judgment and concentration. Few today would argue against those qualities. On the other hand, he lists passion for reputation and patriotism among his desirable intellectual qualities. Here his arguments are not as persuasive.

Ramón y Cajal adheres to the "hard work" theory. He feels that most young investigators can do good work, but that lack of self-confidence leads many to fail.

In *Devices of the Soul* (Micro Review, July/August 2007), Steve Talbott says that self-forgetfulness is the reigning temptation of the technological era. He complains that we do not observe aspects of the world around us that fail to fit into our preconceived models. Ramón y Cajal would agree. He advocates specialization in the sense of mastery. His ideal investigator has the same relationship to his craft as the Waorani hunter has to his prey.

This is a fascinating book. It gives valuable insights into the changes of the last hundred years, and it contains nuggets of advice that are as valuable today as when Ramón y Cajal wrote them. I commend MIT Press for making it available, and I recommend it to any student of science.

Agile Retrospectives: Making Good Teams Great, Esther Derby and Diana Larsen (Pragmatic Bookshelf, 2006, 192 pp., US\$29.95, ISBN 0-9776166-4-9, <http://www.pragprog.com/titles/dlret>)

In the May/June 2001 Micro Review, I reviewed *Project Retrospectives* by Norm Kerth. That book proposes a dramatic improvement over the perfunctory “postmortems” that software teams sometimes engage in when they have finished a project. Derby and Larsen have adapted Kerth’s retrospectives to work with iterative and incremental development methods. Their retrospectives cover a short stretch of development—perhaps a few weeks—and drive changes to goals and processes for the next short stretch.

The project retrospectives that Kerth describes can entail a week spent at a resort. Derby and Larsen describe a process that proceeds on a much tighter schedule—perhaps an hour or so. Short meetings, like short letters, are hard to achieve. They require focus and discipline. This book defines a structure and time line for such meetings. By concrete examples, it shows why each element of the structure is important and how to meet the goals of that element within the allotted time.

As regular readers of this column know, Esther Derby is one of the people who work with Gerald Weinberg to produce the annual Amplify Your Effectiveness conferences. These conferences emphasize the human side of software development, and this book does the same. It shows how to identify and solve many team interaction problems that occur again and again in development projects.

If you engage in any sort of agile development, you should read this extremely concrete and practical handbook.

Oracle Essentials, fourth edition, Rick Greenwald, Robert Stackowiak

and Jonathan Stern (O’Reilly, 2008, 406 pp., US\$39.99, ISBN 0-596-51454-9, <http://www.oreilly.com>)

Jonathan Stern, the author most responsible for the original edition’s structure and clarity, died in March 2007. The remaining authors dedicate this edition to him.

A few years ago, I reviewed the second edition of this book (Micro Review, July/August 2001). As I noted then, I love books like this. I wish there were more of them. I spent over a year working with the Oracle server technology publications group and wrote some of the documentation that the earlier edition of this book is based on. The few hours that I spent reading this book gave me a much better sense of all the pieces and how they fit together than I ever achieved reading or writing Oracle documentation.

This edition does not differ greatly from earlier editions. It covers the new features of Oracle 11g: caching of query results, automatic memory management, real application testing (analyzing data from production runs), total recall (a data archive that allows querying “as of” a specified time), active data guard (simultaneous query and update of a standby database), and many other changes and additions. The authors integrated the new features into the book at the appropriate places, making the book look as if it were written from scratch for this edition.

The economics of the computer book business seem to favor books that cover a wide variety of features in a cursory way. Many books provide excellent task-oriented instructions for end users or even administrators, but give them little insight into the underlying structure and concepts.

The authors of *Oracle Essentials* follow a different path. They present a concise, coherent picture of the entire Oracle system. This picture does not

cover every feature of Oracle, nor does it cover any feature in complete depth. The picture is broad enough and deep enough to give you a good understanding of the main structures, processes, and issues involved in planning for and deploying Oracle, developing and optimizing schemas and applications for it, and administering its use.

One of the hardest tasks in working with Oracle is understanding the kind of big picture that this book presents. Thousands of highly skilled software developers have worked on the system over a period of more than 20 years. Inevitably, layer upon layer of enhancements have distorted and obscured the clarity of the original design. Furthermore, the system is so large and complex that people who start to work with it usually join a group that specializes in one aspect of it. Even the old-timers in the group may not know much about the rest of the system. And because they are comfortable in their corner of the product, they may not even consider it important to orient newcomers to the whole system.

The difficulty with this situation surfaces when different groups of specialists must work together to solve a common problem. Activities ranging from developing new versions of the product to designing new applications suffer from the inability of groups of specialists to understand one another’s problems and issues. For that reason, I think this book will do as much good within the walls of Oracle Corporation as it will outside.

If you work with any aspect of Oracle, or if you’d just like to understand the ins and outs of an important complex technology, this book is a must.

For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.