CARE: An Automobile Crash Data Analysis Tool  
pp. 22-30  
Allen S. Parrish, Brandon Dixon, David Cordes, Susan Vrbsky, and David Brown

The Critical Analysis Reporting Environment provides an efficient tool for transportation safety engineers and policymakers to use in analyzing the categorical crash data typically obtained from police reports. CARE has proven successful in the traffic safety community for two reasons: its simplicity and its efficiency. It is currently being used in several states.

Andhra Pradesh: Lessons for Global Software Development  
pp. 31-37  
Kyle Eischen

Although extremely poor economically, the Indian state of Andhra Pradesh has been a supplier of skilled software labor for more than a decade and is one of the most innovative sites in applying IT for social development. Many of AP’s e-government programs are just beginning to fully take root, and they should have a significant impact in a relatively short time.

Agility through Discipline: A Debate  
pp. 44-46  
Kent Beck and Barry Boehm

While Beck asserts that agility is only possible through greater discipline on the part of everyone involved, Boehm counters that you don’t broaden the definition of “discipline” by rejecting parts of it. Yet, from inside Extreme Programming, it seems that the only way to achieve the desired results is to view the world in “both-and” terms instead of “either-or” terms.

Iterative and Incremental Development: A Brief History  
pp. 47-56  
Craig Larman and Victor R. Basili

Although many view iterative and incremental development as a modern practice, its application dates as far back as the mid-1950s. Prominent software-engineering thought leaders from each succeeding decade supported IID practices, and many large projects used them successfully. These practices may have differed in their details, but all had a common theme—to avoid a single-pass sequential, document-driven, gated-step approach.

Using Risk to Balance Agile and Plan-Driven Methods  
pp. 57-66  
Barry Boehm and Richard Turner

Both agile and plan-driven approaches have situation-dependent shortcomings that, if not addressed, can lead to project failure. The challenge is to balance the two approaches to take advantage of their strengths in a given situation while compensating for their weaknesses.

The authors present a risk-based approach for structuring projects to incorporate both agile and plan-driven approaches in proportion to a project’s needs.

Developing Complex Projects Using XP with Extensions  
pp. 67-73  
Martin Lippert, Petra Becker-Pechau, Holger Breitling, Jörn Koch, Andreas Kornstädt, Stefan Roock, Axel Schmolitzky, Henning Wolf, and Heinz Züllighoven

Using methodological extensions to adapt XP for major projects offers high security and reliability without limiting agile software development’s advantages. The authors describe their use of XP extensions that focus on development’s planning and controlling aspects, demonstrating that a suitably adapted agile development process is applicable to long-term, large-system projects.

Introducing an Agile Process to an Organization  
pp. 74-78  
Mike Cohn and Doris Ford

The transition from a plan-driven to an agile process affects not only the development team members, but also other teams, departments, and management. Any new process will likely attract developers excited to try it while repelling those opposed to change. Thus, how an agile process is introduced into an organization significantly affects its ultimate success.

Migrating Agile Methods to Standardized Development Practice  
pp. 79-85  
Mark Lycett, Robert D. Macredie, Chaitali Patel, and Ray J. Paul

Situated process and quality frameworks offer a way to resolve the tensions that arise when introducing agile methods into standardized software development engineering. For these to be successful, however, organizations must grasp the opportunity to re-integrate software development management, theory, and practice.