

Partial Contents

Part I: A Software Engineering Culture

- 1: Software Culture Concepts
- 2: Standing On Principle
- 3: Recognizing Achievements Great and Small
- 4: So Much to Learn, So Little Time

Part II: In Search of Excellent Requirements

- 5: Optimizing Customer Involvement
- 6: Tools for Sharing the Vision

Part III: Improving Your Processes

- 7: Process Improvement Fundamentals
- 8: Process Improvement Case Study
- 9: Software Process Maturity
- 10: Software Development Procedures

Part IV: The Bug Stops Here

- 11: The Quality Culture
- 12: Improving Quality by Software Inspection
- 13: Structured Testing

Part V: Methods, Measures, and Tools

- 14: The CASE for Iteration
- 15: Control Change Before It Controls You
- 16: Taking Measures to Stay on Track
- 17: Case Study: Measurement in a Small Software Group
- 18: If It Makes Sense, Do It

Part VI: What to Do on Monday

- 19: Action Planning for Software Managers
- 20: Action Planning for Software Engineers

Epilogue

- Appendix A: Sources for Continued Software Learning
- Appendix B: Contact Information for Selected Resources

Bibliography

Author Index

Subject Index

Creating a Software Engineering Culture

"... deals with how real people react to changes, to mandates, to new methods, and to both success and failure in their work. ... not just a theoretical discussion of how software engineering is supposed to work, but rather some informed observations of how things really do work in an actual company that wants to improve software."

—Capers Jones

Chairman, Software Productivity Research, Inc.

"... an insightful description of a healthy software engineering culture. His culture builder and culture killer tips present good commonsense advice to both software developers and managers on how to evolve a professional culture. ... a very readable book."

—Bill Curtis, Co-Founder and Chief Scientist, TeraQuest

"... the software engineering culture within your organization is critical to the success of development projects. ... A must-read for anyone interested in improving the way they develop software."

—Scott Ambler, *Software Development*

"Throughout the book, every step in the process, from the first idea down to the post release bug fixes, is stated, developed and critiqued from a team perspective. The author is a software process engineer who has clearly learned from his experience, both positive and negative."

—Charles Ashbacher

Mathematics and Computer Education

"Firmly rooted in the experiences of its author, this book offers the reader not just practical ideas of what to do, but also of ways to learn and think while doing them."

—Brian O'Laughlin

Software QA

About the Author



Karl E. Wieggers is a principal of Process Impact, a consulting firm based in Portland, Oregon. An author of numerous articles and a frequent speaker at conferences, Wieggers was formerly a software process improvement coordinator at Eastman Kodak Company.

Creating a Software Engineering Culture

by Karl E. Wieggers



ISBN: 0-932633-33-1

©1996 384 pages hardcover

\$45.95 (incl. \$6.00 for UPS in US)

Winner of a Software Development Productivity Award

Written in a remarkably clear style, *Creating a Software Engineering Culture* presents a comprehensive approach to improving the quality and effectiveness of the software development process.

In twenty chapters spread over six parts, Wieggers promotes the tactical changes required to support process improvement and high-quality software development.

Throughout the text, Wieggers identifies scores of culture builders and culture killers, and he offers a wealth of references to resources for the software engineer, including seminars, conferences, publications, videos, and on-line information.

With case studies on process improvement and software metrics programs and an entire Part

on action planning (called "What to Do on Monday"), this practical book guides the reader in applying the concepts to real life.

Topics include software culture concepts, team behaviors, the five dimensions of a software project, recognizing achievements, optimizing customer involvement, the project champion model, tools for sharing the vision, requirements traceability matrices, the capability maturity model, action planning, testing, inspections, metrics-based project estimation, the cost of quality, and much more!

"has a good chance of joining the select few books that ... become standard references for the software engineering world."

—Capers Jones

Software Productivity Research, Inc.

Read more about this book at
<http://www.dorsethouse.com/books/cse.html>