

# Einsatz von Software-Engineering-Techniken und CASE-Tools bei der Erstellung von CIM-Systemen in Europa

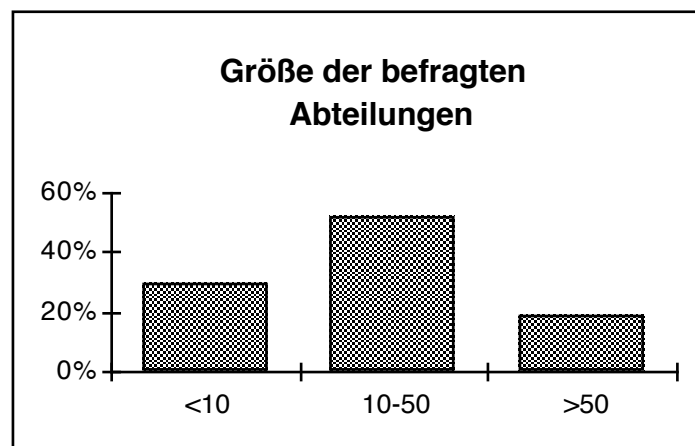
T. Lüth<sup>1</sup>, R. Mackay<sup>2</sup>

<sup>1</sup>Universität Karlsruhe, Postfach 6980, D-W7500 Karlsruhe, Deutschland

<sup>2</sup>C.E.C. Directorate-General XIII, Rue de la Loi 200, B-1049 Brussels, Belgium

*Im Bereich der computerintegrierten Fertigung CIM werden seit den letzten Jahren in steigendem Maß große und sehr komplexe Programmpakete entworfen. Derartige Entwicklungsvorhaben erfordern eine hohe Sorgfalt bei dem Entwurf und der Implementierung der Programmsysteme. In der ersten Jahreshälfte wurde nun eine europaweite Befragung von Firmen und Forschungseinrichtungen, die sich mit der Entwicklung von Software-Werkzeugen für CIM im Rahmen der ESPRIT-CIM Projekte beschäftigen, durchgeführt. Sie sollte klären, in wie weit Methoden des Software-Engineering bei der Programmentwicklung bereits verwendet werden und wie die bekannten Methoden von den Anwendern als praxisrelevant sowie die auf dem Markt erhältlichen CASE-Tools im täglichen Einsatz beurteilt werden.*

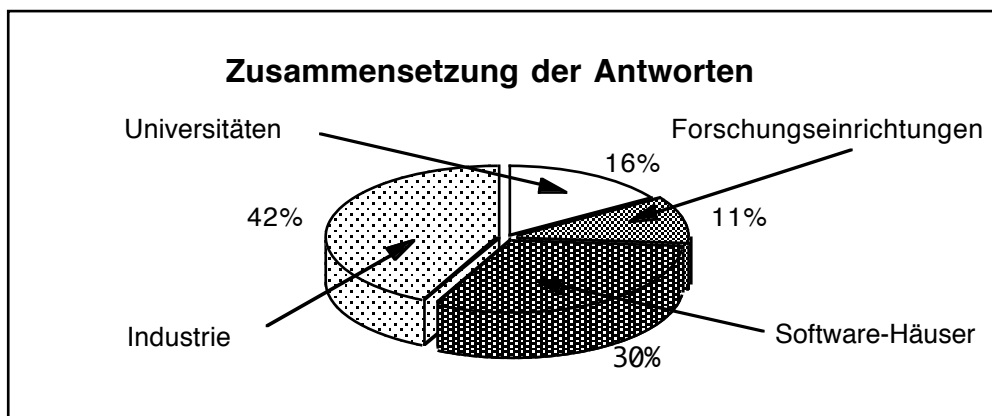
Die Entwicklung von Programmsystemen für die *computerintegrierte Fertigung* CIM erfordert wegen des großen Umfangs, den diese CAD/CAM-Werkzeuge erreichen, in besonderem Maß die Anwendung von Prinzipien, Methoden und Techniken auf den Entwurf und die Implementierung, d.h. den Einsatz von Methoden des Software-Engineering (SE). Für die einzelnen Abschnitte im Phasenmodell (software life cycle), das die verschiedenen Stadien eines Softwareproduktes von seiner Entstehung bis zum Ende seiner Verwendung enthält, sind seit einigen Jahren kommerzielle Produkte erhältlich, die den Entwicklungsprozeß unterstützen sollen.



**Bild 1** Mit leichter Streuung haben die Firmen/Abteilungen, mit denen die Befragung durchgeführt wurde eine Größe von zehn bis fünfzig Personen.

Eine europaweite Umfrage sollte klären, in wie weit die Methoden und CASE-Tools von den Produzenten großer Softwarepakete eingesetzt werden. Die Befragung wurde unter den Forschungseinrichtungen und Firmen durchgeführt, die sich an ESPRIT-Forschungsvorhaben (ESPRIT - The European strategic programme for research and development in information technology) der EG im Bereich der computerintegrierten Fertigung C.I.M. bereits beteiligt haben. Diese Firmen verfügen in der Regel über große Erfahrung bei der Entwicklung von Software in diesem Bereich. Ihre Antworten stehen sicherlich stellvertretend für fast alle Entwickler großer Softwarepakete. Die Auswertung stützt sich auf die Rückantworten von ca. 30% der ESPRIT-CIM Teilnehmer, die bis Ende Juni '92 eintrafen. Bei den größeren Firmen wurde der Fragebogen von der Abteilung, die für das entsprechende Projekt zuständig ist, ausgefüllt.

Zur genaueren Analyse der Ergebnisse wurde eine Aufteilung der eingegangenen Fragebögen in die vier Gruppen *Universitäten (UNI)*, *Forschungseinrichtungen (FE)*, *Software-Häuser (SH)* und andere *Firmen für Automatisierungstechnik (IND)* vorgenommen. Die Unterteilung ist notwendig, da die vier Gruppen bei ihrer Software-Entwicklung nicht in gleicher Weise alle Stadien im Phasenmodell des Software-Engineering unterstützen. Bei einer Interpretation der angegebenen Durchschnittswerte ist es notwendig, das Zahlenverhältnis der Gruppenstärke zu berücksichtigen. Von den 90 Fragebögen entfielen 16% auf die Universitäten und 11% auf andere Forschungseinrichtungen. Die Software-Häuser waren mit 30% beteiligt und den größten Anteil mit 42% hatten die restlichen Firmen für Automatisierungstechniken.

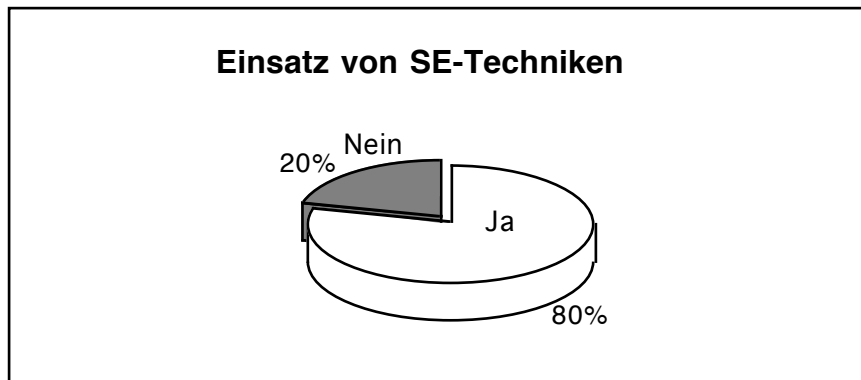


**Bild 2** Zusammensetzung der Untersuchung durch die einzelnen Gruppen: Universitäten, Forschungseinrichtungen, Software-Häuser und andere Firmen. Es wurden 90 Fragebögen ausgewertet.

## Wann und wofür werden SE-Methoden verwendet?

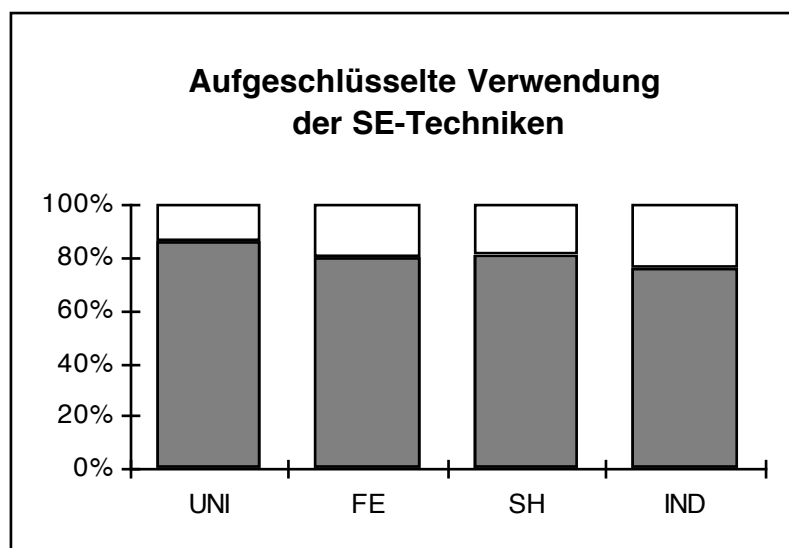
Die wichtigste Frage zielt auf den bisherigen Einsatz von Software-Engineering-Techniken oder Methoden bei der Software-Entwicklung. Diese Frage wurde von 80% der Befragten positiv beantwortet. Interessant ist hierbei, daß es keine großen Unterschiede zwischen den universitären Einrichtungen und der Industrie gab. Offensichtlich ist die Notwendigkeit, Software-Engineering-Techniken konsequent bei der Erstellung von Programmen einzusetzen, inzwischen weitestgehend erkannt. Warum dennoch bisher 20% der Befragten auf den

Einsatz der SE-Methoden verzichtet wird aus den entsprechenden Antworten weiter unten deutlich.



**Bild 3** Der Einsatz von Software-Engineering-Techniken liegt bei allen Teilnehmern der Umfrage bei ca. 80%.

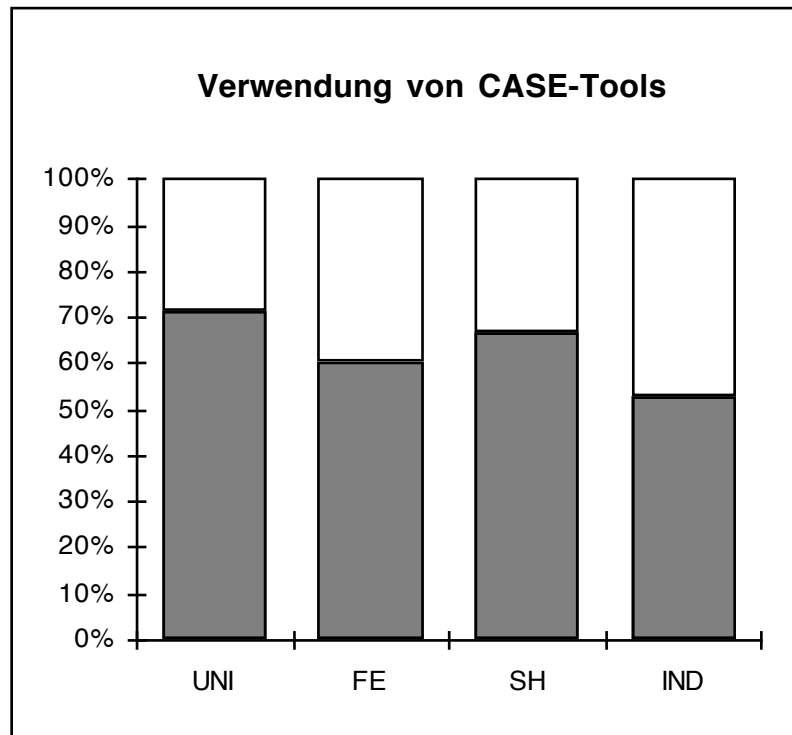
Die positive Beantwortung der Frage nach dem Einsatz der SE-Methoden bedeutet nicht, daß die Techniken durchgehend im Software-Entwicklungsprozeß eingesetzt werden. Die in der Umfrage gegebenen Antworten ließen erkennen, daß hauptsächlich in der Analysephase die SE-Methoden zum Einsatz kommen.



**Bild 4** Der Einsatz von SE-Techniken liegt bei Universitäten (UNI), Forschungseinrichtungen (FE), Softwarehäuser (SH) und Firmen (IND) gleichmäßig zwischen 75% und 85%.

Während bei der Frage nach der Verwendung von SE-Techniken ein einheitliches Bild aller vier Gruppen entsteht, zeigen sich im Gegensatz dazu bei der Frage nach der Verwendung von CASE-Tools für den rechnerunterstützten Einsatz der SE-Techniken, deutliche Unterschiede zwischen den einzelnen Gruppen. Offensichtlich ist auch, daß die bisher vorhandenen CASE-Tools insgesamt noch nicht den Verbreitungsgrad wie die SE-Techniken besitzen; so verwenden beispielsweise in der Industrie nur etwa die Hälfte aller Befragten

CASE-Tools. Auch hier gilt wieder, daß die Werkzeuge nur in einzelnen Phasen der Entwicklung eingesetzt werden können.



**Bild 5** CASE-Tools werden von den vier betrachteten Gruppen Universitäten (UNI), Forschungseinrichtungen (FE), Softwarehäuser (SH) und Firmen (IND) in unterschiedlichem Rahmen zur Erstellung von Software verwendet.

Ein weiterer Punkt der Befragung richtete sich darauf, in welcher Phase der Erstellung von Software CASE-Tools von den Befragten verwendet werden. Es wurden dabei die Phasen 1. Analysephase, 2. Entwurfsphase und erster Prototyp, 3. Implementierungsphase sowie 4. Testphase und Qualitätskontrolle unterschieden. Eine fünfte Sparte wurde anderen Aufgaben wie der Erstellung von Dokumentationen, dem Produktmanagement und der Wartung zugeordnet.

Verständlich ist, daß die Universitäten keine CASE-Tools für die Testphase einsetzen, da in Universitäten hauptsächlich Prototypen entworfen werden. Interessant ist jedoch, daß in der Industrie, sofern CASE-Tools Verwendung finden, diese immerhin zu 70% in der Analysephase, zu 90% in der Entwurfsphase und nur zu 30% in der Implementierungsphase eingesetzt werden. Universitäten und Forschungseinrichtungen setzen ihre CASE-Tools in etwa zu je 65% und die Software-Firmen zu je 75% in den ersten drei Phasen ein. Nur etwa ein Fünftel aller Befragten setzt für die Testphase oder die Qualitätskontrolle CASE-Tools ein. Der gleiche Wert wurde auch für die besondere Sparte 5 erreicht.

## Welche SE-Methoden werden verwendet?

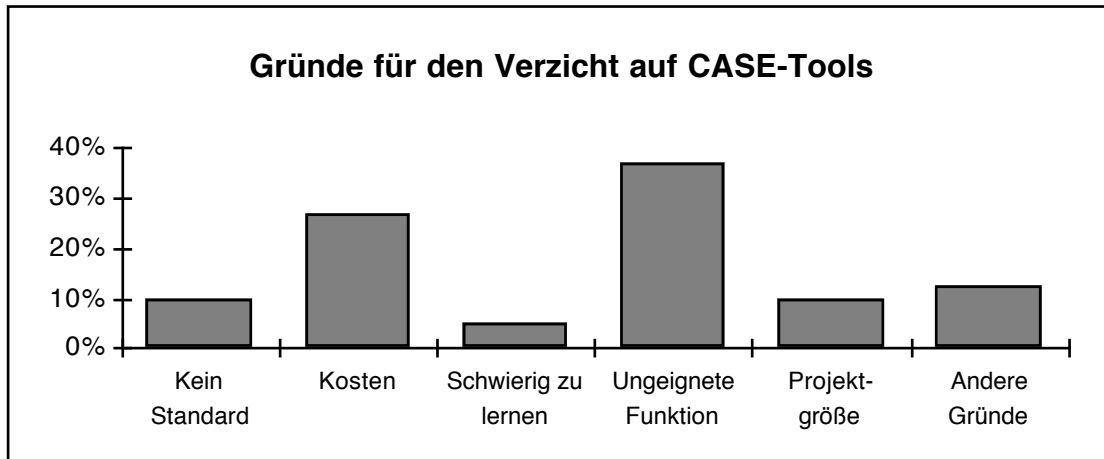
Unabhängig, ob manuelle SE-Techniken oder CASE-Tools zum Einsatz kommen, gibt es weitere Methoden, um innerhalb einer Abteilung, die an einem größeren Software-Projekt

arbeitet, eine einheitlichen Software-Entwicklungsstrategie zu verfolgen. Interessanterweise wurden hierbei sowohl von den Universitäten als auch von den anderen Forschungseinrichtungen in allererster Linie die Verwendung von einheitlichen Programmiersprachen und Programmbibliotheken genannt. Ganz anders sieht es bei den Software-Häusern aus, die gleichermaßen auf die Verwendung von hausinternen Programmierrichtlinien, Qualitätsplänen, Simulations- und Testläufen sowie auf weitere Verfahren setzen. Die an der Umfrage beteiligten Firmen setzen in erster Linie auf Qualitätspläne und mit einigem Abstand auf Kundenbewertungen und Programmcode-Kontrollen. Auf Grund der stark differierenden Antworten läßt sich hier kein einheitliches Bild für alle vier Gruppen finden.

Bei der Frage, welche Methoden oder Werkzeuge zum Einsatz kommen, wurde in allen vier Gruppen von mehr als der Hälfte die Methoden SADT/IDEF genannt. Daneben sind zu über 75% bei den Software-Häusern eigene Verfahren Bestandteil der Softwareentwicklung. Eigenentwicklungen machen bei den anderen drei Gruppen jedoch nur etwa 10% aus. Kritisch beurteilen muß man hierbei den Begriff der Eigenentwicklung einer SE-Methode. Ein Verfahren, das von nur einer Firma entwickelt und benutzt wird, kann kaum als Methode bezeichnet werden. Neben SADT/IDEF wurden ebenfalls in etwa 50% der Antworten Data-Flow-Diagrams, Entity-Relationship-Diagrams, State-Diagrams, wissensbasierte Methoden oder objektorientierte Methoden genannt. 63% der Befragten verwenden darüberhinaus Methoden, die sich nicht in die vorher genannten Verfahren eingliedern lassen.

## **Was spricht für und was spricht gegen einen Einsatz?**

Die Frage nach dem Grund des Einsatzes von Software-Engineering-Techniken oder Tools wurde von den Forschungseinrichtungen und den Software-Häusern gleichermaßen beantwortet. An erster Stelle steht mit 65% der Wunsch nach einem hausinternen Entwicklungsstandard. Auf den Plätzen zwei und drei rangierend dann mit etwa 30% eine verbesserte Produktqualität und die Einhaltung von Kundenwünschen. Anders sieht es bei den Antworten der Industrie aus. Hier beantworteten 30% die Frage nach der Motivation des Tool-Einsatzes mit der geeigneten Funktionalität der Tools. Die Angabe von nur 30% sollte auf die Entwickler von CASE-Tools alarmierend wirken, da nur ein Drittel der größten Zielgruppe die Funktionalität der Werkzeuge als Grund für den Einsatz angaben. 20% der Abteilungen in der Industrie verwendeten Werkzeuge für einen hausinternen Entwicklungsstandard.

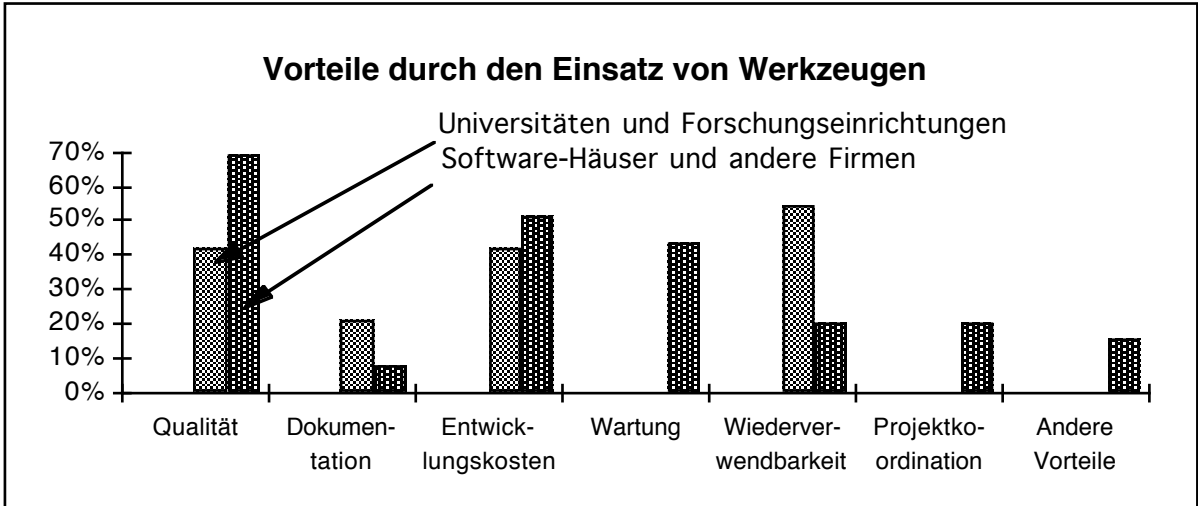


**Bild 6** Verteilung der genannten Gründe, die bei den Befragten gegen den Einsatz von CASE-Tools gesprochen haben.

Auf die Frage, warum in gewissen Phasen der Softwareentwicklung keine CASE-Tools verwendet werden, wurde von allen Gruppen eine ungeeignete Funktionalität (37%) der Tools genannt. An zweiter Stelle folgen die Anschaffungskosten und die Einarbeitungskosten (27%).

## Erzielte Vorteile und offensichtliche Nachteile

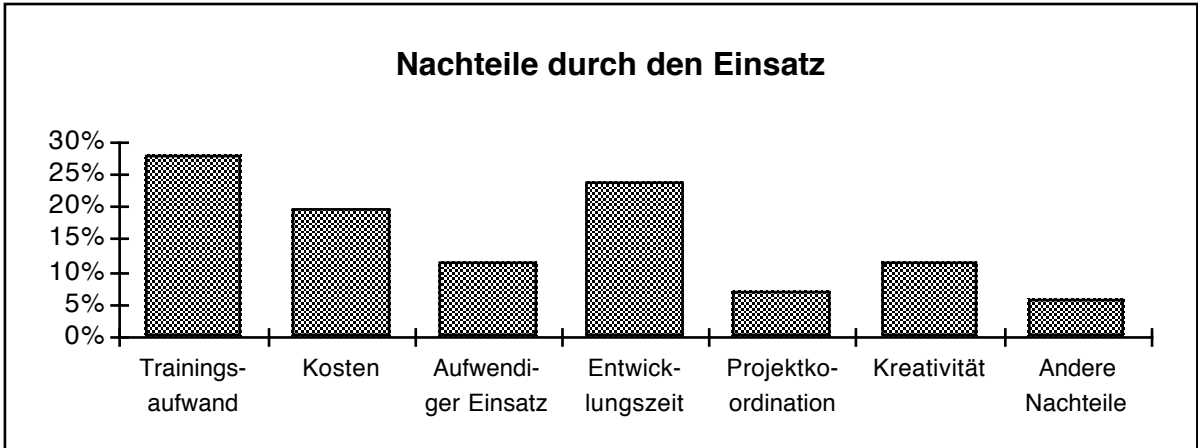
Obwohl die Frage nach den Vor- und Nachteilen des Einsatzes der vorhergehenden Fragestellung ähnelt, wurde diese Frage wesentlich konkreter und detaillierter beantwortet. Während für die kommerziellen Anwender die Punkte Qualitätssteigerung (69%), gesunkene Entwicklungskosten (51%) und eine leichtere Wartung der Software-Pakete (43%) die wichtigsten Vorteile sind, sehen die Programmierer in den nicht am Verkauf von Software interessierten Gruppen die Vorteile in einer hohen Modularität und Wiederverwendbarkeit der Quelltexte (54%). Eine bessere Qualität und gesunkene Entwicklungskosten wurden nur von etwa 40% der Befragten als offensichtlichen Nutzen des Einsatzes angesehen.



**Bild 7** Festgestellte Haupt-Vorteile, die sich aus dem Einsatz von Software-Engineering-Werkzeugen ergeben (Mehrfachnennungen sind möglich).

Neben dem direkten Nutzen durch den Einsatz von Software-Engineering-Techniken entstehen dem Benutzer selbstverständlich auch Nachteile. Bei der Angabe der Nachteile zeigte sich wieder ein geschlossenes Bild, obwohl die Resonanz auf die Frage deutlich geringer als bei den Vorteilen war. Aus der Sicht der meisten Anwender scheinen die Vorteile die Nachteile zu überwiegen.

Bei den festgestellten Nachteilen wurde der Trainingsaufwand von ca. 30% angegeben. Mit 25% der Antworten folgte die verlängerte Entwicklungszeit und nur etwa 20% gaben direkt gestiegene Entwicklungskosten als Nachteil an. Einige der Software-Häuser und die Forschungseinrichtungen sehen darüberhinaus die Kreativität der Programmierer durch die Verwendung von Tools eingeschränkt.



**Bild 8** Festgestellte Haupt-Nachteile die sich aus dem Einsatz von Software-Engineering-Werkzeugen ergeben (Mehrfachnennungen sind möglich).

## **Abschließende Bemerkungen**

Ziel der Untersuchung war es, festzustellen, in wie weit die Methoden des Software-Engineering bereits Einzug in den Alltag der Software-Erstellung gehalten haben. Es hat sich gezeigt, daß SE inzwischen relativ weit verbreitet ist, aber in vielen Bereichen ein Nachholbedarf besteht. Kommentare wie "...Software-Engineering ist nicht unsere Sache...", "...wir haben das bereits hinter uns, es brachte keinerlei Vorteile...", "...für die Programmierer ist das eine unangenehme Sache..." verdeutlichen auch noch einen weiteren, emotionalen Aspekt, der bei Entscheidungen für oder gegen des Einsatz von SE-Methoden oder -Werkzeugen eine wichtige Rolle spielt.

Weiterhin fiel auf, daß bei der Nennung von Produktnamen kein Produkt mehr als fünf mal - d.h., weniger als 6% der Befragten verwenden es - genannt wurde und in erster Linie die preiswerteren PC-basierten CASE-Tools am häufigsten zum Einsatz kommen. Auch die Gründe, warum auf den Einsatz von CASE-Tools verzichtet wird, zeigt den Bedarf nach billigeren und bedienerfreundlicheren CASE-Tools. Bisher müssen laut Befragung die Benutzer von CASE-Tools schon von vorne herein eine bessere Ausbildung besitzen. Es genügt nicht immer, die bisherigen Arbeitskräfte einzuarbeiten. Auch decken die bisher auf dem Markt erhältlichen Tools noch nicht alle Phasen der Software-Entwicklung geeignet ab.

Innerhalb der ESPRIT-Programme wird selbstverständlich auch die Entwicklung von Software-Engineering-Methoden und CASE-Tools unterstützt und vorangetrieben. Teilweise sind es dieselben Firmen, die sowohl CASE-Tools als auch Werkzeuge für die computerintegrierte Fertigung entwickeln. Die Übermittlung von Wissen und Erfahrung auf dem Gebiet des Software-Engineering erweist sich jedoch schon innerhalb einer Firma als überaus schwierig. Im Rahmen der in nächster Zukunft anlaufenden ESSI-Projekte (The European software and systems initiative) soll daher ein großflächiger Informationsaustausch durch die Unterstützung experimenteller Einsätze von Software-Werkzeugen gefördert werden.

Die Untersuchung steht in Zusammenhang mit der Gründung einer europäischen Interessengemeinschaft, die sich für einen verstärkten Einsatz von CASE-Tools bei der Entwicklung von CIM-Systemen einsetzt. Für diese Gemeinschaft werden immer noch Interessenten gesucht. Die Umfrage wurde durch die europäische Gemeinschaft initiiert und gefördert und durch das Institut für Prozeßrechentechnik und Robotik (Prof. Dr.-Ing. U. Rembold und Prof. Dr.-Ing. R. Dillmann) unterstützt. Zum Abschluß sei noch den Herren C. Le Pape (Fa. ILOG, Frankreich) und T. Vanmunster (Fa. LMS, Belgien) gedankt, ohne deren Hilfe und Fachkenntnis die Auswertung der Fragebögen nicht möglich gewesen wäre.