
Implementing Software Engineering Practices

FLETCHER J. BUCKLEY



WILEY

A WILEY-INTERSCIENCE PUBLICATION

JOHN WILEY & SONS

New York • Chichester • Brisbane • Toronto • Singapore

CONTENTS

List of Figures	xvii
List of Tables	xix
Preface	xxi
Acknowledgments	xxiii
1 Introduction, 1	
1.1 Purpose, 2	
1.2 Audience, 2	
1.3 The Approach to Improving The Process, 2	
1.3.1 The Metaprocess, 3	
1.3.2 Vehicles, 3	
1.4 Disclaimer, 4	
1.5 Overview of the Rest of the Text, 4	
2 The Baseline Process, 5	
2.1 Producing the First Approved Product, 5	
2.1.1 Announce Your Intentions, 5	
2.1.2 Do Your Homework, 6	
2.1.3 Establish a Framework, 7	
2.1.4 Coordinate, 7	
2.1.5 Obtain Approval, 8	

- 2.2 Implementing and Improving the Product, 8
 - 2.2.1 Help the First Implementers, 8
 - 2.2.2 Institute an Organizational Memory, 9
 - 2.2.3 Revise the Software Development Manual, 10
 - 2.2.4 Initiate Formal Training, 10
 - 2.2.5 Establish a Technical Support Group, 10

- 3 The Framework, 11**
 - 3.1 The Overall Policy, 11
 - 3.1.1 Scope of the Initial Application, 12
 - 3.1.2 Organization, 13
 - 3.2 Software Development Manual, 14
 - 3.2.1 Example Software Development Life Cycle, 14
 - 3.3 Exercise, 16

- 4 The Requirements Definition Phase, 17**
 - 4.1 What Is To Be Done, 17
 - 4.1.1 Software Requirements Specifications, 18
 - 4.1.1.1 Approaches, 18
 - 4.1.1.2 Feasibility, 18
 - 4.1.1.3 Risk Reduction, 19
 - 4.1.2 External Interface Specification, 20
 - 4.1.3 Acceptance Test Plans, 20
 - 4.1.4 Software Development Plans, 21
 - 4.1.5 Specialty Plans, 21
 - 4.1.6 Software Standards and Procedures Manual, 22
 - 4.2 Who Is To Do It, 22
 - 4.3 The Final Review, 23
 - 4.4 Example Policy, 23
 - 4.5 Cautions, 23

- 5 Establishing the Requirements, 25**
 - 5.1 Policy, 25
 - 5.2 Partitioning the Overall Job, 26
 - 5.3 Initiation of the Individual SRS, 28
 - 5.3.1 Design Constraints, 29
 - 5.3.2 Assumptions and Dependencies, 31
 - 5.3.3 Performance Requirements, 31
 - 5.3.4 Attributes, 32
 - 5.3.5 Specific Requirements, 32

- 5.3.6 Appendixes, 34
- 5.4 Supporting Methodologies, 34
- 5.5 Checking the Product, 34
- 5.6 Exercises, 35

- 6 Establishing the External Interfaces, 36**
 - 6.1 Initiation of the EIS, 36
 - 6.1.1 Initial Message Identification, 36
 - 6.1.2 Hardware Interfaces, 37
 - 6.1.3 Message Descriptions, 38
 - 6.2 Formatting Techniques, 38
 - 6.3 Document Completion, 40
 - 6.4 Support Tools, 40
 - 6.5 Exercises, 40

- 7 Planning the Acceptance Test, 41**
 - 7.1 The Software Test and Evaluation Master Plan, 42
 - 7.1.1 Informal Test, 42
 - 7.1.2 Integration, 43
 - 7.1.3 Formal Tests, 43
 - 7.2 Software Test Plans, 44
 - 7.2.1 Documentation Considerations, 45
 - 7.3 Exercises, 47

- 8 Software Development Planning, 49**
 - 8.1 Example Policy, 49
 - 8.1.1 Organizational Philosophy, 52
 - 8.1.2 Management Control, 52
 - 8.1.3 Work Breakdown Structures, 54
 - 8.1.4 Schedules, 55
 - 8.1.5 Personnel Allocation Planning, 56
 - 8.1.6 Status Reports and Reviews, 56
 - 8.1.7 Control of Subcontractors, 57
 - 8.2 Exercises, 58

- 9 Software Configuration Management, 59**
 - 9.1 Organizational Choices, 59
 - 9.2 SCM Tools, 61
 - 9.3 Guidance Documents, 63

x CONTENTS

- 9.4 Concerns, 63
 - 9.4.1 Double-Baselining, 64
 - 9.4.2 Baseline Divergence, 65
- 9.5 Conclusion, 66
- 9.6 Exercises, 66

10 Software Quality Assurance, 67

- 10.1 Reasons for SQA, 67
- 10.2 SQA Roles, 68
 - 10.2.1 Management Visibility, 68
 - 10.2.2 Organizational Conflicts, 68
- 10.3 Implementation Concerns, 70
- 10.4 Organizational Implications, 71
- 10.5 Exercises, 71

11 The Software Standards and Procedures Manual, 73

- 11.1 Overview, 73
 - 11.1.1 Common Culture, 74
 - 11.1.2 Layers of Detail, 74
- 11.2 Initial Coverage, 75
- 11.3 Format, 76
- 11.4 Summary, 78
- 11.5 Exercises, 79

12 The Software Requirements Review, 80

- 12.1 Purpose of the Review, 80
- 12.2 Material to be Reviewed, 81
- 12.3 Material Review Criteria, 83
- 12.4 Pre-SRR Scheduling Actions, 84
- 12.5 Conduct of the Review, 86
- 12.6 Post-SRR Actions, 86
- 12.7 Additional Thoughts, 87
- 12.8 Exercises, 87

13 The Preliminary Design Phase, 88

- 13.1 What Is To Be Done, 88
 - 13.1.1 Software Top-Level Design Description, 89
 - 13.1.2 Software Integration Plan, 90
 - 13.1.3 Software Test Procedures (Draft), 92
 - 13.1.4 User Documentation (Draft), 92

- 13.1.4.1 Operator Manuals, 93
- 13.1.4.2 Maintenance Manuals, 93
- 13.1.4.3 Training Material, 94
- 13.1.5 Changes To Previously Approved Documents, 94
- 13.2 Who Is To Do It, 95
- 13.3 Final Review, 95
- 13.4 Exercise, 96

14 Establishing the Top-Level Design, 97

- 14.1 Approaches, 97
 - 14.1.1 Top-Down Design, 98
 - 14.1.2 Completion Criteria, 98
 - 14.1.3 Traceability, 99
 - 14.1.4 Other Methodologies, 100
- 14.2 Format Considerations, 100
 - 14.2.1 Principles, 101
 - 14.2.2 A Specific Format, 102
- 14.3 Implementation Issues, 102
- 14.4 Other Sections, 103
- 14.5 Conclusion, 103
- 14.6 Exercises, 103

15 Planning the Integration of the Software, 104

- 15.1 Approaches, 104
- 15.2 The Software Integration Plan, 105
 - 15.2.1 An Example Policy, 106
- 15.3 Hardware/Software Integration Builds, 107
- 15.4 Metrics, 107
 - 15.4.1 Functional Coverage, 108
 - 15.4.2 Domain Coverage, 108
 - 15.4.3 Branch Coverage, 109
 - 15.4.4 Decision Point Coverage, 109
 - 15.4.5 Error Messages, 110
 - 15.4.6 Other Metrics, 110
 - 15.4.7 Tradeoffs, 110
 - 15.4.8 Other Considerations, 110
 - 15.4.9 Caution, 111
- 15.5 Turnover, 111
- 15.6 Exercises, 111

16 Developing User Documentation, 112

- 16.1 Software Operator Manuals, 112
 - 16.1.1 Part I Format Considerations, 113
 - 16.1.2 Part II Format Considerations, 114
- 16.2 Software Maintenance Manuals, 114
 - 16.2.1 Format Considerations, 115
- 16.3 Software Training Material, 117
- 16.4 Overall, 118
- 16.5 Exercises, 118

17 The Rest of the Reviews and Audits, 119

- 17.1 Preliminary Design Review, 119
 - 17.1.1 Parallels to the SRR, 120
 - 17.1.2 Material To Be Reviewed, 120
 - 17.1.3 Management Documents, 121
- 17.2 Critical Design Review, 121
- 17.3 Test Readiness Review, 122
- 17.4 Functional Audit, 123
- 17.5 Physical Audit, 123
 - 17.5.1 Planning, 124
 - 17.5.2 Implementation, 124
 - 17.5.3 Closeout Actions, 125
- 17.6 In-Process Audits, 125
- 17.7 Exercises, 126

18 The Rest of the Phases, 127

- 18.1 Detailed Design Phase, 127
- 18.2 Code and Unit Test Phase, 128
- 18.3 Integration Phase, 128
- 18.4 Acceptance Test Phase, 128
 - 18.4.1 Anomaly Resolution, 129
 - 18.4.2 Regression Testing, 129
 - 18.4.3 Hardware Availability, 130
 - 18.4.4 Management Metrics, 131
 - 18.4.5 Interface with Support Elements, 131

A Acronyms and Abbreviations, 133

B References, 135

- B.1 IEEE Standards, 135

- B.2 Department of Defense, 136
- B.3 Books, 137
- B.4 Articles, 137

C SAICS Software Requirements Specification, 138

- C.1 Introduction, 138
 - C.1.1 Purpose, 138
 - C.1.2 Scope, 138
 - C.1.3 Background Information, 139
 - C.1.4 Definitions, Acronyms, and Abbreviations, 139
 - C.1.4.1 Definitions, 139
 - C.1.4.2 Acronyms and Abbreviations, 139
 - C.1.5 References, 140
- C.2 General Description, 140
 - C.2.1 Product Perspective, 140
 - C.2.1.1 Hardware Configuration, 140
 - C.2.1.2 SAICS Native Assembler/Linking Loader System, 141
 - C.2.1.2.1 *SAICS Native Assembler/Linking Loader Functions, 141*
 - C.2.1.2.2 *SAICS Native Assembler/Linking Loader Components, 141*
 - C.2.2 Command Interpreter Functions, 142
 - C.2.3 User Characteristics, 143
- C.3 Specific Requirements, 143
 - C.3.1 Functional Requirements, 143
 - C.3.1.1 Functional Requirement Number 1: Prompt the User, 143
 - C.3.1.2 Functional Requirement Number 2: Indicate Syntax Errors, 143
 - C.3.1.3 Functional Requirement Number 3: Route Valid Commands, 144
 - C.3.2 External Interface Requirements, 144
 - C.3.2.1 User Interfaces, 144
 - C.3.2.2 Hardware Interfaces, 144
 - C.3.2.3 Software Interfaces, 145
 - C.3.3 Performance Requirements, 145
 - C.3.4 Design Constraints, 146

C.3.4.1 Standards Compliance, 146

D SAICS Software Top-Level Design Document, 147

- D.1 Introduction, 147
 - D.1.1 Purpose, 147
 - D.1.2 Scope, 147
 - D.1.3 Reference, 147
- D.2 Requirements, 148
- D.3 System Overview, 148
- D.4 Component Designs, 149
 - D.4.1 Command Interpreter Control (CICM), 149
 - D.4.1.1 Function, 149
 - D.4.1.2 Input, 149
 - D.4.1.3 Output, 149
 - D.4.1.4 Other Components Called, 149
 - D.4.1.5 Logic, 149
 - D.4.2 Get the Next Command (CIGC), 149
 - D.4.2.1 Function, 149
 - D.4.2.2 Input, 149
 - D.4.2.3 Output, 150
 - D.4.2.4 Subroutines Called, 150
 - D.4.2.5 Logic, 150
 - D.4.3 Execute the Command (CIXC), 150
 - D.4.3.1 Function, 150
 - D.4.3.2 Input, 150
 - D.4.3.3 Output, 150
 - D.4.3.4 Subroutines Called, 151
 - D.4.3.5 Logic, 151
- D.5 Other, 151

E SAICS Software Detailed Design Document, 153

- E.1 Reference, 153
- E.2 Purpose, 153
- E.3 Design Overview, 153
 - E.3.1 Input to the Command Interpreter, 154
 - E.3.2 Outputs from the Command Interpreter, 154
 - E.3.3 Logic Overview, 154
- E.4 Subroutine Designs, 154
 - E.4.1 Command Interpreter Control (CICM), 154

- E.4.1.1 Global Input, 154
 - E.4.1.2 Global Output, 154
 - E.4.1.3 Logic, 154
 - E.4.1.4 Subroutines Referenced, 155
 - E.4.2 Get the Next Command (CIGC), 155
 - E.4.2.1 Global Input, 156
 - E.4.2.2 Global Output, 156
 - E.4.2.3 Logic, 156
 - E.4.2.4 Subroutines Referenced, 156
 - E.4.3 Execute the Command (CIXC), 156
 - E.4.3.1 Global Input, 156
 - E.4.3.2 Global Output, 156
 - E.4.3.3 Logic, 156
 - E.4.3.4 Subroutines Referenced, 157
 - E.4.4 Identify the Command (CICN), 157
 - E.4.4.1 Global Input, 157
 - E.4.4.2 Global Output, 157
 - E.4.4.3 Logic, 157
 - E.4.5 Invoke the Implementing Subsystem (CIIS), 158
 - E.4.5.1 Global Input, 158
 - E.4.5.2 Global Output, 158
 - E.4.5.3 Logic, 158
 - E.4.5.4 Subroutines Referenced, 159
 - E.5 Other Design Information, 159
 - E.6 Budgets, 160
- F Standard for Free-Form PDL, 161**
 - F.1 Scope, 162
 - F.2 Definitions and Acronyms, 162
 - F.2.1 Definitions, 162
 - F.2.2 Acronyms, 162
 - F.3 PDL Language Requirements, 162
 - F.3.1 Indenting and Labeling, 162
 - F.3.2 The Control Structures, 163
 - F.3.2.1 Selection, 163
 - F.3.2.2 Iteration, 163
 - F.3.2.3 Module Call, 163
 - F.3.2.4 Meta-Language Conventions, 163
 - F.3.3 Constructs, 164

- F.3.3.1** IF Construct, 164
- F.3.3.2** IF-THEN-ELSE Construct, 164
- F.3.3.3** CASE Construct, 165
- F.3.3.4** WHILE Construct, 166
- F.3.3.5** CALL Construct, 167