This book introduces Software Quality Assurance (SQA) and provides an overview of standards used to implement SQA. It defines ways to assess the effectiveness of how one approaches software quality across key industry sectors such as telecommunications, transport, defense, and aerospace.

The book illustrates how basic concepts of software quality assurance can be utilized in businesses, government agencies, and small organizations that develop software products in many areas such as telecommunications, transport, defense, and aerospace. At the same time, it provides an overview of standards used to implement software quality assurance and defines ways to maximize performance. Topics covered include quality culture and requirements, software engineering standards and models, software review and audit, verification and validation, software configuration management, measurement, risk management, and more. In addition, this book:

- Applies ISO and IEEE software standards as well as the Capability Maturity Model Integration (CMMI)
- Illustrates the application of software quality assurance practices through the use of practical examples, quotes from experts, and tips from the authors
- Includes supplementary website with an instructor’s guide and solutions

Claude Y. Laporte, PhD, has coordinated the development, implementation, and deployment of systems and software engineering processes and project management processes, and has trained software engineers in America, Europe, and Asia. Since 2000, he has been a professor at the École de technologie supérieure (ÉTS), a Canadian engineering school, where he teaches software engineering. In 2013, Professor Laporte was awarded an honorary doctorate for his contributions to software engineering. He is the Project Editor of the set of ISO/IEC 29110 systems and software engineering life cycle standards and guides developed specifically for Very Small Entities (VSEs). He has also written two French software engineering textbooks with Dr. April. Dr. Laporte is a co-author of another book targeted at managers of small systems engineering organizations.

Alain April, PhD, is a full professor of software engineering and IT at ÉTS University, Québec, Canada. He specializes in software quality assurance and IT process mapping/conformity in the industry transfer of Big Data HPC applications based on Spark, H2O.ai, and other cloud computing technologies applied to healthcare, construction, banking, and financial industries. Professor April has been developing healthcare HPC applications in the area of genomic visualization, genotyping sequencing, and whole genome sequencing, extending Berkeley’s Adam data structure for HPC. These applied research projects deploy large-scale machine learning algorithms in research hospitals for specific use cases, such as type 2 diabetes early prediction and leukemia treatments in children.
IEEE Press Editorial Board
Tariq Samad, Editor in Chief

Giancarlo Fortino  Xiaou Li  Ray Perez
Dmitry Goldgof  Andreas Molisch  Linda Shafer
Don Heirman  Saeid Nahavandi  Mohammad Shahidehpour
Ekram Hossain  Jeffrey Nanzer  Zidong Wang

About IEEE Computer Society
IEEE Computer Society is the world’s leading computing membership organization and the trusted information and career-development source for a global workforce of technology leaders including: professors, researchers, software engineers, IT professionals, employers, and students. The unmatched source for technology information, inspiration, and collaboration, the IEEE Computer Society is the source that computing professionals trust to provide high-quality, state-of-the-art information on an on-demand basis. The Computer Society provides a wide range of forums for top minds to come together, including technical conferences, publications, and a comprehensive digital library, unique training webinars, professional training, and the TechLeader Training Partner Program to help organizations increase their staff’s technical knowledge and expertise, as well as the personalized information tool myComputer. To find out more about the community for technology leaders, visit http://www.computer.org.

IEEE/Wiley Partnership
The IEEE Computer Society and Wiley partnership allows the CS Press authored book program to produce a number of exciting new titles in areas of computer science, computing, and networking with a special focus on software engineering. IEEE Computer Society members continue to receive a 15% discount on these titles when purchased through Wiley or at wiley.com/ieeecs.

To submit questions about the program or send proposals, please contact Mary Hatcher, Editor, Wiley-IEEE Press: Email: mhatcher@wiley.com, Telephone: 201-748-6903, John Wiley & Sons, Inc., 111 River Street, MS 8-01, Hoboken, NJ 07030-5774.
Software Quality Assurance

Claude Y. Laporte
Alain April
Contents

Preface xiv
Acknowledgments xxi

1. Software Quality Fundamentals 1
   1.1 Introduction 1
   1.2 Defining Software Quality 2
   1.3 Software Errors, Defects, and Failures 4
       1.3.1 Problems with Defining Requirements 10
       1.3.2 Maintaining Effective Communications Between Client and Developer 13
       1.3.3 Deviations from Specifications 14
       1.3.4 Architecture and Design Errors 15
       1.3.5 Coding Errors 15
       1.3.6 Non-Compliance with Current Processes/Procedures 16
       1.3.7 Inadequate Reviews and Tests 17
       1.3.8 Documentation Errors 17
   1.4 Software Quality 19
   1.5 Software Quality Assurance 20
   1.6 Business Models and the Choice of Software Engineering Practices 22
       1.6.1 Description of the Context 23
       1.6.2 Anxiety and Fear 24
       1.6.3 Choice of Software Practices 25
       1.6.4 Business Model Descriptions 25
       1.6.5 Description of Generic Situational Factors 26
       1.6.6 Detailed Description of Each Business Model 27
   1.7 Success Factors 32
   1.8 Further Reading 33
   1.9 Exercises 34

2. Quality Culture 35
   2.1 Introduction 35
   2.2 Cost of Quality 39
   2.3 Quality Culture 49
   2.4 The Five Dimensions of a Software Project 53
vi Contents

2.5 The Software Engineering Code of Ethics 56
  2.5.1 Abridged Version: Preamble 58
  2.5.2 The Example of the Code of Ethics of the Ordre des ingénieurs du Québec 60
  2.5.3 Whistle Blowers 61

2.6 Success Factors 62
2.7 Further Reading 63
2.8 Exercises 63

3. Software Quality Requirements 66

3.1 Introduction 66
3.2 Software Quality Models 69
  3.2.1 Initial Model Proposed by McCall 71
  3.2.2 The First Standardized Model: IEEE 1061 73
  3.2.3 Current Standardized Model: ISO 25000 Set of Standards 77
3.3 Definition of Software Quality Requirements 86
  3.3.1 Specifying Quality Requirements: The Process 91
3.4 Requirement Traceability During the Software Life Cycle 95
3.5 Software Quality Requirements and the Software Quality Plan 95
3.6 Success Factors 96
3.7 Further Reading 97
3.8 Exercises 97

4. Software Engineering Standards and Models 101

4.1 Introduction 101
4.2 Standards, Cost of Quality, and Business Models 108
4.3 Main Standards for Quality Management 109
  4.3.1 ISO 9000 Family 109
  4.3.2 ISO/IEC 90003 Standard 115
4.4 ISO/IEC/IEEE 12207 Standard 117
  4.4.1 Limitations of the ISO 12207 Standard 121
4.5 ISO/IEC/IEEE 15289 Standard for the Description of Information Elements 121
4.6 IEEE 730 Standard for SQA Processes 123
  4.6.1 Activities and Tasks of SQA 125
4.7 Other Quality Models, Standards, References, and Processes 129
  4.7.1 Process Maturity Models of the SEI 130
  4.7.2 Software Maintenance Maturity Model (S^mm) 135
  4.7.3 ITIL Framework and ISO/IEC 20000 138
  4.7.4 CobiT Process 142
  4.7.5 ISO/IEC 27000 Family of Standards for Information Security 143
  4.7.6 ISO/IEC 29110 Standards and Guides for Very Small Entities 144
  4.7.7 ISO/IEC 29110 Standards for VSEs Developing Systems 155
4.8 Specific Standards for an Application Domain 156
4.8.1 DO-178 and ED-12 Guidance for Airborne Systems 156
4.8.2 EN 50128 Standard for Railway Applications 159
4.8.3 ISO 13485 Standard for Medical Devices 161
4.9 Standards and the SQAP 163
4.10 Success Factors 165
4.11 Further Reading 165
4.12 Exercises 166

5. Reviews 167
5.1 Introduction 167
5.2 Personal Review and Desk-Check Review 172
5.2.1 Personal Review 172
5.2.2 Desk-Check Reviews 175
5.3 Standards and Models 179
5.3.1 ISO/IEC 20246 Software and Systems Engineering: Work Product Reviews 179
5.3.2 Capability Maturity Model Integration 180
5.3.3 The IEEE 1028 Standard 181
5.4 Walk-Through 184
5.4.1 Usefulness of a Walk-Through 184
5.4.2 Identification of Roles and Responsibilities 186
5.5 Inspection Review 187
5.6 Project Launch Reviews and Project Assessments 189
5.6.1 Project Launch Review 190
5.6.2 Project Retrospectives 192
5.7 Agile Meetings 197
5.8 Measures 199
5.9 Selecting the Type of Review 202
5.10 Reviews and Business Models 205
5.11 Software Quality Assurance Plan 205
5.12 Success Factors 206
5.13 Tools 208
5.14 Further Reading 208
5.15 Exercises 208

6. Software Audits 210
6.1 Introduction 210
6.2 Types of Audits 215
6.2.1 Internal Audit 215
6.2.2 Second-Party Audit 215
6.2.3 Third-Party Audit 217
Contents

6.3 Audit and Software Problem Resolution According to ISO/IEC/IEEE 12207 217
   6.3.1 Project Assessment and Control Process 218
   6.3.2 Decision Management Process 218

6.4 Audit According to the IEEE 1028 Standard 218
   6.4.1 Roles and Responsibilities 220
   6.4.2 IEEE 1028 Audit Clause 221
   6.4.3 Audit Conducted According to IEEE 1028 222

6.5 Audit Process and the ISO 9001 Standard 225
   6.5.1 Steps of a Software Audit 226

6.6 Audit According to the CMMI 230
   6.6.1 SCAMPI Assessment Method 231

6.7 Corrective Actions 233
   6.7.1 Corrective Actions Process 234

6.8 Audits for Very Small Entities 238

6.9 Audit and the SQA Plan 239

6.10 Presentation of an Audit Case Study 241

6.11 Success Factors 246

6.12 Further Reading 247

6.13 Exercises 247

7. Verification and Validation 249

7.1 Introduction 249

7.2 Benefits and Costs of V&V 255
   7.2.1 V&V and the Business Models 257

7.3 V&V Standards and Process Models 257
   7.3.1 IEEE 1012 V&V Standard 258
   7.3.2 Integrity Levels 260
   7.3.3 Recommended V&V Activities for Software Requirements 262

7.4 V&V According to ISO/IEC/IEEE 12207 263
   7.4.1 Verification Process 265
   7.4.2 Validation Process 265

7.5 V&V According to the CMMI Model 266

7.6 ISO/IEC 29110 and V&V 267

7.7 Independent V&V 268
   7.7.1 IV&V Advantages with Regards to SQA 271

7.8 Traceability 271
   7.8.1 Traceability Matrix 273
   7.8.2 Implementing Traceability 276

7.9 Validation Phase of Software Development 277
   7.9.1 Validation Plan 279

7.10 Tests 281
# Contents

7.11 Checklists 282  
  7.11.1 How to Develop a Checklist 283  
  7.11.2 How to Use a Checklist 285  
  7.11.3 How to Improve and Manage a Checklist 286  
7.12 V&V Techniques 287  
  7.12.1 Introduction to V&V Techniques 287  
  7.12.2 Some V&V Techniques 288  
7.13 V&V Plan 289  
7.14 Limitations of V&V 290  
7.15 V&V in the SQA Plan 291  
7.16 Success Factors 292  
7.17 Further Reading 293  
7.18 Exercises 293  

8. Software Configuration Management 295  

8.1 Introduction 295  
8.2 Software Configuration Management 296  
8.3 Benefits of Good Configuration Management 297  
  8.3.1 CM According to ISO 12207 298  
  8.3.2 CM According to IEEE 828 299  
  8.3.3 CM According to the CMMI 299  
8.4 SCM Activities 301  
  8.4.1 Organizational Context of SCM 301  
  8.4.2 Developing a SCM Plan 302  
  8.4.3 Identification of CI to be Controlled 303  
8.5 Baselines 309  
8.6 Software Repository and Its Branches 311  
  8.6.1 A Simple Branching Strategy 315  
  8.6.2 A Typical Branching Strategy 316  
8.7 Configuration Control 318  
  8.7.1 Requests, Evaluation, and Approval of Changes 319  
  8.7.2 Configuration Control Board 321  
  8.7.3 Request for Waivers 322  
  8.7.4 Change Management Policy 322  
8.8 Configuration Status Accounting 323  
  8.8.1 Information Concerning the Status of CI 323  
  8.8.2 Configuration Item Status Reporting 325  
8.9 Software Configuration Audit 325  
  8.9.1 Functional Configuration Audit 327  
  8.9.2 Physical Configuration Audit 327  
  8.9.3 Audits Performed During a Project 328  
8.10 Implementing SCM in Very Small Entities with ISO/IEC 29110 329  
8.11 SCM and the SQAP 330
# Contents

8.12 Success Factors 331  
8.13 Further Reading 333  
8.14 Exercises 333

9. Policies, Processes, and Procedures 335  
9.1 Introduction 335  
9.1.1 Standards, the Cost of Quality, and Business Models 341  
9.2 Policies 341  
9.3 Processes 345  
9.4 Procedures 351  
9.5 Organizational Standards 352  
9.6 Graphical Representation of Processes and Procedures 353  
9.6.1 Some Pitfalls to Avoid 356  
9.6.2 Process Mapping 357  
9.6.3 ETVX Process Notation 357  
9.6.4 IDEF Notation 366  
9.6.5 BPMN Notation 370  
9.7 Process Notation of ISO/IEC 29110 376  
9.8 Case Study 383  
9.9 Personal Improvement Process 388  
9.10 Policies, Processes, and Procedures in the SQA Plan 393  
9.11 Success Factors 394  
9.12 Further Reading 395  
9.13 Exercises 396

10. Measurement 397  
10.1 Introduction—the Importance of Measurement 397  
10.1.1 Standards, the Cost of Quality, and Software Business Models 401  
10.2 Software Measurement According to ISO/IEC/IEEE 12207 402  
10.3 Measurement According to ISO 9001 403  
10.4 The Practical Software and Systems Measurement Method 404  
10.5 ISO/IEC/IEEE 15939 Standard 411  
10.5.1 Measurement Process According to ISO 15939 412  
10.5.2 Activities and Tasks of the Measurement Process 412  
10.5.3 An Information Measurement Model of ISO 15939 412  
10.6 Measurement According to the CMMI Model 418  
10.7 Measurement in Very Small Entities 421  
10.8 The Survey as a Measurement Tool 421
Contents

10.9 Implementing a Measurement Program 425
  10.9.1 Step 1: Management Commitment Build-Up 426
  10.9.2 Step 2: Staff Commitment Build-Up 427
  10.9.3 Step 3: Selection of Key Processes to be Improved 427
  10.9.4 Step 4: Identification of the Goals and Objectives Related to the Key Process 427
  10.9.5 Step 5: Design of the Measurement Program 427
  10.9.6 Step 6: Description of the Information System to Support Measurement 428
  10.9.7 Step 7: Deployment of the Measurement Program 428

10.10 Practical Considerations 430
  10.10.1 Some Pitfalls with Regards to Measurement 432

10.11 The Human Side of Measurement 435
  10.11.1 Cost of Measurement 438

10.12 Measurement and the IEEE 730 SQAP 439
  10.12.1 Software Process Measurement 440
  10.12.2 Software Product Measurement 441

10.13 Success Factors 443

10.14 Further Reading 443

10.15 Exercises 444

11. Risk Management 445

11.1 Introduction 445
  11.1.1 Risk, the Cost of Quality and Business Models 451
  11.1.2 Costs and Benefits of Risk Management 453

11.2 Risk Management According to Standards and Models 454
  11.2.1 Risk Management According to ISO 9001 454
  11.2.2 Risk Management According to ISO/IEC/IEEE 12207 455
  11.2.3 Risk Management According to ISO/IEC/IEEE 16085 456
  11.2.4 Risk Management According to the CMMI Model 459
  11.2.5 Risk Management According to PMBOK® Guide 461
  11.2.6 Risk Management According to ISO 29110 462
  11.2.7 Risk Management and the SQA According to IEEE 730 465

11.3 Practical Considerations for Risk Management 466
  11.3.1 Risk Evaluation Step 468
  11.3.2 Risk Control Step 474
  11.3.3 Lessons Learned Activity 477

11.4 Risk Management Roles 478

11.5 Measurement and Risk Management 479

11.6 Human Factors and Risk Management 483

11.7 Success Factors 485

11.8 Conclusion 486

11.9 Further Reading 487

11.10 Exercises 487
12. Supplier Management and Agreements 489

12.1 Introduction 489
12.2 Supplier Requirements of ISO 9001 490
12.3 Agreement Processes of ISO 12207 491
12.4 Supplier Agreement Management According to the CMMI 494
12.5 Managing Suppliers 496
12.6 Software Acquisition Life Cycle 497
12.7 Software Contract Types 499
  12.7.1 Fixed Price Contract 501
  12.7.2 Cost plus Percentage of Cost 502
  12.7.3 Cost plus Fixed Fee 502
  12.7.4 Risk Sharing 502
12.8 Software Contract Reviews 505
  12.8.1 Two Reviews: Initial and Final 505
  12.8.2 Initial Contract Review 506
  12.8.3 Final Contract Review 509
12.9 Supplier and Acquirer Relationship and the SQAP 510
12.10 Success Factors 511
12.11 Further Reading 512
12.12 Exercises 512

13. Software Quality Assurance Plan 514

13.1 Introduction 514
13.2 SQA Planning 518
  13.2.1 Purpose and Scope 518
  13.2.2 Definitions and Acronyms 518
  13.2.3 Reference Documents 519
  13.2.4 SQAP Overview—Organization and Independence 520
  13.2.5 SQAP Overview—Software Product Risk 524
  13.2.6 SQAP Overview—Tools 525
  13.2.7 SQAP Overview—Standards, Practices, and Conventions 525
  13.2.8 SQAP Overview—Effort, Resources, and Schedule 526
  13.2.9 Activities, Outcomes, and Tasks—Product Assurance 528
  13.2.10 Activities, Outcomes, and Tasks—Process Assurance 529
  13.2.11 Additional Considerations 531
  13.2.12 SQA Records 536
13.3 Executing the SQAP 537
13.4 Conclusion 539
13.5 Further Reading 539
13.6 Exercises 540
Contents

Appendix 1. Software Engineering Code of Ethics and Professional Practice (Version 5.2) 541

Appendix 2. Incidents and Horror Stories Involving Software 549

Glossary – Abbreviations – Acronyms 555
References 576
Index 591
Preface

This book addresses the global challenge of the improvement of software quality. It seeks to provide an overview of software quality assurance (SQA) practices for customers, managers, auditors, suppliers, and personnel responsible for software projects, development, maintenance, and software services.

In a globally competitive environment, clients and competitors exert a great deal of pressure on organizations. Clients are increasingly demanding and require, among other things, software that is of high quality, low cost, delivered quickly, and with impeccable after-sales support. To meet the demand, quality, and deadlines, the organization must use efficient quality assurance practices for their software activities.

Ensuring software quality is not an easy task. Standards define ways to maximize performance but managers and employees are largely left to themselves to decide how to practically improve the situation. They face several problems:

- increasing pressure to deliver quality products quickly;
- increasing size and complexity of software and of systems;
- increasing requirements to meet national, international, and professional standards;
- subcontracting and outsourcing;
- distributed work teams; and
- ever changing platforms and technologies.

We will focus on the issue of SQA in industry and in public organizations. Industry and public organizations do not have access to a complete and integrated reference (i.e., one book) that can help them with assessing and improving activities specific to SQA. The SQA department must meet service standards for its customers, the technical criteria of the field, and maximize strategic and economic impacts.

The purpose of this book is to enable managers, clients, suppliers, developers, auditors, software maintainers, and SQA personnel to use this information to assess
the effectiveness and completeness of their approach to SQA. Some of the issues raised here include:

- What are the processes, practices, and activities of SQA and software improvement?
- Can the current standards and models serve as a reference?
- How do we ensure that managers and their staff understand the value of SQA activities and their implementation?

To answer these questions, we drew upon over 30 years of practical experience in software engineering and SQA in different organizations such as telecom, banking, defense, and transportation. This industry experience has convinced us of the importance of supporting the presentation of concepts and theory with references and practical examples. We have illustrated the correct and effective implementation of numerous quality assurance practices with real case studies throughout the book.

In many organizations, SQA is a synonym for testing. SQA, as presented in this book, covers a large spectrum of proven practices to provide a level of confidence that quality in software development and maintenance activities is independent of the life cycle selected by an organization or a project.

In this book, we will extensively use the term “software quality assurance” and the acronym SQA. As defined in the IEEE Standard for Software Quality Assurance Processes, IEEE 730-2014, a function is a set of resources and activities that achieve a particular purpose [IEE 14]. The SQA function can be executed by a software project team member. It could also be executed by an independent party (e.g., within a quality assurance (QA) department responsible for hardware, software, and supplier quality).

STRUCTURE AND ORGANIZATION OF THIS BOOK

The book is divided into 13 chapters that cover the basic knowledge of SQA as identified, among others, by the IEEE 730 Standard for SQA Processes of the Institute of Electrical and Electronics Engineers (IEEE), the ISO/IEC/IEEE 12207 software life cycle processes standard, the Capability Maturity Model® Integration for Development (CMMI®-DEV) developed by the Software Engineering Institute as well as the ISO Guide to the Software Engineering Body of Knowledge (SWEBOK®). Numerous practical examples are used to illustrate the application of SQA practices.

CHAPTER 1: SOFTWARE QUALITY FUNDAMENTALS

This chapter presents an overview of the knowledge required by SQA practitioners. From this overview, the book develops every aspect of the field and cites the important references that deepen each specific topic. We use the concept of business models to
explain the significant differences in the selection of SQA practices. In this chapter, we also establish terms and their definitions as well as useful concepts that are used throughout the book.

CHAPTER 2: QUALITY CULTURE

This chapter introduces the concept of cost of quality, followed by practical examples. It also introduces the concept of quality culture and its influence on the SQA practices used. We also present five dimensions of a software project and how these dimensions can be used to identify the degrees of freedom a project manager has to ensure its success. In this chapter, we present an overview of software engineering ethics and the techniques to manage the expectations of managers and customers with respect to software quality.

CHAPTER 3: SOFTWARE QUALITY REQUIREMENTS

This chapter adds to the concepts and terminology already presented. It deals with software quality models as well as ISO standards on software quality models. These models propose classifications of software quality requirements and steps to define them. Practical examples describe how to use these models to define the quality requirements of a software project. Finally, we introduce the concept of requirements traceability and the importance of quality requirements for the SQA plan.

CHAPTER 4: SOFTWARE ENGINEERING STANDARDS AND MODELS

This chapter presents the most important international standards of ISO and models about software quality, such as the CMMI® developed by the Software Engineering Institute. A new ISO standard for very small organizations is also presented. The SQA practitioner and specialist will find proven practices from standards and models. This chapter provides the framework that can be useful for the following major software activities: (1) development, (2) maintenance, and (3) IT services. Finally, a short discussion on the standards specific to certain domains of application is presented, followed by recommendations for a SQA plan.

CHAPTER 5: REVIEWS

This chapter presents different types of software reviews: personal review, the “desk check,” the walk-through, and the inspection. We describe the theory about reviews and then provide practical examples. It introduces reviews in an agile context. Subsequently, we describe other reviews specific to a project: the project launch review
and lessons learned review. The chapter concludes with a discussion on the selection of one type of review depending on your business domain and how these techniques fit into the SQA plan.

CHAPTER 6: SOFTWARE AUDITS

This chapter describes the audit process and the software problem resolution process. Sooner or later in the career of a software practitioner, audits will be conducted in a software project. Standards and models describing audits are presented followed by a practical case. The chapter concludes with a discussion of the role of audits in the SQA plan.

CHAPTER 7: VERIFICATION AND VALIDATION

This chapter describes the concept of software verification and validation (V&V). It describes its benefits as well as the costs of using V&V practices. Then, the standards and models that impose or describe V&V practices for a project are described. Finally, the description of the contents of a V&V plan is presented.

CHAPTER 8: SOFTWARE CONFIGURATION MANAGEMENT

This chapter describes an important component of software quality: software configuration management (SCM). The chapter begins by presenting the usefulness of SCM and typical SCM activities. It presents repositories and branching techniques involved in source code management, as well as the concepts of software control, software status, and software audits. Finally, this chapter concludes with a proposal for the implementation of SCM in a small organization and ends with a discussion of the role of SCM in the SQA plan.

CHAPTER 9: POLICIES, PROCESSES, AND PROCEDURES

This chapter explains how to develop, document, and improve policies, processes, and procedures to ensure the effectiveness and efficiency of the software organization. It explains the importance of documentation presenting a few notations, as examples, to document processes and procedures. The chapter ends by presenting the Personal Software Process (PSP) developed by the Software Engineering Institute to ensure individuals have a disciplined and structured approach to software development that enables them to significantly increase the quality of their software products.
CHAPTER 10: MEASUREMENT

This chapter explains the importance of measurement, standards, and models, and presents a methodology to describe the requirements for a measurement process. It presents how measurement can be used by small organizations and small projects. Then, an approach to implement a measurement program, to detect the potential pitfalls, and the potential impact of human factors, when measuring, is discussed. The chapter concludes with a discussion of the role of measurement in a SQA plan.

CHAPTER 11: RISK MANAGEMENT

This chapter presents the main models and standards that include requirements for the management of risks. It discusses the risks that may affect the quality of software and techniques to identify, prioritize, document, and mitigate them. It also presents the roles of stakeholders in the risk management process and discusses the human factors to consider in the management of software risks. The chapter concludes with a discussion on the critical role of risk in the development of a SQA plan.

CHAPTER 12: SUPPLIER MANAGEMENT AND AGREEMENTS

This chapter deals with the important topic of supplier management and agreements. It discusses the major reviews and recommendations of the CMMI®. Subsequently, it lists the different types of software agreements and the benefits of the risk sharing agreement are illustrated using a practical example. This chapter concludes with recommendations for the content of the SQA plan when suppliers are involved.

CHAPTER 13: SOFTWARE QUALITY ASSURANCE PLAN

This chapter summarizes the topics presented in the whole book by using the concepts presented in each chapter to assemble a comprehensive SQA plan that conforms to the IEEE 730 recommendation. It ends by presenting additional recommendations and practical examples.

APPENDICES

Appendix 1 – Software Engineering Code of Ethics and Professional Practice (Version 5.2)
Appendix 2 – Incidents and Horror Stories involving Software
ICONS USED IN THE BOOK

Different icons are used throughout this book to illustrate a concept with a practical example; to focus on a definition; to present an anecdote, a tool, or checklist; or simply to provide a quote or a website. Consult the table below for the meaning of each icon.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="example_icon" alt="Practical example" /></td>
<td>Practical example: An example of the practical application of a theoretical concept</td>
</tr>
<tr>
<td><img src="quote_icon" alt="Quote" /></td>
<td>Quote: A quote from an expert</td>
</tr>
<tr>
<td><img src="definition_icon" alt="Definition" /></td>
<td>Definition: A definition of an important term</td>
</tr>
<tr>
<td><img src="web_icon" alt="Reference on the Web" /></td>
<td>Reference on the Web: An internet site to learn more about a specific topic</td>
</tr>
<tr>
<td><img src="tools_icon" alt="Tools" /></td>
<td>Tools: Examples of tools that support the techniques presented</td>
</tr>
<tr>
<td><img src="anecdote_icon" alt="Anecdote" /></td>
<td>Anecdote: A short story of a little known fact, or a curious point on the subject discussed</td>
</tr>
<tr>
<td><img src="checklist_icon" alt="Checklist" /></td>
<td>Checklist: A list of items to check, or not to be forgotten, during the execution of a presented technique</td>
</tr>
<tr>
<td><img src="tip_icon" alt="Tip" /></td>
<td>Tip: A tip from the authors or from another professional</td>
</tr>
</tbody>
</table>

WEBSITE

Supplementary material for teaching as well as for use in organizations (e.g., presentation material, solutions, project descriptions, templates, tools, articles, and links) is available on the website: www.sqabook.org.

Given that international standards are updated on a regular basis, the website will also highlight the latest developments that contribute to SQA practices.

EXERCISES

Each chapter contains exercises. Solutions are available on the website.

NOTES

Many software engineering standards from ISO and IEEE have been cited in this book. These standards are updated on a regular basis, typically every five years, to reflect evolving software engineering practices. The accompanying website,
Preface

www.sqabook.org, contains complementary information as well as the latest developments that impact or contribute to SQA practices described in each chapter and will evolve over time.

Since software engineering standards can be cited in an agreement between a customer and a supplier and add additional legal requirements to the agreement, we have not paraphrased the text of standards in our book, we have directly quoted the text from the standards.
Acknowledgments

We would like to thank Professor Normand Séguin of the University of Quebec in Montreal (UQAM), Mr. Jean-Marc Desharnais for allowing us to use an excerpt that describes the implementation process of a measurement program, and many graduate students of the Masters in Software Engineering from the École de technologie supérieure (ÉTS) who reviewed the chapters of this book and contributed through their vast industry experience, analogies, and case studies to enrich the content.

We are also very grateful to Kathy Iberle for letting us use her description of business models and their application in different business domains [IBE02, IBE03]. The business models are very helpful in understanding the risks facing a specific business domain as well as the breadth and depth of software engineering practices used to mitigate the risks. Finally, we would like to thank Karl Wiegers and Daniel Galin for allowing us to use figures from their books.