

Contents

Introduction	xxi
Part I: Philosophy	
Chapter 1: Buy, Not Build	3
Cost versus Benefit	4
Creating a Competitive Advantage	5
Base Class Libraries	6
Open Source Software	7
Taking Advantage of Your Platform	7
Design	7
Risk	8
Services	9
Third-Party Components	9
Summary	10
Chapter 2: Test-Driven Development	11
Tests Define Your Contract	13
Tests Communicate Your Intent	16
Summary	19
Chapter 3: Continuous Integration	21
Integrate Early and Often	22
Keeping Testers Working	22
Keeping Developers Working	22
Barriers to Continuous Integration	23
Build Servers	25
Automating a Build Process	26
Expanding the Build Process	29
Setting Up a CI Server	31
Multiple Builds of the Same Project	35
Coordinating Build Output	35
Notifying People about Build Results	36

Contents

Fix a Broken Build before Integrating Changes	37
Summary	39

Part II: Process

Chapter 4: Done Is Done **43**

Discuss Design Decisions	44
Every Class Has a Test Fixture	45
Each Fixture Exercises Only One Class	48
Code Coverage Is High	49
No Compiler Warnings	51
Static Analysis Tools Generate No Errors	52
Before Committing, Update	53
Documentation in Place	53
Summary	56

Chapter 5: Testing **57**

Why Testing Doesn't Get Done	58
How Testing Will Make You a Better Developer	59
Your Designs Will Be Better	59
You'll Have to Write Less Code to Achieve Your Goals	59
You Will Learn More about Coding	60
You Will Develop Better Relationships with Your Testers	61
You Will Make Your Project Sponsors Happy	61
Code Coverage	64
Why Measure Code Coverage	65
Code Coverage Tools	67
Strategies for Improving Code Coverage	72
Types of Tests	74
Unit Tests	75
Integration Tests	84
Functional Tests	89
Performance Testing	93
Test Automation	96
Strategies for Test Automation	96
Testing Frameworks	97
Automation and Your Organization	104
Summary	106

Chapter 6: Source Control	107
Some Source Control History	108
Concurrent Versioning System (CVS)	110
Picking the Right SCCS	110
Performance and Scalability	111
Locking Models	111
Cost	112
Atomic Commits	112
Branching	113
Merging	114
Integration with Other Tools	114
Reporting	115
Extensibility	115
Making Your Choice	115
Organizing Your Source Tree	117
Using TreeSurgeon	117
Packaging	121
Adding Folders to Work with Source Control	121
Making the Most of Branching	122
Version Branches	123
Integration Branches	127
Personal Branches	129
Task Branches	130
Merging Branches	131
Summary	133
Chapter 7: Static Analysis	135
Using Static Analysis Tools	135
NDepend	136
FxCop	137
Simian	139
Who Benefits from Static Analysis?	139
How to Integrate Static Analysis into Your Process	140
Summary	143
Part III: Code Construction	
Chapter 8: Contract, Contract, Contract!	147
Public Interfaces	152

Contents

Data Contracts	155
Summary	158
Chapter 9: Limiting Dependencies	159
Limiting Surface Area	162
Dependency Injection	165
Inversion of Control	167
Summary	170
Chapter 10: The Model-View-Presenter (MVP) Model	173
Why MVP?	173
What Is MVP?	174
Constructing the MVP Application	177
Testing MVP Applications	184
Summary	187
Chapter 11: Tracing	189
Different Kinds of Messages	189
Log Sources and Log Sinks	190
Activities and Correlation IDs	192
Defining a Policy	193
Making Messages Actionable	196
Summary	198
Chapter 12: Error Handling	199
Result Code Reading	200
Exception Throwing	201
Importance of a Policy	203
Defining a Policy	204
Where to Handle Errors	207
Summary	210
Part IV: Putting It All Together	
Chapter 13: Calculator Project: A Case Study	213
Wrapping Up	220
Index	223