
Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

For further volumes:
<http://www.springer.com/series/7592>

Gerard O'Regan

Introduction to Software Quality

 Springer

Gerard O'Regan
SQC Consulting
Mallow, Cork, Ireland

Series Editor
Ian Mackie

Advisory Board

Samson Abramsky, University of Oxford, Oxford, UK
Karin Breitman, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
Chris Hankin, Imperial College London, London, UK
Dexter Kozen, Cornell University, Ithaca, USA
Andrew Pitts, University of Cambridge, Cambridge, UK
Hanne Riis Nielson, Technical University of Denmark, Kongens Lyngby, Denmark
Steven Skiena, Stony Brook University, Stony Brook, USA
Iain Stewart, University of Durham, Durham, UK

ISSN 1863-7310
ISBN 978-3-319-06105-4
DOI 10.1007/978-3-319-06106-1
Springer Cham Heidelberg New York Dordrecht London

ISSN 2197-1781 (electronic)
ISBN 978-3-319-06106-1 (eBook)

Library of Congress Control Number: 2014936841

© Springer International Publishing Switzerland 2014

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To
Kevin and Maura and the four princesses
(Eve, Grace, Jane and Tara)*

Preface

Overview

The objective of this book is to provide an introduction to the software quality field to students and practitioners, and it is based on the author's experience in software quality and software process improvement at leading industrial companies. The principles of software quality management and software process improvement are discussed.

The goal is to cover both theory and practice, and to give the reader a grasp of the fundamentals of the software quality field, as well as guidance on how to apply the theory in an industrial environment.

Organization and Features

The first chapter provides an introduction to the fundamentals of the quality management field, and provides historical background on several pioneers such as Deming, Juran, and Crosby.

Chapter 2 provides a broad overview of software engineering and discusses various software lifecycles and the phases in software development. It includes a discussion on requirements elicitation, software design, implementation, testing, and maintenance.

Chapter 3 provides an introduction to project management and discusses project estimation, project planning and scheduling, project monitoring and control, risk management, and managing project quality.

Chapter 4 discusses requirements and design and is concerned with requirements engineering and management, architectural design, and design and development.

Chapter 5 discusses configuration management and discusses the fundamental concept of a baseline. Configuration management is concerned with identifying those deliverables that must be subject to change control, and controlling changes to them.

Chapter 6 discusses software inspections which play a key role in building quality into a product. The well-known Fagan inspection process which was developed at IBM in the 1970s is discussed, as well as lighter review and walk-through methodologies.

Chapter 7 is concerned with software testing and discusses the various types of testing that may be carried out. It includes a discussion on test planning, test case definition, test tracking, test metrics, test reporting, and testing in an e-commerce environment.

Chapter 8 is concerned with the selection and management of a software supplier. It discusses how candidate suppliers may be formally evaluated, and how the selected supplier may be managed during the project.

Chapter 9 discusses software quality assurance and the importance of process quality. It is a premise in the quality field that conformance to the defined process is essential in the delivery of high-quality product, and this chapter discusses audits, and describes how they are carried out.

Chapter 10 is concerned with metrics and problem solving, and this includes a discussion of the balanced score card which assists in identifying appropriate metrics for the organization. The Goal, Question, Metrics (GQM) approach is discussed, and this is useful in defining metrics that are related to the organization goals. This chapter includes a collection of sample metrics for an organization. Problem solving tools such as fishbone diagrams, pareto charts, and trend charts are also discussed.

Chapter 11 discusses the ISO 9000 standard, which is an important standard for product and service delivery. This family of standards includes ISO 9001 and ISO 9004. The main features of the standard are discussed as well as guidance on its implementation.

Chapter 12 discusses software process improvement. It begins with a discussion of a software process, and discusses the benefits that may be gained from a software process improvement initiative. Various models that support software process improvement are discussed, and these include the CMMI, ISO 9000, PSP, and TSP.

Chapter 13 gives an overview of the CMMI model and discusses its five maturity levels and their constituent process areas. It includes a discussion of both the staged and continuous representations.

Chapter 14 describes the activities and teams required to set up a CMMI improvement initiative for an organization. These include the CMMI Steering Group, the SEPG team, and process specific teams.

Chapter 15 discusses the SCAMPI appraisal methodology. This includes the formal SCAMPI Class A appraisal often employed by large organizations to obtain a CMMI rating that allows them to benchmark themselves against other organizations, and SCAMPI Class B and C appraisals that are less expensive and time consuming but may not be used for benchmarking.

Chapter 16 discusses various tools to support the organizations in the various software engineering activities. The focus is first to define the process, and then to find tools to support the process. Tools to support project management are discussed as well as tools to support requirements engineering, configuration management, design and development activities, and software testing.

Chapter 17 discusses formal methods, which consist of a set of mathematical techniques to specify and derive a program from its specification. Formal methods may be employed to rigorously state the requirements of the proposed system; they

may be employed to derive a program from its mathematical specification; and they provide a rigorous proof that the implemented program satisfies its specification. They have been mainly applied to the safety critical field.

Chapter 18 presents the Z specification language, which is one of the most widely used formal methods. It was developed at Oxford University in the UK.

Chapter 19 presents the unified modelling language (UML) which is used to present several views of the system architecture. Chapter 20 is the concluding chapter in which we summarize the journey that we have travelled in this book.

Audience

The main audience of this book are computer science students who are interested in learning about software quality, and in learning on how to build high-quality and reliable software on time and on budget. It will also be of interest to industrialists including software engineers, quality professionals, and software managers as well as the motivated general reader.

Mallow, Cork, Ireland

Gerard O'Regan

Acknowledgments

I am deeply indebted to family and friends who supported my efforts in this endeavour.

Contents

1	Introduction	1
1.1	Introduction	1
1.1.1	The Software Engineering Challenge	2
1.2	History of Software Failures	4
1.3	Background to Software Quality	5
1.3.1	What Is Software Quality?	5
1.3.2	Early Quality Management	6
1.3.3	Total Quality Management	6
1.3.4	Software Quality Control	7
1.4	History of Quality	8
1.4.1	Shewhart	8
1.4.2	Deming	10
1.4.3	Juran	12
1.4.4	Crosby	15
1.4.5	Watts Humphrey	17
1.4.6	Miscellaneous Quality Gurus	19
1.5	Modern Software Quality Management	20
1.5.1	Software Inspections	20
1.5.2	Software Testing	21
1.5.3	Software Quality Assurance	21
1.5.4	Problem Solving Techniques	22
1.5.5	Cost of Quality	24
1.5.6	Software Process Improvement	25
1.5.7	Software Metrics	26
1.5.8	Customer Satisfaction	26
1.5.9	Assessments (Appraisals)	29
1.5.10	Total Quality Management	29
1.6	Miscellaneous	30
1.6.1	Organization Culture and Change	30
1.6.2	Law of Negligence	31
1.6.3	Quality and the WEB	31
1.7	Review Questions	32
1.8	Summary	32

2	Software Engineering	35
2.1	Introduction	35
2.2	What Is Software Engineering?	38
2.3	Challenges in Software Engineering	40
2.4	Software Processes and Lifecycles	42
2.4.1	Waterfall Lifecycle	43
2.4.2	Spiral Lifecycles	44
2.4.3	Rational Unified Process	45
2.4.4	Agile Development	46
2.5	Activities in Waterfall Lifecycle	47
2.5.1	Business Requirements Definition	48
2.5.2	Specification of System Requirements	48
2.5.3	Design	49
2.5.4	Implementation	50
2.5.5	Software Testing	50
2.5.6	Maintenance	52
2.6	Software Inspections	53
2.7	Software Project Management	53
2.8	CMMI Maturity Model	54
2.9	Formal Methods	55
2.10	Review Questions	56
2.11	Summary	56
3	Project Management	59
3.1	Introduction	59
3.2	Project Start Up and Initiation	61
3.3	Estimation	62
3.3.1	Estimation Techniques	63
3.3.2	Work Breakdown Structure	64
3.4	Project Planning and Scheduling	65
3.5	Risk Management	67
3.6	Quality Management in Projects	69
3.7	Project Monitoring and Control	70
3.8	Managing Issues and Change Requests	71
3.9	Project Board and Governance	72
3.10	Project Reporting	73
3.11	Project Closure	73
3.12	Prince 2 Methodology	74
3.13	Review Questions	76
3.14	Summary	76
4	Requirements, Design and Development	77
4.1	Introduction	77
4.2	Requirements Engineering	78
4.2.1	Requirements Elicitation and Specification	79
4.2.2	Requirements Analysis	81

4.2.3	Requirements Verification and Validation	81
4.2.4	Managing Changes to Requirements	82
4.2.5	Requirements Traceability	82
4.3	Architecture Design	84
4.4	Design and Development	86
4.5	Review Questions	87
4.6	Summary	87
5	Configuration Management	89
5.1	Introduction	89
5.2	Configuration Management System	93
5.2.1	Identify Configuration Items	93
5.2.2	Document Control Management	93
5.2.3	Source Code Control Management	94
5.2.4	Configuration Management Plan	95
5.3	Change Control	95
5.4	Configuration Management Audits	98
5.5	Review Questions	98
5.6	Summary	99
6	Software Inspections	101
6.1	Introduction	101
6.2	Economic Benefits of Software Inspections	103
6.3	Informal Reviews	104
6.4	Structured Walkthrough	104
6.5	Semi-formal Review Meeting	105
6.6	Fagan Inspections	108
6.6.1	Fagan Inspection Guidelines	109
6.6.2	Inspectors and Roles	110
6.6.3	Inspection Entry Criteria	110
6.6.4	Preparation	110
6.6.5	The Inspection Meeting	112
6.6.6	Inspection Exit Criteria	114
6.6.7	Issue Severity	114
6.6.8	Defect Type	114
6.7	Automated Software Inspections	116
6.8	Review Questions	117
6.9	Summary	117
7	Software Testing	119
7.1	Introduction	119
7.2	Test Process	121
7.3	Test Planning	125
7.4	Test Case Design and Definition	126
7.5	Test Reporting and Project Sign-off	127
7.6	Testing and Quality Improvement	128

7.7	Traceability of Requirements	129
7.8	Test Tools	130
	7.8.1 Test Management Tools	130
	7.8.2 Miscellaneous Testing Tools	131
7.9	E-commerce Testing	131
7.10	Review Questions	133
7.11	Summary	133
8	Supplier Selection and Management	135
8.1	Introduction	135
8.2	Planning and Requirements	136
8.3	Identifying Suppliers	137
8.4	Prepare and Issue RFP	137
8.5	Evaluate Proposals and Select Supplier	138
8.6	Formal Agreement	138
8.7	Managing the Supplier	139
8.8	Acceptance of Software	139
8.9	Rollout	140
8.10	Review Questions	140
8.11	Summary	140
9	Software Quality Assurance	143
9.1	Introduction	143
9.2	Audit Planning	146
9.3	Audit Meeting	147
9.4	Audit Reporting	148
9.5	Follow Up Activity	149
9.6	Audit Escalation	149
9.7	Review of Audit Activities	149
9.8	Review Questions	149
9.9	Summary	150
10	Software Metrics	151
10.1	Introduction	151
10.2	The Goal Question Metric Paradigm	152
	10.2.1 Goal	153
	10.2.2 Question	154
	10.2.3 Metrics	154
10.3	The Balanced Scorecard	154
10.4	Metrics for an Organization	156
	10.4.1 Customer Satisfaction Metrics	157
	10.4.2 Process Improvement Metrics	158
	10.4.3 Human Resources and Training Metrics	160
	10.4.4 Project Management Metrics	162
	10.4.5 Development Quality Metrics	164
	10.4.6 Quality Audit Metrics	167

10.4.7	Customer Care Metrics	168
10.4.8	Miscellaneous Metrics	170
10.5	Implementing a Metrics Program	172
10.5.1	Data Gathering for Metrics	173
10.6	Problem-Solving Techniques	174
10.6.1	Fishbone Diagram	175
10.6.2	Histograms	177
10.6.3	Pareto Chart	178
10.6.4	Trend Graphs	179
10.6.5	Scatter Graphs	180
10.6.6	Metrics and Statistical Process Control	181
10.7	Review Questions	182
10.8	Summary	183
11	ISO 9000	185
11.1	Introduction	185
11.2	Motivation for ISO 9000	186
11.3	ISO 9000	187
11.3.1	Quality Management System	187
11.3.2	Management Responsibility	189
11.3.3	Resource Management	190
11.3.4	Product or Service Realization	191
11.3.5	Measuring, Analysis, and Improvement	193
11.4	Implementing ISO 9001	195
11.5	ISO 9000 and Improvement	195
11.5.1	Self-Assessment Process	195
11.5.2	ISO 9001 Certification Process	195
11.6	Review Questions	197
11.7	Summary	198
12	Software Process Improvement	199
12.1	Introduction	199
12.2	What Is a Software Process?	200
12.3	What Is Software Process Improvement?	202
12.4	What Are the Benefits of Software Process Improvement?	203
12.5	What Models Are Used in Software Process Improvement?	204
12.6	Process Mapping	206
12.7	Process Improvement Initiatives	206
12.8	Barriers to Success	207
12.9	Review Questions	208
12.10	Summary	208
13	Capability Maturity Model Integration	211
13.1	Introduction	211
13.2	The CMMI	214

13.3	CMMI Maturity Levels	217
13.3.1	CMMI Representations	220
13.4	Categories of CMMI Processes	222
13.5	CMMI Process Areas	222
13.6	Components of CMMI Process Areas	223
13.6.1	SG 1 – Manage Requirements	227
13.7	SCAMPI Appraisals	229
13.8	Review Questions	231
13.9	Summary	231
14	Setting Up a CMMI Initiative	233
14.1	Introduction	233
14.2	Approach to Continuous Improvement	234
14.3	CMMI Improvement Structure and Teams	236
14.3.1	Setting Up the SEPG Team	237
14.3.2	Setting Up the Steering Group	240
14.3.3	Setting Up Dedicated Improvement Sub-teams	241
14.3.4	Role of the CMMI Project Manager	242
14.3.5	Risks to Success	242
14.4	Planning the Improvement Cycle	243
14.4.1	Appraisals	243
14.4.2	CMMI Project Plan	244
14.4.3	CMMI Project Schedule	245
14.4.4	CMMI Kick-off Session	245
14.5	Implementation of Improvements	246
14.5.1	Process Mapping	246
14.5.2	Layout of Templates	248
14.5.3	Layout of Procedures and Guidelines	248
14.6	Piloting the Process	249
14.7	Rolling Out Process	249
14.8	Review Questions	250
14.9	Summary	250
15	SCAMPI Appraisals	253
15.1	Introduction	253
15.2	Planning and Requirements for the Appraisal	256
15.2.1	Analyze Requirements	256
15.2.2	Develop Appraisal Plan	257
15.2.3	Select and Prepare Team	258
15.2.4	Obtain and Analyze Initial Evidence	259
15.2.5	Prepare for Conducting Appraisal	260
15.3	Conducting the Appraisal	261
15.3.1	Prepare Participants	261
15.3.2	Examine Objective Evidence	262
15.3.3	Document Objective Evidence	264

15.3.4	Verify Objective Evidence	265
15.3.5	Validate Preliminary Findings	266
15.3.6	Generate Appraisal Results	266
15.4	Reporting the Results	267
15.4.1	Deliver Appraisal Results	267
15.4.2	Archive Appraisal Results	267
15.5	Review Questions	268
15.6	Summary	268
16	Software Engineering Tools	271
16.1	Introduction	271
16.2	Tools for Project Management	272
16.3	Tools for Requirements	274
16.4	Tools for Design and Development	278
16.5	Tools for Configuration Management and Change Control	280
16.6	Tools for Code Analysis and Code Inspections	283
16.7	Tools for Testing	284
16.8	Review Questions	286
16.9	Summary	286
17	Formal Methods	289
17.1	Introduction	289
17.2	Why Should We Use Formal Methods?	291
17.3	Applications of Formal Methods	292
17.4	Tools for Formal Methods	293
17.5	Approaches to Formal Methods	294
17.5.1	Model-Oriented Approach	294
17.5.2	Modelling	295
17.5.3	Axiomatic Approach	296
17.6	Proof and Formal Methods	296
17.7	The Future of Formal Methods	298
17.8	The Vienna Development Method	298
17.9	VDM [*] , the Irish School of VDM	299
17.10	The Z Specification Language	301
17.11	The B Method	302
17.12	Predicate Transformers and Weakest Preconditions	302
17.13	The Process Calculii	303
17.14	Finite State Machines	304
17.15	The Parnas Way	305
17.16	Usability of Formal Methods	306
17.16.1	Why Are Formal Methods difficult?	307
17.16.2	Characteristics of a Usable Formal Method	308
17.17	Review Questions	308
17.18	Summary	309

18	Z Formal Specification Language	311
18.1	Introduction	311
18.2	Sets	314
18.3	Relations	315
18.4	Functions	316
18.5	Sequences	318
18.6	Bags	319
18.7	Schemas and Schema Composition	320
18.8	Reification and Decomposition	322
18.9	Proof in Z	323
18.10	Review Questions	324
18.11	Summary	325
19	Unified Modelling Language	327
19.1	Introduction	327
19.2	Overview of UML	328
19.3	UML Diagrams	330
19.3.1	Advantages of UML	335
19.4	Rational Unified Process	335
19.5	Review Questions	337
19.6	Summary	337
20	Epilogue	339
20.1	The Future of Software Quality	341
	Glossary	343
	References	347
	Index	351

List of Figures

Fig. 1.1	Standish research – Project cost estimation accuracy in 1998	3
Fig. 1.2	Shewhart’s control chart	9
Fig. 1.3	Shewhart’s PDCA cycle	9
Fig. 1.4	W.E. Deming	10
Fig. 1.5	Joseph Juran	13
Fig. 1.6	Cost of poor quality – % of sales	13
Fig. 1.7	Estimation accuracy – Breakthrough and control	14
Fig. 1.8	Watts Humphrey (Courtesy of Watts Humphrey)	18
Fig. 1.9	Cost of quality	25
Fig. 1.10	Customer satisfaction process	27
Fig. 1.11	Customer satisfaction metrics	28
Fig. 2.1	Standish report – Results of 1995 and 2009 survey	37
Fig. 2.2	Standish 1998 report – Estimation accuracy	41
Fig. 2.3	Waterfall V lifecycle model	43
Fig. 2.4	SPIRAL lifecycle model Public domain	44
Fig. 3.1	Simple process map for project planning	65
Fig. 3.2	Sample Microsoft project schedule	66
Fig. 3.3	Simple process map for project monitoring and control	70
Fig. 3.4	Prince 2 project board	72
Fig. 3.5	Project management triangle	74
Fig. 3.6	Prince 2 processes	75
Fig. 4.1	C.A.R. Hoare (Public domain)	85
Fig. 4.2	David Parnas (Public domain)	85
Fig. 5.1	Simple process map for change requests	96
Fig. 5.2	Simple process map for configuration management	97
Fig. 6.1	Michael Fagan	102
Fig. 6.2	Template for semi-formal review	107
Fig. 6.3	Template for Fagan inspection	113
Fig. 6.4	Sample-defect types in a project (ODC)	116

Fig. 7.1	Simplified test process	122
Fig. 7.2	Sample test status	124
Fig. 7.3	Cumulative defects	127
Fig. 7.4	Phase containment effectiveness metric	128
Fig. 9.1	Sample audit process	145
Fig. 10.1	GQM example	153
Fig. 10.2	The balanced scorecard	155
Fig. 10.3	Balanced score card and implementing strategy	155
Fig. 10.4	Customer survey arrivals	157
Fig. 10.5	Customer satisfaction measurements	158
Fig. 10.6	Process improvement measurements	159
Fig. 10.7	Status of process improvement suggestions	159
Fig. 10.8	Age of open process improvement suggestions	160
Fig. 10.9	Process improvement productivity	160
Fig. 10.10	Employee headcount in current year	161
Fig. 10.11	Employee turnover in current year	161
Fig. 10.12	Schedule timeliness metric	162
Fig. 10.13	Effort timeliness metric	163
Fig. 10.14	Requirements delivered	163
Fig. 10.15	Total number of issues in project	164
Fig. 10.16	Open issues in project	165
Fig. 10.17	Age of open defects in project	165
Fig. 10.18	Problem arrivals per month	166
Fig. 10.19	Phase containment effectiveness	166
Fig. 10.20	Annual audit schedule	167
Fig. 10.21	Status of audit actions	167
Fig. 10.22	Audit action types	168
Fig. 10.23	Customer queries (arrivals/closures)	169
Fig. 10.24	Outage time per customer	170
Fig. 10.25	Availability of system per month	171
Fig. 10.26	Configuration management	171
Fig. 10.27	CMMI maturity in current year	172
Fig. 10.28	Fishbone cause-and-effect diagram	176
Fig. 10.29	Histogram	178
Fig. 10.30	Pareto chart outages	179
Fig. 10.31	Trend chart estimation accuracy	180
Fig. 10.32	Scatter graph amount inspected rate/error density	181
Fig. 10.33	Estimation accuracy and control charts	182
Fig. 11.1	ISO 9000 quality management system	188
Fig. 12.1	Process as glue for people, procedures and tools	201
Fig. 12.2	Sample process map	202

Fig. 13.1	Process as glue for people, procedures and tools	212
Fig. 13.2	ISO/IEC 12207 standard for software engineering processes	213
Fig. 13.3	CMMI Worldwide maturity 2013	216
Fig. 13.4	CMMI maturity levels	218
Fig. 13.5	CMMI capability levels	221
Fig. 13.6	CMMI – Continuous representation	222
Fig. 13.7	CMMI staged model	226
Fig. 13.8	Specific practices for SG1 – manage requirements	227
Fig. 14.1	Steps in process improvement	235
Fig. 14.2	Continuous improvement cycle	235
Fig. 14.3	CMMI Level 2 improvement structure and teams	239
Fig. 14.4	Sample process map	247
Fig. 15.1	Appraisals	254
Fig. 15.2	SCAMPI CMMI L3 rating of practices	255
Fig. 15.3	SCAMPI (classes of appraisals)	256
Fig. 15.4	Sample schedule for SCAMPI Class C appraisal	258
Fig. 16.1	Dashboard views in Planview Enterprise	275
Fig. 16.2	Planview Process Builder	276
Fig. 16.3	IBM Rational DOORS tool	278
Fig. 16.4	IBM Rational Software Modeler	279
Fig. 16.5	Sparx Enterprise Architect	281
Fig. 16.6	LDRA code coverage analysis report	283
Fig. 16.7	HP Quality Center	285
Fig. 17.1	Deterministic finite state machine	305
Fig. 18.1	Specification of positive square root	312
Fig. 18.2	Specification of a library system	313
Fig. 18.3	Specification of borrow operation	313
Fig. 18.4	Specification of vending machine using bags	320
Fig. 18.5	Schema inclusion	321
Fig. 18.6	Merging schemas ($S_1 \vee S_2$)	321
Fig. 18.7	Schema composition	322
Fig. 18.8	Refinement commuting diagram	323
Fig. 19.1	Simple object diagram	332
Fig. 19.2	Use-Case diagram of ATM machine	333
Fig. 19.3	UML sequence diagram	333
Fig. 19.4	UML activity diagram	334
Fig. 19.5	Rational unified process	336
Fig. 19.6	Phases and workflows in rational unified process	337

List of Tables

Table 1.1	ISO 9126-Quality characteristics	6
Table 1.2	Shewhart cycle	9
Table 1.3	Deming-14 step programme	11
Table 1.4	Deming – Five deadly diseases	12
Table 1.5	Juran’s ten step programme	14
Table 1.6	Juran’s breakthrough and control	15
Table 1.7	Crosby’s 14 step programme	16
Table 1.8	Crosby’s maturity grid	17
Table 1.9	Cost of quality categories	24
Table 1.10	Sample customer satisfaction questionnaire	28
Table 1.11	Total quality management	30
Table 3.1	Estimation techniques	63
Table 3.2	Example work-breakdown structure	64
Table 3.3	Sample project management checklist	67
Table 3.4	Risk management activities	68
Table 3.5	Activities in managing issues and change requests	71
Table 3.6	Project board roles and responsibilities	73
Table 3.7	Key processes in Prince 2	75
Table 4.1	Symptoms of poor requirements development and management	79
Table 4.2	Characteristics of good requirements	79
Table 4.3	Managing change requests	83
Table 4.4	Sample trace matrix	83
Table 4.5	Views of system architecture	86
Table 5.1	Features of good configuration management	90
Table 5.2	Symptoms of poor configuration management	91
Table 5.3	Software configuration management activities	91
Table 5.4	Build plan	92
Table 5.5	CMMI requirements for configuration management	92
Table 5.6	Sample configuration management audit checklist	98
Table 6.1	Informal review	104
Table 6.2	Structured walkthroughs	105

Table 6.3	Activities for semi-formal review meeting	106
Table 6.4	Overview Fagan inspection process	108
Table 6.5	Strict Fagan inspection guidelines	110
Table 6.6	Tailored (Relaxed) Fagan inspection guidelines	110
Table 6.7	Inspector roles	111
Table 6.8	Fagan entry criteria	111
Table 6.9	Inspection meeting	112
Table 6.10	Fagan exit criteria	114
Table 6.11	Issue severity	114
Table 6.12	Classification of defects in Fagan inspections	115
Table 6.13	Classification of ODC defect types	115
Table 7.1	Types of testing	123
Table 7.2	Sample test schedule	126
Table 8.1	Supplier selection and management	136
Table 9.1	Auditing activities	144
Table 9.2	Sample auditing checklist	147
Table 9.3	Sample audit report	148
Table 10.1	BSC objectives and measures for IT service organization	156
Table 10.2	Implementing metrics	173
Table 10.3	Identifying data to be gathered	173
Table 10.4	Phase containment effectiveness	174
Table 11.1	Motivation for ISO 9000 implementation	186
Table 11.2	ISO 9001 clauses	187
Table 11.3	ISO 9001 implementation	196
Table 11.4	Simple ISO 9000 self-assessment	197
Table 12.1	Benefits of software process improvement (CMMI)	204
Table 13.1	Motivation for CMMI implementation	215
Table 13.2	Benefits of CMMI implementation	217
Table 13.3	CMMI maturity levels	219
Table 13.4	CMMI capability levels for continuous representation	221
Table 13.5	CMMI process categories	223
Table 13.6	CMMI process areas	224
Table 13.7	CMMI generic practices	228
Table 13.8	Implementation of generic practices	230
Table 14.1	Continuous improvement cycle	237
Table 14.2	CMMI improvement structure and teams	238
Table 15.1	Phases in a SCAMPI appraisal	255
Table 15.2	Indicators of practice implementation	260
Table 16.1	Tool evaluation table	272
Table 16.2	Key capabilities of Planview Enterprise	275

Table 16.3	Tools for requirements development and management	276
Table 16.4	Tools for software design	279
Table 16.5	Integrated development environment	282
Table 17.1	Criticisms of formal methods	291
Table 17.2	Techniques for validation of formal specification	307
Table 17.3	Why are formal methods difficult?	307
Table 17.4	Characteristics of a usable formal method	308
Table 18.1	Schema composition	322
Table 19.1	Classification of UML things	329
Table 19.2	UML diagrams	330
Table 19.3	Simple class diagram	331
Table 19.4	Advantages of UML	335