**Improving Requirements Engineering Communication in Multiproject Environments**

by Barbara Paech, Jörg Dörr, and Mathias Koehler, pp. 40–47. In complex, multiproject environments, communication is the key to successful requirements engineering. An information model helps with this problem by capturing stakeholders’ documents and responsibilities during RE. The information model effectively and practically ensures that stakeholders of dependent projects are mutually aware of critical communication needs. The authors present the information model developed at Nokia Smart Traffic Products and show how they defined it in a two-day workshop.

**Sharing Requirements Engineering Experience Using Patterns**

by Lars Hagge and Kathrin Lappe, pp. 24–31. Patterns offer a general vehicle for sharing requirements engineering experience. Four patterns for basic RE activities enable knowledge transfer at the level of practical experience. An analysis method for deriving patterns from observations can also be used as a basis for building an online pattern repository to make patterns available to project leaders on the job.

**A Linguistic-Engineering Approach to Large-Scale Requirements Management**

by Johan Natt och Dag, Vincenzo Gervasi, Sjaak Brinkkemper, and Björn Regnell, pp. 32–39. Developing large, complex software products aimed at broad markets involves identifying and maintaining the link between product requirements and the continuous, massive inflow of customers’ wishes. The manual linkage performed today is cumbersome, time-consuming, and error-prone. Automating this support through linguistic engineering could save considerable time and improve software quality. This article includes an evaluation with real requirements from industry.

**Evidence-Based Software Engineering**

**Evidence-Based Software Engineering for Practitioners**

by Tore Dybå, Barbara A. Kitchenham, and Magne Jørgensen, pp. 58–65. Software engineers might make incorrect decisions about adopting new techniques if they don’t consider scientific evidence about the techniques’ efficacy. Procedures used for evidence-based medicine can also apply to software engineering. Such evidence-based software engineering fits well with current ideas concerning software process improvement and could be an important means for closing the gap between research and practice. However, EBSE presents difficulties for practitioners because current software engineering research is limited and not reported in a manner that assists evidence accumulation and evaluation evidence.

**Soup or Art? The Role of Evidential Force in Empirical Software Engineering**

by Shari Lawrence Pfleeger, pp. 66–73. Project managers make decisions about resources, tools, techniques, and more, often aiming to identify “good” or “best” practices. But such decisions should be based on solid evidence rather than common wisdom or vendor hype. Choosing the right kind of evidence helps build strong arguments to support more effective decision-making.

**Web Metadata Standards: Observations and Prescriptions**

by David Bodoff, Mordechai Ben-Menachem, and Patrick C.K. Hung, pp. 78–85. The proliferation of Web metadata standards presents an opportunity to recall key lessons from related disciplines, such as software engineering, software reuse, library science, and knowledge representation. However, the haste to advance these standards also entails a danger that such lessons might be overlooked. In addition, the apparent confluence of metadata standards and artificial intelligence will likely yield mutual benefits but might also divert researchers’ energies away from interoperability and other key issues. This article reviews various Web metadata standards and calls attention to lessons learned from related disciplines.

**In-House Software Development: What Project Management Practices Lead to Success?**

By June M. Verner and William M. Evanco, pp. 86–93. Because most software engineering research focuses on technical matters, information about in-house software development project management practices is lacking. A survey of 122 development projects investigates the practices that lead to successful outcomes. A clear vision of the final product, good requirements, accurate risk assessment and management, and postmortem reviews can all help increase the odds of success.