

## Raising Your Software Consciousness

Steve McConnell



In 1970, Charles Reich published a best-selling book called *The Greening of America*.<sup>1</sup> In it, Reich identifies three kinds of awareness or consciousness, which he calls Consciousness I, Consciousness II, and Consciousness III.

Consciousness I (“Con I”) is the pioneer mentality. People who operate at Con I place great value on independence and self-satisfaction. They don’t easily tolerate other people telling them what to do. They are highly self-reliant and self-sufficient. Reich believes that Con I dominated the American psyche during America’s first centuries and that this focus on self-reliance was a significant factor in

America’s development.

Consciousness II is the gray flannel suit mentality—corporation man. People who operate at Con II understand the importance of getting along with others and playing by the rules. They believe rules are good for society, and they think everyone should follow them. Reich believes that Con II became more dominant than Con I in the mid-twentieth century.

Consciousness III is the mentality of enlightened independence. The Con III person operates on the basis of principles, with little regard for the rules that predominate in Con II and without the selfishness that predominates in Con I. By the time *Greening* was published, Reich argued that Con II’s time was over. He believed Con III was in its ascendancy and would soon replace Con II.

Although *The Greening of America* struck a resonant chord when it was published, his-

tory has not been kind to the book. In 1999, *Slate* magazine’s readers voted it the silliest book of the 20th century. Reich’s Con III was a hippie nirvana, and the “greening” Reich predicted was a nationwide movement toward the hippie culture of the 1960s and 1970s—psychedelic drugs, bell-bottom pants, and all. As the hippie culture faded into obscurity in the 1980s, so did the credibility of Reich’s predictions.

### Can’t get no satisfaction

Reich’s political predictions may not have withstood the test of time, but his classification of Con I, Con II, and Con III provides a useful model for the software industry today.

Con I in software is associated with a focus on self-reliance. Software experts often refer to software developers operating at this level of awareness as mavericks, cowboy programmers, Lone Rangers, and prima donnas. Software developers at this level tend to have little tolerance for other people’s ideas. They like to work alone. They don’t like following standards. The “Not Invented Here” syndrome thrives.

Con I’s advantage is that little training is needed, and the lone-wolf approach works adequately in environments that employ only small numbers of programmers who work independently on small projects. Con I’s disadvantage is that it scales poorly to projects that need teams of programmers rather than isolated individuals.

Con II in software is associated with a focus on rules. Many software developers eventually discover the limitations of Con I’s self-reliant development style and see the

### DEPARTMENT EDITORS

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

**Country Report:** Deependra Moitra, Lucent Technologies  
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

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

Group Managing Editor  
**Crystal Chweh**

Associate Editors  
**Jenny Ferrero and  
Dennis Taylor**

Staff Editors  
**Shani Murray, Scott L. Andresen,  
and Kathy Clark-Fisher**

Magazine Assistants  
**Dawn Craig**  
software@computer.org

**Pauline Hosillos**

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**

Assistant Publisher  
**Dick Price**

Membership/Circulation  
Marketing Manager  
**Georgann Carter**

Advertising Assistant  
**Debbie Sims**

### CONTRIBUTING EDITORS

Greg Goth, Denise Hurst, Anne Lear, Keri Schreiner,  
and Margaret Weatherford

**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.

advantages of working in groups. Over time, they learn rules that allow them to coordinate their work with others. Some groups of developers create their own informal rules through trial and error, and these groups can be highly effective. Other groups buy a prebuilt methodology. Sometimes the rules are provided by consultants, as in the classic “17 three-ring binders” methodologies. Other times, the rules are taken from books, such as *The Rational Unified Process*,<sup>2</sup> the *Extreme Programming* series,<sup>3</sup> or my own *Software Project Survival Guide*.<sup>4</sup> Developers at this level of awareness tend to focus on the details of adhering to the rules. They argue about which interpretations of the rules are correct and focus on “following the methodology.”

The advantage of Con II is that a developer needs to be trained to use only a single approach. If a good approach is chosen, the developer can leverage a relatively small amount of training across many projects. The disadvantage is that a Con II developer is ill-equipped to succeed on projects that fall outside the specific methodology in which the developer was trained.

Con III in software is associated with a focus on principles. At this level of awareness, developers understand that the rules of any prepackaged methodology are, at best, approximations of principles. Those approximations might apply most of the time, but they won't apply all of the time. The disadvantage of Con III is that extensive education and training are needed to introduce a developer to the principles underlying effective software development, and that training is not easily obtained. Con III's advantage is that, once that training has been obtained, the developer is equipped with a full range of software engineering tools that support success on a wide range of projects.

### Love the one you're with

The software industry has a long history of trying and ultimately rejecting “one size fits all” methodologies. These methodologies are Con II soft-

ware approaches, and they fail outside narrowly defined areas of applicability—predictably—precisely because they are Con II. The world of software is far too varied to be addressed by a single set of rules.

For example, compare the practices you would use to develop a heart pacemaker control to those you would use to develop a video store management program. If a software malfunction caused you to lose one video out of 1,000, it might affect the store's profitability by a fraction of a percent, but the impact is negligible. If a malfunction caused one pacemaker out of 1,000 to fail, however, you've got a real problem. Generally speaking, widely distributed products must be developed more carefully than narrowly distributed ones. Products whose reliability is important must be developed more carefully than products whose reliability doesn't much matter.

These different kinds of software require different development practices. Practices that would be considered to be overly rigorous, bureaucratic, and cumbersome for video store management software might be considered irresponsibly quick and dirty—or reckless—for an embedded pacemaker control. The Con III developer will use different practices to develop a heart pacemaker control than to develop an video inventory tracking system. The Con II developer will try to apply a one-size-fits-all methodology to both projects, with the likelihood that the methodology won't work particularly well for either one.

### Are you experienced?

Reich identified the three levels of consciousness as the zeitgeists of different eras, but I see Con I, Con II, and Con III as three distinct steps along a path of personal software engineering maturity. Most software developers begin their careers at Con I and eventually journey to Con II. In many environments, Con II supports effective work, and no further development is needed. In some environments, however, a further progression toward Con III is needed.

The by-the-book methodologies of Con II seem to be a reasonable learning path for developers at Con I who are not yet well versed in a wide range of software practices. The specific details of the rules-based practices probably don't matter all that much. People who are trying to raise themselves from Con I to Con II simply need to take a first step away from the chaos of a completely unmanaged project. They must learn a set of rules and get some experience applying those rules before they can advance to the Con III level, where they understand software project dynamics well enough to break the rules when needed. This whole process is part of the natural progression from apprentice to journeyman to master. ☺

**References**

1. C. Reich, *The Greening of America*, Random House, New York, 1970.
2. P. Kruchten, *The Rational Unified Process: An Introduction*, 2nd ed., Addison-Wesley, Reading, Mass., 2000.
3. K. Beck, *Extreme Programming: Embrace Change*, Addison-Wesley, Reading, Mass., 2000.
4. S. McConnell, *Software Project Survival Guide*, Microsoft Press, Redmond, Wash., 1998.

**Upcoming Topics**

**January/February '02:**  
**Building Systems Securely  
 from the Ground Up**

**March/April '02:**  
**The Software Engineering  
 of Internet Software**

**May/June '02:**  
**Knowledge Management  
 in Software Engineering**

**July/August '02:**  
**The Business of  
 Software Engineering**

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@quaerendo.com

**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.umn.edu

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

**Experience Reports:** Wolfgang Strigel,  
 Software Productivity Center; strigel@spc.ca

**EDITORIAL BOARD**

- Don Bagert, Texas Tech University
- Richard Fairley, Oregon Graduate Institute
- Martin Fowler, ThoughtWorks
- Robert Glass, Computing Trends
- Andy Hunt, Pragmatic Programmers
- Warren Keuffel, independent consultant
- Brian Lawrence, Coyote Valley Software
- Karen Mackey, Cisco Systems
- Deependra Moitra, Lucent Technologies, India
- Don Reifer, Reifer Consultants
- Suzanne Robertson, Atlantinc Systems Guild
- Dave Thomas, Pragmatic Programmers
- Karl Wiegers, 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 Knaflic, 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
- Nancy Mead, Software Engineering Institute
- Stephen Mellor, Project Technology
- Susan Mickel, AgileTV
- 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, Ken Sakamura, Nigel  
 Shadbolt, 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,  
 Gabriella Sannitti di Baja, Sallie Sheppard, Mike  
 Williams, Zhiwei Xu