



25th anniversary.....

IEEE Software's 25th-Anniversary Top Picks

Over the past 25 years, from 1984 through 2008, *IEEE Software* published more than 1,200 peer-reviewed full-length articles. As part of our 25th-anniversary celebration, *Software's* editorial and advisory boards embarked upon the ambitious task of distilling those articles into a compact list of recommended reading.

About the Selection Process

To select the articles, we relied on three sources: nominations, Web analytics, and citation statistics. We invited the members of *Software's* boards as well as former editors in chief to nominate their favorite full-length, peer-reviewed articles that appeared either in a special focus section or as a standalone piece. We excluded from the competition non-peer-reviewed content such as columns, short essays, news articles, letters, point-counterpoint pieces, interviews, and guest editor introductions.

The call for nominations generated 37 titles. We augmented this initial set using the remaining two sources. Web analytics consisted of download data from IEEE's digital libraries: IEEE Xplore and the Computer Society Digital Library. The heterogeneous download data covered the period from 2004 to 2008. The data were filtered, sanitized, consolidated, and reduced through set thresholds. The end result of this step was a set of 56 articles with relatively high PDF downloads. Citation statistics were based on Thomson ISI and Scopus data. Daniel O'Leary analyzed the Thomson ISI data from 1984 to mid-2008. His analysis is also published as a separate piece in *Software's* current issue (see p. 12). This set had a strong bias for older articles,

so we also requested assistance from editorial board member Diomidis Spinellis, who provided us with year-by-year citation statistics from an alternative source, Scopus, covering the period 2000–2008. We then merged the two sets of citation statistics to come up with 119 articles.

The next step involved condensing the three overlapping sets of 212 articles into a shortlist for a selection committee to review. We started with the list of 37 nominated articles. Then we identified criteria and thresholds to select articles for addition to the shortlist from the remaining two sets. We moved seven of the 56 articles in the high-downloads set and six of the remaining 119 articles in the highly cited set to the shortlist. The end result of this step was a shortlist of 50 articles.

To optimize the available review resources, we then placed each of the 50 articles in the shortlist into one of three reader-interest levels on the basis of the number of nominations the article received and the number of downloads and citations it generated. Five of the nominated articles with the strongest evidence of reader interest were assessed at level 1. Ten nominated articles were assessed at level 2. The remaining 35 articles with sufficient evidence of reader interest in terms of citations or downloads,

but not necessarily both, were assessed at level 3. Three selection committee members evaluated each level 3 article. Two selection committee members evaluated each level 2 article. Level 1 articles received a single evaluation each. The committee's evaluations rated the articles according to historical importance, appeal to *Software* readers, information value, applicability, popularity, and relevance. We didn't ask the committee to reassess an article's publication worthiness, just its relative worthiness of being singled out as recommended reading. We then reviewed the committee's evaluations and deliberated in several iterations to reach consensus. In one instance, we consulted with an outside expert for confirmation. Finally, we settled on 35 articles that represented, in the committee's opinion, the best of the bunch. This reduced set is our final selection (see Table 1).

The Selections

The selection committee was diverse. Including the three chairs, it comprised 22 members of the editorial and advisory boards, representing all our coverage areas as well as varied interests, geographical regions, professional roles, and generations. The final selection reflects this diversity. Our goal wasn't to eliminate subjectivity—doing so in such an endeavor is impossible. A different set of experts would undoubtedly come up with a different list. Thus it's entirely possible that we've missed articles that you think deserve to get on the final list. If we missed your favorite full-length articles, write to us and let us know about

the hidden gems. Don't forget to explain why the articles should have been included.

So, even if we can't guarantee that all these articles will appeal to everyone, our 25th-Anniversary Top Picks list hopefully contains something for every taste and orientation. Whether or not you agree with these articles' ultimate messages, you'll find them packed with wisdom and timeless ideas. We especially invite you to check out the older articles that have stood the test of time: you might be surprised by the degree of relevance they still possess many years after publication.

We've tagged the articles so that you can quickly identify your particular interests by topic or by category. Although some articles could easily fall into more than one category, we have only indicated the most fitting category for each article. If you're interested in a topic overview, scan the topic column and look for titles with the tag "O." If you're looking for original analyses based on experiences, observations, or surveys, look for the tag "E." Articles expounding pioneering ideas that had a significant

impact are tagged "P;" recent cutting-edge work is tagged "C," and reflections on specific themes, areas, or the profession in general are tagged "R."

Happy reading. ☺

Hakan Erdogmus, editor in chief

Frances Paulisch, chair of the Advisory Board

John Grundy, associate editor in chief

These articles will be freely available, three at a time, on a rotating basis throughout 2009 at www.computer.org/software/top-picks.

Acknowledgments

We wish to extend our sincere thanks to the Selection Committee: Elisa Baniassad, David J. Blaine, Annie Combelles, John Favaro, Robert Glass, Ann Hickey, Gargi Keeni, Neil Maiden, Grigori Melnik, Maurizio Morisio, Linda Rising, Martin Robillard, Helen Sharp, Forrest Shull, Diomidis Spinellis, Wolfgang Strigel, Laurence Tratt, and Uwe Zdun. Special thanks go to Daniel O'Leary and Diomidis Spinellis, who provided the citation data. We're also grateful to several members of *Software* advisory and editorial boards as well as past editors in chief for nominating their favorite articles.

Table 1

Our 25th-anniversary top picks for full-length peer-reviewed articles*

Title	Authors	Issue	Year	Topic	Category
Reusability: The Case for Object-Oriented Design	Bertrand Meyer	Mar./Apr.	1987	Design, programming	P
Characterizing the Software Process: A Maturity Framework	Watts S. Humphrey	Mar./Apr.	1988	Process	P
Seven Myths of Formal Methods	Anthony Hall	Sep./Oct.	1990	Requirements, quality	R
Prospects for an Engineering Discipline of Software	Mary Shaw	Nov./Dec.	1990	Professionalism	R
Software Risk Management: Principles and Practices	Barry W. Boehm	Jan./Feb.	1991	Project management	P
A Critical Look at Software Capability Evaluations	Terry B. Bollinger, Clement McGowan	Jul./Aug.	1991	Process	E
Operational Profiles in Software-Reliability Engineering	John D. Musa	Mar./Apr.	1993	Quality	P
The 4+1 View Model of Architecture	Philippe B. Kruchten	Nov./Dec.	1995	Architecture	P
Architectural Mismatch: Why Reuse Is So Hard	David Garlan, Robert Allen, John Ockerbloom	Nov./Dec.	1995	Architecture	E
Anchoring the Software Process	Barry Boehm	Jul./Aug.	1996	Process	P
Beyond Blaming: Congruence in Large Systems Development Projects	Jean McLendon, Gerald M. Weinberg	Jul./Aug.	1996	Project management	R
Using Patterns to Improve Our Architectural Vision	Norman L. Kerth, Ward Cunningham	Jan./Feb.	1997	Design, architecture	R
Process Control for Error-Free Software: A Software Success Story	Buford D. Tackett, Buddy Van Doren	May/June	1999	Quality, process	E

*O: Overview of a topic. E: Analyses based on experiences, observations, or surveys. P: Pioneering ideas that had an impact. C: Recent, cutting-edge ideas. R: Reflections on a topic or the profession.

Table 1 (continued)**Our 25th-anniversary top picks for full-length peer-reviewed articles***

Title	Authors	Issue	Year	Topic	Category
What Is Software Testing? And Why Is It So Hard?	James A. Whittaker	Jan./Feb.	2000	Testing	O
Strengthening the Case for Pair Programming	Laurie Williams, Robert R. Kessler, Ward Cunningham, Ron Jeffries	Jul./Aug.	2000	Process, programming	E
Attacking Malicious Code: A Report to the Infosec Research Council	Gary McGraw, Greg Morrisett	Sep./Oct.	2000	Security	E
Surviving Global Software Development	Christof Ebert, Philip De Neve	Mar./Apr.	2001	Process	E
Tactical Approaches for Alleviating Distance in Global Software Development	Erran Carmel, Ritu Agarwal	Mar./Apr.	2001	Process, project management	O
Extreme Programming from a CMM Perspective	Mark C. Paulk	Nov./Dec.	2001	Process	O
Usage-Centered Engineering for Web Applications	Larry L. Constantine, Lucy A.D. Lockwood	Mar./Apr.	2002	Requirements	P
The Pragmatics of Model-Driven Development	Bran Selic	Sep./Oct.	2003	Design, architecture	R
What Models Mean	Ed Seidewitz	Sep./Oct.	2003	Design, architecture	O
Software Development Worldwide: The State of the Practice	Michael Cusumano, Alan MacCormack, Chris F. Kemerer, Bill Crandall	Nov./Dec.	2003	Process	E
Righting Software	James R. Larus, Thomas Ball, Manuvir Das, Robert DeLine, Manuel Fähndrich, Jon Pincus, Sriram K. Rajamani, Ramanathan Venkatapathy	May/June	2004	Quality, tools	P
Measuring the ROI of Software Process Improvement	Rini van Solingen	May/June	2004	Project management	O
The Incremental Funding Method: Data-Driven Software Development	Mark Denne, Jane Cleland-Huang	May/June	2004	Project management	P
Provoking Creativity: Imagine What Your Requirements Could Be Like	Neil Maiden, Alexis Gizikis, Suzanne Robertson	Sep./Oct.	2004	Requirements	P
Integrated Requirements Engineering: A Tutorial	Ian Sommerville	Jan./Feb.	2005	Requirements	O
Evidence-Based Software Engineering for Practitioners	Tore Dybå, Barbara A. Kitchenham, Magne Jørgensen	Jan./Feb.	2005	Empirical software engineering	R
Architecture Reviews: Practice and Experience	Joseph F. Maranzano, Sandra A. Rozsypal, Gus H. Zimmerman, Guy W. Warnken, Patricia E. Wirth, David M. Weiss	Mar./Apr.	2005	Design, architecture	E
Architecture Decisions: Demystifying Architecture	Jeff Tyree, Art Akerman	Mar./Apr.	2005	Architecture	P
The Golden Age of Software Architecture	Mary Shaw, Paul Clements	Mar./Apr.	2006	Architecture	R
Professionalism and Test-Driven Development	Robert C. Martin	May/June	2007	Process, programming, professionalism	R
Tests and Requirements, Requirements and Tests: A Moebius Strip	Robert C. Martin, Grigori Melnik	Jan./Feb.	2008	Requirements, testing	C
Using Static Analysis to Find Bugs	Nathaniel Ayewah, William Pugh, David Hovemeyer, J. David Morgenthaler, John Penix	Sep./Oct.	2008	Quality, tools	C

*O: Overview of a topic. E: Analyses based on experiences, observations, or surveys. P: Pioneering ideas that had an impact. C: Recent, cutting-edge ideas. R: Reflections on a topic or the profession.