

---

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Component-Based Software Development	2
1.1.1	Component Definition	2
1.1.2	Core Principles of Component-Based Development	4
1.1.3	Component Meta-model	7
1.1.4	Component Engineering vs. Application Engineering	9
1.2	Component-Based Software Testing	11
1.2.1	Challenges in Component-Based Software Testing	12
1.2.2	The ARIANE 5 Failure	14
1.2.3	The Lessons Learned	15
1.3	Model-Based Development and Testing	16
1.3.1	UML and Testing	16
1.3.2	Model-Based Testing	18
1.3.3	Test Modeling	18
1.4	Summary and Outline of This Book	18
<b>2</b>	<b>Component-Based and Model-Driven Development with UML</b>	21
2.1	Principles of the Kobra Method	22
2.1.1	Decomposition	24
2.1.2	Embodiment	26
2.1.3	Composition	27
2.1.4	Validation	27
2.1.5	Spiral Model vs. Waterfall Model	27
2.2	Context Realization	29
2.2.1	Usage Model	30
2.2.2	Enterprise or Business Process Model	33
2.2.3	Structural Model	33
2.2.4	Activity and Interaction Model	35
2.3	Component Specification	38
2.3.1	Structural Specification	39

2.3.2	Functional Specification .....	41
2.3.3	Behavioral Specification .....	42
2.4	Component Realization .....	44
2.4.1	Realization Structural Specification .....	46
2.4.2	Realization Algorithmic Specification .....	48
2.4.3	Realization Interaction Specification .....	48
2.5	Component Embodiment .....	50
2.5.1	Refinement and Translation .....	53
2.5.2	The Normal Object Form .....	55
2.5.3	Component Reuse .....	56
2.5.4	COTS Component Integration .....	58
2.5.5	System Construction and Deployment .....	60
2.6	Product Family Concepts .....	61
2.6.1	Decision Models .....	62
2.6.2	Framework Engineering .....	64
2.6.3	Application Engineering .....	68
2.7	Documentation and Quality Assurance Plan .....	69
2.8	Summary .....	70
<b>3</b>	<b>Model-Based Testing with UML .....</b>	<b>73</b>
3.1	Model-Based vs. Traditional Software Testing .....	74
3.1.1	White Box Testing Criteria .....	75
3.1.2	Black Box Testing Criteria .....	77
3.2	Model-Based Testing .....	80
3.2.1	Usage Modeling .....	80
3.2.2	Use Case Diagram-Based Testing .....	81
3.2.3	Use Case and Operation Specification-Based Testing ..	84
3.2.4	Structural Modeling .....	88
3.2.5	Structural Diagram-Based Testing .....	95
3.2.6	Behavioral Modeling with Statecharts .....	98
3.2.7	Statechart Diagram-Based Testing .....	99
3.2.8	Behavioral Modeling with Activity Diagrams .....	102
3.2.9	Activity Diagram-Based Testing .....	104
3.2.10	Interaction Modeling .....	106
3.2.11	Interaction Diagram-Based Testing .....	109
3.3	Test Modeling .....	112
3.3.1	Structural Aspects of Testing .....	112
3.3.2	Behavioral Aspects of Testing .....	113
3.3.3	UML Testing Profile Mapping .....	115
3.3.4	Extension of the Testing Profile .....	118
3.4	Summary .....	119

<b>4</b>	<b>Built-in Contract Testing</b> .....	121
4.1	Concepts of Built-in Testing .....	123
4.1.1	Assertions .....	123
4.1.2	Built-in Testing .....	124
4.2	Motivation for Built-in Contract Testing .....	127
4.2.1	Objective of Built-in Contract Testing .....	127
4.2.2	Component Contracts .....	129
4.3	Model and Architecture of Built-in Contract Testing .....	130
4.3.1	Explicit vs. Implicit Servers .....	133
4.3.2	The Testing Interface .....	134
4.3.3	Optimal Design of the Testing Interface .....	140
4.3.4	Tester Components .....	146
4.3.5	Optimal Design of a Tester Component .....	148
4.3.6	Component Associations in Built-in Contract Testing ..	152
4.4	Development Process for Built-in Contract Testing .....	157
4.4.1	Identification of Tested Interactions .....	163
4.4.2	Definition and Modeling of the Testing Architecture ..	164
4.4.3	Specification and Realization of the Testing Interfaces ..	167
4.4.4	Specification and Realization of the Tester Components	169
4.4.5	Integration of the Components .....	174
4.5	Summary .....	177
<b>5</b>	<b>Built-in Contract Testing and Implementation Technologies</b>	179
5.1	Instantiation and Embodiment of Built-in Contract Testing ..	183
5.2	Built-in Contract Testing with Programming Languages .....	187
5.2.1	Procedural Embodiment Under C .....	188
5.2.2	Object-Oriented Embodiment Under C++ and Java ....	191
5.3	Component Technologies .....	200
5.3.1	JavaBeans and Enterprise JavaBeans .....	201
5.3.2	COM, DCOM, ActiveX, COM+, and .NET .....	203
5.3.3	CORBA, OMA and CCM .....	204
5.3.4	Component Technologies and Built-in Contract Testing	206
5.4	Built-in Contract Testing and Web Services .....	209
5.4.1	Checking Web Services Through Contract Testing .....	210
5.4.2	Testing of Readily Initialized Server Components .....	212
5.5	Implementation Technologies for Built-in Contract Testing ..	214
5.5.1	The XUnit Testing Framework .....	215
5.5.2	JUnit and Built-in Contract Testing .....	216
5.5.3	The Testing and Test Control Notation – TTCN-3 .....	219
5.5.4	TTCN-3 and Built-in Contract Testing .....	223
5.6	Summary .....	226

<b>6</b>	<b>Reuse and Related Technologies</b> .....	229
6.1	Use and Reuse of Contract Testing Artifacts .....	231
6.1.1	Development-Time Reuse .....	232
6.1.2	Runtime Reuse .....	235
6.2	Component Certification and Procurement .....	238
6.2.1	The CLARiFi Component Broker Platform .....	239
6.2.2	Customer Self-certification .....	240
6.3	Product Families and Testing .....	242
6.3.1	Testing of Product Families .....	244
6.3.2	Testing as a Product Family Development .....	253
6.4	Summary .....	254
<b>7</b>	<b>Assessing Quality-of-Service Contracts</b> .....	255
7.1	Quality-of-Service Contracts in Component-Based Development .....	256
7.2	Timing Analysis and Assessment with Components .....	260
7.2.1	Typical Timing Problems .....	261
7.2.2	Timing Analysis Approaches .....	263
7.3	Extended Model of Built-in Contract Testing .....	265
7.3.1	Testing Interface for the Extended Model .....	267
7.3.2	Tester Component for the Extended Model .....	268
7.3.3	Optimization-Based Timing Analysis .....	272
7.3.4	Application to the RIN System .....	274
7.4	QoS Contract Testing for Dynamic Updates .....	279
7.5	Built-in Quality-of-Service Runtime Monitoring .....	280
7.6	Summary .....	283
	<b>Glossary</b> .....	285
	<b>References</b> .....	297
	<b>Index</b> .....	307