

Institut für Sozialanthropologie, Universität Bern
Eingereicht bei Prof. Dr. Heinzpeter Znoj
30. April 2022
Zeichen: 59'130

Software als Erweiterung des Selbst von Softwareentwickler:innen

Der Versuch einer Aktualisierung der Theorie Goffmans mithilfe der Akteur-Netzwerk-
Theorie von Latour und anhand des Anwendungsbeispiels der Softwareentwicklung

Bachelorarbeit von Kerrie Lesli Stauffer

Kerrie Lesli Stauffer
kerrie.stauffer@students.unibe.ch
Matrikelnummer 16-815-128
Major Sozialanthropologie
Minor Kriminologie und Informatik

INHALTSVERZEICHNIS

EINLEITUNG UND FRAGESTELLUNG	3
LATOURE AKTEUR NETZWERK THEORIE IN «DER BERLINER SCHLÜSSEL» UND DEREN ANWENDUNG AUF DIE SOFTWAREENTWICKLUNG	6
AKTIONSPROGRAMM UND USER STORIES	7
MITTLER:INNEN, SOZIALE BEZIEHUNGEN UND NETZWERKE.....	8
MEHR ALS EINE ÜBERSETZUNG: GEGENPROGRAMM UND DIE AUSLAGERUNG VON EIGENSCHAFTEN AUF MITTLER:INNEN.....	10
VORSCHRIFTEN UND USER PERSONAS	13
NETZWERKE UND KONSKRPTIONEN.....	15
GOFFMANS THEORIE DER SELBSTDARSTELLUNG	17
DARSTELLUNG UND TRANSFORMATION.....	17
STÖRFAKTOREN ALS GEGENPROGRAMM	18
ANALYSEEBENEN UND PERSPEKTIVEN	19
TRANSFORMATION UND AUSLAGERUNG DES SELBST	21
ANWENDUNG DER ERWEITERTEN THEORIE GOFFMANS AUF DIE SOFTWAREENTWICKLUNG	23
AUSLAGERUNG DES SELBST.....	24
FAZIT UND AUSBLICK.....	28
LITERATURVERZEICHNIS	30
SELBSTSTÄNDIGKEITSERKLÄRUNG.....	32

Einleitung und Fragestellung

In seiner *Theorie der Selbstdarstellung*¹ erklärt der Soziologe Erving Goffman, wie sich *Individuen* oder *Ensembles* während *Interaktionen* (in unmittelbarer physischer Präsenz voneinander) selbst darstellen (Goffman 2017). Interessant finde ich dabei insbesondere, dass Goffman aufzeigt, wie die Eigendarstellung unter anderem davon abhängig ist, welche Informationen Einzelpersonen im Voraus oder während der Interaktion zu ihrem Gegenüber haben beziehungsweise welche sie aus der Situation herauslesen können (ebd.: 5). Damit zeigt Goffman auf, dass bei limitierten Informationen frühere Erfahrungen oder auch Klischeevorstellungen zu Hilfe gezogen werden, um die *Selbstdarstellung* möglichst der Interaktion mit dem Gegenüber anzupassen (ebd.: 5). Ein Beispiel für Werte, die in solchen Situationen übertragen werden können, sind rassistische Vorurteile.

Die Literatur zum Thema rassistische Vorurteile in Algorithmen und Datensets hat sich in den letzten Jahren stark vervielfacht und das Problem wird vermehrt anerkannt (Silva und Kenney 2018: 9). Um das Einbinden von solchen Vorurteilen in Software im Allgemeinen zu vermeiden, ist es essentiell zu verstehen, wie solche Vorurteile und Glauben von Entwickler:innen bewusst oder unbewusst in die Programme einfließen (ebd.: 9). Dieses Verständnis ist gerade auch wichtig, da digitale Technologien (und damit auch Softwares) immer ausgeklügelter werden, sowie gleichzeitig eine immer grössere werdende Rolle im alltäglichen Leben (sozial, politisch oder auch ökonomisch) übernehmen (ebd.: 9).

Aufgrund des von Goffman in seiner Theorie aufgegriffenen Einflusses, den Klischeevorstellungen auf die Interaktion haben können, stellte sich mir die Frage, ob sich seine *Theorie der Selbstdarstellung* auch anwenden lässt, um dieses Einfließen von Vorurteilen in die Softwareentwicklung zu analysieren. Deshalb verfolge ich in meiner Arbeit das Interesse, Goffmans *Theorie der Selbstdarstellung*, die ursprünglich auf direkte Interaktionen zwischen menschlichen Subjekten angewendet wurde, auf die Interaktion mit nicht-menschlichen Wesen, sprich Software, auszuweiten. Somit setze ich mich indirekt auch mit der Frage der Relevanz von Goffmans Theorie in einer zunehmend digitalisierten Gesellschaft² auseinander.

¹ In dieser Arbeit werde ich Begrifflichkeiten, die den Theorien von Erving Goffman und Bruno Latour entspringen, in Kursivschrift verwenden, um sie hervorzuheben.

² Das Wort digital verwende ich hier vereinfacht als Überbegriff für all das, was im Endeffekt auf Binärcode zurückgeführt kann. Die Digitalisierung entspricht folglich der Weiterentwicklung von Binärcode (und allem was

In meinem Versuch Erving Goffmans *Theorie der Selbstdarstellung* auf nicht-menschliche Wesen auszuweiten, nehme ich mir die *Akteur-Netzwerk-Theorie* des Soziologen und Philosophen Bruno Latour in «Der Berliner Schlüssel» zu Hilfe. Die *Akteur-Netzwerk-Theorie* von Latour ist für mich insofern von Interesse, da sie nicht zwischen menschlichen und nicht-menschlichen Wesen unterscheidet (Latour 1996). Sie liefert ferner den Schlüssel, um bei einer erfolgreichen Verknüpfung mit Goffmans Theorie, auch bei Goffman nicht-menschliche Wesen einzubeziehen. Um aufzuzeigen welchen Nutzen eine solche Aktualisierung von Goffmans Theorie haben kann, werde ich sie im Anschluss an die Erweiterung als Analysetool in Bezug auf die Softwareentwicklung anwenden.

Die Fragestellung meiner Arbeit lautet folglich, ob und wie die *Theorie der Selbstdarstellung* Goffmans mithilfe von Latours *Akteur-Netzwerk-Theorie* so erweitert werden kann, dass sie auch in Situationen, wo Software die Interaktionen indirekt übernimmt, angewendet werden kann. In anderen Worten will ich versuchen, die Software als Erweiterung des *Selbst* von Entwickler:innen anzuschauen.

Um diese Frage zu beantworten, gehe ich das Thema wie bereits angedeutet in verschiedenen Stufen an. Als erstes möchte ich die *Akteur-Netzwerk-Theorie*, wie sie von Latour in «Der Berliner Schlüssel» erläutert wird, thematisieren. Ziel ist es aufzuzeigen, wie sich die Theorie auf den Prozess der Softwareentwicklung anwenden lässt. In einem nächsten Schritt geht es mir darum, Ähnlichkeiten zwischen der *Theorie der Selbstdarstellung* aus Goffmans «Wir alle spielen Theater» und der Theorie Latours aufzuzeigen. Schliesslich diskutiere ich, inwiefern Goffmans Theorie mithilfe von Latour in Bezug auf nicht-menschliche Individuen und nicht in direkter physischer Präsenz angewendet kann. In einem letzten Abschnitt wende ich schlussendlich die Theorie Goffmans zur Analyse des Prozesses der Softwareentwicklung an. Ich werde aufzeigen, dass die Software als Erweiterung des *Selbst* der Darstellenden angesehen werden kann.

Den Begriff Software möchte ich an dieser Stelle auf Webapplikationen (Webanwendungen) einschränken. Mit einer Webapplikation ist eine Sammlung aus dynamischen wie auch

darauf aufbaut), welche Informations- und Kommunikationsprozesse stark vereinfacht hat (Miller und Horst 2012: 5).

statischen Webseiten gemeint³. Dynamisch heisst, dass die Webseite vom Server geändert wird, bevor sie an den Browser gesendet wird⁴. Die statischen Websites im Gegensatz werden unverändert weitergeleitet⁵.

Wenn ich vom Prozess der Softwareentwicklung spreche, beziehe ich mich hauptsächlich auf das Programmieren von Applikationen, im Team oder allein. Dazu gehören aber auch Meetings mit Kund:innen sowie der Projektleitung, in welchen die Software konzipiert wird oder der Projektstand besprochen werden. Es handelt sich dabei grundsätzlich um sogenannte «custom made» Software. Also um Software, die explizit nach den Wünschen der Kundschaft programmiert wird.

Diese Eingrenzung der Begriffe der Software und des Softwareentwicklungsprozesses nehme ich vor, da ich auch Eigenerfahrung aus meiner Arbeit als Softwareentwicklerin in die Arbeit einfließen lassen werde. Diese Eigenerfahrung beschränkt sich auf die Entwicklung der eben erklärten Webapplikationen und beruht auf meiner mittlerweile einjährigen Erfahrung als Softwareentwicklerin. Teile meiner Arbeit werden daher zum Autoethnographischen tendieren, denn der Begriff der Autoethnographie beschreibt den Forschungsansatz, in welchem persönliche Erfahrungen und Erlebnisse systematisch hinterfragt und analysiert werden, um grössere Zusammenhänge besser zu verstehen (Ellis, Adams und Bochner 2011: 273).

³ 2021: Grundlegendes zu Webanwendungen. Adobe. <<https://helpx.adobe.com/de/dreamweaver/using/web-applications.html>>. 20. Februar 2022.

⁴ ebd.

⁵ ebd.

Latours Akteur Netzwerk Theorie in «Der Berliner Schlüssel» und deren Anwendung auf die Softwareentwicklung

Als ersten Schritt möchte ich die *Akteur-Netzwerk-Theorie* von Latour erläutern und demonstrieren, inwiefern sie sich auf den Prozess der Softwareentwicklung anwenden lässt. Einerseits um aufzuzeigen, dass die Theorie von Latour auf die Softwareentwicklung anwendbar ist. Andererseits um die Theorie vorzustellen, bevor ich in einem nächsten Schritt versuchen werde sie mit Goffmans *Theorie der Selbstdarstellung* zu verknüpfen.

Latour nimmt seiner Analyse vorweg, dass weder Subjekte noch Objekte existieren (Latour 1996: 38). Vielmehr stellen *Elemente*, die *Assoziationsketten* bilden, die Grundlage von Latours Analyse dar (ebd.: 39). Bei diesen *Assoziationsketten* handelt es sich vereinfacht gesagt um Ketten von Menschen und nicht-menschlichen Wesen (ebd.: 39). Latour nennt die *Assoziationsketten* auch *Vorrichtungen* (ebd.: 47).

Wenn *Elemente* solcher *Assoziationsketten* isoliert betrachtet werden, ist es schwierig sie als Menschen oder nicht-menschliche Wesen zu charakterisieren (ebd.: 40). Dies weil die Trennung in die beiden Kategorien von menschlichen und nicht-menschlichen Wesen laut Latour künstlich ist (ebd.: 40). Latour greift daher anstelle des Gegensatzes Mensch und Technik, auf den Gegensatz zwischen Verbindung und Ersetzung von *Elementen* der *Assoziationsketten* zurück (ebd.: 21). Sprich auf den Gegensatz der *Assoziation* (UND) und der *Substitution* (ODER) (ebd.: 21). Das heisst Latour betrachtet die *Elemente* der *Assoziationsketten* als verwandel- und ersetzbar (ebd.: 40). Genau diese Verwandel- und Ersetzbarkeit, auch als *Transformation* bezeichnet, ist von eigentlichem Interesse für Latour, denn isoliert betrachtet sind die Elemente für ihn nur schwer charakterisierbar (ebd.: 40). Das Aufheben des künstlichen Gegensatzes ist Latour auch deshalb wichtig, da die Aufhebung erst erlaubt diese *Transformation* richtig nachzuvollziehen (ebd.: 40). Was eine solche *Transformation* von *Elementen* einer *Assoziationskette* alles mit sich bringen kann, und wie *Transformationen* in Bezug auf die Softwareentwicklung aussehen können, möchte ich im Folgenden näher erläutern.

AKTIONSPROGRAMM UND USER STORIES

Das Skript eines *Elements* einer *Assoziationskette*, nennt Latour *Aktionsprogramm* (Latour 1996: 47). Jedes Wesen, oder eben auch *Element* einer *Assoziationskette*, kann mit einem solchen *Aktionsprogramm* versehen werden (ebd.: 21). Ein *Aktionsprogramm* drückt, in Worte übersetzt, den Sinn der verwandelten sozialen Interaktion aus (ebd.: 47). Beispiel für eine verwandelte soziale Interaktion ist das Hereingelassen werden von dem:der Concierge, welches durch den Berliner Schlüssel, der fortan den Zugang zum Gebäude reguliert, ersetzt wird (ebd.: 47). In Worte übersetzt, entspricht der Sinn des Schlüssels der Bitte, die Haustüre tagsüber offenzulassen, aber nachts zu schliessen, und stellt damit folglich das *Aktionsprogramm* des Schlüssels dar (ebd.: 47). Unterbrechungen eines solchen *Programms* werden als *Gegenprogramm* bezeichnet (ebd.: 21).

Gerne möchte ich hier bereits einen ersten Einschub vornehmen, um aufzuzeigen, inwiefern das Konzept und die Begriffe von Latour auf die Softwareentwicklung übertragbar sind. Das von Latour beschriebene *Aktionsprogramm* ist, meines Erachtens, vergleichbar mit sogenannten «user stories» in der Softwareentwicklung. «User stories» werden oftmals verwendet, um Funktionalitäten und Bedienungsmöglichkeiten eines Produktes, beziehungsweise der Software festzulegen (Sommerville, Ian 2020: 13). Sie sind textbasierte Beschreibungen von Handlungen, die Benutzer:innen einer Software vielleicht werden vornehmen wollen⁶ (ebd.: 13). «User stories» sind auf der einen Seite eine Hilfestellung um Funktionen der Software zu planen, auf der anderen Seite helfen sie aber auch sicherzustellen, ob die Anforderungen der Kundschaft richtig verstanden worden sind (ebd.: 13).

Eine Software ersetzt oftmals eine Tätigkeit, die vorher von Menschen ausgeführt wurde. Folglich kann gesagt werden, dass sie in der Regel soziale Interaktionen ersetzt. Als Beispiel möchte ich hier die Parallelen zwischen Identifizierung in Person (beispielsweise das Einchecken im Hotel) und dem Online-Login nennen, wie sie der Autor David Pogue in seinem Artikel erwähnt (Pogue 2011). Es kann somit abgeleitet werden, dass im Prozess der Softwareentwicklung, die soziale Interaktion in einer ersten *Transformation* der

⁶ Beispiel für eine *user story* ist folgendes Szenario: Heidi möchte sich in ihr Konto der Software xy einloggen. Leider hat sie aber ihr Passwort vergessen. Sie drückt daher auf den Link «Passwort vergessen». Nachdem sie ihre E-Mail-Adresse eingegeben hat, erhält sie ein Link zugesendet, der es ihr erlaubt ein neues Passwort festzulegen.

Assoziationskette in Worte übersetzt wird («user stories»), bevor sie dann in einem weiteren Übersetzungsschritt in eine technische Interaktion transformiert wird.

MITTLER:INNEN, SOZIALE BEZIEHUNGEN UND NETZWERKE

Wie anfangs bereits erwähnt, betrachtet Latour Transformationen anhand des Gegensatzes von Verbindung und Ersetzung (Latour 1996: 21). Um deren *Transformationen* besser zu umschreiben, teilt er die *Assoziationsketten* in zwei Dimensionen auf. Die *UND-Dimension* stellt dabei Verbindungen dar, während Ersetzungen in der *ODER-Dimension* dargestellt werden (Latour 1996: 40). Eine Verschiebung in einer Dimensionen wird auch Erweiterung genannt (ebd.: 58). Die Erweiterung in der *UND-Dimension*, sprich die Änderung des *Programms* oder Anwendung eines *Gegen-Gegenprogramms*, hat ihren Preis (ebd.: 58). Das *Aktionsprogramm* wird nämlich gleichzeitig durch Übersetzungen in der *ODER-Dimension* vergrößert (ebd.: 58). Die Verschiebung in den Verbindungen wird folglich mit einer Verschiebung in den Ersetzungen ermöglicht (ebd.: 58).

Aufgrund der Verschiebung der Verbindungen variiert also die Bindung von Aktant:innen an ein *Aktionsprogramm* von der einen *ODER-Dimension* zur nächsten (ebd.: 58f). Anders ausgedrückt, verändern sich die Verbindungen zwischen den *Elementen* mit jeder Änderung von *Aktions-* oder *Gegenprogramm*. Die Beziehung zwischen Anwohner:innen und Concierges beispielweise, verändert sich durch die Einführung des Berliner Schlüssels. Gleichzeitig verändert sich bei einer Veränderung am Schlüssel auch die Beziehung zwischen Schlüssel und Anwohner:in. Weiter ist es so, dass nicht nur der Bindungsgrad an eine Aussage sondern auch Kompetenz, Performanz und Definition sich neu aushandeln lassen (ebd.: 60). Da diese Wandlung der Akteur:innen sowohl auf Menschen wie auch nicht-menschliche Wesen anwendbar ist, sieht Latour die Aufteilung in diese beiden Kategorien, beim Betrachten von *Transformationen* innerhalb einer *Assoziationskette*, als hinderlich (ebd.: 60–61).

In seiner Analyse rund um den Berliner Schlüssel führt Latour aus, dass Hauseigentümer:innen neue Beziehungen eingehen mussten, um den Schlüssel überhaupt erst herstellen zu können (ebd.: 49). Mit den neuen Beziehungen sind neben zwischenmenschlichen Beziehungen auch Fachwissen wie zum Beispiel Mathematik gemeint

(ebd.: 49). Latour zeigt auf, wie der Schlüssel folglich zu mehr als nur einem Werkzeug wird, denn er autorisiert, verpflichtet und verbietet (ebd.: 49). Er ist eben ein *Mittler*, der soziale Beziehungen in anderer Form fortsetzt und ist dabei gleichzeitig Mittel und Zweck (ebd.: 49). Er stellt eine Fortsetzung sozialer Beziehungen mit anderen Mitteln dar und ist somit ein sozialer Akteur (ebd.: 49). Laut Latour kann Technik als Vermittlung der Beziehungen zwischen Menschen, Dingen und Tieren definiert werden (ebd.: 17). Wie bereits dargelegt, ersetzt Software (welche auch in den technischen Bereich gehört) oftmals zwischenmenschliche und somit soziale Interaktionen. Die Software vermittelt also die ehemals soziale Interaktion (ebd.: 48). Wir haben es dementsprechend mit einer *Mittlerin* zu tun, die die Beziehung zwischen Entwickler:in und Nutzer:in nicht nur übersetzt, sondern überhaupt erst konstituiert⁷ (ebd.: 48f). Weil es sich bei der Software um eine *Mittlerin* nach Latour handelt, werden folglich auch bei der Softwareentwicklung textbasierte Skripte verwendet, «user stories», die den Sinn einer *Vorrichtung*, hier Software, in einem Zwischenschritt in Worte übersetzen⁸.

Auch beim Analysebeispiel der Softwareentwicklung gehen Firmen oder Auftraggeber:innen neue Beziehungen ein, analog zu Latours Beispiel des Berliner Schlüssels. Dies geschieht anhand neuer Beziehungen mit den Personen, die sie anstellen, um Software herzustellen (Entwickler:innen und Projektmanager:innen), sowie mit der technischen Sprache, mit welcher sie im Verlauf der Gespräche mit den Entwickler:innen und der Projektleitung vertrauter werden. Gleichzeitig müssen sich Entwickler:innen, je nach Inhalt des Projekts, neue Fachkenntnisse aneignen. Durch das Eingehen dieser neuen Beziehungen wird also eine *Mittlerin* (Software) erstellt, die eine bisher soziale Interaktion (beispielsweise Identifizierung via Ausweis) in eine technische übersetzt (Online-Login via erstelltem Benutzerkonto). Es verändern sich folglich die Beziehungen der *Elemente* der *Assoziationskette*, in welche die Software eingebettet ist. Software ist somit eine soziale Akteurin, die unter anderem verpflichten (Passwort muss gemerkt werden), autorisieren (Zugang zu Funktionalitäten kann gewährt werden) und verbieten kann (Zugriff auf Applikation kann verwehrt oder eingeschränkt werden).

⁷ Um nicht die ganze Funktionalität von Latours Beispiel des Berliner Schlüssel erläutern zu müssen, habe ich seine Aussagen zum Schlüssel als *Mittler* hier direkt auf mein Eigenbeispiel der Software übertragen.

⁸ Es kann sein, dass diese «user stories» nirgends explizit niedergeschrieben werden, aber zumindest gedanklich wird immer eine Abfolge von Handlungen (als Worte) im Kopf festgelegt anhand derer die Funktionalität der Software implementiert wird.

Schlussendlich gibt es in Latours Theorie nicht nur *Ketten von Assoziationen* von verschiedenen *Elementen (Vorrichtungen)* (Latour 1996: 39), sondern auch Ketten von *Mittler:innen* verschiedener *Assoziationsketten*, welche wiederum ein *Netzwerk* bilden (ebd.: 51).

Um die Existenz von *Netzwerken* in der Softwareentwicklung aufzuzeigen, möchte ich das Beispiel von Frameworks herbeiziehen. Frameworks sind eine Art Grundgerüst, das bestimmte Elemente und Funktionen zur Verfügung stellt, so dass man nicht jedes Mal alles von Grund auf neu programmieren muss⁹. Beispiel für ein Framework, das ich persönlich oft verwende, ist Angular. Angular erlaubt es einem in kurzer Zeit eine funktionierende Webapplikation zu erstellen¹⁰. Wie bereits etabliert, sind Softwares *Mittlerinnen*. Da Frameworks ebenfalls Softwares sind, stellen auch sie *Mittler:innen* dar. Wenn nun also ein Framework in die eigene Software eingebaut wird, entsteht ein *Netzwerk*, da mit dem Framework und dem selbstgeschriebenen Code der hergestellten Software zwei *Mittler:innen* verbunden werden. Daraus ergibt sich, dass auch das Konzept der *Netzwerke* auf die Softwareentwicklung anwendbar ist.

MEHR ALS EINE ÜBERSETZUNG: GEGENPROGRAMM UND DIE AUSLAGERUNG VON EIGENSCHAFTEN AUF MITTLER:INNEN

Bei seiner Analyse von *Assoziationsketten* geht Latour anhand eines anderen Beispiels, dem gusseisernen Schlüsselanhänger in Hotels, noch einen Schritt weiter. Er zeigt nämlich auf, dass mit dem Schlüsselanhänger in Hotels nicht nur die Bitte, den Schlüssel an der Rezeption abzugeben in eine andere Art der Interaktion übersetzt wird (Gewicht am Schlüssel), sondern auch Moral und Disziplin ausgelagert werden (Latour 1996: 53). Sprich mit dem Anbringen des Gewichts am Schlüssel verlassen sich Hoteliers fortan nicht mehr auf die Moral und Disziplin der Gäste, sondern lagern diese in den Schlüsselanhänger aus (ebd.: 53). Ausgangslage der Situation betreffend des Schlüsselanhängers ist die schriftliche Aufforderung an die Gäste, den Schlüssel an der Hotelrezeption abzugeben (ebd.: 53). Wie

⁹ Hery-Mossman, Nicole 2021: Was ist ein Framework? - einfach erklärt. CHIP. <https://praxistipps.chip.de/was-ist-ein-framework-einfach-erklart_41348>. 30. März 2022.

¹⁰ Introduction to the Angular Docs. Angular. <<https://angular.io/docs>>. 28. März 2022.

nachdrücklich eine solche geschriebene Aussage wirkt, hängt einerseits von der Wortwahl ab, andererseits jedoch auch davon, was die angesprochenen Personen aus der Aussage machen (ebd.: 53). Wenn beispielsweise die Sprache schon gar nicht verstanden wird, hat die Bitte wenig bis gar keinen Nachdruck (ebd.: 54). Um die Nachdrücklichkeit der Aussage zu verstärken, gibt es im Szenario des Schlüsselanhängers zwei Handlungsmöglichkeiten. Es können entweder die Gäste ähnlich gemacht werden, so dass sie beispielweise alle die Sprache der Aussage lesen können, oder die Aussage anderweitig gewichtiger gemacht werden, um den Gästen das gewünschte Verhalten aufzuzwingen (ebd.: 54). Beispiele dafür sind das Schild, genauer der verwendete Imperativ oder auch das Gewicht des Schlüsselanhängers (ebd.: 54).

Die *Programme* der Hoteliers verkomplizieren sich auf die Antwort der Unwilligkeit (oder auch Vergesslichkeit) der Gäste, die Schlüssel abzugeben (ebd.:54). Wenn wir uns zurückerinnern, sind solche Unterbrechungen eines *Aktionsprogramms* als *Gegenprogramme* definiert (ebd.: 21). Um ihnen entgegenzuhalten, werden erneute Änderungen eingesetzt, sogenannte *Gegen-Gegenprogramme* (ebd.: 54). Denn je grösser die Anzahl *Gegenprogramme*, die sich nicht verwirklichen lassen, desto eher wird die Aussage befolgt, sprich das *Aktionsprogramm* erfolgreich ablaufen (ebd.: 54).

Durch das Übertragen des *Programms* in anderes Material wurde es übersetzt und transformiert (ebd.:55). Die Aussage ist eine andere, da sie nicht einfach übermittelt, sondern (in ein anderes Material) übersetzt worden ist (ebd.: 55). Aufgrund dieser Übersetzung des *Programms* ändert sich auch das Verhalten der Adressat:innen (ebd.:55). Im Fall des Hotelschlüssels ändert sich also das *Gegenprogramm* der Gäste (ebd.: 55). Anders ausgedrückt wurde das *Gegenprogramm* der Gäste durchkreuzt. Sie befolgen nun das *Programm* nicht mehr aus Pflicht, sondern aus Egoismus (ebd.: 55). Das *Aktionsprogramm* der Hoteliers wird befolgt, damit sie das Gewicht nicht herumtragen müssen und beachten die geschriebene Aufforderung zur Schlüsselabgabe gegebenenfalls gar nicht erst (ebd.: 54f). Es findet eine Auslagerung von Moral und Disziplin statt, die sie folglich dazu verleitet, das *Aktionsprogramm* nicht mehr zu unterbrechen (ebd.: 55).

Diese Auslagerungen von menschlichen Eigenschaften, sowie die Interaktion von *Aktions-* und *Gegenprogramm* ist auch bei der Softwareentwicklung erkennbar. Anhand des Beispiels des Logins/Erstellen eines Benutzerkontos wird ersichtlich, dass die Konzepte von *Aktions-*

und *Gegenprogramm* sich gut auf den Bereich der Software übertragen lassen.

Benutzerkontos dienen dazu, Benutzer:innen zu identifizieren und ihnen Zugang zu einem in der Regel passwortgeschützten Bereich zu erlauben. Damit sich nicht andere (fremde) Personen Zugang auf das Benutzerkonto verschaffen können, werden Personen darauf hingewiesen, ein möglichst sicheres Passwort auszuwählen. Ein möglichst sicheres Passwort besitzt eine hohe Entropie, das heisst, dass es schwieriger ist, das Passwort durch Erraten, Ausprobieren oder Anwenden von anderen Methoden zu knacken¹¹. Wichtig für eine hohe Sicherheit sind die Anzahl Zeichen, mit Gross- und Kleinbuchstaben, Zahlen und Sonderzeichen und das nicht Wiederverwenden von Passwörtern¹².

Nun ist es aber so, dass manch eine Person zu faul ist oder sich ein schweres Passwort nur schwer merken kann, weshalb sie einen Hinweis darauf, ein sicheres Passwort zu wählen, oftmals einfach ignoriert (*Gegenprogramm*). Dies hat dazu geführt, dass vielerorts beim Erstellen eines Benutzerkontos, ein Passwort gar nicht erst akzeptiert wird, wenn es Bedingungen wie die oben erwähnten gar nicht erfüllt (*Gegen-Gegenprogramm*).

Analog zu Latours Ausführungen zum Schlüsselanhänger, kann also auch in diesem Fall gesagt werden, dass die Moral und Disziplin ausgelagert wurden. In unserem Beispiel wurde sie jedoch, statt auf einen gusseisernen Anhänger, auf die Software übertragen. Die Software wurde (in unserem Beispiel) so konzipiert, dass Nutzer:innen sich gezwungen sehen, die Vorgaben für ein sicheres Passwort umzusetzen. User:innen können demnach nicht selber entscheiden, ob sie moralisch sein wollen und ein sicheres Passwort wählen oder nicht, denn die Software lässt ein unsicheres Passwort erst gar nicht zu. Inwiefern dies dann tatsächlich die Passwortsicherheit erhöht, ist eine andere Frage und es besteht natürlich auch immer die Gefahr, dass Nutzer:innen sich das Passwort einfach auf einem am Bildschirm angehefteten Post-it notieren¹³.

¹¹ 2020: Passwort-Entropie. ComputerWeekly.de.

<<https://www.computerweekly.com/de/definition/Passwort-Entropie>>. 30. März 2022.

¹² Schulz, Sven 2022: Was ist ein sicheres Passwort? Die besten Tipps. CHIP. <https://praxistipps.chip.de/was-ist-ein-sicheres-passwort-die-besten-tipps_3482>. 30. März 2022.

¹³ David Pogue illustriert dies auch in seinem kurzweiligen Artikel des «Scientific American» (Pogue 2011).

VORSCHRIFTEN UND USER PERSONAS

Wie bereits anhand des Schlüsselanhängers und anschliessend bei der Software aufgezeigt, werden durch *Transformationen* Eigenschaften zwischen den Aktant:innen umverteilt. Im Beispiel des Programmierens einer Benutzeroberfläche, die es erlaubt, ein Benutzerkonto mit Passwort zu erstellen, sind es Moral beziehungsweise Disziplin, die von den Nutzer:innen auf die Software übertragen werden.

Das Bedienen von nicht-menschlichen *Mittler:innen* ohne Intelligenz kann aber bedeuten, dass gewisse Anforderungen an Benutzer:innen entstehen (Latour 1996: 68). Das durch die Voraussetzungen der nicht-menschlichen Delegierten den Menschenwesen aufgezwungene Verhalten nennt Latour auch *Vorschrift* (ebd.: 68). Sie erklärt die moralische und ethische Dimension mechanischer *Vorrichtungen*¹⁴ (ebd.: 68). Latour fasst es auch prägnant zusammen unter: «Die nicht-menschlichen Wesen übernehmen die Einstellungen derer, die sie entwickelt haben» (ebd.: 69). In anderen Worten hängt die *Vorschrift* von *Mittler:innen* stark davon ab, wer sie entwickelt. Nicht-menschliche Wesen sind deshalb als *anthromorph* zu betrachten (ebd.: 72). Sprich sie besitzen eine Menschengestalt oder geben Menschen eine Gestalt (ebd.: 72). Dies, weil sie von Menschen fabriziert wurden, Handlungen von Menschen ersetzen, die Position eines Menschen einnehmen und der menschlichen Handlung Gestalt geben (ebd.: 72). *Mittler:innen* können folglich in Szenarien diskriminieren, die von den Personen, die sie konzipiert haben, nicht vorhergesehen oder angedacht wurden (ebd.: 70ff).

Wie bereits etabliert, kann eine Software als *Mittler:in* nach Latour betrachtet werden. Folglich kann auch bei der Softwareentwicklung ausgesagt werden, dass die *Vorschrift*, also die Voraussetzungen an Nutzer:innen beziehungsweise die ausgelagerten Eigenschaften, stark davon abhängen, wer die Software entwickelt hat. Oder wie Design-Anthropologin Ariel Sim es im Vortrag über User Experience und Design-Anthropologie aufzeigt, ist es so, dass Designer:innen (oder eben auch Entwickler:innen) beim Planen und Erstellen eines Systems gewisse Entscheidungen treffen, welche einen grossen Einfluss auf das System als Endprodukt haben¹⁵. Anders ausgedrückt, hängt die Funktionalität oder auch die Bedienung

¹⁴ *Vorrichtung* ist wie schon am Anfang erwähnt ein anderer Begriff für *Assoziationsketten* (Latour 1996: 47)

¹⁵ Sim, Ariel 2019: How Might We Future Proof User Experience.
<https://www.youtube.com/watch?v=eqTTD2tI_ml>. 30. März 2020.

eines Systems demnach stark von den während dem Designprozess getroffenen Entscheidungen ab¹⁶.

Die Kluft zwischen den im *Aktionsprogramm* vorgesehenen Benutzer:innen, jene die die *Vorschrift* erfüllen und den tatsächlichen, kann ganz erheblich oder aber auch komplett aufgehoben sein (Latour 1996: 74). Je nachdem, wie gut die Benutzer:innen antizipiert wurden und wie präzise ihr zugewiesener Platz im Szenario ist (ebd.: 74). Massnahmen zur Verringerung dieser Kluft zwischen antizipierten und tatsächlich vorliegenden Nutzer:innen nennt Latour *Vor-Einschreibungen* (ebd.: 75). Je mächtiger eine Software ist, sprich je mehr Funktionalität sie hat, desto höher muss oftmals das Kompetenzlevel sein, um sie bedienen zu können. Das heisst, die Anforderungen an die Nutzer:innen oder auch die *Vorschrift* nach Latour sind relativ hoch. In der Softwareentwicklung werden daher zum Teil sogenannte «user personas»¹⁷ erstellt. Diese sollen helfen, eine an den Bedürfnissen, wie auch Fähigkeiten der potenziellen Benutzer:innen angepasste Benutzeroberfläche zu erstellen (Sommerville, Ian 2020: 54). Dies ist auch gerade bei Szenarien wichtig, in welchen nicht ganz klar ist, wer die potenziellen User:innen überhaupt sein werden und welchen Hintergrund sie haben (ebd.: 54). Das Aufsetzen von solchen ausgedachten Personen kann insbesondere auch deshalb helfen, weil bei unklarem Zielpublikum, Entwickler:innen mit grosser Wahrscheinlichkeit unterschiedliche Vorstellungen davon haben, wer das Produkt bedienen wird und welche Fähigkeiten diese Personen besitzen (ebd.: 54). Mithilfe von «user personas» kann also sichergestellt werden, dass alle Personen im Entwicklungsteam dieselben potenziellen Nutzer:innen im Kopf haben (ebd.: 54). Folglich können dank «user personas» nicht nur potenzielle Schwierigkeiten beim Verständnis und der intuitiven Bedienung von Software frühzeitig erkannt werden (ebd.: 54f), sondern es kann auch klar aufgezeigt werden, dass die Vorstellungen, die Entwickler:innen über potenzielle Benutzergruppen haben, in das Produkt hineinfließen und damit auch die Interaktion zwischen Nutzer:innen und Software stark prägen. «User personas» sind also ein in der Softwareentwicklung verwendetes Hilfsmittel, um die *Vorschrift* so zu gestalten, dass sie von möglichst vielen Nutzer:innen erfüllt wird. Nach Latour entsprechen sie folglich einer *Vor-Einschreibung*.

¹⁶ ebd.

¹⁷ Analog zu den «user stories» werden auch «user personas» meiner Erfahrung nach nicht zwingend schriftlich festgehalten, sondern eher während Besprechungen in Worte gefasst oder implizit in Gedanken erstellt.

NETZWERKE UND KONSKRIPTIONEN

Jede *Vorrichtung*, sprich *Assoziationskette*, ist von verschiedenen Zonen umgeben, die *Umschreibungen* genannt werden (Latour 1996: 75). Die *Umschreibung* definiert nur, mit welchen Ressourcen eine *Vorrichtung* ausgestattet ist, um die Beziehung zu den Nutzer:innen vorzubereiten, jedoch garantiert sie nicht, dass die Benutzer:innen gehorchen werden (ebd.: 76). Das Vorschreiben von Verhaltensweisen durch automatisieren oder durch ein Skript kann vieles bewirken, jedoch ist die schlussendliche Wirkungskraft zu einem Grossteil auch auf das aufeinander Ausrichten und Abstimmen verschiedener *Vorrichtungen* zurückzuführen (ebd.: 77).

Ein Ensemble von *Vorrichtungen*, die aufeinander abgestimmt sind, erlaubt es Benutzer:innen mühelos zu navigieren, da sie genügend Kompetenzen *vor-einschreiben* (ebd.: 77). In anderen Worten erlauben die aufeinander abgestimmten *Vor-Einschreibungen* der *Vorrichtungen*, den Nutzer:innen eine mühelose Navigation, dank verringerter Kluft. *Konskription* ist der Begriff, den Latour verwendet, um solche aufeinander abgestimmte Ensembles zu umschreiben (ebd.: 77). *Konskriptionen* reduzieren die Anzahl von Momenten, in denen Worte benötigt werden. Sie sind entweder in Menschen ein- oder nicht-menschliche Wesen ausgegliedert und die Handlungen werden schweigsam vertraut ausgeführt (ebd.: 77ff). Inwiefern gewisse Dinge (Moral etc.) in nicht-menschliche Körper ausgegliedert oder in menschlichen Körper inkorporiert werden, unterliegt der Entscheidung der Entwickler:innen (ebd.: 82).

Als Beispiel für ein *Netzwerk* von *Mittler:innen*, habe ich bereits früher die Verwendung von Frameworks genannt. Wenn mehrere *Mittler:innen* (Frameworks von verschiedenen Teams von Entwickler:innen) zusammengeführt und in der eigenen Software eingebaut werden, so dass das Endprodukt mühelos benutzt werden kann, liegt eine *Konskription* nach Latour vor. In einem nächsten Schritt liesse sich dann auch die Hardware als Teil der *Konskription* betrachten.

Kurz zusammengefasst habe ich in diesem Kapitel aufgezeigt, dass die *Akteur-Netzwerk-Theorie* von Latour auf den Prozess der Softwareentwicklung anwendbar ist. Software kann als *Mittlerin* betrachtet werden, welche die Beziehung zwischen Nutzer:in und Entwickler:in

konstituiert. Anhand des Beispiels des Erstellens eines Benutzerkontos habe ich aufgezeigt, dass Eigenschaften von menschlichen Wesen in die Software ausgelagert werden.

Goffmans Theorie der Selbstdarstellung

Im folgenden Teil meiner Arbeit geht es mir darum, Parallelen zwischen Goffmans *Theorie der Selbstdarstellung* in «Wir alle spielen Theater» und der bereits eingeführten *Akteur-Netzwerk-Theorie* von Latour in «Der Berliner Schlüssel» aufzuzeigen. Im Anschluss davon möchte ich diskutieren, welche Erkenntnisse diese Verbindungen mit sich bringen, bevor ich die gezogenen Schlüsse schliesslich in einem letzten Unterkapitel auf die Softwareentwicklung anwenden werde.

DARSTELLUNG UND TRANSFORMATION

Goffmans *Theorie der Selbstdarstellung* entspricht laut seinen eigenen Angaben einem Modell, das charakteristisch sei für einen Grossteil der unter natürlichen Bedingungen stattfindenden sozialen Interaktionen (Goffman 2017: 218). *Interaktionen* definiert er als die gegenseitige Beeinflussung, von Individuen auf ihre Handlungen während ihrer unmittelbaren physischen Präsenz (ebd.: 18).

Bei seiner Untersuchung solcher *Interaktionen* legt Goffman den Fokus darauf, wie Individuen sich gegenseitig während einer *Interaktion* beeinflussen (ebd.: 18). Bei seinem Modell geht es darum, die bewegenden Momente einer *Interaktion* zu analysieren (ebd.: 218). Damit ist bei Goffman, ähnlich einer *Transformation* nach Latour, das Event (*Interaktion*), welches die beteiligten Parteien (Individuen bei Goffman und Aktant:innen bei Latour) verändert hervorgehen lässt, im Zentrum des Interesses.

Bei Goffmans Theorie wird die Deutung der Situation durch die Handlung einer einzelnen Person beeinflusst, wenn diese anderen gegenübertritt (ebd.: 9). Wenn nun eine *Darstellung* als die Gesamttätigkeit einer Person in einer bestimmten Situation definiert wird (ebd.: 18), sind es also die *Darstellungen* der an einer Situation beteiligten Personen, die die Deutung der Situation beeinflussen.

Da die *Darstellung* einer Tätigkeit sich immer in bestimmten Punkten von der Tätigkeit selbst unterscheiden wird, ist eine *Darstellung* zwingend falsch (ebd.: 61). Dies argumentiert Goffman auch anhand der Tatsache, dass Einzelpersonen Tätigkeiten nur mithilfe von Zeichen darstellen können (ebd.: 61). In anderen Worten muss die Tätigkeit also zuerst in Zeichen übersetzt werden, um dargestellt werden zu können. Daher liegt jeder *Darstellung* eine Übersetzung zugrunde.

Latour definiert in seiner Theorie genau solche Übersetzungen als *Transformationen* (Latour 1996: 55). Obwohl eine *Darstellung* nach Goffman einer *Transformation* nach Latour zu entsprechen scheint, sollte jedoch nicht vergessen werden, dass der jeweilige Kontext der beiden Theorien widersprüchlich ist. Bei Goffman ist die soziale Interaktion auf bestimmte Rahmenbedingungen (ohne Einflüsse von aussen und auf Individuen) eingeschränkt (Goffman 2017: 18). Latour hingegen kritisiert genau diese Einschränkung und sagt, dass der Begriff der Interaktion auch das Nicht-Menschliche umfassen sollte, damit die künstlichen Rahmenbedingungen, die bei Theorien zu Interaktionen oftmals vorausgesetzt wurden (so auch bei Goffman) weggelassen werden können (Latour 2001: 20ff). Im Folgenden werde ich *Transformationen* nach Latour und *Darstellungen* nach Goffman jedoch als gleichwertig anschauen, um aufzuzeigen, dass Goffman ebenjene (von Latour erwähnte) Erweiterung um nicht-menschliche Akteur:innen benötigt, um im digitalen Zeitalter nicht an Aktualität zu verlieren.

Wie schon etabliert, sind *Interaktionen* bei Goffman die gegenseitigen Beeinflussungen der jeweiligen Handlungen von Individuen während ihrer unmittelbaren physischen Präsenz (Goffman 2017: 18). *Interaktionen* sind folglich das Zusammenspiel der *Darstellungen* der an der *Interaktion* beteiligten Personen. Wenn nun also eine *Darstellung* von Goffman einer *Transformation* nach Latour entspricht, und zusammenhängende *Darstellungen* eine *Interaktion* konstituieren, entspricht diese *Interaktion* einer *Assoziationskette* von Latour.

STÖRFAKTOREN ALS GEGENPROGRAMM

Zu den *Darstellungen* erläutert Goffman weiter, dass es innerhalb einer *Interaktion* immer wieder zu Ereignissen kommen kann, die eine *Darstellung* als fragwürdig oder gar unglaubwürdig auftreten lassen (Goffman 2017: 15). Eine unglaubwürdige oder unterbrochene *Darstellung*, bedeutet in Latours Theorie eine Verhinderung der *Transformation* und dadurch eines *Aktionsprogramms*. Was Latour also als das *Gegenprogramm* bezeichnet, behandelt Goffman als *Störfaktor*. Als Konsequenz kann ausgesagt werden, dass alles, was Teil der *Darstellung* ist zum *Aktionsprogramm* und alle *Störfaktoren* einem *Gegenprogramm* angehören.

Ein klarer Unterschied zwischen Goffman und Latours Theorien zeigt sich hier, indem bei Goffman auch die Darstellenden selbst in der Lage sind, der Glaubwürdigkeit ihrer

Darstellung zu schaden (ebd.: 60). Inwiefern *Mittler:innen* selbst in der Lage sind, ein *Gegenprogramm* durchzuführen, ist jedoch fraglich. Ich argumentiere an dieser Stelle jedoch, dass dieser Unterschied nicht so erheblich ist, denn je schlechter ein:e *Mittler:in* ist, desto weniger ausgeklügelt muss ein *Gegenprogramm* sein, um die *Transformation* erfolgreich zu verhindern. Ein:e *Mittler:in* kann in Latours Analyse also ebenfalls, wenn auch nur indirekt, an einer erfolglosen *Transformation* schuld sein.

ANALYSEEBENEN UND PERSPEKTIVEN

Bei den von Goffman betrachteten *gesellschaftlichen Einrichtungen*, der Ort, an dem die beschriebenen Interaktionen stattfinden, handelt es sich um verhältnismässig geschlossene Systeme (Goffman 2017: 218). Die Beziehungen zwischen *Einrichtungen* untereinander werden als Teil einer anderen Sachlage behandelt, nämlich bei der Analyse der institutionellen Integration (ebd.: 218). Vereinfacht gesagt unterteilt Goffman in verschiedene Analyseeinheiten, wobei jene der Institutionen nicht derselben wie die der *Interaktionen* angehört.

Eine *Institution*, Analyseeinheit auf Niveau der Beziehungen zwischen *Einrichtungen*, kann nach Goffman unter verschiedenen Gesichtspunkten betrachtet werden (ebd.: 218f). Dabei sind alle Tatsachen für jede der Perspektiven relevant, jedoch in einer anderen Prioritätsskala und Ordnung (ebd.: 218f).

In der *technischen Perspektive* steht die Un-/Wirksamkeit einer *Institution* bei der Erzielung vorbestimmter Zwecke im Vordergrund (ebd.: 218f). Die Handlungen, die man von seinem Gegenüber verlangen kann, stellen den Fokus der *politischen Perspektive* dar (ebd.: 219). Von einem *strukturellen Gesichtspunkt* aus stehen jedoch die horizontalen und vertikalen Statusunterscheidungen im Mittelpunkt (ebd.: 219). Bei der vierten Perspektive, der *kulturellen*, geht es schliesslich darum, von welchen moralischen Werten die Tätigkeiten innerhalb der Institutionen beeinflusst werden (ebd.: 219). Mit «Wir alle spielen Theater» argumentiert Goffman, dass neben den vorhin erwähnten bereits bestehenden vier Perspektiven (*technische, politische, strukturelle, kulturelle*) nun noch eine fünfte dazukommt, nämlich die *dramaturgische Perspektive* (ebd.: 219).

Bei dieser von Goffman neu eingeführten *dramaturgischen Sichtweise*, liegt das Interesse in den Techniken der Eindrucksmanipulation, die innerhalb einer *Institution* verwendet werden

(ebd.: 219). Goffman besagt also, dass das Zusammenspiel zwischen verschiedenen *Einrichtungen* aus diesen fünf Blickwinkeln betrachtet werden kann. Er ist sich dabei bewusst, dass sich die Perspektiven überschneiden, da es ja lediglich um eine andere Priorisierung verschiedener Eigenschaften geht (ebd.: 219).

Da Goffman aussagt, dass *gesellschaftliche Einrichtungen* unter dem Aspekt der Eindrucksmanipulation untersucht werden können (ebd.: 217), kann angenommen werden, dass die erwähnten Perspektiven nicht nur bei der Analyse auf der Ebene *Institution*, sondern auch auf anderen Ebenen, wie beispielsweise derjenigen der *Einrichtung*, anzuwenden sind.

Ähnlich zu Goffman, der zwischen der Ebene der *Institution*, den *gesellschaftlichen Einrichtungen*, sowie der *Interaktion* unterscheidet (ebd.: 218), differenziert Latour ja ebenfalls zwischen verschiedenen Ebenen. Namentlich jener der *Mittler:in* (*Assoziationsketten*), einer Kette von *Mittler:innen* (*Netzwerk*), sowie *Konskriptionen* (Latour 1996: 51).

Wie bereits etabliert, lässt sich die Ebene der *Assoziationsketten*, mit jeweils einem:einer *Mittler:in* in deren Zentrum, vergleichen mit der Ebene der *Interaktion* nach Goffman. Wenn wir uns zurückerinnern, definiert Latour *Konskriptionen* als Ensembles von *Assoziationsketten*, die aufeinander abgestimmt sind. *Konskriptionen* scheinen folglich *gesellschaftlichen Einrichtungen* zu entsprechen, wo die *Darstellungen* und somit auch *Interaktionen* stattfinden.

Mit der Ebene der *Institution* hat Goffman noch einen Schritt mehr abstrahiert, als Latour das getan hat. Da in meiner Arbeit aber vor allem die *Interaktion* und *Transformation* im Vordergrund stehen, spielt dies nicht so eine bedeutende Rolle und daher werde ich dies zum jetzigen Zeitpunkt einfach als gegeben nehmen.

In «Wir alle spielen Theater» beschreibt Goffman *gesellschaftliche Einrichtungen* als Orte, an denen regelmässig eine bestimmte Art von Tätigkeit ausgeführt wird (Goffman 2017: 217). Diese *Einrichtungen* können aus den fünf oben aufgezeigten Perspektiven analysiert werden. Die Anwendung dieser Perspektiven schränkt Goffman jedoch wie erwähnt nicht nur auf die Einrichtungsebene ein.

Wie im vorherigen Kapitel dargelegt, beschreibt Latour *Transformationen* unter anderem mit Hilfe des Beispiels des Schlüsselanhängers, bei welchem Moral auf einen Gegenstand ausgelagert wird (Latour 1996: 53). Wenn nun bei Latour, je nach *Transformation*, die Verschiebung der Moral oder Disziplin im Vordergrund steht (ebd.: 53) und bei Goffmans *kultureller Perspektive*, die Betrachtung eben dieser Moral im Zentrum steht (Goffman 2017: 218), heisst dies, dass Latour den Schlüsselanhänger aus einer *kulturellen Perspektive* nach Goffman betrachtet hat. Manchmal steht bei Latour mit den *Vor-Einschreibungen*, sprich den vorgängigen Verteilungen von Kompetenzen (Latour 1996: 75) und den damit gesetzten Erwartungen, aber auch die *politische Perspektive* im Fokus. In anderen Teilen seiner Analyse ist dies jedoch auch die *technische*, nämlich wenn er das Verhältnis zwischen *Aktionsprogramm*, *Gegenprogramm* und *Gegen-Gegenprogramm* betrachtet. Denn die Vorhersehbarkeit und damit der Grad der Wirksamkeit hängt von deren Zusammenspiel ab (ebd.: 54). Latour wendet also implizit die Perspektiven von Goffman bei seiner Analyse an.

TRANSFORMATION UND AUSLAGERUNG DES SELBST

Laut Goffman (Goffman 2017: 231) machen sich Beobachter:innen ein Bild von Einzelpersonen, und schreiben ihnen somit ein *Selbst* zu. Dieses zugeschriebene *Selbst* entspringt jedoch nicht der Person, die es besitzt, respektive darstellt, sondern der Gesamtheit deren Handlungen (ebd.: 231). Das *Selbst* wird also von Merkmalen erzeugt, die beobachtenden Personen helfen, die Geschehnisse zu interpretieren (ebd.: 231).

Wenn nach Goffman das *Selbst* als die Gesamtheit der Handlungen einer Person definiert wird (ebd.: 231) und die *Darstellung* als die Gesamttätigkeit einer Person in bestimmten Situationen (ebd.: 18), folgt daraus, dass das *Selbst* aus der *Darstellung* einer Person konstituiert wird.

Wie bereits dargelegt, kann die *Darstellung* nach Goffman mit der *Transformation* von Latour gleichgesetzt werden. Ein Problem, das bisher jedoch noch besteht, ist die Tatsache, dass Goffmans *Interaktionen* auf menschliche Individuen in unmittelbarer physischer Präsenz begrenzt sind, während *Transformationen* nach Latour von dieser Einschränkung ausgenommen sind.

In seiner Theorie führt Goffman aus, dass das *Selbst* (als dargestellte Rolle) kein organisches Ding mit spezifischem Ort ist (ebd.: 231). Folglich kann das *Selbst* also auch unabhängig vom

Körper, den es darstellt, existieren. Schliesslich sagt Goffman selbst, dass der Körper der Darsteller:in, nur ein vorübergehender Aufhänger für etwas während der *Darstellung* oder *Interaktion* gemeinsam Hergestelltes ist (ebd.: 231).

Ausgehend von der Annahme, dass das *Selbst* ortsunabhängig existiert, möchte ich schlussfolgern, dass das *Selbst* also vom Körper der darstellenden Person ausgelagert werden kann. Ähnlich wie die Moral bei Latour während der *Transformation* in eine:n *Mittler:in* ausgelagert wird, kann das *Selbst* (oder zumindest Teile davon) während der *Darstellung* (die einer *Transformation* entspricht) auf nicht-menschliche Gegenstände übertragen werden. Damit löst sich nicht nur der Widerspruch der Restriktion von Goffmans Theorie auf menschliche Individuen auf, sondern auch das Problem der physischen Präsenz. Wenn nämlich das *Selbst* auf einen Gegenstand ausgelagert wird, kann mit dem *Selbst* der darstellenden Person via Gegenstand interagiert werden, ohne dass die darstellende Person selbst physisch anwesend ist. Die *Interaktion* findet also über das nicht-menschliche Wesen statt (bzw. anhand der *Interaktion* mit dem nicht-menschlichen Wesen).

Mithilfe des Vergleichs der Theorien von Latour und Goffman, lässt sich also aufzeigen, dass sich Goffmans *Theorie der Selbstdarstellung* auch auf *Interaktionen* mit nicht-menschlichen Wesen anwenden lässt. Denn wie vorhin aufgezeigt, stellen *Interaktionen* mit nicht-menschlichen Wesen eine *Interaktion* mit dem ausgelagerten *Selbst* der Darsteller:innen dar. Dass sich Latours *Akteur-Netzwerk-Theorie* auf die Softwareentwicklung anwenden lässt, haben wir ja schon gesehen. Nun da wir wissen, dass sich Goffmans Theorie auch bei nicht-menschlichen Wesen anwenden lässt, möchte ich im nächsten Abschnitt anhand der Verknüpfung der erweiterten Theorie Goffmans und den bereits gezogenen Parallelen zwischen der *Akteur-Netzwerk-Theorie* aufzeigen, was dies in Bezug auf die Analyse des Prozesses der Softwareentwicklung genau bedeutet.

Anwendung der erweiterten Theorie Goffmans auf die Softwareentwicklung

Ziel dieses Unterkapitels ist es aufzuzeigen, wie Goffmans Theorie in Verbindung mit Latour helfen kann, den Prozess der Softwareentwicklung zu analysieren. Es geht somit darum, die auf nicht-menschliche Wesen erweiterte *Selbstdarstellung*, auf dem Bereich der Software anzuwenden.

Anhand der bisher gezogenen Parallelen zwischen Latour und Goffman, lassen sich bereits einige der Begrifflichkeiten aus den aufgezeigten Verbindungen zwischen Softwareentwicklungsprozess und Latour übernehmen. So sind die dem *Aktionsprogramm* gleichgesetzten «user stories» Teil der Darstellung. Denn wie im vorherigen Kapitel dargelegt, gehört alles in Bezug zum *Aktionsprogramm* zur Darstellung und alle *Gegenprogramme* werden als Störfaktoren angesehen. Somit sind Weigerungen oder Ignoranz gegenüber den Hinweisen bezüglich des Erstellens sicherer Passwörter, wiederum als *Störfaktoren* anzusehen.

Nach Goffman sind Individuen, die gemeinsam eine Rolle aufbauen, Teil eines *Ensembles* (Goffman 2017: 75). Die nebst dem *Ensemble* an der *Interaktion* teilnehmenden Personen bilden, aufgrund ihrer diversen Reaktionen, wiederum selbst ein *Ensemble* (ebd.: 85). Es findet eine sogenannte *dramatische Interaktion* zwischen diesen beiden *Ensembles* statt (ebd.: 85).

In seiner Analyse lässt Goffman der Vereinfachung halber ausser Acht, dass das *Ensemble*, welches als Zuschauer bezeichnet wird, selbst in der Regel auch eine Darstellung bietet (ebd.: 86). Ähnlich zu Goffman in Bezug auf die Zuschauer, werde ich für den Moment nicht darauf eingehen, inwieweit User:innen bei der Benutzung von Software, oder die Kund:innen im Entwicklungsprozess ihrer «custom made» Software, sich selbst darstellen. Von Interesse ist für mich zu diesem Zeitpunkt hauptsächlich die Darstellung auf Seite der Entwickler:innen und wie diese durch die Software ortsunabhängig erweitert wird.

AUSLAGERUNG DES SELBST

Bei den Ausführungen zur Auslagerung der Moral in der Softwareentwicklung habe ich das Beispiel der Passwortsicherheit angeführt. Beim Forcieren des Einhaltens von verschiedenen Massnahmen zum Erstellen von Passwörtern geht es aber nicht nur darum, dass die Moral von User:innen in Software ausgelagert wird. Ich will aufzeigen, dass zusätzlich Annahmen oder auch Vorurteile der Entwickler:innen über die Nutzer:innen in dieser Auslagerung zum Ausdruck kommen. Konkret meine ich dabei beispielweise die Annahme von Entwickler:innen, dass sich Nutzer:innen sowieso nicht an Hinweise zum Erstellen eines sicheren Passworts halten würden. Folglich stellen sich, durch diese Auslagerung ihrer Vorurteile, Entwickler:innen als besorgt um die Sicherheit des Benutzerkontos der User:innen dar. Dies, indem sie ihnen vorschreiben, die Sicherheitsmassnahmen befolgen zu müssen. Der Teil des Selbst, der sich als um die Passwortsicherheit der Benutzer:innen besorgt ausgibt, wird folglich ebenfalls in die Software ausgelagert. Ob es den Entwickler:innen dabei wirklich von Belang ist, ob das gewählte Passwort sicher ist, sie einfach Produkthanforderungen erfüllen, oder sie schlicht konventionelle Massnahmen ergreifen, ist jedoch unklar.

Ähnlich sieht es mit den «user personas» aus. Beim Erstellen dieser «personas» wird oftmals auf stereotypisierte User:innen zurückgegriffen. Beispiele für solche «personas» können eine IT-Administratorin sein, die ein grosses technisches Know-how besitzt und daher Zugriff auf erweiterte Funktionen erhält, um die gewöhnlichen Nutzer:innen unterstützen zu können. Ein anderes Beispiel kann aber auch eine Person sein, die zwar mit den inhaltlichen Themen sehr vertraut ist, deren technisches Know-how aber sehr beschränkt ist. Entwickler:innen bauen folglich die Software nach den antizipierten «personas», welche nach den Erwartungen (inkl. unbewussten Vorurteilen) erstellt wurden, die sie gegenüber den vordefinierten Gruppen von Nutzer:innen haben. Folglich fliessen dort viele Annahmen, auch vom angeeigneten Wissen abhängige, in die Software ein.

Ein weiteres Beispiel dafür, dass die etablierte Auslagerung des *Selbst* auf nicht-menschliche Wesen dazu dienen kann, den Einfluss von bewussten oder unbewussten Annahmen zu analysieren, sind Screenreader. Screenreader lesen in der Regel den auf einer Webseite

vorhandenen Text oder Alternativtext¹⁸ bei graphischen Elementen vor¹⁹. Ich wage zu behaupten, dass die meisten Leute nicht besonders viel Erfahrung in der Benutzung von Screenreader haben, da sie sie selbst nicht benutzen. Ähnlich sieht es mit dem Wissen, wie solche Screenreader tatsächlich funktionieren, aus. Auch ich bin von dieser Aussage nicht ausgeschlossen. Wenn ich an einer Applikation arbeite, denke ich in der Regel nicht daran, diese explizit für Screenreader zu testen. Als Person, deren visuelle Wahrnehmungsfähigkeit nicht massgeblich eingeschränkt ist, steht das Besuchen, beziehungsweise Verwenden, einer Webseite mithilfe eines Screenreaders nämlich oftmals nicht im Fokus, und tendiert meiner Erfahrung nach vergessen zu gehen. Die Tatsache, dass die Barrierefreiheit nach meiner Wahrnehmung selten eine explizit gewünschte Voraussetzung ist, hilft dabei auch nicht. Aufgrund der meist eher beschränkten Ressourcen (sei dies Budget, Zeit o.Ä.), programmiert man nämlich in der Regel nichts, was nicht zumindest implizit den Bedingungen dem:der Auftraggeber:in entspricht²⁰.

So kann es vorkommen, dass die graphische Benutzeroberfläche für Menschen mit Sehbehinderung unzugänglich ist. Dies, weil oftmals unbewusst davon ausgegangen wird, dass die Webseite unabhängig von Screenreader, also von Personen ohne Sehbehinderung, verwendet wird.

Viele Frameworks²¹ haben bereits eingebaute Funktionen, die die Barrierefreiheit erhöhen sollen. Das bereits erwähnte Framework Angular erlaubt es einem zum Beispiel sogenannte «aria-labels» zu definieren²². Solche Labels können eingesetzt werden, um die für Personen mit Sehbehinderung nicht oder nur schwer identifizierbaren Elemente besser interpretierbar

¹⁸ Alternativtexte sind eine textliche Beschreibung eines graphischen Elements (bspw. Bild), die auch bei Ladeproblemen angezeigt werden.

Grafiken und Bilder. Barrierefreiheit | Schulung, Begleitung und Tests.

<<https://www.netz-barrierefrei.de/wordpress/barrierefreies-internet/barrierefreie-redaktion/grafiken-und-bilder/>>. 28. April 2022.

¹⁹ 2018: What Is A Screen Reader And How Does It Work?. University of Connecticut.

<<https://accessibility.its.uconn.edu/2018/08/22/what-is-a-screen-reader-and-how-does-it-work/>>. 20. April 2022.

²⁰ Oftmals unter YAGNI-Prinzip zusammengefasst, kurz für «You ain't gonna need it».

YAGNI-Prinzip. t2informatik.

<<https://t2informatik.de/wissen-kompakt/yagni-prinzip/>>. 23. April 2022.

²¹ Zur Erinnerung: Frameworks sind eine Art Grundgerüst, welches bestimmte Elemente und Funktionen zur Verfügung stellt.

Hery-Mossman, Nicole 2021: Was ist ein Framework? - einfach erklärt. CHIP. <https://praxistipps.chip.de/was-ist-ein-framework-einfach-erklart_41348>. 30. März 2022.

²² 2022: Accessibility in Angular. Angular.

<<https://angular.io/guide/accessibility>>. 26. April 2022.

zu gestalten²³. Jedoch nützen solche Funktionen nur, insofern sie auch korrekt bzw. überhaupt eingesetzt werden.

Gegebenenfalls kann es auch sein, dass gewisse Nutzungsszenarien nicht abgedeckt werden, denn wie schon erwähnt kann aufgrund von beschränktem Budget nicht alles perfekt implementiert werden. Oder auch weil die zugängliche Implementation von Kund:innen explizit nicht gewünscht wird. Einschränkungen im Budget, der Zeit, oder auch fehlendes Wissen, sind demnach *Störfaktoren* in der Darstellung von Entwickler:innen, die sich als möglichst kompetent in der Softwareentwicklung darstellen möchten. Ebenso ist das fehlende Wissen um die Funktionalität der Screenreader als *Störfaktor* anzusehen. Schliesslich beinhaltet Kompetenz die Abdeckung aller möglichen Nutzungsszenarien, da so das Risiko von misslungenen Zugriffen auf die Software vermieden werden kann.

Als letztes konkretes Analysebeispiel möchte ich hier Farbpaletten erwähnen, die für graphische Elemente verwendet werden. Ich probiere jeweils mit dem Adobe Color Barrierefreiheits-Tools²⁴ die verwendeten Farben anzupassen, damit möglichst wenig Probleme für Personen mit Farbenblindheit auftreten. Je mehr Farben jedoch benötigt werden, desto schwieriger ist das Zusammenstellen einer Farbpalette ohne Konflikte. Wenn dann noch gewisse Farbwünsche der Kundschaft berücksichtigt werden sollen, wird es nur noch schwieriger. Da das Verwenden einer barrierefreien Farbpalette in der Regel aber ebenfalls nicht unbedingt eine explizit (von Kund:innen) gewünschte Voraussetzung ist, kann es auch schnell einmal vergessen werden. Zusätzlich ist es als nicht-farbenblinde Person auf die Schnelle schwer abzuschätzen, wie einschränkend gewisse Farbkombinationen sind. Bei den erwähnten «user personas» werden auch nicht unbedingt «personas» erstellt für Personen, die eben beispielsweise farben- oder ganz blind sind. Zumindest nicht bei Projekten in der Grösse, in der ich gearbeitet habe. Es geht dabei oftmals eher darum, das technische Kompetenzlevel, und je nach Projekt, auch die Verständlichkeit der präsentierten Inhalte an die antizipierten Nutzer:innen anzupassen. Daher wird das Problem auch auf diese Weise nicht behoben.

²³ Kearney, Meggin, Dave Gash und Alice Boxhall 2020: Introduction to ARIA. web.dev. <<https://web.dev/semantics-aria/>>. 20. Februar 2022.

²⁴ Barrierefreiheits-Tools. Adobe Color. <<https://color.adobe.com/de/create/color-accessibility>>. 21. April 2022.

In einem kurzen Einschub möchte ich hier erwähnt haben, dass ich, in dem ich anhand meiner Eigenerfahrung aufzeige, welche Fehler und falschen Annahmen ich selbst mache, die Darstellung meines *Selbst* teilweise dekonstruiere. In Goffmans Terminologie ausgedrückt, gewähre ich Einblick auf die *Hinterbühne* der Darstellung meines *Selbst*, in der Rolle als Softwareentwicklerin. Der Begriff der *Hinterbühne* beschreibt einen zur Vorstellung gehörenden Ort, der dem Publikum in der Regel nicht zugängliche ist, und an dem die Eindrücke der Darstellung bewusst widerlegt werden (Goffman 2017: 104f). Damit gehe ich gewissermassen das Risiko ein, meine Rolle der kompetenten Softwareentwicklerin unglaubwürdig zu machen. Würde ich aber mich selbst nicht kritisch hinterfragen, würde dies meine Rolle als Studentin der Sozialanthropologie unglaubwürdiger machen. Denn die kritische Selbstreflexion ist gerade im Bereich des Autoethnografischen essenziell. Eine Frage, die sich nun stellt, ist, ob ich mich entschieden habe, der Rolle als Sozialanthropologin in dieser Arbeit stärker zu gewichten, weil das Publikum (hier Personen, die diese Arbeit vermutlich lesen werden) eher Personen sind, bei denen ich meine Rolle als Studentin der Sozialanthropologie übernehme und weniger Personen, die mich aus der Rolle der Softwareentwicklern sehen.

Im Verlauf dieser Arbeit habe ich bereits mehrfach erwähnt, dass bei der Softwareentwicklung oftmals auch in Teams an derselben Applikation gearbeitet wird. Auch in diesem Setting findet die erweiterte Theorie Anwendung. Diese Teamarbeit entspricht nämlich der Zusammenarbeit des *Ensembles* bei Goffman (ebd.: 85). Ein *Ensemble* arbeitet zusammen um eine bestimmte Situation darzustellen (ebd.: 217). Ähnlich dazu arbeiten Entwickler:innen zusammen, um via Software jeweils einen bestimmte Situation (bspw. erfolgreiches Login) darzustellen. In anderen Worten konstituieren diese ausgelagerten *Selbsts* der Entwickler:innen eine gemeinsame Darstellung, nämlich die der funktionierenden Software. Wie Goffman ausführt, ist die Definition einer Situation schliesslich integraler Bestandteil einer Darstellung, die von verschiedenen Teilnehmenden gemeinsam geschaffen und erhalten wird (ebd.: 73).

Fazit und Ausblick

Als Rekapitulation der Arbeit kann gesagt werden, dass erfolgreich Parallelen zwischen Goffmans *Theorie der Selbstdarstellung* und Latours *Akteur-Netzwerk-Theorie* aufgezeigt wurden. Als Konsequenz dessen, konnten die beiden Theorien verbunden werden, um Goffmans Theorie zu erweitern. Die Erweiterung von Goffmans *Theorie der Selbstdarstellung* mithilfe von Latour erlaubt, anhand der aufgezeigten Auslagerung des *Selbst*, argumentieren zu können, dass Goffmans Theorie auch auf nicht-menschliche Akteur:innen angewendet werden kann. Dies anhand der Tatsache, dass aufgezeigt wurde, wie Eigenschaften des Selbst auf nicht-menschliche Akteur:innen ausgelagert werden können.

Angewandt auf die Softwareentwicklung bedeutet dies, dass Goffmans Theorie verwendet werden kann, um die Einflüsse der von den Entwickler:innen getroffenen Entscheidungen oder deren Annahmen auf die Software zu untersuchen. Diese Übertragung ist auch deswegen möglich, da in einem ersten Schritt bereits aufgezeigt wurde, dass Software als *Mittler:in* nach Latour betrachtet werden kann.

Einen Punkt den ich nochmals aufgreifen möchte, ist die Tatsache, dass in dieser Arbeit die darstellende Funktion vom Publikum respektive der Nutzer:innen aussen vor gelassen wurde. Auf dieser Seite der *Interaktion* existieren jedoch auch diverse Möglichkeiten das Thema weiter zu vertiefen. Gerade in Bezug auf soziale Medien oder Dating Apps findet eine offensichtliche Selbstdarstellung statt. Spannend wäre dabei beispielsweise zu untersuchen, inwiefern die in die Apps oder Webseiten einprogrammierten Annahmen von Entwickler:innen, die Selbstdarstellung von Nutzer:innen einschränken.

Ein letzter weiterführender Gedanke, den ich hier anbringen möchte, ist ein Hinweis darauf, dass ich in dieser Arbeit das behandelte Thema noch lange nicht aufs Ganze ausgeschöpft habe. Es gibt noch weitere Aspekte der verwendeten Theorien, die sich in Bezug auf die Softwareentwicklung ausdiskutieren lassen. Eine vollumfängliche Analyse hätte den Rahmen dieser Arbeit schlicht gesprengt.

Abschliessend kann gesagt werden, dass Goffmans Theorie auch bei Interaktionen in einer zunehmenden digitalisierten Gesellschaft Anwendung findet. Gerade wenn sie erweitert wird und somit auch auf nicht-menschliche Wesen und uneingeschränkt von direkter physischer Anwesenheit Anwendung findet, bleibt ihre Relevanz weiterhin vorhanden. Sie bietet einen möglichen Ansatz zu untersuchen, wie Annahmen und Weltansichten von

Entwickler:innen in die Software einfließen können. Ein Thema, das mit der stetig zunehmenden Digitalisierung, mit Sicherheit weiterhin von grosser Relevanz sein wird.

Literaturverzeichnis

- 2018: What Is A Screen Reader And How Does It Work?. University of Connecticut.
<<https://accessibility.its.uconn.edu/2018/08/22/what-is-a-screen-reader-and-how-does-it-work/>>. 20. April 2022.
- 2020: Passwort-Entropie. ComputerWeekly.de.
<<https://www.computerweekly.com/de/definition/Passwort-Entropie>>. 30. März 2022.
- 2021: Grundlegendes zu Webanwendungen. Adobe.
<<https://helpx.adobe.com/de/dreamweaver/using/web-applications.html>>. 20. Februar 2022.
- 2022: Accessibility in Angular. Angular.
<<https://angular.io/guide/accessibility>>. 26. April 2022.
- Barrierefreiheits-Tools. Adobe Color.
<<https://color.adobe.com/de/create/color-accessibility>>. 21. April 2022.
- Ellis, Carolyn, Tony E. Adams und Arthur P. Bochner 2011: Autoethnography: An Overview. *Historical Social Research* 36 (4): 273-290.
- Goffman, Erving ¹⁷2017: Wir alle spielen Theater. München: Piper.
- Grafiken und Bilder. Barrierefreiheit | Schulung, Begleitung und Tests.
<<https://www.netz-barrierefrei.de/wordpress/barrierefreies-internet/barrierefreie-redaktion/grafiken-und-bilder/>>. 28. April 2022.
- Hery-Mossman, Nicole 2021: Was ist ein Framework? - einfach erklärt. CHIP.
<https://praxistipps.chip.de/was-ist-ein-framework-einfach-erklart_41348>. 30. März 2022.
- Introduction to the Angular Docs. Angular.
<<https://angular.io/docs>>. 28. März 2022.
- Kearney, Meggin, Dave Gash und Alice Boxhall 2020: Introduction to ARIA. web.dev.
<<https://web.dev/semantics-aria/>>. 20. Februar 2022.
- Latour, Bruno 1996: Der Berliner Schlüssel. Erkundungen eines Liebhabers der Wissenschaften. Berlin: Akademie Verlag.

Latour, Bruno 2001: Eine Soziologie ohne Objekt? Anmerkungen zur Interobjektivität. *Berliner Journal für Soziologie* 11 (2): 237-252.

Miller, Daniel und Heather A. Horst 2012: The Digital and the Human: A Prospectus for Digital Anthropology. In: Miller, Daniel und Heather A. Horst (Hg.): *Digital Anthropology*. London: Berg Pub Ltd.

Pogue, David 2011: Password Prevented. *Scientific American* 305 (3): 36–37.

Schulz, Sven 2022: Was ist ein sicheres Passwort? Die besten Tipps. CHIP. <https://praxistipps.chip.de/was-ist-ein-sicheres-passwort-die-besten-tipps_3482>. 30. März 2022.

Silva, Selena und Martin Kenney 2018: Algorithms, Platforms, and Ethnic Bias: An Integrative Essay. *Phylon* 55 (1&2): 9-37.

Sim, Ariel 2019: How Might We Future Proof User Experience. <https://www.youtube.com/watch?v=eqTTD2tI_ml>. 30. März 2020.

Sommerville, Ian 2020: *Engineering Software Products. An Introduction to Modern Software Engineering*. Hoboken, NJ: Pearson Education, Inc.

YAGNI-Prinzip. t2informatik. <<https://t2informatik.de/wissen-kompakt/yagni-prinzip/>>. 23. April 2022.

Selbstständigkeitserklärung

„Ich erkläre hiermit, dass ich die vorliegende Arbeit mit dem Titel «Software als Erweiterung des Selbst von Softwareentwickler:innen» selbstständig verfasst, bisher weder ganz oder in Teilen als Prüfungsleistung vorgelegt und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Ich bin mir bewusst, dass ich andernfalls ein Plagiat beziehungsweise einen Betrug begangen habe, der als schweres akademisches Fehlverhalten Sanktionen nach sich zieht.“

Bern, 30.04.2022



Kerrie Lesli Stauffer