Zeitschrift: IABSE congress report = Rapport du congrès AIPC = IVBH Kongressbericht

Band: 11 (1980)

Artikel: Software Engineering. Eine Notwendigkeit im Bauwesen

Autor: Hartmann, Dietrich

DOI: https://doi.org/10.5169/seals-11331

VII

Software Engineering — eine Notwendigkeit im Bauwesen

Software Engineering — A Necessity in Civil Engineering

Ingéniérie du logiciel — une nécessité en génie civil

DIETRICH HARTMANN
Priv.-Doz. Dr. -Ing.
Universität Dortmund
Dortmund, Bundesrepublik Deutschland

ZUSAMMENFASSUNG
Die Erstellung von SOFTWARE hat — verursacht durch den Preisrückgang bei der HARDWARE — industrielle Ausmasse angenommen. Nur durch den gezielten Einsatz der neu entstandenen Ingenieurdisziplin SOFTWARE ENGINEERING wird es möglich sein, die SOFTWARE Entwicklung und -Wartung in den Griff zu bekommen. Es wird definiert, was SOFTWARE ENGINEERING bedeutet und welche Konsequenzen sich für die Ausbildung, Praxis und Forschung im Bauwesen ergeben.

SUMMARY
Computer HARDWARE is becoming ever cheaper. On the other hand SOFTWARE costs have risen enormously. This trend can only be stopped if the principles, methods and tools of the so-called SOFTWARE ENGINEERING are applied. The article shows what is meant by SOFTWARE ENGINEERING and describes its consequences for program development in civil engineering.

RESUME
Le matériel devient toujours meilleur bon marché, le logiciel au contraire toujours plus cher. L'augmentation des frais dans le domaine du logiciel ne peut être contrôlée que par l'emploi de la science du logiciel (Software Engineering). Il s'agit dans cet exposé de montrer la signification de l'ingéniérie du logiciel et les conséquences de cette nouvelle discipline.

1. EINLEITUNG

Wer die neueste Entwicklung auf dem Computer-Markt studiert, stößt immer wieder auf Meldungen, welche besagen, daß die EDV im Bauwesen im Vergleich zu anderen Industriezweigen nach wie vor "Entwicklungsgebiet" ist [1]. Diese Feststellung wurde auch auf der im September 1979 in München stattgefundenen SYSTEMS 79 bestätigt, einer EDV-Messe, die sich immer mehr zum "EDV-Mekka" fürs Bauwesen entpuppt. Belegt werden derartige Behauptungen mit Hilfe statistischer Zahlen aus dem Jahr 1975 (vgl. [2]): der Anteil des Bauwesens am Bruttosozialprodukt beträgt danach rund 15%; der Anteil an den eingesetzten EDV-Anlagen dagegen nur 5%.

Es ist jedoch nicht überzeugend, den Anteil der eingesetzten Rechenanlagen im Bauwesen dazu zu benutzen, um das Bauwesen als "Entwicklungsgebiet" abzuqualifizieren. Man muß berücksichtigen, daß gerade im Bauwesen eine sehr große Zahl von kleineren Ingenieur- und Architekturbüros zu den 15% Anteil am Bruttosozialprodukt beiträgt und es für diese Büros sicherlich gar nicht so sinnvoll war, vor 1975 (als Bezugszeitpunkt der o.g. Erhebung) Rechner oder EDV zu betreiben, da aufgrund hoher Anschaffungs- oder Mietkosten die Wirtschaftlichkeit der Büros in Frage gestellt worden wäre. Der wichtigste Grund für den geringen EDV-Einsatz waren also hauptsächlich die hohen Kosten für die HARDWARE. Ein zweiter Grund war sicherlich auch der geringe Komfort im Hinblick auf die zu erfüllenden, in ihrer Art unterschiedlichsten Aufgaben, wie Planung, Statik, Dimensionierung, Zeichnungserstellung und Abrechnung. Nun haben sich in den letzten 5 Jahren die Verhältnisse enorm geändert: die HARDWARE wird billiger und billiger, der Komfort steigt. Im folgenden sollen insbesondere die durch den Preisverfall der HARDWARE eingetretene Situation dargestellt, zukünftige Trends aufgezeigt und Konsequenzen für Aus- und Fortbildung, Forschung und Entwicklung im EDV-Bereich des Bauwesens gezogen werden.

2. HARDWARE UND SOFTWARE SITUATION HEUTE

Es ist allgemein bekannt, wodurch der Preisverfall der HARDWARE bewirkt wurde: Die neue LSI-Technologie[1)], durch die Tausende elektronischer Bauelemente auf nur wenigen mm^2 großen Plättchen - den sogenannten chips - untergebracht werden können, erlaubt es, unsere heutigen Computer industriell und dadurch billig herzustellen. Die HARDWARE wurde aber nicht nur "spottbillig", sondern gleichzeitig auch noch schneller, betriebssicherer, robuster und in den Abmessungen erheblich kleiner. Gerade die Miniaturisierung der Elektronik hat den Markt in den letzten Jahren total verändert und Rechnertypen herausgebracht, die mit Sicherheit das Arbeiten im Bauwesen erheblich beeinflussen werden.

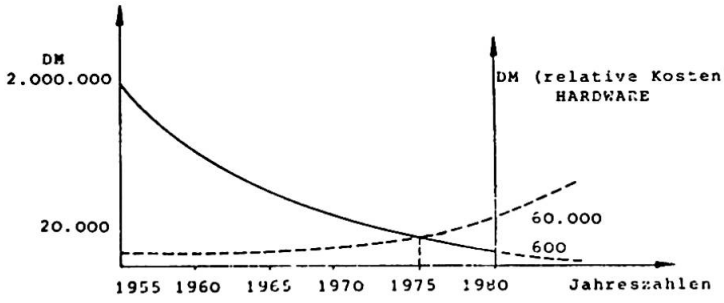
Um einen Überblick über den derzeitigen Stand der Technik zu erhalten, werden die verschiedenen Rechnertypen tabellarisch zusammengestellt:

[1)] LSI = Large Scale Integrated

Eigenschaften / Rechnertyp	Kosten	Output	Speicher	Sprache
programmierbarer Taschenrechner	100 DM bis 1000 DM	Papierrollen Printer	Magnet Karteien	system-abhängig mnemotech.
Personalcomputer	1000 DM bis 10000 DM	Printer, Plotter, Display	Kassette Floppy Disk	BASIC
Microcomputer	10000 DM bis 40000 DM	Printer, Plotter, Display	Floppy Disk	BASIC FORTRAN
Minicomputer	80000 DM bis 600000 DM	Printer Plotter Display	Disk magnetband	BASIC FORTRAN PASCAL COBOL
Mainframe Computer	500000 DM bis 10000000 DM	Printer Plotter Display	Disk Magnetband	keine Einschränkung

Aus der obigen Tabelle ist zu entnehmen, daß selbst für kleinere Büros die Möglichkeit besteht, für erschwingliches Geld funktionstüchtige - wenn auch relativ langsame - "Datenverarbeitungssysteme" betreiben zu können. Es ist zu erwarten, daß gerade im Mini- und Mikrocomputerbereich in den nächsten 5 bis 10 Jahren erhebliche Anstrengungen unternommen werden, um die HARDWARE noch leistungsfähiger, d.h. noch schneller zu machen und mit mehr Kapazität und Komfort auszustatten.

Wie stark sich der Preisverfall der HARDWARE ausgewirkt hat, läßt sich am besten aus der folgenden Graphik ablesen, in der die relativen Kosten für die HARDWARE der letzten 25 Jahre aufgetragen sind (vgl. [3]).



In der Weise wie die Kosten für die "schlaue Kiste" Computer geschrumpft sind, ist jedoch das Bedürfnis gestiegen, den Computer zu nutzen - eben weil er so billig geworden ist! Selbst der Laie weiß jedoch, daß die HARDWARE allein nicht ausreicht, um EDV einzusetzen. Man braucht "Programme" also eine zweite "WARE-Form", die sogenannte SOFTWARE. Die Erstellung von SOFTWARE ist aber arbeits- und lohnintensiv - und demnach teuer. Um zu zeigen, wie sich die SOFTWARE-Kosten entwickelt haben, sind diese in die obige Graphik (gestrichelt) eingetragen worden (vgl. [3]).

Die Erstellung von SOFTWARE hat - wie hieraus zu ersehen ist - ein erhebliches wirtschaftliches Gewicht bekommen: betrug 1955 das Kostenverhältnis HARDWARE : SOFTWARE noch 100:1, so hat sich dieses Verhältnis heute genau ins Gegenteil verkehrt, nämlich 1:100! Verursacht durch den Preissturz der HARDWARE und ihre große Verbreitung, hat die Erstellung von SOFTWARE industrielle Außmaße angenommen. War früher die HARDWARE ein Engpaß der Daten-

verarbeitung, so jetzt zunehmend die SOFTWARE. Dieser Entwicklung muß mehr entgegengewirkt werden als bisher, will man die Kostenschere nicht noch weiter aufklaffen lassen. In den letzten Jahren hat sich deswegen folgerichtig so etwas wie eine neue Disziplin, das SOFTWARE ENGINEERING herausgebildet, mit dem man die Schwierigkeiten bei der Entwicklung, Wartung und Anwendung von SOFTWARE in den Griff bekommen möchte. Was hierunter im einzelnen zu verstehen ist und welche Folgerungen sich für die EDV im Bauwesen ergeben, soll in den folgenden Kapiteln abgehandelt werden.

3. SOFTWARE ENGINEERING - RETTUNG IN DER NOT

Jeder Bauingenieur, der eine Brücke oder ein Hochhaus bauen will, weiß genau, wie er sein "Ingenieurprodukt" zu erstellen hat. Er verwendet Methoden, die auf wissenschaftlich fundierten Grundlagen - wie der Elastizitäts- oder Plastizitätstheorie - und durch einschlägige Vorschriften abgesichert sind. Bei der Herstellung seines Produktes greift er außerdem noch auf Methoden des Managements und der Kostenkontrolle zurück, um möglichst wirtschaftliche Lösungen zu erzielen.

Ganz anders ist die Situation bei der Erstellung von SOFTWARE. Im Gegensatz zu den traditionellen technischen Disziplinen, wie etwa dem Bauingenieurwesen oder Maschinenbau - im anglo-amerikanischen mit CIVIL ENGINEERING bzw. MECHANICAL ENGINEERING bezeichnet - gibt es keine eingenständige Ingenieurdisziplin SOFTWARE ENGINEERING. Wenn demnach der Begriff SOFTWARE ENGINEERING verwendet wird, so ist dies eigentlich mehr als Provokation zu verstehen, da das "SOFTWARE ENGINEERING" zur Zeit noch längst nicht das Niveau der klassischen Ingenieurfächer erreicht hat. Der Begriff SOFTWARE ENGINEERING charakterisiert vielmehr eine veränderte Einstellung und Arbeitshaltung zur SOFTWARE - hervorgerufen durch die enormen Kostensteigerungen im SOFTWARE-Bereich.

Obwohl das SOFTWARE ENGINEERING erst in der Entwicklungsphase ist, soll dennoch kurz so etwas wie eine Definition gegeben werden. Wir wollen unter SOFTWARE ENGINEERING im folgenden analog zu [4] die Bereitstellung von Prinzipien, Methoden und Werkzeugen für die SOFTWARE-Entwicklung und -Wartung auf der Basis wissenschaftlicher Erkenntnisse und praktischer Erfahrungen verstehen. Ziel ist es dabei, die SOFTWARE-Qualität entscheidend zu verbessern und somit der unheilvollen Kostenexplosion im SOFTWARE-Bereich entgegenzuarbeiten.

Prinzipe sind allgemeingültige Grundsätze des Denkens oder Handelns im Sinne einer Norm. (Beispiele: Modularisierung, begleitende Dokumentation).

Methoden sind planmäßige Vorgehensweisen im Hinblick auf ein Ziel, meistens auf der Grundlage eines allgemeinen Prinzips. (Beispiele: Strukturierte Programmierung, Programmierstandards, Netzplantechnik und Terminplanung).

Werkzeuge sind Arbeitsmittel, die für die Erstellung von SOFTWARE verfügbar sind. (Beispiele: Programmiersprachen, Programm-

bibliotheken, Makroprozessoren oder Generatoren, Pre- und Postprocessing).

Ziel des SOFTWARE ENGINEERING ist die bewußte Qualitätsverbesserung der SOFTWARE, wobei folgende Qualitätskriterien maßgebend sind:

- Effizienz
- Benutzerkomfort
- Zuverlässigkeit
- Portabilität
- Änderbarkeit

Es liegt auf der Hand, daß die einzelnen Qualitätskriterien nicht immer gleichzeitig durchsetzbar sind und somit Zielkonflikte auftreten. Beispielsweise ist es nicht möglich ein Programm zugleich hinsichtlich seiner Effizienz optimal zu machen und mit höchstem Benutzerkomfort auszustatten, da diese beiden Eigenschaften konträr zueinander sind. Es kommt auf den jeweiligen Einzelfall an, ob einem bestimmten Qualitätskriterium der Vorzug zu geben ist oder ein ausgewogenes Mittel verschiedener Qualitätsmerkmale sinnvoller ist. Allen Beteiligten, d.h. dem Hersteller von SOFTWARE und dem Anwender muß jedoch klar sein, daß zwischen Qualität, Kosten und Zeitaufwand zahlreiche Abhängigkeiten und Rückkopplungen bestehen.

Welche Konsequenzen ergeben sich aus dem SOFTWARE ENGINEERING für Ausbildung, Praxis und Forschung des Bauingenieurwesens? Im folgenden soll hierauf eine kurze Antwort gegeben werden.

4. FOLGERUNGEN FÜR AUSBILDUNG UND FORSCHUNG

Fragen wir zunächst einmal, inwieweit die heutige Ausbildung unserer Bauingenieure den veränderten Verhältnissen durch die Einwirkung der modernen EDV Rechnung trägt? Die Antwort lautet: Wir sind noch weit von einer qualifizierten EDV-Ausbildung entfernt. Im wesentlichen beschränkt man sich derzeit lediglich darauf, den Studenten eine Programmiersprache (etwa FORTRAN) beizubringen. Dies reicht aber angesichts des heute ablesbaren Trends, daß die HARDWARE immer billiger und die "SOFTWARE" ständig teurer wird, nicht mehr aus. Wer verantwortungsbewußt denkt, sollte deswegen folgende Forderungen an die Ausbildung der Bauingenieure stellen:

a) es muß für eine schon zu Beginn des Studiums einsetzende, fachspezifische und praxisorientierte EDV-Grundausbildung der Bauingenieurstudenten gesorgt werden.

b) die EDV-Grundausbildung muß das gleiche Gewicht erhalten wie auch andere Grundlagenfächer.

c) im Hauptstudium müssen die Grundlagen sinnvoll durch eine kooperative Zusammenarbeit mit den klassischen Bauingenieurfächern vertieft werden. Insbesondere müssen Kenntnisse des SOFTWARE ENGINEERING, das Arbeiten mit großen Programmsystemen, Probleme des CAD und der graphischen Datenverarbeitung vermittelt werden.

Für die Praxis müssen insbesondere die SOFTWARE-Probleme klei-

ner und mittlerer Anwender abgebaut werden. Da dem Anwender die Übersicht fehlt, brauchte man quasi ein "Testlabor" für Programme, das mit Mitteln des SOFTWARE ENGINEERING objektiv die angebotenen Programme auf ihre Qualität hin überprüft. Da ein solches Labor bei realistischer Einschätzung der Gegebenheiten Utopie ist, müssen von den Entwicklern wenigstens Mindestanforderungen erfüllt werden, die dem Anwender Möglichkeiten des Vergleichs an die Hand geben:

a) die Programme müssen einheitlich dokumentiert und strukturiert programmiert werden,

b) es muß eine genaue Angabe darüber erfolgen, wie Änderungen vorgenommen werden können.

Für die computergestützte Forschung sind allgemein gültige EDV-Richtlinien auszuarbeiten, die - wie andere Normen - anerkannt und eingehalten werden. Wichtigste Forderungen sind:

a) die Verpflichtung, eine ausreichende Portabilität der SOFTWARE zu gewährleisten, um den Aufwand für die Erstellung von neuer SOFTWARE zu reduzieren.

b) die Erstellung einer einheitlichen Dokumentation für strukturiert programmierte SOFTWARE,

c) die Einrichtung einer koordinierenden Stelle, die bereits existierende SOFTWARE hinsichtlich ihrer Qualität zu überprüfen und gegebenenfalls Verbesserungen vorzunehmen hat.

5. ABSCHLIESSENDE BEMERKUNG

Der Verfasser dieses Aufsatzes ist der Meinung, daß vor allem durch eine verbesserte fachspezifische Ausbildung und durch den gezielten Einsatz der Prinzipe, Methoden und Werkzeuge des SOFTWARE ENGINEERING in Praxis und Forschung in wenigen Jahren jeglicher Verdacht, das Bauwesen sei ein "Entwicklungsgebiet" der EDV, ad absurdum geführt wird.

REFERENZEN

[1] COMPUTERWOCHE: Deutsches Bauwesen ist noch DV-Entwicklungsgebiet, 5.10.1979, S. 32.

[2] SYSTEMS 77: Computersysteme und ihre Anwendung, Branchen-Seminar Bauwesen, München, 1977, S. 13.

[3] WILSON, E.L.: Role of small computer systems in structural engineering, Electronic Computation 7th Conference, American Society of Civil Engineers, New York, 1979.

[4] GEWALD, K.; HAAKE, G.; PFADLER, W.: Software Engineering, Reihe Datenverarbeitung, R. Oldenbourg Verlag, München, Wien, 1979.