

# INHALTSVERZEICHNIS

ZUM GELEIT	IX
VORWORT	XI
<b>1 EINFÜHRUNG</b>	<b>1</b>
<i>Warum ein Buch über Produktivität?</i>	2
<i>Worum es in diesem Buch geht</i>	4
<i>Und wie geht es weiter?</i>	7
<b>Teil 1</b> <b>MECHANISMEN</b>	<b>9</b>
<hr/>	
<b>2 BESCHLEUNIGUNG</b>	<b>11</b>
<i>Launcher</i>	12
<i>Beschleuniger</i>	24
<i>Makros</i>	44
<i>Zusammenfassung</i>	47
<b>3 FOKUS</b>	<b>49</b>
<i>Ablenkungen beseitigen</i>	50
<i>Suche übertrumpft Navigation</i>	53
<i>Dinge suchen, die schwer zu finden sind</i>	56
<i>Verwenden Sie Rooted Views</i>	59
<i>Verwenden Sie dauerhafte Attribute</i>	61
<i>Verwenden Sie projektbezogene Verknüpfungen</i>	62
<i>Multiplizieren Sie Ihre Monitore</i>	63
<i>Teilen Sie Ihren Arbeitsbereich virtuell auf</i>	63
<i>Zusammenfassung</i>	66
<b>4 AUTOMATISIERUNG</b>	<b>67</b>
<i>Erfinden Sie das Rad nicht neu</i>	69
<i>Speichern Sie Dinge lokal</i>	70
<i>Automatisieren Sie die Interaktion mit Websites</i>	71
<i>Interagieren Sie mit RSS-Feeds</i>	72
<i>Mit Ant nicht nur Build-Prozesse steuern</i>	73

	<i>Rake für allgemeine Aufgaben verwenden</i>	75
	<i>Mit Selenium Webseiten auslesen</i>	77
	<i>Mit Bash Exceptions zählen</i>	79
	<i>Ersetze BAT durch PowerShell</i>	80
	<i>Mit Mac OS X Automator alte Downloads löschen</i>	81
	<i>Die Subversion-Befehlszeile zähmen</i>	83
	<i>Einen SQL-Splitter mit Ruby bauen</i>	84
	<i>Warum Automatisierung wichtig ist</i>	86
	<i>Rasieren Sie keine Grunzochsen</i>	89
	<i>Zusammenfassung</i>	90
<b>5</b>	<b>KANONITÄT</b>	91
	<i>DRY in der Versionskontrolle</i>	93
	<i>Verwenden Sie einen kanonischen Build-Server</i>	95
	<i>Indirektion</i>	97
	<i>Nutzen Sie die Virtualisierung</i>	106
	<i>DRY und der Impedance-Mismatch</i>	107
	<i>DRY in der Dokumentation</i>	116
	<i>Zusammenfassung</i>	124
<b>Teil 2</b>	<b>PRAXIS</b>	125
<hr/>		
<b>6</b>	<b>TESTGETRIEBENES DESIGN</b>	127
	<i>Tests entstehen lassen</i>	129
	<i>Testabdeckung</i>	137
<b>7</b>	<b>STATISCHE ANALYSE</b>	141
	<i>Bytecode-Analyse</i>	142
	<i>Quellcode-Analyse</i>	145
	<i>Metriken generieren mit Panopticode</i>	147
	<i>Analyse dynamischer Sprachen</i>	150
<b>8</b>	<b>GUTE MITBÜRGER</b>	153
	<i>Die Kapselung durchbrechen</i>	154
	<i>Konstruktoren</i>	156
	<i>Statische Methoden</i>	156
	<i>Kriminelles Verhalten</i>	162
<b>9</b>	<b>YAGNI</b>	165

<b>10</b>	ALTE PHILOSOPHEN	173
	<i>Aristoteles' essenzielle und akzidenzielle Eigenschaften</i>	174
	<i>Ockhams Skalpell</i>	176
	<i>Das Gesetz der Demeter</i>	181
	<i>Softwareüberlieferungen</i>	182
<b>11</b>	AUTORITÄTEN INFRAGE STELLEN	185
	<i>Böse Affen</i>	186
	<i>Fluent-Interfaces</i>	188
	<i>Antiobjekte</i>	190
<b>12</b>	META-PROGRAMMIERUNG	193
	<i>Java und Reflection</i>	194
	<i>Java mit Groovy testen</i>	196
	<i>Fluent-Interfaces schreiben</i>	198
	<i>Wohin führt uns Meta-Programmierung?</i>	200
<b>13</b>	COMPOSED-METHOD UND SLAP	201
	<i>Composed-Method im Einsatz</i>	202
	<i>SLAP</i>	208
<b>14</b>	POLYGLOTTES PROGRAMMIEREN	215
	<i>Wie sind wir hierher gekommen? Und was heißt »hier«?</i>	216
	<i>Wo gehen wir hin? Und wie kommen wir dort hin?</i>	220
	<i>Olas Pyramide</i>	226
<b>15</b>	PERFEKTE WERKZEUGE	229
	<i>Das Streben nach dem perfekten Editor</i>	230
	<i>Die Kandidaten</i>	234
	<i>Das richtige Werkzeug für meinen Job</i>	236
	<i>Die falschen Werkzeuge vermeiden</i>	244
<b>16</b>	SCHLUSSFOLGERUNG: DAS GESPRÄCH FORTFÜHREN	249
	ANHANG: BAUSTEINE	253
	INDEX	263