

Analphabeten können nicht Programmieren

Daß Deutschland in dem weiten Bereich, der heute unter dem Kürzel IT in aller Munde ist, international tatsächlich bedeutungslos ist, wird von Politik und Gesellschaft einerseits sorgenvoll registriert, zugleich aber mit geradezu fahrlässiger Selbstverständlichkeit gelassen, wie es ist. Und egal, ob Hardware oder Software im Fokus der Kritik sind: Es gibt, mit einer Ausnahme, praktisch kein bedeutendes deutsches IT-Unternehmen, das weltweit aufgestellt ist. Die Zeit großer deutscher Ingenieursleistungen liegt einige Jahre zurück und sie wurden in traditionellen Bereichen wie der chemischen oder der Autoindustrie erreicht und bleiben, wie es aktuell aussieht, auf sie beschränkt. Ein Zuse macht noch keine IT-Industrie.

Immerhin hat die Politik seit kurzem erkannt, daß es mit der IT schon in Deutschland selber nicht so weit her ist. Unter dem Losungswort ›Digitalisierung‹ wurde daher zur Offensive geblasen. Erstes Ziel waren und sind die Schulen im ganzen Land; sie sollen, umfassend mit entsprechender Hardware ausgestattet, zuerst die Schüler und dann das Land nach vorne bringen. Als wären Schiefertafeln und Kreide der Grund allen Übels.

Doch schon hier zeigt sich das geringe Verständnis der Verantwortlichen für die Misere, in der sich Deutschland befindet und warum seine Schüler so weit hinterherhinken: Denn nicht fehlende Strippen und WLAN-Router sorgen dafür, daß die IT-Industrie dieses Landes bedeutungslos ist - es ist die stiefmütterlich behandelte Entwicklung von Software, dem Lebenssaft des Computers, die Deutschland ins Hintertreffen geraten und in den vergangenen zwei Jahrzehnten zum Entwicklungsland degradierte. Ausgerechnet in der Kerndisziplin aller IT - zur Erinnerung: Ohne Software läuft auf dem Computer gar nichts - ausgerechnet in der Kerndisziplin kommen Produkte aus Deutschland nicht vor.

Als Grund für diese essentielle Schwäche wird, falls überhaupt jemand offen darüber spricht, die mangelhafte Ausbildung der Schüler in den

sogenannten MINT-Fächern angeführt: Mathematik, Informatik, Naturwissenschaften und Technik; was, darüber besteht, selbst wenn die sehr unterschiedlichen Verhältnisse in Berlin und Bremen auf der einen und Bayern auf der anderen Seite berücksichtigt werden, kein Zweifel, ja auch stimmt. Erst vor kurzem erzählte mir eine Mathematiklehrerin aus der Ukraine überrascht, daß es in Deutschland schick wäre, kein Mathe zu können und das nicht nur bei Schülern; etwas, das in Kiew oder Charkiv undenkbar sei.

Doch bei all den lebhaften Diskussionen über die MINT-Fächer und deren Bedeutung wird übersehen, daß es noch einen weiteren Grund gibt, warum Deutschland hinsichtlich Software so schlecht aufgestellt ist. Und dieser Grund liegt in einer ganz anderen Ecke der Bildungslandschaft; in einer Ecke, wo man üblicherweise Ursachen für Schwierigkeiten im Umgang mit Mathematik und Naturwissenschaften, Informatik und Technik eher nicht vermutet: In der Vernachlässigung der deutschen Sprache.

Nicht daß über die Geringschätzung des Deutschen nicht schon geklagt worden wäre. Da wird die Sprache der jungen Leute als flach kritisiert, es werden Anglizismen moniert und wer hätte nicht schon einmal über den Sprachverfall nachgedacht. Die Schulkinder können zum Teil nicht einmal mehr in Schreibrift in ihr Tagebuch schreiben, Rechtschreibung wird in den Schulen immer später verlangt, von der Grammatik will man gar nicht erst reden. Doch als Grund für das Desaster an der IT-Front wird das Desaster im Bereich Deutsche Sprache nirgends genannt. Da bestimmt das hergebrachte Diktum, Menschen seien entweder sprachlich-künstlerisch oder mathematisch-logisch begabt, die Sicht auf die Dinge.

Indes ist die Grundlage aller laufenden Software das Programm; oder wie es heute heißt: Die App. Und Apps werden in Sprachen geschrieben, die nicht zufällig Programmiersprachen heißen, ausdrücklich Sprachen. Denn wie jede andere Sprache haben Programmiersprachen ein Vokabular, sie werden von einer Grammatik beherrscht und ihre Texte haben ein Schriftbild. Das Vokabular entspricht im wesentlichen den zugelassenen Zeichen und dem Befehlssatz; er ist begrenzt und fest definiert. Die Grammatik basiert auf dem, was übrig geblieben ist von den zähen und vergeblichen Versuchen, eine Universalgrammatik menschlicher Sprachen zu finden: Dem Regelwerk für Sprachen, die eine Maschine verarbeiten

kann; es ist, wie schon das Vokabular, begrenzt und fest definiert. Vokabular und Grammatik dienen der Kommunikation mit der Maschine. Und wenngleich Programmiersprachen, ähnlich wie mathematische Formeln, nicht oder nur selten laut gesprochen werden - mit den entstandenen Texten werden Algorithmen beschrieben und auf diesem Weg dienen sie zugleich der Verständigung mit Schülern, Kollegen und schließlich sogar mit sich selber, wenn man später noch einmal nachlesen will, was genau denn programmiert werden sollte. Daher ist auch das Schriftbild von zentraler Bedeutung: Es muss lesbar sein und verständlich.

Wenn nun aber Programmiersprachen für die Softwareentwicklung essentiell sind, dann sind es auch die drei Elemente: Vokabular, Grammatik und Schriftbild. Ohne sie bleibt einem Schüler der Zugang zu den Ausdrucksformen dieser Sprachen verschlossen; zumindest wird der Zugang deutlich schwerer gemacht - mit gravierenden Folgen für die Softwareentwicklung. Denn anders als in der Kommunikation in einer natürlichen Sprache wie Deutsch oder Englisch, kennt der Rechner kein Pardon beim sprachlichen Ausdruck: Was nicht korrekt formuliert worden ist, nimmt er nicht an. Und der Hinweis, der beim menschlichen Gegenüber fast immer hilft: ›Du weiß doch, was ich gemeint habe ?‹, diese Hilfestellung ist hier nicht einmal erwünscht. In keinem Fall soll die Maschine nach Gutdünken deuten. Rechtschreibung und Grammatik sind beim Programmieren also entscheidend. Und was in einem krausen Schriftbild ausgedrückt wurde, versteht ein Programmierer selber schon nach wenigen Tagen nicht mehr, von den Kollegen erst gar nicht zu reden.

Ein Sinn fürs Schriftbild, ein Pflichtgefühl gegenüber richtiger Schreibung, ein Gefühl für Grammatik - sie sind die Voraussetzung fürs Programmieren; zumindest sollte der Schüler die Bedeutung aller drei Elemente verstehen. Das mag überraschen. Für gewöhnlich wird Programmierung mit logischem Verständnis verknüpft und man wird einwenden können, am Ende sei das logische Denken entscheidend und nicht Schriftbild, Rechtschreibung oder Grammatik. Doch genau das ist ein Irrtum. Wer die Mittel der Sprache und das Gesagte trennt, hat im Grunde Sprache nicht wirklich verstanden. Es ist nicht nur die Logik des ausgedrückten Gedankens, die den Umgang mit der Sprache bestimmt - es ist auch der sprachliche Ausdruck an sich. - Was damit gemeint ist ?

Man nehme etwa das Schriftbild. Ja, es ist, wie schon angedeutet, für die Verständigung nötig. Aber tatsächlich ist es noch mehr. In einem Seminar zur Programmiersprache R, einer Sprache zur Formulierung statistischer Auswertungen, sprach ich vor einem gemischten Publikum - teils IT-Fachleute, teils Anfänger - von der ›Zeichenlust‹, die man bräuchte, um ein guter Programmierer werden zu können. Die Spezialisten stutzten kurz, stimmten mir aber dann lachend umgehend zu; die Anfänger blieben skeptisch: Zeichenlust? Was sollte das sein? - Nun, Zeichenlust ist die Freude am Schriftbild! Das reicht von der Versstruktur eines Gedichts über eine mathematische Formel zum Schriftbild eines Computer-Programms. Wer diese Freude nicht teilt, der wird in den Bereichen nie zu etwas bringen, denn die Zeichen sind nun einmal das Handwerkszeug von Dichtern, Mathematikern und eben Programmierern. Es ist wie bei jedem Künstler: Das Werkzeug ist die Verlängerung seines Gedankens, seiner Idee; beide sind nicht voneinander zu lösen. Umgekehrt wird daher ein Schuh draus: Weil auch Mathematik und insbesondere Logik ein Sprachsystem bilden, ist der herrschende Sprachverfall eine zentrale Ursache für den Niedergang Deutschlands in beiden Bereichen.

Somit ist klar: Der heutige, zunehmend ärmlischer werdende Sprachunterricht mit Politikern und Pädagogen, die Schrift, Rechtschreibung und Grammatik für unwichtig, gar hinderlich für die Ausdrucksmöglichkeit halten, dieser Sprachunterricht ist maßgeblich verantwortlich für die schlechte Position Deutschlands bei der Softwareentwicklung. Wer seine Muttersprache nicht richtig liest oder schreibt und ihre Grammatik verflucht, statt sie in Ehrfurcht und voller Dankbarkeit für den Halt, den sie im Wortschungel bietet, zu lernen, wird auch in keiner Programmiersprache lesen und schreiben. Die Laxheit im Umgang mit Sprache, wie sie sich seit den Partytagen der 1960er Jahre breitgemacht hat, führt also nicht nur zur Dummheit - das macht sie auch; sie führt in einem scheinbar so weit entfernten Gebiet wie der Handhabung des Computers zum Analphabetismus.

Aber besteht der Zusammenhang überhaupt speziell im Deutschen? Gilt er nicht generell? Und müsste nicht eigentlich Englisch noch mehr in den Vordergrund rücken, da praktisch alle Programmiersprachen auf der englischen Sprache basieren? - Ja, der Zusammenhang gilt generell: Es

geht um das Erlernen von Sprachen im Allgemeinen, nicht nur ums Deutsche. Aber mit dem Deutschen ist es vielleicht doch noch etwas mehr, weil sie zu den Sprachen mit einer komplexen Grammatik gezählt wird. Wer sie nicht mehr wirklich lernt und durch eine grammatikalisch schwächere Sprache ersetzt, bleibt schließlich auf flacherem Niveau hängen und entwickelt keinen Spürsinn für die Geflechte einer reichen Grammatik. Ihm entgehen die Tiefen, in die beide Sprachen führen und verführen können. Und damit auch ein Gefühl für die Tiefen der technischen Sprachen.

Und nicht nur für sie. Auch die Mathematik bleibt, zumindest in ihren formalen Ausdrucksweisen verschlossen. Denn der Zusammenhang besteht natürlich auch hier. Ist es ein Wunder, daß die großen Mathematiker der beiden vergangenen Jahrhunderte zuvorderst aus Frankreich und Deutschland stammten? Galois, Poincaré, Lebesgue; Cantor, Hilbert, Hausdorff. Und man bedenke: Bertrand Russell sprach fließend Deutsch. – Aber der Niedergang der Mathematik in Deutschland ist ein anderes, nicht weniger trauriges Thema.