

Zur Ausblendung sozialer Dimensionen im Qualitätsbegriff der Informatik

Sara John und Doris Allhutter

Einleitung

Trotz zahlreicher feministisch-technikkritischer Beiträge zur Entwicklung technologischer Artefakte als soziokulturelle und vergeschlechtlichte Prozesse gibt es bis dato im deutschsprachigen Raum kaum konkrete Analysen von technologischen Entwicklungsprozessen vor diesem theoretischen Hintergrund. Vielmehr scheint die Kluft zwischen theoretischen Gender-Diskursen und deren Anwendung in der Softwareentwicklung zu wachsen. Zwar wurde feministische Kritik insofern aufgegriffen, als mittlerweile partizipative Entwicklungsmethoden, die – auch aus einer geschlechterkritischen Perspektive – auf Qualitätsmerkmale wie Usability, Brauchbarkeit und Design eingehen, durchaus anerkannt sind, doch impliziert das Verständnis von Artefakten als sozial verhandelte Konstruktionen eine tiefer gehende Infragestellung zentraler Paradigmen des technologischen Entwicklungsprozesses und des ihm zugrunde liegenden Qualitätsbegriffs.

Traditionelle technikzentrierte Qualitätsstandards blenden soziale Dimensionen, wie etwa vergeschlechtlichte Annahmen über zukünftige NutzerInnen oder gesellschaftliche Konstruktionen, die in der Gestaltung von Software wirksam werden und sich in technologischen Artefakten materialisieren, aus.

Ziel des an der Wirtschaftsuniversität Wien durchgeführten Forschungsprojekts »Gendered Software Design« war es daher, unreflektierte soziale Konstruktionen und damit implizite Qualitätsaspekte in Entwicklungsprozessen sichtbar zu machen (vgl. Allhutter/Hanappi-Egger/John 2007). Konkret wurde untersucht, wie sich Geschlechter-Diskurse über Designentscheidungen in die Entwicklung von Computerspielen und Suchmaschinen einschreiben.

Softwarequalität als technikzentrierter Begriff

Seit den 1970er Jahren ist in der Informatik eine Verschiebung von traditionell ingenieurwissenschaftlichen zu stärker kontextbezogenen, partizipativen und evolutionären Zugängen zu beobachten. Begriffe wie User Centered Design, Human Factors Engineering oder Usability Engineering wurden bereits in den frühen 1980er Jahren geprägt, allerdings ist die Einbindung von Usability Engineering in Entwicklungsprozesse nach wie vor unzureichend (Petrasch 2001 a, 3). Ausgehend von verschiedenen Software-Engineering Konzepten greifen davon abgeleitete Qualitätsstandards auf unterschiedliche Bezugssysteme zurück. Jørgensen (1999, 909) nennt die folgenden drei Qualitätsdefinitionen:

- Softwarequalität ist die »Gesamtheit von Merkmalen und Merkmalswerten eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen« (gemäß ISO 9126),

- Softwarequalität wird an der Zufriedenheit der UserInnen gemessen, oder
- Softwarequalität wird anhand der Fehler oder des unerwarteten Verhaltens der Software gemessen.

Im Zentrum des ersten Qualitätsbegriffs stehen demnach Qualitätsmerkmale wie Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz und Übertragbarkeit. Die zweite Definition bezieht sich auf Bedürfnisse von UserInnen und Qualität wird mit der diesbezüglichen Zufriedenheit der UserInnen gleichgesetzt. Der dritte Qualitätsbegriff misst die Fehlerzahl von Softwareprodukten im Verhältnis zu den programmierten »lines of code«.

In Fachliteratur und einschlägigen Normen und Standards existiert bisher kein einheitlicher und konsistenter Qualitätsbegriff und es herrscht eine »erstaunliche Inkonsistenz auf der terminologischen Ebene« (Petrasch 2001b, 104).

Im Entwicklungsprozess wird Softwarequalität gemeinhin auf zwei Ebenen verortet: erstens als Managementprozess im Rahmen des Softwarequalitätsmanagements und zweitens als technikkulturelle bzw. ethische Frage im Rahmen einer professionellen Ethik, die sich an EntwicklerInnen richtet. Die Aufgabe des Qualitätsmanagements ist es demnach, quantitative sowie qualitative Merkmale, die im Rahmen einer Anforderungsspezifikation festgelegt wurden, entsprechen den Erwartungen der beteiligten *Stakeholder* zu erfüllen (Abran u. a. 2004, 8/5). Eine formale Definition von Softwarequalität soll zu einem Übereinkommen über Qualitätsansprüche sowie zu einer klaren Kommunikation mit SoftwareentwicklerInnen darüber, was unter Qualität zu verstehen ist, beitragen (ebd., 11/1). Im Rahmen der professionellen Ethik wird darüber hinaus von EntwicklerInnen erwartet »to share a commitment to software quality as part of their culture« (ebd.). Professionelle Ethik umfasst dabei allgemeine Grundsätze, die EntwicklerInnen auffordern, ethisch zu handeln, und das Qualitätsmanagement dazu anhalten, entsprechende Rahmenbedingungen zur Verfügung zu stellen: »The ultimate effect of the work should be to the public good.«¹

Grundsätzlich soll Qualität durch Managementprozesse, wie etwa Qualitätssicherungsprozesse und verschiedene Prüfverfahren wie »Verifikation« und »Validierung« erreicht und durch entsprechende Methoden unterstützt werden. Im Rahmen der Verifikation sei dabei sicherzustellen, dass das Produkt *korrekt* entwickelt wird, d. h. dass der Output einer Software auch der Spezifikation entspricht. Validierung ist der Versuch sicherzustellen, dass das *richtige* Produkt entwickelt wird, d. h. dass das Produkt die spezifizierten Verwendungszwecke erfüllt (Abran u. a. 2004, 11/4f.).

Im Großen und Ganzen implizieren technikzentrierte Softwarequalitätskonzepte, dass Qualitätskriterien genau definiert und mit allen *Stakeholders* abgeklärt werden können. Im weiteren Verlauf seien sie auch entsprechend dieser Vereinbarungen spezifizier- und umsetzbar und können schlussendlich gemessen und objektiv überprüft werden. Insgesamt gehen die beschriebenen Zugänge relativ einheitlich davon aus, dass so etwas wie ein richtiger, objektiver Qualitätsbegriff und formale Lösungsansätze dieses »beste« Ergebnis zu erreichen, tatsächlich existieren. Jedoch ist der

»modelling process [...] complicated further by the fact that for any need there are many distinct (and valid) responses to that need; for each such response there are many distinct (and valid) formal models for the software that will realize that response; and for each formal model there are many distinct (and correct) realizations« (Blum 1994, 83).

Andrade u. a. (2004, 282) schlagen daher die Integration und Konzeptualisierung von unterschiedlichen Standpunkten vor: »Conceptual models are abstractions of the universe of discourse in which the problem occurs, as well as a possible model of a possible conceptual solution to that problem. They represent the problem from the viewpoint of the ›problem owner.« Auch Schinzel (2004) kritisiert die »Konstituierung vermeintlich objektiver Ordnungen« formalisierter Realitätsausschnitte. Formale Informatik-Modelle werden nur bedeutsam durch »soziale Prozesse der Entwicklung und Aneignung, wobei die subjektiven Perspektiven der Beteiligten entscheidenden Einfluss haben« (Floyd/Klischewski 1998).

In diesen Sichtweisen deutet sich im Gegensatz zum traditionellen Qualitätsverständnis der Informatik an, dass in der Softwareentwicklung auf diskursiv hergestellte Problemwahrnehmungen zurückgegriffen wird, für die sich darüber hinaus aus verschiedenen Perspektiven auch unterschiedliche Lösungswege darstellen. Hanappi-Egger (2003) weist in diesem Zusammenhang auf die zentrale Rolle hegemonialer Wahrnehmungsmuster in der Informatik-Ausbildung hin. SoftwareentwicklerInnen erwerben im Rahmen ihrer Ausbildung und beruflichen Praxis ein spezifisches Know-how in Form von informationstechnologischen Theorien, Tools und Methoden. Zusätzlich unterstützen soziale Aneignungsprozesse in diesem Setting die Einschreibung disziplinärer wissens- und kontextbasierter Herangehensweisen. Das bedeutet einerseits, dass sich EntwicklerInnen einen spezifischen Zugang zu Problemlösungen aneignen, die in technologischen Artefakten repräsentiert werden. Andererseits greifen sie in der Gestaltung technologischer Artefakte darüber hinaus auf vielfältige, in unterschiedlichen sozialen Interaktionen angeeignete gesellschaftliche Diskurse – darunter auch Geschlechter-Diskurse – zurück. Sozial konstruierte Vorstellungen, beispielsweise über zukünftige NutzerInnen, beeinflussen das Technikdesign und die ihm zugrunde liegenden Qualitätsstandards. EntwicklerInnen treffen permanent unbewusste Vorannahmen über UserInnen und deren Präferenzen: »Innovators are from the very start constantly interested in their future users. They construct many different representations of these users, and objectify these representations in technological choices«, betont Akrich (1995, 168). In die Entwicklung von technologischen Artefakten fließen daher *Gender Scripts*, d. h. vergeschlechtlichte Repräsentationen von AnwenderInnen ein, die diesen beispielsweise bestimmte Rechte, Kompetenzen oder Fähigkeiten zuschreiben (Hanappi-Egger 2004). Nach Akrich (1995) versuchen EntwicklerInnen sich den Präferenzen von UserInnen außerdem dadurch anzunähern, dass sie ihre eigenen Vorlieben oder Kenntnisse als repräsentativ für jene der zukünftigen UserInnen verstehen. Diese als »I-Methodolgy« bezeichnete Vorgehensweise führt allerdings zu Ausschlussmechanismen in der Nutzung technologischer Artefakte.

Qualitätsverständnis in der Praxis: explizite und implizite Qualitätsmerkmale

Auf welche Qualitätskonzepte greifen EntwicklerInnen nun in der Praxis tatsächlich zurück? Um dieser Frage nachzugehen, wurden im Rahmen des Forschungsprojekts zwei Softwareentwicklungen über einen längeren Zeitraum hinweg begleitet. Konkret ging es zum einen um die Entwicklung eines Computerspiels, zum anderen wurde ein Team im Bereich der Suchmaschinen-Technologien untersucht.

Wie sich in der Untersuchung zeigte, wurde in beiden Entwicklungsprozessen explizit auf klassische Qualitätsdefinitionen zurückgegriffen, die Softwarequalität als

bestimmte festgelegte und überprüfbare Merkmale wie Funktionalität oder Performance, als fehlerfreies Verhalten der Software und als Usability konzipieren. Dafür welche Qualitätskriterien von ProjektmanagerInnen, GrafikerInnen, ProgrammiererInnen usw. in den Vordergrund gestellt wurden, erwies sich der spezifische professionelle Zugang dieser unterschiedlichen Teammitglieder als ausschlaggebend. Zwar wurden weitgehend alle klassischen Qualitätskriterien aus Perspektive der unterschiedlichen Funktionen der Teammitglieder im Projekt genannt. Allerdings fehlte beiden Teams ein insgesamt konsistentes geteiltes Qualitätsverständnis. Generell wurde Qualität im Rahmen der untersuchten Produktentwicklungen am *State of the Art* gemessen oder daran, was die Unternehmensleitung als qualitativ »gut« definiert.

Bei genauerer Analyse der Qualitätskonzepte, auf die im Prozess tatsächlich zurückgegriffen wurde, konnten relativ widersprüchliche Qualitätsvorstellungen identifiziert werden. Neben dem explizit geäußerten Verständnis, das klassische Standards in Form einer Checkliste von technikzentrierten, objektivierten Merkmalen aufzählt, wurden immer auch als nicht operationalisierbar verstandene und/oder subjektiviert dargestellte Qualitätsmerkmale angeführt.

Im Rahmen der Fallstudie »Spieleentwicklung« zeigte sich eine starke Aufspaltung des Qualitätsbegriffs in einerseits objektivierte, professionelle Kriterien und andererseits subjektivierte, vage Präferenzen, die als nicht greifbar und damit als nicht objektivierbar oder operationalisierbar verstanden wurden. Im expliziten Anspruch der SpieleentwicklerInnen wurde Qualität in Form von Kriterien, die die Funktionalität, das Handling, die Ästhetik des Spiels sowie die Interaktion mit dem Spiel betreffen, artikuliert. Der implizite Qualitätsanspruch, der nach Meinung der EntwicklerInnen ein Computerspiel *eigentlich* als qualitativ gut ausweist, wurde dagegen in Form von subjektiviert dargestellten Präferenzen als atmosphärische oder emotionale Ebene des Spiels beschrieben: Während die erste Kategorie im Rahmen einer technisch-kreativen Lösung als Ausdruck technischer Qualifikationen erreicht werden könne und an technischen Standards messbar sei, könne man sich der zweiten Ebene nur durch künstlerisches »Talent« nähern und diese sei eher Ausdruck subjektiver Präferenzen denn operationalisierbarer Merkmale.

Im Bereich der Entwicklung von Suchmaschinen stellt sich das Verhältnis zwischen expliziten und impliziten Qualitätsstandards dagegen als Diskrepanz zwischen einem wissenschaftlich anspruchsvollen, allerdings auf einer abstrakten Ebene angesiedelten Qualitätsbewusstsein der EntwicklerInnen, und impliziten, eher pragmatischen und weniger reflektierten Umsetzungspraktiken in der konkreten Entwicklung von Anwendungen, dar. Nach Aussage der EntwicklerInnen spiegelt die Systematik von Suchmaschinen keine neutrale Ordnung von Suchergebnissen wider. Subjektivität ist in der Übersetzung komplexer sozialer Zusammenhänge in Informationssysteme aufgrund mangelnder Objektivierbarkeit unumgänglich. Dennoch wurden Überlegungen zur Gestaltungsmacht der EntwicklerInnen durch diese (notwendigerweise) subjektiven Zugänge nicht explizit mit Qualitätskonzepten in Verbindung gebracht. Im untersuchten Entwicklungsprozess wurde – wie später noch gezeigt werden wird – schließlich eher unbewusst auf Praktiken wie die Reproduktion althergebrachter Lösungsmuster zurückgegriffen. Subjektivität und mangelnde Objektivierbarkeit scheinen der Operationalisierung im Entwicklungsprozess auch in diesem Fall entgegen zu stehen.

Im Vergleich stellten die TeilnehmerInnen also in der Spieleentwicklung implizite, subjektivierte Kriterien als zentral dar, wobei keine Denkansätze darüber vorlagen, wie diese abseits vom Talent der EntwicklerInnen umgesetzt werden könnten. In der Suchmaschinenentwicklung näherte man sich Qualitätsfragen – neben der Aufzählung der üblichen Standards – explizit auch im Rahmen einer wissenschaftlichen Vorgehensweise auf einem höheren Abstraktionslevel an. Wenn es allerdings um die Umsetzung in einer konkreten Anwendung ging, konnte der hohe Abstraktionsgrad oftmals nicht auf die Implementierungsebene heruntergebrochen werden. Zwar manifestierte sich der Zusammenhang zwischen explizitem und implizitem Qualitätsverständnis in den beiden Fallstudien auf sehr unterschiedlichen Ebenen. Doch spielte die implizite Qualitätsebene insofern in beiden Fällen eine wichtige Rolle, als es im Entwicklungsprozess an Methoden fehlt, um einerseits subjektivierte Qualitätsmerkmale zu operationalisieren und es andererseits zu ermöglichen, theoretische Konzepte auf die konkrete Projektebene zu übersetzen. In diesem Zusammenhang kristallisierten sich aus geschlechtertheoretischer Perspektive wichtige Ansatzpunkte dafür heraus, wie implizite Qualitätsaspekte durch stärker reflektierte Vorgehensweisen in den Entwicklungsprozess einbezogen werden können. Konkret wird im Folgenden beispielhaft beschrieben, wie sich gesellschaftliche (Geschlechter-)Diskurse in die in der Software abgebildeten »Realitätsausschnitte« eingeschrieben haben.

Vergeschlechtlichte »Realitätsausschnitte«

Obwohl Usability in beiden Fallstudien immer wieder als zentrales Qualitätskriterium genannt wurde, erwies sich die Spezifikation von zukünftigen Nutzungskontexten für die entwickelte Software interessanterweise in beiden Projekten als Leerstelle, d. h. zukünftige UserInnen wurden von den EntwicklerInnen kaum explizit mitgedacht. Auf der Suche nach generischen oder allgemeinen Lösungen blieben Zielgruppen grundsätzlich weitgehend undifferenziert. Dennoch spielten Vorstellungen der EntwicklerInnen über »durchschnittliche« *User* (die in der Regel als männlich imaginiert werden) und eine entsprechende Marginalisierung von anderen UserInnengruppen eine wichtige Rolle. Im Rahmen der Spieleentwicklung wurden stereotype Repräsentationen von UserInnen beispielsweise über stereotype, vergeschlechtlichte Annahmen in das Computerspiel eingeschrieben.

Soziale und vergeschlechtlichte Konstruktionen werden allerdings nicht nur auf dem Wege von Imaginationen über UserInnen in Artefakte inkorporiert, sondern stehen auch in engem Zusammenhang mit dem in einer Software abgebildeten »Realitätsausschnitt«. Gedacht sei hier beispielsweise an Annahmen über geschlechtsspezifische Arbeitsteilung bei der Modellierung von Workflow Management Software, an populärkulturelle, soziale Konstruktionen in Computerspielen oder an die Hierarchisierung unterschiedlicher Wissensgebiete bzw. Informationsangebote in Informationssystemen.

So zeigten sich im Rahmen der Fallstudie »Spieleentwicklung« neben der Funktion »genderter« *user patterns*, über die NutzerInnenbilder der EntwicklerInnen in Spiele einfließen, wie bei der Entwicklung von Spielecharakteren geschlechtsspezifische Darstellungskonventionen angelegt wurden. Dabei orientieren sich die EntwicklerInnen in der Modellierung und Animation von Figuren an nach Geschlecht differenzierten »Realitätskonventionen«, die UserInnen als »männlich« und als

»weiblich« konstruierte Figuren als »fotorealistisch« glaubhaft machen sollen. Die männliche Hauptfigur des Spiels sollte ein »normaler Durchschnittstyp« sein, um eine Identifikation des als männlich imaginierten Spielers zu ermöglichen. Als Referenzmaterial wurden daher zur Entwicklung dieses *Charakters* Fotos eines Schauspielers herangezogen und die EntwicklerInnen zielten darauf ab, den männlichen Charakter anatomisch korrekt zu modellieren. Im Gegensatz dazu wurde als Referenzmaterial für die zentrale weibliche Figur bereits ein *3D-Character* gewählt. Das Ziel war, den weiblichen Charakter »hübsch und sexy« zu gestalten, weshalb er bis zu einem gewissen Grad nicht anatomisch korrekt sein durfte. Interne Diskussionen über persönliche Präferenzen der Teammitglieder in Bezug auf das Haar, die Beine, die Farbe und Form der Augen, die Größe der Nase und die Kleidung des weiblichen Charakters, wurden als sehr intensiv beschrieben.

Ebenso wurden im Spielverlauf »gegenderte« Narrative wirksam, über die in der Wahrnehmung der EntwicklerInnen offensichtlich soweit gesellschaftliche Einigkeit herrscht, dass ein Teil der Spielelogik nur nachvollzogen werden kann, wenn die Narration im Spiel entlang traditioneller Geschlechterstereotype entschlüsselt wird. Beispielsweise soll in einer Spielsequenz herausgefunden werden, wer der/die VerräterIn ist. Entsprechend der stereotypisierenden Kausallogik des Spiels, muss der/die UserIn den weiblichen Charakter als Verräterin ausschließen. Für eine Frau sei nämlich die Tasche mit der Belohnung zu schwer, deshalb müsse es die männliche Nebenfigur gewesen sein.

Im Bereich der Entwicklung von Suchmaschinen-Technologien verdeutlicht sich die Problematik der sozialen Konstruiertheit von Informationssystemen im Zusammenhang mit der Abbildung von Wissensgebieten oder der Selektion von Suchergebnissen. In diesem Prozess werden »Wissen« und Information durch die Entwicklung von Datenmodellen geordnet, zueinander in Beziehung gesetzt und für UserInnen aufbereitet. Auch in die Struktur solcher Klassifizierungen fließen gesellschaftliche (Geschlechter-)Diskurse ein, die sich in Informationssystemen manifestieren und damit soziale Zuschreibungen reproduzieren. Suchmaschinen greifen sowohl in der Aufbereitung von Abfragemöglichkeiten als auch in der Darstellung von Suchergebnissen auf Klassifikationen, d. h. Ordnungssysteme, die Sach- oder Wissensgebiete abbilden und in Form von Überordnungen, Unterordnungen oder Vernetzungen organisieren, zurück. Durch die Struktur der Abfragemöglichkeiten sowie die Suchprozesse selbst werden als relevant definierte Informationen von als nicht-relevant definierten Informationen differenziert. In der Aufbereitung von Suchergebnissen wird diese Auswahl gruppiert und zueinander in Beziehung gesetzt. Diese »Infrastrukturen« stellen historisch gewachsene Ordnungssysteme dar und sind Ergebnis sozialer Aushandlungsprozesse (Bowker/Star 1999). Wie bereits weiter oben angesprochen spiegeln diese Ordnungssysteme die gesellschaftlich geprägten Sichtweisen derjenigen wider, die sie entwickeln. Am Beispiel eines Informationssystems über Kursangebote zeigte sich im Rahmen der Fallstudie, wie traditionelle Geschlechter-Diskurse Eingang in ein solches System finden: Das Informationssystem sollte sich nämlich speziell auch an Frauen als UserInnen richten, woraufhin Kontextinformationen über Kinderbetreuungsangebote in das System integriert wurden. Rekonstruiert man den Weg, den die Anforderung, dass Frauen als UserInnen mitbedacht werden sollen, nimmt, zeigt sich also wie sich die diskursive Verknüpfung von Frau und Kinderbetreuung in der Ordnungslogik der Suchmaschinen-Anwendung mani-

festiert. Das Unterfeld Kinderbetreuung in der Kategorie »Ort« oder in der Kategorie »Rahmenbedingungen« findet erst durch das Mitbedenken von Frauen als UserInnen und durch die Reproduktion der Kinderbetreuung als weiblicher Tätigkeitsbereich Eingang in das Kategoriensystem. Tatsächlich wird im Rahmen von Kursangeboten relativ selten eine Kinderbetreuung zur Verfügung gestellt, d. h. Kinderbetreuungs-pflichten werden gesellschaftlich im Zusammenhang mit Kursbesuchen als nicht relevant konstruiert. Diese soziale Konstruktion bildet sich im hierarchischen Aufbau der Kategorisierung und der Unterordnung der Kategorie Kinderbetreuung unter Kategorien wie »Ort« oder »Rahmenbedingungen« in der Suchmaschine erneut ab und wird damit reproduziert.

Softwarequalität als Prozess

An diesen Beispielen zeigt sich deutlich, dass Softwareentwicklung einen sozialen Aushandlungsprozess darstellt, in dem Designentscheidungen vor dem Hintergrund gesellschaftlicher Diskurse verhandelt werden. Qualitätskonzepte spielen dabei im Sinne von in technischen Ausbildungen und der professionellen Praxis erworbenem Know-how, das sich im Qualitätsverständnis von EntwicklerInnen manifestiert, sowie auch im Sinne eines impliziten Wissens über qualitätsrelevante Kriterien, die über technikzentrierte Standards hinausgehen, eine wichtige Rolle. Im Kontrast dazu verstehen traditionelle technikzentrierte Qualitätskonzepte Technikentwicklung als objektives Verfahren. Sie konzipieren Qualitätssicherung im Rahmen von Software-Engineering Konzepten als Managementprozess bzw. gemäß professioneller Ethikkodizes als Aufgabe des/der einzelnen EntwicklerIn. Beide Ansätze gehen nicht auf die Funktion sozialer Aushandlungsprozesse in Produktentwicklungen ein, die sowohl eine Auseinandersetzung mit strukturellen Aspekten wie organisationalen Hierarchien als auch mit diskursiven Aspekten wie technikzentrierten und sozialkonstruierten hegemonialen Problemlösungsmustern voraussetzt.

Zwar zeichneten sich die untersuchten Entwicklungskontexte durch flache Teamhierarchien aus, dennoch zeigte sich in den Fallstudien, dass die Definition dessen, was Qualität im jeweiligen Entwicklungsprojekt bedeutet, durch Machtstrukturen – im Sinne von organisationalen Hierarchien sowie auch im Sinne gesellschaftlicher Hegemonien – geprägt ist.

Während das explizite Qualitätsverständnis über objektivierte, professionalisierte Zugänge operationalisiert wird, wobei gleichzeitig jedoch soziale Aspekte ausgeklammert bleiben, wird das implizite Qualitätsverständnis als »subjektive Geschmackssache« zu einem nicht operationalisierbaren Bereich konstruiert.

Soziale und vergeschlechtlichte Dimensionen von Softwarequalität werden auf diesem Wege weitgehend ausgeblendet und können somit nicht verhandelt oder diskutiert werden. Stattdessen wird unbewusst auf Verfahren zurückgegriffen, die dann als Einfallstor für *Gender Scripts* und soziale Konstruktionen dienen. Durch das Anwenden von I-Methodology und das Zurückgreifen auf imaginäre NutzerInnenbilder werden die sozialen Vorstellungen und Stereotype, die sich die EntwicklerInnen als Mitglieder der Gesellschaft und während ihrer professionellen Sozialisation angeeignet haben, aktiviert und unbewusst und unreflektiert in die Software eingeschrieben. Darüber hinaus fließen soziale und »gegenderte« Konstruktionen auch in die Modellierung von »Realitätsausschnitten« ein. Durch ihre Materialisierung in technolo-

gischen Artefakten werden sie diskursiv wirksam und reproduzieren unhinterfragte hierarchische Gesellschaftsstrukturen (vgl. dazu auch Allhutter 2007).

Um *Gender Scripts* und unreflektierte soziale Konstruktionen sichtbar und somit auch hinterfragbar zu machen, ist es notwendig, das bisherige explizite Qualitätsverständnis um implizite Qualitätsvorstellungen zu erweitern. Eine Erweiterung des Qualitätsverständnisses bedeutet, implizite Qualitätskriterien nicht als subjektive Präferenzen abzuhandeln, sondern versteckt vorhandene Vorstellungen explizit in den Entwicklungsprozess einzubringen und zur Diskussion zu stellen, damit ein bewusster Aushandlungsprozess stattfinden kann.²

Ziel ist es also, ein erweitertes Verständnis von Softwarequalität zu erreichen und so soziale Aspekte als zentralen Teil von Qualität festzuschreiben und zu professionalisieren. Traditionelle Definitionen sind vorrangig auf die technische Machbarkeit ausgerichtet. Diese finden sich je nach professionellem Zugang in unterschiedlichen Ausprägungen in den expliziten Qualitätsverständnissen der Entwicklungsteams. Zum Beispiel waren die ProgrammiererInnen vorrangig daran interessiert, die Fehlerquote im Code zu senken und unerwartetes Softwareverhalten auszuschließen, während sich *Interface-DesignerInnen* eher an *Usability-Richtlinien* orientierten.

Im Gegensatz zu diesen technologiezentrierten Qualitätsverständnissen erweist sich eine prozessorientierte Definition von Softwarequalität, welche die wechselseitige Beeinflussung von technologischen und sozialen Aspekten grundsätzlich einbezieht, als zielführend. In diesem Sinne müssen Aspekte, die bisher lediglich als implizites Qualitätsverständnis vorhanden waren, als zentrale Bestandteile von Softwarequalität anerkannt und professionalisiert werden. Im Sinne einer dekonstruktivistischen Strategie gilt es, soziale Komponenten im Entwicklungsprozess sichtbar zu machen und entsprechende Reflektionsmöglichkeiten in den Prozess zu integrieren (vgl. dazu genauer Allhutter/Hanappi-Egger 2005; 2006).

Der geschlechtertheoretische Zugang ermöglichte es im Rahmen der Untersuchung, die Relevanz diskursiver Einschreibungen in technologische Artefakte zu verdeutlichen und erweist sich im Entwicklungsprozess als geeignet, soziale Dimensionen von Softwarequalität zugänglich zu machen. Die Dekonstruktion subjektiver und kollektiver Realitätskonstruktionen kann dabei als Methode dienen, komplexe soziale Zusammenhänge herunterzubrechen und zu einem gemeinsamen Qualitätsverständnis zu gelangen.³

Literatur

- Abran, Alain/Moore, James W./Bourque, Pierre/Dupuis, Robert (2004) Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, Los Alamitos, Kalifornien.
- Allhutter, Doris/Hanappi-Egger, Edeltraud (2005) Making the Invisible Visible: Mind-Scripting as Method of Deconstructing (IT-) System Design. Proceedings und CD-ROM »The 13th International Conference of Women Engineers and Scientists«, Seoul/Korea.
- Allhutter, Doris/Hanappi-Egger, Edeltraud (2006) The Hidden Social Dimensions of Technologically Centred Quality Standards: Triple-Loop Learning as Process Centred Quality Approach; in: Dawson, R./Georgiadou, E./Linecar, P./Ross, M./Staples, G. (Hg.) Perspectives in Software Quality, The British Computer Society, 179-195.
- Allhutter, Doris/Hanappi-Egger, Edeltraud/John, Sara (2007) Gendered Software Design: Zur Sichtbarmachung von Gender Scripts in technologischen Artefakten. Forschungsbericht, Abteilung Gender and Diversity in Organizations, Wirtschaftsuniversität Wien.
- Allhutter, Doris (2007) Digitale Pornografie als technologisches Artefakt. Informationsethische

- Diskurse, soziotechnologische Praktiken und die EU-Politik zu *schädigenden* Internethalten, Dissertation, Universität Wien.
- Akrich, Madeleine (1995) User Representations: Practices, Methods and Sociology; in: Rip, A./ Misa, T.J./ Schot, J. (Hg.) *Managing Technology in Society. The Approach of Constructive Technology Assessment*, London/ New York, 167-184.
- Andrade, J./ Ares, J./ García, R./ Pazos, J./ Rodríguez, S./ Silva, A. (2004) A Methodological Framework for Viewpoint-Oriented Conceptual Modeling; in: *IEEE Transactions on Software Engineering*, 30/5, 282-294.
- Blum, B. (1994) A Taxonomy of Software Development Methods; in: *Association for Computing Machinery (Hg.) Communications of the ACM*, 37/11, 82-94.
- Bowker, Geoffrey C./ Star, Susan Leigh (1999) *Sorting Things Out. Classification and its Consequences*, Cambridge, MA/ London.
- Floyd, Chistiane/ Klischewski, Ralf (1998) Modellierung – ein Handgriff zur Wirklichkeit. Zur sozialen Konstruktion und Wirksamkeit von Informatik-Modellen; in: Pohl, K./ Schürr, A./ Vossen, G. (Hg.) *Modellierung '98 – Proceedings*, Universität Münster, 21-26.
- Hanappi-Egger, Edeltraud (2003) A Gendered View on New Technologies, *Proc. of NETIES03*, Cyprus, 10.-13. 10. 2003, Nikosia, Zypern.
- Hanappi-Egger, Edeltraud (2004) The Role of Gender in the Social Shaping of New Technology. 20th EGOS Colloquium, Ljubljana, Slovenien.
- Haug, Frigga (1999) Erinnerungssarbeit – ein Leitfaden zur Methode; in: *Vorlesungen zur Einführung in die Erinnerungsarbeit*, Berlin/ Hamburg, 199-227.
- Jørgensen, M. (1999) Software quality measurement; in: *Advances in Engineering Software*, 30, 907-912.
- Petrasch, Roland (2001a) Über den Software-Qualitätsbegriff – Teil 2: Anforderungen und Merkmale am Beispiel Software-Ergonomie (Usability), Download unter: <http://pi.informatik.uni-siegen.de/stt/24_2/03_Technische_Beitraege/petrasch.pdf> (Abfrage: 1. 2. 2007).
- Petrasch, Roland (2001b) Software-Qualität. Transformation von Anforderungen zu Merkmalen; in: Petrasch, R./ Riedemann, E. (Hg.) *Schriften zum Software-Qualitätsmanagement. Erfahrungs- und Forschungsberichte, Reihe: Software-Qualitätsmanagement: Theorie & Praxis Band 1*, Berlin.
- Schinzel, Britta (2004) Das unsichtbare Geschlecht der Neuen Medien, Download unter: <<http://mod.iig.uni-freiburg.de/publikationen/online-publikationen/unsichtbare-geschlechter-medien.pdf>> (Abfrage: 1. 2. 2007).

Anmerkungen

- 1 Vgl. *Software Engineering Code of Ethics and Professional Practice*, Online unter: <<http://www.acm.org/serving/se/code.htm>> (Abfrage: 1. 2. 2007).
- 2 Im Rahmen der Untersuchung wurde deutlich, dass in diesem Zusammenhang grundsätzlich auch die Strukturen, in denen Software entwickelt wird, sowie die Interessen der AuftraggeberInnen, die durchaus in Widerspruch zu Ansprüchen von tatsächlichen UserInnen stehen können, zu diskutieren sind.
- 3 Im beschriebenen Forschungsprojekt wurde Frigga Haugs (1999) Methode der »Erinnerungsarbeit« zu diesem Zweck weiterentwickelt und unter dem Begriff »Mind Scripting« zur Analyse technologischer Entwicklungsprozesse und der Einschreibung kollektiv geteilter Vorstellungen in Artefakte adaptiert (vgl. dazu genauer Allhutter/ Hanappi-Egger 2005; 2006).