There has been a growing interest in applying appropriate software engineering practices to the development of computational science and engineering (CSE) software. Recent research shows two important trends: First, in most cases, the development of CSE software doesn’t follow traditional software engineering best practices. Second, in many cases, existing software engineering best practices aren’t applicable to CSE software without significantly tailoring and adapting them. These two trends are clearly related to each other and they must be addressed to improve both the productivity of CSE software development and the reliability of CSE software. Unfortunately, historically there have only been a small number of people with enough expertise and interest in both CSE software development and traditional...
software engineering to properly address the problem. Therefore, a great need exists to investigate this problem further by gathering experts from software engineering together with experts from CSE and develop solutions. This need motivated the creation of the Software Engineering for Computational Science and Engineering (SE-CSE) workshop series, from which the articles in this special issue are drawn.1

At the workshop, we discussed why previous attempts to apply software engineering practices from the business/IT domain to CSE software weren’t as successful as many had hoped. This lack of success often results from a mismatch between the CSE environment and the traditional business/IT environment. Some important differences between those environments can include the following:2–6

- CSE software often explores unknown science or engineering, so many requirements emerge during development and can’t be known or adequately planned for a priori.
- Often the scientist or engineer doesn’t know the expected results of the software’s execution, making it difficult to perform traditional tests using test oracles.
- The software-development life cycle is often either extremely long (tens of years in development and evolution) or short (code written quickly, executed once, and discarded); therefore, choosing an appropriate process to manage software development can be difficult.

By bringing researchers together in this workshop, we’re building a community to address these problems. As CSE software becomes an increasingly important factor in critical decisions of various types, this kind of interaction is vital for improving the quality and reliability of CSE software. During the workshop, researchers presented numerous papers that spurred further discussions on possible solutions. Here I provide a brief summary of those discussions, along with an introduction to the articles in this special issue.

Key Discussions
Most of the SE-CSE workshop’s discussion focused on better understanding some of the reasons why traditional software engineering practices aren’t commonly adopted by CSE developers. One issue raised was the fact that there needs to be better communication of software engineering successes among CSE teams. For example, there are cases where a CSE team might have (re)discovered a software engineering practice on their own, without ever formally learning it.

We need to be able to socialize these successes within the CSE community so that others can benefit from such findings. It’s important to identify principles that will apply regardless of the domain. One of the primary problems that CSE developers perceive with current software engineering curricula is that there’s too much content that doesn’t directly apply to the development of CSE software. Furthermore, the practices that are included in software engineering curricula often haven’t been appropriately tailored for the CSE environment. This situation results, in part, because we’re still not confident about which characteristics are important for differentiating traditional software engineering environments from CSE environments. As a result, CSE developers often don’t see the relevance of software engineering for their projects.

To address this problem, we discussed what it would take for CSE developers to adopt appropriate software engineering practices. The group identified a number of items that could help. The most important conclusion was that for any proposed solutions to have credibility, they must be developed and advocated by members of the CSE community rather than by external software engineering researchers. Second, in many cases, funding agencies don’t require adequate software engineering processes to be followed. Portions of the budget for CSE projects must be allocated for using appropriate software engineering practices, such as testing. Finally, it’s important to address this problem from the beginning. That is, junior CSE developers (including graduate and postdoctoral students) must be mentored by more senior CSE developers regarding appropriate software engineering practices. Another way of addressing the problem early is to offer software engineering for CSE tutorials and workshops for graduate students and junior developers at appropriate venues such as the Supercomputing (SC) conference.

Introducing the Articles
This special issue contains four articles that address various aspects of the software engineering life cycle, ranging from high-level process issues to more detailed language issues.

The first article, “Velo: A Knowledge-Management Framework for Modeling and Simulation,” by Ian Gorton and his coauthors, describes a knowledge-management system specifically for use by developers of simulation software. Velo helps simulation teams keep track of
the large amount of data and artifacts that result from computational simulations.

The second article, “What Do We Know about Scientific Software Development’s Agile Practices?” by Magnus Thorstein Sletholt and his coauthors, is a literature review that investigates the use of agile software-development methods for scientific software. Other researchers have posited that agile approaches can be a good fit for the development of scientific software. This article examines that claim in more detail.

The third article, “System Testing a Scientific Framework Using a Regression-Test Environment,” by Hanna Remmel and her coauthors, investigates a specific phase of the software-development life cycle, system testing. This article describes the need for system testing in a scientific environment and illustrates the concepts with an example.

Finally, the last article, “This Isn’t Your Parents’ Fortran: Managing C++ Objects with Modern Fortran” by Damian Rouson and his coauthors, focuses on the specific problem that occurs when interface Fortran code creates C++ objects. Allocation and deallocation of these objects isn’t straightforward. This article provides a solution for managing that complexity.

More details about the workshop, along with the full list of papers presented, are available at www.cs.ua.edu/~SECSE11. The website also provides links to previous years’ workshops. If you have thoughts on how to better integrate appropriate software engineering practices into CSE development or questions about the workshop series, or if you would like further information, I look forward to hearing from you at carver@cs.ua.edu.

Acknowledgments
I would like to thank the following reviewers who provided input during the paper review and selection process: Roscoe Bartlett, Oak Ridge National Laboratories (US); Diane Kelly, Royal Military College (Canada); Lorin Hochstein, ISI (US); and Judith Segal, The Open University (UK).

References

Jeffrey C. Carver is an assistant professor in the Department of Computer Science at the University of Alabama. His research interests include empirical software engineering, software engineering for computational science and engineering, qualitative methods for software engineering, and human factors in software engineering. Carver has a PhD in computer science from the University of Maryland. Contact him at carver@cs.ua.edu. He is a senior member of IEEE and ACM.

Selected articles and columns from IEEE Computer Society publications are also available for free at http://ComputingNow.computer.org.
PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine Computer, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEBSITE: www.computer.org

OMBUDSMAN: To check membership status or report a change of address, call the IEEE Member Services toll-free number, +1 800 678 4333 (US) or +1 732 981 0060 (international). Direct all other Computer Society-related questions—magazine delivery or unresolved complaints—to help@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION: To obtain more information on any of the following, contact Customer Service at +1 714 821 8380 or +1 800 272 6657:

• Membership applications
• Publications catalog
• Draft standards and order forms
• Technical committee list
• Technical committee application
• Chapter start-up procedures
• Student scholarship information
• Volunteer leaders/staff directory
• IEEE senior member grade application (requires 10 years of practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

Computer: The flagship publication of the IEEE Computer Society, Computer, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications. Periodicals: The society publishes 13 magazines, 18 transactions, and one letters. Refer to membership application or request information as noted above.


Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Committees: TCs provide professional interaction in more than 45 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers two software developer credentials. For more information, visit www.computer.org/certification.

NEXT BOARD MEETING
1–3 Feb., Long Beach, CA, USA

EXECUTIVE COMMITTEE

President: John W. Walz
President-Elect: David Alan Grier
Past President: Sorel Reisman
VP, Standards Activities: Charlene (Chuck) Walrad
Secretary: Andre Ivanov (2nd VP)
VP, Educational Activities: Elizabeth L. Burd
VP, Member & Geographic Activities: Sattupathuv Sankaran
VP, Publications: Tom M. Conte (1st VP)
VP, Professional Activities: Paul K. Joannou
VP, Technical & Conference Activities: Paul R. Croll
Treasurer: James W. Moore, CSDP
2011–2012 IEEE Division VIII Director: Susan K. (Kathy) Land, CSDP
2012–2013 IEEE Division V Director: James W. Moore, CSDP
2012 IEEE Division Director-VII Director-Elect: Roger U. Fujii

BOARD OF GOVERNORS

Term Expiring 2012: Elizabeth L. Burd, Thomas M. Conte, Frank E. Ferrante, Jean-Luc Gaudiot, Paul K. Joannou, Luis Kun, James W. Moore, William (Bill) Pitts
Term Expiring 2013: Pierre Bourque, Dennis J. Frailey, Atsushi Goto, André Ivanov, Dejan S. Milojevic, Paolo Montuschi, Jane Chu Prey, Charlene (Chuck) Walrad

EXECUTIVE STAFF

Executive Director: Angela R. Burgess
Associate Executive Director; Director, Governance: Anne Marie Kelly
Director, Finance & Accounting: John Miller
Director, Information Technology & Services: Ray Kahn
Director, Membership Development: Violet S. Doan
Director, Products & Services: Evan Butterfield
Director, Sales & Marketing: Chris Jensen

COMPUTER SOCIETY OFFICES

Phone: +1 202 371 0101 • Fax: +1 202 728 9614
Email: hq.ofc@computer.org
Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314
Phone: +1 714 821 8380
Email: help@computer.org

MEMBERSHIP & PUBLICATION ORDERS
Phone: +1 800 272 6657 • Fax: +1 714 821 4641 • Email: help@computer.org

IEEE OFFICERS

President: Moshe Kam
President-Elect: Gordon W. Day
Past President: Pedro A. Ray
Secretary: Roger D. Pollard
Treasurer: Harold L. Flescher
President, Standards Association Board of Governors: Steven M. Mills
VP, Educational Activities: Tariq S. Durrani
VP, Membership & Geographic Activities: Howard E. Michel
VP, Publication Services & Products: David A. Hodges
VP, Technical Activities: Donna L. Hudson
IEEE Division V Director: James W. Moore, CSDP
IEEE Division VIII Director: Susan K. (Kathy) Land, CSDP
President, IEEE-USA: Ronald G. Jensen

revised 1 Dec. 2011