article summaries

MARCH/APRIL 2003, VOLUME TWENTY, NUMBER TWO

PROCESS

Why the Vasa Sank: 10 Problems and Some Antidotes for Software Projects

by Richard E. Fairley and Mary Jane Willshire, pp. 18–25. In 1628, the newest ship in the Royal Swedish Navy took its maiden voyage. After sailing about 1,300 meters, a light gust of wind caused the Vasa to capsize. The reasons that the Vasa was constructed to be unstable, and the reasons it was launched when known to be unstable, are as relevant to our modern-day attempts to build large, complex software systems as they were to the 17th-century art and craft of building warships. This article describes the problems encountered in that project, interprets them in terms of modern software projects, and presents some antidotes for those problems.

Managing Code Ownership

by Martin E. Nordberg III, pp. 26–33. Solving the problem of failed project ownership is not merely a process of assigning developers to subsystems. This article describes four code ownership models, ranging from one individual owning an entire system to the collective ownership that is a core Extreme Programming practice. No single code ownership model is best. In fact, many projects would benefit from dynamically adjusting the ownership model over time as development progresses through, for example, the four phases of the Unified Process.

A Process Model for Component-Oriented Software Engineering

by Ali H. Dogru and Murat M. Tanik, pp. 34–41. The investigation of reuse has a long history, eventually maturing into the "build by integration" paradigm. Meanwhile, component technologies also improved along with engineering practices. What is missing is a methodology that uses components within such a paradigm, thus bridging the gap. The authors propose a model devoted to complete component orientation, rather than modifying object-oriented approaches to accommodate components. They investigate component-based technologies to address new needs and, consequently, to integrate adequate practices toward a consistent process model.

Blending CMM and Six Sigma to Meet Business Goals

by Mala Murugappan and Gargi Keeni, pp. 42–48. Tata Consultancy Services blended Six Sigma concepts with the various SW-CMM key process areas, thereby creating a quality management system. This helped TCS improve its customer focus and sustain process improvement initiatives by explicitly linking them to business goals. The TCS team implemented the QMS on the lines of Level 2 and 3 requirements of SW-CMM, using Six Sigma concepts to reinforce quantitative process and product measurements and analysis, process improvements for defect prevention, and process optimization. This article describes TCS's approach, highlighting the benefits gained by blending Six Sigma and CMM to provide quality deliverables to its customers.

FEATURES

Task Descriptions as Functional Requirements

by Soren Lauesen, pp. 58–65. The Tasks & Support approach uses task descriptions that specify what the user and computer shall accomplish together without being explicit about who performs which parts of a task. The requirement is simply to support the identified tasks. Stakeholders can easily validate and later verify such requirements. This approach is just as successful for product development and large-scale work restructuring as it is for buying commercial off-the-shelf products. Although the resulting requirements are of higher quality than traditional requirements, they are much faster to produce.

The Intelligent Alarm Management System

by Jun Liu, Khiang Wee Lim, Weng Khuen Ho, Kay Chen Tan, Rajagopalan Srinivasan, and Arthur Tay, pp. 66–71. Nuisance alarms often clutter and obscure a process plant operator's view of critical information, with potentially severe consequences. The Intelligent Alarm Management System suppresses nuisance alarms and provides valuable advisory information to help the operator focus quickly on important alarm information and take correct, quick actions. Test results show that IAMS effectively suppresses nuisance alarms, making alarm systems more helpful.

Accelerating COTS Middleware Acquisition: The i-Mate Process

by Anna Liu and Ian Gorton, pp. 72–79. COTS middleware speeds e-business application deployment but can be difficult to select. The i-Mate tool provides a proven, structured software engineering process for COTS middleware acquisition. Using i-Mate in six major projects for a wide range of organizations led to highly visible, accountable, and ultimately reliable selections of COTS middleware products in greatly compressed time scales. This significantly reduced the risks associated with inappropriate product selections and made these projects more likely to succeed.