



ELSEVIER

The Journal of Systems and Software 61 (2002) 171–172

 **The Journal of
Systems and
Software**

www.elsevier.com/locate/jss

Guest Editor's Corner

Best practices in software engineering

Hossein Saiedian

Elec. Engineering and Computer Science, School of Engineering, University of Kansas, Lawrence, KS 66045, USA

This special issue of the *Journal of Systems and Software (JSS)* includes four of the best practice papers of the 22nd International Conference on Software Engineering (ICSE) as well as two additional invited articles presenting work in practical software engineering. All articles share a common theme: they represent case studies of collaborative projects between members of the academic and industrial or government institutions, thus highlighting the fact that progress in practical software engineering is and must be a joint collaborative effort.

ICSE, the flagship conference of the software engineering community, is well known for being a world-wide forum where researchers, practitioners, and educators together define and identify new directions of research and practice in software engineering, assess the state of the art, and exchange ideas. The 22nd ICSE was held in Limerick, Ireland, during June 4–11, 2000.

The conference included many excellent papers; a number of the best research papers were selected for consideration for a special issue of the *ACM Transactions on Software Engineering and Methodology* and four of the practice papers were recommended for further consideration for the *JSS*. The authors of selected papers were requested to revise their original papers and thus were provided with an opportunity to expand on their papers beyond the version published in the proceedings. All of the articles underwent a second review.

The selected and invited articles, while quite technical, emphasize technologies and concepts that are relevant to day to day software engineering practices and therefore are in line with the *JSS* mission of publishing those contributions that possess deep analytical value but the conclusions are supported by some form of application and a conscious effort to ground their findings in practice. A summary of each paper appears below.

Classification and Evaluation of Defects in a Project Retrospective. Marek Leszak (Lucent Technologies, Germany), Dewayne Perry (University of Texas at Austin, USA), and Dieter Stoll (Lucent Technologies, Germany) first present root-cause analysis of the defect modification requests (MRs) discovered while building, testing, and deploying a release of a network element as part of an optical transmission network. They discuss the novel approach they have taken to defect and root cause classification. Results of their analyses describe the defects and root causes that they found and the countermeasures created either to prevent those defects and their root causes or to detect them at the earliest possible point in the development process. They then present their findings on the correlation between defects detected and the adherence to our development process. Third the authors report on their experience with analyzing static code properties and their relation to observed defect numbers and defect densities. The authors conclude the report with lessons learned from the three investigations of the case study.

COTS-Based Software Development. The work described by Maurizio Morisio (Politecnico di Torino, Italy), Carolyn Seaman (University of Maryland Baltimore County, USA), Victor Basili (University of Maryland at College Park, USA), Amy Parra (Fraunhofer Center, Maryland, USA), Steve Kraft (NASA Goddard Space Flight Center, USA), and Steve Condon (Computer Science Corporation, USA) is an investigation of COTS-based software development within a particular NASA environment, with an emphasis on the processes used. Fifteen projects using a COTS-based approach were studied and their actual process was documented. This process is evaluated to identify essential differences in comparison to traditional software development. The main differences, and the activities for which projects require more guidance, are requirements definition and COTS selection, high level design, integration and testing. Starting from these empirical observations, a new process and set of guidelines for

E-mail address: saiedian@eecs.ukans.edu.

COTS-based development are developed and briefly presented.

Component-Based Software Development. Ivica Crnkovic (Mälardalen University, Sweden) and Magnus Larsson (ABB Automation Products, Sweden) discuss the practical challenges of component-based software development. The reuse orientation, founded on component-based development, provides many advantages, but it requires specific approach in design planning, extensive development, support of a more complex maintenance process, and in general more consideration being given to components. The authors' work discusses the requirements and problems of component-based approach by illustrating a case study of a successful example of the development of a component-based system. The authors discuss the following topics: (1) designing components and systems for reuse and with reuse, (2) trade-off between reusability and efficiency, (3) evolutions of component and system requirements and their impact on the development process. The authors also address the question of moving to new technologies which require the re-creation of the components or the inclusion of standard components available on the market. The increased complexity of the development and maintenance processes are described. The authors conclude that the component-based development require capability of managing complex processes.

CMM vs. Formal Methods. Carol Smidts, Xin Huang (University of Maryland), and James Widmaier (DOD) report on an experiment in producing “reliable software”. Two companies, one at the CMM Level 4 and another one with expertise in formal methods and automatic code generation develop the same product. The authors analyze the resulting products via different metrics. The analysis reveals the strength and weaknesses of each development and some of the widely held misconceptions.

Practical Security Engineering. The work of Rayford Vaughn (Mississippi State University), Ronda Henning and Kevin Fox (both with Harris Corporation) presents lessons learned and observations noted about the state of security-engineering practices by three information security practitioners with different perspectives – two in industry and one in academia. The authors offer insight into the problems faced today by security practitioners in bringing a comprehensive solution to a customer. Defense in depth as a prevention strategy is discussed as is the subjective nature of today's security engineering practice. Without the availability of strong metrics and high quality software systems, the provision of security becomes an art – relying heavily on the expertise and experience of the engineer. It is also a volatile field requiring periodic examination of the solution set in that vulnerabilities and attacks can changed quickly over time.

Interprocess Communication in Distributed Systems. David Andrews (University of Kansas), Paul Austin (Xerox Corporation), Peter Costello and David LeVan (both with Lockheed Martin Corporation) present a case study of the design and implementation of the interprocess communications facility developed for the AN/BSY-2 distributed computer system, the computer system for the Seawolf submarine. The interprocess communications facility is identified as a critical design challenge for the AN/BSY-2 system, as the system incorporated new component and network technology along with new run time system services as well as application programs. The requirements specified for the interprocess communications included aggressive performance, as well as functional capabilities that had not been previously fielded. The AN/BSY-2 system represents the largest real time system successfully fielded.

I thank the organizers of the conference, the authors, and the reviewers for all their efforts, and hope you enjoy their contributions.