

Inhaltsverzeichnis

	Danksagung	11
	Einleitung	13
1	Einführung	17
1.1	Versionsverwaltung	17
1.1.1	Lokale Versionsverwaltung	18
1.1.2	Zentrale Versionsverwaltung	19
1.1.3	Verteilte Versionsverwaltung	20
1.2	Geschichtliches	22
2	Die Grundlagen	23
2.1	Installation	23
2.2	Das erste Repository	25
2.3	Git-Konfiguration	27
2.4	Der erste Commit	28
2.4.1	Versionierte Dateien mit »git mv« verschieben	44
2.5	Änderungen rückgängig machen mit Reset und Revert	44
2.5.1	Revert	44
2.5.2	Reset	45
2.6	Git mit GUI	48
2.6.1	Commits mit Git GUI	50
2.7	Wie Git arbeitet	51
2.8	Git-Hilfe	56
2.9	Zusammenfassung	57
3	Arbeiten mit Branches	59
3.1	Allgemeines zum Branching	59
3.2	Branches anlegen	60
3.3	Branches mergen	67
3.4	Merge-Konflikte	71
3.5	Mergetools	75
3.6	Merge-Strategien	78
3.6.1	resolve	78

3.6.2	recursive	79
3.6.3	octopus	79
3.6.4	ours	79
3.6.5	subtree	79
3.7	Rebasing	80
3.8	Stash und Clean	85
3.8.1	Den Arbeitsordner säubern	89
3.8.2	Dateien ignorieren	91
3.9	Zusammenfassung	92
4	Verteilte Repositories	95
4.1	Projekt mit einem Remote-Repository	96
4.2	Branch-Management	103
4.3	Tracking-Branches	105
4.4	Projekt mit drei Remote-Repositories	108
4.5	Der Workflow mit drei Repositories	110
4.6	Zusammenfassung	113
5	Git-Hosting	115
5.1	GitHub	117
5.1.1	Repository anlegen	117
5.1.2	SSH-Keys anlegen und hinzufügen	120
5.1.3	SSH-Agent konfigurieren	122
5.1.4	Lokales Git-Repository konfigurieren	124
5.1.5	Repository klonen	125
5.1.6	Der GitHub-Workflow	126
5.1.7	GitHub-Repositories um externe Tools erweitern	140
5.2	GitLab	141
5.2.1	Installation	141
5.2.2	Konfiguration	141
5.3	gitolite	145
5.4	Weitere Git-Hosting-Lösungen	146
5.5	3rd-Party-Tools für Continuous Integration	146
5.5.1	Der Workflow	147
5.5.2	Jenkins	148
5.5.3	Travis-CI	153
5.5.4	GitLab-CI	155
5.6	Zusammenfassung	157

6	Workflows	159
6.1	Interaktives Rebasing	159
6.1.1	Branches pseudo-sichern	160
6.1.2	Den letzten Commit verändern	161
6.1.3	Mehrere Commits verändern	163
6.1.4	Reihenfolge der Commits anpassen	165
6.1.5	Commits ergänzen	166
6.1.6	Commits squashen	167
6.1.7	Commits autosquashen	169
6.1.8	Commits dropfen	170
6.1.9	Commit aufteilen	170
6.2	Workflow mit einem Branch und Repository für eine Person	171
6.3	Workflow mit mehreren Personen, einem Repository und einem Branch	173
6.4	Git Flow	175
6.4.1	Feature-Branches	176
6.4.2	Release-Branches	178
6.4.3	Release taggen	179
6.4.4	Hotfix-Branches	180
6.4.5	Zusammenfassung zu Git Flow	181
6.5	Git Flow mit mehr als einem develop-Branch	182
6.6	GitHub-Workflow	183
6.7	Git Flow mit mehreren Repositories	184
6.8	Git-Flow-Alternative	185
6.9	Auswahl des Workflows	186
7	Hooks	187
7.1	Client-seitige Hooks	187
7.1.1	Commit-Hooks	188
7.1.2	E-Mail-Hooks	190
7.1.3	Weitere Hooks	191
7.2	Server-seitige Hooks	191
7.2.1	pre-receive-Hook	192
7.2.2	update-Hook	192
7.2.3	post-receive-Hook	192
7.2.4	Beispiel-Hooks	192
7.3	Git-Attribute	194

8	Umstieg von Subversion	197
8.1	Zentrale vs. verteilte Repositories	197
8.2	Checkout vs. Clone	198
8.3	svn commit vs. git commit & git push	198
8.4	svn add vs. git add	198
8.5	Binärdateien im Repository	199
8.6	SVN- in Git-Repository konvertieren	199
8.6.1	git-svn	199
8.6.2	Nach der Umwandlung	202
8.6.3	Committen mit git-svn	203
8.7	Zusammenfassung	204
9	Tipps und Tricks	205
9.1	Alias setzen und nutzen	205
9.2	Mehr aus dem Log holen	206
9.2.1	Begrenzte Ausgaben	206
9.2.2	Schönere Logs	208
9.3	Ausgeführte Aktionen im Repository mit git reflog	209
9.4	Garbage Collection mit git gc	212
9.5	Finde den Schuldigen mit git blame	212
9.6	Wortweises diff mit word-diff	213
9.7	Datei-Inhalte suchen mit git grep	214
9.8	Änderungen häppchenweise stagen und committen	215
9.9	Auf Fehlersuche mit git bisect	217
9.10	Arbeiten mit Patches	219
9.10.1	Patches erstellen	219
9.10.2	Patches anwenden	221
9.11	Repositories in Repositories mit git submodules	223
9.12	Komplette Historie neu schreiben mit git filter-branch	226
9.13	Notizen erstellen mit git notes	228
9.14	Tippfehler in Git-Befehlen automatisch korrigieren	229
9.15	Liquid Prompt für Git	230
9.15.1	Installation	230
9.15.2	Im Einsatz mit Git	231
9.16	Zusammenfassung	232
10	Grafische Clients	235
10.1	Git GUI	235
10.2	Gitk	237

10.3	SourceTree	240
10.4	GitHub Desktop	243
10.5	Gitg	244
10.6	Tig	245
10.7	TortoiseGit	247
10.8	GitKraken	249
10.9	Weiteres	250
A	Befehlsreferenz	251
A.1	Repository und Arbeitsverzeichnis anlegen	251
A.2	Erweiterung und Bearbeitung der Historie	252
	A.2.1 Arbeiten im Staging-Bereich	252
	A.2.2 Arbeiten mit Commits und Branches	253
A.3	Statusausgaben und Fehlersuche	256
A.4	Verteilte Repositories	257
A.5	Hilfsbefehle	258
A.6	Sonstige	259
	Stichwortverzeichnis	261