

## Art, Science, and Engineering

Steve McConnell

**E**ngineering's use of mathematics and science exposes it to the criticism that it is dry—that it saps the artistic elements out of structures that are engineered. The same criticism has been applied to software engineering. How true is this criticism? Does software engineering exclude aesthetics? Or are programmers' debates about the most beautiful way to format source code really just the tip of a deep software-aesthetics iceberg?



Far from being antithetical to aesthetics, traditional engineering is largely concerned with all aspects of design, including aesthetic aspects. Its designs aren't just limited to shapes and colors. Engineers design everything from electronic circuits to load-bearing beams to vehicles that land on the moon. As Samuel C. Florman says in the *Existential Pleasures of Engineering* (St. Martin's Griffin, 1994), "Creative design is the central mission of the professional engineer."

Consider a comparison of two well-known buildings, the Reims Cathedral in France and the Sydney Opera House in Australia. The Reims Cathedral, shown in Figure 1, was completed about 1290; the Sydney Opera House in 1973. The Reims Cathedral was designed to use materials whose properties were understood (more or less) at the time.

The Sydney Opera House was constructed 700 years after the Reims Cathedral. As you can see in Figure 2, it's stylistically quite different from the Reims

Cathedral. Its architects used modern materials such as steel and reinforced concrete, and they employed engineering techniques including computer modeling to determine how little material could safely be used.

Which building you prefer is a matter of taste, but which building can actually be built is a matter of engineering. It would be possible for modern builders to construct another Reims cathedral, but it would not have been possible for 13th-century builders to construct a Sydney Opera House. The reason is not the 13th century's lack of art but of engineering. Engineering without art can be ugly, but art without engineering can be impossible. Engineering does not constrain artistic possibilities. The lack of engineering constrains artistic possibilities.

So it is with modern software systems. The level of engineering prowess determines how large a system can be built successfully, how easy it will be to use, how fast it will operate, how many errors it will contain, and how well it will cooperate with other systems. Software includes many aesthetic elements, and software developers have no lack of artistic ambition. What we in the software industry sometimes lack is the engineering technique that enables us to realize some of our grandest aesthetic aspirations.

### Maturation of engineering disciplines

Mary Shaw at Carnegie Mellon University has identified a progression that fields travel through on their way to professional engineering ("Prospects for an Engineering Discipline of Software," *IEEE Software*, Nov. 1990).

## DEPARTMENT EDITORS

**Bookshelf:** Warren Keuffel,  
wkeuffel@computer.org

**Country Report:** Deependra Moitra,  
d.moitra@computer.org

**Design:** Martin Fowler, ThoughtWorks,  
fowler@acm.org

**Loyal Opposition:** Robert Glass, Computing Trends,  
rglass@indiana.edu

**Manager:** Don Reifer, Reifer Consultants,  
dreifer@sprintmail.com

**Quality Time:** Jeffrey Voas, Cigital,  
voas@cigital.com

## STAFF

Group Managing Editor  
**Dick Price**

Senior Lead Editor  
**Dale C. Strok**  
dstrok@computer.org

Associate Lead Editors  
**Crystal Chweh, Jenny Ferrero, and  
Dennis Taylor**

Staff Lead Editor  
**Shani Murray**

Staff Editor  
**Scott Anderson**

Magazine Assistants  
**Dawn Craig and Angela Williams**  
software@computer.org

Art Director  
**Toni Van Buskirk**

Cover Illustration  
**Dirk Hagner**

Technical Illustrator  
**Alex Torres**

Production Artists  
**Carmen Flores-Garvey and Larry Bauer**

Acting Executive Director  
**Anne Marie Kelly**

Publisher  
**Angela Burgess**

Membership/Circulation  
Marketing Manager  
**Georgann Carter**

Advertising Assistant  
**Debbie Sims**

## CONTRIBUTING EDITORS

Greg Goth, Kirk Kroeker, Nancy Mead,  
Ware Myers, Paula Powers, Judy Shane,  
Gil Shif

**Editorial:** All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

**To Submit:** Send 2 electronic versions (1 word-processed and 1 postscript or PDF) of articles to Magazine Assistant, *IEEE Software*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; software@computer.org. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.



**Figure 1. The Reims Cathedral.**

The Reims Cathedral was built when civil engineering was in its “commercial” stage. In software, many commercial-stage organizations achieve respectable levels of quality and productivity by using carefully selected, well-trained personnel. They rely on familiar practices and change them incrementally in pursuit of better products and better project performance.

Some of the problems that commercial production encounters can't be solved by trial and error, and, if the economic stakes are high enough, a

corresponding science will develop. As the science matures, it develops theories that contribute to commercial practice, and this is the point at which true professional engineering practice emerges. At that point, progress arises from application of scientific principles as well as from practical experimentation. The practitioners working in the field at that point must be well educated in both the theory and practice of their profession.

### **A science for software development**

Software science has been lagging behind commercial software development for years. Extremely large software systems were developed in the 1950s and 1960s, including the Sage missile defense system, the Sabre airline reservation system, and IBM's OS/360 operating system. Commercial development of these large systems proceeded much faster than supporting research did.

But are we really asking software science to provide the right things? For many classes of applications, including inventory management systems, payroll programs, general-ledger software, operating system design, database management software, language compilers—the list is nearly endless—the same basic applications have been written so many times that these systems shouldn't require as much unique design effort as they usually get. Shaw



**Figure 2. The Sydney Opera House.**

points out that in mature engineering fields routine design involves solving familiar problems and reusing large portions of prior solutions. Often these “solutions” are codified in the form of equations, analytical models, or prebuilt components.

Science has yet to provide software development with a set of equations that describe how to run a project successfully. Perhaps it never will. But science doesn't necessarily have to consist of formulas and mathematics. In *The Structure of Scientific Revolutions* (University of Chicago Press, 1970), Thomas Kuhn points out that a scientific paradigm can consist of a set of solved problems. Reusable software project artifacts are a set of solved problems—solved requirements problems, estimation problems, planning prob-

lems, design problems, and so on.

We know that software engineering is different from other engineering disciplines in some respects; perhaps the kind of science that underpins it is different as well. The one thing we know that is not different is that software engineers care about the aesthetics of their work. Historically, perhaps that has focused on source code, and perhaps in the future software engineers will care just as much about the aesthetics of their reusable estimates, plans, requirements, designs, and other “solved problems.”

### Acknowledgment

The images used here were obtained from IMSI's MasterPhotos Collection, 1895 Francisco Blvd. East, San Rafael, CA 94901-5506, USA.

# Software Engineering that affects our society.

We are the Software Engineering Institute (SEI), a federally funded research and development center sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. As an integral component of Carnegie Mellon University, we have access to leading-edge technology and provide career-minded professionals with a progressive work environment that is conducive to their personal achievement.

## A web site that can change your future.

If you would like to learn more about the work we do and the opportunities we offer, visit our web site:

[www.sei.cmu.edu](http://www.sei.cmu.edu)

Included below are the technical initiatives that have career opportunities currently available.

- CERT® Coordination Center (CERT/CC) for information security
- Capability Maturity Model® for Software (SW-CMM®) framework
- Capability Maturity Model Integration (CMMI)<sup>sm</sup>
- Personal Software Process<sup>sm</sup> (PSP<sup>sm</sup>) and Team Software Process<sup>sm</sup> (TSP<sup>sm</sup>) methods
- Application of product line practices to software development
- Integration of systems from commercial off-the-shelf (COTS) software

We offer a competitive salary, tuition reimbursement and benefits package along with the chance to work on projects that have major impact on the future of software technology. Please email your resume referencing job code: jobs-IEEE ehh@sei.cmu.edu or fax 412-268-5758.

Equal Opportunity Employer.  
We do not accept resumes from recruiting firms.



**Carnegie Mellon**  
**Software Engineering Institute**

#### EDITOR IN CHIEF:

**Steve McConnell**

10662 Los Vaqueros Circle  
Los Alamitos, CA 90720-1314  
software@construx.com

#### EDITOR IN CHIEF EMERITUS:

Alan M. Davis, Omni-Vista

#### ASSOCIATE EDITORS IN CHIEF

**Design:** Maarten Boasson, Quaerendo Invenietis  
boasson@science.uva.nl

**Construction:** Terry Bollinger, Mitre Corp.  
terry@mitre.org

**Requirements:** Christof Ebert, Alcatel Telecom  
christof.ebert@alcatel.be

**Management:** Ann Miller, University of Missouri, Rolla  
millera@ece.umar.edu

**Quality:** Jeffrey M. Voas, Cigital  
voas@cigital.com

#### EDITORIAL BOARD

Don Bagert, Texas Tech University  
Andy Bytheway, Univ. of the Western Cape  
Richard Fairley, Oregon Graduate Institute  
Martin Fowler, ThoughtWorks  
Robert Glass, Computing Trends  
Natalia Juristo, Universidad Politécnica de Madrid  
Warren Keuffel, independent consultant  
Brian Lawrence, Coyote Valley Software  
Karen Mackey, Cisco Systems  
Stephen Mellor, Project Technology  
Deependra Moitra, Lucent Technologies, India  
Don Reifer, Reifer Consultants  
Wolfgang Strigel, Software Productivity Centre  
Karl Wieggers, Process Impact

#### INDUSTRY ADVISORY BOARD

Robert Cochran, Catalyst Software, chair  
Annie Kuntzmann-Combelles, Q-Labs  
Enrique Draier, PSINet  
Eric Horvitz, Microsoft Research  
David Hsiao, Cisco Systems  
Takaya Ishida, Mitsubishi Electric Corp.  
Dehua Ju, ASTI Shanghai  
Donna Kasperson, Science Applications International  
Pavle Knaffic, Hermes SoftLab  
Günter Koch, Austrian Research Centers  
Wojtek Kozaczynski, Rational Software Corp.  
Tomoo Matsubara, Matsubara Consulting  
Masao Matsumoto, Univ. of Tsukuba  
Dorothy McKinney, Lockheed Martin Space Systems  
Susan Mickel, AtomicTangerine  
Dave Moore, Vulcan Northwest  
Melissa Murphy, Sandia National Laboratories  
Kiyoh Nakamura, Fujitsu  
Grant Rule, Software Measurement Services  
Girish Seshagiri, Advanced Information Services  
Chandra Shekaran, Microsoft  
Martyn Thomas, Praxis  
Rob Thomsett, The Thomsett Company  
John Vu, The Boeing Company  
Simon Wright, Integrated Chipware  
Tsuneo Yamaura, Hitachi Software Engineering

#### MAGAZINE OPERATIONS COMMITTEE

Sorel Reisman (chair), James H. Aylor, Jean Bacon,  
Thomas J. Bergin, Wushow Chou, William I.  
Grosky, Steve McConnell, Daniel E. O'Leary, Ken  
Sakamura, Munindar P. Singh, Francis Sullivan,  
James J. Thomas, Yervant Zorian

#### PUBLICATIONS BOARD

Rangachar Kasturi (chair), Angela Burgess (pub-  
lisher), Jake Aggarwal, Laxmi Bhuyan, Mark Chris-  
tensen, Lori Clarke, Mike T. Liu, Sorel Reisman,  
Gabiella Sannitti di Baja, Sallie Sheppard, Mike  
Williams, Zhiwei Xu