

**SOFTWARE
ENGINEERING
for
IMAGE
PROCESSING
SYSTEMS**

Phillip A. Laplante



CRC PRESS

Contents

Chapter 1	Software Engineering: An Overview	1
1.1	Introduction	1
1.2	A Case for a Software Engineering Approach to Building Imaging Systems	2
1.3	The Role of the Software Engineer	3
1.4	The Nature of Software for Imaging Systems	4
1.5	Case Study: A Visual Inspection System	5
1.6	Misconceptions about Software Engineering	7
1.7	Exercises	8
Chapter 2	Imaging Software and Its Properties	9
2.1	Classification of Software Qualities	9
2.1.1	Reliability	9
2.1.2	Correctness	12
2.1.3	Performance	12
2.1.4	Usability	13
2.1.5	Interoperability	13
2.1.6	Maintainability	13
2.1.7	Portability	14
2.1.8	Verifiability	14
2.1.9	Summary of Software Properties and Associated Metrics	14
2.2	Basic Software Engineering Principles	15
2.2.1	Rigor and Formality	15
2.2.2	Separation of Concerns	15
2.2.3	Modularity	15
2.2.4	Anticipation of Change	18
2.2.5	Generality	20
2.2.6	Incrementality	20
2.2.7	Traceability	20
2.3	Exercises	22
Chapter 3	Software Process and Life Cycle Models	23
3.1	Software Processes and Methodologies	23
3.2	Software Life Cycle Models	23
3.2.1	The Waterfall Model	24
3.2.1.1	Software Conception	25
3.2.1.2	Requirements Specification	25

3.2.1.3	Software Design.....	26
3.2.1.4	Software Development.....	26
3.2.1.5	Testing.....	27
3.2.1.6	Software Maintenance	27
3.2.1.7	Backtracking Transitions in the Waterfall Life Cycle	27
3.2.1.8	Waterfall Model Summary	28
3.2.2	V Model	28
3.2.3	The Spiral Model	28
3.2.4	Evolutionary Model	30
3.2.5	Incremental Model	31
3.2.6	Fountain Model	31
3.2.7	Lightweight Methodologies	32
3.2.8	Unified Process Model.....	34
3.2.9	Capability Maturity Model	34
3.2.9.1	CMM-1: Initial.....	34
3.2.9.2	CMM-2: Repeatable	34
3.2.9.3	CMM-3: Defined	35
3.2.9.4	CMM-4: Managed	35
3.2.9.5	CMM-5: Optimizing.....	35
3.2.9.6	CMM-I.....	35
3.2.10	Prototyping and Risk	35
3.3	Software Standards	36
3.3.1	DOD-STD-2167A	37
3.3.2	DOD-STD-498.....	37
3.3.3	ISO 9000-3.....	38
3.3.4	ISO/IEC	39
3.4	Exercises.....	40
 Chapter 4 Software Requirements		41
4.1	Requirements Engineering Process	41
4.2	Types of Requirements	42
4.3	Requirements Users	43
4.4	Formal Methods in Software Specification	44
4.4.1	Limitations of Formal Methods.....	45
4.4.2	Z	46
4.4.3	Finite State Machines.....	46
4.4.4	Statecharts	49
4.4.5	Petri Nets.....	51
4.5	Specification of Imaging Systems: A Survey of Current Practices	53
4.5.1	Multiresolution Block-Matching System Specification Using a Block Diagram and Flowchart.....	55
4.5.2	Collision Testing of Graphical Objects Using Pseudo-Code.....	56
4.5.3	Functional Representation of Machine Vision System Using a Structured Approach.....	56

4.5.4	Markov Random Fields Image Reconstruction Using Object-Oriented Design	57
4.6	Case Study.....	58
4.6.1	Structured Analysis and Design.....	59
4.6.2	Structured Analysis	59
4.7	Object-Oriented Analysis.....	61
4.8	Object-Oriented vs. Structured Analysis.....	62
4.8.1	Recommendations on Specification Approach for Imaging Systems	64
4.9	Organizing the Requirements Document	64
4.9.1	Writing Good Requirements.....	66
4.10	Requirements Validation and Review	67
4.11	Some Surprises about Current Software Specification Practices.....	68
4.11.1	Surprise 1	68
4.11.2	Surprise 2	68
4.11.3	Surprise 3	68
4.11.4	Surprise 4	69
4.12	Exercises.....	69
Chapter 5	Software System Design.....	71
5.1	The Design Activity	71
5.2	Procedural-Oriented Design.....	72
5.2.1	Parnas Partitioning	72
5.2.2	Structured Design.....	74
5.2.2.1	Transitioning from Structured Analysis to Structured Design.....	74
5.2.2.2	Data Dictionaries	76
5.2.2.3	Problems with SASD in Imaging Applications	77
5.2.2.4	Real-Time Extensions of SASD.....	78
5.2.3	Design in Procedural Form Using Finite State Machines.....	78
5.3	Object-Oriented Design	80
5.3.1	Benefits of Object Orientation.....	81
5.3.1.1	Open-Closed Principle	81
5.3.1.2	Once and Only Once	82
5.3.1.3	Dependency Inversion Principle.....	82
5.3.1.4	Liskov Substitution Principle	82
5.3.2	Design Patterns	83
5.3.3	Object-Oriented Design Using Unified Modeling Language.....	84
5.3.4	Modeling Time Explicitly.....	84
5.3.5	Visual Inspection System Case Study	88
5.4	Hardware Considerations in Imaging System Design	93
5.4.1	Processors.....	94
5.4.2	Non-von Neumann Architectures	95
5.4.2.1	Single Instruction Single Data	95
5.4.2.2	Single Instruction Multiple Data.....	95

5.4.2.3	Multiple Instruction Single Data	96
5.4.2.4	Multiple Instruction Multiple Data	96
5.4.3	Interrupt Handling	96
5.4.4	Memory	97
5.4.5	Input and Output	97
5.5	Fault-Tolerant Design	99
5.5.1	Spatial Fault Tolerance	99
5.5.2	Using a Kalman Filter in the Case Study System	99
5.5.3	Checkpoints	102
5.5.4	Recovery Blocks	102
5.5.5	Software Black Boxes	104
5.5.6	N-Version Programming	105
5.5.6.1	Built-In Test Software	105
5.5.6.2	CPU Testing	106
5.5.6.3	Memory Testing	106
5.5.6.4	Other Devices	107
5.6	Exercises	107
Chapter 6	The Software Production Process	109
6.1	Programming Languages	109
6.1.1	Parameter Passing Techniques	110
6.1.2	Call-by-Value and Call-by-Reference	110
6.1.3	Global Variables	110
6.1.4	Recursion	111
6.1.5	Dynamic Memory Allocation	111
6.1.6	Typing	112
6.1.7	Exception Handling	112
6.1.8	Modularity	113
6.1.9	Brief Survey of Languages	114
6.1.9.1	Ada 95	114
6.1.9.2	Assembly Language	115
6.1.9.3	C	115
6.1.9.4	C++	116
6.1.9.5	Fortran	116
6.1.9.6	Java	117
6.2	Writing and Testing Code	118
6.2.1	Example: The Unix/Linux C Compiler	119
6.2.2	Handling Compiler Errors	120
6.2.3	Some Debugging Tips: Unit-Level Testing	120
6.2.4	Extended Syntax and Semantic Checking	120
6.2.5	Symbolic Debugging	121
6.2.6	Test-First Coding	122
6.2.7	Know the Compiler	122
6.3	Coding Standards	123
6.4	Reviews and Audits	124

6.5	Documentation	126
6.6	Exercises.....	127
Chapter 7 Software Measurement and Testing.....		129
7.1	The Role of Metrics.....	129
7.1.1	Lines of Code.....	129
7.1.2	McCabe's Metric.....	130
	7.1.2.1 Measuring Software Complexity.....	130
	7.1.2.2 Determining the Limit on Number of Test Cases	132
7.1.3	Halstead's Metrics.....	132
7.1.4	Function Points	133
7.1.5	Feature Points.....	137
7.1.6	Metrics for Object-Oriented Software.....	137
7.1.7	Objections to Metrics.....	138
7.2	Faults, Failures, and Bugs.....	138
7.3	The Role of Testing	139
7.4	Testing Techniques.....	139
7.4.1	Unit-Level Testing.....	139
	7.4.1.1 Black Box Testing	139
	7.4.1.2 White Box Testing.....	141
7.4.2	Testing Object-Oriented Software	142
7.4.3	System-Level Testing.....	142
	7.4.3.1 Cleanroom Testing.....	143
	7.4.3.2 Stress Testing	143
7.5	Design of Testing Plans	144
7.6	Exercises.....	144
Chapter 8 Hardware-Software Integration and Maintenance		145
8.1	Goals of System Integration	145
8.2	System Unification.....	145
8.3	System Verification	146
8.4	System Integration Tools	146
8.4.1	Multimeter	147
8.4.2	Oscilloscope	147
8.4.3	Logic Analyzer.....	147
	8.4.3.1 Timing Instructions.....	148
	8.4.3.2 Timing Code	148
8.4.4	In-Circuit Emulator	149
8.4.5	Software Simulators.....	149
8.4.6	Hardware Prototypes.....	149
8.5	Software Integration.....	150
	8.5.1 A Simple Integration Strategy	150
	8.5.2 Patching.....	150
	8.5.3 The Probe Effect	152

9.9	Cost Estimation Using COCOMO	180
9.9.1	Basic COCOMO	180
9.9.2	Intermediate and Detailed COCOMO	182
9.9.3	COCOMO II	183
9.10	Exercises.....	183
	Glossary	185
	References.....	203
	Index.....	209