

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	25
2.1	Installation	25
2.2	Das erste Repository	28
2.3	Git-Konfiguration	29
2.4	Der erste Commit	30
2.4.1	Versionierte Dateien mit »git mv« verschieben	47
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	52
2.7	Wie Git arbeitet	54
2.8	Git-Hilfe	59
2.9	Zusammenfassung	59
3	Arbeiten mit Branches	61
3.1	Allgemeines zum Branching	61
3.2	Branches anlegen	62
3.3	Branches mergen	69
3.4	Merge-Konflikte	73
3.5	Mergetools	77

3.6	Merge-Strategien	80
3.6.1	resolve	80
3.6.2	recursive	81
3.6.3	octopus.....	81
3.6.4	ours	81
3.6.5	subtree.....	81
3.7	Rebasing	82
3.8	Stash und Clean.....	87
3.8.1	Den Arbeitsordner säubern	91
3.8.2	Dateien ignorieren	93
3.9	Zusammenfassung	94
4	Verteilte Repositorys	97
4.1	Projekt mit einem Remote-Repository	98
4.2	Branch-Management	107
4.3	Tracking-Branches	109
4.4	Projekt mit drei Remote-Repositorys	112
4.5	Der Workflow mit drei Repositorys	114
4.6	Zusammenfassung	118
5	Git-Hosting.....	119
5.1	GitHub	121
5.1.1	Repository anlegen	121
5.1.2	SSH-Keys anlegen und hinzufügen	124
5.1.3	SSH-Agent konfigurieren	126
5.1.4	Lokales Git-Repository konfigurieren	128
5.1.5	Repository klonen.....	129
5.1.6	Der GitHub-Workflow	130
5.1.7	GitHub-Repositorys um externe Tools erweitern.....	146
5.2	GitLab.....	146
5.2.1	Installation	146
5.2.2	Konfiguration	147
5.3	gitolite.....	150
5.4	Weitere Git-Hosting-Lösungen.....	151
5.5	3rd-Party-Tools für Continuous Integration	152
5.5.1	Der Workflow	153
5.5.2	Jenkins.....	154
5.5.3	Travis-CI	158
5.5.4	GitLab-CI.....	161
5.6	Zusammenfassung	164

6	Workflows	165
6.1	Interaktives Rebasing	165
6.1.1	Branches pseudo-sichern	166
6.1.2	Den letzten Commit verändern	167
6.1.3	Mehrere Commits verändern	169
6.1.4	Reihenfolge der Commits anpassen	171
6.1.5	Commits ergänzen	172
6.1.6	Commits squashen	173
6.1.7	Commits autosquashen	175
6.1.8	Commits dropen	176
6.1.9	Commit aufteilen	176
6.2	Workflow mit einem Branch und Repository für eine Person	177
6.3	Workflow mit mehreren Personen, einem Repository und einem Branch	179
6.4	Git Flow	181
6.4.1	Feature-Branches	182
6.4.2	Release-Branches	184
6.4.3	Release taggen	185
6.4.4	Hotfix-Branches	186
6.4.5	Zusammenfassung zu Git Flow	187
6.5	Git Flow mit mehr als einem develop-Branch	188
6.6	GitHub-Workflow	189
6.7	Git Flow mit mehreren Repositories	190
6.8	Git-Flow-Alternative	191
6.9	Auswahl des Workflows	192
7	Hooks	193
7.1	Client-seitige Hooks	193
7.1.1	Commit-Hooks	194
7.1.2	E-Mail-Hooks	196
7.1.3	Weitere Hooks	197
7.2	Server-seitige Hooks	197
7.2.1	pre-receive-Hook	198
7.2.2	update-Hook	198
7.2.3	post-receive-Hook	198
7.2.4	Beispiel-Hooks	198
7.3	Git-Attribute	200

8	Umstieg von Subversion	203
8.1	Zentrale vs. verteilte Repositories	203
8.2	Checkout vs. Clone	204
8.3	svn commit vs. git commit & git push	204
8.4	svn add vs. git add	204
8.5	Binärdateien im Repository	205
8.6	SVN- in Git-Repository konvertieren	205
	8.6.1 git-svn	205
	8.6.2 Nach der Umwandlung	208
	8.6.3 Commmitten mit git-svn	209
8.7	Zusammenfassung	210
9	Tipps und Tricks	211
9.1	Aliasse setzen und nutzen	211
9.2	Mehr aus dem Log holen	212
	9.2.1 Begrenzte Ausgaben	212
	9.2.2 Schönere Logs	214
9.3	Ausgeführte Aktionen im Repository mit git reflog	215
9.4	Garbage Collection mit git gc	218
9.5	Finde den Schuldigen mit git blame	218
9.6	Wortweises diff mit word-diff	219
9.7	Datei-Inhalte suchen mit git grep	220
9.8	Änderungen häppchenweise stagen und committen	221
9.9	Auf Fehlersuche mit git bisect	223
9.10	Arbeiten mit Patches	225
	9.10.1 Patches erstellen	225
	9.10.2 Patches anwenden	227
9.11	Repositories in Repositories mit git submodules	229
9.12	Komplette Historie neu schreiben mit git filter-branch	232
9.13	Tippfehler in Git-Befehlen automatisch korrigieren	234
9.14	Git Worktree	235
9.15	Liquid Prompt für Git	237
	9.15.1 Installation	237
	9.15.2 Im Einsatz mit Git	238
9.16	Zusammenfassung	239
10	Grafische Clients	241
10.1	Git GUI	241
10.2	Gitk	243

10.3	SourceTree	246
10.4	GitHub Desktop	248
10.5	Gitg	249
10.6	Tig	250
10.7	TortoiseGit	252
10.8	GitKraken	254
10.9	Weiteres	255
11	Nachvollziehbare Git-Historien	257
11.1	Gut dosierte Commits	257
11.2	Gute Commit-Messages	259
12	Frequently Asked Questions	267
A	Befehlsreferenz	271
A.1	Repository und Arbeitsverzeichnis anlegen	271
A.2	Erweiterung und Bearbeitung der Historie	272
	A.2.1 Arbeiten im Staging-Bereich	272
	A.2.2 Arbeiten mit Commits und Branches	273
A.3	Statusausgaben und Fehlersuche	276
A.4	Verteilte Repositories	277
A.5	Hilfsbefehle	278
A.6	Sonstige	279
	Stichwortverzeichnis	281