

PROF. DR. PAUL SCHMITZ

INFORMATIK-FACHBERICHT 91/1

Georg Herzwurm  
Dirk Berkau

Auswahl PC-gestützter Software-Entwicklungs-  
umgebungen - dargestellt am Beispiel von  
Excelerator, Information Engineering Workbench,  
ProKit\*WORKBENCH und Systems Engineer



LEHRSTUHL FÜR INFORMATIK  
DER UNIVERSITÄT ZU KÖLN

REGIONALES RECHENZENTRUM  
AN DER UNIVERSITÄT ZU KÖLN

PROF. DR. PAUL SCHMITZ

INFORMATIK-FACHBERICHT 91/1

Georg Herzwurm  
Dirk Berkau

Auswahl PC-gestützter Software-Entwicklungs-  
umgebungen - dargestellt am Beispiel von  
Excelerator, Information Engineering Workbench,  
ProKit\*WORKBENCH und Systems Engineer

Dieser Fachbericht basiert auf einer Diplomarbeit von Dirk Berkau, die am Lehrstuhl für Informatik angefertigt und im Dezember 1990 der Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität zu Köln vorgelegt worden ist.

Georg Herzwurm ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Informatik der Universität zu Köln.

© Alle Rechte vorbehalten

Albertus-Magnus-Platz, D - 5000 Köln 41

<u>Gliederung</u>	<u>Seite</u>
Verzeichnis der Abbildungen	4
Verzeichnis der Abkürzungen	5
1 Einleitung	6
1.1 Begriffsabgrenzung und Problemstellung	6
1.2 Zielsetzung und Aufbau der Arbeit	7
2 Grundlagen von Software-Entwicklungsumgebungen	9
2.1 Bedeutung	9
2.2 Architektur	12
2.3 Markt	13
3 Kriterien für die Auswahl von Software-Entwicklungsumgebungen	13
3.1 Generelle Kriterien für die Auswahl von Standardsoftware	14
3.2 Spezielle Kriterien für die Auswahl von Software- Entwicklungsumgebungen	20
3.2.1 Unterstützung der Phasen des Software-Lebenszyklus	20
3.2.1.1 Entwurfs- und Analysephase	20
3.2.1.2 Realisierungsphase	22
3.2.1.3 Erprobungs- und Konsolidierungsphase	23
3.2.1.4 Pflege- und Wartungsphase	24
3.2.2 Unterstützung phasenübergreifender Aufgaben	25
3.2.2.1 Informationsverwaltung	25
3.2.2.2 Dokumentation	26
3.2.2.3 Software-Qualitätssicherung	27
3.2.2.4 Prototyping	28
3.2.2.5 Projektmanagement	30
3.2.2.6 Konfigurationsmanagement	31
3.2.2.7 Allgemeine Bürodienste	31
3.2.3 Gestaltungsorientierte Auswahlkriterien	32
3.2.3.1 Methoden	32
3.2.3.2 Kapazität	38
3.2.3.3 Outputqualität	38
3.2.3.4 Integration	39
4 Beschreibung der ausgewählten Software-Entwicklungsumgebungen	40
4.1 Excelerator	41
4.2 Information Engineering Workbench	42

4.3	ProKit*WORKBENCH	43
4.4	Systems Engineer	43
5	Bewertung der ausgewählten Software-Entwicklungsumgebungen	44
5.1	Bewertung anhand genereller Kriterien für die Auswahl von Standardsoftware	44
5.2	Bewertung anhand spezieller Kriterien für die Auswahl von Software-Entwicklungsumgebungen	49
5.2.1	Unterstützung der Phasen des Software-Lebenszyklusses	49
5.2.1.1	Entwurfs- und Analysephase	49
5.2.1.2	Realisierungsphase	50
5.2.1.3	Erprobungs- und Konsolidierungsphase	51
5.2.1.4	Pflege- und Wartungsphase	51
5.2.2	Unterstützung phasenübergreifender Aufgaben	51
5.2.2.1	Informationsverwaltung	51
5.2.2.2	Dokumentation	55
5.2.2.3	Software-Qualitätssicherung	55
5.2.2.4	Prototyping	56
5.2.2.5	Projektmanagement	57
5.2.2.6	Konfigurationsmanagement	58
5.2.2.7	Allgemeine Bürodienste	58
5.2.3	Gestaltungsorientierte Auswahlkriterien	58
5.2.3.1	Methoden	58
5.2.3.2	Kapazität	63
5.2.3.3	Outputqualität	63
5.2.3.4	Integration	64
5.3	Zusammenfassende Bewertung der ausgewählten Software-Entwicklungsumgebungen	65
6	Neue Entwicklungen auf dem Markt für Software-Entwicklungsumgebungen	67
	Literaturverzeichnis	69
	Anhang I - Nutzwertanalyse der ausgewählten Software-Entwicklungsumgebungen	87
	Anhang II - Beispielausdrucke der ausgewählten Software-Entwicklungsumgebungen	103

Verzeichnis der Abbildungen

Seite

Abb. 1: Kriterien zur Auswahl von Software-Entwicklungsumgebungen	14
Abb. 2: Kurzbeschreibung Excelerator, Information Engineering Workbench, ProKit*WORKBENCH, Systems Engineer	41
Abb. 3: Informationsverwaltung Excelerator und ProKit*WORKBENCH	52
Abb. 4: Informationsverwaltung Information Engineering Workbench und Systems Engineer	54

Verzeichnis der Abkürzungen

Abb.	Abbildung
AD/Cycle	Application Development Cycle
ADW	Application Development Workbench
ANSI	American National Standards Institute
BOF	Benutzeroberfläche
BWL	Betriebswirtschaftslehre
CASE	Computer Aided Software Engineering
CGA	Color Graphics Array
COCOMO	Constructive cost Model
CSP/AD	Cross System Product / Application Development
CUA	Common User Access
DIN	Deutsches Institut für Normung e. V.
DFD	Datenflußdiagramm
EXC	Excelerator
HIPO	Hierarchy plus Input-Process-Output
ID	Identifikation
IEEE	The Institute of Electrical and Electronics Engineers
IEW	Information Engineering Workbench
Inf.-Verw.	Informationsverwaltung
KB	Kilo Byte
PC	Personal Computer
PKWB	ProKit*WORKBENCH
POSE	Picture Oriented Software Engineering
RBIS	Rechnergestütztes betriebliches Informationssystem
SA	Structured Analysis
SAA	System-Anwendungs-Architektur
SE	Systems Engineer
SEU	Software-Entwicklungsumgebung
SPU	Software-Produktionsumgebung
SSAD	Structured Systems Analysis and Design
TDM	Tausend Deutsche Mark
WYSIWYG	What you see is what you get
4GL	Fourth Generation Language

## 1 Einleitung

### 1.1 Begriffsabgrenzung und Problemstellung

Die Automatisierung der Softwareentwicklung wird unter dem Begriff Computer Aided Software Engineering (CASE) zusammengefaßt.<sup>1)</sup> Für Produkte zur Unterstützung dieser Automatisierung finden sich in der Literatur eine Reihe von Synonymen.<sup>2)</sup> Im Rahmen dieser Arbeit wird der Begriff Software-Entwicklungsumgebung (SEU) verwendet. Der Terminus Entwicklung soll verdeutlichen, daß kein Schwerpunkt auf einen bestimmten Teil der Softwareentwicklung gesetzt wird, wie es beim Gebrauch der Begriffe Entwurfs- oder Produktionsumgebungen der Fall sein könnte. Der Wortteil Umgebung soll andeuten, daß es sich nicht um ein isoliertes Werkzeug zur Unterstützung einer einzelnen Aufgabe der Softwareentwicklung handelt. Dieses wäre keine Umgebung, sondern lediglich ein Software Tool<sup>3)</sup>.<sup>4)</sup> Gleichwohl soll die Illusion genommen werden, SEU unterstützen grundsätzlich das gesamte Aufgabenspektrum der Softwareerstellung, wie zahlreiche Definitionen hoffen lassen.<sup>5)</sup> Da die zur Zeit angebotenen Produkte, insbesondere für den PC-Markt, nur selten dieser Anforderung gerecht werden,<sup>6)</sup> wird eine Software-Entwicklungsumgebung als eine Gruppe von integrierten Werkzeugen verstanden, die die Entwurfs- und Analysephase sowie mindestens eine weitere Phase des Software-Entwicklungszyklusses<sup>7)</sup> unter einer grafischen Benutzeroberfläche abdecken.<sup>8)</sup> Obwohl eine SEU physisch aus Hard- und Softwarekomponenten besteht,<sup>9)</sup> soll im Rahmen der Arbeit lediglich die Softwarekomponente untersucht werden,<sup>10)</sup> weil i. d. R. nur diese von den

---

1) Vgl. McClure /CASE Experience/ 235

2) Vgl. Balzert /Anforderungen/ 103; Seibt /Gegenwärtige Probleme/ o. S.

3) Vgl. IEEE/ANSI /IEEE Standard Glossary/ 32

4) Vgl. Fisher /CASE/ 6; McClure /CASE Experience/ 235-236

5) Vgl. Acly /Looking Beyond CASE/ 39; Österle /Computer Integrated Software-Engineering/ 14

6) Vgl. Löffler, Warner /Integration/ 31; Lehnhardt, Kimmel /Vollständige SPU/ 27

7) Diese Arbeit orientiert sich am Modell der Softwareentwicklungsphasen von Schmitz und Seibt. Vgl. Schmitz, Seibt /Einführung/ 8-10; Seibt /RBIS Skriptum/ 31-70

8) Vgl. Herzwurm /Möglichkeiten und Grenzen/ 1

9) Vgl. Kulling /Problemstellung und Perspektiven/ 1-06

10) Auf die Software- und Hardwarearchitektur von SEU wird in Kapitel 2.2 eingegangen.

CASE-Anbietern vertrieben wird.

Um in der Arbeit eine konsistente Bezeichnung der Personengruppen zu gewährleisten, wird der Anbieter des CASE-Produkts als Anbieter bezeichnet. Der Anwender bzw. Benutzer<sup>1)</sup> des CASE-Produkts und gleichzeitige Entwickler des mit der SEU zu erstellenden Systems wird ausschließlich Entwickler genannt und der Anwender bzw. Benutzer des zu erstellenden Systems als Anwender bzw. Benutzer bezeichnet.

## 1.2 Zielsetzung und Aufbau der Arbeit

Diese Arbeit basiert auf Erfahrungen, die bei der Auswahl einer Software-Entwicklungsumgebung für die Ausbildung der Wirtschaftsinformatikstudierenden an der Universität zu Köln von den Autoren im Januar 1991 gemacht wurden. Eine ideale SEU zur Entwicklung sämtlicher Softwaretypen und zur Unterstützung aller Aufgaben der Softwareentwicklung, außerdem lauffähig auf allen Rechnerklassen, ist am Markt nicht verfügbar.<sup>2)</sup> Der potentielle Käufer einer SEU muß deshalb das für seine individuellen Bedürfnisse und Zielvorstellungen geeignetste Produkt aus dem umfangreichen Marktangebot auswählen.<sup>3)</sup> Hierzu ist ein Selektionsprozeß notwendig, der objektiv, möglichst fehlerfrei und nachprüfbar ist.<sup>4)</sup> Im Rahmen des Auswahlprozesses an der Universität zu Köln wurde zunächst ein Grobkriterienkatalog mit K.O.-Kriterien entworfen, um somit eine Reduzierung der in Frage kommenden SEU auf eine überschaubare Anzahl zu erreichen. Da der Personal Computer (PC) aufgrund seiner wirtschaftlichen Vorteile, der kurzen Antwortzeiten, der Grafikmöglichkeiten und der daraus resultierenden Benutzerfreundlichkeit für die Ausbildung am geeignetsten erschien, wurde sich bei der Auswahl auf PC-gestützte SEU beschränkt. Die verbliebenen SEU wurden durch die Erstellung kleiner, praxisnaher Fallbeispiele und mit Hilfe eines Feinkriterienkatalogs verglichen. Dabei wurde versucht, einen sehr detaillierten Feinkriterienkatalog zu entwickeln, der möglichst sämtliche für den Entwickler relevanten Eigenschaften der SEU berücksichtigt,<sup>5)</sup> aber dennoch mit einem vertretbarer

- 
- 1) Der Anwender wird als die Organisationseinheit bezeichnet, in der mit der Software gearbeitet wird; der Benutzer wird als der Mensch, der unmittelbar mit der Software arbeitet, definiert. Vgl. DIN /DIN 66285/ 2
  - 2) Vgl. Fisher /CASE/ 225; McClure /CASE Experience/ 236
  - 3) Vgl. Fisher /CASE/ 172
  - 4) Vgl. Schmitt /Auswahl / 5-05; Schmitz, Schwichtenberg, Sodeur /Kombinationsverfahren/ 91
  - 5) Vgl. Ewald /Software/ 68; Schmitz, Schwichtenberg, Sodeur



Gesamtaufwand abzarbeiten ist<sup>1)</sup>). Aufgrund der strategischen Bedeutung einer Investition in eine SEU<sup>2)</sup> ist für die Feinauswahl, bei der nur noch wenige Produkte untersucht werden, ein höherer Beurteilungsaufwand vertretbar.<sup>3)</sup> Dies rechtfertigt den Umfang des Kriterienkatalogs, für dessen Ausfüllen einschließlich der Erstellung eines kleinen Fallbeispiels ein Zeitbedarf von 5 Personentagen je SEU kalkuliert werden muß. Die Einzelergebnisse ermöglichen in einer abschließenden Zusammenfassung<sup>4)</sup> einen Vergleich der SEU hinsichtlich der Eignung für einen bestimmten Entwickler bzw. für ein bestimmtes Entwicklungsziel - hier die Ausbildung von Studierenden. Die vorliegende Arbeit soll aber auch anderen öffentlichen oder privaten Unternehmungen eine Hilfestellung für die Feinauswahl einer SEU bieten. Hierzu ist lediglich eine Gewichtung der Kriterien entsprechend der individuellen Präferenzen notwendig; in Einzelfällen kann auch das Hinzufügen weiterer Kriterien erforderlich sein, da der vorliegende Kriterienkatalog zwar einen Großteil der relevanten Eigenschaften von SEU abdeckt, jedoch keinen Anspruch auf Vollständigkeit erhebt. Da es sich bei den Auszubildenden um Studierende an der Wirtschafts- und Sozialwissenschaftlichen Fakultät handelt, wird desweiteren ein Entwickler unterstellt, der sich auf kommerzielle Probleme spezialisiert hat. Die Anwendung des Kriterienkatalogs wird an vier in die Feinauswahl gelangten SEU demonstriert. Anschließend erfolgt eine Verdichtung der Ergebnisse im Rahmen einer Nutzwertanalyse. Der gesamte Kriterienkatalog umfaßt ca. 280 Einzelkriterien und ist über 150 Seiten lang. Aus diesen Gründen können lediglich die bedeutendsten Kriterien kommentiert werden. Ein Überblick über die Einzelergebnisse befindet sich im Anhang I der Arbeit.

Das auf diese Einleitung folgende zweite Kapitel beschreibt die Grundlagen von SEU. Diskutiert wird die Bedeutung der SEU für die Softwareentwicklung und die einer Investition für den Entwickler. Danach wird die Hard- und Softwarearchitektur von SEU erläutert und der Markt für PC-gestützte SEU vorgestellt. Im dritten Kapitel werden Kriterien für die Auswahl von SEU aufgestellt und ihre Bedeutung kommentiert. Aus Gründen der Übersichtlichkeit erfolgt eine Unterteilung in generelle Kriterien, die allgemein für Standardsoftware

---

/Kombinationsverfahren/ 33

- 1) Vgl. Frank /Standardsoftware/ 55; Schmitz, Schwichtenberg, Sodeur /Kombinationsverfahren/ 91
- 2) Die Gründe für den strategischen Charakter werden in Kapitel 2.1 dargelegt.
- 3) Vgl. Schmitt /Auswahl/ 5-09
- 4) Vgl. Schmitz, Schwichtenberg, Sodeur /Kombinationsverfahren/ 5, 9

gelten, und in spezielle Kriterien, die entweder nur für SEU relevant sind oder für diese eine besondere Bedeutung haben. Im einzelnen werden die Vollständigkeit der Unterstützung des Software-Lebenszyklusses und die Unterstützung der phasenübergreifenden Aufgaben untersucht. Der letzte Abschnitt dieses Kapitels geht auf die Gestaltung dieser Unterstützung ein. Die Unterteilung des Kapitels entspricht der Grobgliederung des Fragebogens. In Kapitel vier werden die ausgewählten SEU kurz vorgestellt. Im fünften Kapitel folgt eine Untersuchung der ausgewählten SEU anhand der vorher entwickelten Kriterien. In einer abschließenden Zusammenfassung werden die grundlegenden Unterschiede der untersuchten Produkte aufgezeigt. Das Schlußkapitel beinhaltet die Darlegung von Entwicklungstendenzen auf dem CASE-Markt. Insbesondere werden die Ankündigung des AD/Cycle-Konzepts der IBM und die Methodenunterstützung durch den Einsatz von Expertensystemen diskutiert.

## 2 Grundlagen von Software-Entwicklungsumgebungen

### 2.1 Bedeutung

Die Softwarebranche muß die Wirtschaftlichkeit des Softwareentwicklungsprozesses<sup>1)</sup> erhöhen und die Qualität<sup>2)</sup> ihrer Produkte verbessern, will sie die Softwarekrise<sup>3)</sup> sowie die steigende Komplexität der zu lösenden Aufgabenstellungen bewältigen.<sup>4)</sup> SEU sollen die Lösung dieser Probleme unterstützen; durch sie wird der Einsatz strukturierter Techniken<sup>5)</sup> erst praktikabel und wirtschaftlich vertretbar.<sup>6)</sup> Der Softwareentwickler wird von Routinearbeiten befreit,<sup>7)</sup> seine Arbeit systematisiert. Ließen die mit Papier und Bleistift ausgerüsteten Entwickler nichts unversucht, das erneute Zeichnen eines Diagramms

---

1) Vgl. Crozier u. a. /Critical analysis/ 486; Schmitz /Methoden, Verfahren und Werkzeuge/ 72-73

2) Vgl. Bröhl /SE-Umgebung/ 96; McClure /CASE Experience/ 236

3) Begrenzte Softwareerstellungskapazitäten bei steigender Nachfrage nach Anwendungssoftware, bedingt durch sinkende Hardwarekosten, führten in die Softwarekrise: Software wird später und teurer als erwartet fertiggestellt, sie ist nur schwer verständlich und leistungsschwach. Vgl. Sommerville /Software Engineering/ 2

4) Vgl. Löffler, Warner /Integration/ 30

5) Vgl. Martin, McClure /Structured Techniques/

6) Vgl. Orr u. a. /Methodology/ 233; Sommerville, Morisson /Software Development/ 2

7) Vgl. Schmitz /Komplexes anschaulich darstellen/ 33

zu vermeiden, so sind die mit einer SEU arbeitenden Entwickler eher zu Änderungen motiviert.<sup>1)</sup> Desweiteren kann die Einhaltung von Normen sowie von Unternehmungs- oder Projektstandards erzwungen werden<sup>2)</sup>. Die Bereitstellung von grafischen Kommunikationsmedien leistet einen Beitrag zur Überwindung von Sprachbarrieren<sup>3)</sup> zwischen dem Anwender bzw. Benutzer und dem Systementwickler und reduziert somit das Risiko, Software zu erstellen, die den Bedürfnissen des Anwenders oder des Benutzers nicht entspricht.<sup>4)</sup> Gleichzeitig wird die Kommunikation der Teammitglieder untereinander gefördert.<sup>5)</sup> SEU bieten die Möglichkeit, den Software-Erstellungsprozeß zu modularisieren.<sup>6)</sup> Die Verfügbarkeit einer maschinellen Überprüfung<sup>7)</sup> der Vollständigkeit, Eindeutigkeit und Konsistenz des Entwurfs ermöglicht das frühzeitige Entdecken und Beheben von Entwurfsfehlern<sup>8)</sup>. Weiterhin kann die Dokumentation und Implementierung weitgehend automatisiert vorgenommen werden.<sup>9)</sup> Der erhebliche Kostenfaktor Wartung wird reduziert.<sup>10)</sup> Änderungen und Wiederverwendungen von Entwürfen oder Softwaremodulen werden wirtschaftlich vertretbar.<sup>11)</sup> SEU lassen durch die Bereitstellung von Projektmanagementfunktionen<sup>12)</sup> eine Kalkulation von Ressourcen, Kosten und Terminen zu. Die Folgen eines Einsatzes von SEU bewirken neben der Automatisierung des Softwareerstellungsprozesses auch die Modifikation seiner Struktur: die stärkere Konzentration des Softwareentwicklers auf die analytischen Tätigkeiten<sup>13)</sup> und die Einführung des Rapid Prototyping<sup>14)</sup> sind in diesem Zusammenhang zu nennen. Zusammenfassend bleibt festzuhalten: der Entwickler soll durch CASE nicht ersetzt werden, sondern die Effizienz

- 
- 1) Vgl. Martin, McClure /Structured Techniques/ 315
  - 2) Vgl. Hering /Software-Engineering/ 3, 5; Orr u. a. /Methodology/ 233
  - 3) Vgl. Scheer /EDV-orientierte BWL/ 137-138
  - 4) Vgl. Fisher /CASE/ 50; Maag /Methoden/ 28
  - 5) Vgl. McClure /Automation/ 37; Sullivan-Trainor /Buyers' Scorecard/ 69
  - 6) Vgl. Fisher /CASE/ 6, 231; Schulz /Software-Entwurf/ 31
  - 7) Vgl. McClure /Automation/ 6, 11, 37, 189
  - 8) Vgl. Chikofsky, Rubenstein /Reliability Engineering/ 15; Wirtz /Softwareentwurf/ 326
  - 9) Vgl. Fisher /CASE/ 213; Martin, McClure /Structured Techniques/ 120
  - 10) Vgl. Scheer /EDV-orientierte BWL/ 126
  - 11) Vgl. McClure /CASE-Experience/ 236; Schmitt /Auswahl/ 5-03
  - 12) Vgl. Balzert /Entwicklung/ 488; McClure /Automation/ 61
  - 13) Vgl. Leppert, Stork /Entwicklungsumgebungen/ 229; Fisher /CASE/ 230
  - 14) Vgl. McClure /CASE Experience/ 242

seiner Arbeit soll gesteigert werden.<sup>1)</sup> Die Investition in eine SEU bedeutet für die Software entwickelnde Unternehmung eine strategische Entscheidung: durch die aufwands- und zeitgerechtere Projektabwicklung<sup>2)</sup> wird die Wettbewerbsfähigkeit der Unternehmung positiv beeinflusst.<sup>3)</sup> Desweiteren wird die Abhängigkeit vom einzelnen Programmierer aufgrund der erhöhten Lesbarkeit der erstellten Software reduziert.<sup>4)</sup>

Die Einführung einer SEU ist jedoch für die Unternehmung mit hohen Investitionskosten verbunden: neben den Anschaffungskosten entstehen Kosten für die Anpassung der Organisationsstruktur an die neue Produktionsweise<sup>5)</sup> und entgehen Einnahmen durch Mitarbeiterfreistellungen für Werkzeug- und Methodenschulungen<sup>6)</sup>. Infolge des erhöhten Zeitaufwands in der Einarbeitungsphase und der Tatsache, daß die Kostenersparnis sich erst in den Pflege- und Wartungsphasen bemerkbar macht<sup>7)</sup>, sind meßbare Produktivitätssteigerungen in den ersten Investitionsperioden nicht zu erwarten. Die Gültigkeit und Zuverlässigkeit von Produktivitätsmessungen ist allerdings problematisch,<sup>8)</sup> weil die Produktivität durch eine Reihe von Größen beeinflusst wird, so daß ein Produktivitätsvergleich erschwert wird.<sup>9)</sup> Wirtschaftlichkeitsanalysen der Anbieter, insbesondere wenn sie nur einen Teil der entstehenden Einführungskosten berücksichtigen und den Lernkurveneffekt<sup>10)</sup> ignorieren, wecken beim Käufer nicht zu erfüllende Erwartungen. Insgesamt betrachtet entwickelt sich die Softwareindustrie durch die Computerunterstützung der Softwareentwicklung von einer arbeits- zu einer nun auch kapitalintensiven Branche.

---

1) Vgl. Coqui /Katalysator "CASE"/ B7

2) Vgl. Coqui /Katalysator "CASE"/ B7

3) Vgl. Balzert /Anforderungen/ 114

4) Vgl. Seibt/ Anwendungserfahrungen/ o. S.

5) Vgl. Frank /Standardsoftware/ 34-35; Hamilton, Staff /Look before you leap/ 22

6) Vgl. McClure /CASE Experience/ 236; Sayles /Training and CASE/ 23-24

7) Vgl. Stobbe /Softwareentwicklungsumgebungen/ 48

8) Vgl. Andresen /Warum scheitern Software-Entwicklungsumgebungen/ 249

9) Vgl. Griese u. a. /Ergebnisse/ 516

10) Aufgrund der geringen Vertrautheit der Mitarbeiter mit neuen Methoden und Werkzeugen ist in der Einarbeitungsphase eher mit Produktivitätssenkungen zu rechnen.

## 2.2 Architektur

Zentraler Bestandteil der SEU ist ein Datenbanksystem mit einem zugrundeliegenden, objektorientierten Datenmodell.<sup>1)</sup> In dieses werden die Entwurfsdaten möglichst projektübergreifend abgelegt.<sup>2)</sup> Zur Eingabe dieser Informationen dienen grafikorientierte Editoren,<sup>3)</sup> die in eine einheitliche Benutzeroberfläche eingebunden sind. Durch die Einbindung in die Benutzerschnittstelle kann die Verwendung der methodenspezifischen Entwurfssprachen für den Entwickler verborgen ablaufen.<sup>4)</sup> Analysatoren prüfen die Eingabe, transformieren sie und speichern sie in der Datenbanksprache; Generatoren sorgen für die Rückübersetzung sowie für die automatisierte Quellcode-Erstellung in verschiedene Programmiersprachen.<sup>5)</sup>

Bei der Gestaltung der Hardwarearchitektur ist eine Aufgabenteilung zwischen einem oder mehreren PC oder Workstation und einem Mini- oder Großrechner sinnvoll.<sup>6)</sup> Eine Workstation mit hochauflösender Farbgrafik, leistungsstarkem Prozessor, Mausunterstützung und angeschlossenem Drucker ist infolge der Grafikfähigkeit und der kurzen Antwortzeiten besonders für interaktive Entwicklungsaktivitäten geeignet.<sup>7)</sup> Der Mini- oder Großrechner mit hoher Speicherkapazität kann die Aufgaben der Code-Generierung, der Verbindung zu Sprachen der vierten Generation, der Programm- und Systemtests, der Konsistenzprüfungen, der Normalisierungsprozesse und der Informationskonsolidierung besser lösen.<sup>8)</sup> Auf der Workstation sollte ein lokales Repository<sup>9)</sup>, auf dem Großrechner das Projekt- oder Produktrepository implementiert sein.<sup>10)</sup> Die Kommunikation der Entwickler

---

1) Vgl. Preßmar /Strategien/ 20

2) Vgl. Schulz /Klassifizierungs- und Bewertungsschema/ 193

3) Vgl. Curth, Wyss /Information engineering/ 195; Österle /CASE/ 353

4) Vgl. Preßmar /Strategien/ 22

5) Vgl. Preßmar /Strategien/ 21-22

6) Vgl. Bröhl /SE-Umgebung/ 97; Herzwurm /Möglichkeiten und Grenzen/ o. S.

7) Vgl. McClure /Automation/ 130; Nastansky /PC-basierte Enduser-Tools/ 73

8) Vgl. McClure /Automation/ 125-130; Preßmar /Strategien/ 20

9) Der Begriff Repository wird gleichbedeutend mit Entwicklungsdatenbank verwendet. Viele Anbieter, u.a. IBM, verwenden diesen Begriff zur Benennung ihrer Entwicklungsdatenbank.

10) Vgl. McClure /Automation/ 128-131

untereinander oder zum Großrechner erfolgt sinnvoll über ein lokales Netz.<sup>1)</sup>,  
2)

### 2.3 Markt

Auf dem Markt für SEU bieten ca. 60 Anbieter PC-basierte Produkte an. Die Preise in diesem Marktsegment schwanken zwischen TDM 1,8 für die SEU POSE von Computer Systems Advisor Inc. bis zu TDM 35 für das Starter-Kit der Information Engineering Workbench von KnowledgeWare.<sup>3)</sup> Die meisten der SEU sind entweder auf den technischen oder auf den kaufmännischen Bereich spezialisiert. Sie unterstützen ähnliche Methoden,<sup>4)</sup> stellen diese jedoch unterschiedlich zusammen und setzen andere Schwerpunkte<sup>5)</sup>. Die SEU sind derzeit nicht standardisiert<sup>6)</sup> und werden ständig weiterentwickelt.<sup>7)</sup> Die Ankündigung des AD/Cycle-Konzepts der IBM<sup>8)</sup> begünstigt den PC-basierten Sektor des CASE-Markts.<sup>9)</sup>

### 3 Kriterien für die Auswahl von Software-Entwicklungsumgebungen

Zur Gewährleistung einer besseren Übersichtlichkeit werden die Kriterien gemäß folgender Abbildung gegliedert:

- 
- 1) Vgl. Preßmar /Strategien/ 20; Tobiasch /Anforderungen/ 28
  - 2) Das Nutzen sowohl der Vorteile des PC als auch der des Großrechners wird auch bei dem von IBM angekündigten AD/Cycle-Konzept ermöglicht werden.
  - 3) Vgl. Fersko-Weiss /CASE Tools/ 220-221
  - 4) Vgl. Warner, Woschinski /Qual der Wahl/ 58-59
  - 5) Vgl. Fisher /CASE/ 172
  - 6) Vgl. Gibson /CASE Philosophy/ 209
  - 7) Vgl. Warner, Woschinski /Qual der Wahl/ 58
  - 8) Vgl. Kapitel 3.2.3.4
  - 9) Vgl. Fersko-Weiss /CASE Tools/ 214

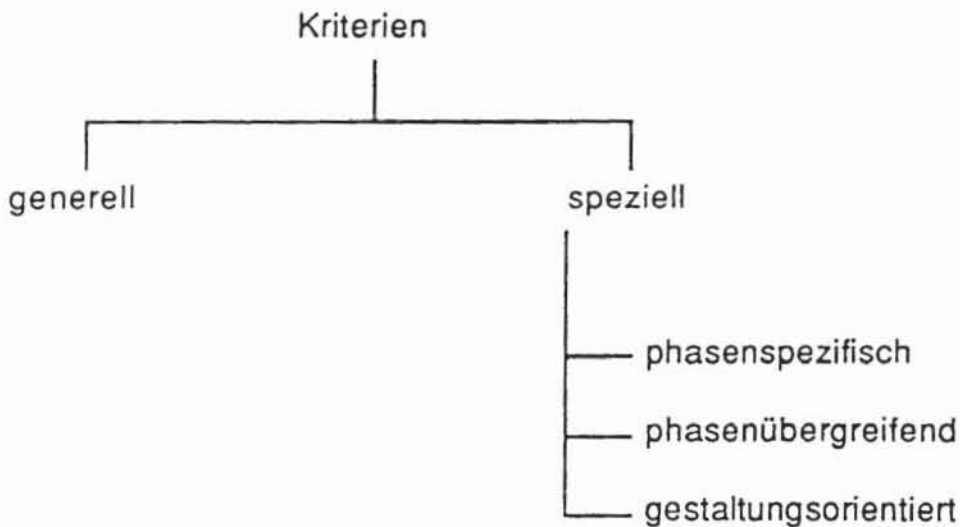


Abb. 1: Kriterien zur Auswahl von Software-Entwicklungsumgebungen

### 3.1 Generelle Kriterien für die Auswahl von Standardsoftware

Für SEU gelten neben speziellen Kriterien auch solche, die für jede Standardsoftware relevant sind.

So ist die Frage der notwendigen und empfohlenen Konfiguration, die sowohl die Anforderungen an die Hardware als auch an die Systemsoftware einschließt,<sup>1)</sup> schon aus Kostengesichtspunkten für eine Kaufentscheidung interessant. Eine höhere Bedeutung kommt diesem Kriterium zu, wenn beim Entwickler eine Systemlandschaft bereits vorhanden ist.<sup>2)</sup> Soll die SEU zu Demonstrationszwecken beim Anwender installiert werden können, kommen ihr geringe Systemanforderungen zugute: sie reduzieren die Installierungsprozedur, weil weniger oder keine Anpassungen des Systems notwendig werden.

Wird die SEU in der Unternehmung für einen langen Zeitraum installiert, so ist die Frage nach dem Installierungskomfort relativ unbedeutend. Wird sie hingegen häufig auf unterschiedlicher Hardware installiert, unter Umständen auch beim Anwender, so ist der Zeitbedarf der Installierungsprozedur<sup>3)</sup> sowie die überprüfbarkeit der korrekten Installation<sup>4)</sup> eine wichtige Eigenschaft. Ein

---

1) Vgl. McClure /Automation/ 180; Stahlknecht, Warner /Untersuchung/ 7

2) Vgl. Frank /Standardsoftware/ 35-36

3) Vgl. Frank /Standardsoftware/ 39

4) Vgl. DIN /DIN 66285/ 6; Software-Prüfstelle des TÜV Berlin /Rahmenprüfplan/ 8

interessantes Kriterium ist gerade bei SEU mit hohem Speicherbedarf die Möglichkeit einer partiellen Installierung oder einer Deinstallierung.

Die Akzeptanz beim Entwickler erhöht sich, wenn das Datenmodell des Repository modifizierbar ist.<sup>1)</sup> Die Gestaltbarkeit der einzelnen Werkzeuge hinsichtlich der zur Verfügung stehenden Symbole und der Benutzerführung sowie der Möglichkeit, eigene Diagramme zu entwerfen,<sup>2)</sup> ermöglichen eine Anpassung der SEU an die Belange des Entwicklers. Somit können unternehmensinterne, bewährte Methoden und Vorgehensweisen weiter gepflegt werden. Durch die Verwendung eines gewohnten Vokabulars und bekannter Symbole wird der Entfremdungseffekt bei der Anwendung der neuen Technik reduziert und die Entwicklerakzeptanz gefördert. Die Portabilität der SEU ist besonders für Unternehmungen mit verschiedenen Hardwarelandschaften und vor dem Hintergrund einer Portierung auf die Anlage eines Anwenders ein wichtiges Argument.<sup>3)</sup>

Eng mit der Änderbarkeit ist die Offenheit der SEU verbunden. CASE-Produkte können zur Zeit nicht die Gesamtheit der an sie gestellten Aufgaben erfüllen.<sup>4)</sup> Auch über einen längeren Zeitraum wird keine SEU Vollständigkeit bieten können, denn in der Zukunft werden sich wiederum neue Anforderungen ergeben.<sup>5)</sup> Deshalb sollten SEU offen sein und somit Ausbaufähigkeit<sup>6)</sup> zum Schutz der in sie getätigten Investitionen<sup>7)</sup> bieten. Sie sollten in der Unternehmung bereits vorhandene<sup>8)</sup> oder neue Entwicklungswerkzeuge einbinden oder mit ihnen kommunizieren können.<sup>9)</sup> Wettbewerbsvorteile besitzen SEU, die in der Lage sind, Schnittstellen zu Standardsoftware, zu relationalen

- 
- 1) Vgl. Index Technology /A Guided Tour of Customizer/ 6-9, 19
  - 2) Vgl. Index Technology /A Guided Tour of Customizer/ 12-17, 19; Stobbe /Softwareentwicklungsumgebungen/ 40
  - 3) Vgl. Curth, Wyss /Information engineering/ 197; Hering /Software-Engineering/ 3
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 276; Kimmel, Lehnhard /Vollständige SPU/ 28
  - 5) Vgl. Balzert /Anforderungen/ 106; Schulz /Klassifizierungs- und Bewertungsschema/ 196
  - 6) Vgl. Hering /Software-Engineering/ 3; Ruf /Basis des Schnittstellen-Management-Ansatzes/ 97
  - 7) Vgl. Althammer, Bernath /SAA/ 207
  - 8) Vgl. Boehm u. a./Improving Productivity/ 31; Ruf /Basis des Schnittstellen-Management-Ansatzes/ 97
  - 9) Vgl. Curth, Wyss /Information engineering/ 203; Österle /CASE/ 358-359



Datenbanken oder zu einer anderen Entwicklungsdatenhaltung anzubieten.<sup>1)</sup> Ein durch das AD-Cycle-Konzept der IBM in greifbare Nähe gerücktes Ziel ist die Verbindungsmöglichkeit mehrerer CASE-Tools unterschiedlicher Hersteller. Die Austauschbarkeit von Methoden, Vorgehensweisen und Lebenszyklusmodellen ist eine weitere zu fordernde Eigenschaften für CASE-Produkte.

Einen wesentlichen Einfluß auf die Akzeptanz der neuen Technologie besitzt die Gestaltung der Benutzerschnittstelle.<sup>2)</sup> Zur Beurteilung von CASE-Produkten sind die Verständlichkeit, die Benutzbarkeit, die Gestaltung der Hilfefunktion, die Fehlerbehandlung und das Antwortzeitverhalten relevante Kriterien.<sup>3)</sup> Da zunehmend Spezialisten aus den Anwendungsbereichen Aufgaben in der Systementwicklung übernehmen, kommt einer für den Entwickler verständlichen Benutzeroberfläche eine steigende Bedeutung zu.<sup>4)</sup> So sind überschaubare Masken sowie selbsterklärende, aussagekräftige und möglichst nationalsprachliche Kommandonamen zu fordern.<sup>5)</sup> Eine aufgabenangemessene<sup>6)</sup> SEU zeichnet sich durch eine gute Benutzbarkeit aus, so daß der Entwickler sich auf seine eigentlichen Aufgaben konzentrieren kann<sup>7)</sup>. Die SEU muß einen brauchbaren Kompromiß finden zwischen einer an die Erfahrung des Entwicklers anpaßbaren Schnittstelle<sup>8)</sup> und einer konsistenten Benutzeroberfläche<sup>9)</sup> mit einheitlicher Bildschirmgestaltung und Funktionstastenbelegung.<sup>10)</sup> Daneben sollte das CASE-Produkt eine permanente Orientierung im Programm durch die Bereitstellung von Überblicksdiagrammen, Statuszeilen und die grafische Hervorhebung des bearbeiteten Inkrements gewährleisten.<sup>11)</sup> Um dem Entwickler ein häufiges Wechseln zwischen den

- 
- 1) Vgl. Chikofsky, Rubenstein /Reliability Engineering/ 13; Krüger /Anwendungserfahrungen/ o. S.
  - 2) Vgl. Zwerina /Masken und Formulare/ 172
  - 3) Vgl. vertiefend DIN /DIN 66234/ 1-5; DIN /DIN 66285/ 6- 7; Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 23
  - 4) Vgl. Leppert, Stork /Entwicklungsumgebungen/ 215
  - 5) Vgl. McClure /Automation/ 217, 231; Schäfer /Benutzerschnittstelle/ 422
  - 6) Vgl. DIN /DIN 66234/ 2
  - 7) Vgl. Stobbe /Softwareentwicklungsumgebungen/ 39
  - 8) Vgl. Ackermann /Empirie des Softwareentwurfs/ 253-254; Koslowski /Unterstützung partizipativer Systementwicklung/ 103
  - 9) Vgl. Eberleh /Menüauswahl/ 132; Schäfer /Benutzerschnittstelle/ 421
  - 10) Vgl. Gierlach, Jankowski /Operationalisierung/ 190
  - 11) Vgl. Eberleh /Menüauswahl/ 127; Gane /Computer-Aided Software Engineering/ 132

einzelnen Diagrammen zu ersparen,<sup>1)</sup> sollten alle relevanten Informationen durch die Verwendung von Bildschirmteilungs- oder Fenstertechniken auf möglichst einer Bildschirmseite plaziert werden können.<sup>2)</sup> Grafiken und Mausunterstützung sind für CASE-Produkte heute selbstverständlich,<sup>3)</sup> untersucht werden soll der Bedienungskomfort der Diagrammtypen. Außerdem stellt die Unterstützung des WYSIWYG-Prinzips<sup>4)</sup> eine wesentliche Forderung an eine ergonomische SEU dar.<sup>5)</sup> Die ideale Benutzeroberfläche sollte eine Hilfefunktion überflüssig machen, auf keinen Fall aber sollte die Hilfefunktion für sich selbst Hilfe beanspruchen. Eine gute Hilfefunktion berücksichtigt das Benutzerprofil und den Kontext ihres Aufrufs.<sup>6)</sup> Weiter ist zu fordern, daß sie nur in offensichtlich notwendigen Fällen ohne Aufruf erscheint. Ausführliche Informationen über die Wirkungsweise und Anwendung von Werkzeugen und Funktionen sollten nur auf Verlangen geboten werden,<sup>7)</sup> um den Entwickler nicht mit für ihn unnötigen Informationen zu belasten. Zum Schutz der Entwicklungsergebnisse muß die SEU wiederum robust gegen unvorhergesehene Eingaben sein.<sup>8)</sup> So ist bei Aktionen mit schwerwiegenden Folgen das Erbitten einer Bestätigung sinnvoll.<sup>9)</sup> Manche CASE-Produkte heben das Symbol, bei dem der Fehler auftrat, visuell hervor. Die Fehlermeldungen sollten die Fehlerursache verständlich und hilfreich beschreiben und Lösungsalternativen zur Fehlerkorrektur aufzeigen.<sup>10)</sup> Der Korrekturaufwand kann beispielsweise durch die Stornierbarkeit von Befehlen minimiert werden.<sup>11)</sup> Soll die Ressource Arbeitskraft im Softwarehaus effizient genutzt werden, bedarf es außerdem eines akzeptablen Antwortzeitverhaltens in der SEU.<sup>12)</sup> Unan-

- 
- 1) Vgl. Ackermann /Emperie des Softwareentwurfs/ 253-254
  - 2) Vgl. Gierlach, Jankowski /Operationalisierung/ 191; Schäfer /Benutzerschnittstelle/ 421
  - 3) Vgl. Ernst /Methodenunterstützung/ 45
  - 4) Bei der Realisierung des WYSIWYG-Prinzips stimmt die Bildschirmdarstellung mit der Druckausgabe überein. WYSIWYG = What you see is what you get
  - 5) Vgl. Schulz /Software-Entwurf/ 56
  - 6) Vgl. Bauer, Schwab /Anforderungen an Hilfesysteme/ 198
  - 7) Vgl. Gierlach, Jankowski /Operationalisierung/ 191
  - 8) Vgl. DIN /DIN 66285/ 6; McClure /Automation/ 217-218
  - 9) Vgl. DIN /DIN 66285/ 7; McClure /Automation/ 226-227
  - 10) Vgl. Bauer, Schwab /Anforderungen an Hilfesysteme/ 199; McClure /Automation/ 217, 219, 226-227
  - 11) Vgl. DIN /DIN 66285/ 6; Gierlach, Jankowski /Operationalisierung/ 191
  - 12) Vgl. Hering /Software-Engineering/ 1-2; Koslowski /Unterstützung

gemessenes Dialogverhalten mindert die Mitarbeitermotivation.<sup>1)</sup> Kann dies bei der Abspeicherung oder Auswertung großer Datenmengen nicht realisiert werden, sollte für den Entwickler wenigstens keine unproduktive Zwangspause entstehen.<sup>2)</sup>

Die Bewertung der Dokumentation und der Produktbeschreibung gibt Aufschluß über den Reifegrad des Softwareprodukts.<sup>3)</sup> Eine gute Dokumentation und Produktbeschreibung macht den Softwareentwickler unabhängig vom CASE-Anbieter<sup>4)</sup>. Bei der Bewertung der Vollständigkeit dieser Unterlagen wird untersucht, ob die Benutzerdokumentation alle Informationen für eine sinnvolle Anwendung der SEU liefert<sup>5)</sup> und ob die Systemdokumentation die Installierung, den Betrieb, die Wartung und Pflege sowie den Test des Produkts ermöglicht<sup>6)</sup>. Daneben werden die Richtigkeit, Widerspruchsfreiheit und Aktualität untersucht.<sup>7)</sup> Damit der Entwickler nur die zur Lösung seines Problems notwendigen Textstellen studieren muß, kommen der Übersichtlichkeit und Strukturiertheit der Dokumentation eine besondere Bedeutung zu<sup>8)</sup>.

Ein weiterer für den Erfolg des CASE-Produkts wichtiger Faktor ist die Einführungsunterstützung.<sup>9)</sup> Notwendiger als das Erlernen der Bedienung der SEU selbst sind Methodenschulungen für die Entwickler.<sup>10)</sup> Um auf die besonderen Belange der einzelnen Entwickler eingehen zu können, sollten die Einführungsveranstaltungen möglichst in der Unternehmung des Softwareentwicklers stattfinden.<sup>11)</sup> Schulungsunterlagen ermöglichen ein Selbststudium und

---

partizipativer Systementwicklung/ 103

- 1) Vgl. Koslowski /Unterstützung partizipativer Systementwicklung/ 103
- 2) Vgl. Gierlach, Jankowski /Operationalisierung/ 195
- 3) Vgl. Frank /Standardsoftware/ 40
- 4) Vgl. Frank /Standardsoftware/ 40
- 5) Vgl. DIN /Beiblatt 1 zu DIN 66230/ 2; Sommerville /Software Engineering/ 230
- 6) Vgl. DIN /Beiblatt 1 zu DIN 66230/ 5; Sommerville /Software Engineering/ 230
- 7) Vgl. Asam, Drenkard, Maier /Qualitätsprüfung/ 113; Sommerville /Software Engineering/ 233
- 8) Vgl. DIN /DIN 66285/ 5; Ewald /Software/ 96
- 9) Vgl. Sayler /Training and CASE/ 23; Skubch /CASE-Einführungen/ VI-3-07
- 10) Vgl. McClure /CASE Experience/ 242; Sayler /Training and CASE/ 23
- 11) Vgl. Frank /Standardsoftware/ 40

eine effektive Vorbereitung auf die Einführungsveranstaltung.<sup>1)</sup> Desweiteren ist die Benennung von Kontaktpersonen in der Anbieterunternehmung zu fordern, die beim erstmaligen Einsatz der SEU konsultiert werden können, falls Probleme auftreten.<sup>2)</sup>

Ein wesentliches Kriterium einer jeden Kaufentscheidung ist der Preis. Beim Vergleich sollen allerdings nicht nur Beträge gegenübergestellt werden, denn es gibt eine Vielzahl von Konditionsgestaltungen. So soll untersucht werden, ob die o. g. Einführungsunterstützungen im Preis enthalten sind<sup>3)</sup> und ob Probeinstallierungen möglich sind<sup>4)</sup>. Auch die Erlaubnis einer parallelen mehrfachen Nutzung muß geklärt werden.<sup>5)</sup> Desweiteren muß die Frage der Wartungskosten einschließlich Wartungszeitraum, Übergabe der neuesten Produktversionen, Aktualisierung von Handbüchern und Nachschulungen bei der Beurteilung des Preises berücksichtigt werden.<sup>6)</sup> Weiterhin ist zu klären, ob nach Ablauf des im Kaufpreis eingeschlossenen Wartungsintervalls eine Wartungsoption besteht.<sup>7)</sup>

Die Entscheidung für ein CASE-Produkt führt zu einem hohen Investitionsvolumen. Es ist daher verständlich, daß ein kompetenter und potenter Geschäftspartner gesucht wird.<sup>8)</sup> Der Investor orientiert sich am Alter der vertriebenen SEU, an der Anzahl ihrer Installationen und der wirtschaftlichen Lage des Anbieters.<sup>9)</sup> Interessant ist für ihn auch die Entfernung zum nächsten Kundenstützpunkt des Verkäufers.<sup>10)</sup>

---

1) Vgl. Frank /Standardsoftware/ 40; Sullivan-Trainor /Buyers' Scorecard/ 69

2) Vgl. Frank /Standardsoftware/ 51

3) Vgl. Frank /Standardsoftware/ 44

4) Vgl. Balzert /Entwicklung/ 8; Frank /Standardsoftware/ 51

5) Vgl. Frank /Standardsoftware/ 53

6) Vgl. Frank /Standardsoftware/ 42

7) Vgl. Frank /Standardsoftware/ 43

8) Vgl. Fisher /CASE/ 258

9) Vgl. Ewald /Software/ 88; Warner, Woschinski /Qual der Wahl/ 59

10) Vgl. Ewald /Software/ 88

## 3.2 Spezielle Kriterien für die Auswahl von Software-Entwicklungsumgebungen

### 3.2.1 Unterstützung der Phasen des Software-Lebenszyklusses

Phasenmodelle unterteilen die in der Softwareentwicklung anfallenden Tätigkeiten und ordnen sie bestimmten Phasen zu.<sup>1)</sup> Es haben sich im Software Engineering<sup>2)</sup> eine Anzahl von Vorgehensweisen und Phasenmodellen herausgebildet, die sich in Bezeichnungen, Zuordnungen der einzelnen Aufgaben und deren Detaillierungsgrad unterscheiden.<sup>3)</sup> Eine allgemein anerkannte Phasenabgrenzung hat sich noch nicht herausgebildet.<sup>4), 5)</sup> Eine spezielle Abgrenzung ist beim Einsatz eines CASE-Produkts nicht unbedingt erforderlich, es sollte jedoch das gesamte Tätigkeitspektrum der Softwareentwicklung abgedeckt, die Qualitätssicherung und die Dokumentation der Phasenergebnisse vollständig unterstützt werden.<sup>6)</sup> Um die Vollständigkeit der Werkzeuge im Hinblick auf die Phasenabdeckung zu verifizieren,<sup>7)</sup> orientiert sich diese Arbeit am Software-Lebenszyklusmodell von Schmitz und Seibt<sup>8)</sup>, weil dieses organisatorische Aspekte berücksichtigt, die Wartungs- und Pflegephasen explizit einschließt und die durch den Einsatz von SEU ermöglichte Überprüfung der Anforderungen des Entwurfs bereits zu frühen Zeitpunkten vorschlägt.<sup>9)</sup>

#### 3.2.1.1 Entwurfs- und Analysephase

In dieser Phase, die setzt sich aus der Initialisierung, Voruntersuchung, dem Grobkonzept und Detailentwurf zusammen,<sup>10)</sup> soll die Basis der Implementie-

---

1) Vgl. Koslowski /Unterstützung partizipativer Systementwicklung/ 128

2) Vgl. IEEE/ANSI /IEEE Standard Glossary/ 32

3) Vgl. Balzert /Entwicklung/ 15

4) Vgl. Schmitz, Seibt /Einführung/ 10

5) So hat sich auch der Software Life Cycle des American National Standards Institute nicht durchsetzen können. Vgl. IEEE/ANSI /IEEE Standard Glossary/ 32

6) Vgl. Chikosfky, Rubenstein /Reliability Engineering/ 12; Wallmüller /Methode und Werkzeug/ 424

7) Vgl. McClure /Automation/ 21, 71; Österle /CASE/ 358

8) Vgl. Schmitz, Seibt /Einführung/ 9; Seibt /RBIS Skriptum/ 31-70

9) Vgl. Seibt /Phasen und Aktivitäten/ 12; Seibt /Phasenkonzept/ 255; Seibt /Systemlebenszyklus/ 326

10) Vgl. Schmitz, Seibt /Einführung/ 9

rung erstellt werden.<sup>1)</sup> Die Qualität dieser Grundlage beeinflusst die nachfolgenden Phasen des Softwareerstellungsprozesses, insbesondere die Wartungsphasen, denn eventuelle Fehler im Entwurf verursachen einen erheblichen Mehraufwand, falls sie erst in späteren Phasen entdeckt und behoben werden.<sup>2)</sup> Der Entwurf der Software beeinflusst somit die Wirtschaftlichkeit des Entwicklungsprozesses, neben der Zuverlässigkeit, Änderbarkeit und Portabilität des Softwareprodukts.<sup>3)</sup>

Zur Verifikation einer vollständigen Unterstützung der Entwurfs- und Analysephase müssen zunächst die softwaremäßigen Voraussetzungen aus den Einzeltätigkeiten dieser Phase abgeleitet und in einem zweiten Schritt deren Erfüllung geprüft werden. So werden für die Zielformulierung und -begründung des neuen Softwareprodukts, für die Analyse und Strukturierung der Informationen aus der Istaufnahme von "Bedürfnissen, Zielen, Strukturen und Abläufen im Benutzerbereich"<sup>4)</sup> und für die Erstellung von Pflichtenheft und Aufgabenabgrenzung sprachliche Beschreibungsmittel wie Textverarbeitungssysteme, Matrizen, eventuell strukturierte Checklisten und grafische Beschreibungsmittel benötigt.<sup>5)</sup> Für die Grobkonzepte und Detailentwürfe müssen Beschreibungsmittel für verschiedene Sichtweisen zur Verfügung stehen: so sollten für die datenorientierte Sicht Werkzeuge für konzeptionelle Datenmodellierung und logische Datenstrukturen vorhanden sein.<sup>6)</sup> Ebenso ist ein Darstellungsmittel für den Fluß der Daten durch das System und zur Prozeßbeschreibung zu fordern.<sup>7)</sup> Für die Planung der Benutzerschnittstellen müssen Generatoren und Simulatoren zur Verfügung stehen.<sup>8)</sup> In sämtlichen Phasen sollten die Prinzipien der Softwareerstellung unterstützt werden, insbesondere das Top-Down-Prinzip und das Prinzip der Modularität.<sup>9)</sup> Alle In-

- 
- 1) Vgl. Koslowski /Unterstützung partizipativer Systementwicklung/ 116
  - 2) Vgl. Boehm /Software-Produktion/ 33; Schmitz /Methoden, Verfahren und Werkzeuge/ 74
  - 3) Vgl. McClure /Automation/ 72; Stobbe /Softwareentwicklungsumgebungen/ 35
  - 4) Seibt /RBIS Skriptum/ 49
  - 5) Vgl. Hildebrand /Software Tools/ 259; Wirtz /Softwareentwurf/ 325
  - 6) Vgl. Crozier u. a. /Critical analysis/ 487; Vetter /Konzeptionelle Datenmodellierung/ 386-387
  - 7) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 86, Scheibl /Kommerzielle Software-Entwicklung/ 292
  - 8) Vgl. Fisher /CASE/ 38; Scheibl /Kommerzielle Software- Entwicklung/ 305
  - 9) Vgl. Fisher /CASE/ 41; Gewalt, Haake, Pfadler /Software Engineering/ 67, 72;

formationen müssen in ein integriertes Repository eingespeist werden können, auf das aus allen Phasen zugegriffen werden kann.<sup>1)</sup> Wie alle Phasen muß auch die Entwurfs- und Analysephase geplant und kontrolliert werden.<sup>2)</sup> Deshalb sind Planungswerkzeuge für den Ressourceneinsatz und Analysehilfsmittel zu fordern. Diagramme sollten einfach und schnell erstellt werden, um für Design Reviews zur Verfügung zu stehen<sup>3)</sup>, Testgeneratoren sollten die Aufstellung von Testfällen ermöglichen<sup>4)</sup>. Produktivitätssteigerungen in der Entwurfs- und Analysephase können durch die Wiederverwendung von Entwurfsspezifikationen, Masken, Berichten, Datenmodellen und Modulen erreicht werden.<sup>5)</sup> Voraussetzung hierfür ist eine Ausrichtung auf deren Wiederverwendbarkeit bereits in der Entwicklung.<sup>6)</sup> Dies geschieht durch die Anwendung von Standards und geeigneten Dokumentationen,<sup>7)</sup> um die Änderungen und die Zusammensetzung zu unterstützen. Zum Auffinden von wiederverwendbaren Teilen wird ein möglichst projektübergreifendes Bibliothekssystem benötigt.<sup>8)</sup> Sollen Programmteile unplanmäßig wiederverwendet werden, bedarf es zusätzlicher Werkzeuge für die Respezifikation und Restrukturierung.<sup>9)</sup>

### 3.2.1.2 Realisierungsphase

Die Realisierungsphase umfaßt alle Tätigkeiten, die dazu dienen, den Entwurf in ein ablauffähiges Programm umzusetzen und organisatorische Voraussetzungen für den Einsatz der neuen Software zu schaffen.<sup>10)</sup> Sie umfaßt auf DV-technischer Seite die Code-Generierung, das Testen und die Dokumentation der erstellten Programme. Durch eine Automatisierung der Code-Generierung können Entwicklungszeit gespart, die Zuverlässigkeit der Pro-

---

1) Vgl. Hildebrand /Software-Life-Cycle/ 47

2) Vgl. Boehm /Software-Produktion/ 25; Schmitz, Seibt /Einführung/ 10

3) Vgl. Fisher /CASE/ 44

4) Vgl. Seibt /RBIS Skriptum/ 59

5) Vgl. Boehm u. a. /Improving Productivity /37; Sommerville, Morrison /Software Development/ 324

6) Vgl. McClure /Automation/ 265

7) Vgl. Martin /Information Engineering, Book II/ 276; McClure /Automation/ 267

8) Vgl. Balzert /Prinzipien/ 4; Hruschka /Evolutions-Zyklen/ 87

9) Vgl. Lempp /vorhandene Software nutzen/ 43

10) Vgl. Seibt /RBIS Skriptum/ 62-64

gramme erhöht und Wartungskosten gesenkt werden.<sup>1)</sup>

Deshalb ist die Integration von automatisierten Code-Generatoren oder zumindest eine Schnittstelle zu solchen zu fordern.<sup>2)</sup> Diese sind nach ihrem Automatisierungsgrad<sup>3)</sup> und den erstellbaren Zielsprachen<sup>4)</sup> zu bewerten. Weiterhin ist die Möglichkeit wünschenswert, die Programme auf den Großrechner zu portieren. Der durch den Code-Generator produzierte Quellcode sollte von dem manuell erstellten Code getrennt sein.<sup>5)</sup> Die SEU sollte außerdem über die Fähigkeit verfügen, die Programme automatisiert zu dokumentieren.<sup>6)</sup> In der Realisierungsphase müssen umfangreiche Qualitätssicherungsmaßnahmen durchgeführt werden: Baustein-, Verfahrens- und Systemtests.<sup>7)</sup> Hierzu werden Testdatengeneratoren, möglichst dynamische Debugger, Teststatistikprogramme und Testvergleichswerkzeuge benötigt.<sup>8)</sup> Für die organisatorische Seite der Realisierung werden Textverarbeitung sowie Dokumentations- und Planungswerkzeuge benötigt.<sup>9)</sup>

### 3.2.1.3 Erprobungs- und Konsolidierungsphase

In dieser Phase finden die Übergabe des neuen Programms an den Anwender, seine Einführung und regelmäßige Kontrolle mit sich eventuell anschließenden Feinanpassungen statt.<sup>10)</sup> Größtenteils besteht diese Phase aus organisatorischen Aktivitäten,<sup>11)</sup> die weitgehend nur durch Planungswerkzeuge und Textverarbeitung von der SEU unterstützt werden können.

Für den DV-technischen Teil sollte die SEU Testwerkzeuge für den System-

---

1) Vgl. Gibson /CASE Philosophy/ 216; McClure /Automation/ 142-143

2) Vgl. Chikofsky, Rubenstein /Reliability Engineering/ 13

3) Vgl. Crozier u. a. /Critical analysis/ 491; McClure /Automation/ 75

4) Vgl. McClure /Automation/ 77; Österle /Computer Integrated Software-Engineering/ 14

5) Vgl. Sommerville /Software Engineering/ 138

6) Vgl. McClure /CASE Experience/ 77, 238

7) Vgl. Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 75; Seibt /RBIS Skriptum/ 63 (Schmitz, Bons, van Megen ordnen im Gegensatz zu Seibt den Systemtest der Einführungsphase zu.)

8) Vgl. Hildebrand /Software Tools/ 261; McClure /Automation/ 142

9) Vgl. Seibt /RBIS Skriptum/ 62

10) Vgl. Seibt /RBIS Skriptum/ 65-70

11) Vgl. Hildebrand /Software Tools/ 261; Seibt /RBIS Skriptum/ 65-70



test in echter Umgebung und zur Performanceüberwachung bereitstellen.<sup>1)</sup> Weiterhin müssen geringfügige Codekorrekturen für Tuning- und Optimierungsmaßnahmen ermöglicht werden.<sup>2)</sup> Weitreichende Korrekturen sollten nicht im Quellcode vorgenommen werden. Hierzu sollte immer ein kleiner, vollständiger Softwareentwicklungszyklus durchlaufen werden. So wird die Konsistenz zwischen Entwurf und Programmcode gewährleistet. Bei der Datenübernahme in das neue System des Anwenders können CASE-Produkte durch Bereitstellung von Konvertierungsprogrammen Hilfestellung bieten.<sup>3)</sup>

#### 3.2.1.4 Pflege- und Wartungsphase

Nach erfolgter Entwicklung und Übergabe der neuen Software beginnt die Betriebsphase des Software-Lebenszyklusses, die immer wieder durch Pflege- und Wartungsphasen unterbrochen wird, um Fehler zu beheben, kleine funktionale Änderungen vorzunehmen oder die Performance zu verbessern.<sup>4)</sup> Weil ein Großteil der Ressourcen in der Softwareentwicklung für die Pflege- und Wartungsphasen aufgebracht werden muß,<sup>5)</sup> sollte Software nicht nur zum Zeitpunkt der Übergabe optimal funktionieren, darüberhinaus sollte sie einfach zu modifizieren sein.<sup>6)</sup> Die Pflege- und Wartungsaktivitäten beinhalten sämtliche Aufgaben eines Software-Entwicklungszyklusses, jedoch wird mehr Einarbeitungszeit und mehr Aufwand für das Nachvollziehen von Änderungen benötigt.<sup>7)</sup>

Wurde die Software mit einer SEU erstellt, so können in der Entwicklungsdatenbank gespeicherte Teile der Systembeschreibung wiederverwendet werden.<sup>8)</sup> Diagramme ermöglichen ein schnelles Verstehen des Entwurfs, die Nutzung von Analysewerkzeugen eine Vollständigkeitsprüfung der Änderun-

- 
- 1) Vgl. Hildebrand /Software Tools/ 261; Stobbe /Softwareentwicklungsumgebungen/ 38
  - 2) Vgl. Hildebrand /Software Tools/ 261; Seibt /RBIS Skriptum/ 69
  - 3) Vgl. Hildebrand /Software Tools/ 261; Seibt /RBIS Skriptum/ 69
  - 4) Vgl. Seibt /Phasen und Aktivitäten/ 1, 3
  - 5) Vgl. DeMarco /Structured Analysis/ 298; Preßmar /Strategien/ 11
  - 6) Vgl. Ernst /Methodenunterstützung/ 41
  - 7) Vgl. Hildebrand /Software Tools/ 261; Koslowski /Unterstützung partizipativer Systementwicklung/ 125
  - 8) Vgl. Chikofsky, Rubenstein /Reliability Engineering/ 15; Gibson /CASE Philosophy/ 216

gen<sup>1)</sup> und die Dokumentationsfunktionen eine Aktualisierung der Dokumentation.<sup>2)</sup> Komplexer wird das Problem der Wartung, wenn die Software nicht mit Hilfe einer SEU entwickelt wurde, sondern lediglich der Quellcode vorliegt. Reverse-Engineeringwerkzeuge bieten in diesem Fall die Möglichkeit, eine große Menge von bereits existierendem Code wiederzuverwenden.<sup>3)</sup> Zur Erfüllung dieser Aufgabe müssen Werkzeuge zur Nachdokumentation, zur Respezifizierung und zur Restrukturierung bereitgestellt werden.<sup>4)</sup> Ungeachtet, ob eine Spezifikation in der Entwicklungsdatenhaltung für die Wartung vorliegt oder erst erstellt werden muß, erheben die Pflege- und Wartungsphasen hohe Anforderungen an die Qualitätssicherung<sup>5)</sup> und an das Versionsmanagement<sup>6)</sup>.

### 3.2.2 Unterstützung phasenübergreifender Aufgaben

Bereits bei der Vollständigkeitsüberprüfung der Unterstützung der einzelnen Phasen der Softwareentwicklung wird deutlich, daß bestimmte Aufgaben in mehreren, einige sogar in jeder Phase anfallen. Zur Vermeidung von Wiederholungen sollen diese Aufgaben in einem gesonderten Kapitel behandelt werden.

#### 3.2.2.1 Informationsverwaltung

Kernstück einer SEU ist die Informationsverwaltung.<sup>7)</sup> Hier werden in vorgefertigten Masken die aus verschiedenen Sichtweisen gewonnenen Entwicklungsergebnisse in einheitlicher Form gespeichert.<sup>8)</sup>

- 
- 1) Vgl. Zaleski /Software-Wartung/ 30
  - 2) Vgl. Hering /Software-Engineering/ 77; Zaleski /Software-Wartung/ 30
  - 3) Vgl. McClure /CASE Experience /238-239; Parthier /Marktbild CASE/ 25
  - 4) Vgl. Martin /Information Engineering, Book I/ 125; Parthier /Marktbild CASE/ 25; Zaleski /Software-Wartung/ 30
  - 5) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 287; McClure /Automation/ 145
  - 6) Vgl. Balzert /Entwicklung/ 16; Leppert, Stork /Entwicklungsumgebungen/ 206
  - 7) In SEU wird die Informationsverwaltung häufig als Entwicklungsdatenbank, Data Dictionary, Encyclopedia oder Repository bezeichnet.
  - 8) Vgl. Crozier u. a. /Critical analysis/ 487; Schulz /Klassifizierungs- und Bewertungsschema/ 192

Zur Beurteilung der Informationsverwaltung ist zu prüfen, welche Angaben in ihr gespeichert werden können: neben Definitionen der Daten sollten Beziehungen, Regeln und physische Informationen in ihr abgelegt werden können.<sup>1)</sup> Zusätzlich sind in einer gut vordefinierten Datenstruktur die Festlegung von Domänen, eine Alias<sup>2)</sup>-Verwaltung sowie die Speicherung von Informationen über die Anzahl der erwarteten Datensätze in dem zu entwickelnden System möglich.<sup>3)</sup> Zur Projektüberwachung dienen Prüfungsfelder, die den Entwickler und Zeitpunkt der letzten Änderung festhalten.<sup>4)</sup> Trotz der Informationsvielfalt müssen die Informationen strukturiert gespeichert werden, um die Bildung einer Teilmenge für eine Analyse oder für Suchfunktionen zu ermöglichen.<sup>5)</sup> Beim Füllen der Entwicklungsdatenbank kann die Einhaltung von Standards unterstützt werden, indem Vorschläge für die Namensvergabe gemacht oder die Einhaltung von Konventionen erzwungen werden.<sup>6)</sup> Zur Analyse der Arbeitsergebnisse sollten die Einträge in der Entwicklungsdatenbank über vorgefertigte oder selbst zu erstellende Berichte oder durch direkte Abfragen aufgelistet werden können.<sup>7)</sup> Wesentlicher Beurteilungsfaktor ist die Integration der Informationsverwaltung in die SEU. Hier ist zu untersuchen, ob ein direkter Zugang aus den Diagrammen ermöglicht wird, ob Diagrammänderungen in der Entwicklungsdatenbank fortgeschrieben werden können und inwieweit die Qualitätssicherung, insbesondere die Konsistenzüberwachung, bei Einträgen gewährleistet wird.<sup>8)</sup>

### 3.2.2.2 Dokumentation

Ohne die Verfügbarkeit einer SEU erfordert die schriftliche Präsentation der Entwurfsergebnisse einen erheblichen Zeitaufwand. Sie wird deshalb von vielen Entwicklern als belastend empfunden und nur oberflächlich ausge-

- 
- 1) Vgl. Crozier u. a. /Critical analysis/ 492; McClure /Automation/ 57, 66
  - 2) Vgl. IEEE/ANSI /IEEE Standard Glossary/ 8
  - 3) Vgl. DeMarco /Structured Analysis/ 126, 143; Gane /Computer-Aided Software Engineering/ 46-48
  - 4) Vgl. Crozier u. a. /Critical analysis/ 494
  - 5) Vgl. Österle /CASE/ 358-359
  - 6) Vgl. DeMarco /Structured Analysis/ 158; Heuer /Rolle von Dictionary-Systemen/ 121
  - 7) Vgl. McClure /Automation/ 60; Ruf /Basis des Schnittstellen-Management-Ansatzes/ 234
  - 8) Vgl. Crozier u. a. /Critical analysis/ 493-494; Scheibl /Kommerzielle Software-Entwicklung/ 332

führt.<sup>1)</sup> Dennoch ist die Dokumentation Voraussetzung zur Anwendung und Bestandteil des Softwareprodukts.<sup>2)</sup> Nicht zum Zweck der Verfügbarkeit der Informationen oder aus Motivationsgründen wird eine entwicklungsbegleitende Dokumentationserstellung empfohlen, ihre Verfügbarkeit dient auch der Kommunikation der in die Produktentwicklung involvierten Personen und dem Festhalten von Zwischenergebnissen der einzelnen Entwicklungsphasen.<sup>3)</sup>

Eine SEU mit guter Dokumentationsfunktion muß die Erstellung einer Programmdokumentation ermöglichen, die sicherstellt, daß das Programm in vollem Umfang genutzt werden kann. Zusätzlich sollte eine vollständige Programmentwicklungsdokumentation mit ihrer Hilfe erstellt werden können.<sup>4)</sup> Zur Erfüllung dieser Aufgaben ist die Verfügbarkeit eines Textverarbeitungssystems zur Berichtserstellung und zur Einbindung von Grafiken notwendig.<sup>5)</sup> Zur Gewährleistung der Richtigkeit und der Aktualität sollten die Grafiken auf Konsistenz und Vollständigkeit geprüft werden, bevor sie in die Dokumentation eingebracht werden. Um eine übersichtliche Dokumentation zu ermöglichen, sollte die SEU deren strukturierten Aufbau unterstützen; ein grafischer Überblick der Dokumentation, Inhalts- und Stichwortverzeichnis wären wünschenswert.<sup>6)</sup> Die Dokumente selbst sollten an Unternehmungsstandards angepaßt werden können.<sup>7)</sup>

### 3.2.2.3 Software-Qualitätssicherung

Grafiken und Berichte werden nicht nur für die Dokumentation benötigt, gute Grafiken erleichtern auch das Verständnis der Projektbeteiligten in Walk-Throughs<sup>8)</sup>, Berichte dienen der Entwurfsanalyse. Durch die Verfügbarkeit von qualitätssichernden Instrumenten in der SEU werden Überprüfungen

- 
- 1) Vgl. Balzert /Anforderungen/ 113; Seibt /Phasen und Aktivitäten/ 75
  - 2) Vgl. DIN /DIN 66231/ 1; Schulz /Software-Entwurf/ 16
  - 3) Vgl. McClure /Automation/ 21; Seibt /Phasen und Aktivitäten/ 70
  - 4) Vgl. DIN /DIN 66230/ 1-12; DIN /DIN 66231/ 1-23
  - 5) Vgl. Fisher /CASE/ 147; Gane /Computer-Aided Software Engineering/ 89
  - 6) Vgl. Leinweber /Planung und Dokumentation/ VI-11-08; Sommerville /Software Engineering/ 231-234;
  - 7) Vgl. Boehm u. a. /Improving Productivity/ 37
  - 8) Überprüfungsprozeß, in dem ein Entwickler seinen Entwurf oder seine Codierung vorstellt und andere in das Projekt eingeschlossene Personen seine Arbeit hinterfragen. Vgl. IEEE/ANSI /IEEE Standard Glossary/ 37

praktikabel und wirtschaftlich.<sup>1)</sup> Sie können frühzeitig projektbegleitend durchgeführt werden,<sup>2)</sup> wodurch die Folgen von Fehlern in Grenzen gehalten werden. Bereits bei der Eingabe von Symbolen in Diagramme kann die Einhaltung von Konventionen durch interaktive Syntaxüberprüfungen erzwungen werden. Während die Befürworter dieser interaktiven Syntaxchecks den Entwickler so früh wie möglich vor falschen und sinnlosen Eingaben bewahren möchten,<sup>3)</sup> sehen ihre Gegner durch eine solche früh greifende Sicherungsmaßnahme den Entwurfscharakter der Diagrammerstellung in Frage gestellt<sup>4)</sup>. Zur Sicherung der Vollständigkeit, Konsistenz und Eindeutigkeit der Einträge aus den Diagrammen in die Entwicklungsdatenbank sollten die Diagrammeinträge mit den Repositoryinhalten durch Berichte abgeglichen werden können.<sup>5)</sup> Zur Beurteilung der Qualitätssicherungsfunktion in der SEU wird das Angebot an Berichten untersucht. Die Reports sollten sowohl einzelne Diagramme als auch die Konsistenz zwischen ihnen überprüfen. Die Qualitätssicherung soll phasenübergreifend eingesetzt werden,<sup>6)</sup> sie darf sich deshalb nicht lediglich auf den Entwurf beschränken. Von einer den gesamten Softwarelebenszyklus abdeckenden SEU ist auch eine Qualitätssicherung der codierten Programme zu fordern. Hier können die CASE-Produkte mit der Bereitstellung von Testwerkzeugen beispielsweise Testdatengeneratoren, der Simulationsmöglichkeit von Testumgebungen sowie Testauswertungs- und Testdokumentationswerkzeugen Unterstützung bieten.<sup>7)</sup>

#### 3.2.2.4 Prototyping

Zur Akzeptanzsicherung können die Benutzerschnittstellen durch Prototypen-Vorstellung vom zukünftigen Benutzer verifiziert werden. Somit besteht die Chance, Mißverständnisse noch vor der Codierung zu beseitigen.<sup>8)</sup> Dem

---

1) Vgl. Balzert /Entwicklung/ 451; Martin, McClure /Structured Techniques/ 145

2) Vgl. Balzert /Prinzipien/ 1, 6; Gewalt, Haake; Pfadler /Software Engineering/ 161

3) Vgl. DeMarco /Structured Analysis/ 158; McClure /Automation/ 43

4) Vgl. Crozier u. a. /Critical analysis/ 493; Curtis, Krasner, Iscoe /A Field Study/ 1278

5) Vgl. DeMarco /Structured Analysis/ 158; Martin, McClure /Structured Techniques/ 177

6) Vgl. Griese u. a. /Ergebnisse/ 523

7) Vgl. Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 122-125; Sneed /automatische Qualitätssicherung/ 87

8) Vgl. Fisher /CASE/ 30; Wedekind /Rapid Prototyping/ 58

Prototyping kommt jedoch nicht nur eine qualitätssichernde Funktion zu, es dient auch als Kommunikationsmedium der Einbindung des Benutzers in die Systementwicklung.<sup>1)</sup> Diese Funktion ist besonders in den Fällen hilfreich, in denen der Benutzer seine Anforderungen nicht präzise beschreiben kann oder diesbezüglich unsicher ist.<sup>2)</sup>

Generell unterscheidet man Prototypen, die bis zum vollständigen Anwendungsprodukt weiterentwickelt werden sollen, und solche, die lediglich als Kommunikationsmittel dienen.<sup>3)</sup> Letztere müssen schnell entwickelt und ebenso schnell geändert werden können.<sup>4)</sup> Ihre Erstellung darf nicht zu aufwendig sein, damit auch das Verwerfen eines Prototypen wirtschaftlich vertretbar ist.<sup>5)</sup> Inkrementelle Übersetzer bieten deshalb die Möglichkeit, bei Änderungen lediglich die modifizierten Teile des Prototypen zu übersetzen und diese in die bestehende Version zu integrieren.<sup>6)</sup> Die meisten Prototyper unterstützen lediglich den Benutzerdialog,<sup>7)</sup> andere nur die mathematischen Berechnungen des zu entwickelnden Systems. Bei der Beurteilung des Funktionsumfangs sind die darstellbaren Elemente zu vergleichen: werden nur Bildschirmmasken oder auch Berichte modelliert, welche Details des Bildschirmlayouts, beispielsweise Fenster, Pull-down-Menüs oder Zeigeoperationen, können dargestellt werden?<sup>8)</sup> Die meisten Maskengeneratoren liefern jedoch lediglich Oberflächen mit vorgegebenen Positionen für Texte und Ein- oder Ausgabefelder.<sup>9)</sup> Besonders geeignet sind Prototyper, die nicht nur Editoren zur Gestaltung der Benutzeroberfläche bereitstellen, sondern zusätzlich eine Simulation des Systemverhaltens bei Benutzereingaben demonstrieren können.<sup>10)</sup> Ein solcher Prototyp reagiert auf Benutzereingaben, kontrolliert die Korrektheit der Inputs und blendet bei Bedarf Hilfstexte ein.<sup>1)</sup> Zur Planung

- 
- 1) Vgl. Fisher /CASE/ 29; Koslowski /Unterstützung partizipativer Systementwicklung/ 167
  - 2) Vgl. Martin /Information Engineering, Book III/ 109; Tobiasch /Anforderungen/ 15
  - 3) Vgl. Martin /Information Engineering, Book III/ 110
  - 4) Vgl. Timmermann /Lisp-CASE/ 1106; Vonk /Prototyping/ 47, 54
  - 5) Vgl. Wedekind /Rapid Prototyping/ 58
  - 6) Vgl. Lektorat des B.I.-Wissenschaftsverlags /Duden Informatik/ 278; Hensel /Perspektiven/ o. S.
  - 7) Vgl. Vonk /Prototyping/ 139
  - 8) Vgl. Gibson /CASE Philosophy/ 216; Hoppe /Prototypenentwicklung/ 278
  - 9) Vgl. Gane /Computer-Aided Software Engineering/ 62
  - 10) Vgl. Österle /CASE/ 358-359

rektheit der Inputs und blendet bei Bedarf Hilfstexte ein.<sup>1)</sup> Zur Planung einer Abfolge mehrerer Masken sollten Diagramme verfügbar sein, die eine Navigationsroute des Dialogs abbilden.<sup>2)</sup>

### 3.2.2.5 Projektmanagement

Zur Planung von Ressourcen, Kosten und Terminen und zur Aufgabenverteilung unter den Entwicklern sollten, insbesondere bei der Abwicklung größerer Projekte, Projektmanagement-Werkzeuge zur Verfügung stehen.<sup>3)</sup> Die Planung ist vor allem vor dem Hintergrund juristischer Konsequenzen der Nicht-Einhaltung von Termin- und Kostenvereinbarungen zwischen Entwickler und Anwender bedeutsam. Daneben werden Werkzeuge für die Durchführung und Kontrolle von Projekten benötigt. So kann beispielsweise eine Wirtschaftlichkeitskontrolle vor Fehlinvestitionen, das Nicht-Erreichen eines Teilziels vor Termenschwierigkeiten warnen<sup>4)</sup>. Zur Ressourcenplanung sollten das Gesamtprojekt in Phasen und Aktivitäten unterteilt, Aufgaben den Teammitgliedern zugeordnet und Termine geplant werden.<sup>5)</sup> Grafische und automatisierte Hilfsmittel sollten den Projektmanager bei diesen Aufgaben unterstützen. So kann ein in die SEU integrierter Netzplan leicht den geänderten Bedürfnissen des Projekts angepaßt werden.<sup>6)</sup> Zur Ressourcenplanung und deren Dokumentation ist die Verfügbarkeit von Schätzinstrumenten wünschenswert.<sup>7)</sup> Die Planung eines Softwareprojekts steigt ihren vollen Nutzen erst dann, wenn der Projektfortschritt zur Kontrolle des Projektplans dokumentiert wird. Hierzu müssen Meilensteinberichte<sup>8)</sup> formuliert werden können sowie Kosten- und Projektfortschrittserfassungssysteme in der SEU vorhanden sein.<sup>9)</sup> Diese

---

1) Vgl. Fisher /CASE/ 127-128

2) Vgl. Crozier u. a. /Critical analysis/ 494; Hoppe /Prototypenentwicklung/ 282

3) Vgl. Curth, Wyss /Information engineering/ 221-222; Ruf /Basis des Schnittstellen-Management-Ansatzes/ 242

4) Vgl. Boehm /Software-Produktion/ 22

5) Vgl. Curth, Wyss /Information engineering/ 221; Gewalt, Haake, Pfadler /Software Engineering/ 234

6) Vgl. Hering /Software-Engineering 102; Stahlknecht, Warner /Untersuchung/ 28

7) Vgl. Seibt /COCOMO/ 59; Seibt /Function-Point-Methode: Vorgehensweise/ 3-9; Seibt /Function-Point-Methode/ 109

8) Meilensteinberichte dokumentieren den Zielerreichungsgrad der von der Projektplanung gesteckten Ziele.

9) Vgl. Hering /Software-Engineering/ 98; McMenamin, Palmer /Strukturierte

Aufgaben können teilweise durch Textverarbeitungssysteme, Tabellenkalkulationsprogramme und Audit Attributes<sup>1)</sup> in der SEU realisiert werden.<sup>2)</sup> Zum Schutz vor unberechtigten Änderungen der Entwicklungsergebnisse müssen ein Zugriffsschutz und Lock-Mechanismen<sup>3)</sup> vorhanden sein.<sup>4)</sup>

#### 3.2.2.6 Konfigurationsmanagement

Im Laufe des Software-Lebenszyklusses führen der Projektfortschritt, sich ändernde Umweltbedingungen oder anwenderspezifische Sonderwünsche zu neuen Versionen eines Softwaresystems.<sup>5)</sup> Die Verwaltung der verschiedenen Versionen, die Kontrolle sowie die Dokumentation der Änderungen einschließlich deren Gründe sind Aufgabe des Konfigurationsmanagement.<sup>6)</sup> Die SEU sollte zur Versionskontrolle mehrere Versionen eines Projekts gleichzeitig in der Entwicklungsdatenbank speichern können. Zum Nachvollziehen von Änderungen einschließlich deren Gründe und Verantwortlichen sind für die SEU spezielle Dokumentationsinstrumente zu fordern, die den Entwickler mit möglichst wenig Eingaben belasten.

#### 3.2.2.7 Allgemeine Bürodienste

Neben den Werkzeugen, die unmittelbar der Softwareentwicklung dienen, sollte die SEU auch allgemeine Bürodienste anbieten. So sollte jedes CASE-Produkt eine Textverarbeitung, Tabellenkalkulations, Kalenderfunktion und Electronic Mail integrieren oder zumindest eine Schnittstelle zu diesen Programmen besitzen.<sup>7)</sup>

---

#### Systemanalyse/ 33

- 1) Audit Attributes entsprechen Prüfungsfeldern, vgl. 3.2.2.1 Informationsverwaltung
- 2) Vgl. McClure /Automation/ 149
- 3) Möglichkeit, Dokumente vor den Änderungen durch Teamkollegen zu schützen.
- 4) Vgl. Curth, Wyss /Information engineering/ 203; Gane /Computer-Aided Software Engineering/ 90
- 5) Vgl. Hanschke, Kobialka, Timm /Dokumentverwaltung/ 495; Tobiasch /Anforderungen/ 19
- 6) Vgl. Gane /Computer-Aided Software Engineering/ 91; Sommerville /Software Engineering/ 140
- 7) Vgl. Boehm u. a. /Improving Productivity/ 32, Österle /Computer Integrated Software-Engineering/ 15



### 3.2.3 Gestaltungsorientierte Auswahlkriterien

Zur Auswahl von SEU reicht es nicht aus, nur die Funktionalität der Produkte zu untersuchen. Es muß auch die Gestaltung der Lösungsansätze berücksichtigt werden.

#### 3.2.3.1 Methoden

Durch die Anwendung von Methoden<sup>1)</sup> bemühen sich die Softwareentwickler, die Komplexität ihrer Aufgaben durch Abstraktion und Strukturierung zu bewältigen.<sup>2)</sup> Die Methodenunterstützung der CASE-Produkte reicht von der Bereitstellung der Diagrammtypen über eine Unterstützung durch Hilfehinweise oder Checklisten bis zur Einbettung der Methoden in einen Vorgehensplan.<sup>3)</sup> Die Beantwortung einer Grundsatzfrage stellt die Entscheidung dar, ob die SEU sich auf die Unterstützung bestimmter Methoden konzentrieren oder methodenneutral gestaltet sein soll. Die Befürworter einer Methodenneutralität argumentieren mit der Flexibilität, für jede Aufgabe eine adäquate Methode anbieten zu können. Außerdem bestehe die Möglichkeit, auch unternehmungseigene, in der Vergangenheit bewährte Methoden einzusetzen.<sup>4)</sup> Weiterhin kann eine methodenorientierte SEU durch die Entwicklung neuer Methoden in Frage gestellt werden.<sup>5)</sup> Für eine Methodenorientierung sprechen optimal auf einzelne Methoden zugeschnittene Werkzeuge, der höhere Unterstützungsgrad und effektivere Qualitätssicherungsmaßnahmen.<sup>6)</sup> Eine zahlreiche Methoden unterstützende SEU realisiert die Vielfalt entweder durch die Bereitstellung von die einzelnen Methoden nur sehr allgemein unterstützenden Werkzeugen oder durch eine hohe Anzahl von auf einzelne Methoden abgestimmten Werkzeugen. Im ersten Fall bezahlt der Entwickler die Flexibilität mit einer nur geringen Methodenunterstützung, im zweiten er-

- 
- 1) Methoden werden definiert als die "Gesamtheit aller Vorschriften zur Bewältigung einer Klasse von Problemen" und als "planmäßig angewendete, begründete Vorgehensweise" zur Zielerreichung. Vgl. Balzert /Entwicklung/ 22; Schmitz /Methoden, Verfahren und Werkzeuge/ 73; Wirtz /Softwareentwurf/ 325
  - 2) Vgl. Fisher /CASE/ 32; Schulz /Software-Entwurf/ 28
  - 3) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43
  - 4) Vgl. Fisher /CASE/ 226; Tobiasch /Anforderungen/ 25
  - 5) Vgl. Balzert /Entwicklung/ 486; Leppert, Stork /Entwicklungsumgebungen/ 209
  - 6) Vgl. Balzert /Entwicklung/ 486; McClure /Automation/ 81

hält er eine überladene SEU mit Werkzeugen, die er eventuell niemals einsetzen wird.<sup>1)</sup> Das Argument der fehlenden Flexibilität von methodenorientierten SEU verliert an Überzeugungskraft, wenn man berücksichtigt, daß der einzelne Entwickler i. d. R. höchstens eine Methode beherrscht und zu einem Methodenvergleich hinsichtlich der Eignung zur Lösung einer konkreten Aufgabe nur selten in der Lage ist.<sup>2)</sup> Er kommt somit nur in Ausnahmefällen in den Genuß einer angebotenen Methodenvielfalt.

Bei der Entscheidung für eine SEU sollte das Angebot der unterstützten Methoden ein großes Gewicht erhalten; die Wahl der richtigen Methode ist wichtiger als beispielsweise die Performance eines CASE-Produkts.<sup>3)</sup> Es existieren eine Vielzahl von Methoden, eine universelle hat sich jedoch noch nicht herausgebildet.<sup>4)</sup>

Um mehrere Sichtweisen des Projekts zu ermöglichen, bieten viele SEU einen Methodenverbund, der eine Modellierung von Datenflüssen, Datenstrukturen und eine prozeßorientierte Gestaltung gestattet.<sup>5)</sup> Aus Gründen der Flexibilität kann es sinnvoll sein, mehrere Methoden und Werkzeuge zur Bewältigung derselben Aufgabe anzubieten, um dem Entwickler eine Auswahl unter Berücksichtigung seiner Fähigkeiten und Kenntnisse anzubieten.<sup>6)</sup> Im folgenden soll der Unterstützungsgrad von drei verbreiteten Methoden insbesondere im Hinblick auf die Bereitstellung der Diagrammtypen untersucht werden: Structured Analysis, Entity-Relationship Modelling und Structured Design. Die Unterstützung weiterer Methoden wird nur im Fragebogen überprüft.

### Structured Analysis

Eine der sowohl in den USA als auch in Europa verbreitetsten Methoden ist Structured Analysis in der Notation von DeMarco oder von Gane und Sarson;<sup>7)</sup> sie wird von den meisten SEU unterstützt.<sup>8)</sup> Diese Methode strukturiert

- 
- 1) Vgl. Ruf /Basis des Schnittstellen-Management-Ansatzes/ 95
  - 2) Vgl. McClure /Automation/ 81
  - 3) Vgl. Maag /Methoden/ 28
  - 4) Vgl. Stobbe /Softwareentwicklungsumgebungen/ 32; Wallmüller /Methode und Werkzeug/ 424
  - 5) Vgl. Fisher /CASE/ 218; Warner, Woschinski /Qual der Wahl/ 58
  - 6) Vgl. Schmitz /Methoden, Verfahren und Werkzeuge/ 77
  - 7) Vgl. McClure /Automation/ 91; Prakash /Using CASE/ 80
  - 8) Vgl. Fisher /CASE/ 217

das Gesamtsystem unter funktionalen Gesichtspunkten mit dem Ziel, eine prägnante und redundanzfreie Spezifikation zu erreichen.<sup>1)</sup> In den ersten Schritten der Methode soll ein logisches<sup>2)</sup> Modell des neuen Systems erstellt werden.<sup>3)</sup> Während DeMarco dieses durch die Abbildung des bestehenden physischen Modells erreicht, aus dem er das logische Ist-Modell ableitet, um anschließend zum logischen Soll-Modell zu gelangen, verlangen Gane und Sarson die Beschreibung des physischen Ist-Modells nicht. Sie leiten das logische Soll-Modell direkt aus dem logischen Ist-Modell ab.<sup>4)</sup>

Zur Darstellung der generellen Regeln des zu entwickelnden Systems bedient man sich einer datenfluß-orientierten Darstellungsform, die keine technischen Komponenten enthält und wegen der Vertrautheit des Blickwinkels sowie der geringen Abstraktion vom Benutzer leicht nachvollziehbar ist.<sup>5)</sup> Datenflußdiagramme sind sowohl für die Modellierung eines Überblicks als auch für die Darstellung von Details geeignet.<sup>6)</sup> Die Darstellungsformen des Datenflußdiagramms sind im wesentlichen auf vier Symbole reduziert: Prozeß, External Entity, Datenspeicher und Datenfluß, deren Aussehen in den verschiedenen Notationen<sup>7)</sup> differiert.<sup>8)</sup> Um die Übersichtlichkeit der Diagramme zu gewährleisten,<sup>9)</sup> sollten die Prozesse durch ein hierarchisch untergeordnetes Datenflußdiagramm verfeinert werden können.<sup>10)</sup> Hierbei kann eine grafische Hervorhebung der verfeinerten Prozesse und die automatisierte Numerierung

- 
- 1) Vgl. DeMarco /Structured Analysis/ 16; Fisher /CASE/ 60
  - 2) Der Begriff logisch wird in der Softwareentwicklung gebraucht, wenn die zugrundeliegende Essenz eines Systems im Gegensatz zu ihrer physischen Realisation angesprochen wird. Vgl. Gane /Computer-Aided Software Engineering/ 6
  - 3) Vgl. DeMarco /Structured Analysis/ 16
  - 4) Vgl. Martin, McClure /Structured Techniques/ 413
  - 5) Vgl. Gane /Computer-Aided Software Engineering/ 5; Schulz /Software-Entwurf/ 82
  - 6) Vgl. Martin, McClure /Structured Techniques/ 152
  - 7) In der Structured Analysis haben sich die Notationen von DeMarco und die von Gane und Sarson durchgesetzt.
  - 8) Vgl. DeMarco /Structured Analysis/ 40; Gane, Sarson /Structured Systems Analysis/ 38-45
  - 9) Es wird empfohlen, in einem Diagramm möglichst nicht mehr als 7 Symbole darzustellen. Vgl. DeMarco /Structured Analysis/ 82 oder McMenamin, Palmer /Strukturierte Systemanalyse/ 60
  - 10) Vgl. Fisher /CASE/ 65; Gane, Sarson /Structured Systems Analysis/ 46

die hierarchische Struktur verdeutlichen.<sup>1)</sup> Die SEU sollte gewährleisten, daß die Änderungen in einem Datenflußdiagramm auf detaillierter Ebene in den hierarchisch höheren Diagrammen fortgeschrieben werden.<sup>2)</sup> Zum Zwecke der Ableitung des logischen Soll-Modells aus der bestehenden Systembeschreibung kommt der visuellen Hervorhebung von Symbolen und dem Komfort bei Änderungen, Hinzufügungen und Löschungen von Symbolen eine besondere Bedeutung zu. Ausführliche Kontrollen sollten dafür sorgen, daß die methodenspezifischen Regeln bei der Modellierung eingehalten werden.<sup>3)</sup> Um die Datenflußdiagramme nicht mit Informationen zu überladen, werden Informationen in der Entwicklungsdatenbank dargestellt.<sup>4)</sup> Datenlexika sollten die Beschreibungen der nicht weiter zu zerlegenden Prozesse und der Diagrammsymbole ermöglichen.<sup>5)</sup> Durch Kontrollberichte sollte die Beschreibung aller Symbole in der Entwicklungsdatenbank gesichert werden.<sup>6)</sup> Datenflußdiagramme sind zur Darstellung von Prozeßstrukturen nicht geeignet, hierzu sollte das CASE-Produkt geeignete Werkzeuge, beispielsweise Structure Charts oder Strukturdiagramme, bereitstellen.<sup>7)</sup> Für eine detaillierte Funktionsbeschreibung, auch als MiniSpec bezeichnet, sind Entscheidungstabellen, Entscheidungsbäume, strukturierte Sprache oder Pseudocode zur Verfügung zu stellen.<sup>8)</sup>

Im Gegensatz zu DeMarco binden Gane und Sarson in das logische Modell ein Datenmodell ein, das den Inhalt der Datenspeicher in der dritten Normalform darstellt.<sup>9)</sup> Anschließend soll eine physische Datenbasis entworfen werden.<sup>10)</sup> DeMarco verzichtet auf die Datenmodellierung. Er unterscheidet sich von Gane und Sarson durch die Entwicklung mehrerer physischer Soll-

- 
- 1) Vgl. DeMarco /Structured Analysis/ 80; Gane, Sarson /Structured Systems Analysis/ 46
  - 2) Vgl. DeMarco /Structured Analysis/ 78; Martin, McClure /Structured Techniques/ 163
  - 3) Vgl. Fisher /CASE/ 69; Scheibl /Kommerzielle Software- Entwicklung/ 331
  - 4) Vgl. DeMarco /Structured Analysis/ 16; McMenamin, Palmer /Strukturierte Systemanalyse/ 5
  - 5) Vgl. Gane, Sarson /Structured Systems Analysis/ 75-90; Schulz /Software-Entwurf/ 87-88
  - 6) Vgl. DeMarco /Structured Analysis/ 106
  - 7) Vgl. Martin, McClure /Structured Techniques/ 163
  - 8) Vgl. DeMarco /Structured Analysis/ 16, 85; McMenamin, Palmer /Strukturierte Systemanalyse/ 66
  - 9) Vgl. Gane, Sarson /Structured Systems Analysis/ 169-197
  - 10) Vgl. Martin, McClure /Structured Techniques/ 413

Modelle, die mit Hilfe von Analysen ausgewertet werden, um schließlich die geeignetste Alternative zu wählen.<sup>1)</sup>

### Entity-Relationship Ansatz

Das Entity-Relationship Modell dient dem logischen Datenbankentwurf. Aufgrund der Unabhängigkeit von Datenspeichern und Zugriffsmethoden sowie der grafischen Beschreibung ist dieses Modell einfach zu verstehen<sup>2)</sup> und somit für die Einbeziehung des Benutzers in die Softwareentwicklung geeignet.<sup>3)</sup> In einem ersten Schritt werden die Unternehmungsdaten in einem Diagramm abgebildet, das Entity-Relationship Modell entsteht.<sup>4)</sup> Hierbei werden die für das zu entwickelnde Softwaresystem interessanten Objekte als Entities abgebildet, Beziehungstypen und Schlüssel festgelegt.

Der Grafikeditor sollte die benötigten Symbole u. a. Entity, Beziehung, Schlüssel und Domäne zur Verfügung stellen. Eine Möglichkeit zur Darstellung von Entity-Sub-Types oder Entity-Super-Types ist zur Unterstützung des Top-Down Prinzips wünschenswert.<sup>5)</sup> Anschließend werden die Kardinalitäten und Abhängigkeiten festgelegt.<sup>6)</sup> Der Editor sollte deshalb die Kardinalität der Beziehungen und die Abhängigkeit der Entities illustrieren. Zusätzlich ist die Darstellung der Domänen<sup>7)</sup> im Diagramm von Interesse. Nach Fertigstellung des Entity-Relationship Modells wird dieses in ein Datenstrukturmodell transferiert; die Unterstützung hierarchischer Modelle, Netzwerke oder Relationenmodelle ist zu fordern.<sup>8)</sup> Ein wichtiges Kriterium ist die Unterstützung bei der Übersetzung des konzeptionellen Schemas in logische Datenstrukturen. Zur Vermeidung von Redundanzen und Anomalien und somit zur Wahrung der Integrität der gespeicherten Informationen, werden diese in mehrere einfache Einheiten zerlegt.<sup>9)</sup> Zur Lösung dieser Aufgabe sollten die Regeln zur Norma-

---

1) Vgl. Martin, McClure /Structured Techniques/ 413

2) Vgl. Barker /CASE\*Method/ 1-1, 2-13

3) Vgl. Scheer /Wirtschaftsinformatik/ 20

4) Vgl. Chen /Entity-Relationship Approach/ 9; Orr u. a. /Methodology/ 231

5) Vgl. Barker /CASE\*Method/ 6-4; Scheer /Wirtschaftsinformatik/ 17-19

6) Vgl. Orr u. a. /Methodology/ 231

7) Vgl. Chen /Entity-Relationship Approach/ 20; Vetter /Aufbau betriebswirtschaftlicher Informationssysteme/ 45

8) Vgl. Chen /Entity-Relationship Approach/ 6-8

9) Vgl. Scheer /Wirtschaftsinformatik/ 29; Vetter /Aufbau betrieblicher Informationssysteme/ 129

lisierung der Daten angewendet werden können.<sup>1)</sup> Die Normalisierung der Daten ist in der Praxis umstritten, denn eine vollständig normalisierte Datenbank besitzt mehr Datensätze, benötigt somit mehr Bearbeitungszeit.<sup>2)</sup> Andererseits reduziert sie infolge der abgebauten Redundanz den Speicherbedarf und erleichtert die Kommunikation mit Datenmengen unterschiedlicher Datenstruktur.<sup>3)</sup> Deshalb wird häufig die Normalisierung nur auf logischer Ebene bis zur dritten Normalform durchgeführt, physisch jedoch werden die Daten in der ersten Normalform implementiert. Während eine vollständig automatisierte Normalisierung für CASE-Produkte nicht realisiert ist,<sup>4)</sup> können diese Umgebungen den Normalisierungsprozeß durch interaktive Benutzerführung des Entwicklers und durch Kontrollen unterstützen.

Auch beim Entity-Relationship Ansatz existieren verschiedene Notationen; die bekanntesten sind die von Bachman und Chen.<sup>5)</sup>

#### Structured Design

Diese Methode stellt Regeln zur Modularisierung von Programmen zur Verfügung.<sup>6)</sup> In einem ersten Schritt sollen die Problemkomponenten in einem Datenflußdiagramm abgeleitet werden.

Das Datenflußdiagramm sollte daher neben den Symbolen zur Modellierung des Datenflusses auch logische Verzweigungen darstellen können.<sup>7)</sup> Um die Transformation des Datenflußmodells in ein Structure Chart zu erleichtern, sollten Eingabe-, Verarbeitungs- und Ausgabekomponenten im Diagramm grafisch getrennt werden können.<sup>8)</sup> In der Transformationsanalyse erfolgt die Umsetzung der Problemkomponenten in hierarchisch angeordnete funktionale Bestandteile. Es gibt Werkzeuge zur automatisierten Transformation; die SEU sollte zumindest die vollständige Erfassung des Datenflußdiagramms im

- 
- 1) Vgl. Scheer /Wirtschaftsinformatik/ 30; Vetter /Aufbau betrieblicher Informationssysteme/ 123-173
  - 2) Vgl. Martin /Information Engineering, Book II/ 234
  - 3) Vgl. Codd /Relational Model of Data/ 381
  - 4) Vgl. Gane /Computer-Aided Software Engineering/ 60
  - 5) Vgl. Martin, McClure /Structured Techniques/ 315-324
  - 6) Vgl. Schulz /Software-Entwurf/ 82
  - 7) Vgl. Yourdon, Constantine /Structured Design/ 175
  - 8) Vgl. Martin, McClure /Structured Techniques/ 443; Yourdon, Constantine /Structured Design/ 175-176

Structure Chart überprüfen können.<sup>1)</sup> Um die Komplexität der hierarchischen Struktur zu minimieren, sind Werkzeuge zu fordern, die helfen, die Modulkopplung zu minimieren und die Strukturiertheit innerhalb der Module zu maximieren. Ziel des Structured Design sollte die Programmzerlegung in kleine, einfach verbundene und unabhängige Module sein,<sup>2)</sup> damit die Implementierungs-, Pflege- und Wartungskosten niedrig gehalten werden können<sup>3)</sup>. Die Abbildung der Programmhierarchie in einem Structure Chart sollte die elementaren Symbole und Kontrollstrukturen sowie Erweiterungen wie wiederverwendbare Module, das Ansprechen von Modulteilern, rekursive und homologe Strukturen darstellen können.<sup>4)</sup> Unterstützung ist auch bei der Zusammenfassung der logischen Verarbeitungsschritte in physikalische Einheiten wünschenswert.<sup>5)</sup>

### 3.2.3.2 Kapazität

Insbesondere bei der Unterstützung von größeren Projekten wird die Frage der Kapazität bedeutungsvoll. Gemessen wird sie an der maximal möglichen Anzahl der in der Entwicklungsdatenbank speicherbaren Objekte innerhalb eines Projekts. Weitere Vergleichskennzahlen sind die Anzahl der möglichen Diagramme und die Anzahl der in diesen darstellbaren Symbole.<sup>6)</sup>

### 3.2.3.3 Outputqualität

Eine hohe Outputqualität wirkt sich positiv auf die Motivation der Mitarbeiter aus und präsentiert dem Anwender die Arbeitsergebnisse in professioneller Weise.<sup>7)</sup> Zur Beurteilung der Outputqualität wird die Möglichkeit beurteilt, gezoomte Bilder auszudrucken.<sup>8)</sup> Desweiteren wird untersucht, welche Hard-

---

1) Vgl. Martin, McClure /Structured Techniques/ 443

2) Vgl. Yourdon, Constantine /Structured Design/ 26; Schulz /Software-Entwurf/ 82

3) Vgl. Yourdon, Constantine /Structured Design/ 26; Schulz /Software-Entwurf/ 64, 81

4) Vgl. Schulz /Software-Entwurf/ 74-75, 77; Yourdon, Constantine /Structured Design/ 291, 396-402

5) Vgl. Martin, McClure /Structured Techniques/ 435

6) Vgl. McClure /Automation/ 39

7) Vgl. McDonnell Douglas /Product Description/ 5

8) Vgl. Martin, McClure /Structured Techniques/ 145

warevoraussetzungen für den Ausdruck der Grafiken erforderlich sind.<sup>1)</sup> Bewertet wird auch die Möglichkeit, Schriftzüge quer zu drucken sowie Schrift- und Symbolgrößen zu bestimmen und Umlaute zu auszugeben.<sup>2)</sup>

#### 3.2.3.4 Integration

Weil ein Werkzeug allein die Aufgaben der Software-Entwicklung nicht lösen kann, werden in Software-Entwicklungsumgebungen mehrere Werkzeuge eingebunden.<sup>3)</sup> Um Fehler und Zeitverzögerungen bei Transformationsprozessen zwischen den einzelnen Werkzeugen zu vermeiden, wird eine Integration der einzelnen Werkzeuge gefordert.<sup>4)</sup> Die Werkzeuge müssen so aufeinander abgestimmt sein, daß der Output eines Tools von einem anderen Werkzeug als Eingabe verwendet werden kann.<sup>5)</sup> Damit nicht sämtliche Werkzeuge verändert werden müssen, ist es sinnvoll, deren Eingabe und Ausgabe über ein Kommunikationsmedium zu verbinden: die Entwicklungsdatenbank.<sup>6)</sup> Zur Beurteilung des Integrationsgrads wird untersucht, inwieweit mehrere Diagramme auf einem Bildschirm dargestellt werden können und inwieweit aus einem Editor in einen anderen gesprungen werden kann. Für die Softwareentwicklung von Bedeutung ist die Frage, ob aus der Entwicklungsdatenbank Informationen in Beziehung gesetzt werden können, die mit verschiedenen Werkzeugen gewonnen wurden.<sup>7)</sup> Desweiteren ist eine logische Integration der Werkzeuge zu fordern, d.h. die Werkzeuge sollten methodisch aufeinander abgestimmt sein, sich zumindest nicht widersprechen.<sup>8)</sup> Die SEU sollte mehrere PC miteinander verbinden können oder multiuserfähig sein. Eventuell. sollte eine gemeinsame Entwicklungsdatenhaltung auf einem Großrechner unterstützt werden.<sup>9)</sup>

---

1) Vgl. Martin, McClure /Structured Techniques/ 138

2) Vgl. Martin, McClure /Structured Techniques/ 125

3) Vgl. Ruf /Basis des Schnittstellen-Management-Ansatzes/ 17

4) Vgl. Balzert /Anforderungen/ 105; Schulz /Klassifizierungs- und Bewertungsschema/ 193

5) Vgl. Schmitz /Methoden, Verfahren und Werkzeuge/ 77; Stobbe /Softwareentwicklungsumgebungen/ 39

6) Vgl. Stobbe /Softwareentwicklungsumgebungen/ 40; Stork /Toolintegration/ 51

7) Vgl. Martin /Information Engineering, Book II/ 171

8) Vgl. Ruf /Basis des Schnittstellen-Management-Ansatzes/ 96

9) Vgl. McClure /Automation/ 125; Tobiasch /Anforderungen/ 19



SEU legen häufig einen Schwerpunkt auf die Unterstützung einer oder mehrerer Phasen des Software-Lebenszyklusses, die restlichen Phasen werden vernachlässigt. Die Literatur unterscheidet entsprechend des Unterstützungsschwerpunkts der SEU in Upper-, Middle- und Lower-CASE-Umgebungen.<sup>1)</sup> Die Integration einzelner SEU in den Gesamtprozeß der Softwareentwicklung soll durch das Anwendungsentwicklungskonzept der IBM erreicht werden.<sup>2)</sup> Im September 1989 kündigte die IBM ein standardisiertes, offenes Rahmenwerk zur Anwendungsentwicklung<sup>3)</sup> auf der Basis der IBM System-Anwendungs-Architektur (SAA)<sup>4)</sup> an: den Application Development Cycle (AD/Cycle).<sup>5)</sup> Jede Phase der Softwareentwicklung wird durch Produkte entweder von IBM, anderen Softwareanbietern oder vom Kunden selbst unterstützt. Die Integration erfolgt über eine Anwendungsentwicklungsplattform, in der die Richtlinien für die Benutzerschnittstellen, die Regelungen für die Integration, die Kommunikationsdienste zwischen PC und Großrechner und die Steuerung des Datenaustauschs mit dem zentralen, auf einem Großrechner implementierten Repository realisiert werden.<sup>6)</sup> Obwohl der AD/Cycle heute noch eher eine Vision, ein Plan oder ein großes Versprechen<sup>7)</sup> darstellt, jedoch noch keine verfügbare Technologie, ist die zukünftige Integrierbarkeit in den AD/Cycle wegen der hohen Durchsetzungswahrscheinlichkeit des IBM-Konzepts<sup>8)</sup> eine entscheidene Eigenschaft einer SEU. Als Indikator dient die Stellung des CASE-Anbieters zur IBM in Hinblick auf den AD/Cycle.

#### 4 Beschreibung der ausgewählten Software-Entwicklungsumgebungen

Nach Sichtung und Eingrenzung des Marktes kamen folgende Produkte in die Feinauswahl:

- 
- 1) Vgl. Gibson /CASE Philosophy/ 209
  - 2) Vgl. IBM /Neu von IBM/ o. S.; Merlyn, Boone /AD/Cycle/ 59
  - 3) Vgl. Merlyn, Boone /AD/Cycle/ 59; Richards /AD/Cycle/ 154
  - 4) Vgl. IBM /Neu von IBM/ o. S.
  - 5) Vgl. Fersko-Weiß /CASE-Tools/ 214
  - 6) Vgl. IBM /Neu von IBM/ o. S.
  - 7) Vgl. Hamilton /No easy answers/ o. S.
  - 8) Vgl. Merlyn, Boone /AD/Cycle/ 59

	<b>Exceleator</b>	<b>Information Engineering Workbench</b>	<b>ProKit* WORK-BENCH</b>	<b>Systems Engineer</b>
<b>Hersteller</b>	Index Technology	Knowledge Ware	McDonnell Douglas	LBMS
<b>Vertreiber</b>	Exceleator Deutschland	Ernst & Young CASE Services	McDonnell Douglas	ExperTeam
<b>untersuchte Version</b>	IS / 1.9	5.020C	2.0	2.1
<b>zugrundeliegende Methoden bzw. herstellereigene Vorgehensweise</b>	methoden-unabhängig; keine eigene Vorgehensweise	Information Engineering nach James Martin; eigene Vorgehensweise Navigator	Structured System Analysis and Design nach Gane und Sarson; eigene Vorgehensweise STRADIS	methoden-unabhängig; eigene Vorgehensweise Systems Engineering

Abb. 2: Kurzbeschreibung Exceleator, Information Engineering Workbench, ProKit\*WORKBENCH, Systems Engineer

#### 4.1 Exceleator

Der Exceleator der Index Technology Corporation ist seit 1984 auf dem Markt und ist aus der Weiterentwicklung einer reinen Zeichenhilfe<sup>1)</sup> entstanden.<sup>2)</sup> Er besteht aus sieben Hauptkomponenten: Graphics, XLDictionary, Screens & Reports, Documentation, Analysis, XLD Interface und Housekeeping. Mit den Grafikerzeugnissen wird die Modellierung von Diagrammen unterstützt. Hier ermöglicht eine Palette von Diagrammeditoren unterschiedliche Sichtweisen auf das zu erstellende Softwareprodukt. Im einzelnen werden Datenflußdiagramme, Structure Charts, Datenstrukturdiagramme, Entity-Relationship-Diagramme, State Transition Diagramme, Strukturdiagramme, Präsentationsdiagramme und Work Breakdown Strukturen unterstützt. Im XLDictionary, der

1) Vgl. Österle /Computer Integrated Software-Engineering/ 13

2) Neben der hier geprüften Exceleator/IS-Version für kommerzielle Anwendungen wird noch eine spezielle Variante für Echtzeitprobleme namens Exceleator/RTS angeboten.

Entwicklungsdatenbank, werden sämtliche Entwicklungsdaten gespeichert und aktualisiert. Desweiteren können diese Informationen ausgewertet und kontrolliert werden. Dokumentationshilfen ermöglichen die Dokumentationsstrukturierung. In einem Masken- und Berichtsgestalter können die Benutzerschnittstellen, Bildschirmmasken und Berichtslayouts entworfen und demonstriert werden. Die Korrektheit des Entwurfs und die Einhaltung methodenspezifischer Regeln wird im Analysis-Modul untersucht. Das Data Dictionary Interface bietet die nötigen Schnittstellen, um anderen Entwicklern einen Zugriff auf das Projekt zu gewähren und um komplette Projektversionen zu exportieren oder importieren. Projektorganisation und Systemkonfiguration werden im Housekeeping-Modul durchgeführt.

#### 4.2 Information Engineering Workbench

Die Information Engineering Workbench von KnowledgeWare besteht aus einem Set von CASE-Tools, die auch einzeln erworben werden können. Die Planning Workstation dient der Unterstützung der strategischen Unternehmensplanung, die Analysis Workstation stellt Werkzeuge für die Entwicklung der logischen Systemspezifikation zur Verfügung, die Design Workstation bietet Hilfestellungen für den Schritt von der logischen Repräsentation zur physischen Ausführung und mit der Construction Workstation können COBOL-Source Code, Datenbankschemata, Datenbankdefinitionen, Dateizugriffe, Bildschirmmasken sowie Dokumentationen generiert werden. Obwohl die Analysis und Design Workstation bereits seit 1984 auf dem Markt angeboten werden, gibt es die Information Engineering Workbench erst seit 1989 als integrierte Lösung.<sup>1)</sup> Die Construction Workstation nimmt eine Sonderstellung ein, weil sie nicht wie die anderen Workstations auf der Benutzeroberfläche GEM, sondern auf der von Windows (2.11) basiert. Kernstück der Information Engineering Workbench ist die sogenannte Enzyklopädie mit ihrem Knowledge Coordinator, die für die horizontale und vertikale Integration der einzelnen Workstations sorgt. Die Information Engineering Workbench unterstützt eine Vielzahl von Diagrammtechniken wie z. B. Association Matrix, Property Matrix, Decomposition Diagram, Entity Relationship Diagram, Data Flow Diagram, Action Diagram, Structure Chart Diagram, Data Base Diagram, Data Structure Diagram und Minispezifikation.

---

1) Seit 1990 ist in Deutschland auch die Application Development Workbench (ADW), eine leicht verbesserte Version unter OS/2 Presentation Manager, erhältlich.

### 4.3 ProKit\*WORKBENCH

Die ProKit\*WORKBENCH von McDonnell Douglas ist zunächst für den Eigenbedarf des Anbieters entwickelt worden, sie wird seit 1987 auf dem Markt angeboten. Das Produkt besteht aus den Modulen: Analyser, Data Modeler, Designer, Prototyper, Repository, Reporting, Projekt, Version, Import/Export, Backup, Interface, DOS und Help. Die SEU ist auf die Structured Analysis Methode von Gane und Sarson abgestimmt. So unterstützt der Analyser ein Datenflußdiagramm in der Notation von Gane und Sarson. Der Data Modeler bietet ein Entity-Relationship Modell. Im Prototyper lassen sich Menü-, Bildschirm- und Reportmasken darstellen und die Reaktion des Systems auf Benutzereingaben demonstrieren. Die Illustration des Datentransfers und des Kontrollflusses kann im Designer durch ein Structure Chart erfolgen. Die Entwicklungsinformationen werden zentral im Repository gehalten, das entsprechend der oben beschriebenen Entwurfsmodule strukturiert ist. Ein Report-Modul ermöglicht eine Auflistung der Inhalte des Repository. Für die Projekt- und Versionsverwaltung stehen ebenfalls Module zur Verfügung. Die ProKit\*WORKBENCH stellt verschiedene Utility-Funktionen zur Verfügung, so unterstützt das Import/Export-Modul den Transfer des Repository, das Backup-Modul ermöglicht ein schnelles Archivieren von Projekten und Versionen. Die Module Interface und DOS erlauben die Ausführung von externen Programmen. Daneben verfügt die SEU über eine ausführliche Online-Hilfefunktion.

### 4.4 Systems Engineer

Systems Engineer von LBMS ist eine recht junge SEU, die erst seit 1990 auf dem deutschen Markt erhältlich ist und auf einem früheren CASE-Produkt von LBMS, Auto-Mate Plus, basiert. Die wichtigsten Veränderungen zu diesem Vorgängerprodukt ergeben sich durch die Wahl von Windows 3.0 als Umgebungssoftware. Systems Engineer wurde speziell für eine Multiuser-Umgebung in Client-/Server-Architektur konzipiert und bietet somit mehreren Entwicklern einen gleichzeitigen Zugriff auf die unmittelbar aktuell gehaltene Projektdatenbank SQLBASE.<sup>1)</sup> Schwerpunkte bei der Methodenunterstützung bilden Datenmodellierung (Entity-Relationship Ansatz), Datenflußdiagramm, Funktions- und Prozeßmodellierung (Pseudocode, Dialogdesign, Prototyping etc.) und Projektdokumentation (Anforderungen, Lösungen, Präsentationsgrafiken).

---

1) Darüberhinaus ist jedoch auch eine Singleuser-Version verfügbar.

## 5 Bewertung der ausgewählten Software-Entwicklungsumgebungen

### 5.1 Bewertung anhand genereller Kriterien für die Auswahl von Standardsoftware

Im Vergleich zu seinen Konkurrenzprodukten sind die Systemanforderungen<sup>1)</sup> des Excelerator bescheiden: er kommt in sehr kleinen Projekten ohne Speichererweiterung aus, benötigt lediglich 10 MB des Festplatten- und 459 KB des konventionellen Hauptspeichers. Die Prokit\*WORKBENCH stellt wesentlich höhere Anforderungen an die Konfiguration: sie braucht mindestens 512 KB Erweiterungsspeicher und nimmt ca. 26 MB des externen Speichers für sich in Anspruch. Besonders kritisch sind die beanspruchten 570 KB im konventionellen Hauptspeicher. Wer eine große Festplatte verwalten möchte, ohne diese partitionieren zu müssen, und deshalb DOS 4 oder höher nutzt, stößt auf Engpässe im konventionellen Speicher. Das Arbeiten mit einer DOS-Version 3.3 oder kleiner wird empfohlen. Bei der Installierung des Systems Engineer müssen mindestens 15 MB auf der Festplatte verfügbar sein, allerdings werden tatsächlich nur 10 MB belegt. Zwar ist der Systems Engineer auch mit weniger als den empfohlenen 4 MB Hauptspeicher lauffähig, er zeigt in diesem Fall jedoch erhebliche Performance-Schwächen. Die Hersteller der Information Engineering Workbench raten zu 8 MB Hauptspeicher, gleichwohl werden auch mit 5 MB zufriedenstellende Ergebnisse erzielt. Der Bedarf an externem Speicher ist mit dem der ProKit\*WORKBENCH und des Systems Engineer vergleichbar. Das durchgeführte Fallbeispiel benötigte, auf der ProKit\*WORKBENCH, auf der Information Engineering Workbench und auf dem Systems Engineer entwickelt, wesentlich mehr Speicherplatz als das gleiche Projekt im Unterverzeichnis des Excelerator. Zur Zeit bietet nur KnowledgeWare mit ADW eine OS/2-Version an. Die anderen SEU sind nur auf dem DOS-Betriebssystem lauffähig; Index Technology kündigte ein OS/2-basiertes Release an, McDonnell Douglas und LBMS planen ebenfalls eine OS/2-Version. Alle SEU unterstützen eine Reihe von Druckern und Bildschirmen; beim Excelerator fällt auf, daß er keine CGA-Karte benutzen kann. Dem Systems Engineer kommt hier die Bandbreite von Windows 3.0 zugute.

Als die SEU mit dem geringeren Speicherplatzbedarf ist der Excelerator wesentlich schneller zu installieren als seine Kontrahenten. Ist dieses CASE-Produkt nach 7 Minuten auf den externen Speicher kopiert, so muß man für

---

1) Die empfohlenen Werte bezüglich der Systemanforderungen beruhen auf Herstellerangaben.

die Installation der ProKit\*WORKBENCH eine halbe Stunde einkalkulieren und für die Information Engineering Workbench etwa weitere 10 Minuten. Dafür weist die ProKit\*WORKBENCH einen höheren Installationskomfort auf, beispielsweise eine sofortige Eingabekontrolle in der Installationsprozedur sowie eine abschließende Installationskontrolle. Besonders vorteilhaft ist die Möglichkeit, nicht benötigte Teile der Umgebung wieder zu deinstallieren; eine Möglichkeit, die zum Beispiel bei der Information Engineering Workbench völlig fehlt. Systems Engineer erlaubt dem Benutzer über ein Menü nur ausgewählte Teile der SEU zu installieren bzw. zu deinstallieren.

In allen SEU, mit Ausnahme des Systems Engineer, kann das Profil der Diagramme gestaltet werden: Farben, Textgrößen, und Label-Darstellung lassen sich auswählen. Die Information Engineering Workbench und der Systems Engineer bieten darüberhinaus durch ihre Fenstertechnik dem Benutzer die Gelegenheit, die Oberfläche nach seinen individuellen Bedürfnissen zu gestalten. Die ProKit\*WORKBENCH verfügt mit der Extend Repository-Funktion über die Möglichkeit, zusätzliche Attribute in das Metadatenmodell der SEU einzubauen, und diese zusätzlichen Attribute zu modifizieren. Eine Änderbarkeit der bestehenden Attribute der Metadatenstruktur wird nicht unterstützt. Diese Gestaltungsmöglichkeit bietet der Excelsator zwar nicht direkt, jedoch bietet Index Technology über ein Zusatzprodukt, den Customizer, die Gelegenheit, die Metadatenstruktur oder die Diagrammsymbole der SEU nach individuellen Bedürfnissen zu gestalten. Der Excelsator-Kunde kann somit seine SEU anpassen, indem er Objekte in ihr Repository aufnimmt, Diagramme durch die Einführung neuer Symbole erweitert und Menüs ändert oder sogar neue eigengestaltete Diagrammtypen in seine SEU aufnimmt.<sup>1)</sup> Dieses hohe Maß an Gestaltbarkeit ist ein wesentlicher Vorteil gegenüber den anderen Produkten.

Alle untersuchten SEU bieten Schnittstellen zu anderen Entwicklungswerkzeugen. Während die ProKit\*WORKBENCH über das TELON Interface und das TRANSFORM Interface den Zugang zu Codegeneratoren und Anschluß an die unternehmungseigene Datenbankabfragesprache Pro IV bietet, verfügt der Excelsator über Schnittstellen zum hauseigenen Analysewerkzeug PC Prism, zum Projektmanagement System ABT's Projekt Workbench und zur Micro Focus Workbench für die COBOL-Code-Generierung. Desweiteren wird von beiden SEU ein Zugang zum Applikationsgenerator CSP/AD und dem Datenbank-System DB2 der IBM offeriert. Die mit der ProKit\*WORKBENCH

---

1) Vgl. Index Technology /A Guided Tour of Customizer/ 6- 9, 12-17, 19

erstellten DB2-Datenstrukturen können an die Datenbanken ORACLE, INGRES und INFORMIX angepaßt werden. Auch die Information Engineering Workbench unterstützt verschiedene Anwendungsgeneratoren wie CSP/AD von IBM oder den eigenen Mainframe-COBOL-Generator GAMMA. Alle Objekte der Enzyklopädie können als ASCII-Daten anderen Programmen zur Verfügung gestellt werden. Schnittstellen existieren beispielsweise zum DATMANAGER, FOCUS, ROCHADE und TELON. Der Systems Engineer verfügt über alle Schnittstellen seines Vorgängerproduktes Auto-Mate Plus. So können Daten unter anderem in folgende Datenbanken, Data Dictionary und Dateiformate exportiert werden: ADABAS, DATAMANAGER, CSP/AD, INFORMIX, TELON, UNISYS/LINC. Einen Vorteil bietet das Produkt von Index Technology durch die Möglichkeit, mit Hilfe des XL/Programmer Interface einen direkten Zugriff auf das Repository zu realisieren. McDonnell Douglas gibt die Spezifikation ihrer Entwicklungsdatenbank nicht bekannt. Auch der Knowledge Coordinator der Information Engineering Workbench unterliegt strengster Geheimhaltung. In Ausnahmefällen unterstützt LBMS einen direkten Zugriff des Kunden auf die Projektdatenbank.

In der Benutzerfreundlichkeit zeigen die getesteten Produkte ebenfalls relevante Unterschiede. Im Teilbereich Verständlichkeit jedoch sind sie ähnlich zu bewerten: alle Produkte sind ohne Vorkenntnisse auf dem Gebiet der Softwareentwicklung, wenn überhaupt, nur schwer verständlich. In der Menügestaltung bietet der Excelerator mit der Verfeinerung von Menüpunkten, ohne das Menü verlassen zu müssen, einen kleinen Vorteil. Er vermeidet eine unüberschaubare Anzahl von Untermenüs. Bei der Beurteilung der Benutzbarkeit fällt die inkonsistente Tastenbelegung in der ProKit\*WORKBENCH auf: so bewirkt in einigen Menüs die <Enter>-Taste einen Aufruf, ein anderes mal wird durch sie lediglich der Cursor eine Zeile tiefer gesetzt und der Aufruf muß über die <F8>-Taste erfolgen. Im Prototypen erfolgt der Funktionsaufruf teilweise über Kombinationen mit der <Alt>-, teilweise mit der <Ctrl>-Taste. Weil der Entwickler sich in der Diagrammbearbeitung an den Gebrauch von Tastenschlüsseln in Kombination mit der <CTRL>-Taste gewöhnt, ist die Gefahr eines Vertauschens groß. Ein solches Vertauschen beim Aufruf der mit der Tastenkombination <Alt-R> aufzurufenden Funktion Edit Image Detail wird mit der Rückkehr in das Hauptmenü ohne Sicherung bestraft. Dies verrät neben der Inkonsistenz auch eine ungenügende Robustheit. Positiv zu beurteilen sind die Pull-down-Menüs beim Systems Engineer und der Information Engineering Workbench. Während beim Systems Engineer nahezu sämtliche Befehle auch über die Tastatur eingegeben werden können, ist der Benutzer der Information Engineering Workbench auf den Gebrauch einer Maus angewie-

sen. Beim Excelerator kann die Oberfläche mit Hilfe des Zusatzprodukts Customizer gestaltet werden, in der Benutzbarkeit hat der Entwickler die Wahlmöglichkeit zwischen Menübenutzung und Tastenschlüsseleingabe. Die Orientierung in der Grafik wird mit Hilfe eines kleinen Überblicksdiagramms sehr gut gelöst. Schlecht dagegen ist die Grafikedarstellung dieser SEU: aufgrund der niedrigen Bildschirmauflösung können die Beschriftungen auf dem Bildschirm nicht verkleinert werden, so daß ein mit der Zoomfunktion im CLOSE UP-Level erstelltes Diagramm in der Größe LAYOUT bereits bei geringer Komplexität auch in seiner groben Struktur nicht erkennbar ist. Das WYSIWYG-Prinzip wird im Gegensatz zu den Kontrahenten nicht verfolgt. Die Information Engineering Workbench enthält zwar eine Scale-Funktion, mit der die Größe der Diagramme auf dem Bildschirm verändert werden kann, diese ist jedoch vergleichsweise umständlich zu bedienen, und die Darstellung auf dem Bildschirm ist von unbefriedigender Qualität. Als nahezu vorbildlich ist dagegen die Benutzbarkeit des Systems Engineer zu bezeichnen; dies resultiert nicht zuletzt aus der Einhaltung des SAA-/CUA-Standards der IBM. Auch die Auto-route-line Funktion, mit der verhindert werden kann, daß Linien andere Objekte kreuzen oder die Option straight-line, mit der Linien begradigt werden können, sind erwähnenswert. Die Hilfefunktion des Systems Engineer ist besser als die der Konkurrenten. Die Hilfestellungen dieser SEU sind wesentlich ausführlicher und effizienter; mit Hilfe des online implementierten Hilfesystems wird das Handbuch größtenteils ersetzbar. Sowohl der Systems Engineer als auch die Information Engineering Workbench bieten neben der üblichen Werkzeughilfe auch Online-Informationen über fundamentale Grundbegriffe der zugrundeliegenden Methoden. Die meist nur einzeiligen Hilfestellungen des Excelerator sind sehr konkret, aber in vielen Situationen unzureichend. Der Vergleich der Fehlerbehandlung führt zu ähnlichen Ergebnissen: auch hier sind die Fehlererklärungen des Excelerator am wenigsten effektiv. Im Antwortzeitverhalten zeigen alle Produkte angemessene Ergebnisse. Der von Standardanwendungsprodukten Verwöhnte muß sich jedoch bei CASE-Produkten auf insgesamt längere Wartezeiten einstellen. Bei einer Installierung im Netz mit durchschnittlicher Hardware-Ausstattung können beim Systems Engineer teilweise unangenehm lange Pausen entstehen; die Option von Windows, beispielsweise während des Ausdrucks weiterarbeiten zu können, schafft in diesem Fall nur einen unzureichenden Ausgleich für die schwache Performance. Die anderen Produkte geben dem Entwickler nicht die Möglichkeit, während der Wartezeit die Arbeit fortzusetzen.

Die Untersuchung der Dokumentationen und Produktbeschreibungen der SEU ergab keine nennenswerten Unterschiede. Lediglich in der übersicht-



lichkeit und Strukturiertheit der Dokumentation verdienen die ProKit\*WORKBENCH und der Systems Engineer eine bessere Benotung, weil ihr Anwendungshandbuch besser strukturiert ist als das der Konkurrenten und ohne permanente Querverweise auskommt. Die äußere Gestaltung der Handbücher für die ProKit\*WORKBENCH, die Information Engineering Workbench und den Systems Engineer ist nicht zu beanstanden; diese Handbücher befinden sich in einem stabilen Ordner. Der Excelerator stellt seine Handbücher in mehreren einfach gebundenen Heften oder Büchern zur Verfügung, hierbei erfolgt eine nicht immer sinnvolle Aufteilung: so sind beispielsweise die Beschreibungen der ersten zwei Funktionen des Analysis-Moduls im Facilities & Funktions Reference Guide zu finden, zum Studium der Erläuterungen für die nächsten drei Funktionen wird der Data & Report Reference Guide benötigt, die letzte Funktion wird in der Einleitung des Data & Reports Reference Guide beschrieben. Die ausführlichste Dokumentation stellt die Information Engineering Workbench zur Verfügung, die kürzeste der Systems Engineer.

Die Einführung wird von allen CASE-Anbietern unterstützt: sie bieten Schulungen und Telefonsupport an. Bei McDonnell Douglas gehört ein Methodenhandbuch von Gane und Sarson, an dem die Unternehmung die Lizenzrechte erworben hat, zum Lieferumfang.

In der Konditionengestaltung ergeben sich keine entscheidungsrelevanten Unterschiede zwischen dem Excelerator, der ProKit\*WORKBENCH und dem Systems Engineer; die Stückpreise liegen um 20.000 DM. Lediglich die Information Engineering Workbench erscheint mit 21.000 DM pro Workstation auf den ersten Blick wesentlich teurer, allerdings ergeben sich hier diverse Gestaltungsfreiräume, z. B. durch den Erwerb eines Starter-Kits (Planning, Analysis und Design Workstation) für 35.000 DM. Um bezüglich dieses Kriteriums eindeutige Aussagen zu machen, sind sicherlich individuelle Verhandlungen mit den Vertreibern erforderlich, da sich beispielsweise bei höheren Stückzahlen ganz andere Preisrelationen ergeben können.

Bei der Bewertung von anbieterspezifischen Kriterien erzielen Index Technology und KnowledgeWare gute Ergebnisse, weil sie die ältesten CASE-Produkte vertreiben und die höchsten Installationszahlen vorweisen können. So waren per September 1990 in der Bundesrepublik 2000 Module der Information Engineering Workbench und 200 Installationen des Excelerator im Vergleich zu 30 Installationen der ProKit\*WORKBENCH und 13 Installationen bei

Systems Engineer<sup>1)</sup> laut Anbieterangaben vorhanden. Weltweit existieren 45.000 Module der Information Engineering Workbench, der Excelerator wurde 45.000 mal, die ProKit\*WORKBENCH 3.000 mal und der Systems Engineer 1.000 mal installiert.<sup>2)</sup> Infolge des unterschiedlichen Alters der SEU und aufgrund der problematischen Aussagekraft der Installationszahl sollten diese Zahlen jedoch nicht überbewertet werden. Alle Unternehmungen unterhalten Kundenstützpunkte in der Bundesrepublik, ihre wirtschaftliche Lage gibt keinen Anlaß zur Besorgnis.

## 5.2 Bewertung anhand spezieller Kriterien für die Auswahl von Software-Entwicklungsumgebungen

### 5.2.1 Unterstützung der Phasen des Software-Lebenszyklusses

#### 5.2.1.1 Entwurfs- und Analysephase

Alle Produkte unterstützen die Entwurfs- und Analysephase, allerdings in unterschiedlicher Weise. Sowohl der Excelerator als auch die ProKit\*WORKBENCH und der Systems Engineer unterstützen im Gegensatz zur Information Engineering Workbench die Arbeit mit strukturierten Checklisten und Matrizen nicht, der Excelerator bietet jedoch über das Zusatzprodukt PC Prism die Möglichkeit, Unternehmungsziele und -aufgaben, Organisationsstrukturen und Informationsquellen und -anforderungen über menügesteuerte Matrizen zu entwickeln und miteinander in Beziehung zu setzen.<sup>3)</sup> Die ProKit\*WORKBENCH zeigt sich in mehreren Bereichen komfortabler, aber weniger flexibel als ihre Mitbewerber. So bietet die ProKit\*WORKBENCH im Bereich der konzeptionellen Datenmodellierung einen hohen Komfort, weil sie in der Lage ist, Attribute in das Entity-Relationship Diagramm zu integrieren. Der Excelerator, die Information Engineering Workbench und der Systems Engineer stellen für die Datenmodellierung neben einem Entity-Relationship Diagramm auch ein Datenstrukturdiagramm zur Verfügung. In der Prozeßmodellierung bieten der Excelerator und die Information Engineering Workbench eine größere Auswahl als die Konkurrenz: neben Structure Charts und Minispezifikation wie in der ProKit\*WORKBENCH und im Systems Engineer stellen sie zusätzlich ein Struktur Diagramm, das dem Jackson Diagramm

- 
- 1) Die Zahlen für den Systems Engineer repräsentieren den Stand im Februar 1991.
  - 2) Die Zahlen beruhen auf Anbieterangaben.
  - 3) Vgl. Index Technology /PC Prism/ 1

gleich sowie einige andere Diagramme zur Verfügung. Die Projektplanung wird in der ProKit\*WORKBENCH nicht unterstützt, im Excelerator sind Ansätze in Form von Work Breakdown Diagrammen zu finden. Die Information Engineering Workbench und der Systems Engineer halten Werkzeuge bereit, die lediglich Dokumentationscharakter besitzen. Die Software-Wiederverwendung wird in allen SEU lediglich durch die Abspeicherung der Entwürfe im Repository und der Option zur Bildung von Programmbibliotheken gewährleistet; Hilfen zum Auffinden wiederverwendbarer Module werden nicht geboten. Bei McDonnell Douglas steht ein Modul zur Respezifikation kurz vor der Produktreife; es wird einigen Kunden auf Wunsch bereits zur Verfügung gestellt. Dieses Modul ist in der Lage, Zusammenhänge eines COBOL-Programms in einem Structure Chart darzustellen sowie Programmroutinen in Modulbeschreibungen einzulesen. Es eröffnet somit die Möglichkeit des Reverse Engineering, nicht mit SEU erstellte Software zu warten und Teile von ihr wiederzuverwenden.

#### 5.2.1.2 Realisierungsphase

Die fehlende Unterstützung dieser Phase ist ein Mangel vieler CASE-Produkte. Der Excelerator und die ProKitWORKBENCH besitzen lediglich Code-Generatoren zur Übersetzung der Schnittstellenentwürfe; zum Zwecke der Erstellung kompletter Softwareprogramme wird nur eine Schnittstelle zu marktgängigen Generatoren angeboten. Der Systems Engineer offeriert in dieser Hinsicht die wenigsten Möglichkeiten; Routinen zur Codegenerierung existieren überhaupt nicht und Schnittstellen zu AD/Cycle-Produkten wie CSP/AD sind lediglich angekündigt, die bestehenden Schnittstellen sind hauptsächlich über das Vorgängerprodukt Auto-Mate Plus zu erreichen und betreffen primär den hauseigenen Anwendungsgenerator Application Engineer<sup>1)</sup>. Der Excelerator verfügt über eine Schnittstelle zur Micro Focus Workbench für die COBOL-Programmierung und zu DB2 für die Überführung in physische Datenstrukturen.<sup>2)</sup> Die ProKit\*WORKBENCH beinhaltet eine Verbindung zu TELON für die COBOL und PL/I Programmierung sowie zur hauseigenen 4GL-Sprache PRO IV.<sup>3)</sup> Beide SEU bieten eine Schnittstelle zum Anwendungsgenerator CSP/AD der IBM. Bedingt durch die fehlende Integration einer automatisierten Code-Generierung werden von beiden Anbietern ebenfalls keine Testwerkzeuge angeboten. Die Information Engineering Workbench stellt mit

---

1) Siehe hierzu auch Kapitel 5.1

2) Vgl. Index Technology /Data & Reports/ 1-11 - 1-12

3) Vgl. McDonnell Douglas /Application Manual/ 18-3 - 18-4

der Construction Workstation die größten Möglichkeiten der Codegenerierung zur Verfügung. Dabei werden die Ergebnisse der früheren Phasen zur Generierung von COBOL-Source-Code verwendet. Ausgeprägte Testwerkzeuge werden jedoch auch in der Construction Workbench nicht geboten.

#### 5.2.1.3 Erprobungs- und Konsolidierungsphase

Für die Erprobungs- und Konsolidierungsphase sind keine speziellen Werkzeuge integriert, alle SEU dienen vornehmlich dem Einsatz in den frühen Phasen der Software-Entwicklung.

#### 5.2.1.4 Pflege- und Wartungsphase

Da aus Konsistenzgründen bei erforderlichen Änderungen der betroffene Systemteil einen gesamten Software-Entwicklungszyklus durchlaufen soll, ist die Unterstützung dieser Phase in den drei vorherigen Kapiteln bereits diskutiert worden. Besondere Bedeutung erlangt in der Wartungsphase das kurz vor der Marktreife stehende Zusatzprodukt von McDonnell Douglas für die Respezifizierung des COBOL-Code von Programmen, die nicht mit der ProKit\*WORKBENCH erstellt wurden. Somit eröffnet diese SEU nicht nur die Möglichkeit, seit Jahren eingesetzte Software wiederzuverwenden, sondern darüberhinaus diese Software zu warten und somit einen Beitrag zum Abbau von Altlasten zu leisten. In Verbindung mit CSP/AD können in der Information Engineering Workbench bestehende COBOL- und VSAM-Beschreibungen, die in COBOL-Bibliotheken und -Quellisten definiert sind, in die Enzyklopädie geladen werden, wobei automatisiert Datenstrukturdiagramme entstehen; diese können dann wieder zur Neugenerierung einer CSP/AD-Anwendung genutzt werden.

### 5.2.2 Unterstützung phasenübergreifender Aufgaben

#### 5.2.2.1 Informationsverwaltung

Während einige Eigenschaften der Informationsverwaltung in allen SEU ähnlich gelöst sind, wie beispielsweise die Vielfalt der speicherbaren Objekte einschließlich der Änderungsfunktionen, der Beschreibungsmöglichkeiten der Objekte oder das Anbieten einer Suchfunktion, so zeigen sich Unterschiede in den Auswertungsmöglichkeiten der Informationen und besonders in der Integration der Informationsverwaltung mit den Grafikeditoren. Dies resultiert aus der unterschiedlichen Konstruktion, die mit Hilfe der nachfolgenden verein-

fachten Datenflußdiagramme kurz erläutert werden soll.

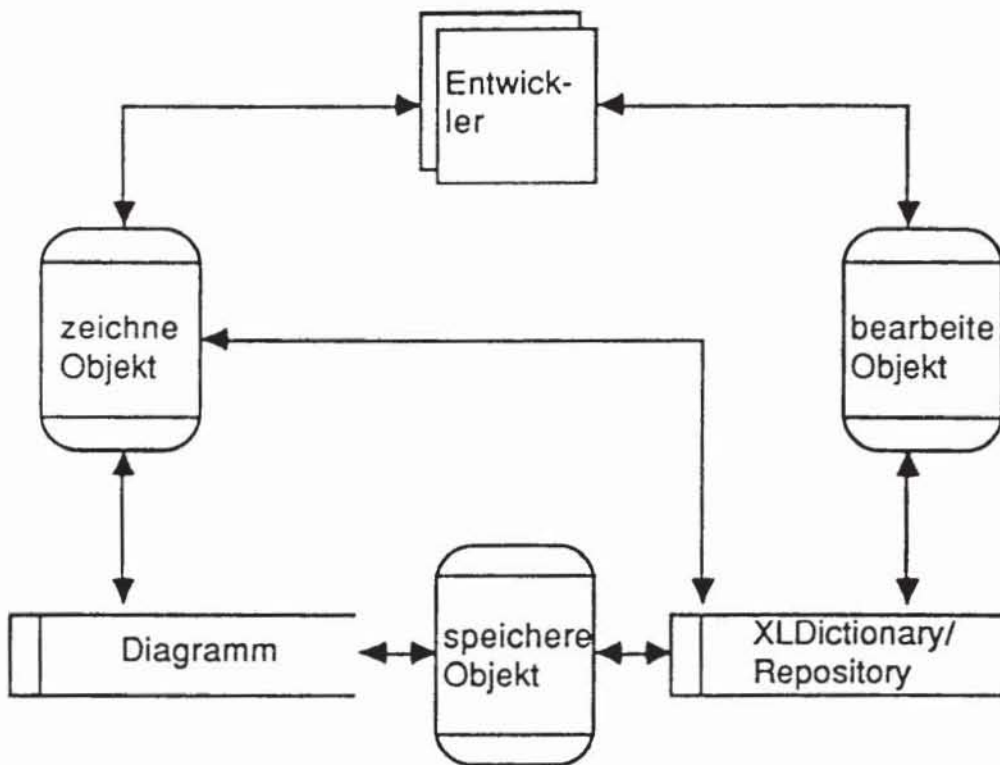


Abb. 3: Informationsverwaltung Excelsator und ProKit\*WORKBENCH

Der Excelsator und die ProKit\*WORKBENCH besitzen eine relativ lockere Verbindung zwischen Diagrammen und Entwicklungsdatenbank, die bewußt Inkonsistenzen zuläßt, welche allerdings mit Hilfe vorgefertigter Berichte nachträglich festgestellt werden können. Aber auch zwischen diesen beiden Produkten bestehen Unterschiede im Detail: während der Excelsator die Diagramme separat abspeichert und lediglich eine Verbindung zum XLDictionary bietet, werden die Grafikinformatoren in der ProKit\*WORKBENCH ebenfalls im Repository abgelegt. Aus diesem Grund sind die Verbindungen der Objekte zueinander in der ProKit\*WORKBENCH bereits automatisiert eingetragen, während diese Arbeiten beim Excelsator vom Entwickler manuell ausgeführt werden müssen. In einem Diagramm des Excelsator können Informationen über ein Objekt mittels der describe-Funktion direkt ins XLDictionary eingetragen, wieder abgerufen oder modifiziert werden. Nach einer solchen Aktion wird der Benutzer direkt in die Grafik zurückgeführt. In der ProKit\*WORKBENCH erfolgt der Aufruf der dem Objekt zugeordneten Seite des Repository über eine Schlüsselfunktion. Bevor die Informationsverwaltung wieder verlassen wird, können andere Seiten des Repository aufgerufen werden, d.h. der Entwickler wird nicht direkt in die Diagramme zurückgeleitet. In diesem Punkt ist die ProKit\*WORKBENCH flexibler als ihr Konkurrent aus Cambridge. Flexibilität beweist der Excelsator durch die Möglichkeit, Symbole einzufügen

und mit Hilfe der label-Funktion zu beschriften, ohne daß ein Eintrag im XLDictionary erfolgt. Das Einfügen von Symbolen ohne Speicherung im Repository ermöglicht die ProKit\*WORKBENCH ebenfalls, jedoch müssen die Symbole eine ID<sup>1)</sup> besitzen, wenn sie beschriftet werden sollen. Die ID bewirkt einen Eintrag in das Repository. Diese Konstruktion verwirkt die Möglichkeit, zunächst einen groben Entwurf eventuell als Diskussionsgrundlage zu erstellen, bevor mit der Eingabe eines durchdachten Modells begonnen wird. Während im Excelerator beim Löschen von Grafiksymbolen gefragt wird, ob nur das Symbol in der Grafik oder auch der Eintrag im XLDictionary gelöscht werden soll, werden Löschungen in Diagrammen der ProKit\*WORKBENCH in ihrem Repository nicht berücksichtigt. Wenn direkt in der Informationsverwaltung Objekte erstellt werden, so erscheinen diese bei allen SEU in keiner Grafik. Bei Änderungen und Löschungen von Objekten verhalten sich die Produkte unterschiedlich. Der Label eines Symbols im Excelerator wird bei Löschungen des korrespondierenden Datensatzes im XLDictionary wie folgt modifiziert: es ändert sich lediglich die Farbe des Symbols, um zu zeigen, daß eine Verbindung des Symbols zum Datensatz der Entwicklungsdatenbank nicht mehr besteht. In der ProKit\*WORKBENCH dagegen werden die Änderungen direkt auf die Diagramme übertragen, Löschungen im Repository führen zunächst zum Aufzeigen von Konsequenzen, nach deren Bestätigung die entsprechenden Symbole von der Grafik verschwinden; verbundene Symbole, die ohne das Gelöschte ihren Sinn verlieren, werden gleichzeitig eliminiert.

---

1) Mit der Identification (ID) wird ein Objekt in der Entwicklungsdatenbank eindeutig bezeichnet.

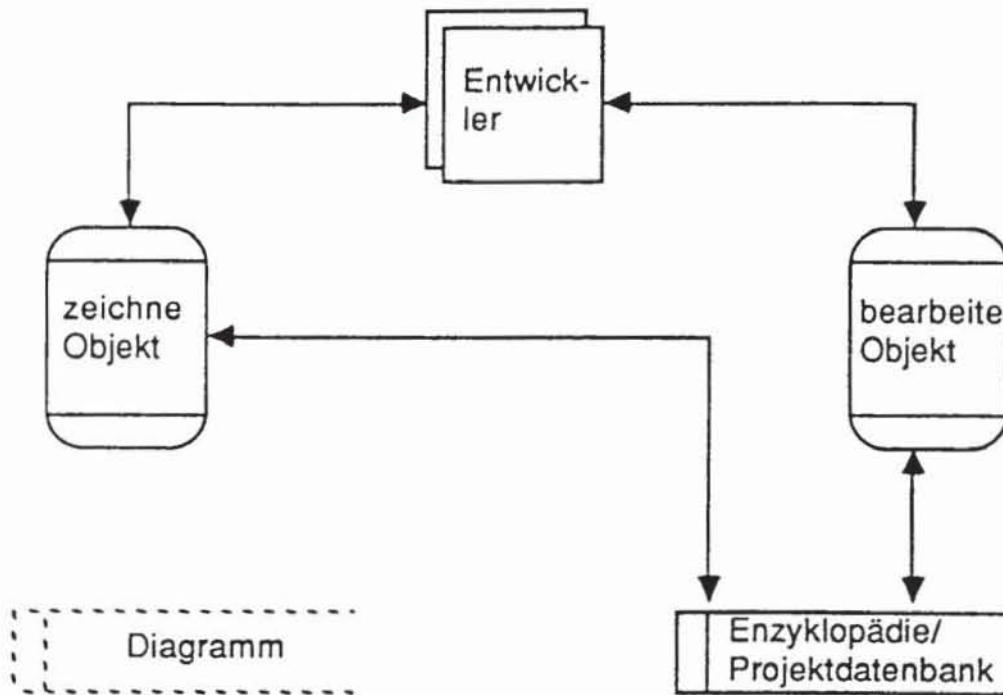


Abb. 4: Informationsverwaltung Information Engineering Workbench und Systems Engineer

Eine prinzipiell andere Architektur weisen die Information Engineering Workbench und der Systems Engineer auf. So werden die Diagramme bei der Information Engineering Workbench unmittelbar aus der Enzyklopädie generiert und nicht als separate Datei gespeichert. Der Knowledge Coordinator sorgt für eine ständige Konsistenz der Diagramme mit der Enzyklopädie sowohl für alle Diagramme einer Workstation als auch für die Diagramme der verschiedenen Workstations untereinander. Da der Systems Engineer primär für den Einsatz im Netz konzipiert ist, wird hier ebenfalls eine ständige übereinstimmung der Diagramme mit der Projektdatenbank gewährleistet. In beiden SEU erfolgt nach vorangegangener Warnung über die Konsequenzen einer Löschung des Objektes eine Eliminierung aller Einträge sowohl im Diagramm als auch in der Entwicklungsdatenbank. Ob diese Konstruktion, die Inkonsistenzen zwischen Diagrammen und Entwicklungsdatenbank vermeidet, oder aber das Konzept des Excelerator und auch teilweise der ProKit\*WORKBENCH, das dem Entwickler die Möglichkeit des kreativen Experiments ohne Belastung der Entwicklungsdatenbank bietet, die bessere Alternative darstellt, muß vom Entscheidungsträger selbst beurteilt werden. Neben einer reichhaltigen Auswahl von Berichten bieten vor allem der Excelerator, die Information Engineering Workbench und die ProKit\*WORKBENCH die Flexibilität, Berichte selbst zu erstellen.

#### 5.2.2.2 Dokumentation

Alle SEU unterstützen die Dokumentation bereits durch ihre Grafikwerkzeuge, insbesondere durch die Möglichkeit des Ausdrucks. Besonders zu erwähnen ist die Verfügbarkeit von Präsentationsgrafiken beim Excelsior, der ProKit-WORKBENCH und dem Systems Engineer, die es ermöglichen, für die nicht mit den Softwareentwicklungsmethoden geübten Projektbeteiligten die Entwürfe zu illustrieren. Unterstützung bieten alle bewerteten CASE-Produkte durch die Bereitstellung bereits vorgefertigter Berichte und bei der Eingrenzung von Objektarten und -bereichen von gestaltbaren Berichten. Das Layout der Berichte ist im Excelsior begrenzt gestaltbar, die Anpassung des Layout an Unternehmensstandards der Entwickler ist bei allen Produkten, abgesehen von festen Kopf- bzw. Fußzeilen, nicht möglich. Beim Excelsior, der Information Engineering Workbench und der ProKit\*WORKBENCH kann eine direkte Verbindung zu einem Textverarbeitungssystem erstellt werden; der Excelsior bietet zusätzlich ein Xerox Ventura Publisher Interface. Der Systems Engineer kann über die Zwischenablage unter Windows mit anderen Windows-Applikationen kommunizieren. Im Excelsior können Dokumente verschiedenster Klassen wie Grafiken, Texte und Berichte miteinander verknüpft werden. Die Dokumente werden in einem Documentation Graph strukturiert, die Symbole mit Objekten der Entwicklungsdatenbank verbunden. Die Namen der beschriebenen Dokumente sind in Datensätzen enthalten. Auf diese Weise kann die Dokumentation partiell oder insgesamt mit einem Befehl erstellt werden. Diese Funktion wird von den anderen Produkten nicht unterstützt. Lediglich die Construction Workstation der Information Engineering Workbench bietet eine Hilfestellung bei der Generierung von Dokumentationen, die Kombination von Text und Grafik ist jedoch auch hier nicht möglich.

#### 5.2.2.3 Software-Qualitätssicherung

Die untersuchten SEU konzentrieren sich vornehmlich auf die Entwurfs- und Analysephase, deshalb ist es nicht verwunderlich, daß sie, abgesehen von einem Test Plan-Report der ProKit\*WORKBENCH, lediglich die Qualität des Entwurfs sichern. Bereits bei der Kontrolle der Eingabesyntax in Diagramme zeigen sich Unterschiede: während der Excelsior, die Information Engineering Workbench und die ProKit\*WORKBENCH sofort die Syntax überprüfen, um falsche Eingaben wie beispielsweise eine Verbindung zweier External Entities<sup>1)</sup> in einem Datenflußdiagramm zu unterbinden, kann im Excelsior

---

1) Durch External Entities werden aus der Sicht des Systems



zunächst gegen Regeln verstoßen werden. Die Regelverstöße können anschließend in einem Prüfbericht, Data Flow Diagram Verification, aufgelistet werden. Diese Flexibilität ist zu begrüßen, gleichwohl sollte von der Report-Funktion häufig Gebrauch gemacht werden, damit Fehler möglichst frühzeitig erkannt werden. Die Information Engineering Workbench und der Systems Engineer unterscheiden zwischen harten und weichen Regeln; d.h. bestimmte Verstöße werden direkt unterbunden, andere werden zugelassen jedoch mit einer Warnmeldung versehen. Alle Produkte bieten eine Reihe von Berichten, um Regelverstöße, Unvollständigkeiten und Inkonsistenzen aufzudecken. Bei der Beurteilung dieses Berichtsangebots fällt die umfangreiche Hilfestellung des Excelsior bei der Untersuchung der Datenstrukturen und deren Gebrauch in seiner Extended Analysis auf. 32 Berichte analysieren den Inhalt von Datensätzen, die Schlüsselverwendung, den Zusammenhang von Datenmodell, Informationsverwaltung und Datenflußdiagramm, die Einhaltung der Normalisierungsregeln und die Verwendung der Daten in den Benutzerschnittstellen. Die ProKit\*WORKBENCH verfügt dagegen über umfangreichere Berichte für die Überprüfung der Vollständigkeit und Konsistenz des Entwurfs in ihrer Report-Funktion. Eine Gegenüberstellung von Elementen aus verschiedenen Diagrammklassen in Form einer Matrix wird von keinem der beiden Produkte angeboten. Auch die Information Engineering Workbench weist mehr als 20 Berichte auf. Beim Systems Engineer ist das Fehlen einer Dokumentation der verfügbaren Berichte zu konstatieren, so daß dem Entwickler die vollständigen Möglichkeiten dieser Option oft verborgen bleiben.

#### 5.2.2.4 Prototyping

Alle Produkte unterstützen nicht die Erstellung eines vollständigen Anwendungsausschnitts, sondern konzentrieren sich auf die Benutzerschnittstellen. Die ProKit\*WORKBENCH beschränkt sich im Gegensatz zu zahlreichen anderen PC-gestützten SEU nicht nur auf die Simulation von Menüeingaben, sondern bietet zusätzlich eine Demonstration des Funktionstastengebrauchs; durch diesen wird zusätzlich die Simulation einer anderen Zielumgebung ermöglicht. Außerdem kann mit dieser SEU das Systemverhalten bei Benutzereingaben simuliert werden, denn die einzelnen Menüfelder sind mit Kontrollflüssen zu den entsprechenden Masken verbunden. Analog zur ProKit\*WORKBENCH erlaubt auch der Systems Engineer die Kombination verschiedener Menüs und bietet zusätzlich eigene Diagramme zur Dialogge-

---

Außenstehende gekennzeichnet. Eine Verbindung zweier External Entities stellt eine Beziehung außerhalb des Systems dar und braucht im System nicht modelliert zu werden. Vgl. McDonnell Douglas /SSAD/ o. S.

staltung an. Während der Systems Engineer ausführlich die Bildschirmmaskenerstellung einschließlich der Simulation der Systemreaktion auf Benutzereingaben unterstützt, beschränkt sich der Screen-Editor im Excelerator bei der Simulation auf eine feststehende Kette aus 6 Masken; diese Funktion wird nicht als Prototyping, sondern als Screen Design bezeichnet. Die Benutzereingaben im Screen Data Entry werden in anzulegende Dateien gespeichert, auf die bei der Vorführung des Berichts demonstration zugegriffen werden kann. Auch bei der Information Engineering Workbench erfolgt keine echte Verknüpfung von Bildschirmmasken; es handelt sich vielmehr beim sogenannten Presentation Diagram Editor um einen mehr oder weniger komfortablen Screen Painter, der allerdings erst in der Design Workstation zur Verfügung steht. Beim Systems Engineer stört die Tatsache, daß zwar eine Simulation von Benutzereingaben unterstützt wird, die Eingaben des Benutzers allerdings nicht über die zugeordneten Tasten, sondern durch das Anklicken der entsprechenden Bezeichnung mit der Maus erfolgen. Die Erstellung des Prototypen ist beim Excelerator und der ProKit\*WORKBENCH für den Ungeübten nicht einfach. In beiden SEU muß sich der Entwickler zunächst auf eine andere Benutzeroberfläche einstellen, die nicht so komfortabel wie die Oberfläche der restlichen Umgebung ist; dies gilt insbesondere für die ProKit\*WORKBENCH. Im Gegensatz dazu ist die Bedienung der Screen-Painter der Information Engineering Workbench und des Systems Engineer sehr benutzerfreundlich. Bei der Prototypgestaltung kommt allen Produkten die Integration mit der Informationsverwaltung zugute, da Datenattribute übernommen werden können. Positiv hebt sich die ProKit\*WORKBENCH durch die Möglichkeit der Verbindung mit dem Datenflußdiagramm ab, weil der Benutzer die im entsprechenden Datenfluß enthaltenen Elemente angezeigt bekommt. Eine Übersetzung in Programmcode ist beim Excelerator und bei der ProKit\*WORKBENCH nur begrenzt, beim Systems Engineer überhaupt nicht realisiert; die ProKit\*WORKBENCH liefert jedoch die Ausgabe für eine Produktion von Menüs, Bildschirmmasken und Formularen für CICS und IMS Umgebungen. Bei der Information Engineering Workbench besitzt wiederum die Construction Workstation die Fähigkeit, COBOL-Source-Code zu generieren.

#### 5.2.2.5 Projektmanagement

Die Projektplanung wird von der ProKit\*WORKBENCH nicht unterstützt. Der Excelerator stellt lediglich eine Work Breakdown Structure, ein grafisches Beschreibungsmittel zur Projektstrukturierung, zur Verfügung. Eine weitere Hilfestellung zur Projektplanung wird nicht geboten. Dies gilt auch für den Sy-

stems Engineer und mit wenigen Ausnahmen in der Planning Workstation ebenfalls für die Information Engineering Workbench. Zur Sicherung von Entwicklungsergebnissen können in allen Produkten Zugriffsrechte vergeben werden. Der Excelerator bietet darüberhinaus die Möglichkeit, bestimmte Diagramme vor den Eingriffen anderer Projektbeteiligter mit Hilfe eines Lock-Mechanismusses zu schützen.

#### 5.2.2.6 Konfigurationsmanagement

Die ProKit\*WORKBENCH enthält eine Versionsverwaltung, mit der bis zu acht Projektversionen in der Entwicklungsdatenbank gehalten werden können; die zu bearbeitende Version ist jeweils zu aktivieren. Diesen Dienst bieten die Konkurrenten nicht. Dagegen ist die Änderungsdokumentation im Excelerator, in der Information Engineering Workbench und im Systems Engineer etwas vorteilhafter, weil hier neben dem Datum der letzten Änderung der Name des Verantwortlichen und nicht wie in der ProKit\*WORKBENCH der Name der Workstation gespeichert wird. Das Zählen der Änderungen im Excelerator kann zwar im Einzelfall von Vorteil sein (z. B. aus Sicherheitsgründen), demotiviert jedoch den Entwickler, Änderungen vorzunehmen und wirkt deshalb eher störend. Bei der Information Engineering Workbench besteht lediglich die Möglichkeit, Enzyklopädien verschiedener Entwickler bzw. Projekte zu konsolidieren. Der Systems Engineer bietet noch nicht einmal diesen Service.

#### 5.2.2.7 Allgemeine Bürodienste

Abgesehen von der direkten Verbindung zu einem Textverarbeitungssystem bieten alle SEU keine weitere Unterstützung von allgemeinen Bürodiensten. Dem Systems Engineer kommt an dieser Stelle wiederum die Verfügbarkeit verschiedener Zubehörprodukte von Windows zugute.

### 5.2.3 Gestaltungsorientierte Auswahlkriterien

#### 5.2.3.1 Methoden

Der Excelerator ist eine methodenneutrale SEU, d.h. bei der Nutzung können eine Reihe unterschiedlicher Methoden angewendet werden. Diese Neutralität ist besonders wertvoll, wenn der Entwickler eine unternehmungsspezifische Methode anwenden möchte, mit der er vertraut ist und die sich in bereits abgewickelten Projekten bewährt hat. Die ProKit\*WORKBENCH ist auf die

Structured Analysis Methode von Gane und Sarson<sup>1)</sup> zugeschnitten; in diesem Zusammenhang muß erwähnt werden, daß diese Methode ihrerseits andere Methoden einschließt. So sind die Ansätze des Relationenmodells von Codd und der Structured Design-Methode von Yourdon und Constantine in ihr enthalten. Desweiteren unterstützt die ProKit\*WORKBENCH den Entity-Relationship Ansatz in den Notationen von Chen und Bachman. Somit ist das Spektrum der angebotenen Methoden ausreichend groß; es lassen sich datenfluß-, daten- und prozeßorientierte Sichtweisen mit diesem CASE-Produkt modellieren. Die Information Engineering Workbench ist ursprünglich zur Unterstützung des Information Engineering nach James Martin konzipiert, stellt aber darüberhinaus ausreichende Techniken und Werkzeuge zur Anwendung anderer Methoden bereit. Ernst & Young bietet mit dem Navigator auch eine eigene Methode an. Dies gilt auch für den Systems Engineer (Systems Engineering), dessen Hersteller LBMS ursprünglich ein Methodenhaus war.

### Structured Analysis

Die ProKit\*WORKBENCH ist auf diese Methode zugeschnitten und bietet daher diesbezüglich eine hervorragende Unterstützung. So gehört ein ausführliches Methodenhandbuch<sup>2)</sup> zum Lieferumfang. Die von McDonnell Douglas für die Abwicklung technischer Großprojekte entwickelte Vorgehensweise STRADIS kann zusätzlich erworben werden.<sup>3)</sup> Auch der Systems Engineer zeigt seine Stärken bei der Unterstützung dieser Methode. Während mit dem Datenflußdiagramm-Editor des Excelerator auch Symbole dargestellt werden können, die nicht zur Structured Analysis Methode gehören, wie Kontrollspeicher und -flüsse, so ist der entsprechende Editor der ProKit\*WORKBENCH auf die Notation von Gane und Sarson abgestimmt: es können keine Kontrollsymbole dargestellt, dafür Suchargumente in das Datenflußdiagramm aufgenommen werden, Wiederholungen von Symbolen werden durch Wiederholungskennzeichen markiert. Ähnliches gilt für den Systems Engineer, der im Gegensatz zur Information Engineering Workbench keine Wahl zwischen verschiedenen Notationen zuläßt. Dafür fehlt bei der Information Engineering Workbench die Möglichkeit, Symbole in einem Diagramm mehrfach zu verwenden, was beispielsweise bei großen Diagrammen sehr leicht zu einer unübersichtlichen Vielfalt an Datenflüssen führen kann. Desweiteren können bei der ProKit\*WORKBENCH Symboltypen in unterschiedlichen Farben darge-

---

1) Siehe Gane, Sarson /Structured Systems Analysis/ 1 ff.

2) McDonnell Douglas besitzt die Lizenzrechte an dem Methodenhandbuch Structured Systems Analysis von Gane und Sarson.

3) Vgl. Müßig /Software Engineering Environment System/ 450

stellt und bestimmbare Symbole visuell hervorgehoben werden. Die Highlight-Funktion erleichtert das Verfolgen der Datenströme und somit das Entwickeln eines Soll-Modells aus einem Ist-Modell. Durch die Spezialisierung auf nur eine Methode ist eine Kontrolle der Einhaltung der Methodenregeln bereits bei der Eingabe möglich. Der Excelerator bietet diese methodenspezifischen Möglichkeiten nicht. Auch die Überprüfungen der Information Engineering Workbench und des Systems Engineer beschränken sich mehr auf methodenunabhängige, syntaxorientierte Fehler (z. B. Datenspeicher ohne Verbindungen zu Prozessen). Mit allen SEU ist es möglich, Prozesse durch hierarchisch untergeordnete Datenflußdiagramme darzustellen. Der Excelerator numeriert die Prozesse automatisiert und bildet die ein- und ausströmenden Datenflüsse im verfeinerten Diagramm ab. Auf diese Weise ist der Entwickler nicht gezwungen, ständig die Diagramme zu wechseln, und die Gewährleistung der Vollständigkeit des Entwurfs wird gesichert. Auch beim Systems Engineer werden bei der Verfeinerung eines Prozesses sämtliche Verbindungen und ihre zugehörigen Objekte automatisiert mit in die nächste Diagrammebene übernommen. Außerdem werden die Prozeßgrenzen markiert. Dies hat zusammen mit der Option, mehrere Fenster und damit mehrere Datenflußdiagrammebenen auf einem Bildschirm darstellen zu können, ein Höchstmaß an Übersichtlichkeit zur Folge. In der Information Engineering Workbench werden Kennzeichen gesetzt, von wo bestimmte Informationen höherer Diagrammebenen kommen. Auch hier machen sich die Vorteile der Fenstertechnik bemerkbar. Die ProKit\*WORKBENCH bietet diesen Vorteil nicht, dafür ist es bei diesem Produkt möglich, die Prozeßgrenzen und die mit dem Prozeß verbundenen Objekte im verfeinerten Datenflußdiagramm darzustellen. Eine automatisierte Aktualisierung hierarchisch höherer Diagramme bei Änderungen ist bei keiner der getesteten SEU implementiert. In allen CASE-Produkten kann aus den Diagrammeditoren auf die Informationsdatenbasis zugegriffen werden. Die nur von Gane und Sarson geforderte Datenmodellierung mit dem Ziel, die Inhalte der Datenspeicher in der dritten Normalform zu präsentieren, wird von allen ebenso wie die Prozeßspezifikation unterstützt. Für die Entwicklung der Funktionsstruktur steht beim Excelerator und der ProKit\*WORKBENCH lediglich ein Texteditor zur Verfügung, der zur Entwicklung von Pseudocode verwendet werden soll. Beide Editoren verfügen über Felder zur Auflistung der Ein- und Ausgabedaten, ein Zugang zu einem Datenflußdiagramm ist jedoch nicht realisiert. Ebenso fehlt eine zur Bewältigung komplexer Programmstrukturen hilfreiche Möglichkeit, Entscheidungstabellen oder -bäume darzustellen. Letzteres gilt zwar auch für die Information Engineering Workbench und den Systems Engineer, beide verfügen

jedoch über einen komfortablen Editor, der mit der Informationsverwaltung verbunden ist. Dies macht sich besonders beim Systems Engineer positiv bemerkbar, denn für den Fall, daß beispielsweise Datennamen in der Projektdatenbank geändert werden, schlagen sich diese veränderten Namen auch im Pseudocode (System Design Language) nieder. Durch die Gestaltbarkeit der Schlüsselwörter in den Menüs von COBOL bis zu SQL ist es außerdem möglich, unternehmungsspezifische Besonderheiten zu berücksichtigen.

### Entity-Relationship Ansatz

Alle SEU unterstützen den Entity-Relationship Ansatz. Der Excelerator bietet die Wahlmöglichkeit zwischen der Notation von Chen und Merise; hierbei ist die Konvertierung bestehender Dokumente gewährleistet. Bei der ProKit\*WORKBENCH kann der Entwickler zwischen einer erweiterten Notation von Chen, einer erweiterten Notation von Bachman oder einem relationalen Entity-Relationship Diagramm wählen. Eine Konvertierung wird nicht geboten, dafür lassen sich in einer Projekt-Version Modelle in allen drei Diagrammtypen entwickeln. Bei der Information Engineering Workbench kann durch die individuelle Wahl der Diagrammsymbole auch eine Mischung verschiedener Notationen vorgenommen werden. Der Systems Engineer bietet nur fest vorgegebene Symbole entsprechend der eigenen Methode Systems Engineering; der klassische Entity-Relationship Ansatz kann jedoch auch verfolgt werden. In der Datenmodellierung ist die Funktionalität der SEU von McDonnell Douglas vielen vergleichbaren CASE-Produkten auf dem PC-Markt überlegen. So können in den Diagrammen neben den Grundelementen Entity und Beziehung zusätzlich Attribute, Optionalitäten, qualifizierte Kardinalitäten sowie Primär- und externe Zugriffsschlüssel gezeigt werden. Desweiteren lassen sich Entity-Sub-Types und Entity-Super-Types darstellen, zu denen jeweils Beziehungen gezeichnet werden können. In der Entity-Hierarchie lassen sich Verbindungen von einem Eltern-Entity zu mehreren Kinder-Entities darstellen. Ebenso ermöglicht der Editor das Zeigen rekursiver Beziehungen. Existenziale Abhängigkeiten können lediglich im Repository vermerkt werden. Auch die Information Engineering Workbench bietet wirksame Werkzeuge für die Datenmodellierung. Es sind nahezu alle gängigen Beziehungstypen darstellbar, und es wird zwischen kann- und muß-Beziehungen sowie zwischen fundamental, associative und attributiv Entities unterschieden. Kritisch anzumerken ist jedoch der Umstand, daß alle Beziehungen binär sein müssen. Der Excelerator bleibt in der Datenmodellierung hinter der ProKit\*WORKBENCH und der Information Engineering Workbench zurück: er kann keine Attribute,

keine n-ary-Beziehungen<sup>1)</sup>, keine Optionalitäten, keine Abhängigkeiten und keine Schlüssel im Diagramm darstellen. Der Eintrag der Kardinalitäten erfolgt kompliziert als Label der Verbindungslinien zwischen Objekt und Relation. Eine Bildung von Datenhierarchien ist ebenfalls nicht möglich. Die schwächste Datenmodellierungskomponente bietet jedoch der Systems Engineer, der beispielsweise nicht in der Lage ist, n:m-Beziehungen darzustellen. Zum Entwurf von Datenstrukturen stehen in der ProKit\*WORKBENCH Informationen zur Modellierung einer relationalen Datenstruktur zur Verfügung, in den anderen SEU wird ein Datenstrukturdiagramm bereitgestellt. Eine Automatisierung der Normalisierung wird von allen SEU nicht unterstützt, die ProKit\*WORKBENCH bietet hierbei lediglich Unterstützung in Form einer automatisierten Schlüsselübergabe in 1:N-Beziehungen durch das Entity mit der höheren Kardinalität. Auch die Information Engineering Workbench bietet zumindest Unterstützung bei der Normalisierung, während Systems Engineer hierfür keinerlei Funktionen bereitstellt. Eine Kontrolle der Einhaltung von Normalisierungsregeln wird von allen Produkten, mit Ausnahme des Systems Engineer, geboten.

### Structured Design

Im Excelerator und in der ProKit\*WORKBENCH können Datenflußdiagramme aufgeteilt werden, um Eingabe-, Verarbeitungs- und Ausgabebereich zu trennen. In der Information Engineering Workbench ist es beispielsweise möglich, Transaction Center durch Attributierung zu kennzeichnen. Eine automatisierte Entwicklung eines Structure Chart aus dem Datenflußdiagramm bietet keine der untersuchten SEU. Die ProKit\*WORKBENCH hat neben der Aufrufmöglichkeit eines Datenflußdiagramms aus dem Designer den Vorteil, die Prozeß-Beschreibungen aus dem Repository in die Modulbeschreibung des Structure Chart zu übernehmen. Auch in diesem Zusammenhang ist die gute Verbindung der Enzyklopädie der Information Engineering Workbench mit den entsprechenden Editoren zu loben. Alle CASE-Produkte können eine Reihe der methodenspezifischen Symbole und die Kontrollkonstrukte Sequenz, Selektion und Iteration darstellen.<sup>2)</sup> Erweiterte Konstrukte wie rekursive Strukturen und homologe Systeme werden von keiner der SEU unterstützt.

- 
- 1) n-ary-Beziehungen verbinden im Gegensatz zu binären Beziehungen mehr als zwei Entities.
  - 2) Die einzelnen Unterschiede können dem Auswertungsbogen im Anhang I entnommen werden.

### 5.2.3.2 Kapazität

Die Anzahl der darstellbaren Objekte in der Entwicklungsdatenbank ist bei allen Produkten lediglich durch die Hardwarekonfiguration limitiert. Alle SEU können unbegrenzt Diagramme generieren, auch hier wird die Kapazität ausschließlich durch die Kapazität der Hardware begrenzt. Deutliche Unterschiede ergeben sich bei der möglichen Anzahl der Symbole pro Diagramm: im Excelerator können pro Diagramm 75 Symbole eingetragen werden, während die Diagramme der ProKit\*WORKBENCH je nach Konfiguration der Hardware bis zu 1600 Symbole<sup>1)</sup> aufnehmen. Eine solch große Kapazität sollte vor dem Hintergrund, übersichtliche Diagramme zu erstellen, nicht überbewertet werden. Dagegen kommt der Anzahl der Verfeinerungsebenen eine höhere Bedeutung zu. Die Produkte liefern vergleichbare Werte: der Systems Engineer bietet neun Verfeinerungsebenen, der Excelerator kann Datenflußdiagramme zehnmal verfeinern, die ProKit\*WORKBENCH bietet zwei Stufen mehr und bei der Information Engineering Workbench sind keine Beschränkungen festzustellen.

### 5.2.3.3 Outputqualität

Bei der Gestaltung des Output zeigen sich der Excelerator und der Systems Engineer flexibel: optional ist der Seitenumbruch des Ausdrucks während der Diagrammerstellung sichtbar, Schrift- und Symbolgröße beim Excelerator sind je nach Drucker wählbar, hierbei erhält der Entwickler sofort Informationen über die Auswirkungen auf Seitenanzahl und Beschriftung der Symbole seiner Auswahl. Auch in der ProKit\*WORKBENCH können Symbol- und Schriftgröße gewählt werden, jedoch werden die Konsequenzen für Seitenanzahl und Symbolbeschriftung nicht gezeigt. Die Information Engineering Workbench bietet dem Benutzer nur wenig Gestaltungsfreiheit bezüglich des Layout der Diagramme. Die ProKit\*WORKBENCH ist ihren Kontrahenten überlegen bei der Flexibilität der Gestaltung der Diagrammgröße und der Möglichkeit, Diagramme im Querformat auszudrucken. Durch die Realisierung des WYSIWYG-Prinzips ist eine genaue Platzierung der Labels möglich, einen Vorteil, den der Excelerator nicht bietet. Alle Produkte offerieren die Möglichkeit, einen schnellen Ausdruck im Draft-Modus des Druckers zu erstellen, hierbei können der Excelerator und der Systems Engineer ein mehrseitiges

---

1) Bei 512 KB expanded memory können bis zu 800 Symbole pro Diagramm, bei zusätzlichem Erweiterungsspeicher bis zu 1600 Symbole dargestellt werden. Vgl. McDonnell Douglas /Application Manual/ 21-2



Diagramm auf eine Seite gestaucht ausdrucken. Das äußere Erscheinungsbild der Ausdrücke reicht von einer hervorragenden Qualität bei der ProKit\*WORKBENCH über eine immer noch sehr gute Qualität beim Systems Engineer bis zu einer als schlecht einzustufenden Qualität beim Excelsior und bei der Information Engineering Workbench.<sup>1)</sup>

#### 5.2.3.4 Integration

Alle SEU besitzen eine integrierte Entwicklungsdatenbank, die die Werkzeuge mit Informationen versorgt und Informationen für sie speichert. Während die Möglichkeit, mehrere Diagramme auf einem Bildschirm darzustellen, nur von der Information Engineering Workbench und dem Systems Engineer unterstützt wird, kann in allen SEU zwischen den Diagrammtypen gesprungen werden. Ebenfalls bieten alle Produkte die Auswertung von Daten, die mit verschiedenen Werkzeugen gewonnen wurden. Da die Structured Analysis Methode von Gane und Sarson mehrere Methoden und damit auch Werkzeuge integriert, sind die Werkzeuge der ProKit\*Workbench logisch optimal aufeinander abgestimmt. Entscheidet man sich beim Systems Engineer für die hauseigene Methode, so kann auch hier von einer überzeugenden Abstimmung der einzelnen Werkzeuge gesprochen werden. Von besonderer Bedeutung ist hier die Möglichkeit der Cross-Referenz zwischen den Objekten der Entwicklungsdatenbank. Der Excelsior und die Information Engineering Workbench bieten eine höhere Anzahl von Werkzeugen an, die sich teilweise überschneiden. Die Information Engineering Workbench erlaubt verschiedene Sichten auf Diagramme, deren Integrationsgrad als hervorragend zu bezeichnen ist. Die Integration mehrerer Entwickler durch die Bereitstellung einer gemeinsamen Entwicklungsdatenbank, auf die alle Entwickler gleichzeitig zugreifen können, wird zur Zeit nur vom Systems Engineer angeboten. Alle Anbieter bemühen sich um eine Integration ihrer SEU in den AD/Cycle. Während Index Technology und KnowledgeWare als Product Part of AD/Cycle an der Gestaltung der Gesamtkonzeption und der Schnittstellen beteiligt sind, haben sich McDonnell Douglas und LBMS als AD/Cycle Vendor Program Participant zur Anpassung ihrer SEU an die Schnittstelle des AD/Cycle entschlossen.

---

1) Beispielausdrücke der untersuchten SEU befinden sich im Anhang II dieser Arbeit.

### 5.3 Zusammenfassende Bewertung der ausgewählten Software-Entwicklungsumgebungen

Die vier Produkte wurden anhand eines ca. 280 Einzelkriterien umfassenden Kriterienkatalogs in Form einer Nutzwertanalyse bewertet. Dabei wurden Zielerreichungsgrade von 0 bis 4 vergeben.

In einer ersten Auswertung wird Indifferenz bezüglich der Bedeutung der Einzelkriterien unterstellt. Hierbei ergeben sich lediglich geringe Unterschiede in der Gesamtbewertung. Mit knappem Vorsprung belegt die Information Engineering Workbench den ersten Platz:

- Platz 1:	Information Engineering Workbench	1,99 Punkte
- Platz 2:	Systems Engineer	1,98 Punkte
- Platz 3:	Exceleator	1,93 Punkte
	ProKit*WORKBENCH	1,93 Punkte

In einer weiteren Auswertung<sup>1)</sup> werden die Einzelkriterien unter Berücksichtigung der besonderen Anforderungen im Rahmen der Ausbildung gewichtet. Dabei ergeben sich deutliche Verschiebungen zugunsten des Systems Engineer. Die Ergebnisse im einzelnen:

- Platz 1:	Systems Engineer	2,15 Punkte
- Platz 2:	Information Engineering Workbench	2,06 Punkte
- Platz 3:	Exceleator	2,05 Punkte
	ProKit*WORKBENCH	2,05 Punkte

Zusammenfassend lassen sich folgende Untersuchungsergebnisse festhalten:

- Die SEU erzielen in der Untersuchung ein ähnliches Ergebnis: im Durchschnitt 2,08 von erreichbaren 4 Punkten. Die geringen Punktzahlen verdeutlichen, daß alle vier SEU zwar einerseits eine Reihe von Kriterien erfüllen, allerdings andererseits zahlreiche Wünsche der Entwickler unerfüllt lassen. Bei der Beurteilung ist zu beachten, daß es sich bei den untersuchten SEU um bereits vorselektierte Produkte handelt, die zu den besten am Markt angebotenen zählen.
- Die geringe absolute Differenz der Punktzahlen illustriert, daß keines der Produkte absolut überlegen ist. Bei der Interpretation des Ergebnisses sollte die Subjektivität der Wertung berücksichtigt werden, so daß zum einen die Bewertung von einem Team unter Einbeziehung möglichst vieler Personen, die mit der SEU arbeiten werden, durchgeführt werden

---

1) Vgl. Anhang I

sollte,<sup>1)</sup> zum anderen eine Toleranz von 0,2 Punkten für alle vier Bewertungen einkalkuliert werden muß.<sup>2)</sup> Eine klare Entscheidung kann also erst bei einer Differenz von mindesten 0,4 Punkten getroffen werden.

Zur Beurteilung der ausgewählten SEU kann zusammenfassend festgestellt werden, daß der Excelerator und die Information Engineering Workbench eine höhere Anwendungsbreite, die ProKit\*WORKBENCH und der Systems Engineer eine höhere Anwendungstiefe besitzen.

Der Excelerator erweist sich als änderungsfreundlich mit Hilfe eines Zusatzprodukts und offen. Er eignet sich für einen Entwickler, der sich entweder nicht auf eine Methode festlegen will oder eine unternehmensinterne Methode anwenden möchte.

Die Information Engineering Workbench bietet eine ganze Reihe von Werkzeugen zur Unterstützung der verschiedensten Methoden an und unterstützt vor allem das Information Engineering nach James Martin oder die Methode Navigator von Ernst & Young in hervorragender Weise. Sie zählt zu den professionellsten SEU und eignet sich vornehmlich für Unternehmungen, die bereits Erfahrungen in der Anwendungen moderner Methoden und Werkzeuge der Softwareentwicklung besitzen. Interessant ist die Information Engineering Workbench auch für Unternehmungen, die zur Entlastung ihrer Zentralrechner beabsichtigen, kleinere Projekte vollständig auf dem PC zu entwickeln.

Die ProKit\*WORKBENCH ist auf die Structured Analysis Methode in der Notation von Gane und Sarson abgestimmt. Ihr Spektrum ist zwar auf solche Werkzeuge begrenzt, die für diese Methode vorgesehen sind, dafür werden diese allerdings hervorragend unterstützt. Ein Entwickler, der sich auf diese Methode spezialisieren kann und möchte, sollte diese Umgebung in die engere Wahl ziehen.

Der Systems Engineer zielt primär auf Unternehmungen ab, die die Vorgehensweise von LBMS, Systems Engineer, anzuwenden beabsichtigen oder aber schwerpunktmäßig funktionsorientierte Softwareentwicklung betreiben. Infolge der phasenübergreifenden Benutzerfreundlichkeit kann mit einer hohen Akzeptanz bei den Softwareentwicklern gerechnet werden, wodurch sich die SEU weiterhin auch für Unternehmungen eignet, die in einem produktiven

---

1) Vgl. Schmitz, Schwichtenberg, Sodeur /Kombinationsverfahren/ 13

2) Bröhl bewertet Schwankungen bis zu einer Höhe von 10% als nicht relevant. Vgl. Bröhl /SE-Umgebung/ 100

Lernprozeß erste Erfahrungen im Bereich CASE sammeln möchten.

Die Wahl der richtigen SEU hängt somit entscheidend von der zu unterstützenden Methode ab. Dabei nimmt der Grad der Flexibilität mit dem Grad der Methodenunterstützung ab.

Abschließend sei darauf hingewiesen, daß trotz aller Bemühungen der Autoren, die Untersuchung der SEU objektiv und richtig vorzunehmen, Fehler nicht auszuschließen sind und eine Gewährleistung der Korrektheit aller Angaben nicht übernommen werden kann. Dieser Fachbericht soll einem Entscheidungsträger helfen, die für ihn geeignetste SEU auszuwählen, indem Handlungsempfehlungen für den Selektionsprozeß sowie erste Anhaltspunkte für eine Beurteilung der Produkte geliefert und dabei die bei einer durchgeführten Auswahl erzielten Erfahrungen weitergegeben werden. Der Bericht soll allerdings kein Ersatz für eine eigenständige Auseinandersetzung mit diesen komplexen Werkzeugen sein.

Auch wenn die derzeit auf dem Markt befindlichen SEU noch vieles an Wünschen offenlassen, so ist dennoch allen Anwendern zu raten, erste Erfahrungen mit diesen Werkzeugen zu sammeln und nicht darauf zu warten, daß eine perfekte SEU in den Verkauf gelangt.

## 6 Neue Entwicklungen auf dem Markt für Software-Entwicklungsumgebungen

Der Einsatz von SEU in der Unternehmung stellt für den Verantwortlichen ein Experiment mit hohem Risiko dar, denn die Einführung dieser Produkte ist eine Entscheidung strategischen Charakters. Sämtliche am Markt befindlichen CASE-Produkte weisen noch Schwachstellen auf: so fehlt es den untersuchten Produkten beispielsweise an der Unterstützung der Realisierungsphase durch eingebundene automatisierte Code-Generatoren.<sup>1)</sup> Weiterhin lassen sie umfangreiche Testhilfen vermissen, unterstützen mit Ausnahme des Systems Engineer den Multiuserbetrieb in unzureichendem Maße und stellen den Entwickler vor Kapazitätsprobleme.

Im September 1989 kündigte die IBM den AD/Cycle an.<sup>2)</sup> Wenn die dem AD/Cycle angeschlossenen CASE-Anbieter die Schnittstellen unterstützen,

---

1) Auch die Information Engineering Workbench unterstützt lediglich COBOL

2) Vgl. Fersko-Weiß /CASE-Tools/ 214

wird eine Kommunikation von CASE-Werkzeugen unterschiedlicher Anbieter ermöglicht. Das AD/Cycle-Konzept der IBM könnte einen Großteil der oben aufgezählten Probleme der Produkte elegant lösen. Diese Idee ist nicht neu, gewinnt allerdings infolge der Marktmacht des Initiators die notwendige Durchsetzungskraft.

Ein weiteres großes Manko auf dem CASE-Sektor ist die fehlende oder unzureichende Methodenunterstützung.<sup>1)</sup> Bedenkt man, daß ein Großteil der Entwickler zur Zeit keine tiefen Methodenkenntnisse und -erfahrungen besitzt, wird die Notwendigkeit einer Methodenführung transparent. Hier kann der Einsatz von Künstlicher Intelligenz Unterstützung bieten.<sup>2)</sup> Weitere Einsatzgebiete für Expertensysteme im CASE-Bereich ergeben sich im Rahmen der Anforderungsanalyse, des Systementwurfs, der Fehleranalyse sowie der Entwurfsentscheidungen und deren Bewertung.<sup>3)</sup> Ein Projekt am Lehrstuhl für Informatik der Universität zu Köln bemüht sich zur Zeit, eine Wissensbasis zur Unterstützung eines Entwicklers bei der Erstellung von Datenflußdiagrammen im Rahmen der Structured Analysis Methode zu entwickeln.

---

1) Vgl. Herzwurm /Möglichkeiten und Grenzen/ o. S.

2) Vgl. McClure /Automation/ 201, 209

3) Vgl. Gane /Computer-Aided Software Engineering/ 119- 123; Leppert, Stork /Entwicklungsumgebungen/ 234

## Literaturverzeichnis

### Ackermann /Empirie des Softwareentwurfs/

David Ackermann: Empirie des Softwareentwurfs: Richtlinien und Methoden.

In: Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut Peschke, Gabriele Rohr, Norbert A. Streitz (Hrsg.): Einführung in die Software-Ergonomie, Berlin, New York 1988, S. 253-276

### Acly /Looking Beyond CASE/

Ed Acly: Looking Beyond CASE.

In: IEEE Software, No. 2, March 1988, Vol 5, S. 39-43

### Althammer, Bernath /SAA/

Peter Althammer, Ralph Bernath: Software-Entwicklung und SAA.

In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeu-  
unterstützter Software-Entwicklung, Hallbergmoos 1988, S. 205-216

### Andresen /Warum scheitern Software-Entwicklungsumgebungen/

Thomas Andresen: Warum scheitern Software-Entwicklungsumgebungen.

In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeu-  
unterstützter Software-Entwicklung, Hallbergmoos 1988, S. 247-259

### Asam, Drenkard, Maier /Qualitätsprüfung/

Robert Asam, Norbert Drenkard, Hans-Heinz Maier: Qualitätsprüfung von Softwareprodukten, Berlin, München 1986

### Balzert /Anforderungen/

Helmut Balzert: Anforderungen an Software Engineering Environment Systeme.

In: Helmut Balzert (Hrsg.): CASE, Systeme und Werkzeuge, Mannheim, Wien, Zürich 1990, S. 103-113

### Balzert /Entwicklung/

Helmut Balzert: Die Entwicklung von Software-Systemen. Mannheim, Wien, Zürich 1989

Balzert /Prinzipien/

Helmut Balzert: Allgemeine Prinzipien des Software Engineering.  
In: Angewandte Informatik, Heft 1, 1985, S. 1-8

Barker /CASE\*Method/

Richard Barker: CASE\*Method, Entity Relationship Modelling.  
Wokingham u. a. 1989

Bauer, Schwab /Anforderungen an Hilfesysteme/

Joachim Bauer, Thomas Schwab: Anforderungen an Hilfesysteme.  
In: Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut Peschke, Gabriele Rohr, Norbert A. Streitz (Hrsg.): Einführung in die Software-Ergonomie, Berlin, New York 1988, S. 197-212

Boehm /Software-Produktion/

Barry W. Boehm: Wirtschaftliche Software-Produktion.  
In: Wolfgang Heilmann (Hrsg.): Integrierte Datenverarbeitung in der Praxis, Band 37, Wiesbaden 1986

Boehm u. a. /Improving Productivity/

Barry W. Boehm, Maria H. Penedo, E. Don Stuckle, Robert D. Williams, Arthur B. Pyster: A Software Development Environment for Improving Productivity.  
In: IEEE Computer, June 1984, Vol. 17, S. 30-42

Bröhl /SE-Umgebung/

Adolf P. Bröhl: SE-Umgebung für Informationssysteme der Bundeswehr.  
In: Online, Heft 3, 1990, S. 96-102

Chen /Entity-Relationship Approach/

Peter Chen. The Entity-Relationship Approach to Logical Data Base Design.  
Wellesley, Massachusetts 1977

Chikosky, Rubenstein /Reliability Engineering/

Elliot J. Chikofsky, Burt L. Rubenstein: CASE: Reliability Engineering for information Systems.  
In: IEEE Software, Nr. 2, March 1988, Vol. 5, S. 11-16

Codd /Relational Model of Data/

E. F. Codd: A Relational Model of Data for Large Shared Data Banks.  
In: Communications of the ACM, Number 5, June 1970, Volume 13, S.  
377-387

Coqui /Katalysator "CASE"/

Helmuth Coqui: Katalysator "CASE".  
In: Frankfurter Allgemeine Zeitung, Beilage CEBIT, 7.3.89, S. B7

Crozier u. a. /Critical analysis /

M. Crozier, D. Glass, J. G. Hughes, W. Johnston, I. McChesney: Critical  
analysis of tools for computer-aided software engineering.  
In: Information and Software Technology, No. 9, November 1989, Vol.  
31, S. 486-496

Curth, Wyss /Information engineering/

Michael Alexander Curth, Heinz Bruno Wyss: Information engineering:  
Konzeption u. praktische Anwendung.  
München, Wien, 1988, S. 193-222

Curtis, Krasner, Iscoe /A Field Study/

Bill Curtis, Herb Krasner, Neil Iscoe: A Field Study of the Software De-  
sign Process for large Systems.  
In: Communications of the ACM, No. 11, November 1988, Volume 31,  
S. 1268-1287

DeMarco /Structured Analysis/

Tom DeMarco: Structured Analysis and System Specification.  
New York 1978

DIN /Beiblatt zu DIN 66230/

Normenausschuß Informationsverarbeitung (NI) im DIN Deutsches  
Institut für Normung e.V., Normenausschuß Bauwesen (NABau) im  
DIN: Beiblatt 1 zu DIN 66230, Programmdokumentation (Stand: Ja-  
nuar 1981).  
DIN, Deutsches Institut für Normung (Hrsg.): Softwareentwicklung,  
Programmierung, Dokumentation: Normen, 3. Auflage, Berlin, Köln  
1989, S. 186-201

DIN /DIN 66230/

Normenausschuß Informationsverarbeitung (NI) im DIN Deutsches



Institut für Normung e.V., Normenausschuß Bauwesen (NABau) im  
DIN: Beiblatt 1 zu DIN 66230, Programmdokumentation (Stand: Ja-  
nuar 1981).

DIN, Deutsches Institut für Normung (Hrsg.): Softwareentwicklung,  
Programmierung, Dokumentation: Normen, 3. Auflage, Berlin, Köln  
1989, S. 166-185

DIN /DIN 66231/

Normenausschuß Informationsverarbeitung (NI) im DIN Deutsches  
Institut für Normung e.V.: DIN 66231, Pro-  
grammentwicklungsdokumentation (Stand: Oktober 1982).

DIN, Deutsches Institut für Normung (Hrsg.): Softwareentwicklung,  
Programmierung, Dokumentation: Normen, 3. Auflage, Berlin, Köln  
1989, S. 202-228

DIN /DIN 66234/

Normenausschuß Informationsverarbeitung (NI) im DIN Deutsches  
Institut für Normung e.V., Normenausschuß Ergonomie (FNErg) im  
DIN: DIN 66234, Bildschirmarbeitsplätze (Stand: Februar 1988).

DIN, Deutsches Institut für Normung (Hrsg.): Softwareentwicklung,  
Programmierung, Dokumentation: Normen, 3. Auflage, Berlin, Köln  
1989, S. 236-241

DIN /DIN 66285/

Normenausschuß Informationsverarbeitung (NI) im DIN Deutsches  
Institut für Normung e.V.: DIN 66285, Anwendungssoftware, Gütebe-  
dingungen und Prüfbestimmungen, Entwurf (Stand: November 1989).  
Berlin 1989

Eberleh /Menüauswahl/

Edmund Eberleh: Menüauswahl.

In: Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut  
Peschke, Gabriele Rohr, Norbert A. Streitz (Hrsg.): Einführung in die  
Software-Ergonomie, Berlin, New York 1988, S. 121-137

Ernst /Methodenunterstützung/

Thomas Ernst: Methodenunterstützung durch Entwicklungssysteme.  
In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten  
Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeug-  
unterstützter Software-Entwicklung, Hallbergmoos 1988, S. 39-52

Ewald /Software/

Peter Ewald: Software richtig eingekauft.  
Haar bei München 1983

Fersko-Weiß /CASE-Tools/

Henry Fersko-Weiss: CASE Tools for Designing your Applications.  
In: PC Magazine, 30 January 1990, S. 213-251

Fisher /CASE/

Alan S. Fisher: Using Software Development Tools.  
New York, Chichester, Brisbane, Toronto, Singapore 1988

Frank /Standardsoftware/

Joachim Frank: Standard Software.  
Köln-Braunsfeld 1980

Gane /Computer-Aided Software Engineering/

Chris Gane: Computer-Aided Software Engineering. The methodologies, the products, and the future.  
Englewood Cliffs, 1990

Gane, Sarson /Structured Systems Analysis/

Chris Gane, Trish Sarson: Structured Systems Analysis: tools & techniques.  
Saint Louis 1989

Gewald, Haake, Pfadler /Software Engineering/

Klaus Gewalt, Gisela Haake, Werner Pfadler: Software Engineering, Grundlagen und Technik rationeller Programmentwicklung.  
München, Wien 1977

Gibson /CASE Philosophy/

Michael Lucas Gibson: The CASE Philosophy.  
In: BYTE, No. 4, April 1989, Vol. 14, S. 209-220

Gierlach, Jankowski /Operationalisierung/

Dirk Gierlach, Ralf Jankowski: Operationalisierung der Grundsätze ergonomischer Dialoggestaltung nach DIN 66234 Teil 8 - Beispiel Textverarbeitung.  
In: Angewandte Informatik, Heft 5, 1989, 31. Jg., S. 189-196

Griese u. a. /Ergebnisse/

Joachim Griese, Günter Obelode, Paul Schmitz, Dietrich Seibt: Ergebnisse des Arbeitskreises Wirtschaftlichkeit der Informationsverwaltung.

In: zfbf, Heft 7, 1987, 39. Jg., S. 515-537

Hamilton /No easy answers/

Rosemary Hamilton: No easy answers.

In: KnowledgeWare (Hrsg.): In the News, o.O. August 1990, S. 29

Hamilton, Staff /Look before you leap/

Rosemary Hamilton, CW Staff: CASE veterans say: Look before you leap.

In: KnowledgeWare (Hrsg.): In the News, o. O. August 1990, S. 22

Hanschke, Kobilalka, Timm /Dokumentverwaltung/

Dokumentverwaltung in modernen Software-Entwicklungsumgebungen.

In: Angewandte Informatik, Heft 11/12, 1989, 31. Jg., S. 493-499

Hensel /Perspektiven/

Gerhard Hensel: Perspektiven für die 90er Jahre/Prognosen. Teilnehmerunterlagen des Diebold-Seminars: Expertensystemtechnik in CASE-Umgebungen, Frankfurt, 28. Juni 1990

Hering /Software-Engineering/

Ekbert Hering: Software-Engineering.

In: Harald Schumney (Hrsg.): Reihe Informationstechnik, Braunschweig, Wiesbaden 1984

Herzwurm /Möglichkeiten und Grenzen/

Georg Herzwurm: PC-gestützte Software-Entwicklungsumgebungen - Möglichkeiten und Grenzen am Beispiel von Excelerator, IEW/ADW und ProKit\*WORKBENCH.

In: BIFOA (Hrsg.): Teilnehmerunterlagen zum Fachseminar, Anwendungserfahrungen mit CASE und Software-Entwicklungsumgebungen, Köln, 27./28. Sept. 1990

Heuer /Rolle von Dictionary-Systemen/

Klaus Heuer: Die Rolle von Dictionary-Systemen in Software-Produktionsumgebungen.

In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeuggestützter Software-Entwicklung, Hallbergmoos 1988, S. 117-134

Hildebrand /Software Tools/

Knut Hildebrand: Software Tools - Werkzeuge für jedes Problem?  
In: PIK, Heft 11, 1989, 12 Jg., S. 258-262

Hildebrand /Software-Life-Cycle/

Knut Hildebrand: Einsatz im Software-Life-Cycle.  
In: Computer Magazin, August 1989, 18. Jg., S. 47-48

Hoppe /Prototypenentwicklung/

Heinz Ulrich Hoppe: Werkzeuge für die Prototypenentwicklung von Benutzerschnittstellen.  
In: Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut Peschke, Gabriele Rohr, Norbert A. Streit (Hrsg.): Einführung in die Software-Ergonomie, Berlin, New York 1988, S. 227-297

Hruschka /Evolutions-Zyklen/

Peter Hruschka: Mehrere Evolutions-Zyklen führen aus der Software-Krise.  
In: Computerwoche, Nr. 35, 31. August 1990, 17. Jg., S. 84-87

IBM /Neu von IBM/

IBM (Hrsg.): Neu von IBM.  
o. O., 9/1989

IEEE/ANSI /IEEE Standard Glossary/

IEEE/ANSI (Hrsg.): An American National Standard, IEEE Standard Glossary of Software Engineering Terminology, Std 729-1983  
New York 1983

Index Technology /A Guided Tour of Customizer/

Index Technology (Hrsg.): A Guided Tour of Customizer.  
Cambridge 1988

Index Technology /PC Prism/

Index Technology (Hrsg.): PC Prism.  
Cambridge 1989

Koslowski /Unterstützung partizipativer Systementwicklung/

Knut Koslowski: Unterstützung von partizipativer Systementwicklung durch Methoden des Software Engineering.

In: Der Minister für Arbeit, Gesundheit und Soziales des Landes Nordrhein-Westfalen (Hrsg.): Sozialverträgliche Technikgestaltung, Band 3, Opladen 1988

Krüger /Anwendungserfahrungen/

Anwendungserfahrungen mit einem ICASE auf der Basis IEM/IEF bei Hapag Lloyd.

In: BIFOA (Hrsg.): Teilnehmerunterlagen zum Fachseminar, Anwendungserfahrungen mit CASE und Software-Entwicklungsumgebungen, Köln, 27./28. Sept. 1990

Kulling /Problemstellung und Perspektiven/

Felicitas Kulling: CASE: Computer-Aided Software Engineering. Problemstellung und Perspektiven.

In: K.-P. Fähnrich (Hrsg.): ONLINE '89, 12. Europäische Kongreßmesse für Technische Kommunikation, Kongreß VI, Kongreßband, Velbert 1989, S. VI-1-01 - VI-1-12

Lehnhardt, Kimmel /Vollständige SPU/

Annelore Lehnhardt, Helmut Kimmel: Situationsbericht: Welches Leistungsspektrum bieten heute CASE-Produkte? Vollständige SPU - noch ein Traum.

In: Die Computer Zeitung extra, Computer Aided Software Engineering, Stuttgart 1989, S. 27-28

Leinweber /Planung und Dokumentation/

G. Leinweber: Rechnergestützte Planung und Dokumentation mit PREDICT CASE.

In: K.-P. Fähnrich (Hrsg.): ONLINE '89, 12. Europäische Kongreßmesse für Technische Kommunikation, Kongreß VI, Kongreßband, Velbert 1989, S. VI-11-01 - VI-11-13

Lektorat des B.I.-Wissenschaftsverlags /Duden Informatik/

Lektorat des B.I.-Wissenschaftsverlags unter Leitung von Hermann Engesser (Hrsg.): Duden Informatik. Mannheim, 1988

Lempp /vorhandene Software nutzen/

Pieter Lempp: Systematisches Wiederaufbereiten mit CASE und Reverse Engineering. Die vorhandene Software nutzen.  
In: Die Computer Zeitung extra, Stuttgart 1989, S. 43-45

Leppert, Stork /Entwicklungsumgebungen/

M. Leppert, B. Stork: Entwicklungsumgebungen - Status quo und Perspektiven.  
In: W. Sammer, W. Remmele (Hrsg.): Programmierumgebungen: Entwicklungswerkzeuge und Programmiersprachen, Berlin, Heidelberg 1984, S. 205-236

Löffler, Warner /Integration/

Stefan Löffler, Andr Warner: Integration von Software-Entwicklungsumgebungen.  
In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeuggestützter Software-Entwicklung. Hallbergmoos 1988, S. 29-38

Maag /Methoden/

Daniela Maag: Methoden resultieren aus dem Verständnis ihres Entwicklers.  
In: Computerwoche, Nr. 35, 31. August 1990, 17. Jg., S 27-28

Martin /Information Engineering, Book I/

James Martin: Information Engineering, Book I, Introduction.  
Englewood Cliffs, New Jersey 1990

Martin /Information Engineering, Book II/

James Martin: Information Engineering, Book II, Planning and Analysis.  
Englewood Cliffs, New Jersey 1990

Martin /Information Engineering, Book III/

James Martin: Information Engineering, Book III, Design and Construction.  
Englewood Cliffs, New Jersey 1990

Martin, McClure /Structured Techniques/

James Martin, Carma McClure: Structured Techniques: The Basis for CASE.

Revised Edition, Englewood Cliffs, New Jersey 1988

McClure /Automation/

Carma McClure: CASE is Software Automation.  
Englewood Cliffs, New Jersey 1989

McClure /CASE Experience/

Carma McClure: The CASE Experience.  
In: BYTE, Heft 4, April 1989, 14. Jg., S. 235-245

McDonnell Douglas /Application Manual/

McDonnell Douglas (Hrsg.): ProKit\*Workbench, Application Manual,  
Version 2.0.  
1. Edition, St. Louis 1989

McDonnell Douglas /Product Description/

McDonnell Douglas (Hrsg.): Prokit\*Workbench, Product Description,  
Version 2.0.  
1. Edition, St. Louis 1989

McDonnell Douglas /SSAD/

McDonnell Douglas (Hrsg.): Teilnehmerunterlagen zur Schulung:  
Structured Systems Analysis and Design SSAD  
Frankfurt, 18.-20. Juni 1990

McMenamin, Palmer /Strukturierte Systemanalyse/

Stephen M. McMenamin, John F. Palmer: Strukturierte Systemana-  
lyse.  
München, Wien, Hanser, London, Prentice Hall, 1988

Merlyn, Boone /Ins and Outs/

Vaughan Merlyn, Gregory Boone: The Ins and Outs of AD/Cycle.  
In: Datamation, No. 5, 1.3.1990, Vol 36, S. 59-64

Müßig /Software Engineering Environment System/

Michael Müßig: Das synergetische Software Engineering Environment  
System von McDonnell Douglas.  
In: Helmut Balzert (Hrsg.): CASE: Systeme und Werkzeuge. 2. Auflage,  
Mannheim, Wien, Zürich 1990, S. 449-460

Nastansky /PC-basierte Enduser-Tools/

Ludwig Nastansky: Software-Entwicklung mit PC-basierten Enduser-Tools.

In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeug-unterstützter Software-Entwicklung, Hallbergmoos 1988, S. 71-86

Orr u. a. /Methodology/

Ken Orr, Chris Gane, Edward Yourdon, Peter P. Chen, Larry L. Constantine: Methodology: The Experts Speak.

In: BYTE, No. 4, April 1989, Vol 14, S. 221-234

Österle /CASE/

Hubert Österle: Computer aided software engineering - Von Programmiersprachen zu Softwareentwicklungsumgebungen.

In: Karl Kurbel, Horst Strunz (Hrsg.): Handbuch Wirtschaftsinformatik, Stuttgart 1990, S. 346-361

Österle /Computer Integrated Software-Engineering/

Hubert Österle: Auf dem Weg zum Computer Integrated Software-Engineering.

In: Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 1: Erfolgsfaktoren werkzeug-unterstützter Software-Entwicklung. Hallbergmoos 1988, S. 9-28

Parthier /Marktbild CASE/

Ulrich Parthier: Marktbild CASE.

In: Hard and Soft 6, Nr. 10, 6/1989, S. 22-25

Prakash /Using CASE/

Jay Prakash: How Europe Is Using CASE.

In: Datamation, No. 15, 1.8.90, Vol. 36, S. 79-82

Preßmar /Strategien/

Dieter B. Preßmar: Strategien für den Einsatz von Methoden und Werkzeugen in der Softwaretechnologie.

In: Thomas Gutzwiller, Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 2: Entwicklungssysteme und 4. Generationen-Sprachen. Hallbergmoos 1988, S. 9-24



Richards /AD/Cycle/

Neil L. Richards: AD/Cycle - A new unofficial Standard?  
In: Kathy Spurr, Paul Layzell (Hrsg.): CASE on trial, Chichester 1990,  
S. 147-162

Ruf /Basis des Schnittstellen-Management-Ansatzes/

Walter Ruf: Ein Software-Entwicklungssystem auf der Basis des  
Schnittstellen-Management-Ansatzes, Für Klein- und Mittelbetriebe.  
In: H. R. Hansen, H. Krallmann, P. Mertens, A.-W. Scheer, D. Seibt, P.  
Stahlknecht, H. Strunz, R. Thom (Hrsg.): Betriebs- und Wirtschaftsin-  
formatik, Berlin, Heidelberg 1988

Sayles /Training and CASE/

Jonathan S Sayles: Training and CASE - Strange Bedfellows.  
In: KnowledgeWare (Hrsg.): In the News. o. O., August 1990, S. 23

Schäfer /Benutzerschnittstelle/

W. Schäfer: Die Benutzerschnittstelle einer integrierten Softwareent-  
wicklungsumgebung.  
In: Hans-Jörg Bullinger (Hrsg.): Software-Ergonomie '85, Mensch-  
Computer-Interaktion, Stuttgart 1985, S. 420-429

Scheer /EDV-orientierte Betriebswirtschaftslehre/

August-Wilhelm Scheer: EDV-orientierte Betriebswirtschaftslehre.  
4. Auflage, Berlin, Heidelberg, New York, Tokyo 1990, S. 123-139

Scheer /Wirtschaftsinformatik/

August-Wilhelm Scheer: Wirtschaftsinformatik, Informationssysteme im  
Industriebetrieb.  
Berlin, Heidelberg 1988

Scheibl /Kommerzielle Software-Entwicklung/

Hans-Jürgen Scheibl: Kommerzielle Software-Entwicklung, Systems-  
engineering / Software Engineering im Überblick und in praktischen  
Anwendungen.  
In: Wilfried J. Bartz (Hrsg.): Kontakt & Studium, Ebningen bei Böbling-  
en, 1989

Schmitt /Auswahl/

Günter W. Schmitt: Auswahl von Software-Entwicklungswerkzeugen:  
Analyse und Koordination der Unternehmensanforderungen, Investiti-

entscheidungen, Einführungsstrategie.

In: K.-P. Fähnrich (Hrsg.): ONLINE '89, 12. Europäische Kongreßmesse für Technische Kommunikation, Kongreß VI, Kongreßband, Velbert 1989, S. VI-5-01 - VI-5-11

Schmitz /Komplexes anschaulich darstellen/

Jürgen Schmitz: Komplexes anschaulich darstellen.

In: Die Computer Zeitung extra, Computer Aided Software Engineering, Stuttgart 1989, S. 33-34

Schmitz /Methoden, Verfahren und Werkzeuge/

Paul Schmitz: Methoden, Verfahren und Werkzeuge zur Gestaltung Rechnergestützter betrieblicher Informationssysteme (RBIS).

In: Angewandte Informatik, Heft 2, 1982, 24. Jg., S. 72-79

Schmitz, Bons, van Megen /Software-Qualitätssicherung/

Paul Schmitz, Heinz Bons, Rudolf van Megen: Software-Qualitätssicherung - Testen im Software-Lebenszyklus.

2. Auflage, Braunschweig, Wiesbaden 1983

Schmitz, Schwichtenberg, Sodeur /Kombinationsverfahren/

Paul Schmitz, Günter Schwichtenberg, Wolfgang Sodeur: Ein Kombinationsverfahren für Rangfolgeentscheidungen - unter besonderer Berücksichtigung der Auswahl-Problematik bei Rechensystemen.

Schriftenreihe des Rechenzentrums der Universität zu Köln, Heft 23, Köln 1975

Schmitz, Seibt /Einführung/

Paul Schmitz, Dietrich Seibt: Einführung in die anwendungsorientierte Informatik, Band 1: Systemtechnische Grundlagen.

3. Auflage, München 1985

Schulz /Klassifizierungs- und Bewertungsschema/

Arno Schulz: Ein Klassifizierungs- und Bewertungsschema für Software-Engineering-Werkzeuge, insbesondere für CAS-Systeme.

In: Angewandte Informatik, Heft 5, 1986, 28. Jg., . 191-197

Schulz /Software-Entwurf/

Arno Schulz: Software-Entwurf, Methoden und Werkzeuge.

2. Aufl., Wien, München 1990

Seibt /Anwendungserfahrungen/

Dietrich Seibt: Einführung: Gegenwärtige Probleme und Trends beim Einsatz von Software-Entwicklungsumgebungen.  
In: BIFOA (Hrsg.): Teilnehmerunterlagen zum Fachseminar, Anwendungserfahrungen mit CASE und Software-Entwicklungsumgebungen, Köln, 27./28. Sept. 1990

Seibt /COCOMO/

Dietrich Seibt: Das Verfahren COCOMO  
In: Dietrich Seibt (Hrsg.): Schätzverfahren für Software-Projekte - Kontrolle der Entwicklungskosten. München 1985, S. 56-86

Seibt /Function-Point-Methode/

Dietrich Seibt: Die Function-Point-Methode.  
In: Dietrich Seibt (Hrsg.): Schätzverfahren für Software-Projekte - Kontrolle der Entwicklungskosten. München 1985, S. 109-125

Seibt /Function-Point-Methode: Vorgehensweise/

Dietrich Seibt: Die Function-Point-Methode: Vorgehensweise, Einsatzbedingungen und Anwendungserfahrungen.  
In: Angewandte Informatik, Heft 1, 1987, 29. Jg., S. 3-11

Seibt /Gegenwärtige Probleme/

Dietrich Seibt: Einführung: Gegenwärtige Probleme und Trends beim Einsatz von Software-Entwicklungsumgebungen.  
In: BIFOA (Hrsg.): Teilnehmerunterlagen zum Fachseminar, Anwendungserfahrungen mit CASE und Software-Entwicklungsumgebungen. Köln, 27./28. Sept. 1990

Seibt /Phasen und Aktivitäten/

Dietrich Seibt: Phasen und Aktivitäten der Gestaltung rechnergestützter betrieblicher Informationssysteme - Teil II: Betrieb und Wartung/Pflege einer RBIS sowie Entwicklungsprozeßüberlagernde Gestaltungsaktivitäten, Essen, 1983

Seibt /Phasenkonzept/

Dietrich Seibt: Phasenkonzept.  
In: Mertens (Hrsg.): Lexikon der Wirtschaftsinformatik, Berlin u. a. 1987, S. 253-255

Seibt /RBIS Skriptum/

Dietrich Seibt: RBIS Skriptum SS 90.  
Essen, o.O., o.J.

Seibt /Systemlebenszyklus/

Dietrich Seibt: Management des Systemlebenszyklus.  
In: Mertens (Hrsg.) Lexikon der Wirtschaftsinformatik, Berlin u. a., 1987,  
S. 325-327

Skubch /CASE-Einführungen/

H. Skubch: Die Grenzen von CASE: Warum zahlreiche CASE-Einführungen scheitern.  
In: K.-P. Fähnrich (Hrsg.): ONLINE '89, 12. Europäische Kongreßmesse für Technische Kommunikation, Kongreß VI, Kongreßband, Velbert 1989, S. VI-3-01 - VI-3-24

Sneed /automatische Qualitätssicherung/

Harry Sneed: Modell eines Systems für die automatische Qualitätssicherung.  
In: Thomas Gutzwiller, Hubert Österle (Hrsg.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band2: Entwicklungssysteme und 4. Generationen-Sprachen, Hallbergmoos 1988, S. 79-98

Software-Prüfstelle des TÜV Berlin /Rahmenprüfplan/

Software-Prüfstelle des Technischen Überwachungs-Verein Berlin e. V. (Hrsg.): Rahmenprüfplan für Software-Prüfungen gemäß DIN V 66285 der Software-Prüfstelle des Technischen Überwachungs-Verein Berlin e. V.,  
3. Fassung, o. O., August 1986

Sommerville /Software Engineering/

Ian Sommerville: Software Engineering.  
Bonn 1986

Sommerville, Morrison /Software Development/

Ian Sommerville, Ron Morrison: Software Development with Ada.  
Workingham 1987

Stahlknecht, Warner /Untersuchung/

Peter Stahlknecht, Andr Warner: Stand der Entwicklung und des Ein-

satzes von Softwareentwicklungswerkzeugen - Ergebnisse einer empirischen Untersuchung.  
Beitrag Nr. 8602 des Fachbereichs Wirtschaftswissenschaften der Universität Osnabrück. Osnabrück 1986

Stobbe /Softwareentwicklungsumgebungen/

Christine Stobbe: Softwareentwicklungsumgebungen.  
In: W. Sammer, W. Remmele (Hrsg.): Programmierumgebungen: Entwicklungswerkzeuge und Programmiersprachen, Berlin, Heidelberg 1984, S. 31-59

Stork /Toolintegration/

Burkhard Stork: Toolintegration in Software-Entwicklungsumgebungen.  
In: Angewandte Informatik, Heft 2, 1985, 27 Jg., S. 49-57

Strunz /Entscheidungstabellentechnik/

Horst Strunz: Entscheidungstabellentechnik, Grundlagen und Anwendungsmöglichkeiten bei der Gestaltung rechnergestützter Informationssysteme.  
In: E. Grochla, P. Schmitz, N. Szyperski (Hrsg.): Reihe Betriebsinformatik. München, Wien 1977

Sullivan-Trainor /Buyers' Scorecard/

Michael L. Sullivan-Trainor: Buyers' Scorecard, Knowledgeware bests front-end reveals.  
In: Computerworld, April 9, 1990, S. 68-69

Timmermann /LISP-CASE/

Gerd Timmermann: LISP-CASE - wissensbasiertes Software Engineering.  
In: Office Management, Heft 11, 1986, S. 1106-1108

Tobiasch /Anforderungen/

R. Tobiasch: Anforderungen an Software-Entwicklungsumgebungen und Programmiersprachen.  
In: W. Sammer, W. Remmele (Hrsg.): Programmierumgebungen: Entwicklungswerkzeuge und Programmiersprachen, Berlin, Heidelberg 1984, S. 1-29

Vetter /Aufbau betrieblicher Informationssysteme/

Max Vetter: Aufbau betrieblicher Informationssysteme mittels konzept-

tioneller Datenmodellierung.  
Stuttgart 1986

Vetter /Konzeptionelle Datenmodellierung/

Max Vetter: Konzeptionelle Datenmodellierung.  
In: Karl Kurbel, Horst Strunz (Hrsg.): Handbuch Wirtschaftsinformatik,  
Stuttgart 1990, S. 384-401

Vonk /Prototyping/

Roland Vonk: Prototyping, The effective use of CASE technology.  
New York u. a. 1990

Wallmüller /Methode und Werkzeug/

Ernest Wallmüller: Eine Methode und ein Werkzeug für den Softwareentwurf.  
In: Angewandte Informatik, Heft 10, 1985, 27. Jg.,  
S. 424-427

Warner, Woschinski /Die Qual der Wahl/

André Warner, Sigrid Woschinski: Die Qual der Wahl.  
In: Computer Magazin, Oktober 1989, 18. Jg., S. 58-59

Wedekind /Rapid Prototyping/

Hartmut Wedekind: Ein Experiment zum Rapid Prototyping.  
In: Angewandte Informatik, Heft 2, 1985, 27. Jg.,  
S 58-61

Wirtz /Softwareentwurf/

Klaus Werner Wirtz: Methoden und Werkzeuge für den Softwareentwurf.  
In: Karl Kurbel, Horst Strunz (Hrsg.): Handbuch Wirtschaftsinformatik,  
Stuttgart 1990, S. 324-343

Yourdon /Techniques/

Edward Yourdon: Techniques of Program Structure and Design.  
Englewood Cliffs, New Jersey 1975

Yourdon, Constantine /Structured Design/

Edward Yourdon, Larry L. Constantine: Structured Design, Fundamentals of a Discipline of Computer Program and Systems Design.

Second Edition, New York 1978

Zaleski /Software-Wartung/

Mario Zaleski: CASE hilft bei der Software-Wartung.  
In: Online, Heft 3, März 1989, S. 29-30

Zwerina /Masken und Formulare/

Harald Zwerina: Masken und Formulare.  
In: Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut Peschke, Gabriele Rohr, Norbert A. Streitz (Hrsg.): Einführung in die Software-Ergonomie, Berlin, New York 1988, S. 163-174

LEHRSTUHL FÜR INFORMATIK  
DER UNIVERSITÄT ZU KÖLN

REGIONALES RECHENZENTRUM  
AN DER UNIVERSITÄT ZU KÖLN

PROF. DR. PAUL SCHMITZ

ANLAGE ZUM  
INFORMATIK-FACHBERICHT 91/1

Georg Herzwurm  
Dirk Berkau

Auswahl PC-gestützter Software-Entwicklungs-  
umgebungen - dargestellt am Beispiel von  
Excelerator, Information Engineering Workbench,  
ProKit\*WORKBENCH und Systems Engineer

Dieser Fachbericht basiert auf einer Diplomarbeit von Dirk Berkau, die am Lehrstuhl für Informatik angefertigt und im Dezember 1990 der Wirtschafts- und Sozialwissenschaftlichen Fakultät der Universität zu Köln vorgelegt worden ist.

Georg Herzwurm ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Informatik der Universität zu Köln.

© Alle Rechte vorbehalten

Albertus-Magnus-Platz, D - 5000 Köln 41



# Kriterienkatalog

Vorwort	4
1 Generelle Kriterien für die Auswahl von Standard- software	5
1.1 Vorausgesetzte Konfiguration	5
1.2 Installierung	9
1.3 Modifikationsmöglichkeiten	13
1.4 Offenheit	15
1.5 Benutzerfreundlichkeit	17
1.5.1 Verständlichkeit	17
1.5.2 Benutzbarkeit	19
1.5.3 Hilfefunktion	25
1.5.4 Fehlerbehandlung	29
1.5.5 Antwortzeitverhalten	33
1.6 Dokumentation und Produktbeschreibung	35
1.7 Einführungsunterstützung	38
1.8 Konditionengestaltung	40
1.9 Anbieterspezifische Kriterien	42
2 Spezielle Kriterien für die Auswahl von Software- Entwicklungsumgebungen	44
2.1 Unterstützung der Phasen des Software- Lebenszyklusses	44
2.1.1 Entwurfs- und Analysephase	44
2.1.2 Realisierungsphase	52
2.1.3 Erprobungs- und Konsolidierungsphase	55
2.1.4 Pflege- und Wartungsphase	57
2.2 Unterstützung phasenübergreifender Aufgaben	65
2.2.1 Informationsverwaltung	65
2.2.2 Dokumentation	72
2.2.3 Qualitätssicherung	77
2.2.4 Prototyping	83
2.2.5 Projektmanagement	88
2.2.6 Konfigurationsmanagement	91
2.2.7 Allgemeine Bürodienste	93

2.3	Gestaltungsorientierte Auswahlkriterien	95
2.3.1	Methoden	95
2.3.1.1	Structured Analysis (SA)	95
2.3.1.1.1	Benutzerführung	95
2.3.1.1.2	Datenflußdiagramm	96
2.3.1.1.3	Data Dictionary	102
2.3.1.1.4	Entity-Relationship Diagramm	102
2.3.1.1.5	Prozeßspezifikation	102
2.3.1.1.6	Structured English, Pseudo- code	103
2.3.1.1.7	Entscheidungstabellen, Ent- scheidungs bäume	104
2.3.1.2	Entity-Relationship Ansatz	107
2.3.1.2.1	Benutzerführung	107
2.3.1.2.2	Entity-Relationship Diagramm	108
2.3.1.2.3	Data Dictionary	115
2.3.1.2.4	Datenstrukturen	116
2.3.1.3	Structured Design (SD)	118
2.3.1.3.1	Benutzerführung	118
2.3.1.3.2	Datenflußdiagramm	119
2.3.1.3.3	Structure Chart	120
2.3.1.3.4	Transformation	123
2.3.1.4	Structured Analysis and Design Techniques (SADT)	124
2.3.1.4.1	Benutzerführung	124
2.3.1.4.2	Diagramm	125
2.3.1.5	Jackson System Development (JSD)	127
2.3.1.5.1	Benutzerführung	127
2.3.1.5.2	Diagramme	128
2.3.1.6	Hierarchy plus Input-Process- Output Methode (HIPO)	131
2.3.1.6.1	Benutzerführung	131
2.3.1.6.2	Diagramme	132
2.3.1.7	Warnier-Orr Methode	134
2.3.1.7.1	Benutzerführung	134
2.3.1.7.2	Diagramm	135

2.3.1.8	Information Engineering	137
2.3.1.8.1	Benutzerführung	137
2.3.1.8.2	Diagramme	138
2.3.2	Kapazität	142
2.3.3	Outputqualität	144
2.3.4	Integration	147

# Kriterienkatalog

## Vorwort

Im Kriterienkatalog werden die Eigenschaften der SEUs aufgelistet, die einen Einfluß auf eine Investitionsentscheidung haben könnten.

Der Beurteilende sollte vor der Bewertung der einzelnen Kriterien diese nach seinen individuellen Bedürfnissen gewichten. Erst daran anschließend sollten die Fragen des Kriterienkatalogs beantwortet werden. Auf diese Weise wird eine ergebnisorientierte Gewichtung vermieden.

Die Operationalisierung des Kriterienkatalogs orientiert sich an einer Nutzwertanalyse: es sind Zielerreichungsgrade von 0 bis 4 zu vergeben.

Mit Hilfe eines Auswertungsbogens werden die Einzelwertungen zu einer Gesamtwertung verdichtet. Die Berechnungen können bei Verwendung eines Tabellenkalkulationsprogramms automatisiert werden.

1 Generelle Kriterien für die Auswahl von Standardsoftware

1.1 Vorausgesetzte Konfiguration<sup>1)</sup>

Hauptspeicher

Wieviel Hauptspeicher wird von der SEU benötigt?<sup>2)</sup>

Mindestanforderung: \_\_\_\_\_ KB

4: weniger als 640 KB

2: weniger als 4 MB

0: mehr als 8 MB

Basis-Hauptspeicher

Wieviel Speicherplatz benötigt die SEU im konventionellen Hauptspeicher?

Speicherplatz: \_\_\_\_\_ KB

4: weniger als 400 KB

2: weniger als 550 KB

0: mehr als 570 KB

externer Speicher

Wieviel Bedarf an externen Speicher benötigt die SEU?

4: weniger als 10 MB

2: weniger als 50 MB

0: mehr als 100 MB

---

1) Vgl. DIN /DIN 66285/ 5

2) Vgl. Frank /Standardsoftware/ 37-38



### Speicherbedarf der Projekte

Speicherbedarf der mit der SEU entwickelten Projekte

Speicherbedarf eines Fallbeispiels (Bspw. LFI-BANK)

\_\_\_\_\_ KB

4: das Fallbeispiel benötigt wenig Speicherplatz

2:

0: das Fallbeispiel benötigt sehr viel Speicherplatz



### Co-Prozessor

Ist zum Ablauf ein mathematischer Co-Prozessor erforderlich?

4: die SEU hat auch ohne mathematischen Co-Prozessor eine akzeptable Performance

2: die SEU ist auch ohne mathematischen Co-Prozessor lauffähig, ein solcher ist aber aus Performancegründen dringend zu empfehlen

0: die SEU ist ohne mathematischen Co-Prozessor nicht lauffähig

### Grafikkarte

Welche Grafikkarte wird von der SEU vorausgesetzt?

0 Hercules-Karte

0 EGA-Karte

0 CGA-Karte

0 VGA-Karte

4: alle Karten werden unterstützt

2: einige Karten werden unterstützt

0: nur eine Karte wird unterstützt

### Drucker

Welche Drucker werden von der SEU unterstützt?

0 9-Nadel-Matrixdrucker

0 24-Nadel-Matrixdrucker

0 Laserdrucker

0 Plotter

4: es wird eine umfangreiche Auswahl von Druckern und Plottern unterstützt

2: es eine Auswahl von Druckern und/oder Plottern unterstützt

0: es wird nur eine geringe Auswahl von Druckern und/oder Plottern unterstützt



Betriebssystem

Auf welchen Betriebssystemen läuft die SEU?

0 MS-DOS

0 OS/2

0 Unix

4: sie ist auf allen für PCs zur Verfügung stehenden Betriebssystemen lauffähig

2: sie ist auf mehreren für PCs zur Verfügung stehenden Betriebssystemen lauffähig

0: sie ist nur auf einem Betriebssystem lauffähig



## 1.2 Installierung

### Voraussetzungen

Kann die Installierung vom Entwickler selbst durchgeführt werden oder muß die SEU vom Anbieter installiert werden?<sup>1)</sup>

- 4: der Entwickler kann die Installierung leicht mit Hilfe der Dokumentation durchführen.
- 2: der Entwickler kann die Installierung nur mit Unterstützung des Anbieters durchführen
- 0: der Anbieter muß die Installierung durchführen

### Automatisierungsgrad

Wie hoch ist der Automatisierungsgrad der Installierung?

- 4: die Hardware- und Softwarekonfiguration wird selbständig erkannt
- 2: die Hardware- und Softwarekonfiguration wird interaktiv abgefragt
- 0: die Installierung erfolgt manuell

---

1) Vgl. Frank /Standardsoftware/ 39; Software-Prüfstelle des TÜV Berlin /Rahmenprüfplan/ 4

#### Abbruchmöglichkeit

Kann die Installierungsprozedur abgebrochen werden? Wird nach dem Abbruch das System wieder in den Zustand vor Beginn der Installation gebracht?

- 4: es wird der Zustand vor Beginn der Installation hergestellt
- 2: es werden einige Dateien installiert
- 0: es besteht die Gefahr, daß Systemdateien verändert werden

#### Rücksprung

Kann während der Installierungsprozedur zu Menüs des Installierungsprogramms zurückgesprungen werden, um Eingaben zu korrigieren?

- 4: alle Installierungseingaben können während der Prozedur jederzeit korrigiert werden
- 2: bedeutende Installierungseingaben können an bestimmten Stellen in der Prozedur korrigiert werden
- 0: es kann keine Korrektur vorgenommen werden

#### Dokumentation

Wie werden die Installierungseinstellungen dokumentiert?

- 4: am Ende eines jeden Installierungsschritts werden die gewählten Einstellungen angezeigt
- 2: erst am Ende der Installation werden die gewählten Einstellungen angezeigt
- 0: die Installationen werden vom Installierungsprogramm nicht angezeigt

### Installierungskontrolle

Kann nach erfolgter Installierung deren Richtigkeit übergeprüft werden?<sup>1)</sup>

- 4: die Richtigkeit der Installierung wird automatisiert geprüft, das Ergebnis mitgeteilt
- 2: die Richtigkeit kann anhand von vorbereiteten Prüffällen getestet werden
- 0: die Richtigkeit der Installierung kann nicht geprüft werden

### Zeitbedarf der Installierung

Wie groß ist der Zeitbedarf einer Installation?<sup>2)</sup>

\_\_\_\_\_ min

- 4: unter 15 Minuten
- 2: unter 30 Minuten
- 0: mehr als 30 Minuten

### Teilinstallierung

Können nur Teile der SEU installiert werden?

- 4: eine Teilinstallierung ist möglich
- 2: eine Teilinstallierung ist möglich, zuvor muß jedoch einmal die gesamte SEU installiert worden sein.
- 0: nein

---

1) Vgl. DIN /DIN 66285/ 6; Software-Prüfstelle des TÜV Berlin /Rahmenprüfplan/ 8

2) Vgl. Frank /Standardsoftware/ 39



### Deinstallierung

Kann die SEU kurzfristig deinstalliert werden?

4: eine bequeme Deinstallierung ist möglich, bestehende Projekte und ihre Verbindung zur SEU werden nicht gelöscht

2: eine bequeme Deinstallierung ist möglich, bestehende Projekte werden nicht gelöscht, jedoch müssen sie bei einer erneuten Installation der SEU erneut festgelegt werden

0: eine Deinstallierung ist aufwendig

### 1.3 Modifikationsmöglichkeiten

#### Konfigurationsänderung

Kann die Konfiguration nach Abschluß der  
Installierung geändert werden?<sup>1)</sup>

4: eine Konfigurationsänderung ist nachträglich  
menügesteuert möglich, ohne die vollständige  
Installierungsprozedur durchlaufen zu müssen

2: eine Konfigurationsänderung ist durch eine  
teilweise Installierungsprozedur möglich

0: zur Konfigurationsänderung ist eine erneute  
Installierungsprozedur erforderlich

#### Änderung des Datenmodells

Kann das Datenmodell der Entwicklungsdatenbank  
vom Entwickler geändert werden?<sup>2)</sup>

4: Änderungen können mit der SEU durchgeführt  
werden

2: Änderungen können mit einem erwerbba-  
ren Zusatzprodukt durchgeführt werden

0: Änderungen sind nicht möglich

---

1) Vgl. Curth Wyss /Information engineering/ 197; Frank  
/Standardsoftware/ 39

2) Vgl. Index Technology /A Guided Tour of Customizer/ 6-  
9, 19; Index Technology /Customizer/ 1

#### Änderung der Beschreibungswerkzeuge

Können Beschreibungsmittel hinzugefügt, bestehende geändert werden?<sup>1)</sup>

4: Änderungen können mit der SEU durchgeführt werden

2: Änderungen können mit einem erwerbba-  
ren Zusatzprodukt durchgeführt werden

0: Änderungen sind nicht möglich

#### Änderung der Benutzeroberfläche

Können Änderungen der Benutzeroberfläche vorgenommen werden?

4: Änderungen können mit der SEU durchgeführt werden

2: Änderungen können mit einem erwerbba-  
ren Zusatzprodukt durchgeführt werden

0: Änderungen sind nicht möglich

#### Portabilität

Ist die SEU auf verschiedenen Hardwarekonfigurationen lauffähig?

4: die SEU ist auf verschiedenen Rechnern und Rechnerklassen lauffähig

2: die SEU ist auf verschiedenen Rechnern lauffähig

0: die SEU ist auf einen bestimmten Rechner zugeschnitten

---

1) Vgl. Index Technology /A Guided Tour of Customizer/ 12-17, 19; Index Technology /Customizer/ 2; Stobbe /Softwareentwicklungsumgebungen/ 40

#### 1.4 Offenheit

##### Entwicklungswerkzeuge

Wieviel Schnittstellen zu anderen Entwicklungswerkzeugen sind in der SEU enthalten?<sup>1)</sup>

4: es gibt zahlreiche Schnittstellen zu Entwicklungswerkzeugen unterschiedlicher Art

2: es gibt nur Schnittstellen zu einer Art von Entwicklungswerkzeugen

0: es gibt keine Schnittstellen zu anderen Entwicklungswerkzeugen

##### Standardsoftware

Wieviel Schnittstellen gibt es zu Standardsoftwareprodukten?<sup>2)</sup>

4: es kann mit eine Reihe unterschiedlicher Standardsoftwareprodukte kommuniziert werden

2: es kann nur mit einer begrenzten Anzahl von Standardsoftwareprodukten kommuniziert werden

0: es kann mit keinem Standardsoftwareprodukt kommuniziert werden

---

1) Vgl. Österle /CASE/ 358-359

2) Vgl. Krüger /Anwendungserfahrungen/o. S.; Ruf /Basis des Schnittstellen-Mangement-Ansatzes/ 231

fremde Repositories

Ist der Zugriff auf Repositories anderer SEU möglich?<sup>1)</sup>

- 4: ein direkter Zugriff auf fremde Repositories ist möglich
- 2: ein indirekter Zugriff auf fremde Repositories ist möglich
- 0: ein Zugriff auf fremde Repositories ist nicht möglich

eigenes Repository

Ist ein Zugriff auf das eigene Repository während des Betriebs der SEU möglich?

- 4: ein direkter Zugriff auf das eigene Repository ist möglich
- 2: ein indirekter Zugriff auf das eigene Repository ist möglich
- 0: ein Zugriff auf das eigene Repository ist nicht möglich.

---

1) Vgl. McClure /Automation/ 21



## 1.5 Benutzerfreundlichkeit

### 1.5.1 Verständlichkeit

#### Dialoggestaltung

Ist die Dialoggestaltung verständlich oder müssen zusätzliche Hilfen in Anspruch genommen werden?<sup>1)</sup>

- 4: die Dialoggestaltung ist so verständlich, daß Hilfen nicht in Anspruch genommen werden müssen
- 2: in einigen Fällen ist die Dialoggestaltung unverständlich, so daß Hilfen in Anspruch genommen werden müssen
- 0: die Dialoggestaltung ist so unverständlich, daß ständig Hilfen in Anspruch genommen werden müssen

#### Menügestaltung

Sind die Menüs und Bildschirmmasken übersichtlich gestaltet?<sup>2)</sup>

- 4: übersichtlich gestaltete Menüs, bei denen in der Anwendung jeweils nur eine Kommando-  
gruppe verfeinert wird
- 2: übersichtlich gestaltete Menüs, eine Ver-  
feinerung von Kommandogruppen ist nicht  
möglich
- 0: unübersichtliche Menüs

---

1) Vgl. Koslowski /Unterstützung partizipativer Systementwicklung/ 103

2) Vgl. DIN /DIN 66285/ 6; Schäfer /Benutzerschnittstelle/ 422

#### Aussagekraft

Wie aussagekräftig sind die Kommando- und Funktionsnamen?<sup>1)</sup>

4: sie wirken bereits beim ersten Gebrauch der SEU aussagekräftig und einprägsam

2: sie wirken erst nach mehrmaligem Gebrauch der SEU aussagekräftig und einprägsam

0: sie wirken selbst nach mehrmaligem Gebrauch der SEU nicht aussagekräftig und sind nicht einprägsam

#### Nationalsprache

Wird die entsprechende Nationalsprache unterstützt?

4: das Programm und die Dokumentation sind in der Nationalsprache verfügbar

2: das Programm und die Dokumentation sind 1:1 aus einer Fremdsprache übersetzt

0: das Programm und die Dokumentation sind nur in einer Fremdsprache verfügbar

---

1) Vgl. DIN /DIN 66285/ 6; McClure /Automation/ 217, 231; Schäfer /Benutzerschnittstelle/ 423

### 1.5.2 Benutzbarkeit

#### Konsistenz

Ist die Benutzeroberfläche konsistent? Ist die Bildschirmgestaltung und die Funktionstastenbelegung einheitlich?<sup>1)</sup>

4: konsistente Benutzeroberfläche

2: wenige Inkonsistenzen

0: viele Inkonsistenzen

#### SAA/CUA-Standard

Entspricht die Benutzeroberfläche dem SAA/CUA-Standard der IBM?<sup>2)</sup>

4: die SEU entspricht dem SAA/CUA-Standard

2: die SEU entspricht dem SAA/CUA-Standard nur grob

0: die SEU entspricht dem SAA/CUA-Standard nicht

---

1) Vgl. Ackermann /Empirie des Softwareentwurfs/ 253-254; Koslowski /Unterstützung partizipativer Systementwicklung/ 103

2) Vgl. Althammer, Bernard /SAA/ 207; Preßmar /Strategien/ 16

#### Erwartungskonformität

Entspricht die SEU in der Benutzung den Erwartungen des Entwicklers?<sup>1)</sup>

- 4: sie entspricht voll den Erwartungen
- 2: sie entspricht zu weitgehenden Teilen den Erwartungen
- 0: sie entspricht nur in geringem Umfang den Erwartungen

#### Eingabe

Können die Instrumente zur Eingabe gewählt werden? Unterstützt die SEU Maussteuerung, Menütechnik und Funktionstasten?

- 4: alle Eingabeinstrumente werden permanent unterstützt
- 2: alle Eingabeinstrumente werden nicht immer unterstützt
- 0: alle Eingabeinstrumente werden nie gleichzeitig unterstützt

#### Änderbarkeit der BOF

Kann die BOF entsprechend individuellen Bedürfnissen des Entwicklers gestaltet werden?

- 4: die SEU eröffnet dem Entwickler große Freiheiten bei der Gestaltung seiner individuellen Benutzeroberfläche
- 2: die SEU eröffnet dem Entwickler geringe Freiheiten bei der Gestaltung seiner individuellen Benutzeroberfläche oder die Änderbarkeit erfolgt über ein Zusatzprodukt
- 0: die SEU eröffnet keine Gestaltungsmöglichkeiten

---

1) Vgl. DIN /DIN 66234 Teil 8/ 4-5



### Flexibles Benutzermodell

Wird die Anpaßbarkeit an die Erfahrung des Entwicklers unterstützt?<sup>1)</sup>

- 4: eine mehrstufige Differenzierung der Benutzermodelle ist möglich
- 2: die Differenzierung konzentriert sich auf die Alternative der Menübenutzung und der Eingabe von Schlüsseln
- 0: eine Differenzierung erfolgt nicht



### Orientierung

Ist eine Orientierung in der SEU möglich? Wird in jedem Menü der Standpunkt im System angezeigt? Werden Überblicksdiagramme (orientation map), Statuszeilen unterstützt? Wird das bearbeitete Inkrement hervorgehoben?<sup>2)</sup>

- 4: es werden umfangreiche Orientierungshilfen zur Verfügung gestellt
- 2: es werden einige Orientierungshilfen zur Verfügung gestellt
- 0: es werden keine Orientierungshilfen zur Verfügung gestellt.

---

1) Vgl. Eberleh /Menüauswahl/ 132; McClure /Automation/ 221; Schäfer /Benutzerschnittstelle/ 421; Sommerville, Morrison /Software Development/ 42;

2) Vgl. DIN /DIN 66285/ 6; Eberleh /Menüauswahl/ 127; Gane /Computer-Aided Software Engineering/ 132; McClure /Automation/ 219

#### Quittierung

Werden die Kommandos des Entwicklers quittiert?<sup>1)</sup>

4: eine Quittierung erfolgt immer

2: eine Quittierung erfolgt nicht immer

0: eine Quittierung erfolgt nie

#### Lokalität

Wird die Möglichkeit der Informationssammlung auf einer Bildschirmseite unterstützt?<sup>2)</sup>

4: die SEU unterstützt die Fenstertechnik

2: die SEU unterstützt die Teilbarkeit des Bildschirms

0: die SEU unterstützt weder Fenstertechnik noch ist eine Bildschirmteilung möglich

#### Grafikqualität

Wie ist die Auflösung in den Grafiken auf dem Bildschirm?

4: die Grafiken erscheinen auf dem Bildschirm in hoher Auflösung

2: die Grafiken erscheinen auf dem Bildschirm in mittlerer Auflösung

0: die Grafiken erscheinen auf dem Bildschirm mit geringer Auflösung

---

1) Vgl. Gierlach, Jankowski /Operationalisierung/ 191

2) Vgl. Andresen /Warum scheitern Software-Entwicklungsumgebungen/ 254; Gierlach, Jankowski /Operationalisierung/ 191; Schäfer /Benutzerschnittstelle/ 421

#### WYSIWYG

Entspricht die Darstellung auf dem Bildschirm der Druckausgabe?<sup>1)</sup>

- 4: die Bildschirmdarstellung entspricht exakt der Druckausgabe
- 2: die Bildschirmdarstellung weicht nur geringfügig von der Druckausgabe ab
- 0: die Bildschirmausgabe weicht erheblich von der Druckausgabe ab

#### Verbinden von Symbolen

Wie komfortabel können im Diagramm Verbindungen zwischen Symbolen erstellt werden?

- 4: komfortabel
- 2: ja
- 0: nein

#### Anklicken

Wie komfortabel erfolgt das Anklicken von Symbolen?

- 4: es können mehrere Symbole zusammen angeklickt werden
- 2: ein Symbol kann an beliebiger Stelle angeklickt werden
- 0: es müssen bestimmte Stellen des Symbols getroffen werden, um es anzuklicken

---

1) Vgl. Schulz /Software-Entwurf/ 56



### Steuerbarkeit

Ist die Vorgehensweise des Entwicklers frei gestaltbar oder muß er einem vom CASE-Anbieter vorgezeichneten Weg folgen?<sup>1)</sup>

4: der Entwickler ist in seiner Vorgehensweise vollkommen frei

2: der Entwickler wird nicht geführt, ihm bleiben jedoch wegen der geringen Funktionsauswahl wenig Gestaltungsmöglichkeiten

0: der Entwickler hat keine Steuerungsmöglichkeiten

---

1) Vgl. Gierlach, Jankowski /Operationalisierung/ 191



### 1.5.3 Hilfefunktion<sup>1)</sup>

#### Notwendigkeit

Ist das Hilfesystem notwendig?

4: ein Hilfesystem wird nicht benötigt

2: ein Hilfesystem wird nur in Ausnahmefällen benötigt

0: ein Hilfesystem wird permanent benötigt

#### Effektivität

Wie hilfreich ist das Hilfesystem?

4: das Hilfesystem bietet alle Informationen zur Problemlösung

2: das Hilfesystem bietet fast alle Informationen, manchmal sind jedoch weitere Unterlagen zur Problemlösung hinzuzuziehen

0: das Hilfesystem bietet wenig Informationen, zur Problemlösung müssen jedesmal weitere Unterlagen hinzugezogen werden

---

1) Die folgenden Fragen beziehen sich nur auf das Hilfesystem, das online zur Verfügung steht. Die Hilfe in der Dokumentation wird unter dem Punkt "1.6 Dokumentation" behandelt.



### Folgeaktionen

Zeigt das Hilfesystem zu Situationen mögliche Folgeaktionen auf?<sup>1)</sup>

- 4: das Hilfesystem zeigt zu jeder Situation sämtliche Folgeaktionen auf
- 2: das Hilfesystem zeigt nicht zu jeder Situation sämtliche Folgeaktionen auf oder es zeigt nicht immer sämtliche Folgeaktionen auf
- 0: das Hilfesystem zeigt keine Folgeaktionen



### Dynamik

Berücksichtigt die Hilfefunktion den Kontext, aus dem sie aufgerufen wird?<sup>2)</sup>

- 4: die Hilfefunktion berücksichtigt die Funktion, aus der sie aufgerufen wird
- 2: die Hilfefunktion berücksichtigt das Modul, aus der sie aufgerufen wird
- 0: die Hilfefunktion berücksichtigt den Kontext nicht

---

1) Vgl. Tobiasch /Anforderungen/ 16

2) Vgl. Bauer, Schwab /Anforderungen an Hilfesysteme/ 198-200



### Oberfläche

Entspricht die Gestaltung und Bedienbarkeit der Hilfefunktion der Gestaltung und Bedienbarkeit der SEU?<sup>1)</sup>

- 4: die Gestaltung und die Bedienbarkeit der Hilfefunktion ist mit der SEU identisch
- 2: die Gestaltung und die Bedienbarkeit der Hilfefunktion ist fast mit der SEU identisch
- 0: die Gestaltung und die Bedienbarkeit der Hilfefunktion ist unterschiedlich zur SEU



### Benutzerprofil

Berücksichtigt die Hilfefunktion das Benutzerprofil des Entwicklers?<sup>2)</sup>

- 4: die Informationen sind an die individuelle Erfahrung des Entwicklers angepaßt
- 2: die Informationen sind an typisierte Benutzerprofile des Entwicklers angepaßt
- 0: die Informationen sind nicht an der Erfahrung des Entwicklers ausgerichtet

---

1) Vgl. Bauer, Schwab /Anforderungen an Hilfesysteme/ 200

2) Vgl. Bauer, Schwab /Anforderungen an Hilfesysteme/ 198



aktiv/passiv-Wahl

Verhält sich die Hilfefunktion aktiv oder passiv zum Benutzerverhalten des Entwicklers?<sup>1)</sup>

4: das Hilfesystem verhält sich entsprechend der Notwendigkeit einer Hilfe aktiv oder passiv

2: das Hilfesystem verhält sich aktiv, ist jedoch auf passives Verhalten umschaltbar

0: das Hilfesystem verhält sich entweder aktiv oder passiv

---

1) Vgl. Bauer, Schwab /Anforderungen an Hilfesysteme/ 198-202; Gierlach, Jankowski /Operationalisierung/ 191

#### 1.5.4 Fehlerbehandlung

##### Robustheit

Ist die SEU gegen unvorhergesehene Benutzer-  
eingaben des Entwicklers robust?<sup>1)</sup>

4: die SEU ist gegen falsche Eingaben von  
alphanumerischen Zeichen und gegen falschen  
Funktionstastengebrauch robust

2: die SEU ist gegen falsche Eingaben von  
alphanumerischen Zeichen robust

0: die SEU ist nicht robust

##### Zuverlässig

Ist die SEU zuverlässig?

4: hoch

2: mittel

0: schlecht

##### Backup-Funktion

Gibt es eine Backup-Funktion zur Sicherung der  
Informationen bei Systemzusammenbruch, die in  
bestimmten Intervallen automatisiert aufgeru-  
fen wird.

4: es gibt eine solche Backup-Funktion

0: es gibt keine Backup-Funktion

---

1) Vgl. Balzert /Entwicklung/ 488; DIN /DIN 66285/ 6; DIN  
/DIN 66234 Teil 8/ 5; Koslowski /Unterstützung partizi-  
pativer Systementwicklung/ 103

### Bestätigung

Wird bei Aktionen mit schwerwiegenden Folgen (Verlassen des Programms oder des Moduls ohne zu sichern) eine Bestätigung gefordert?<sup>1)</sup>

4: es wird immer eine Bestätigung gefordert

2: es wird manchmal eine Bestätigung gefordert

0: eine Bestätigung wird nicht verlangt

### Auffälligkeit

Sind die Fehlermeldungen auffällig gestaltet?

4: sie erscheinen auffällig sichtbar auf dem Bildschirm und werden mit einem akustischen Signal begleitet, wichtige Fehlermeldungen müssen bestätigt werden; fehlerhafte Symbole in der Grafik werden visuell hervorgehoben

2: alle Fehlermeldungen erscheinen auffällig sichtbar und werden mit einem akustischen Signal begleitet

0: sie sind nicht auffällig gestaltet

---

1) Vgl. DIN /DIN 66285/ 7; McClure /Automation/ 226-227

### Erklärung

Werden die Ursachen des Fehlers erklärt?<sup>1)</sup>

4: die Fehler werden verständlich erklärt

2: die Fehler werden erklärt, ab und zu wird jedoch die Hilfe der Dokumentation benötigt

0: die Fehler werden nicht erklärt

### Korrekturhilfen

Werden in den Fehlerhinweisen Korrekturhinweise geliefert?<sup>2)</sup>

4: die Fehlerhinweise bieten Hilfen zur Korrektur des Fehlers

2: die Fehlerhinweise geben einen Hinweis auf Textstellen in der Dokumentation, die Korrekturvorschläge bieten

0: es werden keine Korrekturhilfen geboten

---

1) Vgl. DIN /DIN 66285/ 7; Gierlach, Jankowski /Operationalisierung/ 192; McClure /Automation/ 217, 219, 226-227;

2) Vgl. DIN /DIN 66285/ 7; McClure /Automation/ 219, 226-227



Korrekturaufwand

Können Befehle storniert (UNDO-Funktion) werden?<sup>1)</sup>

4: es können mehrere Befehle storniert werden

2: es kann lediglich der letzte Befehl storniert werden

0: es können keine Befehle storniert werden

---

1) Vgl. DIN /DIN 66285/ 6; Gierlach, Jankowski /Operationalisierung/ 191; McClure /Automation/ 231



1.5.5 Antwortzeitverhalten<sup>1)</sup>

Informationsaufruf

Wie zeitaufwendig ist der durchschnittliche Informationsaufruf?

4: nicht zeitaufwendig

2: angemessen

0: sehr zeitaufwendig

Diagrammaufruf

Wie zeitaufwendig ist das Laden der Informationen in ein Beschreibungswerkzeug?

4: nicht zeitaufwendig

2: angemessen

0: sehr zeitaufwendig

Speicherung

Wie zeitaufwendig ist das Abspeichern von Informationen aus einem Beschreibungsmittel?

4: nicht zeitaufwendig

2: angemessen

0: sehr zeitaufwendig

---

1) Um eine vergleichbare Basis zur Messung des Antwortzeitverhalten zu bekommen, wird empfohlen ein Fallbeispiel (bspw. LFI-BANK) in die zu untersuchende SEU eingeladen.

Analyse

Wie zeitaufwendig ist die Analyse?

4: nicht zeitaufwendig

2: angemessen

0: sehr zeitaufwendig

Druckausgabe

Wie zeitaufwendig ist die Druckausgabe?

4: nicht zeitaufwendig

2: angemessen

0: sehr zeitaufwendig

## 1.6 Dokumentation und Produktbeschreibung

### Benutzerdokumentation

Reichen die Informationen aus der Benutzerdokumentation für eine sinnvolle Anwendung der SEU aus?<sup>1)</sup>

4: bei der Durchführung der Fallstudie "LFI-Bank" traten keine Probleme auf, die sich nicht mit Hilfe der Benutzerdokumentation lösen ließen

0: bei der Durchführung der Fallstudie "LFI-Bank" traten Probleme auf, die sich nicht mit Hilfe der Benutzerdokumentation lösen ließen

### Systemdokumentation

Reichen die Informationen aus der Systemdokumentation für die Installierung, den Betrieb, die Wartung und Pflege sowie den Test des Produkts aus?<sup>2)3)</sup>

4: die Informationen der Systemdokumentation sind ausreichend

0: die Informationen der Systemdokumentation sind nicht ausreichend

- 
- 1) Vgl. DIN /Beiblatt 1 zu DIN 66230/ 2; Sommerville /Software Engineering/ 230
  - 2) Vgl. DIN /Beiblatt 1 zu DIN 66230/ 5; Sommerville /Software Engineering/ 230
  - 3) Um einen angemessenen Bewertungsumfang zu erreichen, sollte mit Stichproben gearbeitet werden.



### Produktbeschreibung

Entspricht die Produktbeschreibung den Anforderungen des Entwurfs der DIN 66285?<sup>1)</sup>

Im einzelnen werden gefordert:

- 0 die Bezeichnung der Dokumente als Produktbeschreibung
  - 0 eine eindeutige Bezeichnung des Produkts
  - 0 die Bezeichnung der Arbeitsaufgabe
  - 0 die Nennung von Leistungen und Merkmale, hier wird insbesondere getrennt zwischen Option und Variante
  - 0 eine Beschreibung der vorausgesetzten Konfiguration
  - 0 eine Beschreibung der vorausgesetzten Kenntnisse
  - 0 eine Bezeichnung aller Bestandteile der Lieferung
  - 0 eine Angabe, wer die Installierung durchführen soll
  - 0 Angaben über Unterstützung und Wartung
- 4: dem Entwurf zur DIN 66285 wird entsprochen
- 0: dem Entwurf zur DIN 66285 wird nicht entsprochen

---

1) Vgl. DIN /DIN 66285/ 4-5

### Medium

Auf welchem Medium befindet sich die Dokumentation?

4: die Dokumentation ist sowohl als Handbuch als auch online vorhanden

2: die Dokumentation ist nur als Handbuch vorhanden

0: die Dokumentation ist nur online vorhanden

### Richtigkeit

Stimmen die Dokumentationsunterlagen mit dem Softwareprodukt überein? Wurden Widersprüche festgestellt? Ist die Aktualität gegeben?<sup>1)2)</sup>

4: es wurden keine Unstimmigkeiten festgestellt

0: es wurden Unstimmigkeiten festgestellt

### Aufmachung

Wie ist die Übersichtlichkeit und Stukturiert-heit der Dokumentation?<sup>3)</sup>

4: die Dokumentation ist übersichtlich und gut strukturiert

2: die Dokumentation ist nicht in allen Details übersichtlich und gut strukturiert

0: die Dokumentation ist unübersichtlich und nicht strukturiert

---

1) Vgl. Asam, Drenkard, Maier /Qualitätsprüfung/ 113; DIN /DIN 66285/ 5, Ewald /Software/ 96

2) Um einen angemessenen Bewertungsumfang zu erreichen, sollte mit Stichproben gearbeitet werden.

3) Vgl. DIN /DIN 66285/ 5; Ewald /Software/ 96; Sommer-ville /Software Engineering/ 231

1.7 Einführungsunterstützung

Schulungsumfang

Wie umfangreich sind die Einführungsschulungen?

4: es werden Schulungen angeboten, die mehrere Tage dauern

2: es werden Schulungen angeboten, die nur einen Tag dauern

0: es werden keine Schulungen angeboten

Firmeninterne Schulungen

Besteht die Möglichkeit, Schulungen im Unternehmen durchführen zu lassen?<sup>1)</sup>

4: es besteht die Möglichkeit, Schulungen im Unternehmen durchführen zu lassen, es entstehen dabei nur geringfügige Mehrkosten

2: es besteht die Möglichkeit, Schulungen im Unternehmen durchführen zu lassen, es entstehen dabei jedoch erhebliche Mehrkosten

0: es besteht keine Möglichkeit, Schulungen im Unternehmen durchführen zu lassen

---

1) Vgl. Frank /Standardsoftware/ 40

### Unterlagen

Gibt es ausführliche Schulungsunterlagen, die den Entwickler zum Selbststudium befähigen?<sup>1)</sup>

4: es gibt sehr ausführliche Schulungsunterlagen

2: es gibt nur sehr knappe Schulungsunterlagen

0: es gibt keine Schulungsunterlagen

### Hot Line

Gibt es Kontaktpersonen im Unternehmen des CASE-Anbieters, die bei Problemen konsultiert werden können?<sup>2)</sup>

4: es gibt eine Kontaktperson, sie kann meist eine Problemlösung bieten

2: es gibt eine Kontaktperson, sie kann jedoch selten eine Problemlösung bieten

0: es gibt keine Kontaktperson

---

1) Vgl. Frank /Standardsoftware/ 40; Sullivan-Trainor /Buyers' Scorecard/ 69

2) Vgl. Frank /Standardsoftware/ 51

1.8 Konditionengestaltung

Preiskategorie

In welcher Preiskategorie ist das CASE-Produkt einzuordnen?

4: unter TDM 10

2: unter TDM 20

0: TDM 20 oder mehr

Schulungen

Sind im Preis Schulungen für die Mitarbeiter mit eingeschlossen?<sup>1)</sup>

4: es sind umfangreiche Schulungsmaßnahmen mit eingeschlossen

2: es ist nur eine kurze Produkteinführung mit eingeschlossen

0: es sind keine Schulungen mit eingeschlossen

Probeinstallierung

Ist eine Probeinstallierung möglich?<sup>2)</sup>

4: eine Probeinstallierung ist über einen solchen Zeitraum möglich, daß ein Pilotprojekt durchgeführt werden kann

2: eine Probeinstallierung ist für nur für einen kurzen Zeitraum möglich

0: eine Probeinstallierung ist nicht möglich

---

1) Vgl. Frank /Standardsoftware/ 44

2) Vgl. Balzert /Entwicklung/ 8; Frank /Standardsoftware/ 51



### Wartungskosten

Sind in dem Preis Wartungskosten in vollem Umfang eingeschlossen?<sup>1)</sup>

- 4: im Preis sind Wartungskosten einschließlich der Aktualisierung der Dokumentation und Nachschulungen enthalten
- 2: im Preis sind lediglich die Wartung der reinen Software enthalten und eine Dokumentation, die die Neuerungen aufführt
- 0: im Preis sind keine Wartungskosten enthalten

### Wartungsoption

Ist nach Ablauf des evtl. vereinbarten Wartungszeitraum ein Anschlußwartungsvertrag möglich?<sup>2)</sup>

- 4: ja
- 0: nein

---

1) Vgl. Frank /Standardsoftware/ 42

2) Vgl. Frank /Standardsoftware/ 43

### 1.9 Anbieterspezifische Kriterien

Installationen

Anzahl der Installationen (weltweit)<sup>1)</sup>

4: mehr als 10.000 Stck

2: mehr als 1.000 Stck

0: 1.000 Stck oder weniger

Alter

Wie alt ist das Produkt?<sup>2)</sup>

4: älter als 4 Jahre

2: älter als 2 Jahre

0: 2 Jahre oder jünger

Wirtschaftliche Lage

Wie ist die wirtschaftliche Lage des SEU-Anbieters?<sup>3)</sup>

4: die wirtschaftliche Lage ist nicht besorgniserregend

0: die wirtschaftliche Lage ist besorgniserregend

---

1) Vgl. Stahlknecht, Warner /Untersuchung/ 7; Warner /Qual der Wahl/ 59

2) Vgl. Ewald /Software/ 88

3) Vgl. Fisher /CASE/ 258; Frank /Standardsoftware/ 53



Kundenstützpunkt

Wo befindet sich der nächste Kundenstützpunkt  
des Anbieters?<sup>1)</sup>

4: in der gleichen Stadt

2: im gleichen Land

0: im Ausland

---

1) Vgl. Ewald /Software/ 88

2 Spezielle Kriterien für die Auswahl von Software-Entwicklungsumgebungen

2.1 Unterstützung der Phasen des Software-Lebenszyklusses

2.1.1 Entwurfs- und Analysephase

Prinzipien

Werden die wichtigsten Softwareentwicklungs-Prinzipien unterstützt?<sup>1)</sup>

0 Top-Down Prinzip

0 Bottom-Up Prinzip

0 Prinzip der Modularität

4: es werden alle Prinzipien unterstützt

2: es werden nur wenige Prinzipien unterstützt

0: es wird kein Prinzip unterstützt

Textverarbeitung

Ist in der SEU ein Textverarbeitungssystem verfügbar?<sup>2)</sup>

Vgl. 2.2.7 Allgemeine Bürodienste

Matrixbearbeitung

Können Matrizen erstellt und bearbeitet werden?

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Fisher /CASE/ 41; Gewalt, Haake, Pfadler /Software Engineering/ 67

2) Vgl. Hildebrand /Software Tools/ 258; Seibt /RBIS Skriptum/ 55-61

### Checklistenbearbeitung

Können strukturierte Checklisten entworfen und abgearbeitet werden?<sup>1)</sup>

4: besonders komfortabel (bspw. gute Übersichtlichkeit)

2: ja

0: nein

### Präsentationsgrafik

Können Präsentationsgrafiken erstellt werden?<sup>2)</sup>

4: besonders komfortabel (bspw. hohe Auflösung, große Auswahl von Symbolen)

2: ja

0: nein

### Konzeptionelle Datenmodellierung

Können konzeptionelle Datenmodelle erstellt werden?<sup>3)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Scheibl /Kommerzielle Software-Entwicklung/ 248

2) Vgl. Hildbrand /Software Tools/ 259; Seibt /RBIS Skriptum/ 55-61

3) Vgl. Crozier u. a. /Critical analysis/ 487; Vetter /Konzeptionelle Datenmodellierung/ 386-387

logische Datenmodellierung

Können logische Datenmodelle erstellt werden?<sup>1)</sup>

0 relationale Datenmodelle

0 Netzwerke

0 hierarchische Datenmodelle

4: es können relationale und hierarchische Datenmodelle sowie Netzwerke erstellt werden

2: es können entweder relationale Datenmodelle und Netzwerke oder hierarchische Netzwerke erstellt werden

0: nein

physische Datenstrukturierung

Kann die physische Datenstruktur festgelegt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Crozier u. a. /Critical analysis/ 487; Seibt /RBIS Skriptum/ 58; Vetter /konzeptionelle Datenmodellierung/ 386-387

2) Vgl. Seibt /RBIS Skriptum/ 60



### Datenflußdarstellung

Kann der Datenfluß durch das zu erstellende System dargestellt werden?<sup>1)</sup>

0 Datenflußdiagramm

0 HIPO-Diagramm

4: es können sowohl Datenflußdiagramme als auch HIPO-Diagramme dargestellt werden

2: es kann entweder ein Datenflußdiagramm oder ein HIPO-Diagramm dargestellt werden

0: es kann kein Datenfluß dargestellt werden



### Prozeßmodellierung

Können Prozeßmodelle dargestellt werden?<sup>2)</sup>

0 Action Diagram

0 Jackson Diagram

0 Minispezifikationen

0 Struktogramm

0 Structure Chart

0 Warnier-Orr Diagramm

4: es werden eine große Anzahl von Diagrammtypen unterstützt

2: es werden nur wenige Diagrammtypen unterstützt

0: es wird keine Darstellung des Prozeßmodells unterstützt

---

1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 86; Scheibl /Kommerzielle Software-Entwicklung/ 292; Seibt /RBIS Skriptum/ 58, 60

2) Vgl. Seibt /RBIS Skriptum/ 60

Prototyping<sup>1)</sup>

Vgl. 2.2.4 Prototyping

Kapazitätenplanung

Wird die Kapazitätenplanung, insbesondere die voraussichtliche Anzahl der Datensätze, unterstützt?

4: besonders komfortabel

2: ja

0: nein

Informationsverwaltung

Werden die Entwicklungsergebnisse in einer Entwicklungsdatenbank (Repository, Dictionary, Encyclopedia) geführt?<sup>2)</sup>

Vgl. 2.2.1 Informationsverwaltung

Ressourcenplanung

Können Kosten, Zeit, Personal, Rechnerkapazität geplant werden?<sup>3)</sup>

Vgl. 2.2.5 Projektmanagement

- 
- 1) Vgl. Chikofski, Rubenstein /Reliability Engineering/ 13; Hildebrand /Software Tools/ 260; Seibt /RBIS Skriptum/ 58
  - 2) Vgl. Hildebrand /Software-Life-Cycle/ 48
  - 3) Vgl. Scheibl /Kommerzielle Software-Entwicklung/ 291; Seibt /RBIS Skriptum/ 61



### Aufwandsschätzung

Können Aufwandsschätzungen vorgenommen werden?<sup>1)</sup>

Vgl. 2.2.5 Projektmanagement

### Druckausgabe

Besteht die Möglichkeit einer schnellen und bequemen Druckausgabe?<sup>2)</sup>

Vgl. 2.3.2 Outputqualität

### Standardisierung

Können Standards bei der Entwicklung festgesetzt werden?<sup>3)</sup>

4: es können umfangreiche Standards festgesetzt werden, diese werden bei der Entwicklung erzwungen

2: es können Standards festgesetzt werden, diese werden jedoch bei der Entwicklung nicht erzwungen

0: es können keine Standards festgesetzt werden

---

1) Vgl. Boehm /Software-Produktion/ 25; Seibt /RBIS Skriptum/ 55-61

2) Vgl. Fisher /CASE/ 44; Seibt /RBIS Skriptum/ 55-61

3) Vgl. Martin /Information Engineering, Book II/ 276



### Bibliothek

Können Programme, Module, Entwürfe, Bildschirmmasken und Berichte in Bibliotheken zum Zweck der Wiederverwendung abgelegt und aus ihnen entnommen werden?<sup>1)</sup>

- 4: es können Programme, Module, Entwürfe, Bildschirmmasken und Berichtsgestaltungen in Bibliotheken abgelegt und entnommen werden
- 2: es können nur Programme und Module in Bibliotheken abgelegt und entnommen werden
- 0: es existieren keine Bibliotheken in der SEU



### Füllen der Informationsverwaltung

Können aus einem eingelesenen Quellprogramm Informationen in die Informationsverwaltung eingestellt werden?<sup>2)</sup>

- 4: es können umfangreiche Informationen in die Informationsverwaltung eingestellt werden
- 2: es können lediglich Informationen für ein Data Dictionary in die Informationsverwaltung eingestellt werden
- 0: ein Einstellen von Informationen in die Informationsverwaltung ist nicht möglich

---

1) Vgl. McClure /Automation/ 272-273; Seibt /RBIS Skriptum/ 60

2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24



#### Nachdokumentation

Können Programme, die nicht mit Hilfe der SEU entwickelt wurden, nachdokumentiert werden?<sup>1)</sup>

- 4: es kann eine umfangreiche Nachdokumentation erstellt werden
- 2: es kann nur eine begrenzte Nachdokumentation erstellt werden
- 0: es kann keine Nachdokumentation erstellt werden



#### Respezifikation

Können Programme, die nicht mit Hilfe der SEU entwickelt wurden, in grafische Entwurfsspezifikationen übersetzt werden?<sup>2)</sup>

- 4: es können Programme in verschiedene Sichten übersetzt werden
- 2: es können Programme in Prozeßmodelle übersetzt werden
- 0: eine Respezifikation wird nicht unterstützt



#### Restrukturierung

Können Programmstrukturen optimiert werden?<sup>3)</sup>

- 4: Programmstrukturen können im großen Umfang optimiert werden
- 2: Programmstrukturen können nur in geringem Umfang optimiert werden
- 0: eine Restrukturierung wird nicht unterstützt

---

1) Vgl. Lempp /vorhandene Software nutzen/ 43

2) Vgl. Lempp /vorhandene Software nutzen/ 43

3) Vgl. Lempp /vorhandene Software nutzen/ 43

## 2.1.2 Realisierungsphase

### Textverarbeitung

Ist in der SEU ein Textverarbeitungssystem verfügbar?<sup>1)</sup>

Vgl. 2.2.7 Allgemeine Bürodienste

### Ressourcenplanung

Können Kosten, Zeit, Personal, Rechnerkapazität geplant werden?<sup>2)</sup>

Vgl. 2.2.5 Projektmanagement

### Aufwandsschätzung

Können Aufwandsschätzungen vorgenommen werden?<sup>3)</sup>

Vgl. 2.2.5 Projektmanagement

### Verfügbarkeit eines Code-Generators

Ist ein Code-Generator in die SEU integriert oder besteht eine Schnittstelle zu einem marktgängigen Code-Generator?<sup>4)</sup>

4: in die SEU ist ein Code-Generator integriert

2: in der SEU besteht eine Schnittstelle zu einem marktgängigen Code-Generator

0: ein Code-Generator ist nicht verfügbar

---

1) Vgl. Seibt /RBIS Skriptum/ 62-64

2) Vgl. Seibt /RBIS Skriptum/ 62

3) Vgl. Boehm /Software-Produktion/ 25; Seibt /RBIS Skriptum/ 62

4) Vgl. Chikofsky, Rubenstein /Reliability Engineering/ 13

#### Codierobjekt

Welche Teile des Entwurfs können über einen Code-Generator übersetzt werden?

4: der gesamte Entwurf

2: nur Masken

0: kein Teil

#### Code-Qualität

Ist der übersetzte Quellcode lauffähig oder müssen manuell Änderungen vorgenommen werden?

4: es werden korrekte Programme geliefert

2: es müssen manuell geringfügige Änderungen vorgenommen werden

0: es müssen viele Änderungen vorgenommen werden

#### Code-Kennzeichnung

Ist der übersetzte Quellcode im Listing besonders gekennzeichnet?<sup>1)</sup>

4: deutliche Kennzeichnung

2: Kennzeichnung

0: keine besondere Kennzeichnung

---

1) Vgl. Sommerville /Software Engineering/ 138

#### Up- und Download

Ist eine Portierung des erzeugten Quellcode auf eine Großrechneranlage möglich?

4: eine Portierung ist ohne manuelle Änderungen möglich

2: für eine Portierung sind geringfügige Änderungen notwendig

0: eine Portierung ist nicht möglich

#### Dokumentation

Stehen für die Erstellung von Benutzer- und Operatorhandbüchern Berichte (Reports) zur Verfügung?<sup>1)</sup>

Vgl. 2.2.3 Dokumentation

#### Testwerkzeuge

Stehen für den Modul-, Verfahrens- und Systemtest<sup>2)</sup> Werkzeuge zur Verfügung?<sup>3)</sup>

Vgl. 2.2.4 Qualitätssicherung

- 
- 1) Vgl. McClure /CASE Experience/ 77, 238; Seibt /RBIS Skriptum/ 62-63
  - 2) Vgl. Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 75; Seibt /RBIS Skriptum/ 63
  - 3) Vgl. Hildebrand /Software Tools/ 261; McClure /Automation/ 142

### 2.1.3 Erprobungs- und Konsolidierungsphase

#### Textverarbeitung

Ist in der SEU ein Textverarbeitungssystem verfügbar?<sup>1)</sup>

Vgl. 2.2.7 Allgemeine Bürodienste

#### Ressourcenplanung

Können Kosten, Zeit, Personal, Rechnerkapazität geplant werden?<sup>2)</sup>

Vgl. 2.2.5 Projektmanagement

#### Aufwandsschätzung

Können Aufwandsschätzungen vorgenommen werden?<sup>3)</sup>

Vgl. 2.2.5 Projektmanagement

#### Testwerkzeuge

Stehen für den Modul-, Verfahrens- und Systemtest<sup>4)</sup> Werkzeuge zur Verfügung?<sup>5)</sup>

Vgl. 2.2.4 Qualitätssicherung

- 
- 1) Vgl. Hildebrand /Software Tools/ 261; Seibt /RBIS Skriptum/ 65-70
  - 2) Vgl. Seibt /RBIS Skriptum/ 65-66
  - 3) Vgl. Seibt /RBIS Skriptum/ 65-66
  - 4) Vgl. Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 75; Seibt /RBIS Skriptum/ 66, 69
  - 5) Vgl. Hildebrand /Software Tools/ 261; McClure /Automation/ 142

#### Modifikationsmöglichkeiten

Können Änderungen am Quellcode vorgenommen werden, ohne einen kleinen kompletten Softwareentwicklungszyklus durchlaufen zu müssen?

4: besonders komfortabel

2: ja

0: nein

#### Konvertierungsprogramme

Stehen für die Datenübernahme in das neue System Konvertierungsprogramme zur Verfügung?<sup>1)</sup>

4: es stehen eine große Anzahl oder flexible Konvertierungsprogramme zur Verfügung

2: es stehen einige Konvertierungsprogramme zur Verfügung, so daß ein Großteil der Konvertierungsaufgaben gelöst werden kann

0: Konvertierungsprogramme stehen nicht zur Verfügung

---

1) Vgl. Hildebrand /Software Tools/ 261



#### 2.1.4 Pflege- und Wartungsphase

##### Prinzipien

Werden die wichtigsten Softwareentwicklungs-Prinzipien unterstützt?<sup>1)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

##### Textverarbeitung

Ist in der SEU ein Textverarbeitungssystem verfügbar?<sup>2)</sup>

Vgl. 2.2.7 Allgemeine Bürodienste

##### Matrixbearbeitung

Können Matrizen erstellt und bearbeitet werden?<sup>3)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

##### Checklistenbearbeitung

Können strukturierte Checklisten entworfen und abgearbeitet werden?<sup>4)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

- 
- 1) Vgl. Fisher /CASE/ 41; Gewalt, Haake, Pfadler /Software Engineering/ 24, 67
  - 2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Hildebrand /Software Tools/ 258; Seibt /RBIS Skriptum/ 55-61
  - 3) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Scheibl /Kommerzielle Software-Entwicklung/ 248

### Präsentationsgrafik

Können Präsentationsgrafiken erstellt werden?<sup>1)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### Konzeptionelle Datenmodellierung

Können konzeptionelle Datenmodelle erstellt werden?<sup>2)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### logische Datenmodellierung

Können logische Datenmodelle erstellt werden?<sup>3)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### physische Datenstrukturierung

Kann die physische Datenstruktur festgelegt werden?<sup>4)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

- 
- 1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Hildbrand /Software Tools/ 259; Seibt /RBIS Skriptum/ 55-61
  - 2) Vgl. Crozier u. a. /Critical analysis/ 487; Gewalt, Haake, Pfadler /Software Engineering/ 24; Vetter /Konzeptionelle Datenmodellierung/ 386-387
  - 3) Vgl. Seibt /RBIS Skriptum/ 58; Vetter /konzeptionelle Datenmodellierung/ 386-387
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Seibt /RBIS Skriptum/ 60

### Datenflußdarstellung

Kann der Datenfluß durch das zu erstellende System dargestellt werden?<sup>1)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### Prozeßmodellierung

Können Prozeßmodelle dargestellt werden?<sup>2)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### Prototyping

Können Prototypen erstellt werden?

Vgl. 2.2.4 Prototyping

### Kapazitätenplanung

Wird die Kapazitätenplanung, insbesondere die voraussichtliche Anzahl der Datensätze unterstützt?<sup>3)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

- 
- 1) Vgl. Scheibl /Kommerzielle Software-Entwicklung/ 292; Seibt /RBIS Skriptum/ 58, 60
  - 2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Seibt /RBIS Skriptum/ 60
  - 3) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24

### Informationsverwaltung

Werden die Entwicklungsergebnisse in einer Entwicklungsdatenbank (Repository, Dictionary, Encyclopedia) geführt?<sup>1)</sup>

Vgl. 2.2.1 Informationsverwaltung

### Ressourcenplanung

Können Kosten, Zeit, Personal, Rechnerkapazität geplant werden?<sup>2)</sup>

Vgl. 2.2.5 Projektmanagement

### Aufwandsschätzung

Können Aufwandsschätzungen vorgenommen werden?<sup>3)</sup>

Vgl. 2.2.5 Projektmanagement

### Druckausgabe

Besteht die Möglichkeit einer schnellen und bequemen Druckausgabe?<sup>4)</sup>

Vgl. 2.3.2 Outputqualität

- 
- 1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Hildebrand /Software-Life-Cycle/ 48
  - 2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Scheibl /Kommerzielle Software-Entwicklung/ 291; Seibt /RBIS Skriptum/ 61
  - 3) Vgl. Boehm /Software-Produktion/ 25; McClure /Automation/ 145; Seibt /RBIS Skriptum/ 55-61
  - 4) Vgl. Fisher /CASE/ 44; Gewalt, Haake, Pfadler /Software Engineering/ 24; Seibt /RBIS Skriptum/ 55-61



### Standardisierung

Können Standards bei der Entwicklung festgesetzt werden?<sup>1)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase



### Bibliothek

Können Programme, Module, Entwürfe, Bildschirmmasken und Berichte in Bibliotheken zum Zweck der Wiederverwendung abgelegt und aus ihnen entnommen werden?<sup>2)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase



### Füllen der Informationsverwaltung

Können aus einem eingelesenen Quellprogramm Informationen in die Informationsverwaltung eingestellt werden?<sup>3)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase



### Nachdokumentation

Können Programme, die nicht mit Hilfe der SEU entwickelt wurden, nachdokumentiert werden?<sup>4)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

- 
- 1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Martin /Information Engineering, Book II/ 276
  - 2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; McClure /Automation/ 272-273; Seibt /RBIS Skriptum/ 60
  - 3) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Lempp /vorhandene Software nutzen/ 43

### Respezifikation

Können Programme, die nicht mit Hilfe der SEU entwickelt wurden, in grafische Entwurfsspezifikationen übersetzt werden<sup>1)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### Restrukturierung

Können Programmstrukturen optimiert werden?<sup>2)</sup>

Vgl. 2.1.1 Entwurfs- und Analysephase

### Verfügbarkeit eines Code-Generators

Ist ein Code-Generator in die SEU integriert oder besteht eine Schnittstelle zu einem marktgängigen Code-Generator?<sup>3)</sup>

Vgl. 2.1.2 Realisierungsphase

### Codierobjekt

Welche Teile des Entwurfs können über einen Code-Generator übersetzt werden?<sup>4)</sup>

Vgl. 2.1.2 Realisierungsphase

- 
- 1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Lempp /vorhandene Software nutzen/ 43
  - 2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Lempp /vorhandene Software nutzen/ 43
  - 3) Vgl. Chikofsky, Rubenstein /Reliability Engineering/ 13; Gewalt, Haake, Pfadler /Software Engineering/ 24
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24

#### Code-Qualität

Ist der übersetzte Quellcode lauffähig oder müssen manuell Änderungen vorgenommen werden?<sup>1)</sup>

Vgl. 2.1.2 Realisierungsphase

#### Code-Kennzeichnung

Ist der übersetzte Quellcode im Listing besonders gekennzeichnet?<sup>2)</sup>

Vgl. 2.1.2 Realisierungsphase

#### Up- und Download

Ist eine Portierung des erzeugten Quellcode auf eine Großrechneranlage möglich?<sup>3)</sup>

Vgl. 2.1.2 Realisierungsphase

#### Dokumentation

Stehen für die Erstellung von Benutzer- und Operatorhandbüchern Berichte (Reports) zur Verfügung?<sup>4)</sup>

Vgl. 2.2.3 Dokumentation

- 
- 1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24
  - 2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Sommerville /Software Engineering/ 138
  - 3) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; McClure /CASE Experience/ 77, 238; Seibt /RBIS Skriptum/ 62-63



### Testwerkzeuge

Stehen für den Modul-, Verfahrens- und Systemtest<sup>1)</sup> Werkzeuge zur Verfügung?<sup>2)</sup>

Vgl. 2.2.4 Qualitätssicherung



### Modifikationsmöglichkeiten

Können Änderungen am Quellcode vorgenommen werden, ohne einen kleinen kompletten Softwareentwicklungszyklus durchlaufen zu müssen?<sup>3)</sup>

Vgl. 2.1.3 Erprobungs- und Konsolidierungsphase



### Konvertierungsprogramme

Stehen für die Datenübernahme in das neue System Konvertierungsprogramme zur Verfügung?<sup>4)</sup>

Vgl. 2.1.3 Erprobungs- und Konsolidierungsphase

- 
- 1) Vgl. Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 75; Seibt /RBIS Skriptum/ 66, 69
  - 2) Vgl. Hildebrand /Software Tools/ 261; McClure /Automation/ 142
  - 3) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24
  - 4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 24; Hildebrand /Software Tools/ 261



## 2.2 Unterstützung phasenübergreifender Aufgaben

### 2.2.1 Informationsverwaltung



Speicherbare Objekte

Welche Informationen können in der Informationsverwaltung gespeichert werden?<sup>1)</sup>

- 0 Datendefinitionen
- 0 Beschreibungen
- 0 Beziehungen zwischen Daten
- 0 Regeln
- 0 physische Informationen
- 0 Domainen
- 0 Alias
- 0 Frequenzen
- 0 Volumen
- 0 Benutzer

4: es können zahlreiche Informationen gespeichert werden

2: es können einige Informationen gespeichert werden

0: es können lediglich die Datendefinitionen gespeichert werden

---

1) Vgl. Crozier u. a. /Critical analysis/ 492; DeMarco /Structured Analysis/ 126, 143; Gane /Computer-Aided Software Engineering/ 46-48; Martin, McClure /Structured Techniques/ 155

### Berichtswesen

Welche Berichtsmöglichkeiten stehen zur Verfügung?<sup>1)</sup>

- 0 vorgefertigte Berichte
- 0 selbst zu erstellende Berichte
- 0 Direktabfragen ("ad hoc"-Abfragen)
- 4: es steht eine große Auswahl von vorgefertigten Berichten zur Verfügung, zusätzlich werden selbst zu erstellende Berichte und Direktabfragen unterstützt
- 2: es steht nur eine große Auswahl von vorgefertigten Berichten zur Verfügung
- 0: es steht nur eine kleine Auswahl von vorgefertigten Berichten zur Verfügung

### Teilmengenbildung

Ist für die Analyse eine Bildung von Teilmengen möglich?<sup>2)</sup>

- 4: besonders komfortabel
- 2: ja
- 0: nein

---

1) Vgl. McClure /Automation/ 60; Ruf /Basis des Schnittstellen-Management-Ansatzes/ 234

2) Vgl. Österle /CASE/ 358-359

### Suchfunktion

Können Objekte der Informationsverwaltung über eine Suchfunktion aufgelistet (browse) und aus ihr wiederverwendet werden?

- 4: eine große Anzahl von Objekttypen kann über die Suchfunktionen aus verschiedenen Funktionen der SEU abgerufen werden
- 2: es kann entweder nur eine begrenzte Anzahl von Objekttypen abgerufen werden oder die Suchfunktion ist nicht in allen Funktionen verfügbar
- 0: eine Suchfunktion ist nicht verfügbar

### Standardisierungsunterstützung

Können für die Namensvergabe von Objekten Konventionen festgelegt werden, die bei Eingabe von Namen erzwungen werden?<sup>1)</sup>

- 4: Namenskonventionen können festgelegt werden, sie werden bei der Namensvergabe erzwungen
- 2: Namenskonventionen der SEU werden bei der Namensvergabe erzwungen
- 0: die Standardisierung der Namensvergabe wird nicht unterstützt

---

1) Vgl. DeMarco /Structured Analysis/ 158; Gane /Computer-Aided Software Engineering/ 43; Heuer /Rolle von Dictionary-Systemen/ 121



#### Vorschläge für die Namensvergabe

Werden von der SEU Vorschläge für die Namensvergabe angeboten?<sup>1)</sup>

- 4: es werden online eine Reihe von standardisierten Abkürzungen für die nationalsprachliche Namensvergabe zur Verfügung gestellt
- 2: es werden online nur eine begrenzte Anzahl von standardisierten Abkürzungen oder nur in englischer Sprache zur Verfügung gestellt
- 0: es werden keine Namensvergabevorschläge gemacht



#### Verbindung zum Grafikeditor

Kann direkt aus dem Diagramm auf die Informationsverwaltung zugegriffen werden?<sup>2)</sup>

- 4: aus einem Diagramm kann über eine Funktionstaste oder einen Menüpunkt in die Informationsverwaltung eines bestimmten Objekts gewechselt werden, anschließend kann zu anderen Objekten der Informationsverwaltung gewechselt werden
- 2: aus einem Diagramm kann über eine Funktionstaste oder einen Menüpunkt in die Informationsverwaltung eines bestimmten Objekts gewechselt werden, anschließend kann nur in das Diagramm zurückgekehrt werden
- 0: ein Wechsel vom Diagramm ist nur zum Menü der Informationsverwaltung möglich

---

1) Vgl. Gane /Computer-Aided Software Engineering/ 44

2) Vgl. Crozier u. a. /Critical analysis/ 493



#### Abschaltung der Protokollierung

Können Diagramme erstellt werden, ohne daß sofort alle Details in der Informationsdatenbank protokolliert werden?<sup>1)</sup>

- 4: es können Symbole in die Grafik einschließlich Beschriftungen eingefügt werden, ohne daß die Informationsdatenbank gefüllt wird
- 2: es können Symbole in die Grafik jedoch ohne Beschriftungen eingefügt werden, ohne daß die Informationsdatenbank gefüllt wird
- 0: die Protokollfunktion der Informationsverwaltung kann nicht abgestellt werden



#### Protokollierung grafischer Details

Werden grafische Details, bspw. Quelle und Ziel eines Datenflusses, in der Informationsdatenbasis gespeichert?

- 4: die Speicherung von grafischen Details erfolgt automatisiert beim Erstellen der Objekte für jedes Objekt
- 2: die Speicherung von grafischen Details muß manuell erfolgen
- 0: eine Speicherung grafischer Details kann nicht erfolgen

---

1) Vgl. Crozier u. a. /Critical analysis/ 493



#### Führen von Prüfungsfeldern

Werden Informationen über die letzte Änderung von Objekten in der Informationsverwaltung gespeichert?<sup>1)</sup>

- 4: die Speicherung von Änderungsinformationen erfolgt automatisiert
- 2: die Speicherung von Änderungsinformationen muß manuell erfolgen
- 0: eine Speicherung von Änderungsinformationen kann nicht erfolgen



#### Auswirkungen von Diagrammänderungen

Werden Änderungen oder Löschungen in Diagrammen auf die Informationsverwaltung und auf andere Diagramme übertragen?<sup>2)</sup>

- 4: Diagrammänderungen wirken sich auf die Inf.-Verw. und andere Diagramme aus
- 2: Diagrammänderungen wirken sich entweder auf die Inf.-Verw. oder andere Diagramme aus
- 0: Diagrammänderungen wirken sich weder auf die Inf.-Verw. noch auf andere Diagramme aus

---

1) Vgl. Crozier u. a. /Critical analysis/ 494

2) Vgl. Crozier u. a. /Critical analysis/ 493-494



#### Direkter Inf.-Verw.-Zugriff

Können Objekte direkt in der Informationsverwaltung erstellt werden?<sup>1)</sup>

- 4: Objekte können direkt in der Informationsverwaltung erstellt werden, ein entsprechendes Symbol erscheint im Diagramm
- 2: Objekte können direkt in der Informationsverwaltung erstellt werden, ein Symbol erscheint nicht in der Grafik
- 0: Objekte können nicht direkt in der Informationsverwaltung erstellt werden



#### Löschen in der Inf.-Verw.

Wie wirken sich Löschungen auf die Diagramme aus?<sup>2)</sup>

- 4: die entsprechenden Symbole einschließlich ihrer Verbindungen werden in den Diagrammen gelöscht
- 2: es werden nur die Verbindungen zur Informationsverwaltung gelöscht
- 0: Diagramme werden durch die Löschung in der Informationsverwaltung nicht verändert

---

1) Vgl. Crozier u. a. /Critical analysis/ 493

2) Vgl. Crozier u. a. /Critical analysis/ 494

## 2.2.2 Dokumentation

### Textverarbeitung

Ist ein Textverarbeitungssystem verfügbar?<sup>1)</sup>

Vgl. 2.2.7 Allgemeine Bürodienste

### Vorgefertigte Berichte

Stehen für Dokumentationszwecke vorgefertigte Berichte zur Verfügung?

4: es stehen eine große Anzahl von Berichten zur Verfügung, die die Dokumentation unterstützen

2: es stehen einige Berichte zur Verfügung, die die Dokumentation unterstützen

0: es stehen keine vorgefertigten Berichte zur Verfügung, die die Dokumentation unterstützen

### Gestaltbare Berichte

Können für Dokumentationszwecke Berichte selbst erstellt werden?

4: menügesteuert können Berichte gestaltet werden, hierbei können die Objektart und die Untersuchungsmenge eingegrenzt werden

2: menügesteuert können Berichte gestaltet werden, eine mehrdimensionale Eingrenzung ist nicht möglich

0: Berichte können nicht selbst erstellt werden

---

1) Vgl. Balzert /Entwicklung/ 492; Gane /Computer-Aided Software Engineering/ 89; Gewalt, Haake, Pfadler /Software Engineering/ 184-185



### Layoutgestaltung

Wie hoch ist die Flexibilität bei der Gestaltung der Berichte?

4: das Layout ist wie in einer Textverarbeitung frei gestaltbar

2: das Layout kann aus vorgefertigten Modulen zusammengestellt werden

0: das Layout wird vorgegeben

### Standardisierung

Können Unternehmensstandards für die Berichts- und Diagrammgestaltung in der Dokumentation berücksichtigt werden?<sup>1)</sup>

4: Standards können frei gestaltet werden, sie werden auf alle Dokumente übertragen

2: Standards sind begrenzt gestaltbar oder sie können nur auf einige Dokumente übertragen werden

0: die Standardisierung wird nicht unterstützt

### Kopfzeile

Werden in der Kopfzeile jedes Berichts Informationen über Projekt, Bericht und Zeitpunkt der Erstellung ohne besondere Eingabe bei der Berichtserstellung geliefert?

4: es werden umfangreiche Information geliefert

2: es werden einige Informationen geliefert

0: es werden keine Informationen geliefert

---

1) Vgl. Boehm u. a. /Improving Productivity/ 37; Leinweber /Planung und Dokumentation/ VI-11-08

### Strukturierung

Wird die Strukturierung der Dokumentation unterstützt?<sup>1)</sup>

- 4: die Struktur der Dokumentation kann festgelegt werden, zur Übersicht dienen Inhaltsverzeichnis, Glossar oder ein Diagramm
- 2: die Struktur der Dokumentation kann festgelegt werden
- 0: die Strukturierung wird nicht unterstützt

### Integration

Können in die Dokumentation unterschiedliche Dokumente aufgenommen werden?

- 4: es können unterschiedliche Dokumente, auch solche, die nicht mit der SEU erstellt wurden, in die Dokumentation integriert werden
- 2: es können unterschiedliche Dokumente in die Dokumentation integriert werden
- 0: nein

---

1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 180;  
Sommerville /Software Engineering/ 231

#### Fehlerkorrektur

Wird vor Ausdruck der Dokumentation eine Überprüfung des korrekten Dokumentationsaufbaus und die Verfügbarkeit der einzelnen Dokumente unterstützt?

- 4: vor dem Ausdruck der Dokumentation wird eine Überprüfung des korrekten Aufbaus und der Verfügbarkeit der einzelnen Dokumente unterstützt
- 2: vor dem Ausdruck der Dokumentation wird eine Überprüfung der Verfügbarkeit der einzelnen Dokumente unterstützt
- 0: eine Überprüfung der Dokumentation wird nicht unterstützt

#### Teilausdruck

Können Teile der Dokumentation ausgedruckt werden?

- 4: Teile der Dokumentation können bequem über ein Menü zum Ausdruck ausgewählt werden
- 2: die Auswahl von Teilen der Dokumentation ist möglich
- 0: es kann nur die gesamte Dokumentation ausgedruckt werden



### Grafik und Text

Können Grafik und Text gemeinsam in die Dokumentation eingebracht werden?<sup>1)</sup>

- 4: Grafiken können direkt in den Text eingebunden werden
- 2: Grafiken werden auf eine separate Seite gedruckt
- 0: Grafiken und Text müssen separat ausgedruckt werden

---

1) Vgl. Fisher /CASE/ 147

### 2.2.3 Qualitätssicherung



Syntaxcheck bei der Eingabe

Kann der Zeitpunkt des Syntaxchecks der Eingabe gewählt werden?<sup>1)</sup>

4: ein Syntaxcheck bei der Eingabe kann aktiviert werden

2: die Syntaxcheck findet bei der Eingabe statt

0: die Syntax wird bei der Eingabe nicht geprüft



Vollständigkeitscheck

Können Vollständigkeitschecks für ein Diagramm durchgeführt werden?<sup>2)</sup>

(Wird bspw. im DFD überprüft, ob jeder Prozeß einen In- und Output hat?)

4: mehrere Berichte stehen zur Verfügung

2: einige Berichte stehen zur Verfügung

0: die Vollständigkeit kann nicht überprüft werden

---

1) Vgl. DeMarco /Structured Analysis/ 158; McClure /Automation/ 43

2) Vgl. Österle /CASE/ 358-359; Schmitz /Methoden, Verfahren und Werkzeuge/ 76; Schulz /Klassifizierungs- und Bewertungsschema/ 196



#### Konsistenzcheck im Diagramm

Können Konsistenzchecks für einzelne Diagramme durchgeführt werden?<sup>1)</sup>

(Wird bspw. im DFD die doppelte Vergabe von Namen erkannt?)

4: mehrere Berichte stehen zur Verfügung  
oder: eine doppelte Vergabe wird durch die Eingabekontrolle abgefangen

2: einige Berichte stehen zur Verfügung

0: die Konsistenz kann nicht überprüft werden



#### Konsistenzcheck zwischen Diagrammen

Kann die Konsistenz zwischen mehreren Diagrammen getestet werden?<sup>2)</sup>

(Bspw. durch Funktions- und Entity-Matrizen oder durch Abstimmung verschiedener Diagrammebenen)

4: mehrere Berichte stehen zur Verfügung

2: einige Berichte stehen zur Verfügung

0: die Konsistenz kann nicht überprüft werden

---

1) Vgl. DeMarco /Structured Analysis/ 158; Martin, McClure /Structured Techniques/ 177; Schulz /Klassifizierungs- und Bewertungsschema/ 196

2) Vgl. DeMarco /Structured Analysis/ 158; Fisher /CASE/ 149



#### Konsistenzcheck: Diagramm - Data Dictionary

Kann überprüft werden, ob alle Symbole auch als Objekte im Data Dictionary vorhanden sind und ob Objekte nicht als Symbole in Diagrammen auftauchen?<sup>1)</sup>

- 4: alle Diagramme können mit dem Data Dictionary abgestimmt werden
- 2: einige Diagramme können mit dem Data Dictionary abgestimmt werden
- 0: keine Diagramme können mit dem Data Dictionary abgestimmt werden



#### Datenanalyse

Können die Daten in der Datenmodellierung auf strukturelle Probleme untersucht werden?

- 4: mehrere Berichte stehen zur Verfügung
- 2: einige Berichte stehen zur Verfügung
- 0: eine Untersuchung auf strukturelle Probleme kann nicht durchgeführt werden



#### Zusammenhänge

Können die Zusammenhänge zwischen Objekten aus verschiedenen Diagrammtypen durch Zuordnungen verdeutlicht werden?

(Bsp.: Funktion-Entity-Matrix)

- 4: es können verschiedene Zusammenhänge durch Matrizen dargestellt werden
- 2: es kann nur ein Zusammenhang zwischen Matrizen dargestellt werden
- 0: es stehen keine Matrizen zur Verfügung

---

1) Vgl. Fisher /CASE/ 144

### Testdatengeneratoren

Stehen Testdatengeneratoren zur Verfügung?<sup>1)</sup>

4: es stehen Testdatengeneratoren zur Verfügung, die gewährleisten, daß alle Programmschritte mindestens einmal durchlaufen werden

2: es stehen Testdatengeneratoren zur Verfügung, die die Daten nach einem Zufallsalgorithmus ermitteln

0: es stehen keine Testdatengeneratoren zur Verfügung

### Simulation der Testumgebung

Können vorgelagerte Schnittstellen durch Treiber, nachgelagerte Schnittstellen durch Platzhalter simuliert werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Balzert /Entwicklung/ 436; DeMarco /Structured Analysis/ 331

2) Vgl. Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 122





### Testauswertung

Stehen Werkzeuge für die Testauswertung zur Verfügung?<sup>1)</sup>

- 4: es können mehrere Eigenschaften des Programms ausgewertet werden (Soll- Istdatenvergleich, Performancemessung etc.)
- 2: es können nur Soll- Istdatenvergleiche durchgeführt werden
- 0: es kann keine Testauswertung vorgenommen werden



### Testdokumentation

Können die Testergebnisse dokumentiert werden?<sup>2)</sup>

- 4: die Testergebniss können besonders komfortabel, ausführlich und übersichtlich dokumentiert werden
- 2: die Testergebnisse können dokumentiert werden
- 0: die Testergebnisse können nicht dokumentiert werden

---

1) Vgl. Balzert /Entwicklung/ 435-436; DeMarco /Structured Analysis/ 327; Schmitz, Bons, van Megen /Software-Qualitätssicherung/ 160

2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 171



## Debugger

Stehen Debugger zur Verfügung?<sup>1)</sup>

4: es stehen dynamische Debugger zur Verfügung, die während der Programmausführung eine Variablenmanipulation zulassen und die die Position im Quellcode anzeigen

2: es stehen Debugger zur Verfügung

0: es stehen keine Debugger zur Verfügung

---

1) Vgl. Balzert /Entwicklung/ 436; Hildebrand /Software Tools/ 261

#### 2.2.4 Prototyping

##### Abdeckungsbreite

Welche Programmteile können mit dem Prototyper dargestellt werden?<sup>1)</sup>

- 4: es kann ein komplettes Softwaresystem einschließlich Datenbankzugriffen und Berechnungen unter Berücksichtigung von Performancegesichtspunkten dargestellt werden
- 2: es kann lediglich Programmteile dargestellt werden
- 0: es können keine Programmteile mit dem Prototyper dargestellt werden

##### Darstellungsbreite

Für welche Programmteile können Prototypen erstellt werden?

- 0 Benutzerschnittstelle
- 0 Berechnungen
- 0 Sonstige
- 4: es können mehrere Programmteile dargestellt werden
- 2: es kann nur ein Programmteil dargestellt werden
- 0: es kann kein Programmteil dargestellt werden

---

1) Vgl. Martin /Information Engineering, Book III/ 110

### Schnittstellen

Welche Schnittstellen der Dialogführung können dargestellt werden?<sup>1)</sup>

- 4: es können Bildschirmmasken und Berichte dargestellt werden
- 2: es können nur Bildschirmmasken dargestellt werden
- 0: es können keine Schnittstellen der Dialogführung dargestellt werden

### Details

Welche Details in den Bildschirmmasken können dargestellt werden?<sup>2)</sup>

- 0 Fenster
- 0 Zeigeoperationen
- 0 Pull-Down Menüs
- 0 Menüs
- 0 Funktionstasten
- 4: es sind "oberflächen- und dialogorientierte"<sup>3)</sup> Oberflächen darstellbar
- 2: es sind nur dialogorientierte Oberflächen darstellbar
- 0: es sind keine Oberflächen darstellbar

---

1) Vgl. Gibson /CASE Philosophy/ 216

2) Vgl. Gane /Computer-Aided Software Engineering/ 62; Hoppe /Prototypenentwicklung/ 278

3) Hoppe /Prototypenentwicklung/ 280

### Simulation der Eingabe

Kann die Systemreaktion auf die Eingabe simuliert werden?<sup>1)</sup>

- 0 Eingabechecks
- 0 Defaultwerte
- 0 Soll- und Mußfelder
- 0 Hilfetexte

- 4: die Eingabe kann detailliert simuliert werden
- 2: die Eingabe kann simuliert werden
- 0: die Eingabe kann nicht simuliert werden

### Simulation des Dialogs

Kann der Dialog simuliert werden?

- 4: je nach Menüauswahl werden verschiedene Masken aufgerufen
- 2: die Simulation beschränkt sich auf eine Kette von Bildschirmmasken
- 0: der Dialog kann nicht simuliert werden

### Simulation der Zielumgebung

Kann das Zielsystem simuliert werden?

- 4: der Bildschirm und die Tastenbelegung des Zielsystems können simuliert werden
- 2: der Bildschirm oder die Tastenbelegung des Zielsystems können simuliert werden
- 0: eine Simulation der Zielumgebung wird nicht unterstützt

---

1) Vgl. Österle /CASE/ 358-359

### Prototyperstellung

Wie bequem und schnell kann ein Prototyp erstellt werden?

- 4: die Werkzeugoberfläche ist benutzerfreundlich, der Entwickler wird bei der Erstellung durch eine Schnittstelle zur Informationsdatenbasis unterstützt
- 2: die Werkzeugoberfläche ist benutzerfreundlich oder der Entwickler wird bei der Erstellung durch eine Schnittstelle zur Informationsverwaltung unterstützt
- 0: eine einfache und schnelle Erstellung eines Prototypen ist nicht möglich

### Inkrementelle Übersetzer

Stehen für die Erstellung der Prototypen inkrementelle Übersetzer zur Verfügung?<sup>1)</sup>

- 4: ja
- 0: nein

### Navigation Route

Kann der Dialog in einem Diagramm über mehrere Bildschirmmasken verfolgt werden?<sup>2)</sup>

- 4: besonders komfortabel
- 2: ja
- 0: nein

---

1) Vgl. Hensel /Perspektiven/ o. S.

2) Vgl. Crozier u. a. /Critical analysis/ 494; Gane /Computer-Aided Software Engineering/ 62; Hoppe /Prototypenentwicklung/ 282



#### Code-Generierung

Kann der Schnittstellenentwurf in Programmcode übersetzt werden?

- 4: eine Übersetzung erfolgt, so daß die Eingabemaske im Programm erscheint
- 2: es werden lediglich die Eingabefelddefinitionen im Programm abgebildet
- 0: eine Übersetzung erfolgt nicht

### 2.2.5 Projektmanagement

#### Strukturierung

Kann das Projekt nach Phasen und Aktivitäten strukturiert werden?<sup>1)</sup>

4: zur Strukturierung stehen grafische Darstellungsmittel zur Verfügung

2: zur Strukturierung stehen nur sprachliche Darstellungsmittel zur Verfügung

0: eine Strukturierung ist nicht möglich

#### Ressourcenplanung

Können Zeit, Ressourcen und Kosten des Projekts geplant werden?<sup>2)</sup>

4: es können Zeit, Ressourcen und Kosten mit grafischen Darstellungsmitteln geplant werden

2: es können Zeit, Ressourcen und Kosten nur mit sprachlichen Darstellungsmitteln geplant werden

0: es können keine Ressourcen geplant werden

---

1) Vgl. Curth, Wyss /Information engineering/ 221

2) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 234; Koslowski /Unterstützung partizipativer Systementwicklung/ 127



### Aufwandsschätzung

Wird die Aufwandsschätzung des Projekts unterstützt?<sup>1)</sup>

0 Function-Point-Verfahren<sup>2)</sup>

0 COCOMO-Schätzverfahren<sup>3)</sup>

0 andere

4: es stehen mehrere Verfahren zur Aufwandsschätzung zur Verfügung

2: es steht ein Verfahren zur Aufwandsschätzung zur Verfügung

0: es steht kein Verfahren zur Aufwandsschätzung zur Verfügung

### Überwachung

Können Projektfortschritt und Aufwendungen kontrolliert werden?<sup>4)</sup>

4: es steht ein ausführliches Berichtssystem zur Projektüberwachung zur Verfügung

2: es steht ein eingeschränktes Berichtssystem zur Projektüberwachung zur Verfügung

0: es steht kein Berichtssystem zur Projektüberwachung zur Verfügung

### Textverarbeitung

Vgl. 2.2.7 Allgemeine Bürodienste

---

1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 262

2) Vgl. Seibt /Function-Point-Methode/ 109

3) Vgl. Boehm /Software-Produktion/ 85; Seibt /COCOMO/ 56

4) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 272; Hering /Software-Engineering/ 98

Tabellenkalkulation

Vgl. 2.2.7 Allgemeine Bürodienste

Zugriffsrechte

Können Zugriffsrechte vergeben werden?<sup>1)</sup>

4: es steht ein breites Spektrum von abgestuften Zugriffsrechten zur Verfügung

2: es stehen Zugriffsrechte zur Verfügung

0: es stehen keine Zugriffsrechte zur Verfügung

Lock-Mechanismus

Können einzelne Dokumente vor Änderungen von Teamkollegen geschützt werden?<sup>2)</sup>

4: es können einzelne Dokumente geschützt werden

2: es können Dokumentengruppen geschützt werden

0: es können die Dokumente nicht geschützt werden

---

1) Vgl. Balzert /Anforderungen/ 109; Balzert /Prinzipien/ 7; Koslowski /Unterstützung partizipativer Systementwicklung/ 127

2) Vgl. Gane /Computer-Aided Software Engineering/ 90; Gewalt, Haake, Pfadler /Software Engineering/ 189

## 2.2.6 Konfigurationsmanagement

### Versionsverwaltung

Können mehrere Versionen eines Projekts in der Entwicklungsdatenbank verwaltet werden?<sup>1)</sup>

4: es können mehrere Versionen verwaltet werden

2: es können einige Versionen verwaltet werden

0: es kann lediglich eine Version verwaltet werden

### Versionsabgleich

Können Versionen automatisiert verglichen werden?

4: ein Vergleich mehrerer Versionen kann durchgeführt werden

2: ein Vergleich zweier Versionen kann durchgeführt werden

0: ein Vergleich kann nicht durchgeführt werden

---

1) Vgl. Leppert, Stork /Entwicklungsumgebungen/ 206; Stobbe /Softwareentwicklungsumgebungen/ 46; Wirtz /Softwareentwurf/ 328



### Änderungserfassung

Können Änderungen dokumentiert werden?<sup>1)</sup>

4: Änderungen werden automatisiert dokumentiert, der Entwickler erhält einen Prompt zur Eingabe der Änderungsgründe. Der verantwortliche Entwickler und der Zeitpunkt werden automatisiert gespeichert

2: Änderungen können dokumentiert werden

0: Änderungen können nicht dokumentiert werden

---

1) Vgl. Gane /Computer-Aided Software Engineering/ 91;  
Sommerville /Software Engineering/ 140

### 2.2.7 Allgemeine Bürodienste<sup>1)</sup>

#### Textverarbeitung

Wird eine Textverarbeitung zur Verfügung gestellt?

- 4: eine Textverarbeitung ist integriert
- 2: es besteht eine Schnittstelle zu einer Textverarbeitung
- 0: es wird keine Textverarbeitung unterstützt

#### Tabellenkalkulation

Wird ein Tabellenkalkulationsprogramm zur Verfügung gestellt?

- 4: ein Tabellenkalkulationsprogramm ist integriert
- 2: es besteht eine Schnittstelle zu einem Tabellenkalkulationsprogramm
- 0: es wird keine Tabellenkalkulation unterstützt

#### Kalenderfunktion

Wird die Terminplanung durch eine Kalenderfunktion unterstützt?

- 4: eine Kalenderfunktion ist integriert
- 2: es besteht eine Schnittstelle zu einer Kalenderfunktion
- 0: es wird keine Terminplanung unterstützt

---

1) Vgl. Boehm u. a. /Improving Productivity/ 32; McClure /Automation/ 19; Österle /Computer Integrated Software-Engineering/ 15



Electronic Mail

Wird Electronic Mail unterstützt?

4: ein Electronic Mail System ist integriert

2: es besteht eine Schnittstelle zu einem  
Electronic Mail System

0: es wird kein Electronic Mail unterstützt

## 2.3 Gestaltungsorientierte Auswahlkriterien

### 2.3.1 Methoden

#### 2.3.1.1 Structured Analysis (SA)

##### 2.3.1.1.1 Benutzerführung

#### Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

#### Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

#### Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

### 2.3.1.1.2 Datenflußdiagramm



Symbole

Können die Symbole der Methode dargestellt werden?<sup>1)</sup>

- 0 Prozeß
- 0 Datenspeicher
- 0 Datenfluß
- 0 External Entity
- 0 Suchargument<sup>2)</sup>
- 0 Prozedurale Ergänzungen<sup>3)</sup>
- 0 Materialfluß<sup>4)</sup>

4: es stehen alle Symbole zur Verfügung

2: es stehen die Symbole Prozeß, Datenspeicher, Datenfluß und External Entity zur Verfügung

0: die Symbole stehen nicht zur Verfügung

- 
- 1) Vgl. DeMarco /Structured Analysis/ 40; Gane, Sarson /Structured Systems Analysis/ 75-90; Orr u. a. /Methodology/ 224
  - 2) In der Notation von Gane und Syrson können Suchargumente durch ein kleines gleichseitiges Dreieck dargestellt werden. Vgl. Gane /Computer-Aided Software Engineering/ 10; Gane, Sarson /Structured System Analysis/ 46
  - 3) Der Gebrauch von Prozeduralen Ergänzungen sollte in Datenflußdiagrammen möglichst vermieden werden. Dennoch sind sie in der Notation von DeMarco vorgesehen und so wird die Existenz dieser Symbole als Kriterium mit aufgenommen. Vgl. DeMarco /Structured Analysis/ 61
  - 4) In der Notation von Gane und Sarson sind Materialflüsse zugelassen. Sie werden durch dicke Pfeile dargestellt. Vgl. Martin, McClure /Structured Techniques/ 157



#### Attributierung von Datenflüssen

Können die Datenflüsse attributiert werden?

4: komfortabel

2: ja

0: nein

#### Wiederholungskennzeichen

Werden bei der Mehrfachverwendung eines Symbols Wiederholungskennzeichen eingefügt?<sup>1)</sup>

4: es werden Wiederholungskennzeichen eingefügt, aus denen die Anzahl der Wiederholungen ersichtlich wird

2: es wird ein Wiederholungskennzeichen eingefügt

0: es wird kein Wiederholungskennzeichen eingefügt

#### Darstellung sich kreuzender Linien

Wie werden sich kreuzende Linien im Datenflußdiagramm dargestellt?

4: kreuzende Linien werden durch einen Bogen dargestellt

0: Linien werden über andere Linien gelegt

---

1) Vgl. Gane /Computer-Aided Software Engineering/ 7; Gane, Sarson /Structured Systems Analysis/ 38-39; Martin, McClure /Structured Techniques/ 157



### Beschriftungszeitpunkt

Können Prozesse erst nachträglich beschriftet werden?<sup>1)</sup>

- 4: Prozesse können nachträglich beschriftet werden bzw. ihre Namen sind leicht änderbar
- 2: Prozesse können nachträglich beschriftet werden
- 0: Prozesse müssen sofort beschriftet werden



### Notationenauswahl

Kann zwischen der Notation von DeMarco und der von Gane und Sarson gewählt werden?<sup>2)</sup>

- 4: es kann zwischen den Notationen gewählt werden, bereits erstellte Dokumente werden automatisiert konvertiert
- 2: es kann zwischen den Notationen gewählt werden
- 0: es wird nur eine Notation angeboten



### Visuelle Hervorhebung

Können bestimmte Symbole des Datenflußdiagramms visuell hervorgehoben werden?<sup>3)</sup>

- 4: verschiedene Symboltypen oder bestimmte Sachverhalte können in unterschiedlichen Farben dargestellt werden, einzelne Objekte können visuell hervorgehoben (highlight) werden
- 2: es können verschiedene Symboltypen oder bestimmte Sachverhalte in unterschiedlichen Farben dargestellt werden
- 0: eine visuelle Hervorhebung ist nicht möglich

---

1) Vgl. DeMarco /Structured Analysis/ 64

2) Vgl. Martin, McClure /Structured Techniques/ 155-157

3) Vgl. McMenamin, Palmer /Strukturierte Systemanalyse/ 190

### Gruppierung

Können Aktivitäten zu Gruppen zusammengefaßt werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

### Verbindung mehrerer DFDs

Können mehrere Datenflußdiagramme auf unteren Ebenen zu einem Datenflußdiagramm zusammengefaßt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

### Hierarchie

Können Prozesse durch ein hierarchisch untergeordnetes Datenflußdiagramm dargestellt werden?<sup>3)</sup>

4: die Verfeinerung wird durch visuelle Darstellung und automatisierte Prozeßnumerierung unterstützt.<sup>4)</sup> Es kann zwischen Diagrammlevels gewechselt werden.

2: es kann zwischen Diagrammlevels gewechselt werden

0: eine Verfeinerung wird nicht unterstützt

- 
- 1) Vgl. McMenamin, Palmer /Strukturierte Systemanalyse/ 175
  - 2) Vgl. McMenamin, Palmer /Strukturierte Systemanalyse/ 154-157
  - 3) Vgl. Fisher /CASE/ 65; Gane, Sarson /Structured Systems Analysis/ 46
  - 4) Vgl. DeMarco /Structured Analysis/ 80; Gane, Sarson /Structured Systems Analysis/ 46; Fisher /CASE/ 66



#### Gestaltung niedriger Levels

Sind in einem hierarchisch untergeordneten Datenflußdiagramm Informationen aus den höheren Diagrammebenen erkennbar?

- 4: Prozeßgrenzen und Datenflüsse werden in die unteren Hierarchieebenen übernommen
- 2: Prozeßgrenzen oder Datenflüsse werden in die unteren Hierarchieebenen übernommen
- 0: es werden keine Informationen in die unteren Hierarchieebenen übernommen



#### Fortschreiben von Änderungen

Werden Änderungen im Datenflußdiagramm in anderen Ebenen fortgeschrieben?<sup>1)</sup>

- 4: die Änderungen in einem Diagrammlevel werden automatisiert in den hierarchisch höheren Ebenen fortgeschrieben
- 2: die Änderungen werden nicht fortgeschrieben, der Entwickler wird jedoch von der SEU zu Änderung der entsprechenden Diagramme aufgefordert
- 0: die Änderungen werden nicht fortgeschrieben, der Entwickler wird auf entstehende Inkonsistenzen nicht aufmerksam gemacht

---

1) Vgl. Martin, McClure /Structured Techniques/ 163



## Kontrollen

Kann ein Verstoß gegen die Regeln der Methode entweder bei der Eingabe direkt oder durch Prüfberichte kontrolliert werden?<sup>1)</sup>

Bspw.:

- 0 Verbindung zweier External Entities
  - 0 Verbindung zweier Prozesse
  - 0 direkte Verbindung eines External Entity mit einem Datenspeicher
  - 0 External Entity in einer niedrigeren Hierarchstufe als das Kontextdiagramm
  - 0 Datenspeicher im Kontextdiagramm
  - 0 nicht verbundene Datenflüsse, Datenspeicher oder Prozesse
  - 0 Datenflüsse der hierarchisch untergeordneten Datenflußdiagramme, die nicht mit den hierarchisch übergeordneten Datenflußdiagrammen abgestimmt sind.
  - 0 nicht benannte Symbole
- 4: es werden ein hohe Anzahl von Regelverstößen gemeldet
- 2: es werden die meisten der Regelverstöße gemeldet
- 0: die Einhaltung der Regeln wird nicht kontrolliert

---

1) Vgl. Fisher /CASE/ 69-73; Scheibl /Kommerzielle Software-Entwicklung/ 331; Schulz /Software-Entwurf/ 85

2.3.1.1.3 Data Dictionary

Vgl. 2.2.1 Informationsverwaltung

2.3.1.1.4 Entity-Relationship Diagramm

Vgl. 2.3.1.2 Entity-Relationship Diagramm

2.3.1.1.5 Prozeßspezifikation

Vgl. 2.3.1.3 Structured Design

2.3.1.1.6 Structured English, Pseudocode<sup>1)</sup>

Editor

Wird ein Editor für die Generierung von Structured English oder Pseudocode angeboten?

4: es wird ein komfortabler Editor angeboten

2: es wird ein Editor angeboten

0: es wird kein Editor angeboten

Struktur

Werden im Editor Felder für die Auflistung von Eingabe- und Ausgabedaten bereitgestellt?

4: es existieren Felder für den Eintrag von Ein- und Ausgabedaten; beim Eintragen der Ein- und Ausgabedaten kann auf die Informationsdatenbasis zurückgegriffen werden

2: es existieren Felder für den Eintrag von Ein- und Ausgabedaten

0: es steht lediglich ein reiner Texteditor zur Verfügung

Regeln

Können die logischen Konstrukte auf Sequenz, Iteration und Alternative begrenzt werden?<sup>2)</sup>

4: die Begrenzung der Konstrukte kann kontrolliert werden

0: die Begrenzung kann nicht kontrolliert werden

---

1) Im Vergleich zum Pseudocode ist Structured English weniger strukturiert und somit der erzählenden Sprache ähnlicher als Pseudocode. Vgl. Martin, McClure /Structured Techniques/ 234

2) Vgl. DeMarco /Structured Analysis/ 204

2.3.1.1.7 Entscheidungstabellen, Entscheidungsbäume

Syntaktische Elemente

Können alle syntaktischen Elemente dargestellt werden?<sup>1)</sup>

0 Bedingungen

0 Bedingungsanzeiger

0 Aktionen

0 Aktionsanzeiger

4: besonders komfortabel

2: ja

0: nein

Konsolidierung

Können unterschiedliche Regeln, die zu derselben Aktion führen, zu einer Hauptregel zusammengefaßt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Hering /Software-Engineering/ 18; Strunz /Entscheidungstabellentechnik/ 15

2) Vgl. Hering /Software-Engineering/ 20; Strunz /Entscheidungstabellentechnik/ 47; Yourdon /Techniques/ 103



### Entscheidungstabellenverbund

Können mehrere Entscheidungstabellen zu einem Entscheidungstabellenverbund zusammengefaßt werden?<sup>1)</sup>

4: sie können zusammengefaßt werden, zur besseren Übersicht über die Struktur der Entscheidungstabellen kann ein Entscheidungsbaum eingesetzt werden

2: sie können zusammengefaßt werden

0: sie können nicht zusammengefaßt werden

### Vollständigkeitskontrolle

Können die Entscheidungstabellen automatisiert auf Vollständigkeit kontrolliert werden?

4: besonders komfortabel

2: ja

0: nein

### Widerspruchskontrolle

Können die Entscheidungstabellen automatisiert auf Widersprüche kontrolliert werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Hering /Software Engineering/ 18-19; Strunz /Entscheidungstabellentechnik/ 24

2) Vgl. Hering /Software Engineering/ 20; Strunz /Entscheidungstabellentechnik/ 73



### Realisierung

Können die Entscheidungstabellen in lauffähige Programme transformiert werden?<sup>1)</sup>

4: es werden lauffähige Programme erstellt

2: es werden Programme erstellt, bei denen einige Änderung manuell erstellt werden müssen

0: die Entscheidungstabellen können nicht transformiert werden

---

1) Vgl. Hering /Software-Engineering/ 20; Yourdon /Techniques/ 108

2.3.1.2 Entity-Relationship Ansatz

2.3.1.2.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.2.2 Entity-Relationship Diagramm



Symbole

Können die elementaren Symbole dargestellt werden?<sup>1)</sup>

0 Entity

0 Attribut

0 Domaine

0 Beziehung

4: es können alle Symbole dargestellt werden

3: es können Entities, Attribute und Beziehungen dargestellt werden

2: es können lediglich Entities und Beziehungen dargestellt werden

0: es können keine External Entities oder Beziehungen dargestellt werden



Alias

Können einem Entity mehrere Namen gegeben werden?<sup>2)</sup>

4: es können mehrere Namen vergeben werden, diese erscheinen im Diagramm

2: es können mehrere Namen vergeben werden, sie werden nur im Data Dictionary aufgeführt

0: ein Vergabe mehrerer Namen ist nicht möglich

---

1) Vgl. Barker /CASE\*Method/ 1-1; Martin, McClure /Structured Techniques/ 307; Scheer /Wirtschaftsinformatik/ 20

2) Vgl. Barker /CASE\*Method/ 3-1 - 3-2



### Notationenauswahl

Kann zwischen mehreren Notationen gewählt werden?

0 Chen

0 Bachman

0 Merise

0 Sonstige

4: es kann zwischen mehreren Notationen gewählt werden, bereits erstellte Dokumente werden automatisiert konvertiert

2: es kann zwischen Notationen gewählt werden

0: es steht nur eine Notation zur Verfügung



### n-ary Beziehungen

Können Beziehungen zwischen mehr als zwei Entities dargestellt werden?<sup>1)</sup>

4: es können n-ary Beziehungen dargestellt werden, hierbei wird zwischen ausschließenden und einschließenden Beziehungen unterschieden

2: es können n-ary Beziehungen dargestellt werden

0: es können nur binäre Beziehungen dargestellt werden

---

1) Vgl. Chen /Entity-Relationship Approach/ 17; Gane /Computer-Aided Software Engineering/ 28; Martin /Information Engineering, Book II/ 162

### Rekursive Beziehungen

Können rekursive Beziehungen gezeigt werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

### Optionalität

Kann zwischen Kann- und Muß-Beziehungen unterschieden werden?<sup>2)</sup>

4: es kann zwischen Kann- und Muß-Beziehungen im Diagramm unterschieden werden

0: es kann nicht zwischen Kann- und Muß-Beziehungen unterschieden werden

### Abhängigkeiten

Können Abhängigkeiten von Entities im Diagramm dargestellt werden?<sup>3)</sup>

4: bei der Darstellung von abhängigen Entities wird zwischen existenziell abhängigen und ID-abhängigen Entities unterschieden

2: Abhängigkeiten können dargestellt werden

0: Abhängigkeiten können nicht dargestellt werden

---

1) Vgl. Barker /CASE\*Method/ 3-4; Gane /Computer-Aided Software Engineering/ 28; Martin, McClure /Structured Techniques/ 302, 322

2) Vgl. Barker /CASE\*Method/ 2-4, 3-3; Chen /Entity-Relationship Approach/ 23; Scheer /Wirtschaftsinformatik/ 24

3) Vgl. Chen /Entity-Relationship Approach/ 23-24

### Kardinalität

Kann die Kardinalität einer Beziehung festgelegt werden?<sup>1)</sup>

4: es können qualifizierte Kardinalitäten dargestellt werden<sup>2)</sup>

2: es können Kardinalitäten dargestellt werden

0: es können keine Kardinalitäten dargestellt werden

### Kardinalitäts-Platzhalter

Können Platzhalter ("?") für die Kardinalität eingetragen werden?<sup>3)</sup>

4: besonders komfortabel

2: ja

0: nein

### Identifikationsschlüssel

Können Schlüssel zur Identifizierung von Entities und Beziehungen dargestellt werden?<sup>4)</sup>

4: die Aufnahme von Schlüsselattributen in das Diagramm ist optional

2: Schlüsselattribute werden im Diagramm dargestellt

0: Schlüsselattribute können nicht dargestellt werden

---

1) Vgl. Barker /CASE\*Method/ 2-13; Gane /Computer-Aided Software Engineering/ 28; Martin, McClure / Structured Techniques/ 321

2) Vgl. Barker /CASE\*Method/ 7-12

3) Vgl. Martin, McClure /Structured Techniques/ 321

4) Vgl. Chen /Entity-Relationship Approach/ 23; Scheer /Wirtschaftsinformatik/ 27

#### Zugriffsschlüssel

Können externe Zugriffsschlüssel im Diagramm dargestellt werden?

4: besonders komfortabel

2: ja

0: nein

#### Label-Plazierung

Werden die Bezeichnungen der Beziehungen automatisiert an eine bestimmte Position gestellt, so daß die Richtung der Beziehung verdeutlicht wird?<sup>1)</sup>

4: die Plazierung erfolgt so, daß die Richtung der Beziehung erkennbar ist

2: die Plazierung erfolgt nicht automatisiert, die Bezeichnung kann jedoch frei plaziert werden

0: es ist eine Plazierung der Bezeichnung vorgegeben, die nicht variiert werden kann, und es geht aus ihr nicht die Richtung der Beziehung hervor

---

1) Vgl. Barker /CASE\*Method/ 3-4; Martin, McClure /Structured Techniques/ 321





### Beziehungen zwischen Beziehungstypen

Können Beziehungen zwischen Beziehungstypen dargestellt werden?<sup>1)</sup>

4: für die Darstellung von Beziehungen zwischen Beziehungstypen ist ein besonderes Symbol vorgesehen

2: für die Darstellung von Beziehungen zwischen Beziehungstypen wird das Beziehungssymbol verwendet

0: Beziehungen zwischen Beziehungstypen können nicht dargestellt werden



### Hierarchie

Können Entity-Sub-Types und Entity-Super-Types dargestellt werden?<sup>2)</sup>

4: es können Entity-Sub-Types und Entity-Super-Types in verschiedenen Hierarchieebenen dargestellt werden, Beziehungen können sowohl zum Entity-Sub-Typ als auch zum Entity-Super-Typ dargestellt werden<sup>3)</sup>

2: es können Entity-Sub-Types und Entity-Super-Types gebildet werden

0: es können keine Entity-Sub-Types und keine Entity-Super-Types gebildet werden

---

1) Vgl. Scheer /Wirtschaftsinformatik/ 27

2) Vgl. Martin /Information Engineering, Book II/ 164; Martin, McClure /Structured Techniques/ 313; Scheer /Wirtschaftsinformatik/ 17-19

3) Vgl. Gane /Computer-Aided Software Engineering/ 29; Martin /Information Engineering, Book II/ 164



weitere Informationen

Können weitere Informationen im Diagramm dargestellt werden?

- 4: es können optional weitere Informationen im Diagramm dargestellt werden
- 2: es werden weitere Informationen im Diagramm dargestellt
- 0: es werden keine weiteren Informationen im Diagramm dargestellt



Eingabekontrolle

Wird der Entwickler bei Erstellung einer N:M-Beziehung zur Auflösung dieser aufgefordert?<sup>1)</sup>

- 4: der Entwickler wird zur Auflösung der Beziehung aufgefordert, das System besteht aber nicht darauf
- 2: der Entwickler wird an einer Erstellung einer N:M-Beziehung gehindert
- 0: eine Eingabekontrolle bzgl. der Beziehungen existiert nicht

---

1) Vgl. Barker /CASE\*Method/ 6-10; Gane /Computer-Aided Software Engineering/ 22

2.3.1.2.3 Data Dictionary



Informationen

Können Informationen über Entities und Beziehungen im Data Dictionary gespeichert werden?<sup>1)</sup>

0 Name

0 Attribut

0 Domaine

0 Beziehung

0 Kardinalität

0 Optionalität

4: es können alle Informationen im Data Dictionary gespeichert werden

2: es können einige Informationen im Data Dictionary gespeichert werden

0: es können nur wenige Informationen im Data Dictionary gespeichert werden

---

1) Vgl. Martin, McClure /Structured Techniques/ 309

2.3.1.2.4 Datenstrukturen



Datenstrukturdiagramm

Stehen für die logische Datenstruktur Diagramme zur Verfügung

0 Hierarchisches Modell

0 Netzwerkmodell

0 Relationenmodell

4: es stehen alle Modelle zur Verfügung

2: es steht ein Modell zur Verfügung

0: es steht kein Modell für die logische Datenstruktur zur Verfügung



Übersetzung

Wie erfolgt die Übersetzung des Entity-Relationship Modells in eine Datenstruktur?

4: es erfolgt eine automatisierte Übersetzung

2: der Entwickler wird durch die Übersetzungsprozedur geführt

0: die Übersetzung muß manuell erfolgen



Normalisierung

Werden beim Relationenmodell die Normalisierungsregeln beachtet?

4: es findet eine automatisierte Normalisierung statt

3: es wird die Wanderung der Schlüssel automatisiert

2: die Einhaltung der Normalisierungsregeln wird kontrolliert

0: die Normalisierung wird nicht unterstützt

Normalisierungsbeschränkung

Kann die Normalisierung auf bestimmte Records beschränkt werden?

4: besonders komfortabel

2: ja

0: nein

2.3.1.3 Structured Design

2.3.1.3.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.3.2 Datenflußdiagramm



Datenflußdiagramm für SA

Vgl. 2.3.1.1.2 Datenflußdiagramm



Aufteilung

Kann im Datenflußdiagramm zwischen Input (afferent branch), Transformation (central transform) und Output (efferent branch) aufgeteilt werden?<sup>1)</sup>

4: die einzelnen Bereiche können gruppiert und auseinanderdividiert werden

2: die einzelnen Bereiche können durch Linien von einander getrennt werden

0: eine Aufteilung ist nicht möglich



Prozedurale Ergänzungen

Können im Datenflußdiagramm Und- und Oder-Beziehungen dargestellt werden?<sup>2)</sup>

4: ja

0: nein

---

1) Vgl. Martin, McClure /Structured Techniques/ 443;  
Yourdon, Constantine /Structured Design/ 175-176

2) Vgl. Yourdon, Constantine /Structured Design/ 175

2.3.1.3.3 Structure Chart



Symbole

Können die Symbole der Methode dargestellt werden?<sup>1)</sup>

- 0 beliebiger Modul
- 0 wiederverwendbarer Modul
- 0 Datenmodul
- 0 Ein- und Ausgabegerät
- 0 Datenfluß
- 0 Steuerfluß
- 0 nicht definierter Fluß
- 0 Kommentar

4: sämtliche Symbole sind darstellbar

2: die meisten Symbole sind darstellbar

0: die Symbole sind nicht darstellbar



Kontrollkonstrukte

Können Kontrollkonstrukte dargestellt werden?<sup>2)</sup>

- 0 Sequenz
- 0 Selektion
- 0 Iteration

4: sämtliche Kontrollkonstrukte werden unterstützt

0: Kontrollkonstrukte werden nicht unterstützt

---

1) Vgl. Yourdon, Constantine /Structured Design/ 396-402; Schulz /Software-Entwurf/ 74-75



### Rekursive Strukturen

Können rekursive Strukturen gezeichnet werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

### Homologe Systeme

Können homologe Systeme dargestellt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

### Teilmodule

Können Teilmodule kopiert und mehrfach verwendet werden?<sup>3)</sup>

4: Teilmodule können mehrfach verwendet werden, dies wird im Structure Chart und in der Dokumentation vermerkt

2: Teilmodule können mehrfach verwendet werden

0: Teilmodule können nicht kopiert werden

---

1) Vgl. Schulz /Software-Entwurf/ 77; Yourdon, Constantine /Structured Design/ 300

2) Vgl. Schulz /Software-Entwurf/ 77; Yourdon, Constantine /Structured Design/ 305, 397

3) Vgl. Schulz /Software-Entwurf/ 76



### Modularisierung

Erfolgt eine Unterstützung bei der Modularisierung?<sup>1)</sup>

- 4: der Entwickler wird bei der Auswahl des Modularisierungsgrads durch Aufzeigen des Grads der Modulkopplung (coupling) und der internen Strukturiertheit (cohesion) unterstützt
- 2: der Entwickler wird bei der Auswahl des Modularisierungsgrads unterstützt
- 0: der Entwickler wird bei der Auswahl des Modularisierungsgrads nicht unterstützt



### Gruppierung

Können aus den logischen Modulen physische Einheiten gebildet werden?<sup>2)</sup>

- 4: der Entwickler wird bei der Gruppierung von logischen Modulen geführt
- 2: eine Gruppierung von logischen Modulen ist möglich
- 0: eine Gruppierung ist nicht möglich

---

1) Vgl. Fisher /CASE/ 98; Martin, McClure /Structured Techniques/ 455; Schlulz /Software-Entwurf/ 65-67, 68-70, 72

2) Vgl. Martin, McClure /Structured Techniques/ 435

2.3.1.3.4 Transformation



Transformation

Wie wird die Transformation der Informationen aus dem Datenflußdiagramm in das Structure Chart unterstützt?<sup>1)</sup>

- 4: die Transformation erfolgt automatisiert
- 2: die Transformation wird durch Einsicht in das Datenflußdiagramm aus dem Structure Chart oder durch die Möglichkeit unterstützt, sich Listen über die Prozesse des Datenflußdiagramms im Structure Chart zeigen zu lassen
- 0: die Transformation wird nicht unterstützt

---

1) Vgl. Fisher /CASE/ 97; Martin, McClure /Structured Techniques/ 441

2.3.1.4 Structured Analysis and Design Technique (SADT)

2.3.1.4.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfefunktionen?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.4.2 Diagramm



Symbole

Können die elementaren Symbole der Data- und Aktigramme dargestellt werden?<sup>1)</sup>

0 Kasten

0 Pfeil

4: besonders komfortabel

2: ja

0: nein



Hierarchie

Können die Objekte in den Kästen durch ein weiteres Diagramm gleichen Typs verfeinert werden?<sup>2)</sup>

4: die Verfeinerung wird durch visuelle Darstellung unterstützt.<sup>3)</sup> Es kann bequem zwischen Diagrammlevels gewechselt werden.

2: es kann bequem zwischen Diagrammlevels gewechselt werden

0: eine Verfeinerung wird nicht unterstützt

- 
- 1) Vgl. Scheibl /Kommerzielle Software-Entwicklung/ 340; Schulz/ Software-Entwurf/ 60-61; Wirtz /Softwareentwurf/ 332
  - 2) Vgl. Scheibl /Kommerzielle Software-Entwicklung/ 340; Wirtz /Softwareentwurf/ 332
  - 3) Vgl. DeMarco /Structured Analysis/ 80; Gane, Sarson /Structured Systems Analysis/ 46; Fisher /CASE/ 66

### Konsistenz

Kann die Konsistenz zwischen Data- und Akti-  
gramm überprüft werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

### Fortschreiben von Änderungen

Werden Änderungen in Diagrammen in  
anderen Ebenen fortgeschrieben?

4: die Änderungen in einem Diagrammlevel wer-  
den automatisiert in den hierarchisch höhe-  
ren Ebenen fortgeschrieben

2: die Änderungen werden nicht fortgeschrie-  
ben, der Entwickler wird jedoch von der SEU  
zu Änderung der entsprechenden Diagramme  
aufgefordert

0: die Änderungen werden nicht fortgeschrie-  
ben, der Entwickler wird auf entstehende  
Inkonsistenzen nicht aufmerksam gemacht

### Knotenverzeichnis

Kann ein Knotenverzeichnis erstellt werden?

4: ein Knotenverzeichnis wird automatisiert  
erstellt

2: ein Knotenverzeichnis kann erstellt werden

0: ein Knotenverzeichnis kann nicht erstellt  
werden

---

1) Vgl. Scheibl /Kommerzielle Software-Entwicklung/ 340

2.3.1.5 Jackson System Development (JSD)

2.3.1.5.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.5.2 Diagramme

Textverarbeitung

Vgl. 1.7.1 Textverarbeitung

Baumdiagramm

Können die den Objekten zugeordneten Aktionen in einem Jackson-Diagramm dargestellt werden?<sup>1)</sup>

Können alle Kontrollkonstrukte mit dem Diagramm dargestellt werden?<sup>2)</sup>

0 Sequenz

0 Selektion

0 Iteration

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Schulz /Software-Entwurf/ 129-130

2) Vgl. Wirtz /Softwareentwurf/ 337-339



### Regeln

Wird gewährleistet, daß die Kontrollkonstrukte nicht gemischt werden können?<sup>1)</sup>

4: Regelverstöße werden sofort bei der Eingabe gerügt

2: Nach Fertigstellung des Diagramms kann dieses auf Regelverstöße untersucht werden

0: es kann gegen die Regeln verstoßen werden, ohne daß das System den Entwickler hierauf aufmerksam macht

### Programminversion

Kann die Entwicklung einer Programminversion dargestellt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

### System-Spezifikations-Diagramm

Können die Prozesse in einem System-Spezifikations-Diagramm dargestellt werden?<sup>3)</sup>

0 Datenstromverknüpfung (Kreis)

0 Zustandsvektorverknüpfung (Raute)

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Schulz /Software-Entwurf/ 101

2) Vgl. Martin, McClure /Structured Techniques/ 483; Schulz /Software-Entwurf/ 112

3) Vgl. Schulz /Software-Entwurf/ 131



Pseudocode

Vgl. 2.3.1.1.6 Structured English, Pseudocode

2.3.1.6 Hierarchy plus Input-Prozess-Output-Methode (HIPO)

2.3.1.6.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.6.2 Diagramme



Strukturübersicht

Kann der Grobentwurf in einem Baumdiagramm dargestellt werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Hering /Software-Engineering/ 12-13; Scheibl  
/Kommerzielle Software-Entwicklung/ 385; Schulz  
/Software-Entwurf/ 91



### IPO-Diagramm

Können die Funktionselemente durch ein IPO-Diagramm dargestellt werden?<sup>1)</sup>

Sind die Symbole der HIPO-Methode darstellbar?<sup>2)</sup>

- 0 ausgelagerte Funktion
- 0 Mehrfachverwendungsfunktion
- 0 Funktion eines anderen Systems
- 0 Datenfluß
- 0 Steuerfluß
- 0 Datenflußkonnektor
- 0 Steuerflußkonnektor
- 0 Datenbezug
- 0 Datenanzeiger

4: es sind alle Symbole darstellbar

2: es sind die wichtigsten Symbole darstellbar

0: die Symbole sind nicht darstellbar



### Regeln

Wird die Einhaltung der methodenspezifischen Regeln überprüft?<sup>3)</sup>

4: Regelverstöße werden sofort bei der Eingabe gerügt

2: Nach Fertigstellung des Diagramms kann dieses auf Regelverstöße untersucht werden

0: es kann gegen die Regeln verstoßen werden, ohne daß das System den Entwickler hierauf aufmerksam macht

---

1) Vgl. Gewalt, Haake, Pfadler /Software Engineering/ 113-114; Hering /Software-Engineering/ 11; Schulz /Software-Entwurf/ 93

2) Vgl. Hering /Software Engineering/ 12

3) Vgl. Schulz /Software-Entwurf/ 91

2.3.1.7 Warnier-Orr Methode

2.3.1.7.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.7.2 Diagramm



Warnier-Orr Diagramm

Können Daten- und Prozeßstrukturen in Warnier-Orr Diagrammen dargestellt werden?<sup>1)</sup>

Können die elementaren Symbole dargestellt werden?<sup>2)</sup>

- 0 Klammer
- 0 Eintrag
- 0 logische Sequenz (=Sequenz)
- 0 alternative Struktur (Selektion)
- 0 repetitive Struktur (Iteration)
- 0 inklusives Oder
- 0 exklusives Oder
- 0 Endebedingung (als Fußnote "?n")

4: es sind alle Symbole darstellbar

2: es sind die wichtigsten Symbole darstellbar

0: die Symbole sind nicht darstellbar



Hierarchie

Kann das Diagramm entsprechend dem gewünschten Detaillierungsgrad in Diagrammsegmente aufgespalten und wieder zusammengeführt werden?<sup>3)</sup>

4: besonders komfortabel

2: ja

0: nein

---

1) Vgl. Martin, McClure /Structured Techniques/ 491, 498; Schulz /Software-Entwurf/ 138

2) Vgl. Martin, McClure /Structured Techniques/ 197-198, 200; Schulz /Software-Entwurf/ 138

3) Vgl. Fisher /CASE/ 109



Entscheidungstabellen<sup>1)</sup>

Vgl. 2.3.1.1.7 Entscheidungstabellen, Entscheidungsbäume

---

1) Vgl. Schulz /Software-Entwurf/ 137



2.3.1.8 Information Engineering

2.3.1.8.1 Benutzerführung

Methodenbeschreibung

Werden Methodenhandbücher mitgeliefert?

4: Methodenhandbücher gehören zum Lieferumfang

2: Methodenhandbücher können erworben werden

0: Methodenhandbücher sind vom Anbieter nicht zu beziehen

Vorgehensweise

Wird die Methode in eine Vorgehensweise eingebettet?<sup>1)</sup>

4: eine Vorgehensweise ist online integriert

2: eine Vorgehensweise liegt schriftlich vor

0: eine Vorgehensweise wird nicht angeboten

Hilfefunktion

Gibt es zur Unterstützung der Methode Hilfenhinweise?<sup>2)</sup>

4: es stehen ausführliche Hinweise zur Unterstützung der Methode zur Verfügung

2: es stehen knappe Hinweise zur Unterstützung der Methode zur Verfügung

0: die Hilfefunktion bezieht sich nicht auf die Unterstützung der Methode

---

1) Vgl. McClure /Automation/ 79; Scheibl /Kommerzielle Software-Entwicklung/ 43

2) Vgl. McClure /Automation/ 79

2.3.1.8.2 Diagramme

Textverarbeitung<sup>1)</sup>

Vgl. 1.7.1 Textverarbeitung

Entity-Relationship Diagramm<sup>2)</sup>

Vgl. 2.3.1.2.2. Entity-Relationship Diagramm

Data Analysis Diagramm<sup>3)</sup>

Können detaillierte Datenmodelle mit Hilfe von Data Analysis Diagrammen erstellt werden?

4: besonders komfortabel

2: ja

0: nein

Normalisierung<sup>4)</sup>

Vgl. 2.3.1.2.4 Normalisierung

---

1) Vgl. Martin, McClure /Structured Techniques/ 655

2) Vgl. Martin /Information Engineering/ 103

3) Vgl. Martin, McClure /Structured Techniques/ 660

4) Vgl. Martin, McClure /Structured Techniques/ 657

Repository<sup>1)</sup>

Vgl. 2.2.1 Informationsverwaltung

Dekompositionsdiagramme

Können die Funktionen eines Geschäftsbereichs mit Dekompositionsdiagrammen in Prozesse aufgeteilt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

Dependency Diagramme

Können zur Verdeutlichung der Abhängigkeiten der Prozesse Dependency Diagramme entworfen werden?<sup>3)</sup>

4: besonders komfortabel

2: ja

0: nein

Datenflußdiagramm

Können zur Verdeutlichung der Abhängigkeit der Prozesse Datenflußdiagramme gezeichnet werden?<sup>4)</sup>

Vgl. 2.3.1.1.2 Datenflußdiagramm

- 
- 1) Vgl. Martin /Information Engineering/ 104; 107
  - 2) Vgl. Martin /Information Engineering/ 112; Martin, McClure /Structured Techniques/ 657
  - 3) Vgl. Martin /Information Engineering/ 107, 112; Martin, McClure /Structured Techniques/ 657
  - 4) Vgl. Martin /Information Engineering, Book I/107, 112; Martin, McClure /Structured Techniques/ 657

### Prozeß-Entity-Matrix

Kann eine Matrix erstellt werden, die die Entities den Prozessen zuordnet, die diese lesen, aktualisierung oder schaffen?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

### Action Diagramm

Kann der Prozedur-Entwurf durch Action Diagramme unterstützt werden?<sup>2)</sup>

4: besonders komfortabel

2: ja

0: nein

### Entscheidungstabellen, -bäume<sup>3)</sup>

Vgl. 2.3.1.1.7 Entscheidungstabellen, -bäume

### Konvertierung

Können verschiedene Diagramme automatisiert in Action Diagramme konvertiert werden?<sup>4)</sup>

4: besonders komfortabel

2: ja

0: nein

- 
- 1) Vgl. Martin /Information Engineering, Book I/ 107
  - 2) Vgl. Martin /Information Engineering, Book I/ 112;  
Martin, McClure /Structured Techniques/ 657
  - 3) Vgl. Martin /Information Engineering, Book I/ 113;  
Martin, McClure / Structured Techniques/ 657
  - 4) Vgl. Martin, McClure /Structured Techniques/ 659



Prototyping<sup>1)</sup>

Vgl. 2.2.4 Prototyping

---

1) Vgl. Martin /Information Engineering, Book I/ 112;  
Martin, McClure /Structured Techniques/ 660

### 2.3.2 Kapazität



Anzahl Diagramme

Wieviele Diagramme können bei Verwendung eines alleinstehenden Rechner gezeichnet werden?<sup>1)</sup>

4: die Anzahl der Diagramme ist lediglich von der Hardwarekapazität begrenzt

3: mehr als 400

2: mehr als 200

1: mehr als 100

0: 100 oder weniger



Anzahl Symbole

Wieviel Symbole können in einem Diagramm generiert werden?<sup>2)</sup>

4: die Anzahl der Symbole ist lediglich von der Hardwarekapazität begrenzt

3: mehr als 200

2: mehr als 100

1: mehr als 50

0: 50 oder weniger

---

1) Vgl. McClure /Automation/ 39

2) Vgl. Gane /Computer-Aided Software Engineering/ 140



Anzahl Verfeinerungsebenen

Wieviel Verfeinerungen können erstellt werden?<sup>1)</sup>

4: die Anzahl der Verfeinerungsebenen ist lediglich von der Hardwarekapazität begrenzt

3: mehr als 15

2: mehr als 10

1: mehr als 5

0: 5 oder weniger

---

1) Vgl. Gane /Computer-Aided Software Engineering/ 140

### 2.3.3 Outputqualität

Seitenumbruch

Kann bei der Diagrammerstellung der Seitenumbruch berücksichtigt werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

Schriftgröße

Kann die Schriftgröße variiert werden?

4: es können mehrere Schriftgrößen festgelegt werden, die Auswirkungen auf die Beschriftungen der Symbole wird dem Entwickler angezeigt

2: es können einige Schriftgrößen festgelegt werden

0: die Schriftgröße kann nicht variiert werden

Symbolgröße

Kann die Symbolgröße variiert werden?

4: es können mehrere Symbolgrößen festgelegt werden, die Auswirkungen auf den Platzbedarf des Diagramms wird dem Entwickler angezeigt

2: es können einige Schriftgrößen festgelegt werden

0: die Symbolgröße kann nicht variiert werden

---

1) Vgl. Gane /Computer-Aided Software Engineering/ 138



Diagrammgröße

Kann die Diagrammgröße frei bestimmt werden?

4: besonders komfortabel

2: ja

0: nein

Label-Plazierung

Können Labels direkt an die gewünschte Stelle plaziert werden?

4: besonders komfortabel

2: ja

0: nein

Querdruck

Können Schriftzüge auch quer ausgedruckt werden?<sup>1)</sup>

4: besonders komfortabel

2: ja

0: nein

Ausschnittsdruck

Kann ein Ausschnitt des Diagramms ausgedruckt werden?

4: es kann ein Ausschnitt des Diagramm ausgedruckt werden, er kann auch gezoomt worden sein

2: es kann ein Ausschnitt ausgedruckt werden

0: es kann nur das gesamte Diagramm ausgedruckt werden

---

1) Vgl. Martin, McClure /Structured Techniques/ 125



Entwurf

Kann ein Entwurf ausgedruckt werden?

- 4: der Entwurf kann im Draft-Modus des Druckers ausgedruckt werden, das Diagramm wird auf eine Seite gedruckt
- 2: der Entwurf kann im Draft-Modus des Druckers ausgedruckt werden
- 0: der Draft-Modus des Druckers wird nicht unterstützt

#### 2.3.4 Integration

mehrere Diagramme

Können mehrere Diagramme gleichzeitig auf einem Bildschirm dargestellt werden?

4: es können mehrere Diagramme gleichzeitig auf einem Bildschirm dargestellt werden

2: es können zwei Diagramme gleichzeitig auf einem Bildschirm dargestellt werden

0: es kann lediglich ein Diagramm auf einem Bildschirm dargestellt werden

Sprung in einen anderen Editor

Kann aus einem Editor direkt ein anderer Editor aufgerufen werden?

4: es gibt eine Reihe von Querverbindungen

2: es gibt wenige Querverbindungen

0: zum Aufruf eines anderen Diagrammeditors muß in das Hauptmenü der SEU zurückgegangen werden

Auswertungen

Können Auswertungen von Informationen aus verschiedenen Werkzeugen durchgeführt werden? (Bspw. Entity-Funktionen-Matrix)<sup>1)</sup>

4: es stehen mehrere Matrizen zur Verfügung

2: es stehen mehrere Berichte zur Verfügung

0: es stehen keine Auswertungen zur Verfügung

---

1) Vgl. Martin /Information Engineering, Book II/ 171

### logische Integration

Sind die Werkzeuge logisch bzw. methodisch aufeinander abgestimmt?<sup>1)</sup>

4: die Werkzeuge bauen logisch bzw. methodisch aufeinander auf

2: die Werkzeuge überschneiden sich teilweise, widersprechen sich jedoch logisch bzw. methodisch nicht

0: die Werkzeuge widersprechen sich teilweise

### Multiuserfähigkeit

Ist die SEU multiuserfähig?<sup>2)</sup>

4: ja

0: nein

### Gemeinsames Repository

Können mehrere PCs auf eine gemeinsame Entwicklungsdatenbank zugreifen?<sup>3)</sup>

4: ein Zugriff auf eine gemeinsame Entwicklungsdatenbank ist möglich, der Zugriff erfolgt kontrolliert, so daß das System nicht von zwei Entwicklern gleichzeitig geändert werden kann

2: ein Zugriff auf eine gemeinsame Entwicklungsdatenbank ist möglich

0: eine gemeinsame Entwicklungsdatenbank wird nicht angeboten

---

1) Vgl. Ruf /Basis des Schnittstellen-Management-Ansatzes/ 96

2) Vgl. McClure /Automation/ 125

3) Vgl. Tobiasch /Anforderungen/ 19

AD/Cycle

Welche Stellung hat der Anbieter zur IBM hinsichtlich des AD/Cycle?<sup>1)</sup>

4: Product Part of AD/Cycle

2: AD/Cycle Vendor Program Participant

0: keine Beziehung zur IBM

---

1) Vgl. Merlyn, Boone /Ins and Outs/ 60