SOFTWARE ENGINEERING FOR CSE

6 Guest Editors’ Introduction
Jeffrey C. Carver and Tom Epperly
Software Engineering for Computational Science and Engineering

10 Streamlining Development of a Multimillion-Line Computational Chemistry Code
Robin M. Betz and Ross C. Walker
Software engineering methodologies can be helpful in computational science and engineering projects. Here, a continuous integration software engineering strategy is applied to a multimillion-line molecular dynamics code; the implementation both streamlines the development and release process and unifies a team of widely distributed, academic developers.

18 Water Science Software Institute: Agile and Open Source Scientific Software Development
Stan Ahalt, Larry Band, Laura Christopherson, Ray Idaszak, Chris Lenhardt, Barbara Minsker, Margaret Palmer, Mary Shelley, Michael Tiemann, and Ann Zimmerman
An Open Community Engagement Process (OCEP) applies open source mechanics and software engineering to water science research. To operationalize OCEP, the authors conceptualize a Water Science Software Institute whose mission is to support and accelerate water science by transforming both the software and research cultures of the water science community.

28 Leveraging Expertise to Support Scientific Software Process Improvement Decisions
Erika S. Mesh, Gabbie Burns, and J. Scott Hawker
Leveraging the underlying expertise and motivational factors that drive scientific software development practices, a hybrid scientific software process improvement framework (SciSPIF) is being constructed to allow scientific software developers to make SE process improvement decisions tailored to their own goals. Here, the strategy for designing and building SciSPIF is reported, along with preliminary insights.

36 Building CLiiME via Test-Driven Development: A Case Study
Aziz Nanthaamornphong, Jeffrey C. Carver, Karla Morris, Hope A. Michelsen, and Damian W.I. Rouson
The Community Laser-Induced Incandescence Modeling Environment (CLiiME) project used the agile approaches of test-driven development and refactoring to implement a collaborative model infrastructure for its researchers to use, modify, and extend. As the project’s results show, using agile methods can offer several benefits for software development in the CSE community.

48 Systematic Debugging Methods for Large-Scale HPC Computational Frameworks
Alan Humphrey, Qingyu Meng, Martin Berzins, Diego Caminha B. de Oliveira, Zvonimir Rakamaric, and Ganesh Gopalakrishnan
Parallel computational frameworks for high-performance computing are central to the advancement of simulation-based studies in science and engineering. Finding and fixing bugs in these frameworks can be time consuming. If left unchecked, these bugs diminish the amount of new science performed. A systematic study of the Uintah Computational Framework investigates debugging approaches, leveraging the framework’s modular structure.

For more information on these and other computing topics, please visit the IEEE Computer Society Digital Library at www.computer.org/csdl.
58 A Case Study on a Quality Assurance Process for a Scientific Framework
Hanna Remmel, Barbara Paech, Christian Engwer, and Peter Bastian
This case study analyzes the feasibility and acceptance by developers of two parts of the quality assurance process: variability model creation and desk-checking. The study found that both parts are accepted and feasible for development, but a different variability modeling language is needed to represent all important aspects.

90 Computer Simulations
A Heterogeneous Fleet of Vehicles for Automated Humanitarian Missions
Pieter J. Mosterman, David Escobar Sanabria, Enes Bilgin, Kun Zhang, and Justyna Zander

ALSO IN THIS ISSUE
68 Solving Solid and Fluid Mechanics Problems in the Cloud with mOSAIC
Jernej Južna, Peter Češarek, Dana Petcu, and Vlado Stankovski
An application for analysis of structures under static loading is ported to the cloud using the mOSAIC portable platform-as-a-service. The new cloud application benefits from Web availability, elasticity, and fault tolerance, while being independent from the infrastructure-as-a-service providers. This achievement paves the way for porting a range of engineering applications to multiple clouds.

COLUMNS
96 The Last Word
My First E-Reader
Charles Day

DEPARTMENTS
4 Books
Modern Computer Algebra and a Lot More
Jürgen Schnack

78 Education
Computational Pedagogy: A Modeling and Simulation Approach
Osman Yaşar and Jose Maliekal