

Jan Geißelmann, Georg Herzwurm, Andreas Hierholzer, Michael Kunz\*

## Konzept zur Evaluierung von CASE-Tools

### Inhaltsverzeichnis

Vorbemerkung	212
1. Einleitung	212
1.1 Einführung zum CASE-Tool	213
1.2 Getestete Konfiguration	213
2. Bewertung genereller Kriterien	214
2.1. Entwicklungsdatenhaltung	214
2.2. Bürodienste	215
2.3. Teamfähigkeit	216
2.4. Installation	216
2.5. Benutzerfreundlichkeit	216
3. Bewertung spezieller Kriterien	217
3.1. Funktionalität im Hinblick auf ausgewählte Aktivitäten im Systemlebenszyklus	218
3.1.1. Analyse- und Entwurfsaktivitäten	218
3.1.1.1. Strukturierte Methoden	218
3.1.1.2. Objektorientierte Methoden	222
3.1.2. Realisierungsaktivitäten	223
3.1.3. Dokumentation	224
3.2. Funktionsumfang im Hinblick auf Methoden bzw. Vorgehensmodelle	225
3.2.1. Methodenintegration	225
3.2.2. Übergreifende Methoden/Vorgehensmodelle	225
4. Schlußbemerkung	226
4.1. Leistungsprofil	226
4.2. Ausblick	226

---

\* Jan Geißelmann, Dr. Georg Herzwurm, CEMS Master Dipl.-Kfm. Andreas Hierholzer, Dipl.-Kfm. Michael Kunz, Lehrstuhl für Wirtschaftsinformatik, Systementwicklung der Universität zu Köln

## Vorbemerkung

Der nachfolgende Beitrag dient der Beschreibung der Vorgehensweise bei der CASE-Tool-Evaluierung durch den Lehrstuhl. Er kann aber auch als Anregung für eigene durchzuführende Tool-Evaluierungen verwendet werden.

Zur Gewährleistung der Vergleichbarkeit der Ergebnisse ist die schriftliche Fixierung eines Evaluierungskonzeptes v. a. dann erforderlich, wenn wie im vorliegenden Falle ein Team von 7 Personen die Untersuchung durchführt. Trotz aller Bemühungen um einheitliche Bewertungsmaßstäbe differieren die Evaluierungsberichte in Art und Umfang, was einerseits aus der unterschiedlichen Struktur und Komplexität der untersuchten Werkzeuge, andererseits aber auch aus unterschiedlichen Schwerpunktsetzungen und Arbeitsweisen der Bewerter resultiert.

## 1. Einleitung

Die Bewertung und der Vergleich der Leistungsfähigkeit verschiedener CASE-Tools ist ein schwieriges Unterfangen. Um einen möglichst objektiven Vergleichsmaßstab zu haben, benötigt man ein Konzept zur Evaluierung von CASE-Tools, das einheitlich für jede CASE-Tool Untersuchung angewendet werden muß.

In dieser Untersuchung wird angenommen, daß den zukünftigen CASE-Tool Anwendern - neben allgemeinen Angaben zum Hersteller oder Anbieter des Tools - besonders die Unterstützungsmöglichkeiten des Tools für seine Softwareentwicklung interessieren. Schließlich ist für ihn auch die zukünftige Entwicklung des CASE-Tools von Bedeutung.

Die Basis der Untersuchung stellt ein vereinfachtes Softwareentwicklungsprojekt aus der Versicherungsbranche dar. Die Realisierung dieses Projektes dient als Beispiel für die Prüfung der relevanten Eigenschaften des CASE-Tools.

Obwohl lediglich die Leistungsfähigkeit des CASE-Tools im Hinblick auf die Entwicklung kommerzieller Anwendungssysteme geprüft wird, sind die Ergebnisse der Evaluierung branchenübergreifend und nicht nur für Versicherungsunternehmen von Nutzen.

## 1.1 Einführung zum CASE-Tool

Als Einführung zum CASE-Tool interessieren zunächst Angaben über den Hersteller bzw. Anbieter wie

- die Anzahl der Mitarbeiter,
- Hauptsitz und Geschäftsstellen in Deutschland,
- Gründungszeitpunkt sowie
- Kooperationen mit anderen CASE-Tool- oder Hardware-/Software-Herstellern und Art der Kooperationen.

Zum CASE-Tool sollten allgemeine Angaben über

- Anzahl der Installationen weltweit, europaweit und in Deutschland,
- den Einführungszeitpunkt des CASE-Tools,
- die Hardware-Architektur des CASE-Tools (PC, Client-Server, Hostsystem) der Aufbau und Art der Entwicklungsdatenhaltung
- die Software-Architektur des CASE-Tools, insbesondere
  - Art der Benutzerschnittstelle
  - phasenbegleitende und phasenübergreifende Werkzeuge
  - Entwicklungsdatenhaltung
  - Schnittstellen zwischen den Werkzeugen, zu CASE-Tools von Fremdanbietern sowie zum Zielsystem
  - Einbindung der Komponenten des CASE-Tools in das Betriebssystem

## 1.2 Getestete Konfiguration

Wichtig ist, daß zunächst die getestete Hardware- und Software-Konfiguration festgehalten wird.

Hierzu sollten Angaben gemacht werden

- zum Freigabezeitpunkt der getesteten Version,
- zu den getesteten Komponenten des CASE-Tools,

- zur eingesetzten Betriebssystem-Umgebung (DOS, MS-Windows, OS/2, UNIX etc.) und deren Leistungsmerkmalen (z. B. Grafikauflösung) sowie
- zur verwendeten Hardware (Prozessorleistung, Hauptspeicher, etc.).

## 2. Bewertung genereller Kriterien

Der Übersichtlichkeit halber wird eine Aufteilung in generelle und spezielle Kriterien vorgenommen.

Generelle Kriterien beziehen sich nicht explizit auf den Softwareentwicklungsprozeß. Von den zahlreichen in Frage kommenden Kriterien werden als wesentliche die Art und Funktionalität der Entwicklungsdatenhaltung und Bürodienste, Teamfähigkeit, Umfang der Installation und nicht zuletzt die Benutzerfreundlichkeit untersucht.

Spezielle Kriterien befassen sich mit der Funktionalität im Hinblick auf in dieser Untersuchung ausgewählte Aktivitäten im Systemlebenszyklus sowie mit dem Funktionsumfang im Hinblick auf Methoden bzw. Vorgehensmodelle.

### 2.1. Entwicklungsdatenhaltung

Zur Entwicklungsdatenhaltung sollten zunächst einmal Angaben gemacht werden über

- die Art des Zugriffes auf die projektbezogenen Entwicklungsdaten (Online oder Batch),
- den Umfang der Datenübernahme (z. B., ob alle Diagrammeingaben wie Beziehungen, Attribute etc. in die Entwicklungsdatenbank übernommen werden),
- das Zeitverhalten beim Zugriff auf die Entwicklungsdatenbank, d. h., ob ein ununterbrochenes Arbeiten möglich ist,
- die Möglichkeit eines Multiuserzugriffs (d.h., daß von zwei oder mehr Arbeitsplätzen *gleichzeitig* auf die Entwicklungsdaten problemlos zu-

gegriffen werden kann) und die Differenzierung des Zugriffsschutzes (z.B., ob Objekte stets gesperrt werden, wenn sie bearbeitet werden),

- die Art der Entwicklungsdatenbank (z. B. Standard-Datenbanksystem versus Eigenentwicklung, SQL-Basierung) und die Möglichkeit, mit Such-Komandos (z.B. SQL-Befehlen) auf sie zuzugreifen,
- die Art und der Umfang der Schnittstellen der Entwicklungsdatenbank sowie
- die Datensicherheit bei einem Absturz des Computersystems.

Sollte das CASE-Tool das Konfigurationsmanagement unterstützen, so sollten darüber hinaus Angaben gemacht werden über

- die Möglichkeit und den Komfort der Verwaltung von Versionen und Varianten auf unterschiedlichen Ebenen (Projekt, Teilprojekt, Modell, Ergebnis einer Aktivität (Grafik, Symbol, Text etc.)),
- die Unterstützung des Vergleiches zwischen Versionen,
- die automatische Erstellung von Releases,
- die Verwaltung von Querbezügen (Referenzen) zwischen den Dokumenten und
- den Statusmechanismus (Bearbeitung, Freigabe etc.).

## 2.2. Bürodienste

Unter "Bürodienste" soll die Ausübung aller üblicherweise in einem Büro anfallenden Aufgaben verstanden werden, also Anfertigen von Text- oder Grafikdokumenten, Tabellenkalkulation, Kommunikation und ähnliches.

Es sollte daher festgestellt werden, inwieweit das CASE-Tool diese Dienste unterstützt, z. B. mit einem Grafikeditor (für Präsentationen), einer Textverarbeitung (für die Dokumentation), E-Mail etc. und inwieweit die Integration dieser Bürohilfen in die Systemumgebung (z. B. durch DDE/OLE von MS-Windows) gewährleistet ist.

### 2.3. Teamfähigkeit

Wesentlich für die Entwicklung großer Systeme ist die Unterstützung der arbeitsteiligen Prozesse. Ein CASE-Tool muß daher

- multiuserfähig sein,
- Daten zwischen Benutzern austauschen können und
- das Projektmanagement (inkl. Regelungen über Zugriffsberechtigungen) unterstützen.

### 2.4. Installation

Die Installation von komplexen CASE-Tools ist nicht immer trivial. Daher muß geprüft und festgehalten werden, ob

- Probleme auftreten,
- die Hardware (inkl. Netz, Betriebssystem etc.) vollständig und automatisch erkannt wird und
- die Produktdokumentation zur Installation ausreichend ist.

### 2.5. Benutzerfreundlichkeit

Obwohl die Benutzerfreundlichkeit nur subjektiv festgestellt werden kann, ist es durchaus sinnvoll z. B. darauf zu achten, daß das CASE-Tool

- einen Standard für die Benutzerschnittstelle einhält (z.B. SAA CUA-Standard (1989), Motif),
- die volle Bildschirmgröße (z. B. Grafikauflösung 768 x 1024 Bildpunkte) ausnutzt,
- eine taugliche, kontextsensitive Online-Hilfe anbietet, die nicht nur auf Handbücher verweist,
- eine UNDO-Funktion anbietet und

- eine vollständige, strukturierte, übersichtliche und leicht handhabbare Produktdokumentation (inkl. Methodenhandbücher) bereitstellt.

Besondere Aufmerksamkeit sollte bei einem CASE-Tool dem Diagramm-editor gewidmet werden. Es ist insbesondere wichtig, daß dieser

- komfortabel die gängigen Editierfunktionen Kopieren, Ändern, Löschen, Beschriften von Objekten, stufenloses Zoomen, "Lasso"-Funktion zum Markieren etc. ermöglicht,
- Zeichenhilfe leistet (z. B. automatisches Zeichnen, Begradigen von Linien, Kennzeichnen von Linienüberschneidungen, Verhinderung von Überschneidungen von Linien und Symbolen),
- eine Überlagerung von Symbolen unterbindet oder
- die freie Anordnung von Symbolen zuläßt.

### 3. Bewertung spezieller Kriterien

Die Bewertung der speziellen Kriterien gliedert sich in die Bestimmung der Funktionalität im Hinblick auf die Aktivitäten im Systemlebenszyklus und in die Bestimmung des Funktionsumfanges im Hinblick auf Methoden bzw. Vorgehensmodelle.

Im folgendem Abschnitt soll zunächst auf die Unterstützungsleistung des CASE-Tools im Hinblick auf die Aktivitäten im Systemlebenszyklus eingegangen werden. Zur Vereinfachung werden hierbei nur Analyse- und Entwurfsaktivitäten, Realisierungsaktivitäten sowie die Dokumentation der Entwicklung berücksichtigt, da besonders diese von einem CASE-Tool unterstützt werden sollten.

Im darauf folgenden Kapitel wird abschließend der Funktionsumfang im Hinblick auf Methoden bzw. Vorgehensmodelle untersucht. Hierbei interessiert besonders die Güte der Methodenintegration sowie die Unterstützung weiterer (noch) nicht betrachteter, aber vom CASE-Tool bereitgestellter übergreifender Methoden bzw. Vorgehensmodelle.

### 3.1. Funktionalität im Hinblick auf ausgewählte Aktivitäten im Systemlebenszyklus

Zuerst soll ein kurzer Überblick gegeben werden darüber,

- welche Methoden bzw. Vorgehensweisen und
- welche phasenbezogenen und phasenübergreifenden Aktivitäten

grundsätzlich vom CASE-Tool unterstützt werden.

#### 3.1.1. Analyse- und Entwurfsaktivitäten

Bei der Unterstützung von Analyse- und Entwurfsaktivitäten muß zwischen strukturierter und objektorientierter Vorgehensweise unterschieden werden, da sich hieraus unterschiedliche Anforderungen, die an das CASE-Tool gestellt werden, ergeben.

##### 3.1.1.1. Strukturierte Methoden

Von Relevanz ist,

- welche Softwareentwicklungsprinzipien (Bottom-Up, Top-Down, Modularität etc.) besonders gefördert oder mißachtet werden und
- welche Methode/Beschreibungsmittel (ERM, SA etc.), ggf. mit welchen grundsätzlichen Einschränkungen, angeboten werden.

#### Datenmodellierung (Entity-Relationship-Modelling, falls unterstützt)

Im Rahmen der Datenmodellierung kann festgestellt werden,

- ob sie in ein Vorgehensmodell integriert ist,
- welche Notationen verwendet werden (Chen, Bachmann, Merise, Martin etc.),

- ob die elementaren Symbole Entität (Superentität und Subentität), Relation, Attribut, Schlüsselattribut, Kardinalität dargestellt werden können,
- ob eine Entität mehrere Namen (Alias) erhalten kann,
- ob Attribute einer Superentität automatisch an ihre Subentitäten vererbt werden,
- ob assoziative Entitäten darstellbar sind,
- welche Kardinalitäten dargestellt werden können (1:M, 1:1, M:N, 1:C, 1:CN etc., sowie Platzhalter für Kardinalitäten),
- ob rekursive Beziehungen darstellbar sind,
- ob Beziehungen in beiden Richtungen Namen zugewiesen werden können,
- ob existentielle Abhängigkeiten einer Entität (weak/strong) dargestellt werden können,
- ob bei der Attributierung Wertebereiche, Auswahllisten, Feldlängen, Defaults und dabei einem Attribut verschiedene Ausprägungen zugewiesen werden können,
- ob bei der Attributierung Datentypen für Datenelemente definiert werden können und eine Überprüfung zwischen Wertebereich etc. und Datentypen erfolgt,
- ob Fremdschlüssel für Entitäten vergeben werden können - sofern möglich- auch automatisch,
- ob eine werkzeuggestützte und auf bestimmte Tabellen beschränkbare Normalisierung möglich ist,
- ob der Stand der Normalisierung ausgewertet werden kann,

- ob Views/Selects auf das Datenmodell definiert werden können,
- ob besondere (selbsterstellte und/oder bereits vorformulierte) Integritätsbedingungen unterstützt werden und
- ob die Informationen (Entity, Relation, Attribut, Schlüssel, Kardinalität etc.) des ERM im Data Dictionary ablegbar sind.

### **Funktionsmodellierung (Structured Analysis, falls unterstützt)**

Bei der Datenflußdiagrammerstellung kommt es besonders darauf an,

- daß mit dem CASE-Tool die elementaren Symbole Prozeßfunktion, Datenspeicher, Datenfluß, Schnittstelle von und zur Außenwelt im DFD dargestellt werden können,
- welche Notationen verwendet werden (z. B. Gane & Sarson, DeMarco),
- daß hierarchisch untergeordnete DFDs dargestellt werden können und dabei Prozeßgrenzen und Datenflüsse übernommen werden,
- daß einzelne Prozeßfunktionen zu Gruppen zusammengefaßt werden können,
- daß Änderungen automatisch in alle Hierarchieebenen übernommen werden,
- daß alle Diagrammeingaben in die Entwicklungsdatenbank übernommen werden,
- ob für jeden Prozeß (Funktion) bzw. für jedes Blatt/Knoten zwingend eine Kurzinformation (MiniSpecs) angelegt werden muß und wenn ja, in welcher Form (Pseudo-Code, Structured English, Entscheidungsbaum, Entscheidungstabelle)
- daß eine Qualitätssicherungsfunktion Fehleingaben bzw. Methodenverstöße erkennt (z. B. Eintragung einer neuen externen Entität auf der 2. Verfeinerungsebene, Verbindung einer externen Entität und eines Datenspeichers durch einen Datenfluß),

- daß bei der Mehrfachverwendung eines Symbols automatisch Wiederholungskennzeichen eingefügt werden,
- daß Datenflüsse mit Attributen versehen werden können,
- daß der Beschriftungszeitpunkt von Prozessen frei wählbar ist,
- daß bestimmte Symbole des DFD visuell hervorgehoben werden können,
- ob ein Editor zum Generieren von Pseudo-Code bzw. Structured English existiert und wenn ja, ob Felder für die Auflistung von Ein- und Ausgabedaten existieren und ob man logische Konstrukte auf Sequenz, Iteration u. Alternative begrenzen kann und
- ob ein Editor zum Generieren von Entscheidungstabellen bzw. Entscheidungsbäumen existiert und wenn ja,
  - welche syntaktischen Elemente dargestellt werden (Bedingungen, Bedingungsanzeiger, Aktionsanzeiger, Aktionen) können,
  - ob unterschiedliche Regeln, die zu derselben Aktion führen, zu einer Hauptregel zusammengeführt werden können und
  - ob mehrere Tabellen zu einem Verbund zusammengefaßt werden können.

Bei der Erstellung von **Structure Chart** bzw. **Modulstruktur** muß geprüft werden,

- welche Informationen (Flüsse, Interfaces etc.) dargestellt werden können,
- ob Prozesse und andere Informationen aus dem DFD übernommen werden können,
- ob eine Zuordnung zwischen Structure Chart bzw. Modulhierarchie-Komponenten und dem DFD erfolgen kann und schließlich
- ob eine Code-Generierung auf der Ebene der Modulhierarchie erfolgt.

### 3.1.1.2. Objektorientierte Methoden

Die CASE-Tools, die für die Evaluierungsphase der CASE-Studie ausgewählt wurden, unterstützen folgende objektorientierte Methoden:

1. OMT von Rumbaugh
2. OOA/OOD nach Coad/Yourdan
3. OOSA nach Shlaer/Mellor

Alle drei Methoden bieten eine Notation zur Beschreibung einer Klassenstruktur an. Für Klassen, deren Objekte ein dynamisches Verhalten besitzen, wird ein Zustandsübergangsdiagramm angelegt.

Da das Ziel der Evaluierungsphase nicht das Gegenüberstellen der Methoden, sondern das Testen der Tools bzgl. Funktionalität und Handhabbarkeit ist, ist das objektorientierte Modell des Prüfbeispiels auf ein Klassenstrukturdiagramm und ein Zustandsübergangsdiagramm beschränkt worden. Die weiteren Beschreibungsmittel der Methoden werden auf syntaktische Vollständigkeit und Ausgestaltung der Beziehungen mit den oben beschriebenen Diagrammtechniken geprüft.

#### Klassenstrukturdiagramme

Das zu prüfende CASE-Tool sollte das Klassenstrukturdiagramm des Prüfbeispiels mit seinen Attributen und Methoden darstellen können. Darüber hinaus sollten die folgenden Punkte untersucht werden:

- Werden beim Anlegen von Subklassen die Attribute und Methoden der Superklasse automatisch übernommen und angezeigt ?
- Läßt sich die Anzeige der Attribute und Methoden unterdrücken, um nur die Beziehungen zwischen den Klassen darzustellen ?
- Bleibt das vom Entwickler erstellte Layout des Diagramms erhalten, wenn Methoden und Attribute hinzugefügt werden ?
- Können bei der Attributierung Datentypen (z. B. Name-Typ), Wertebereiche und Defaultwerte definiert werden ?
- Können beim Definieren von Methoden die Datentypen der Aufrufparameter deklariert werden ?

## Zustandsübergangsdiagramme

Das zu prüfende CASE-Tool sollte ein beispielhaft für die Klasse Schaden entworfenes Zustandsübergangsdiagramm und - soweit möglich - die Verfeinerung des Zustands "bearbeite Schaden" modellieren können.

## Integration der Teilmodelle

In bezug auf die Integration der Teilmodelle sind die folgenden Fragen zu beantworten:

- Kann einer Klasse ein Zustandsübergangsdiagramm zugeordnet werden?
- Kann einem State eine Methode der zugehörigen Klasse zugeordnet werden?
- Kann einem Event eine Methode der zugehörigen Klasse zugeordnet werden?
- Kann einer Aktion eine Methode der zugehörigen Klasse zugeordnet werden?
- Kann ein Event an andere Klassen, die im Klassenstrukturdiagramm definiert sind, geschickt werden?
- Werden Zustandsübergangsdiagramme von Superklassen automatisch in Subklassen übernommen?
- Kann dieser Mechanismus abgeschaltet werden?

### 3.1.2. Realisierungsaktivitäten

Im Rahmen der Realisierung sollen hier nur Aktivitäten wie Code-Generierung und die Erstellung der Benutzerdialoge betrachtet werden. Daher interessiert,

- ob automatisiert einwandfreier SQL-Code generiert werden kann,
- ob der SQL-Code auf das Zielsystem zugeschnitten ist bzw., ob sich zielsystemspezifische Einstellungen vornehmen und damit Funktionalität/Performance ausschöpfen lassen,

- ob die gefordertern Bildschirmmasken und -dialoge (sowohl zeichen- als auch grafikorientiert) erstellt werden können,
- ob bei der Animation von Dialogen
  - Screens verknüpft werden können,
  - Menühierarchien animiert werden können,
  - Eingabepfungen (Plausibilitätskontrollen etc.) erfolgen,
  - Benutzereingaben simuliert werden können,
- ob Informationen aus dem Data Dictionary übernommen werden,
- ob Verknüpfungen zu Tabellen/Feldern etc. definiert werden können,
- ob Konsistenzchecks zwischen dem Data Dictionary und dem Dialog-Design möglich sind (z. B. Datenfeldlänge),
- was aus dem Dialog generiert werden kann,
- das Aussehen der generierten Bildschirmmasken und
- die Gestaltung der VIEWS.

### 3.1.3. Dokumentation

Unter der Aktivität Dokumentation soll hier die Darstellung der Entwicklungsergebnisse oder Teile davon verstanden werden. Hierfür sollte das CASE-Tool neben den Diagrammen der Daten- und Funktionsmodellierung (ERD, DFD etc.) auch standardisierte und benutzerindividuell gestaltbare Berichte anbieten. Diese Berichte sollten die Informationsbedürfnisse der Entwickler, aber auch der der Auftraggeber hinsichtlich Inhalt, Struktur, Vollständigkeit und Korrektheit berücksichtigen.

Hierbei ist es von besonderem Nutzen, wenn die Diagramme und Reports komfortabel in einem Textverarbeitungssystem weiterbearbeitet werden können. Bei der Einbindung der Dokumentation in CASE-Tool fremde Textverarbeitungssysteme ist z. B. im PC-Bereich unter MS Windows die Unterstützung von DDE/OLE besonders hilfreich.

Darüber hinaus können die Anforderungen an die Dokumentation hinsichtlich Inhalt, Struktur und Vollständigkeit durch die Bereitstellung von sog. Dokumentskeletten (z. B. vorgegebene Gliederung mit Checkfragen im Anforderungskatalog) besonders gut erfüllt werden.

### **3.2. Funktionsumfang im Hinblick auf Methoden bzw. Vorgehensmodelle**

#### **3.2.1. Methodenintegration**

Die Integration der Methoden ist ein wesentliches Leistungsmerkmal eines CASE-Tools. Daher ist festzustellen, ob

- bei der Arbeit (z. B. Attributierung) im Modell A (z. B. Prozeßmodell) auf Ergebnisse (z. B. Attribute) aus Modell B (z. B. Datenmodell) zurückgegriffen werden kann,
- verschiedene Designobjekte unterschiedlicher Modelle miteinander verknüpft werden können, z. B. Datenspeicher mit Entities und dabei
  - Attribute übernommen werden,
  - die Konsistenz der Attribute auch bei einer Änderung/Löschung gewährleistet ist,
  - dies durch Konsistenzchecks abgesichert wird,
- verschiedene Modelle gleichzeitig auf dem Bildschirm dargestellt werden können (z. B. DFD und ERM),
- bei der Löschung eines Symbols alle Verbindungen zu anderen Objekten im Diagramm einerseits und im Data Dictionary andererseits berücksichtigt werden.

#### **3.2.2. Übergreifende Methoden/Vorgehensmodelle**

Sofern angeboten, sollten weitere als das bei der Evaluierung verwendete Vorgehensmodell z. B. Merise, V-Modell erwähnt werden.

## **4. Schlußbemerkung**

Interessant ist am Ende der Evaluierung zum einen die Gegenüberstellung der wesentlichen Stärken und Schwächen des CASE-Tools. Zum anderen kann dann in einem Ausblick auf die zukünftigen vom Hersteller oder Anbieter angekündigten Weiterentwicklungen in naher Zukunft festgestellt werden, ob die ermittelten Schwächen nur von vorübergehender Natur sind.

### **4.1. Leistungsprofil**

Neben den wichtigsten Stärken und Schwächen ist auch eine kurze Empfehlung für den möglichen Einsatzbereich des CASE-Tools (z. B. Klein-/Großanwender, UNIX/DOS/OS/2-Anwender, C++/COBOL-Programmierer, Branche etc.) von Interesse.

### **4.2. Ausblick**

Der Ausblick sollte nur die Weiterentwicklungen der nächsten 6 bis 12 Monate berücksichtigen, da darüber hinausgehende Produktankündigungen des Herstellers eher spekulativen Charakter haben.