

# Table of Contents

<b>Introduction.....</b>	<b>1</b>
Why Git?.....	1
A Book for Professional Developers.....	3
About This Book.....	3
Why Workflows?.....	4
Tips for the Reader.....	5
Examples and Notations.....	6
Acknowledgments.....	8
Standing on the Shoulders of Giants.....	8
<b>Chapter 1: Basic Concepts.....</b>	<b>9</b>
Distributed Version Control, How Different?.....	9
The Repository, the Basis of Distributed Work.....	12
Branching and Merging, Easy!.....	14
Summary.....	16
<b>Chapter 2: Getting Started.....</b>	<b>19</b>
Setting up Git.....	19
Your First Git Project.....	19
Collaboration with Git.....	24
Summary.....	30
<b>Chapter 3: What Is A Commit?.....</b>	<b>33</b>
Access Permissions and Timestamps .....	34
The add and commit Commands.....	34
Revisiting the Commit Hash.....	35
The Commit History.....	36
A Slightly Different Way of Looking at Commits .....	37

Many Different Histories of the Same Project .....	38
Summary.....	41
<b>Chapter 4: Multiple Commits.....</b>	<b>43</b>
The status Command.....	44
The Staging Area Stores Snapshots .....	46
What To Do with Changes That Are Not To Be Committed? .....	49
Leaving out Unversioned Files with .gitignore.....	50
Stashing.....	51
Summary.....	52
<b>Chapter 5: The Repository.....</b>	<b>53</b>
A Simple and Efficient Storage System .....	53
Storing Directories: Blob & Tree.....	54
Identical Data Is Stored Only Once.....	55
Compressing Similar Content .....	56
Is It Bad When Various Files Happen to Get the Same Hash?.....	56
Commits.....	57
Object Reuse in the Commit History .....	57
Renaming, Moving and Copying.....	59
Summary .....	61
<b>Chapter 6: Branches.....</b>	<b>63</b>
Parallel Development .....	63
Bug Fixes in An Older Version.....	64
Branches.....	65
Swim Lanes.....	65
The Active Branch .....	66
Resetting A Branch Pointer .....	68
Deleting A Branch.....	69
Getting Rid of the Commit Objects.....	70
Summary .....	71
<b>Chapter 7: Merging Branches.....</b>	<b>73</b>
What Happens during A Merge? .....	74
Conflicts .....	76
Edit Conflicts.....	77
Conflict Markers .....	78
Resolving Edit Conflicts .....	79
What about the Content Conflict? .....	80
Fast-Forward Merges .....	81
First-Parent History .....	83

Tricky Merge Conflicts .....	85
Regardless, Somehow It Will Work.....	86
Summary.....	87
<b>Chapter 8: A Cleaner History with Rebasing.....</b>	<b>89</b>
The Principle: Copying of Commits.....	89
Avoiding the “Diamond Chain” .....	90
And When It Comes to Conflicts? .....	92
Transplanting A Branch.....	93
What Happens to the Original Commits after Rebasing? .....	95
Why Is It Problematic to Have the Original and Copy Commits in the Same Repository? .....	95
Cherry-Picking .....	96
Summary .....	97
<b>Chapter 9: Exchanges between Repositories.....</b>	<b>99</b>
Cloning A Repository.....	99
How to Tell Git Where the Other Repository Is.....	100
Giving the Other Repository A Name.....	101
Fetching Data.....	102
Remote-Tracking Branches: Monitoring Other Repositories.....	104
Working with Local Branches from Other Repositories.....	105
Pull = Fetch + Merge.....	106
For Diamond Haters: --rebase.....	106
Push, the Opposite of Pull .....	107
Naming Branches.....	109
Summary .....	110
<b>Chapter 10: Version Tagging.....</b>	<b>113</b>
Creating A Tag .....	113
Which Tags Are There? .....	114
Printing the Tag Hashes.....	114
Adding Tags to the Log Output.....	115
In What Version Is It in?.....	115
How to Change A Tag? .....	116
When Do I Need A Floating Tag? .....	116
Summary.....	116
<b>Chapter 11: Dependencies between Repositories.....</b>	<b>119</b>
Dependencies with Submodules.....	120
Dependencies with Subtrees .....	127
Summary.....	131

<b>Chapter 12: Tips and Tricks.....</b>	<b>133</b>
Don't Panic, There Is A Reflog!.....	133
Ignoring Local Changes Temporarily .....	134
Examining Changes to Text Files.....	135
alias - Shortcuts for Git Commands .....	136
Branches as Temporary Pointers to Commits.....	137
Moving Commits to Another Branch.....	138
<b>Chapter 13: Introduction to Workflows.....</b>	<b>141</b>
When Can I Use These Workflows?.....	142
Structure of the Workflows .....	144
<b>Chapter 14: Project Setup.....</b>	<b>147</b>
Overview.....	148
Requirements .....	149
Compact Workflow: Setting Up A Project.....	149
Process and Implementation .....	151
Why Not the Alternatives? .....	164
<b>Chapter 15: Developing on the Same Branch.....</b>	<b>167</b>
Overview .....	168
Requirements.....	169
Workflow: Developing on the Same Branch.....	169
Process and Implementation .....	170
Why Not the Alternatives?.....	174
<b>Chapter 16: Developing with Feature Branches.....</b>	<b>177</b>
Overview .....	178
Requirements.....	178
Workflow “Developing with Feature Branches”.....	179
Process and Implementation .....	180
Why not the Alternatives?.....	190
<b>Chapter 17: Troubleshooting with Bisection.....</b>	<b>199</b>
Overview .....	200
Requirements .....	200
Workflow “Troubleshooting with bisection”.....	201
Process and Implementation .....	201
Why Not the Alternatives?.....	210
<b>Chapter 18: Working with A Build Server.....</b>	<b>213</b>
Overview.....	213
Requirements.....	214
Workflow “Working with A Build Server”.....	215

Process and Implementation .....	216
Why Not the Alternatives?.....	227
<b>Chapter 19: Performing A Release.....</b>	<b>229</b>
Overview .....	230
Requirements .....	230
Workflow “Performing A Release” .....	231
Process and Implementation .....	232
Why Not the Alternatives? .....	241
<b>Chapter 20: Splitting A Large Project.....</b>	<b>245</b>
Overview.....	246
Requirements .....	247
Workflow “Splitting A Large Project”.....	247
Process and Implementation .....	248
Why Not the Alternatives?.....	253
<b>Chapter 21: Merging Small Projects.....</b>	<b>255</b>
Overview.....	255
Requirement.....	256
Workflow “Merging Small Projects”.....	256
Process and Implementation .....	258
Why Not the Alternatives?.....	261
<b>Chapter 22: Outsourcing Long Histories.....</b>	<b>263</b>
Overview .....	263
Requirements .....	264
Workflow “Outsourcing Long Histories”.....	265
Process and Implementation .....	266
Why Not the Alternatives?.....	272
<b>Chapter 23: Using Other Version Controls in Parallel.....</b>	<b>273</b>
Overview.....	274
Requirements .....	275
Workflow “Working with Other Version Controls in Parallel”.....	275
Process and Implementation .....	276
Why Not the Alternatives?.....	285
<b>Chapter 24: Migrating to Git.....</b>	<b>287</b>
Overview .....	287
Requirements .....	288
Workflow “Migrating to Git” .....	288
Process and Implementation .....	290
Why Not the Alternatives?.....	302

<b>Chapter 25: What Else Is There?</b>	<b>305</b>
Interactive Rebasing—Making the History Better	305
Dealing with Patches	306
Sending Patches by Email	307
Bundles—pull in Offline Mode	307
Creating An Archive	308
Graphical Tools for Git	308
Viewing A Repository with A Web Browser	310
Working with Subversion	310
Command Aliases	311
Notes on Commits	311
Extending Git with Hooks	312
Hosting Repositories on Github	312
<b>Chapter 26: Git's Shortcomings</b>	<b>315</b>
High Complexity	315
Complicated Submodules	317
Resource Consumption for Large Binary Files	318
Repositories Can Only Be Dealt with in Its Entirety	319
Authorization Only on the Entire Repository	319
Moderate Graphical Tools for History Analysis	321
<b>Index</b>	<b>323</b>