



Maximizing ROI on Software Development

Vijay Sikka



CONTENTS

Preface	xix
About the Author	xxiii
Acknowledgments	xxv
1 A Brief Review of Software Development History	1
1.1 Classic Software Development Methodologies	2
1.1.1 Waterfall	2
1.1.1.1 Limitations of Waterfall Model	3
1.1.2 Incremental	3
1.1.3 Spiral	4
1.1.3.1 Feasibility Determination	5
1.1.3.2 Lifecycle	5
1.1.3.3 Initial Operational Capability Milestone	5
1.1.3.4 Risk Exposure Assessment	5
1.1.4 Prototyping	6
1.1.4.1 Limitations of Prototyping	6
1.1.5 Cleanroom	6
1.1.5.1 Starting Assumptions	6
1.1.5.2 Boxes	7
1.1.5.3 Steps for Cleanroom	7
1.1.5.4 Verification and Testing in Cleanroom	8
1.1.6 Object-Oriented	8
1.1.6.1 Key Relationships	9
1.1.6.2 Booch	9
1.1.6.3 Coad and Yourdon	9
1.1.6.4 Object Modeling Technique by Rumbaugh	10
1.1.6.5 Jacobson: Objectory and OOSE	10
1.2 Evolving Software Development Methodologies	10
1.2.1 The Agile Themes	10
1.2.1.1 Most Effort Is Spent on Design	11
1.2.1.2 Business Systems Are Inherently Unpredictable	11
1.2.1.3 Implement, Review, Iterate	11
1.2.1.4 Developers Are More Important than Processes	12

1.2.2	Agile Methodologies.....	12
1.2.2.1	eXtreme Programming.....	13
1.2.2.2	Advanced Software Development.....	15
1.2.2.3	SCRUM.....	17
1.2.2.4	Feature Driven Development.....	18
1.2.2.5	Lean Software Development.....	19
1.2.2.6	Dynamic Systems Development Method.....	22
1.2.2.7	Crystal Family.....	22
1.2.2.8	Unified Modeling Language.....	23
1.2.2.9	Rational Unified Process.....	23
1.2.3	Model Driven Architecture.....	24
1.2.3.1	Interoperability between Platforms.....	25
1.2.3.2	Independence from Platforms.....	25
1.2.3.3	Portability.....	25
1.2.3.4	Improved Communication.....	25
1.2.3.5	Enhanced Return on Investment.....	25
1.2.4	Web Service Development.....	26
1.2.4.1	The Lifecycle of Web Services.....	26
1.2.4.2	Design.....	27
1.2.4.3	Development.....	27
1.2.4.4	Testing.....	27
1.2.4.5	Deployment in SOAP or XML Applications.....	27
1.2.4.6	Execution.....	27
1.2.4.7	Management.....	28
1.3	Evolutionary Predictions.....	28
1.3.1	The Future of Agile Development.....	29
1.3.1.1	The Agile Economy.....	29
1.3.1.2	Agile Is a Strategic Capability.....	29
1.3.1.3	Application Examples.....	30
1.3.2	Application Trends.....	30
1.3.2.1	Evolving Applications and Platforms.....	31
1.3.2.2	Application Integration Opportunity.....	31
1.3.2.3	Agile Application Integration.....	32
1.3.2.4	Software Reuse.....	32
1.3.2.5	Management Responsibilities.....	32
1.3.3	Future of Web Services Development.....	33
1.3.3.1	Agile Services.....	33
1.3.3.2	Write Once Use Anywhere.....	33
1.3.3.3	Self-Aware Services.....	33
1.3.3.4	Easy to Upgrade Services.....	33
1.3.3.5	Loosely Coupled Scaleable Services.....	33
1.3.3.6	Platform Independent Services.....	34
1.4	Summary of Chapter.....	34
	References.....	35

2	Software Complexity Crisis	39
2.1	Contemporary Software Development Is Multifaceted and Complex	39
2.1.1	Old Car and Computers Analogy	39
2.1.2	Diverse Development Locations	40
2.1.3	Diverse Development Teams	41
2.1.4	Diverse Platforms	42
2.2	Market Trends Favor Diversity	42
2.2.1	Proliferation of Software Development and Deployment Environments	44
2.2.1.1	Microsoft Platforms.....	44
2.2.1.2	Java Platforms	45
2.2.1.3	Open Source Platforms.....	45
2.2.1.4	Mobile Platforms.....	46
2.2.2	Proliferation of Software Maturity Models	47
2.2.2.1	Software Process Models	47
2.2.2.2	Capability Maturity Model Integrated	48
2.2.2.3	Information Technology Infrastructure Library	50
2.2.2.4	Control Objectives for Information Technology	53
2.2.2.5	Product Line Model.....	55
2.3	Changing Role of Information Technology.....	60
2.3.1	Global Software Development and Test Opportunity	61
2.3.1.1	Widely Available Technical Training	61
2.3.1.2	Software Development Skills are a Commodity.....	62
2.3.1.3	Global Software Development and Test Reduces Costs	62
2.3.1.4	Entire World Is Becoming a Software Consumer.....	62
2.3.2	Outsourcing and Offshoring.....	62
2.3.2.1	Human Factors Are Most Important	63
2.3.2.2	Differences from IT-Enabled and Business Process Outsourcing.....	63
2.3.2.3	Governments Can Play Constructive Role	63
2.4	Business Case for Maximizing Software ROI.....	63
2.4.1	Excuses for Not Calculating ROI	64
2.4.1.1	ROI is Wasteful Number Crunching	64
2.4.1.2	We Do Not Have the Expertise	65
2.4.1.3	We Will Wait Until Later	65
2.4.1.4	ROI Will Be Positive Anyway	65
2.4.1.5	Let Us Use Someone Else's Data	66
2.4.2	Benefits of ROI.....	66
2.4.2.1	Helps Establish Business Value.....	67
2.4.2.2	Helps Establish Metrics	67
2.4.2.3	ROI gives a Healthy Dose of Reality	68
2.4.2.4	Helps Establish Best Practices.....	68
2.4.2.5	Facilitates Data Collection	69

2.4.3	Business Value Capture Using ROI	70
2.4.3.1	Analyze Opportunities	70
2.4.3.2	Determine Strategy	70
2.4.3.3	Make Decisions	70
2.4.3.4	Manage Portfolio	70
2.4.3.5	Continuous Improvement	71
2.4.4	Getting Ready for ROI Analysis	71
2.4.4.1	Plan Effective Timelines	71
2.4.4.2	Establish Performance Objectives	71
2.4.4.3	Guard Against Over-reliance on Numbers	72
2.4.4.4	Do Not Assume Anything	72
2.4.4.5	Make Appropriate Investments	73
2.4.4.6	Get Stakeholder Buy-In	73
2.4.4.7	Get Ready to Measure, Measure, Measure	74
2.4.5	Avoiding Common ROI Mistakes	74
2.4.5.1	Missing Strategic Vision and Stakeholder Buy-In	74
2.4.5.2	Lack of Domain Expertise	75
2.4.5.3	Inaccurate Assumptions	75
2.4.5.4	Incorrect Timing	75
2.4.5.5	Missing Risk and Cost Profile	75
2.4.5.6	Unsubstantiated Data or Supporting Evidence	76
2.4.5.7	Overestimating Gains	76
2.5	Summary of Chapter	76
	References	77
3	Software Development ROI	79
3.1	Days of "Good Enough" Software Development Are Over	79
3.1.1	ROI Applies to Managers and Developers Alike	81
3.1.2	Use a Business Model	81
3.1.3	Priority of Customers May Not Match Those of ROI	82
3.1.4	Scalability and ROI	82
3.1.5	Use Interfaces that Are Neutral to Technology	84
3.1.6	Financial View of Deliverables	84
3.2	Baseline ROI	84
3.2.1	Definition of ROI	85
3.2.2	Components of ROI	85
3.2.2.1	ROI Percentage	86
3.2.2.2	ROI Factor	86
3.2.2.3	Net Benefit	86
3.2.2.4	Net Cost	86
3.2.2.5	Discount Rate and Internal Rate of Return	86
3.2.2.6	Payback Period	87
3.2.2.7	Net Present Value	87
3.2.3	Data Collection for ROI	88
3.2.3.1	Timeframes of ROI	88
3.2.3.2	Mythical 1000 Percent ROI	88

3.2.3.3	Example Case: Sales Force Automation Application.....	89
3.2.3.4	Summary of Case.....	90
3.3	Applied ROI.....	90
3.3.1	Phases of Applied ROI.....	90
3.3.1.1	Baseline ROI Is the First Step.....	90
3.3.1.2	Training.....	91
3.3.1.3	Tools and Automation.....	91
3.3.1.4	Processes and Methodologies.....	92
3.3.1.5	Cosourcing.....	92
3.3.2	TCO and ROI.....	93
3.3.2.1	Calculating TCO.....	93
3.3.2.2	Steps to Calculate TCO.....	94
3.3.3	Methodologies of Applied ROI.....	94
3.3.3.1	Capability and Maturity Model.....	94
3.3.3.2	ISO 9001.....	96
3.3.3.3	Six Sigma.....	97
3.3.3.4	Value of Methodology to ROI.....	99
3.3.4	Tools and Techniques for Applied ROI.....	99
3.3.4.1	eXtreme Programming ROI.....	99
3.3.4.2	Web Services ROI.....	100
3.3.5	When ROI Is Hard to Quantify.....	101
3.3.5.1	Business Values.....	102
3.3.5.2	Market Share and Brand Equity.....	102
3.3.5.3	Competitive Pressures.....	102
3.3.5.4	Regulatory Environment.....	103
3.3.5.5	Support and Serviceability Cost.....	103
3.4	Summary of Chapter.....	103
	References.....	104
4	The Case for Global Software Development and Testing	107
4.1	Scenarios of Global Software Development.....	108
4.1.1	Intraenterprise Onshore and Offshore.....	108
4.1.2	Extraenterprise Onshore and Offshore.....	109
4.2	Benefits of Global Software Development and Testing.....	109
4.2.1	24/7 Software Development Cycle.....	109
4.2.2	Diversified Talent Pool.....	109
4.2.3	Proximity to Software Consumers and Marketplaces.....	110
4.2.4	Cost Savings.....	110
4.2.5	Reliability through Distributed Development.....	110
4.2.6	Handle Peaks and Valleys in Demand.....	111
4.3	Technology Is the Global Enabler.....	111
4.3.1	Project Management Tools.....	111
4.3.2	Communication Tools.....	112
4.3.2.1	Remote Conferencing.....	113
4.3.2.2	Case.....	113

4.3.3	Distributed Code Management and Test Tools	114
4.3.4	Emergence of Common Standards.....	114
4.3.5	Component-Based Software Architecture.....	114
4.3.6	Seamless and Secure Networks.....	114
4.4	Success Criteria for Global Software Development and Testing.....	115
4.4.1	Define, Execute and Measure to a Plan	116
4.4.1.1	Aim High.....	116
4.4.1.2	Find Champions	116
4.4.1.3	Find Stakeholders	116
4.4.1.4	Integrate Objectives.....	117
4.4.2	Management Responsibilities.....	117
4.4.2.1	No Second Class Citizens	117
4.4.2.2	Establish High Quality Hardware Infrastructure.....	117
4.4.2.3	Follow Best Practices for Personnel Management.....	117
4.4.2.4	Follow Best Practices for Partner Management.....	118
4.4.2.5	Establish High Quality Software Infrastructure.....	118
4.4.3	Personal Strategies.....	118
4.4.3.1	Keep Aware of Local Regulations.....	118
4.4.3.2	Patience Is a Must.....	119
4.4.3.3	Rely on Global Brands	119
4.4.3.4	Avoid Driving in Developing Countries	119
4.4.3.5	Be Prepared for Sensory Overloads	120
4.4.3.6	Work Hours Will Not Get Shorter	120
4.5	What to Watch Out for.....	120
4.5.1	Strong Cultural Currents	121
4.5.2	Anti-Americanism.....	122
4.5.3	Varying Infrastructure and Services	122
4.5.4	Language Barriers.....	123
4.5.5	Political Issues	124
4.5.6	Business Issues	124
4.5.6.1	Reciprocity and Investment Laws	124
4.5.6.2	Approval Processes	124
4.5.6.3	Taxation Laws.....	125
4.6	Outsourcing and Offshoring.....	125
4.6.1	Outsourcing	125
4.6.1.1	IT Outsourcing.....	126
4.6.1.2	Offshoring	126
4.6.1.3	Offshore Software Development.....	126
4.6.2	Onshore and Offshore Outsourcing Myths.....	127
4.6.2.1	Internal Staff Responsibility.....	127
4.6.2.2	Saving Enterprise Money	127
4.6.2.3	Responsibility Shift	127
4.6.2.4	Investment Requirement.....	128
4.6.2.5	Management Involvement.....	128
4.6.2.6	Information Protection.....	128
4.6.2.7	Information Sharing.....	129

4.6.2.8 Cheap Low-Skilled Labor.....	129
4.6.2.9 Poor Working Conditions.....	129
4.6.2.10 Old Equipment and Software Privacy.....	129
4.7 Summary.....	130
References.....	131

5 Software Quality and Test ROI.....133

5.1 How to Minimize Risk and Maximize Quality.....	133
5.1.1 Myths of Software Quality and Testing.....	133
5.1.1.1 Quality and Cost.....	133
5.1.1.2 Quality and Agile Programming.....	134
5.1.1.3 Software and Art.....	134
5.1.1.4 CMM and Management.....	135
5.1.1.5 Wasted Time.....	135
5.1.1.6 ROI and Quality.....	135
5.1.1.7 Shortcuts.....	135
5.1.2 Software Inspection Techniques.....	136
5.1.2.1 Testing and Inspection.....	136
5.1.2.2 Manual Code Inspection.....	137
5.1.2.3 Automated Code Inspection.....	137
5.1.3 Organizational Measures.....	138
5.1.3.1 Measure Defects to Improve Quality.....	138
5.1.3.2 Obtain Organization Commitment.....	138
5.1.3.3 Measure Organizational Success with ROI Goals.....	138
5.1.3.4 Use Key Performance Indicators.....	138
5.1.3.5 Use Successive Rings of Applied ROI to Improve ROI.....	139
5.1.4 Continuous Software Testing.....	139
5.1.4.1 Testing Is “Designed into” Software Development.....	139
5.1.4.2 Continuous Testing Stakeholders.....	140
5.1.4.3 Business of Continuous Testing.....	140
5.1.5 Metric versus Measure and a New International Standard... 140	
5.1.5.1 Information Needs Evaluation.....	141
5.1.5.2 Measurement Process Resource Allocation.....	141
5.1.5.3 Measurement Process Planning and Execution.....	141
5.1.5.4 Measurement Process Evaluation, Data Capture, and Analysis.....	141
5.1.5.5 Measurement Process Key Learnings and Communication.....	142
5.2 Software Testing ROI.....	142
5.2.1 Costs of Quality and Testing.....	142
5.2.1.1 Defects per 1000 Lines of Code.....	142
5.2.1.2 Fully Loaded Tester Cost.....	142
5.2.1.3 Developer Time to Fix Defects.....	143
5.2.2 Applied ROI Considerations.....	143

5.2.2.1	Manual versus Automated Testing	143
5.2.2.2	Automated Test Tools	148
5.2.2.3	In-House, Outsource, or Cosource?	150
5.2.3	Calculating Testing ROI	152
5.2.3.1	Tangible ROI Savings	152
5.2.3.2	Intangible ROI Savings	154
5.2.3.3	Refining the Cost Savings Model	155
5.3	Localization and Internationalization ROI	155
5.3.1	Factors for ROI Calculation	157
5.3.1.1	Cost of Localization	157
5.3.1.2	Cost of Internationalization	159
5.3.1.3	Revenue from Localization	159
5.3.1.4	Nonquantifiable Factors	160
5.3.2	Calculating ROI of Localization	160
5.3.2.1	Net Benefit	161
5.3.2.2	Net Cost	161
5.3.2.3	Baseline ROI	161
5.3.3	Applied ROI Considerations	161
5.3.3.1	Reduce Costs	162
5.3.3.2	Understand Target Market	162
5.3.3.3	Use Automation	162
5.3.3.4	Use Effective Process and Methodology	163
5.3.3.5	Use Cosourcing	163
5.3.4	Localization and Internationalization ROI Summary	164
5.4	Software Training ROI	164
5.4.1	Why Training Needs ROI	165
5.4.1.1	ROI Gives Valuable Insights into Training	165
5.4.1.2	Training Provides Business Value	166
5.4.1.3	Training Consumes Resources	166
5.4.1.4	Older Models of Training Are Becoming Obsolete	166
5.4.1.5	Selecting the Best Training Method Is Important	167
5.4.2	Factors for Measuring Training	167
5.4.2.1	During Training Factors	167
5.4.2.2	Post-Training Factors	168
5.4.3	Calculating Training ROI	169
5.4.3.1	Net Costs	170
5.4.3.2	Net Benefits	171
5.4.3.3	Baseline ROI	173
5.4.3.4	Case	173
5.4.4	Applied ROI Considerations	174
5.4.4.1	Online Training	174
5.4.4.2	Outsourcing Collateral and Marketing	175
5.4.5	Training ROI Summary	175
5.5	Summary	178
	References	178

6	How Do You Implement Global Software Development and Testing.....	181
6.1	The Global Software Development and Testing Model	181
6.1.1	Software Development Lifecycle.....	183
6.1.1.1	Recognize Problem.....	183
6.1.1.2	Analyze.....	183
6.1.1.3	Design	183
6.1.1.4	Implement.....	184
6.1.1.5	Test.....	184
6.1.1.6	Maintain.....	184
6.1.2	QA and Testing Roadmap.....	184
6.2	Product Readiness and Deployment Infrastructure	184
6.2.1	Product Readiness.....	185
6.2.1.1	Functionality/Features	185
6.2.1.2	Localization	188
6.2.1.3	Usability.....	188
6.2.1.4	Reliability.....	188
6.2.1.5	Performance.....	189
6.2.1.6	Supportability.....	189
6.2.1.7	Data Migration Issues.....	189
6.2.2	Deployment Readiness.....	189
6.2.2.1	Sizing.....	190
6.2.2.2	Deployment Reviews Done?.....	190
6.2.2.3	Support Infrastructure in Place?.....	190
6.2.3	Development and Deployment Infrastructure	190
6.2.3.1	Define.....	191
6.2.3.2	Develop.....	191
6.2.3.3	Build.....	191
6.2.3.4	Release	192
6.2.3.5	Deploy.....	193
6.2.3.6	Maintain.....	195
6.2.3.7	Resource Requirements.....	195
6.3	Selecting and Managing the Outsourcer	197
6.3.1	Selecting the Outsourcer	198
6.3.1.1	Project Definition.....	198
6.3.1.2	Interview	198
6.3.1.3	Addressing Language, Culture, and Time Barriers	198
6.3.1.4	Statement of Work and Formal Contract.....	199
6.3.2	Managing the Outsourcer.....	200
6.3.2.1	Methods and Systems.....	201
6.3.2.2	Goals and Tasking.....	202
6.3.3	Release Candidate Certification Guidelines.....	203
6.3.3.1	Release Guidelines	203
6.3.3.2	During the Project.....	203
6.3.3.3	End of the Project.....	203

- 6.4 Methodologies and Measurement 204
 - 6.4.1 Three Columns of Best Practices and Guidelines 204
 - 6.4.1.1 In-House SQA 204
 - 6.4.1.2 Productive Experience 204
 - 6.4.1.3 Costs and Timelines 206
 - 6.4.2 Best Practices Criteria Details 206
 - 6.4.2.1 Productive Experience 206
 - 6.4.2.2 Costs and Timeliness 207
 - 6.4.2.3 Outsourcing Companies 207
 - 6.4.2.4 Outsourcing Country 208
 - 6.4.2.5 Hurdles for Team Communication 208
 - 6.4.2.6 Language Issues 208
 - 6.4.2.7 Types of Testing Outsourced 209
 - 6.4.2.8 Replace SQA Department? 209
 - 6.4.2.9 Required Backgrounds of SQA and Testing People 209
 - 6.4.2.10 Standards Bodies 209
 - 6.4.3 Experiences of Outsourcing 210
- 6.5 Summary of Chapter 211
- References 211
- 7 Case Studies 213**
 - 7.1 Anatomy of Our Case Studies 213
 - 7.1.1 Introduction 213
 - 7.1.2 Sources 213
 - 7.2 Case 1: Developing and Testing Natural Language Systems for Mobile Devices 215
 - 7.2.1 Natural Language System Profile 215
 - 7.2.2 Case Introduction 215
 - 7.2.2.1 E-Mail Interface 215
 - 7.2.2.2 Wireless Web Interface 215
 - 7.2.3 Technical Overview 215
 - 7.2.3.1 Agents 216
 - 7.2.3.2 Networks of Agents 216
 - 7.2.4 Supported Platforms 216
 - 7.2.4.1 E-Mail Clients 216
 - 7.2.4.2 Browsers for Wireless Handheld Devices 216
 - 7.2.5 Problem Definition and Solution 217
 - 7.2.5.1 Functional Testing 217
 - 7.2.6 Test Case Example 218
 - 7.2.7 Testing Tools Used 218
 - 7.2.7.1 WebLOAD 4.51 218
 - 7.2.7.2 Query Bulder Tool 219
 - 7.2.7.3 TestRunner Tool 220
 - 7.2.7.4 Searching Functions 220
 - 7.2.8 Output Results from Testing Tool 221
 - 7.2.8.1 Log File 221
 - 7.2.8.2 Report File 222

7.2.9	Bug Tracking System Tools.....	222
7.2.9.1	Rational ClearQuest.....	223
7.2.9.2	Bug Lifecycle.....	223
7.2.10	Return on Investment.....	224
7.2.10.1	No Cosourcing for SQA Testing.....	224
7.2.10.2	Cosourcing for SQA Testing.....	224
7.2.11	Conclusion.....	225
7.3	Case 2: Global Testing of Enterprise CRM Market Leader.....	226
7.3.1	CRM Company Profile.....	226
7.3.2	Case Introduction.....	226
7.3.3	Technical Overview of Unicode.....	227
7.3.3.1	What Is Unicode?.....	227
7.3.3.2	Unicode and ISO/IEC 10646.....	227
7.3.3.3	Encoding Forms.....	227
7.3.4	Supported Platforms.....	228
7.3.5	Problem Definition and Solution.....	229
7.3.6	Test Case Example.....	229
7.3.6.1	Testing Scenario and System Response.....	230
7.3.7	Verification and Validation Testing Strategies.....	230
7.3.7.1	Verification Strategies.....	230
7.3.7.2	Validation Strategies.....	231
7.3.7.3	Sequence of Testing.....	231
7.3.8	Testing Tools Used.....	232
7.3.8.1	Automated Tools: WebLOAD 4.51.....	232
7.3.8.2	Manual Tools: Microsoft Project 2000.....	233
7.3.9	Bug Tracking System Tools.....	233
7.3.9.1	DevTrack.....	233
7.3.9.2	RD System.....	233
7.3.10	Remote Communication Tools Used.....	234
7.3.10.1	Three Testing Sites.....	234
7.3.11	ROI.....	234
7.3.11.1	No Cosourcing for SQA Testing.....	235
7.3.11.2	Cosourcing for SQA Testing.....	235
7.3.12	Case Summary.....	235
7.4	Book Summary.....	236
	References.....	238

Appendix A: The Quality Assurance and Testing Process.....239

Index.....243