

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>9</b>
<b>Stand der Forschung</b>	<b>13</b>
Open Source Software, ein viel beforschtes Feld	13
Soziologische Analysen des Feldes	16
<b>Präzisierung der Fragestellung</b>	<b>23</b>
<b>Struktur der Arbeit</b>	<b>24</b>

## Teil I

### Software: Entstehung und Problematisierung

<b>1</b>	<b>Eine kleine Geschichte der Software</b>	<b>29</b>
<b>1.1</b>	<b>Anfänge von Computer und Software</b>	<b>29</b>
<b>1.2</b>	<b>Kommodifizierung von Software und der Reflex der Community</b>	<b>34</b>
<b>1.3</b>	<b>Popularisierung und Expansion offener Software-Produktion</b>	<b>35</b>
<b>1.4</b>	<b>Legitimationsschwierigkeiten der Expert*innen</b>	<b>39</b>
<b>2</b>	<b>Implikationen von Technik und Software</b>	<b>43</b>
<b>2.1</b>	<b>Techno-Regulation und die Rolle von Software</b>	<b>44</b>
<b>2.2</b>	<b>Software wird kritisch: Critical Algorithm Studies</b>	<b>49</b>
<b>2.3</b>	<b>Surveillance Studies und die Rolle von Algorithmen</b>	<b>51</b>
<b>3</b>	<b>Software als Element moderner Technologien</b>	<b>55</b>
<b>3.1</b>	<b>„Jenseits des mechanischen Bewirkens“</b>	<b>55</b>
<b>3.2</b>	<b>Spezifische Merkmale von Software</b>	<b>58</b>

## Teil II

### Epistemische Regime zur Regulierung der Wissensdifferenz

<b>4</b>	<b>Technik als soziales Verhältnis zwischen Expert*innen und Laien</b>	<b>69</b>
4.1	Technik als stabiler Wirkungszusammenhang	63
4.2	Vertrauen in Technik	64
4.3	Die Macht der Expert*innen	69
<b>5</b>	<b>Differenzierte Expertisegrade und die Erweiterung der Partizipation</b>	<b>75</b>
5.1	Abstufungen einer „Specialist Expertise“	76
5.2	Das Selektionsproblem erweiterter Partizipation	80
<b>6</b>	<b>Die Regulierung der Wissensproduktion durch epistemische Regime</b>	<b>85</b>
6.1	Epistemische Regime: Vorarbeiten und Konzeption des Begriffs	86
6.2	Die soziale Ordnung der Wissensproduktion	89
6.3	Dimensionen epistemischer Regime	91
6.4	Operationalisierung zur Analyse epistemischer Regime der Softwareproduktion	96

## Teil III

### GNU/Linux Communities als epistemische Regime

<b>7</b>	<b>Untersuchungsdesign</b>	<b>107</b>
7.1	Distributionen als Querschnitt verschiedener „Programmierkulturen“	103
7.1.1	„Programmierkulturen“	103
7.1.2	Charakterisierung von Distributionen	105
7.1.3	Auswahl und Beschreibung der untersuchten Distributionen	108
7.2	Sampling	112

<b>7.3</b>	<b>Auswertungsmethode</b>	<b>115</b>
<b>8</b>	<b>Epistemische Regime der gemeinschaftlichen Softwareproduktion</b>	<b>119</b>
<b>8.1</b>	<b>Mitgliedschaft als kollektives Selbstbild</b>	<b>121</b>
8.1.1	Ubuntu – „Linux for Human Beings“	122
8.1.2	Debian – „The Universal Operating System“	129
8.1.3	Arch – „A Simple and Lightweight Distribution“	140
8.1.4	Vergleich der Variable „Mitgliedschaft“	148
<b>8.2</b>	<b>Strukturen der Community</b>	<b>153</b>
8.2.1	Ubuntu – zwischen Community und Unternehmen	155
8.2.2	Debian – Demokratie der Entwickler*innen	164
8.2.3	Arch – informelle Oligarchie	170
8.2.4	Vergleich der Variable „Strukturen und Verfahren“	175
<b>8.3</b>	<b>Beiträge zum gemeinsamen Wissensprodukt</b>	<b>184</b>
8.3.1	Ubuntu – „Whatever skills“: Bugs und Brötchen	186
8.3.2	Debian – Developer unter sich	193
8.3.3	Arch – Userdeveloper und Developeruser	199
8.3.4	Vergleich der Variable „Beiträge“	203
<b>9</b>	<b>Normative Inskriptionen in Software</b>	<b>211</b>
<b>9.1</b>	<b>Ubuntu Linux: In wenigen Schritten zum Betriebssystem</b>	<b>214</b>
9.1.1	Download: „Help shape the future of Ubuntu“	215
9.1.2	Ubuntu: Ausprobieren und installieren	217
9.1.3	Inskription des Ubuntu-Installers	224
<b>9.2</b>	<b>Debian Linux: Zwischen Anfänger- und Expertenoptionen</b>	<b>225</b>
9.2.1	How to get the Debian installer	226
9.2.2	Debian Installer	227
9.2.3	Inskriptionen des Debian-Installers	236
<b>9.3</b>	<b>Arch Linux: Volle Kontrolle – do it yourself</b>	<b>237</b>
9.3.1	Getting and Installing Arch	238
9.3.2	Being the Install-Script	239
9.3.3	Inskription des Arch-Installers	246
<b>9.4</b>	<b>Zusammenfassende Gegenüberstellung der Skripte</b>	<b>247</b>

## Teil IV

### Normative Konfigurationen

<b>10</b>	<b>Zusammenfassung der Ergebnisse</b>	<b>261</b>
<b>10.1</b>	<b>Spezifische (normative) epistemische Regime</b>	<b>255</b>
10.1.1	Usability Regime: Institutionalisierte Inklusion	257
10.1.2	Stability Regime: Demokratie qualifizierter Entwickler*innen	260
10.1.3	Minimales Eliten-Regime	264
<b>10.2</b>	<b>Das Zusammenspiel der Variablen epistemischer Regime</b>	<b>267</b>
10.2.1	Mitgliedschaft ist Selektionsmechanismus und Identitätsmoment	268
10.2.2	Strukturen und Verfahren als formale zentrale Governance-Struktur	270
10.2.3	Die Selektion der Beiträge strukturiert das Wissensprodukt	271
10.2.4	Normative Prägungen der Regime	273
<b>10.3</b>	<b>Software: Das Wissensprodukt als Spiegel und Selektionsmechanismus der Community</b>	<b>274</b>
<b>11</b>	<b>Epistemische Regime verschiedener Experten-Laien-Differenzen</b>	<b>281</b>
<b>11.1</b>	<b>Epistemische Regime als stabile normative Ordnungen</b>	<b>281</b>
<b>11.2</b>	<b>Die epistemischen Rekonfigurationen der Experten-Laien-Differenz</b>	<b>285</b>
11.2.1	Differenzierungen zwischen den Entwickler*innen	287
11.2.2	Beiträge jenseits einer „Contributory Expertise“	290
11.2.3	Differenzierungen von Wissen jenseits der „Interactional Expertise“	292
<b>11.3</b>	<b>Spielräume zwischen Experten-Entwickler und Laien-User</b>	<b>293</b>
11.3.1	Entwickler-User-Differenz	294
11.3.2	Experten-Laien-Differenz	297
<b>12</b>	<b>Schlussfolgerungen</b>	<b>303</b>
	<b>Danksagung</b>	<b>307</b>
	<b>Literatur</b>	<b>311</b>