

Software gibt es doch

Marcus Burkhardt

Abstract:

In den Medien- und Kulturwissenschaften schenkte man Software als Untersuchungsgegenstand bisher kaum Beachtung. Das von Matthew Fuller herausgegebene Lexikon *Software Studies* schickt sich an, diesen Missstand auszuräumen und den Lesern in insgesamt 39 Artikeln den "'stuff' of software" (S. 1) nahe zu bringen. Durch das Aufzeigen der mannigfachen Verflechtungen von Software, Kultur und Gesellschaft erschließt das Lexikon nicht nur ein neues Forschungsfeld, sondern regt zum kritischen Weiterdenken und -forschen an.

How to cite:

Burkhardt, Marcus: „Software gibt es doch [Review on: Fuller, Matthew (Hg.): *Software Studies. A Lexicon*. Cambridge: MIT Press, 2008.]“. In: *KULT_online* 19 (2009).

DOI: <https://doi.org/10.22029/ko.2009.440>

© beim Autor und bei KULT_online

Software gibt es doch

Marcus Burkhardt

Fuller, Matthew (Hg.): Software Studies. A Lexicon. Cambridge: MIT Press, 2008. 334 Seiten, gebunden, 30,99 Euro. ISBN: 978-0-262-06274-9

"Es gibt keine Software". Mit dieser These konfrontierte Friedrich Kittler 1993 seine Leser in einem gleichnamigen Aufsatz. Was Software genannt wird, sei bloß eine Imagination, die den Blick auf das eigentlich wichtige – die Hardware – verdecke. Hardware nämlich bilde die materielle Basis computervermittelter Kommunikation und Weltaneignung, und auf diese Materialität der Kommunikation komme es, so Kittler, schließlich an. Vielleicht liegt hierin begründet, dass die Kultur- und Medienwissenschaften Software seither nur wenig Aufmerksamkeit geschenkt haben. Doch allem Unken Kittlers zum Trotz – nutzen wir heute Computer, dann haben wir es stets mit Software zu tun, und nicht zuletzt deshalb ist Software ein lohnenswertes Thema für die Kulturwissenschaften. Dies zumindest ist die Überzeugung der Autoren des Lexikons Software Studies, welches der Kulturwissenschaftler Matthew Fuller im August 2008 bei MIT Press herausgab.

In insgesamt 39 Beiträgen werden zentrale Begriffe zu Software und zur Auseinandersetzung mit dieser im Stile eines Lexikons eingeführt. Aus dem Weg geräumt werden soll dabei die falsche Vorstellung von der Immaterialität von Software. Zwar reichen die Wurzeln der Software, dies gesteht Fuller in seiner Einleitung zu dem Lexikon gern ein, nicht so weit zurück wie die der Computerhardware, aber hierin sei kein Grund zu sehen, ihr jegliche Materialität abzuspochen. So ist die Grundthese des Lexikons, dass Software entgegen der weitverbreiteten Meinung durchaus eine gewisse Materialität zukommt, die sich auf verschiedenen Ebenen manifestiert und wirksam wird. Im Zentrum stehen dabei die besonderen Charakteristika von Programmiersprachen und Interfaces. Dabei wird gefragt, welche Formen der Programmierbarkeit und des Gebrauchs diese ermöglichen und welche Effekte die Verwendung verschiedener informatischer Konzepte haben. Schließlich läuft diese Auseinandersetzung mit Software auf die Frage hinaus, wie sich aus dem Zusammenspiel von Mensch und programmierbarer sowie programmierter Maschine neue soziale Formationen und Figuren des Wissens herausbilden (vgl. S. 4). Insofern steht das Lexikon durchaus in der Tradition medienmaterialistischer Ansätze wie dem Friedrich Kittlers, was vielleicht auch erklärt, warum Kittler selbst mit einem Beitrag zum Begriff "Code" in dem Band vertreten ist. Doch Software Studies geht über den Medienmaterialismus hinaus. So dienen die Science and Technology Studies als weiterer theoretischer und methodischer Impulsgeber, um sich dem Gegenstand anzunähern. Software wird als etwas begriffen, das eine eigene Geschichte hat und somit nicht nur den

harten, weil technologischen Gesetzen unterliegt. Software ist vielmehr ein kontingentes Produkt, welches durch Medientechnologie ebenso bedingt ist wie durch soziale, institutionelle und kulturelle Rahmenbedingungen. Dies wird, so Fuller, vor allem im Bereich der Software Art deutlich, die sich deshalb als geeignetes Untersuchungsfeld erweist, um über die Verflechtungen von Technologie, Kultur und Gesellschaft, die durch den Begriff Software markiert werden, zu reflektieren.

Ziel des Lexikons ist es, wie Fuller darlegt, den Lesern den "'stuff' of software" (S. 1) näherzubringen. Vordringliche Aufgabe ist dabei, die verschiedenen Ebenen, auf denen sich Software manifestiert, auf denen sie erfahren und durchdacht werden kann, freizulegen und das Wechselspiel zwischen diesen Ebenen aufzuzeigen. Dabei gilt es insbesondere die Bedingungen und Möglichkeiten zu reflektieren, unter die Software unser Verhältnis zur Welt stellt. Diesem Anspruch folgend, finden sich Beiträge zu den konzeptuellen und technologischen Grundlagen des Computers wie Algorithmen, Code, Information, (Programmier-)Sprachen, Funktionen, objektorientierter Programmierung, Listen, Loops, Speicher und Variablen ebenso wie Beiträge zu softwarebasierten Funktionen, Praktiken und Phänomenen wie zum Beispiel dem Kopierbefehl, Import/Export-Routinen, Buttons oder auch den Systemsounds. Darüber hinaus versammelt das Lexikon Beiträge zu Text-Viren, Glitches und zu Obfuscated Code in denen Brüche, Defekte und Grenzphänomene der Computerkultur thematisiert werden. Fragen der transkulturellen Computernutzung werden schließlich in den Einträgen "Internationalization" und "Ethnocomputing" behandelt.

Entsprechend der von Matthew Fuller in der Einleitung dargelegten Grundsätze kreuzen die Autoren aller Beiträge eine Beschreibung technischer Prinzipien und Prozeduren mit einer Analyse der kulturellen Bedeutung und tragen hierdurch auf je eigene Weise der Bedeutung von Software in unserer und für unsere Kultur Rechnung. Auf eindrucksvolle Weise gelingt es so, dem Leser ein breitgefächertes Panorama der Verflechtung von Kultur und Software zu präsentieren.

Zu kurz kommt in dem Band jedoch die Diskussion über Methoden und Theorien, die die unter dem Titel Software Studies angeregte Forschungsrichtung theoretisch fundieren. Neben der Einleitung Fullers stellt der von Matti Tedre und Ron Eglash verfasste Beitrag zu "Ethnocomputing" diesbezüglich eine der wenigen Ausnahmen in dem Band dar. Die beiden Autoren präsentieren weniger fertige Analysen, als dass sie vielmehr Perspektiven und Fragen für eine künftige Erforschung von Software formulieren. Damit tragen sie der Tatsache Rechnung, dass Software Studies ein noch wenig bearbeitetes Untersuchungsfeld ist. So bedarf es nicht nur der Behandlung konkreter Phänomene und Aspekte von Software, sondern auch der Diskussion von theoretischen Zugängen, Methoden und Fragestellungen, die einerseits die künftige Auseinandersetzung mit Software leiten können und andererseits die aktuellen Arbeiten zu diesem Thema im mittlerweile nahezu unüberschaubaren Gemenge medienkulturwissenschaftlicher Publikationen zu digitalen Medien verorten. Hierin liegt dann auch das Problem des Lexikons. Es präsentiert einen bunten Blumenstrauß unterschiedlicher Zugänge und Perspektiven auf ein Thema, über das man bisher noch wenig weiß. Das Format eines Lexikons

scheint nicht unbedingt geeignet, um sich einem neuen Gegenstand anzunähern. Diese Vermutung erhärtet sich bei der Lektüre, bei der erhebliche Lücken in den behandelten Stichworten auffallen. So fehlen die Stichworte Prozedurale Programmierung und Datenbankmanagementsystem ebenso wie Beiträge zu Open Source und Daten. Diese Schwäche lässt sich jedoch zum Vorteil wenden, wenn man dem Lexikon von vorherein seine Unvollständigkeit und Vorläufigkeit zugesteht, wie es Fuller in der Einleitung explizit einfordert. Denn dann regt die Lektüre der Beiträge zum kritischen Weiterdenken und -forschen an und kann so neue Impulse für eine medien- und kulturwissenschaftliche Beschäftigung mit digitalen Medien geben.