

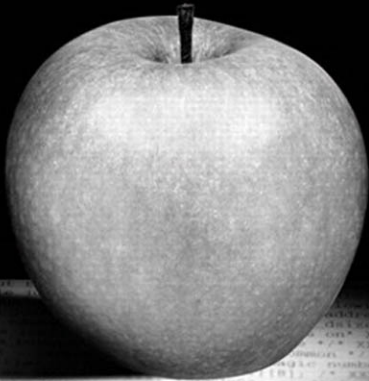
Effective SOFTWARE DEVELOPMENT SERIES

Scott Meyers, Consulting Editor



CODE *Quality*

The Open Source Perspective



Diomidis Spinellis

Code Quality

Code Quality: The Open Source Perspective

Table of Contents

Contents

List of Tables

List of Figures

Foreword

Preface

1 Introduction

1.1 Software Quality

1.1.1 Quality Through the Eyes of the User, the Builder, and the Manager

1.1.2 Quality Attributes

1.1.3 A World of Tensions

1.2 How to Read This Book

1.2.1 Typographical Conventions

1.2.2 Diagrams

1.2.3 Charts

1.2.4 Assembly Code

1.2.5 Exercises

1.2.6 Supplementary Material

1.2.7 Tools

2 Reliability

2.1 Input Problems

2.2 Output Problems

Table of Contents

2.2.1 Incomplete or Missing Output

2.2.2 Correct Results at the Wrong Time

2.2.3 Wrong Format

2.3 Logic Problems

2.3.1 Off-by-One Errors and Loop Iterations

2.3.2 Neglected Extreme Conditions

2.3.3 Forgotten Cases, Condition Tests, or Steps

2.3.4 Missing Methods

2.3.5 Unnecessary Functionality

2.3.6 Misinterpretation

2.4 Computation Problems

2.4.1 Incorrect Algorithm or Computation

2.4.2 Incorrect Operand in an Expression

2.4.3 Incorrect Operator in an Expression

2.4.4 Operator Precedence Problems

2.4.5 Overflow, Underflow, and Sign Conversion-Errors

2.5 Concurrency and Timing Problems

2.6 Interface Problems

2.6.1 Incorrect Routine or Arguments

2.6.2 Failure to Test a Return Value

2.6.3 Missing Error Detection or Recovery

2.6.4 Resource Leaks

2.6.5 Misuse of Object-Oriented Facilities

2.7 Data-Handling Problems

2.7.1 Incorrect Data Initialization

2.7.2 Referencing the Wrong Data Variable

2.7.3 Out-of-Bounds References

2.7.4 Incorrect Subscripting

2.7.5 Incorrect Scaling or Data Units

Table of Contents

2.7.6 Incorrect Data Packing or Unpacking

2.7.7 Inconsistent Data

2.8 Fault Tolerance

2.8.1 Management Strategy

2.8.2 Redundancy in Space

2.8.3 Redundancy in Time

2.8.4 Recoverability

3 Security

3.1 Vulnerable Code

3.2 The Buffer Overflow

3.3 Race Conditions

3.4 Problematic APIs

3.4.1 Functions Susceptible to Buffer Overflows

3.4.2 Format String Vulnerabilities

3.4.3 Path and Shell Metacharacter Vulnerabilities

3.4.4 Temporary Files

3.4.5 Functions Unsuitable for Cryptographic Use

3.4.6 Forgeable Data

3.5 Untrusted Input

3.6 Result Verification

3.7 Data and Privilege Leakage

3.7.1 Data Leakage

3.7.2 Privilege Leakage

3.7.3 The Java Approach

3.7.4 Isolating Privileged Code

3.8 Trojan Horse

3.9 Tools

4 Time Performance

Table of Contents

4.1 Measurement Techniques

4.1.1 Workload Characterization

4.1.2 I/O-Bound Tasks

4.1.3 Kernel-Bound Tasks

4.1.4 CPU-Bound Tasks and Profiling Tools

4.2 Algorithm Complexity

4.3 Stand-Alone Code

4.4 Interacting with the Operating System

4.5 Interacting with Peripherals

4.6 Involuntary Interactions

4.7 Caching

4.7.1 A Simple System Call Cache

4.7.2 Replacement Strategies

4.7.3 Precomputing Results

5 Space Performance

5.1 Data

5.1.1 Basic Data Types

5.1.2 Aggregate Data Types

5.1.3 Alignment

5.1.4 Objects

5.2 Memory Organization

5.3 Memory Hierarchies

5.3.1 Main Memory and Its Caches

5.3.2 Disk Cache and Banked Memory

5.3.3 Swap Area and File-Based Disk Storage

5.4 The Process/Operating System Interface

5.4.1 Memory Allocation

5.4.2 Memory Mapping

Table of Contents

5.4.3 Data Mapping

5.4.4 Code Mapping

5.4.5 Accessing Hardware Resources

5.4.6 Interprocess Communication

5.5 Heap Memory Management

5.5.1 Heap Fragmentation

5.5.2 Heap Profiling

5.5.3 Memory Leaks

5.5.4 Garbage Collection

5.6 Stack Memory Management

5.6.1 The Stack Frame

5.6.2 Stack Space

5.7 Code

5.7.1 Design Time

5.7.2 Coding Time

5.7.3 Build Time

6 Portability

6.1 Operating Systems

6.2 Hardware and Processor Architectures

6.2.1 Data Type Properties

6.2.2 Data Storage

6.2.3 Machine-Specific Code

6.3 Compilers and Language Extensions

6.3.1 Compiler Bugs

6.4 Graphical User Interfaces

6.5 Internationalization and Localization

6.5.1 Character Sets

6.5.2 Locale

Table of Contents

6.5.3 Messages

7 Maintainability

7.1 Measuring Maintainability

7.1.1 The Maintainability Index

7.1.2 Metrics for Object-Oriented Programs

7.1.3 Dependency Metrics on Packages

7.2 Analyzability

7.2.1 Consistency

7.2.2 Expression Formatting

7.2.3 Statement Formatting

7.2.4 Naming Conventions

7.2.5 Statement-Level Comments

7.2.6 Versioning Comments

7.2.7 Visual Structure: Blocks and Indentation

7.2.8 Length of Expressions, Functions, and Methods

7.2.9 Control Structures

7.2.10 Boolean Expressions

7.2.11 Recognizability and Cohesion

7.2.12 Dependencies and Coupling

7.2.13 Code Block Comments

7.2.14 Data Declaration Comments

7.2.15 Appropriate Identifier Names

7.2.16 Locality of Dependencies

7.2.17 Ambiguity

7.2.18 Reviewability

7.3 Changeability

7.3.1 Identification

7.3.2 Separation

7.4 Stability

Table of Contents

- 7.4.1 Encapsulation and Data Hiding
- 7.4.2 Data Abstraction
- 7.4.3 Type Checking
- 7.4.4 Compile-Time Assertions
- 7.4.5 Runtime Checks and Inspection-Time Assertions

7.5 Testability

- 7.5.1 Unit Testing
- 7.5.2 Integration Testing
- 7.5.3 System Testing
- 7.5.4 Test Coverage Analysis
- 7.5.5 Incidental Testing

7.6 Effects of the Development Environment

- 7.6.1 Incremental Builds
- 7.6.2 Tuning Build Performance

8 Floating-Point Arithmetic

8.1 Floating-Point Representation

- 8.1.1 Measuring Error
- 8.1.2 Rounding
- 8.1.3 Memory Format
- 8.1.4 Normalization and the Implied 1-Bit
- 8.1.5 Exponent Biasing
- 8.1.6 Negative Numbers
- 8.1.7 Denormal Numbers
- 8.1.8 Special Values

8.2 Rounding

8.3 Overflow

8.4 Underflow

8.5 Cancellation

Table of Contents

8.6 Absorption

8.7 Invalid Operations

A Source Code Credits

Bibliography

Index

Author Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

Table of Contents

V

W

Y

Z