Stakeholders in Global Requirements Engineering: Lessons Learned from Practice
by Daniela Damian, pp. 21–27. Because of its communication- and collaboration-intensive nature, RE is a key challenge in global software engineering. The stakeholders in distributed projects must specify and manage requirements across cultural, time zone, and organizational boundaries. This article reports on the state of global software engineering practice and offers practical advice to alleviate stakeholders’ challenges, as distilled from empirical studies.

Wiki-Based Stakeholder Participation in Requirements Engineering
by Bjorn Decker, Eric Ras, Jörg Rech, Pascal Jaubert, and Marco Rieth, pp. 28–35. Requirements elicitation and documentation are complex activities. The quality of their products can improve through stakeholder participation, particularly in high-uncertainty projects. However, participative RE, especially in distributed environments, needs a platform that can support effective collaboration. The authors adapted the Wikipedia approach to collaboration in content creation to support active stakeholder participation in RE, including a document structure for wiki-based RE. They discuss challenges and solutions based on their experience.

Stakeholder Risk Assessment: An Outcome-Based Approach
by Richard W. Woolridge, Denise J. McManus, and Joanne E. Hale, pp. 36–45. RE must manage the risks arising from project stakeholders. The Outcome-Based Stakeholder Risk Assessment Model (OB-SRAM) provides guidance in identifying stakeholders, their effects and perceptions, and the potentially negative responses that pose risks to the project and in assessing and prioritizing those risks. This information provides valuable input to a risk mitigation plan that lets you devote more resources and attention to the stakeholders that pose the greatest risk to your project. A case study offers a practical, step-by-step approach to this process.

Determining Stakeholder Needs in the Workplace: How Mobile Technologies Can Help
by Neil Maiden, Norbert Seyff, Paul Grünbacher, Omo Otojare, and Karl Mitteregger, pp. 46–52. Recent advances in mobile computing technologies mean that mobile tools have the potential to determine stakeholder needs in the workplace, with possible benefits to requirements processes. However, little is known about their advantages and weaknesses. This article reports on workplace use of mobile requirements tools for PDAs; results from several studies reveal six lessons learned.

So, You Think You Know Others’ Goals? A Repertory Grid Study
by Nan Niu and Steve Easterbrook, pp. 53–61. The authors’ approach, based on the Repertory Grid Technique, detects when stakeholders have different interpretations of the terms they use to describe their problem domain. By comparing stakeholders’ grids, the authors highlight the mismatches and then generate follow-up questions to resolve them.

Metrics-Based Management of Software Product Portfolios
by Sunita Chulani, P. Santhanam, Brent Hodges, and Kelley Blacksten Anders, pp. 66–72. This metrics framework can help organizations manage their portfolios of software products and applications, with financial and developmental implications. The framework heuristically groups products into three categories—emerging, growth, and stable. The article focuses mainly on one characteristic: quality (as represented by software defects and customer-reported field problems) and its impact on service costs. The authors analyze actual data for 27 software products, highlight lessons learned, and discuss business scenarios that managers can use to exploit this framework.

Misleading Metrics and Unsound Analyses
by Barbara Kitchenham, David Ross Jeffery, and Colin Connaughton, pp. 73–78. Although software measurement is a valuable management tool, software metrics are often not as useful as practitioners hope. Using data from large corporate databases, the authors describe how the ISO/IEC 15393 standard gives inappropriate advice for measuring software engineering processes. They also show how this advice, when combined with the CMM/CMMI level 4 requirement for statistical process control, encourages the use of misleading metrics and inappropriate data aggregation and analysis techniques. They summarize lessons learned and recommend using small, meaningful data sets and effort-estimation models to assess productivity.

A Practical Approach for Quality-Driven Inspections
by Christian Denger and Forrest Shull, pp. 79–86. Software inspection is a rigorous, efficient, and cost-effective process for validating software work products. However, this process presents challenges that might thwart software developers from using it. Major reasons for this are poor or missing customization of inspections for given context characteristics and insufficient stakeholder involvement. The TAQtIC (Tailoring Approach for Quality-Driven Inspections) approach lets organizations implement customized inspections in a sustainable way in a given organizational context.