



**Es hat sich
noch keine
wünschenswerte
Praxis im Um-
gang mit dieser
Technologie
entwickelt.**

Software ist ein ganz besonderer Saft

Wie **ethische Überlegungen** sinnvoll in den Entwicklungsprozess von Software integriert werden können, wird im Projekt „Ethik in der agilen Softwareentwicklung“ am bidt erforscht.

Von **Niina Zuber, Jan Gogoll** und **Severin Kacianka**

Die vielfältigen Einsatzmöglichkeiten der digitalen Technologien und Techniken verführen zu nicht verwunderlichen Glorifizierungen. Ihr Einsatz verheißt eine Reduktion sozialer Ungleichheiten, eine Wiederbelebung demokratischer Strukturen oder gar die Erschaffung eines harmonischen *global village*. Optimisten sehen in digitalen Technologien und Techniken eine Möglichkeit, mit technologischer Unterstützung oder gar durch eine mechanische Substituierung monotoner Tätigkeiten menschliche Freiheitsräume zu vergrößern, indem eine bessere Work-Life-Balance oder mehr sinnstiftende Tätigkeiten ermöglicht werden. Diese Sehnsucht nach einem gesellschaftlichen Zustand, durch den Bürgerinnen und Bürger wirtschaftlich abgesichert sind und sich selbst verwirklichen können, trägt die Ideen klassischer Gesellschaftsutopien eines Adam Smith oder John Stuart Mill weiter. Nicht nur das Telos des Maschineneinsatzes erscheint erstrebenswert, sondern auch der Arbeitsprozess der Systeme selbst verstärkt den Glauben an ein technologisch verursachtes Wunder.

Aufgrund der Tatsache, dass die Ausführungen von Softwaresystemen auf logisch-mathematischen Entscheidungskalkülen beruhen, gelten die Ergebnisse als objektiv und suggerieren, dass sie ein unverzerrtes Urteil abbilden. Problemlösungen, die das Resultat von Softwaresystemen sind, erscheinen sachlich und nüchtern und verdrängen die Notwendigkeit einer menschlichen und deshalb gerade subjektiven Interpretation der Situation. Endlich ist die menschliche Urteilskraft überwindbar, die ihr anhaftende Subjektivität und Willkür scheinen technisch bezwingbar zu sein. Das scheint auch das Versprechen evidenzbasierter Beurteilungssysteme zu sein, etwa des Softwaresystems, das bereits 2017 in den USA im Justizsystem eingesetzt wurde, um Prognosen über die Rückfälligkeit von Straftätern zu erstellen.

Utopie oder Untergang?

Auf der anderen Seite lauert jedoch die Gefahr eines gläsernen Bürgers, roboterartiger Menschen oder vermenschlichter

Roboter. Die Idee autonomer Systeme, die sich ohne menschliche Kontrolle weiterentwickeln und über ihr Fortbestehen eigenständig zu entscheiden vermögen, prophezeit den Untergang des Menschen.

Man muss jedoch gar nicht so weit gehen und sich eine Welt voller Maschinen mit Bewusstsein vorstellen, um Softwaresystemen mit einer rationalen und angemessenen Vorsicht zu begegnen: Unwohlsein resultiert bereits aus der Entwicklung und dem Einsatz von Computerprogrammen, die gesellschaftliche Strukturen stark verändern oder gesellschaftlich etablierte Entscheidungsstrukturen entkräften. Viele Vorfälle sind in den vergangenen Jahren öffentlich diskutiert worden: Vom Datenskandal um Cambridge Analytica oder den Biases, sprich den realen Diskriminierungen, nach Geschlecht und Hautfarbe im Zuge von Gesichtserkennungssystemen in der polizeilichen Fahndungsarbeit bis hin zur Exklusion von Personen mit eingeschränkter Sehkraft aufgrund einer bestimmten Farbwahl oder Schriftgröße auf Homepages.

Was macht gute Software aus?

Es ist daher kaum erstaunlich, dass in allen gesellschaftlichen Bereichen die Rufe nach Verantwortungsübernahme immer lauter und vehementer werden. Dies erzwingt jedoch zunächst die Frage nach der Besonderheit von Softwaresystemen. Sie erwecken den Anschein, etwas Besonderes zu sein. Nicht umsonst wird in den Wissenschaften seit Mitte des 20. Jahrhunderts diskutiert, ob es nicht sogar einer neuen speziellen Ethik bedürfe. Wissen wir eigentlich, was gute Software ausmacht? Sicher, zunächst muss Software technisch funktionieren. Sie muss auch ökonomisch rentabel sein. Aber reicht das aus, um sie mit den Prädikaten „wertvoll“, „vertrauenswürdig“ oder „wünschenswert“ auszuzeichnen?

Software muss noch mehr können, als technisch zuverlässig oder ökonomisch nützlich zu sein. Sie darf keine Benachteiligungen verschärfen oder perpetuieren. Softwaresysteme dürfen nicht grundlos exkludieren oder in die Autonomie der Menschen eingreifen. Autos oder Waschmaschinen sind bereits so voreingestellt, dass Gewohnheiten vom Nutzer gar nicht mehr eigenständig im Detail verändert werden können. Doch was macht Software eigentlich so besonders?

Abstinez im Physischen

Software entzieht sich zunächst der Haptik – sie definiert sich geradezu durch ihre scheinbare Abstinez im Physischen. Die Entwicklung kommt dabei ohne klassische Rohstoffe aus, quasi eine *Creatio ex nihilo*. Häufig benötigen Softwaresysteme nicht einmal mehr eine direkte und aktive Nutzer-Schnittstelle: Software funktioniert oft besser, wenn der Mensch sie weder bemerkt noch bewusst in sie eingreift. Dieses Phänomen wird häufig auch mit den Begriffen *Ambient Intelligence* oder *Ubiquitous Computing* gekennzeichnet. Software ist überall, aber sie entzieht sich der unmittelbaren Wahrnehmung. Selbst wenn sie wahrgenommen wird, bleibt sie häufig unzugänglich und unbegreifbar. Sie ist durchsichtig und zugleich opak. So kann Software in der Kommunikation, in der medizinischen Diagnose, in der Finanzwirtschaft oder

Software darf keine Benachteiligungen verschärfen oder perpetuieren.

in der Logistik gleichermaßen eingesetzt werden. Ihre Wandelbarkeit wird durch ihre scheinbare Materielosigkeit verstärkt. Software zeichnet sich also durch eine sehr flexible Formbarkeit aus. Die goldene Handwerksregel „Zweimal messen, einmal schneiden“ verliert stark an Bedeutung, da nicht funktionierende Software oft mit wenig Aufwand und Kosten korrigiert werden kann. Während z. B. die Autoindustrie möglichst viel testet, um teure Rückruf-Aktionen zu verhindern, kann Software oft mit wenig Aufwand und über das Internet kostensparend geupdated werden.

Überschuss an Zwecken

Software zeichnet sich auch durch einen Überschuss an Zwecken aus: Während sich etwa die Luft- und Raumfahrttechnik im weitesten Sinne mit Dingen beschäftigt, die fliegen, kann Software in jedem Bereich eingesetzt werden. Ein über einen langen Zeitraum klar definierter und konstanter Kontext der Anwendung fehlt. Dies verleiht Software den Charakter einer gerade erst entstehenden Technologie. Eine weitere Besonderheit von Software gegenüber anderen Formen der Ingenieurwissenschaften besteht zudem oft in dem Vertrauen auf aggregierten,

schwer zugänglichen Vorarbeiten anderer. So basiert aktuelle Software häufig auf einer Ansammlung vorher entwickelter Bibliotheken (*libraries*), die es Entwicklern ermöglicht, auf bereits vorhandene Algorithmen zurückzugreifen, um das Rad nicht dauernd neu erfinden zu müssen. Allerdings ist es für die Entwickler fast unmöglich, diese Bibliotheken en détail zu verstehen.

Aus diesen Gründen können wir noch gar nicht wirklich wissen, wie wir mit Software richtig und gut umgehen können sollen. Es hat sich noch keine wünschenswerte Praxis, ein gerechtfertigter Umgang mit eben jener Technologie etablieren können. Ebenso ist die Geschwindigkeit der Veränderung und Weiterentwicklung atemberaubend. Dies führt in eine normative Orientierungslosigkeit. Wir können nur zugeben, dass wir irgendwie immer zu spät sind. Aber wer ist verantwortlich dafür, dass Software so funktioniert, wie wir es für wünschenswert erachten? Sowohl Entwickler und Entwicklerinnen als auch Nutzer und Nutzerinnen werden angesprochen, sich – so mutet es an – anständig zu verhalten: „Do not cause harm!“ richtet sich an die Entwicklergemeinschaft oder „Denken Sie an den Menschen!“. Bekannt ist auch Googles ehemaliges Firmenmotto „Don't be evil“.

Dies ist die Grundlage für das Projekt „Ethik in der agilen Softwareentwicklung“, das am Bayerischen Forschungsinstitut für Digitale Transformation an Fragen ethisch wünschenswerter Softwareentwicklung arbeitet. Ziel ist es zu untersuchen, wie eine ethische Überlegung sinnvoll in den Entwicklungsprozess für Software integriert werden kann.

Dr. des. Niina Zuber und **Dr. Jan Gogoll** forschen am Bayerischen Forschungsinstitut für Digitale Transformation (bidt) der BADW.

Dipl.-Ing. Severin Kacianka ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Software und Systems Engineering der TU München.
