Software
Quality and
Productivity
Software Quality and Productivity
Theory, practice, education and training

Edited by
Matthew Lee
Department of Information Systems
City Polytechnic of Hong Kong
Kowloon Tong
Hong Kong

Ben-Zion Barta
Ministry of Education, Culture and Sports
Jerusalem
Israel

and

Peter Juliff
School of Management Information Systems
Deakin University
Victoria
Australia

SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.
CONTENTS

Preface
International Program Committee, Organizing Committee and Sponsors

PART ONE  Keynote papers

1  Towards the 21st Century's software quality, state-of-the-art and international standard in JTC1/SC7
M. Azuma

2  Software quality and productivity improvement
R.G. Dromey

PART TWO  Requirements, Design and Development Methodologies

3  Progress towards RACE: a 'soft-centred' requirements definition method
D.W. Bustard

4  MOOSE: A method designed for ease of maintenance
R.I. Ferguson, N.F. Parrington and P. Dunne

5  BPR for developing information systems – a case study
E. Coakes and J. Coakes

6  Towards a methodology for 24 hour software production using globally separated development teams
I. Gorton and S. Motwani

7  MetaMet – a soft systemic way toward the quality of information systems
P. Kokol and B. Stiglic

8  Changing programming paradigm – an empirical investigation
J. Miller, G. Darroch, M. Wood, A. Brooks and M. Roper

9  Application-oriented methods for systems development – a review
D. Millington and I.M. Tulloch

10  A proposal for composing software design process
S. Mochizuki and T. Katayama

11  Team-based software development using an electronic meeting system: the quality pay-off
S. Sawyer and P.J. Guinan

12  Designing with non-functional requirements
C.J. Theaker and J. Whitworth
13 Modelling the requirements of process controlled systems  
T.K. Sateesh and P.A.V. Hall  

14 Compiling heterarchical programs by means of Petri nets  
B. Evans  

PART THREE  Object Oriented Analysis, Design and Development  
15 Type checking classes in Object-Z to promote quality of specifications  
J. Chen and B. Durnota  

16 An axiom-based test case selection strategy for object-oriented programs  
T.H. Tse, F.T. Chan and H.Y. Chen  

17 AMI: a case tool based on artificial intelligence techniques for an interactive modelling for analysis  
L. Lasoudris, M. Maurice-Demourieux and R. Voyer  

18 Object-oriented framework for requirement specification  
S. Liao, J. Ma and L. Lasoudris  

19 A structured object-oriented view on systems modeling  
G. Wu and G. Dedene  

20 Construction of a WYSIWYG LaTeX typesetting system using object-oriented design  

21 A re-engineering approach for object-oriented database design  
J. Fong and I. Kwan  

PART FOUR  Software Process and Capability  
22 Software process model specification  
M. Baldi and M.L. Jaccheri  

23 Soft systems methodology – a process view  
G. Tate  

24 BOOTSTRAP – improving the capability of software industry with database support  
J. Jarvinen  

PART FIVE  Project Management  
25 On the optimal timing of software development cycle  
H.A. Wan  

26 Information systems project manager’s critical success factors: in-house verses outsourced development  
H. Rehesaar and A. Rose  

27 Measuring the value and performance of your IT department  
T. Lynch  

PART SIX  Quality, Reliability, Standards  
28 ISO standards in software engineering  
J. Billington
29 An investigation of desired quality attributes for different types of software  
   T. Musson and E. Dodman  
   193

30 A unifying model for software quality engineering  
   I. Tervonen  
   200

31 Quality achievement: current practices and the way ahead  
   J. B. Thompson  
   206

32 Software reliability models for practical applications  
   M. Xie  
   211

33 Assessing the quality of requirements engineering products  
   R.E.M. Champion  
   215

34 Re-certification of software reliability without re-testing  
   C. Wohlin  
   219

35 SAFE: a programming language for software quality  
   B.K. Oliver and R.G. Dromey  
   227

36 Impact of organisational maturity on software quality  
   M. Tan and C.Y. Yap  
   231

37 Using strongest postconditions to improve software quality  
   S. Pan and R.G. Dromey  
   235

PART SEVEN Specifications, Metrics, Assessment

38 Experience in software inspection techniques  
   D.A. Brunskill and A. Samuel  
   243

39 Measuring process improvement: integrating software process assessment  
   with quality cost evaluation  
   M.P. Neilson and T.P. Rout  
   250

40 Mutation versus all-uses: an empirical evaluation  
   of cost, strength and effectiveness  
   W.E. Wong, A.P. Mathur and J.C. Maldonado  
   258

41 Software quality: the user’s point of view  
   M. Xenos and D. Christodoulakis  
   266

42 Complexity measures for system development models  
   E.H. Ferneley, D.A. Howcroft and C.G. Davies  
   273

43 Composite complexity measures (CCM)  
   P.S. Grover and N.S. Gill  
   279

44 Automated testing tool for NATURAL®  
   284

45 Analysis of data collected from software development case studies  
   R. Hunter  
   290

46 A framework for software development technical reviews  
   L.P.W. Kim, C. Sauer and R. Jeffery  
   294
Contents

47 A model-based approach for software test process improvement
   L. Lian, F. Fujita, S. Kusumoto, K.-i. Matsumoto, K. Kikuno and K. Torii
   300

48 Size estimation for C programs with different applications
   C.J. Lokan
   308

49 Cyclomatic complexity as a measure of the structure of
   knowledge-based/expert systems
   T.T. Moores
   314

50 S/C: a software size/complexity measure
   Y.R. Pant, J.M. Verner and B. Henderson-Sellers
   320

51 A critical re-examination of cyclomatic complexity measures
   B. Henderson-Sellers and D. Tegarden
   328

52 Reliability through cooperative work in software testing
   T. Tuikka
   336

PART EIGHT  Reusability

53 Code reusability mechanisms
   X. Castellani and H. Jiang
   345

54 Rule-based reasoning approach for reusable design component retrieval
   S.-T. Lai and C.-C. Yang
   353

55 A multi-layer metrics combination model for reusable software components
   S.-T. Lai and C.-C. Yang
   361

PART NINE  Integrated Environments, Case Tools

56 Environment for three dimensional graphics user interface development
   369

57 PEACE: process evolution and adaptation CASE environment –
   its framework and requirements
   T. Yoshida, A. Murata and M. Azuma
   378

58 Software development environments to support model consistency
   V.E. Veraart and S.L. Wright
   382

PART TEN  Education and Training

59 Process oriented instruction for software engineers (POISE):
   perspectives and experiences
   S. Thummalapalli and R.K. Todi
   389

60 OO methods in teaching: current experience and future possibilities
   G. Lovegrove, P. McLaughlin, J. Chabik and P. Standen
   395

Index of contributors
   401

Keyword index
   402
The International Federation for Information Processing (IFIP) is a body which aims to co-ordinate the various facets of computing/information processing across the world. IFIP has an infrastructure of Technical Committees (TCs), each of which has a number of Working Groups (WGs). TC-3 is concerned with computers in education. Within this, W.G. 3.4 is a Working Group whose brief is education and training for IT professionals and advanced end-users while the brief of W.G. 5.4 is the quality and certification of industrial software.

The specific aims of WG3.4 include:

- the promotion of effective education and training of IT professionals and advanced end-users;
- the maintenance of up-to-date IT professional education curricula;
- the dissemination of knowledge on current best practice in IT education;
- the exploration of technological advances in the delivery of training and education; and
- the consideration of internationally recognised qualifications for professional certification and registration.

The Working Conference entitled “First IFIP/SQI International Conference on Software Quality and Productivity (ICSQP’94)” was produced with the co-operation of IFIP, the Software Quality Research Institute of Australia, the Hong Kong Computer Society and the City Polytechnic of Hong Kong.

The aim of the conference was to provide a forum for discussion among researchers and practitioners interested in the efficient production of quality software systems. The two major themes running through the conference were quality and productivity.

A total of 2 invited papers and 58 papers accepted from those submitted to the conference form the scientific and professional contents for the 3 days of lectures and discussions. To safeguard the quality of the conference, all the accepted papers have been subject to a peer group evaluation process. These papers shall provide the basis for three days of continued interest among the participants and ensure lively debates within the formal sessions of the conference and, not less important, during the free meetings in between the formal sessions.

As would be expected from any contemporary conference focusing on software productivity, many papers addressed the issues involved in object-oriented design and development. The shift to object-orientation as a major, if not predominant, paradigm for software development promises to deliver the increases in productivity which have been the goal of developers for the past two decades.
Preface

Issues relating to education and training bear out the necessity for education providers to have flexible curricula which can change to reflect advances in current theory and practice. The sharing of experiences from such a wide variety of sources is one of the major attractions of these conferences.

The production of quality software systems depends on reliable methods of producing software specifications and their subsequent evaluation against those requirements. Many of the papers addressed issues related to software metrics and the approaches able to be applied to software testing.

There was much discussion of design and development methodologies and their role in the issues of both quality and productivity and of the perennial problem of software reusability. The movement towards object-orientation as a design paradigm has had reusability as one of its major cornerstones. The papers in these proceedings should provide a source of current thinking and practice in this area.

We trust that readers of these proceedings will obtain much of the benefit enjoyed by those who participated in the conference.

We wish to express our thanks for the authors of the paper and to those who laboured long and hard to bring the conference itself to fruition and to enable these proceedings to be compiled.

The editors.
INTERNATIONAL PROGRAM COMMITTEE

Dr Ben Zion BARTA (Chair)  
Professor Sandro BOLOGNA (Co-Chair)  
Professor R. Geoff DROMEY (Co-Chair)  
Professor Peter JULIFF (Co-Chair)  
Dr Matthew K. O. LEE (Co-Chair)  
Prof Phillip Sallis  
Prof M Azuma  
Prof M Maiocchi  
Prof J Billington  
Prof Dave Bustard  
Mr Mike Deutsch  
Prof N Fenton  
Prof C Ghezzi  
Prof Dick Hamlet  
Prof Robin Hunter  
Prof Graham Tate  
Prof Su Yunlin  
Mr Terry Rout  
Dr Min Xie  
Dr Chuk Yau  
Dr June Verner  
Dr T H Tse  
Dr Ernest Lam  
Dr B Kitchenham  
Dr John Edwards  
Dr Trevor Moores  

Israel  
Italy  
Australia  
Australia  
Hong Kong  
New Zealand  
Japan  
Italy  
Australia  
United Kingdom  
United States  
United Kingdom  
Italy  
United States  
United Kingdom  
Hong Kong  
People's Republic of China  
Australia  
Singapore  
Australia  
Hong Kong  
Hong Kong  
Hong Kong  
Hong Kong  
United Kingdom  
Hong Kong

ORGANIZING COMMITTEE

Dr Matthew K. O. LEE (Chair)  
Dr S. L. HUNG (Co-Chair)  
Mr T. P. Rout (Co-Chair)  

Hong Kong  
Hong Kong  
Australia

SPONSORS

International Federation of Information Processing (IFIP)  
W.G.3.4 (Education and Training of IT Professionals and End Users)  
W.G.5.4 (Industrial Software Quality and Certification)  
Software Quality Research Institute (SQI), Australia  
Hong Kong Computer Society

The City Polytechnic of Hong Kong supported the Conference as part of its 10th Anniversary Academic Conference Programme.