

Inhaltsverzeichnis

	Einleitung	11
	Aufbau des Buches	12
	Konvention	13
	Hinweise und Tipps	14
	Feedback	14
	Danksagung	14
1	Einführung	15
1.1	Lokale Versionsverwaltung	16
1.2	Zentrale Versionsverwaltung	17
1.3	Verteilte Versionsverwaltung	18
1.4	Geschichtliches	20
2	Die Grundlagen	23
2.1	Installation	23
2.2	Das erste Repository	26
2.3	Git-Konfiguration	28
2.4	Der erste Commit	29
	2.4.1 Versionierte Dateien mit »git mv« verschieben	46
2.5	Änderungen rückgängig machen mit Reset und Revert	47
	2.5.1 Revert	47
	2.5.2 Reset	48
2.6	Git mit GUI	51
	2.6.1 Commits mit Git GUI	53
2.7	Wie Git arbeitet	54
2.8	Git-Hilfe	60
2.9	Zusammenfassung	60
3	Arbeiten mit Branches	63
3.1	Allgemeines zum Branching	63
3.2	Branches anlegen	65
3.3	Branches mergen	72
3.4	Merge-Konflikte	76
3.5	Mergetools	80

3.6	Merge-Strategien	83
3.6.1	resolve	83
3.6.2	recursive	83
3.6.3	octopus	84
3.6.4	ours	84
3.6.5	subtree	84
3.7	Rebasing	84
3.8	Stash und Clean	90
3.8.1	Das Arbeitsverzeichnis säubern	94
3.8.2	Dateien ignorieren	96
3.9	Zusammenfassung	98
4	Verteilte Repositories	99
4.1	Projekt mit einem Remote-Repository	100
4.2	Branch-Management	109
4.3	Tracking-Branches	111
4.4	Projekt mit drei Remote-Repositories	114
4.5	Der Workflow mit drei Repositories	117
4.6	Zusammenfassung	121
5	Git-Hosting	123
5.1	GitHub	126
5.1.1	Repository anlegen	126
5.1.2	SSH-Keys anlegen und hinzufügen	129
5.1.3	SSH-Agent konfigurieren	131
5.1.4	Lokales Git-Repository konfigurieren	133
5.1.5	Repository klonen	135
5.1.6	Der GitHub-Workflow	135
5.1.7	GitHub-Repositories um externe Tools erweitern	152
5.2	GitLab	152
5.2.1	Installation	153
5.2.2	Konfiguration	153
5.3	Weitere Git-Hosting-Lösungen	158
5.4	CI/CD: Continuous Integration und Continuous Delivery	158
5.4.1	Der Workflow	159
5.4.2	GitHub Actions	161
5.4.3	GitLab CI/CD	164
5.5	Zusammenfassung	168

6	Workflows	169
6.1	Interaktives Rebasing	170
6.1.1	Branches pseudo-sichern	171
6.1.2	Den letzten Commit verändern	171
6.1.3	Mehrere Commits verändern	174
6.1.4	Reihenfolge der Commits anpassen	176
6.1.5	Commits ergänzen	176
6.1.6	Commits squashen	178
6.1.7	Commits autosquashen	180
6.1.8	Commits dropfen	181
6.1.9	Commit aufteilen	181
6.2	Workflow mit einem Branch und Repository für eine Person	183
6.3	Workflow mit mehreren Personen, einem Repository und einem Branch	184
6.4	Git Flow	186
6.4.1	Feature-Branches	187
6.4.2	Release-Branches	190
6.4.3	Release taggen	191
6.4.4	Hotfix-Branches	192
6.4.5	Zusammenfassung zu Git Flow	193
6.5	Git Flow mit mehr als einem develop-Branch	194
6.6	Git Flow mit mehreren Repositories	195
6.7	GitHub-Flow	197
6.8	GitLab-Flow	197
6.9	Weitere Aspekte in Workflows	199
6.10	Zusammenfassung	201
7	Hooks	203
7.1	Client-seitige Hooks	203
7.1.1	Commit-Hooks	204
7.1.2	E-Mail-Hooks	207
7.1.3	Weitere Hooks	207
7.2	Server-seitige Hooks	208
7.2.1	pre-receive-Hook	208
7.2.2	update-Hook	208
7.2.3	post-receive-Hook	209
7.2.4	Beispiel-Hooks	209
7.3	Git-Attribute	211

8	Umstieg von Subversion	215
8.1	Zentrale vs. verteilte Repositories	215
8.2	Checkout vs. Clone	216
8.3	svn commit vs. git commit & git push	216
8.4	svn add vs. git add	216
8.5	Binärdateien im Repository	217
8.6	SVN- in Git-Repository konvertieren	217
8.6.1	git-svn	218
8.6.2	Nach der Umwandlung	221
8.6.3	Committen mit git-svn	221
8.7	Zusammenfassung	223
9	Tipps und Tricks	225
9.1	Große Dateien mit Git LFS verwalten	225
9.2	Partielles Klonen	227
9.3	Aliasse setzen und nutzen	228
9.4	Mehr aus dem Log holen	229
9.4.1	Begrenzte Ausgaben	229
9.4.2	Schönere Logs	231
9.5	Ausgeführte Aktionen im Repository mit git reflog	232
9.6	Garbage Collection mit git gc	235
9.7	Finde den Schuldigen mit git blame	236
9.8	Wortweises diff mit word-diff	236
9.9	Verschobene Zeilen farblich hervorheben mit git diff --color-moved	237
9.10	Datei-Inhalte suchen mit git grep	238
9.11	Änderungen häppchenweise stagen und committen	239
9.12	Auf Fehlersuche mit git bisect	241
9.13	Arbeiten mit Patches	243
9.13.1	Patches erstellen	243
9.13.2	Patches anwenden	245
9.14	Repositories in Repositories mit git submodules	247
9.15	Subtree als Alternative für Submodule	250
9.16	Komplette Historie neu schreiben mit git filter-repo	252
9.17	Tippfehler in Git-Befehlen automatisch korrigieren	253
9.18	Git Worktree	254
9.19	Liquid Prompt für Git	256
9.19.1	Installation	256
9.19.2	Im Einsatz mit Git	257
9.20	Zusammenfassung	258

10	Grafische Clients	261
10.1	Git GUI.	261
10.2	Gitk.	263
10.3	SourceTree	267
10.4	GitHub Desktop.	269
10.5	Gitg.	271
10.6	Tig.	272
10.7	TortoiseGit	274
10.8	GitKraken.	276
10.9	Weiteres	277
11	Nachvollziehbare Git-Historien	279
11.1	Gut dosierte Commits.	279
11.2	Gute Commit-Messages	281
12	DevOps	289
12.1	DevOps im Überblick	289
12.2	Das Problem.	290
12.3	DevOps-Pipeline	294
12.4	DevSecOps	296
12.5	Zusammenfassung	298
13	Frequently Asked Questions	299
A	Befehlsreferenz	305
A.1	Repository und Arbeitsverzeichnis anlegen	305
A.2	Erweiterung und Bearbeitung der Historie	306
	A.2.1 Arbeiten im Staging-Bereich	306
	A.2.2 Arbeiten mit Commits und Branches.	307
A.3	Status-Ausgaben und Fehler-Suche	310
A.4	Verteilte Repositories	311
A.5	Hilfsbefehle	313
A.6	Sonstige	314
	Stichwortverzeichnis	317