

Software Engineering

Herausgeber: Univ.-Prof. Dr. Horst Wildemann

Copyright by TCW Transfer-Centrum GmbH 2000
TCW-report Nr. 24

Die Deutsche Bibliothek CIP-Einheitsaufnahme

Broy, Manfred; Ehler, Herbert; Paech, Barbara; Rumpe, Bernhard; Thurner, Veronika
Software Engineering: Schlüssel zu Prozeßbeherrschung und Informationsmanagement
München, Transfer-Centrum GmbH
ISBN 3-934155-52-9

Verlag: TCW Transfer-Centrum GmbH - Leopoldstr. 146, 80804 München - Tel: 089/360-523-11, Fax: 089/361 023 20
eMail: mail@tcw.de, *Internet:* http://www.tcw.de

Alle Rechte, auch die der Übersetzung in fremde Sprachen, vorbehalten. Kein Teil dieses Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form, auch nicht zum Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet werden.

Software Engineering

Software Engineering

Was der TCW-report „Software Engineering“ leistet

Warum ist die Beherrschung des Software-Engineerings so entscheidend?

Die Antwort ist: Die Qualität, Innovationskraft und Wettbewerbsfähigkeit der Unternehmen und ihrer Geschäftsprozesse wird immer stärker durch das Maß der Integration der Softwareunterstützung bestimmt. Software-Engineering ermöglicht die Verschmelzung der Geschäftsprozesse mit der Informationsverarbeitung und damit die Steigerung der Effizienz und Qualität der Geschäftsabläufe durch Informationstechnik. Dabei geht Software-Engineering mit den entsprechenden Anwendungsgebieten eine nahezu unauflösliche Verbindung ein. Umgekehrt legt die Methodik des Software-Engineerings weitgehend die Form der Geschäftsprozesse fest und erlaubt die Konzeption neuer Dienstleistungen, Produkte, Produktions- und Organisationsformen. Wohl noch nie zuvor in der Geschichte der modernen Technik bestand eine so enge Verbindung zwischen Fragen der Organisation, des Managements und des Marketings auf der einen Seite und der Beherrschung technischer Fragestellungen andererseits. Software-Engineering wird zur entscheidenden „Enabling Technology“, ihre Beherrschung zur Überlebensfrage für alle Unternehmen.

Die Autoren:



Univ.-Prof. Dr. Manfred Broy

Technische Universität München



Dr. Herbert Ehler

Technische Universität München



Dr. Barbara Paech

Technische Universität München



Dr. Bernhard Rumpe

Technische Universität München



Dipl.-Inform. Veronica Thurner

Technische Universität München

Was ist Software eigentlich?

Software, wörtlich übersetzt “weiche Ware”, bezeichnet die Gesamtheit der Computerprogramme und Daten, die zur Erfüllung einer bestimmten Aufgabe, wir sprechen von einer “Anwendung”, eingesetzt werden. Drastisch ausgedrückt heißt das, daß Software nichts anderes ist als formalisierte Handlungsanweisungen für informationsverarbeitende Geräte, wie etwa Computer.

Damit erfüllt Software einen alten Traum der Menschheit. Ideen und Vorstellungen lassen sich in der Gestalt formalisierter Handlungsanweisungen direkt in automatisierte Aktion umsetzen. Am Anfang ist das Wort und das Wort ist bereits die Tat. Dies ist der Kern des Erfolges der modernen Softwaretechnik.

Inhaltsverzeichnis

Worin liegt die Bedeutung der Softwareentwicklung?	1
Warum ist Software ein Wettbewerbsfaktor?	1
Software – Kostentreiber oder Enabling Technology?	3
Softwareentwicklung in der Krise?	3
Warum ist Softwareentwicklung so teuer und schwierig?	4
Software Engineering – Schlüssel zum Erfolg?	6
Auf welchen Kerntätigkeiten basiert die Softwaretechnik?	11
Machbarkeitsstudie – Ist die Entwicklungsaufgabe unter ihren Rahmenbedingungen zu bewältigen?	13
Anforderungsanalyse – Was soll das Softwaresystem leisten?	13
Systemspezifikation – Was genau soll das Softwaresystem leisten?	17
Systementwurf – Wie kann das System strukturiert werden?	22
Implementierungsplattform – Welche Technologie soll eingesetzt werden?	24
Anwendungsarchitektur – Welche Bestandteile soll das System haben?	25
Spezifikation der Systemkomponenten – Was müssen die einzelnen Systembestandteile leisten?	27
Entwurf der Systemkomponenten – Wie können die Systembestandteile strukturiert werden?	28
Implementierung – Wie werden die einzelnen Produktteile erstellt?	30
Test und Validierung – Erfüllt das System die festgelegten Anforderungen?	33
Systemabnahme – Entspricht das System den Vorstellungen der Auftraggeber?	36
Integration – Wie werden System und Systemumgebung zusammengeführt?	38
Evolution – Wie wird Software im Einsatz weiterentwickelt?	40

Läßt sich Softwareentwicklung systematisch organisieren?	43
Softwareentwicklungsprozeß – Welche Tätigkeiten werden wann und mit welchem Ergebnis durchgeführt?	43
Wasserfallmodell – Sind sequentielle Entwicklungsphasen adäquat?	44
V-Prozeßmodell – Wo liegt der Ursprung einer erkannten Reihe von Folgefehlern?	46
Spiralmodell – Wie wird ein komplexer Entwicklungsprozeß in einfachere Schritte gegliedert?	47
Inkrementelles Vorgehen – Läßt sich ein Softwaresystem sukzessive in Einzelteilen entwickeln?	49
Evolutionäres Vorgehen – Wie können Änderungen in den Anforderungen schnell berücksichtigt werden?	50
Prozeßmodelle mit Prototypen – Wie geht man mit Unsicherheiten bei den Anforderungen um?	52
Wann ist welches Prozeßmodell geeignet?	54
Projektplanung und -kontrolle – Was ist im Projekt zu tun und was wurde schon erreicht?	55
Projektplanung – Wer macht was wann und womit?	56
Strukturierung eines Projektes – Wie zerlegt man komplexe Aufgaben in einfache Teilaufgaben?	58
Schätzverfahren – Wieviel Aufwand macht das Projekt, was kostet es und wann ist es fertig?	60
Zeitplanung – Was passiert wann?	65
Ressourceneinsatzplanung – Welche Personen und Betriebsmittel sind wann am Projekt beteiligt?	66
Risikoanalyse – Was kann schief gehen und wie kann man damit umgehen?	68
Qualität - Was kann man dafür tun?	68
Konfigurationsmanagement – Wie behält man den Überblick über eine Fülle von Teilergebnissen und Änderungen?	70
Projektverfolgung – Wie weit ist das Projekt?	73
Teammanagement – Wie wird aus einer Gruppe von Individualisten ein leistungsfähiges Team?	74
Aufgabenverteilung – Welche Person ist für welche Funktion im Projektteam geeignet?	77
Teamstruktur – Wer gehört zum Team und in welcher Rolle?	79
Teamleitung – Wer übernimmt die fachliche Führung und die soziale Vermittlung?	81
Motivation – Was weckt die Einsatzbereitschaft der Projektbeteiligten?	82
Kommunikation und Akzeptanz – Verstehen sich Anwender und Entwickler?	82

Software Engineering

Softwarequalität – Stellt das Softwareprodukt den Kunden zufrieden?	85
Qualitätsmerkmale – Wann ist Software gut?	85
Qualitätssicherung – Wie wird gute Software erstellt?	86
Formale Methoden – Kann Software präzise und fehlerfrei entwickelt werden?	88
Entwicklungsmethoden und CASE-Werkzeuge – Wie können Computer bei der Softwareentwicklung helfen?	92
Qualitätssteigerung durch Wiederverwendung – Macht mehrfache Nutzung Software besser?	96
Projektübergreifende Softwareentwicklung – Wie können Projekte voneinander lernen?	97
Welche Maßnahmen nutzen mehr als einem Projekt?	97
Wie ermöglicht man Wiederverwendung von Softwarekomponenten?	98
Wie verbessert man den Softwareentwicklungsprozeß?	103
Was leistet ISO 9000 für die Softwareentwicklung?	104
Capability Maturity Model – Welchen Reifegrad hat der Entwicklungsprozeß?	106
Welche Rolle spielt die Unternehmenskultur?	109
TQM – Was sind die großen Unternehmensziele?	109
Personalentwicklung und Ausbildung – Wie aktuell ist das Wissen im Unternehmen?	113
Multiprojektmanagement – Wie profitieren Projekte gegenseitig voneinander?	115
Experience Factory – Wie profitiert man von Erfahrungen?	117
Wie setzt man Änderungen im Unternehmen wirkungsvoll um?	118

Software Engineering

Was sind die aktuellen Trends im Software Engineering?	121
Wie stark werden neue Technologien adaptiert?	121
Was sind die aktuellen Trends beim Einsatz von Software?	122
Was steigert die Wertschöpfung: Hardware oder Software?	122
Simulation – Was bringt die virtuelle Realität?	123
Internet und Intranet – Was bringen Kommunikationsnetze?	124
Workflow – Lassen sich Brüche in Geschäftsprozessen vermeiden?	125
Was sind die aktuellen Trends in der Softwaretechnik?	126
Objektorientierung – Ein Paradigma für die Zukunft?	126
Componentware – Software aus dem Baukasten?	127
Standardsoftware versus Individualsoftware – Was ist die richtige Entscheidung?	131
Welche Rolle spielt die Softwaretechnik im Unternehmen der Zukunft?	133
Literaturverzeichnis	135
TCW-report - Themen 1999/2000	139
Impressum	141

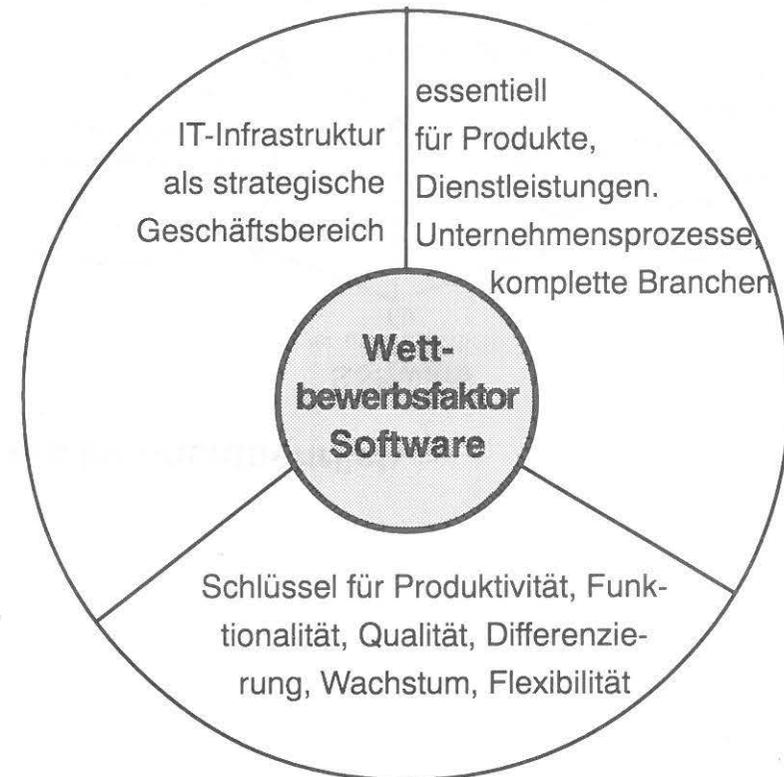
Worin liegt die Bedeutung der Softwareentwicklung?

In Produkten, Dienstleistungen und firmeninternen Arbeitsgängen erhalten Softwaresysteme immer stärker strategische Bedeutung im Hinblick auf Effizienz, Qualität, Produktivität und Flexibilität. Dadurch wird Software immer mehr zu einem kritischen Erfolgsfaktor jedes Unternehmens.

Warum ist Software ein Wettbewerbsfaktor?

Der rasante Fortschritt im Bereich der Computertechnik eröffnet immer weitere Möglichkeiten des Softwareeinsatzes in wissenschaftlichen, technischen und kommerziellen Anwendungsgebieten. Heute ist Software ein integraler Bestandteil fast aller hochwertigen technischen Produkte und Dienstleistungen. Ganze Branchen, wie Banken und Versicherungen, aber auch der gesamte administrative Bereich und nicht zuletzt das Gesundheitswesen sind von modernen Softwaresystemen zur Unterstützung von Geschäftsabläufen vital abhängig. Die Effektivität moderner Unternehmen und Organisationen wird maßgeblich durch den Einsatz effizienter betrieblicher Informationssysteme und Workflow Management Systeme bestimmt. Moderne Informations- und Kommunikationstechnik eröffnen neue Vertriebswege, die zu einer zunehmenden Internationalisierung der Märkte führen. Auch im Fertigungsbereich ist der

Der Wettbewerbsfaktor Software ...

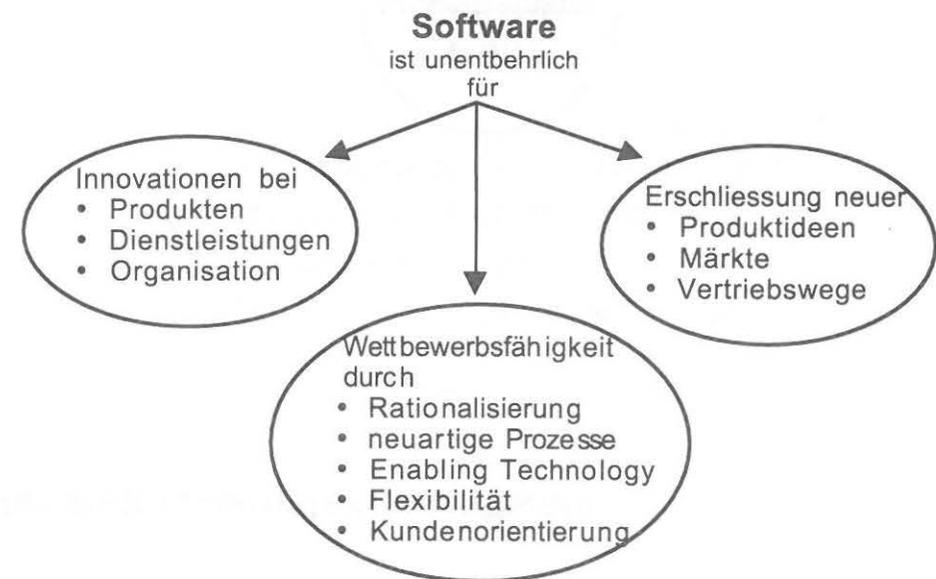


➔ ... läßt sich in drei Hauptwirkungsbereiche unterteilen.

Fortschritt der Automatisierung nicht mehr aufzuhalten. Produktivität und Durchlaufzeiten werden hier maßgeblich von der Leistungsfähigkeit und Zuverlässigkeit der zur Produktionssteuerung eingesetzten Software beeinflusst. Selbst der Leistungsumfang der Erzeugnisse wird nicht zuletzt von der darin integrierten Software bestimmt. Die enormen Innovationen bei fast allen hochwertigen technischen Produkten, wie beispielsweise bei Haushaltsgeräten, im Bereich Unterhaltungselektronik, aber auch im Automobilbereich, wären ohne den Einsatz eingebetteter Softwaresysteme nicht denkbar. Die Entwicklung der Software für diese Produkte stellt einen stetig wachsenden Anteil der Gesamtkosten dar.

Das breite Spektrum für den Einsatz von Computertechnik und dazugehörigen Softwaresystemen hat dazu geführt, daß Software in immer größerem Maß zu einem eigenständigen Wirtschaftsgut geworden ist. Bei den Gesamtkosten von Computersystemen bzw. Anwendungssystemen ist der relative Wertanteil der Software in den letzten Jahrzehnten ständig gestiegen. So ist die Informationstechnik- und Telekommunikationsbranche (IT&T) ein aufstrebender Wirtschaftszweig und stetig wachsender Dienstleistungssektor geworden.

Software ist unentbehrlich ...



⇒ ... als Innovationsmotor.

Durch die zunehmende Integration von Softwareprodukten in Form eingebetteter Systeme stellt in vielen Unternehmen die Software und deren Entwicklung eine strategische Aufgabe dar, die durch eigene oder externe Entwicklungsabteilungen bewältigt werden muß. Während früher größere Unternehmen ihre Software im wesentlichen selbst entwickelten, ist dies heute eher die Ausnahme. Zunehmend wird Software nicht mehr im eigenen Haus entwickelt, sondern bei externen Softwarehäusern in Auftrag gegeben. Dies gilt immer häufiger auch für die Bereitstellung und den Betrieb der gesamten DV-Infrastruktur. Die Vielfalt der in modernen Unternehmen eingesetzten Software und der Aufwand für ihre Entwicklung zwingt immer mehr Unternehmen, sich bei den Eigenentwicklungen auf Kernkompetenzen zu beschränken. Durch diesen Outsourcing-Trend hängt die Existenz einer wachsenden Zahl von Unternehmen von der erfolgreichen Durchführung extern oder intern vergebener Softwareentwicklungsprojekte ab.

Software – Kostentreiber oder Enabling Technology?

Die Kosten der Unternehmen für Informatiklösungen sind horrend. Wen wundert es, daß die Informatikabteilungen nicht selten dem Finanzvorstand zugeordnet sind und typischerweise als Kostenfaktor und -treiber verstanden werden. Übersehen wird dabei leicht, daß die wachsenden Potentiale der Softwarelösungen die Marktfähigkeit der Unternehmen so weitgehend bestimmen, daß eine Reduzierung der Softwarelösungen auf reine Infrastrukturfragen existentielle Risiken für die Unternehmen bergen kann.

Softwareentwicklung in der Krise?

Seit Software für Aufgaben immer höherer Komplexität eingesetzt wird, ist von der Softwarekrise die Rede: Große Projekte werden oft zu spät fertig, da die Entwicklungszeiten unterschätzt werden. Die kalkulierten Kosten und Termine werden zum Teil erheblich überschritten. Die fertige Software hat nicht die geforderte Qualität, sie nicht genügend performant, unzuverlässig, schwer wartbar und wird den Bedürfnissen der Anwender nicht ausreichend gerecht. In manchen Fällen scheitern ambitionierte SW-Projekte völlig. Während die Hardware beständig billiger wird, steigen die Softwarekosten rapide an.

Warum ist Softwareentwicklung so teuer und schwierig?

Die Ursachen für diese Probleme sind vielfältig. Sie liegen nicht allein im technischen Bereich, sondern resultieren auch aus mangelnder Kompetenz und daraus entstehenden Kommunikationsproblemen und Mißverständnissen zwischen Auftraggebern und Softwareentwicklern und sich verändernden Wünschen der Auftraggeber. Der Auftraggeber eines Softwareprojekts ist in der Regel ein Experte im Anwendungsbereich. Seine Sichtweise ist geprägt von den Zielen, die durch den Softwareeinsatz erreicht werden sollen. Häufig ist er jedoch nicht in der Lage, aktuelle und zukünftige Anforderungen an das Softwaresystem in ihren Konsequenzen abzuschätzen und die Anforderungen so präzise, eindeutig und vollständig zu formulieren, daß diese direkt in eine DV-technische Lösung umgesetzt werden können. Obwohl in den letzten Jahrzehnten enorme Fortschritte erzielt wurden, stellen unangemessene Anforderungen auch heute noch entscheidende Risikofaktoren in der Softwareentwicklung dar.

In der Sichtweise des Softwareentwicklers steht die DV-technische Realisierung des Systems im Vordergrund. Er verfügt meist nur über eingeschränktes Know-how aus dem Anwendungsbereich und versteht daher "die Sprache" und die "Essentials" des Auftraggebers nicht ohne weiteres richtig. Hinzu kommt nicht selten eine un-

Die Ursachen für die Probleme in der Softwareentwicklung sind:

