

# **Ein Vorgehensmodell zur Software-Evaluation**

Andreas Winter  
Uwe Dumslaff  
Jürgen Ebert  
Markus Mertesacker

Universität Koblenz-Landau  
Fachbereich Informatik  
Rheinau 1  
5400 Koblenz

23. April 1993

# Stichworte

Vorgehensmodell, Software-Evaluation, Software-Analyse, Auftragsabwicklung im Handwerk, qualitative Evaluation, quantitative Evaluation, Evaluation der Benutzerschnittstelle.

# Zusammenfassung

Es wird ein Vorgehensmodell zur Evaluation von Software vorgestellt. Die Software-Evaluation wird hierzu in einen vom konkreten Untersuchungsziel abhängigen und in einen unabhängigen Untersuchungsteil unterschieden.

Im abhängigen Teil der Software-Evaluation wird die Funktionalität der Software untersucht. Dieser Untersuchung ist ein Modell des Anwendungsbereichs der Software als Bewertungsmaßstab zugrunde zu legen, aus dem Kriterien zur Software-Evaluation abgeleitet werden können. Allgemeine Aspekte, die nicht vom direkten Einsatzbereich der Software abhängen, werden im unabhängigen Teil der Evaluation betrachtet.

Das Vorgehensmodell wird am Beispiel der Evaluation von Software zur Unterstützung der Auftragsabwicklung im Handwerk erläutert.

# Inhaltsübersicht

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Vorgehensmodell zur Software-Evaluation</b>	<b>1</b>
<b>3</b>	<b>Modellabhängige Software-Evaluation</b>	<b>3</b>
3.1	Grundlagen der modellabhängigen Software-Evaluation . . . . .	4
3.2	Quantitative Evaluation . . . . .	6
3.3	Qualitative Evaluation . . . . .	8
<b>4</b>	<b>Modellunabhängige Software-Evaluation</b>	<b>10</b>
4.1	Evaluation der Benutzerschnittstelle . . . . .	11
4.2	Evaluation der Rahmenbedingungen . . . . .	14
<b>5</b>	<b>Zusammenfassung</b>	<b>15</b>
	<b>Literatur</b>	<b>16</b>

# 1 Einleitung

*Software-Evaluation* ist die Analyse und Bewertung von Software-Systemen. Als Anlaß einer Software-Evaluation kann die Unterstützung bei der Auswahl von Software für einen umrissenen Anwendungsbereich genannt werden. Hier interessiert, welches *Leistungsspektrum* durch die betrachteten Software-Produkte geboten wird und es gilt das *Zusammenwirken einzelner Software-Bestandteile* zu beurteilen. Ergänzt wird die Software-Evaluation um eine Betrachtung der *Benutzerschnittstelle* und um die Ermittlung von *Rahmenbedingungen* allgemeiner Art, die beim Einsatz der Software z.B. auf der Hardware- und Systemseite zu berücksichtigen sind.

Das im folgenden dargestellte *Vorgehensmodell zur Software-Evaluation* wurde im Rahmen des Projekts „*Konzeptualisierung von Qualifizierungsmaßnahmen für die Bewirkung von Planungs- und Entscheidungskompetenzen im Bereich der Auftragsabwicklung in energie- und fertigungstechnischen Handwerken*“ entwickelt und auf Branchenlösungen für diese Handwerksbereiche angewandt<sup>1</sup>. Durch den Einsatz von Branchenlösungen kann Handwerksbetrieben ein Unterstützungsmittel geboten werden, das ein planvoll-systematisches Vorgehen bei der kaufmännischen Betriebsführung und deren Zusammenspiel mit der eigentlichen handwerklichen Leistungserstellung ermöglicht [4, 7]. Diese möglichen Unterstützungsmittel galt es zu analysieren und zu beschreiben.

Ziel der Software-Evaluation war es, zu jedem der analysierten Software-Produkte eine Beschreibung zu erstellen, aus der zum einen die Eignung für Qualifikationsmaßnahmen, d.h. zum Einsatz in Aus- und Weiterbildungskursen, und zum anderen der Umfang und die Qualität der Unterstützung der Auftragsabwicklung durch die jeweilige Software hervorgeht.

## 2 Vorgehensmodell zur Software-Evaluation

Charakteristisch für das im folgenden erläuterte Vorgehensmodell zur Software-Evaluation [5] ist die Aufgliederung in eine *modellabhängige* und in eine *modellunabhängige Untersuchung* der Software (vgl. Abbildung 1).

### Modellabhängige Software-Evaluation

Grundlage der *modellabhängigen Software-Evaluation* ist ein Modell des Wirklichkeitsausschnitts, in dem die betrachtete Software eingesetzt werden soll. Entlang eines vom Untersuchungsziel abhängigen Modells kann die Bestimmung des Leistungsspektrums, die Bewertung der durch die Software bereitgestellten Unterstützungsmittel zur Aufgabenerfüllung sowie die Ermittlung von Zusammenhängen zwischen den verschiedenen Software-Bestandteilen erfolgen. Das analysezielbezogene Modell legt einen Vergleichsmaßstab fest, gegen den die zu evaluierenden Software-Produkte untersucht werden.

---

1 Das zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministers für Forschung und Technologie unter der Fördernummer 01 HH 491/5 gefördert. Die Verantwortung für den Inhalt dieses Berichts liegt bei den ausgewiesenen Autoren.

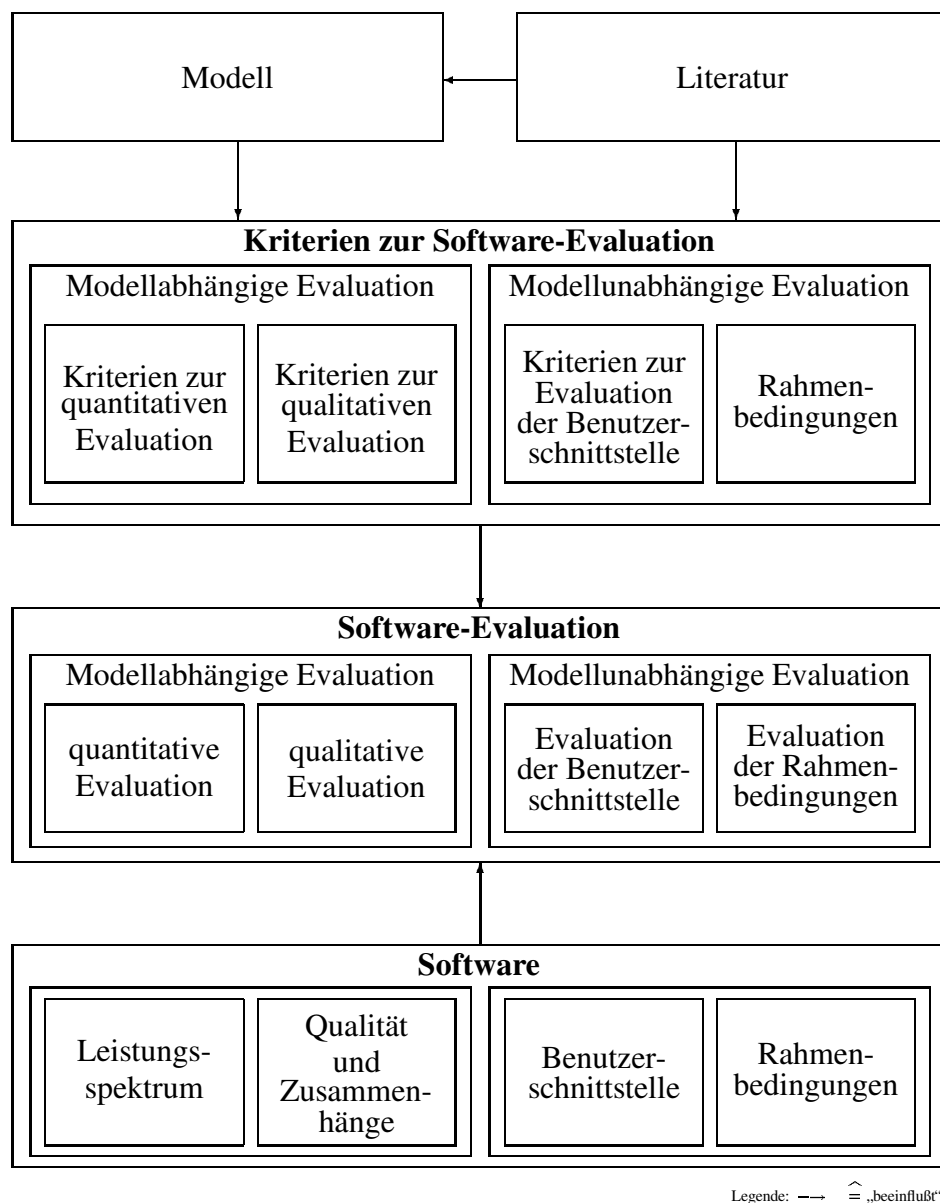


Abbildung 1: Vorgehensmodell zur Software-Evaluation

Die modellabhängige Untersuchung untergliedert sich in eine *quantitative Evaluation*, bei der die Bestimmung des Leistungsumfangs sowie die Aufdeckung der durch die Software gebotenen Unterstützungsmittel im Vordergrund steht, und in eine *qualitative Evaluation*, bei der die Art und Weise der Software-Unterstützung untersucht wird.

### Modellunabhängige Software-Evaluation

Den zweiten Teil des Vorgehensmodells bildet die *modellunabhängige Software-Evaluation*. Hierzu ist kein spezielles, vom Untersuchungsziel abhängiges Modell zu entwickeln, es kann stattdessen auf allgemeine Modelle zurückgegriffen werden, die auch als Vergleichsmaßstab für Software-Betrachtungen in anderen Kontexten einzusetzen sind.

Die modellunabhängige Untersuchung wird in eine Analyse und Bewertung der *Benutzer-*

*schnittstelle* und in eine Betrachtung der *Voraussetzungen und Rahmenbedingungen* beim Software-Einsatz unterteilt. Die Evaluation der Benutzerschnittstelle erfolgt anhand der in der Literatur zur Software-Ergonomie vorgestellten Kriterien (vgl. z.B. [3, 8, 9, 15]) und Evaluationsverfahren (vgl. z.B. [12]), wobei die dort aufgeführten aufgabenbezogenen Anteile hier ausgeklammert und innerhalb der modellabhängigen Evaluation untersucht werden. Bei der Bestimmung der Voraussetzungen und Rahmenbedingungen des Software-Einsatzes wird wieder auf allgemeine Kriterien zu Hard- und Software-Gegebenheiten, Ausstattung der Software mit Handbüchern etc. zurückgegriffen.

### **Durchführung der Software-Evaluation**

Mit Hilfe der hieraus abgeleiteten Kriterien kann eine Untersuchung der Software bezüglich

1. des *Leistungsspektrums*,
2. der *Qualität und Zusammenhänge* der Software-Bestandteile,
3. der *Benutzerschnittstelle* und
4. der *Rahmenbedingungen*

durchgeführt werden. Dies erfolgt für den modellabhängigen Teil der Software-Untersuchung in der *qualitativen* und in der *quantitativen Evaluation* und für den modellunabhängigen Teil in der *Evaluation der Benutzerschnittstelle* und in der *Evaluation der Rahmenbedingungen*.

Da die Untersuchung der Softwareprodukte auf Kriterien basiert, die aus einem analysezielabhängigen Modell bzw. aus der Literatur abgeleitet werden, und somit unabhängig von der untersuchten Software sind, wird bei der Betrachtung unterschiedlicher Software-Produkte ein einheitlicher Bewertungsmaßstab zugrundegelegt. Die hiermit verbundene Offenlegung der Bewertungskriterien erlaubt es weiterhin, die Untersuchung auch später um andere Software-Produkte für ein vergleichbares Anwendungsgebiet zu erweitern.

## **3 Modellabhängige Software-Evaluation**

Bei der modellabhängigen Software-Evaluation werden die funktionalen Aspekte der Software betrachtet. Untersucht wird das Leistungsspektrum der Software, d.h. die Funktionen, die dem Benutzer zur Aufgabenerledigung durch die Software verfügbar gemacht werden. Es gilt festzustellen, welche Aufgaben durch die Software unterstützt und wie die einzelnen Software-Mittel kombiniert werden können.

Ausgangspunkt dieser Software-Untersuchungen ist ein möglichst formales Modell des Anwendungsbereichs der Software. Aus diesem werden Kriterien zur Untersuchung des Leistungsumfangs der Software, zur Bewertung der Möglichkeiten der Software-Unterstützung und zur Betrachtung der Zusammenhänge einzelner Software-Bestandteile abgeleitet. Als Modellierungsmittel bieten sich die aus dem Requirements Engineering bekannten Modellierungsparadigmen [6] für Entwurfsbeschreibungssprachen an. Je nach Analysekontext kann eine Modellierung nach dem Entity-Relationship-Paradigma, dem Datenfluß-Paradigma und/oder dem Kontrollfluß- und Zustandsübergangs-Paradigma erfolgen.

Im konkreten Projektkontext war ein Modell der *Auftragsabwicklung im energie- und fertigungstechnischen Handwerk*<sup>2</sup> zu erstellen. Dieses Modell wird im folgenden als Beispiel kurz erläutert.

### 3.1 Grundlagen der modellabhängigen Software-Evaluation

Zur Evaluation von Software hinsichtlich ihrer Einbindung in Aus- und Fortbildungsmaßnahmen für Handwerker im Bereich der Auftragsabwicklung wird ein Modell benötigt, durch das die Tätigkeiten innerhalb der Auftragsabwicklung beschrieben werden können. Um festzustellen, wie durchgängig der Handwerker bei der Auftragsabwicklung durch die Software unterstützt wird, sind im Modell auch die Datenverbindungen zwischen diesen Aktivitäten abzubilden. Zur Modellierung bietet sich folglich eine Entwurfsbeschreibungssprache nach dem Datenfluß-Paradigma an.

Die Modellierung erfolgte im vorliegenden Projekt durch SADT-Aktivitätendiagramme (vgl. [1, S. 111-133] [13]). Hierbei wird zwischen den Daten unterschieden, die den Informationsfluß zwischen einzelnen Aktivitäten beschreiben, und den Daten, die — unabhängig vom konkret betrachteten Auftrag — im Handwerksbetrieb z.B. über Personal- oder Maschinenauslastung vorliegen. Letztere werden als Kontrolldaten modelliert<sup>3</sup>.

Das SADT-Modell der Auftragsabwicklung im Handwerk wurde nach intensiven Gesprächen mit Fachexperten der beiden behandelten Handwerksbereiche entwickelt. Anschließend erfolgte eine Validierung in mehreren Betriebsbefragungen. Auch hierbei erwies sich die Darstellung des Modells mittels SADT-Aktivitätendiagrammen als geeignetes Kommunikationsmittel zwischen Handwerkern und Informatikern.

Ergänzt wurde die Modellierung mittels SADT-Aktivitäten-Diagrammen (vgl. das Diagramm in Abbildung 2) um eine *kommentierte Aktivitätenliste* (siehe Abbildung 3), die zu jeder nicht weiter verfeinerten Aktivität Software-Unterstützungsmöglichkeiten beschreibt.

#### Exkurs: Modell der Auftragsabwicklung im Handwerk

Die Auftragsabwicklung wird in der Modellierung in die sieben Hauptaktivitäten

1. *Auftrag beschaffen,*
2. *Angebot erarbeiten,*
3. *Vertrag aushandeln,*
4. *Arbeit vorbereiten,*
5. *Auftrag durchführen,*
6. *Auftrag abrechnen und*
7. *Auftrag nachbereiten.*

zerlegt. Diese sind durch Datenflüsse miteinander verbunden. Das zugehörige SADT-Aktivitätendiagramm findet sich in Abbildung 2. Diese Hauptaktivitäten werden in weiteren Diagrammen näher beschrieben (siehe hierzu [5, S. 238–269]).

---

2 Unter dem Begriff „energietechnisches Handwerk“ wurde der Handwerksbereich zusammengefaßt, der sich mit dem Sanitär-, Heizungs- und Klimabereich beschäftigt. „Fertigungstechnisches Handwerk“ umfaßt die Gewerke, die sich mit der Metall- oder Holzverarbeitung befassen.

3 In textuellen Erläuterungen zu den Diagrammen sind die Bezeichner der Aktivitäten *kursiv* und die der Datenflüsse *sans serif* gesetzt.

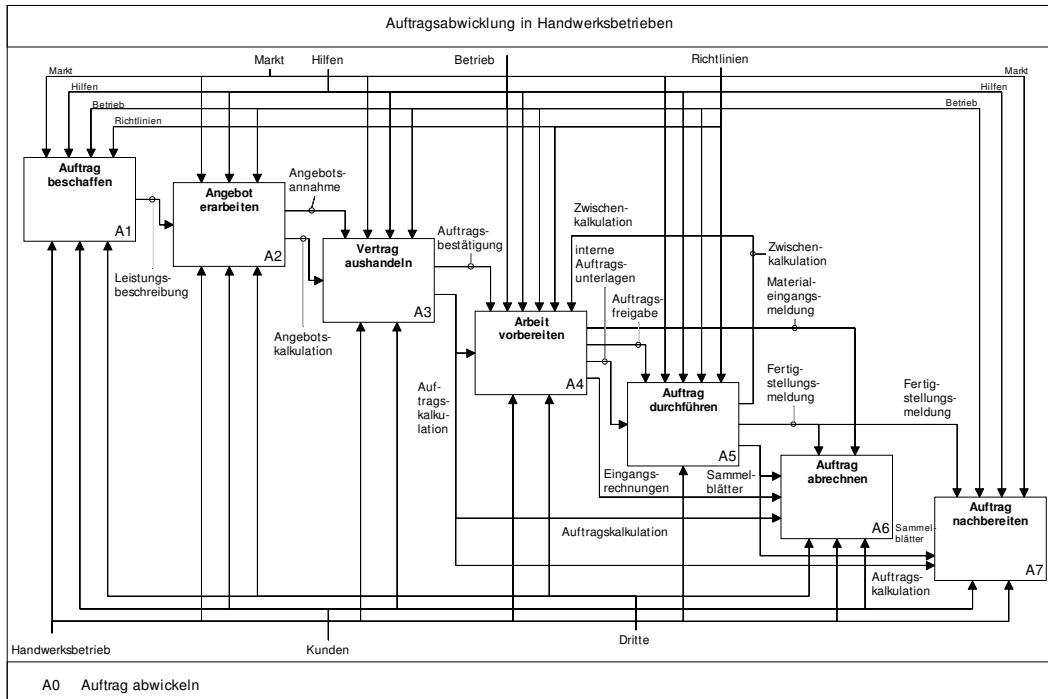


Abbildung 2: SADT-Aktivitätendiagramm

In der Aktivität *Auftrag beschaffen* wird die Kontaktaufnahme zu Kunden und die Beschaffung der Informationen modelliert, die Ausgangspunkt der Erarbeitung eines Angebots sind. Hierunter fallen auch zu erstellende technische Berechnungen.

Mit Vorliegen der so erhaltenen Leistungsbeschreibung kann die Erstellung des Angebots beginnen. Ziel der Aktivität *Angebot erarbeiten* ist die Ermittlung der Angebotskalkulation, die unter anderem eine vorläufige Planung des möglichen Auftrags, Konstruktionen sowie die Preise für die Erbringung der handwerklichen Leistung enthält. Hierin ist weiter die (vorläufige) Annahme des Angebots (Angebotsannahme) durch den Auftraggeber modelliert.

In der Aktivität *Vertrag aushandeln* wird die Angebotskalkulation zur Auftragskalkulation erweitert. Hier werden gegebenenfalls noch einmal Änderungen im Auftragsumfang oder in der Gestaltung der Auftragsausführung vorgenommen. Die Auftragskalkulation ist Grundlage für die Arbeitsvorbereitung, für die Abrechnung und für die Nachbereitung des Auftrags.

Die Vorbereitung der zu erbringenden handwerklichen Leistung in der Aktivität *Arbeit vorbereiten* kann erfolgen, sobald die Auftragsbestätigung erstellt ist. Es geht darum, interne Auftragsunterlagen bereitzustellen, die eine Konkretisierung der bisherigen Planungen und eine Beschreibung zur Durchführung der Arbeiten inkl. Auftragsbegleitpapiere enthalten. Dort erfolgt auch die Beschaffung der benötigten Materialien, so daß von hier aus Eingangsberechnungen an die Auftragsabrechnung weitergegeben werden.

Nach Erteilung der Auftragsfreigabe kann dann mit der Durchführung des Auftrags begonnen werden. Neben der handwerklichen Leistungserstellung liefert die Aktivität *Auftrag durchführen* Dokumente, aus denen der tatsächlich benötigte Material- und Arbeitszeitbedarf (Sammelblätter) hervorgeht. Diese Informationen dienen bei

der Nachbereitung des Auftrags als Grundlage zur Nachkalkulation. Ergebnisse der bisher aufgelaufenen Istkosten (Zwischenkalkulation) werden an die Arbeitsvorbereitung weitergegeben. Durch Anpassung der internen Auftragsunterlagen kann dann auf Abweichungen zwischen geplantem und tatsächlichem Auftragsfortgang reagiert werden.

Die Abrechnung des Auftrags, sowohl gegenüber den Auftraggebern wie auch gegenüber den Lieferanten des Handwerksbetriebs, erfolgt in der Aktivität *Auftrag abrechnen*.

Abgeschlossen wird die hier modellierte Auftragsabwicklung im Handwerksbetrieb durch die Aktivität *Auftrag nachbereiten*. Anhand der Auftragskalkulation und den Sammelblättern können geplante und tatsächlich angefallene Kosten des Auftrags einander gegenübergestellt werden.

Weiter fließen in die Aktivitäten noch Daten ein, die nicht auftragsbezogene Informationen über das Umfeld der Abwicklung eines Auftrags enthalten. Diese, im Modell als Kontrolldaten aufgeführten Datenflüsse, bilden Informationen über den Markt ab, an dem der Handwerksbetrieb aktiv ist und beschreiben den Betrieb selbst, z.B. durch Daten über das Personal oder über die aktuelle Betriebsplanung. Hier werden auch allgemeine Hilfsmittel zur Auftragsabwicklung (Hilfen) sowie Vorschriften und Normen (Richtlinien) notiert, die zu berücksichtigen sind.

Zu jeder Aktivität wird noch vermerkt, durch wen sie ausgeführt wird. Hierzu wurden die „Mechanismen“ Handwerksbetrieb, Kunden und Dritte eingeführt, die jeweils weiter zerlegt wurden (vgl. [5, S. 242]).

Das hier vorgestellte Modell zur Auftragsabwicklung in Handwerksbetrieben beschreibt die Auftragsabwicklung bezogen auf einen einzelnen Auftrag. Eine Verzahnung mit anderen Aufträgen wird indirekt durch die Kontrolldatenflüsse abgebildet, so daß das Modell einen klar umrissenen Ausschnitt des Handwerksbetriebs darstellt. Für das der Software-Analyse zugrundeliegende Modell wurden jedoch alle Aktivitäten ermittelt, die während der Abwicklung eines Auftrags durchlaufen werden können.

**Ende des Exkurses**

## **3.2 Quantitative Evaluation**

Ziel der *quantitativen Evaluation* ist die Ermittlung des Leistungsspektrums der Software. Zu fragen ist, welche Unterstützungsmöglichkeiten durch die betrachtete Software bereitgestellt werden.

Grundlage hierzu ist für das Anwendungsbeispiel der Auftragsabwicklung im Handwerk das zuvor in Kapitel 3.1 beschriebene Datenflußmodell. Durch die Verfeinerung der einzelnen Aktivitäten des Modells erhält man eine hierarchische Gliederung der Tätigkeiten. Einen Ausschnitt der Aktivitäten-Hierarchie für die beiden ersten Hauptaktivitäten aus dem Anwendungsbeispiel findet sich in Abbildung 3 (linke Spalte).

Zur Überprüfung bzw. Bestimmung des Leistungsspektrums der Software reicht es nun, diejenigen Tätigkeiten zu betrachten, die im Modell nicht weiter verfeinert werden. Diese Terminalaktivitäten sind in Abbildung 3 **fett** gesetzt. Im Datenflußmodell



Aktivitäten	Möglichkeiten der Softwareunterstützung
A1 Auftrag beschaffen A11 Kontakt mit dem Absatzmarkt aufnehmen <b>A111 Kontakt zu Anfragern aufnehmen</b> <b>A112 Kontakt zu Ausschreibern aufnehmen</b> A12 Leistungsbeschreibung erstellen/beschaffen A121 Anfrage bearbeiten A1211 Auftrag spezifizieren <b>A12111 Kunden beraten</b> <b>A12112 Baustelle begehen</b> <b>A1212 Angebotskonstruktion erstellen</b> <b>A122 Leistungsverzeichnis beschaffen</b>	Textverarbeitung, Adreßverwaltung, Kunden-/Wartungsdaten anzeigen Textverarbeitung, Adreßverwaltung, Kundendaten anzeigen  Kunden-/Wartungsdaten anzeigen, Beratungshilfen bereitstellen Aufmaßerstellung direkt auf der Baustelle Zugriff auf Vergleichsdaten, technische Berechnungen, CAD Einlesen von Leistungsverzeichnissen über Diskette
A2 Angebot erarbeiten A21 Angebotskalkulation vorbereiten A211 Stücklisten/Massenberechnungen erstellen <b>A2111 Stücklisten erstellen</b> <b>A2112 Materialbedarf ermitteln</b> A212 Angebotsarbeitspläne erstellen <b>A2121 Angebotsarbeitsvorgänge bestimmen</b> <b>A2122 Angebotsarbeitsvorgangfolgen bestimmen</b> <b>A2123 Angebotsarbeitsvorgangszeiten bestimmen</b> A213 Termine und Kapazitäten abschätzen <b>A2131 Termine abschätzen</b> <b>A2132 Kapazitäten abschätzen</b> Anlagenbetriebliche Kooperation	Zugriff auf Vergleichsaufträge, Übernahme aus Leistungsverzeichnis Materialbedarf automatisch aus Stücklisten ableiten  Zugriff auf Vergleichsaufträge, Kalkulationshilfen bereitstellen Zugriff auf Vergleichsaufträge, Kalkulationshilfen bereitstellen Zugriff auf Vergleichsaufträge, Kalkulationshilfen bereitstellen

Abbildung 3: kommentierte Aktivitätenliste (Ausschnitt)

der Auftragsabwicklung im Handwerk wurden hierzu den Terminalaktivitäten Software-Unterstützungsmöglichkeiten zugeordnet. So kann beispielsweise die Unterstützung der Terminalaktivität *Kontakt zu Anfragern aufnehmen* durch eine Textverarbeitung und eine Adressenverwaltung unterstützt werden. Anzeigen der Kunden-/Wartungsdaten erlaubt z.B. bei telephonischen Reparaturanfragen erste Rückschlüsse auf eventuelle Fehlerursachen. Auf diese Datengrundlage könnte auch zur Vorbereitung gezielter Werbemaßnahmen zurückgegriffen werden.

### Analyse und Ergebnisdokumentation

Anhand dieser Aktivitätenliste erfolgt die quantitative Evaluation der Handwerks-Software. Für jede Terminalaktivität des Modells wird überprüft, ob durch die Software die erwarteten Unterstützungsmöglichkeiten geboten werden.

Die Dokumentation der quantitativen Evaluation erfolgt über Tabellen (siehe Abbildung 4), in denen für jede Software markiert wird, inwieweit eine Unterstützung der Terminalaktivitäten durch die Software erfolgen kann. Können die erwarteten Unterstützungsmöglichkeiten in der Software gefunden werden, so wird die Terminalaktivität als durch die Software unterstützbar notiert; wird für die betrachtete Aktivität nur eine partielle Unterstützung gefunden, wird dieses ebenfalls markiert.

Innerhalb der quantitativen Evaluation werden weitgehend objektive Daten, nämlich das Vorhandensein, das teilweise Vorhandensein oder das Fehlen von zuvor dokumentierten Unterstützungsmöglichkeiten erhoben und beschrieben. Diese Erhebung kann folglich entlang des Modells für jedes einzelne Produkt unabhängig von den anderen zu analysierenden Produkten erfolgen.

Quantitative Evaluation	Legende:		
	● gute Unterstützung	○ partielle Unterstützung	— keine Unterstützung
Aktivitäten	Produkt A	Produkt B	Produkt C
A1 Auftrag beschaffen			
A11 Kontakt mit dem Absatzmarkt aufnehmen			
<b>A111 Kontakt zu Anfragern aufnehmen</b>	●	○	○
<b>A112 Kontakt zu Ausschreibern aufnehmen</b>	●	○	○
A12 Leistungsbeschreibung erstellen/beschaffen			
...			
A2 Angebot erarbeiten			
A21 Angebotskalkulation vorbereiten			
A211 Stücklisten/Massenberechnungen erstellen			
<b>A2111 Stücklisten erstellen</b>	●	●	●
<b>A2112 Materialbedarf ermitteln</b>	●	●	●
...			
A213 Termine und Kapazitäten abschätzen			
<b>A2131 Termine abschätzen</b>	○	●	—
<b>A2132 Kapazitäten abschätzen</b>	○	●	—
A214 Zwischenbetriebliche Kooperation planen			
<b>A2141 Anfrage nach Fremdantrag spezifizieren</b>	—	—	○
<b>A2142 Fremdanträge einholen</b>	—	—	○

Abbildung 4: Ergebnis der quantitativen Evaluation (Ausschnitt)

### 3.3 Qualitative Evaluation

Ebenso wie die quantitative Evaluation erfolgt auch die *qualitative Evaluation* bezogen auf das vom Untersuchungsziel abhängige Modell. Ziel dieses Evaluationsteils ist eine weitergehende Untersuchung der in der quantitativen Evaluation betrachteten Aspekte. Es sind auch die Zusammenhänge zwischen den einzelnen Softwarebestandteilen herauszuarbeiten.

Liegt der Software-Evaluation — wie im vorliegenden Anwendungsbeispiel — eine Modellierung nach dem Datenfluß-Paradigma vor, so kann die Untersuchung entlang den Datenflüssen erfolgen, die zwischen den einzelnen Aktivitäten verlaufen. Anhand der in einen Aktivitätenknoten einlaufenden Datenflüsse ist festgelegt, welche Daten zur Bearbeitung der modellierten Aktivität benötigt werden. In der qualitativen Evaluation ist nun zu überprüfen, welche der erwarteten Daten durch die Software zur Bearbeitung dieser Aktivität zur Verfügung gestellt oder erfragt werden.

#### Vorhandene Daten:

Für im System *bereits vorhandene Daten* ist festzustellen, wo sie erfaßt oder errechnet werden, und wie sie an die sie benötigenden Programmteile weitergegeben werden. Der Durchgängigkeit der Softwareunterstützung ist es z.B. nicht dienlich, wenn Ergebnisse bereits durchgeführter technischer Berechnungen noch einmal von Hand erfaßt werden müssen. Die Ermittlung der Datenquellen (und -senken) erlaubt die Auffindung solcher Medienbrüche. Das Augenmerk kann hierbei auf die auftragsbezogenen Daten gelenkt werden. Diese sind im SADT-Modell als Datenflüsse modelliert.

Kontrollflüsse beschreiben in der zur Evaluation zugrundegelegten Modellierung solche Daten, die im Handwerksbetrieb allgemein und nicht bezogen auf einen

einzelnen Auftrag vorliegen. Hier finden sich Personaldaten, Einkaufs- und Verkaufspreise, Informationen über Personalauslastungen und ähnliches. Für jede nicht weiter zerlegte Tätigkeit des Modells ist zu überprüfen, ob die erwarteten Daten zur Bearbeitung dieser Aktivität durch das Software-System bereitgestellt werden können. So können durch die Software nicht bereitgestellte, aber aufgrund des Modells für notwendig erachtete Daten ermittelt werden.

### **Unterstützung der Datenerhebung:**

Zur Bearbeitung einer Aktivität sind auch Daten zu erfassen. Die Software ist somit hinsichtlich der zu erhebenden Daten und der *Unterstützung der Datenerhebung* zu untersuchen. Hier ist danach zu fragen, ob der Benutzer z.B. durch Bereitstellen von teilweise sogar vorausgefüllten Formblättern oder Checklisten bei der Datenerhebung auf der Baustelle unterstützt wird. Als weitere Daten-Eingabemöglichkeit sind Datenschnittstellen z.B. zum Einlesen von Leistungsverzeichnissen zu berücksichtigen.

### **Kontrolle der Benutzereingaben:**

Weitere Untersuchungsaspekte sind die von der Software bereitgestellten Mechanismen zur *Kontrolle der Benutzereingaben* auf Korrektheit und Vollständigkeit. Neben der Analyse, ob überhaupt eine Überprüfung auf Plausibilität der Daten erfolgt, ist der Umgang mit fehlerhaften Eingaben zu betrachten. Es kann in Sonderfällen durchaus sinnvoll sein, mit — für das Software-System — unplausiblen Daten zu arbeiten. So kann es aus betriebspolitischen Gründen richtig sein, bei einem Auftrag bewußt einen Verlust einzukalkulieren. Dies sollte die Software zulassen.

Bei der Evaluation der Software hinsichtlich der Kontrolle von Dateneingaben durch den Benutzer ist also zu fragen, ob diese Überprüfung erfolgt und ob bewußt unplausible Eingaben zugelassen werden. Werden diese erlaubt, ist zu analysieren, ob der Benutzer auch bei weiteren Berechnungen darauf aufmerksam gemacht wird, mit solchen Daten zu arbeiten.

### **Erzeugte Daten:**

Neben den Daten, die zur Bearbeitung einer Aktivität benötigt werden, sind ebenso die Daten zu analysieren, die durch diese Aktivitäten erzeugt werden. Zu überprüfen ist hier, in welcher Form diese Ergebnisse vom Bediener weiter benutzt werden können und ob diese Daten auch noch durch den Benutzer änderbar sind.

Ähnlich wie bei der Frage nach Hilfsmitteln zur Datenerfassung, wird hier untersucht, ob und in welchem Umfang durch die Software die Erstellung verschiedenartiger Dokumente zur Benutzung und Weitergabe der Daten ermöglicht wird. Für die Software-Unterstützung der Auftragsabwicklung in Handwerksbetrieben ist z.B. nach Begleitpapieren wie Materiallisten oder Montageanweisungen zu fragen.

### **Funktionale Korrektheit:**

Ein weiterer Aspekt der qualitativen Evaluation ist die Untersuchung der funktionalen Korrektheit der betrachteten Software. Hinsichtlich der Korrektheitsüberprüfung von Software sei auf die einschlägige Literatur zum Software-Testen (vgl. z.B. [10][14]) hingewiesen.

## Analyse und Ergebnisdokumentation

Die Analyse der qualitativen Aspekte der Software erfolgt entlang eines Frageleitfadens, der aufgrund des analyseziel-abhängigen Modells erstellt wurde. Hiermit wird jede durch Software unterstützte Aktivität hinsichtlich der zuvor genannten Kriterien untersucht (vgl. die Frageleitfäden in Abbildung 5).

Abbildung 5: Frageleitfaden zur qualitativen Evaluation (Ausschnitt)

Die qualitative Evaluation der Handwerks-Software wurde anhand dieses Frageleitfadens und des Modells in der Regel von zwei Personen durchgeführt. Die Ergebnisse dieser Evaluation wurden in einer textuellen Kurzbeschreibung der Software zusammengefaßt.

Durch die Verwendung des zuvor erstellten Modells als Grundlage zur qualitativen Evaluation, liegt für die Untersuchung der Software-Produkte der gleiche — von anderen Produkten unabhängige — Bewertungsmaßstab zugrunde. Wie bei der quantitativen Evaluation kann auch hier durch die Verwendung des Frageleitfadens eine einheitliche Sichtweise auf die untersuchten Produkte gewährleistet werden.

## 4 Modellunabhängige Software-Evaluation

In der modellunabhängigen Software-Evaluation wird die Benutzerschnittstelle der Software betrachtet. Untersucht wird hier, wie sich die Software während der Benutzung dem

Anwender darstellt.

Gegenüber der modellabhängigen Evaluation ist zu diesem Teil der Software-Evaluation kein spezielles Modell des Anwendungsbereichs der Software notwendig. Stattdessen wird zur Evaluation der Benutzerschnittstelle auf bereits vorliegende, allgemeine Ansätze zur Analyse und Bewertung der Benutzbarkeit von Software zurückgegriffen.

Zur Abrundung der Software-Evaluation werden noch die Rahmenbedingungen des Software-Einsatzes erhoben. Hierunter fallen beispielsweise Fragen nach der Ausstattung des Software-Pakets hinsichtlich Dokumentation und Hilfen bei der Software-Einführung.

## 4.1 Evaluation der Benutzerschnittstelle

Zur Evaluation der Benutzerschnittstelle kann auf die in der Literatur zur Software-Ergonomie formulierten Kriterien zurückgegriffen werden. Hier sind beispielsweise die in DIN 66234, Teil 8 [3] oder in ISO 9241, Part 10 [9] normierten, und anhand von Beispielen motivierten Aspekte von Benutzerschnittstellen zu nennen:

### **Aufgabenangemessenheit:**

Die Interaktion zwischen Benutzer und Software-System wird als *aufgabenangemessen* bezeichnet, wenn der Benutzer zur Erledigung seiner Aufgaben nicht durch Eigenschaften der Benutzerschnittstelle unnötig belastet wird [3, S. 2].

So sollte sich der Benutzer nicht um die Art und Weise der Datenspeicherung kümmern müssen. Auch sollten Menü-Folgen und Bildschirm-Formulare an die Arbeitsschritte des Benutzers angepaßt sein.

### **Erwartungskonformität:**

Die Reaktionen des Software-Systems können als *erwartungskonform* bezeichnet werden, wenn sie den Erwartungen des Benutzers aus Benutzerschulung und bisherigem Arbeitsablauf, aber auch aus Erfahrungen mit anderen Software-Produkten entspricht [3, S. 4].

So sollte z.B. die Darstellung und Positionierung der Fehlermeldungen einheitlich erfolgen.

Zur Erwartungskonformität gehört auch, daß die Software möglichst naheliegende Reaktionen zeigt (*Überraschungsarmut*). Beim Navigieren in Menüs sollte beispielsweise ein „Rücksprung“ auf der nächsthöheren Menü-Ebene landen, und nicht über mehrere Menü-Ebenen gehen.

### **Fehlerrobustheit:**

Ein Software-System reagiert *fehlerrobust* auf (erkennbare) fehlerhafte Benutzereingaben, wenn das angestrebte Arbeitsergebnis dennoch mit geringem oder ohne Korrekturaufwand erreicht werden kann [3, S. 5].

Von einem fehlerrobusten Software-Produkt wird beispielsweise gefordert, daß es durch Falscheingaben des Benutzers nicht in einen undefinierten Systemzustand

gerät, oder es gar zu einem Systemabbruch kommt. Ebenso sollten bei Falschein-  
gaben Korrekturmöglichkeiten aufgezeigt werden, evtl. können eindeutig korri-  
gierbare Fehler — unter Information des Benutzers — durch das Software-System  
optional berichtet werden.

### **Individualisierbarkeit:**

*Individualisierbarkeit* beschreibt die Anpaßbarkeit des Software-Systems an den  
Aufgabenbereich und die Erfahrungen des Benutzers [9, S. 18].

Beim Anzeigen von Hilfen sollte der Kenntnisstand des Benutzers berücksich-  
tigt werden können. So könnten für unterschiedliche Benutzergruppen auch die  
Menge und Art dargestellter Informationen verschieden sein. Es sollte dem Benut-  
zer möglich sein, in seinem Arbeitsablauf wiederkehrende Eingaben oder Menü-  
Aufrufe zusammenzufassen und z.B. durch ein Makro aufzurufen.

Die Individualisierbarkeit umfaßt hier noch die *Steuerbarkeit* des Software-Systems.  
Hierunter versteht man, daß der Benutzer Einfluß auf die Geschwindigkeit des Pro-  
grammablaufs und auf die Auswahl von Arbeitsmitteln hat. So sollte der Benutzer  
nicht durch nur kurzzeitiges Anzeigen von Hinweisen oder geforderten Informatio-  
nen angetrieben oder durch Warten auf im Moment nicht benötigte Berechnungen  
gebremst werden. Weiter sollte er die Interaktion mit dem System beliebig unter-  
brechen können [3, S. 3].

### **Selbsterklärungsfähigkeit:**

Mit *Selbsterklärungsfähigkeit* beschreibt man, inwiefern die Interaktion zwischen  
Benutzer und Software-System aus sich heraus verständlich ist.

So kann dem Benutzer beispielsweise durch eindeutige und direkt verständliche  
Hinweise angezeigt werden, welche Daten er eingeben kann. Nach Dateneingaben  
sollte auch eine Rückmeldung über die Verarbeitung dieser Daten erfolgen.

Die Untersuchung der Selbstklärungsfähigkeit der Software beinhaltet auch die  
Betrachtung der *Benutzerführung*. Hierunter versteht man die Hilfen, die dem  
Benutzer zur Arbeit mit der Software bereitgestellt werden [3]. Es sollte bei-  
spielsweise angezeigt werden, welche Kommandos gerade verfügbar sind, und in  
welchem Zustand sich das System befindet.

### **Erlernbarkeit:**

Bei der Untersuchung der *Erlernbarkeit* wird der Aufwand untersucht, den ein Be-  
nutzer aufbringen muß, um mit einem neuen Software-Produkt arbeiten zu können  
[9, S. 19].

Ein Charakteristikum für eine gute Unterstützung der Erlernbarkeit einer Software  
ist die Führung des Benutzers durch den Lernprozeß mit dem Ziel die aufzubrin-  
gende Lernzeit möglichst zu minimieren.

## **Analyse und Ergebnisdokumentation**

Aufbauend auf diesen Aspekten wird ein Frageleitfaden (siehe hierzu auch Abbildung 6)  
zur Evaluation der Benutzerschnittstelle erstellt. Die Software wird nun dahingehend eva-

liefert, inwieweit sie eine Benutzerschnittstelle bereitstellt, die den geforderten Kriterien entspricht.

Evaluation der Benutzerschnittstelle		Produkt
<b>Selbsterklärungsfähigkeit</b>		
Wird der Systemzustand angezeigt ?	ja	nein
angewählter Menüpunkt	<input type="checkbox"/>	<input type="checkbox"/>
übergeordnete Menüpunkte	<input type="checkbox"/>	
Werden Eingabemöglichkeiten erläutert ?		
Liste der verfügbaren Kommandos	<input type="checkbox"/>	
Information über aktuelles Eingabefeld	<input type="checkbox"/>	
Erfolgen Rückmeldungen nach Dateneingaben ?	<input type="checkbox"/>	
Erfolgen Mitteilungen über länger andauernde Berechnungen ?	<input type="checkbox"/>	
Bemerkungen:		
.....		

Evaluation der Benutzerschnittstelle		Produkt
<b>Erwartungskonformität</b>		
Erfolgt eine einheitliche Darstellung und Positionierung der	ja	nein
Fehlermeldungen ?	<input type="checkbox"/>	<input type="checkbox"/>
Statuszeile ?	<input type="checkbox"/>	<input type="checkbox"/>
Entsprechen die Programmreaktionen den Benutzererwartungen ?	<input type="checkbox"/>	<input type="checkbox"/>
Bemerkungen:		
.....		
.....		
.....		
.....		

Evaluation der Benutzerschnittstelle		Produkt
<b>Individualisierbarkeit</b>		
Können durch den Benutzer Eingabefolgen definiert werden ?	ja	nein
auf Funktionstasten	<input type="checkbox"/>	<input type="checkbox"/>
durch Makros	<input type="checkbox"/>	<input type="checkbox"/>
Bemerkungen:		
.....		
.....		
Können Ausgaben durch den Benutzer gestaltet werden ?		
Bildschirm	<input type="checkbox"/>	<input type="checkbox"/>
Drucker	<input type="checkbox"/>	<input type="checkbox"/>
Bemerkungen:		
.....		
.....		

Abbildung 6: Frageleitfaden zur Evaluation der Benutzerschnittstelle (Ausschnitt)

Neben der Bewertung der Unterstützung dieser Kriterien, werden zusätzlich noch Bemerkungen zu den einzelnen Bewertungen erfasst. Diese dienen bei der Erstellung der Dokumentation der Evaluations-Ergebnisse als Beispiele und Begründungen für Bewertungsentscheidungen.

Es empfiehlt sich, die Evaluation der Benutzerschnittstelle nach der Untersuchung der modellabhängigen Evaluation durchzuführen, da bereits während der quantitativen und qualitativen Evaluation ein Gesamteindruck der Software und deren Bedienung zu erhalten ist. Diese Erfahrungen können dann bei der Evaluation der Benutzerschnittstelle vertieft und konkretisiert werden.

So erfolgte beispielsweise die Einschätzung der *Erlernbarkeit* bei der Untersuchung der Handwerks-Software anhand der Erfahrungen, die während der Einarbeitung und dem Umgang mit dem jeweiligen analysierten Produkt gemacht wurden.

Die Ergebnisse der Benutzerschnittstellen-Evaluation beruhen folglich auf subjektiven Entscheidungen der Personen, die die Evaluation durchführen. Wie auch bei der Betrachtung der modellabhängigen Aspekte der Software, wird dennoch durch die Verwendung

eines Frageleitfadens sichergestellt, daß alle Produkte auch bezüglich der Benutzerschnittstelle begründbar nach den gleichen Kriterien und unabhängig von anderen Produkten bewertet werden können.

Als alternatives Vorgehen zur Untersuchung der Benutzerschnittstelle, das analog in das hier vorgestellte Vorgehensmodell integriert werden kann, sei auf das EVADIS-Verfahren von Oppermann et.al. hingewiesen [12], in dem ein sehr ausführliches Mittel zur Evaluation von Benutzerschnittstellen vorgestellt wird. Die Untersuchung der Benutzerschnittstelle war jedoch nicht das vorrangige Ziel der Betrachtung der Handwerks-Software, so daß es in diesem Kontext ausreichte, die Evaluation direkt entlang der vorgenannten Kriterien durchzuführen.

## **4.2 Evaluation der Rahmenbedingungen**

Zur Vervollständigung der Evaluation werden neben der Benutzerschnittstelle und den modellabhängigen Aspekten der Software noch die Voraussetzungen und Rahmenbedingungen betrachtet, die beim Einsatz dieser Produkte zu berücksichtigen sind. Untersucht werden hierbei *Hardware-* und *Software-Voraussetzungen* sowie *Ausstattungsmerkmale*. Ebenso gehören hierzu Software- und Schulungskosten. Diese Merkmale sind unabhängig vom konkreten Untersuchungsziel, so daß die Evaluation der Rahmenbedingungen entlang allgemeiner Kriterien erfolgen kann.

Bezüglich der Hardware-Voraussetzungen wird erhoben, welche Rechnerkonfiguration (Prozessor, Hauptspeicherkapazität, Festplatte, Monitor, Vernetzung etc.) zum Einsatz der Software nötig ist. Die Software-Voraussetzungen umfassen z.B. das Betriebssystem.

Bei Ausstattungsmerkmalen wird unter anderem nach Handbüchern gefragt. Hier ist zwischen Handbüchern zum Nachschlagen während der Software-Benutzung und Handbüchern mit Lehrbuchcharakter zu unterscheiden. In diesem Zusammenhang kann nach dem Service durch den Vertreiber oder Hersteller der Software gefragt werden. Hier interessieren z.B. Beratungsvermögen des Herstellers auch bzgl. möglicher Umstellungen der Betriebsorganisation, Schulungs- und Einweisungskonzepte, Unterstützung bei der Software-Installation und Vorhandensein einer Hotline.

Weiter kann die Software während der Evaluation der Rahmenbedingungen hinsichtlich der *Offenheit* gegenüber anderen Software-Systemen untersucht werden. Die durch die Software verarbeiteten Daten sollten durch andere Software-Produkte weiterverarbeitet werden können. Ebenso sollte es möglich sein, Daten anderer Programme zu übernehmen. In diesem Zusammenhang sind die durch die evaluierte Software bereitgestellten Schnittstellen zu erheben.

### **Analyse und Ergebnisdokumentation**

Die Rahmenbedingungen und Voraussetzungen für den Software-Einsatz können unabhängig von den anderen Evaluationsaspekten erhoben werden. Grundlage dieses Evaluationsteils sind Prospekte und Produktinformationen der Hersteller. Hierin ist häufig beschrieben, welche Anforderungen an Hard- und Software gestellt werden und was im Lieferumfang der Software enthalten ist. Bezüglich der Beratungskompetenz ist eher auf Gespräche mit Herstellern oder Vertreibern zurückzugreifen.



Unterstützt wird die Erhebung dieser Aspekte der Software-Evaluation ebenfalls durch einen Frageleitfaden. In ausgefüllter Form (vgl. Abbildung 7) kann dieser Fragebogen auch zur Ergebnisdokumentation benutzt werden.

Rahmenbedingungen	Produkt														
<table border="1"> <thead> <tr> <th colspan="2">Hardwarevoraussetzungen</th> </tr> </thead> <tbody> <tr> <td>Prozessor/Hauptspeicher</td> <td>: ab 8088 und 640 KB</td> </tr> <tr> <td>Festplatte</td> <td>: ab 80 MB</td> </tr> <tr> <td>Monitor</td> <td>: mind. Hercules-kompatibel</td> </tr> </tbody> </table>		Hardwarevoraussetzungen		Prozessor/Hauptspeicher	: ab 8088 und 640 KB	Festplatte	: ab 80 MB	Monitor	: mind. Hercules-kompatibel						
Hardwarevoraussetzungen															
Prozessor/Hauptspeicher	: ab 8088 und 640 KB														
Festplatte	: ab 80 MB														
Monitor	: mind. Hercules-kompatibel														
<table border="1"> <thead> <tr> <th colspan="2">Softwarevoraussetzungen</th> </tr> </thead> <tbody> <tr> <td>Betriebssystem</td> <td>: MS-DOS ab Version 3.1</td> </tr> </tbody> </table>		Softwarevoraussetzungen		Betriebssystem	: MS-DOS ab Version 3.1										
Softwarevoraussetzungen															
Betriebssystem	: MS-DOS ab Version 3.1														
<table border="1"> <thead> <tr> <th colspan="2">Ausstattungsmerkmale</th> </tr> </thead> <tbody> <tr> <td>Beratung/Einweisung</td> <td>: erfolgt vor Ort durch regionale Service- und Beratungszentren</td> </tr> <tr> <td>Installation</td> <td>: durch Anwender möglich</td> </tr> <tr> <td>Handbuch</td> <td>: Online-Handbuch, Nachschlagewerk, Lehrbuchcharakter</td> </tr> <tr> <td>Hotline</td> <td>: kostenlos für Kunden, nur bei Vertragsverträgen</td> </tr> <tr> <td>Schnittstellen</td> <td>: DATANG</td> </tr> <tr> <td>sonstiges</td> <td>: Soft</td> </tr> </tbody> </table>		Ausstattungsmerkmale		Beratung/Einweisung	: erfolgt vor Ort durch regionale Service- und Beratungszentren	Installation	: durch Anwender möglich	Handbuch	: Online-Handbuch, Nachschlagewerk, Lehrbuchcharakter	Hotline	: kostenlos für Kunden, nur bei Vertragsverträgen	Schnittstellen	: DATANG	sonstiges	: Soft
Ausstattungsmerkmale															
Beratung/Einweisung	: erfolgt vor Ort durch regionale Service- und Beratungszentren														
Installation	: durch Anwender möglich														
Handbuch	: Online-Handbuch, Nachschlagewerk, Lehrbuchcharakter														
Hotline	: kostenlos für Kunden, nur bei Vertragsverträgen														
Schnittstellen	: DATANG														
sonstiges	: Soft														

Abbildung 7: Analyse und Dokumentation der Rahmenbedingungen (Ausschnitt)

Wie bei der quantitativen Evaluation werden bei der Evaluation der Rahmenbedingungen objektive Daten nach gleichen Kriterien für alle evaluierten Produkte erhoben.

## 5 Zusammenfassung

Das hier vorgestellte Vorgehensmodell zur Software-Evaluation beschreibt ein *leitfadenorientiertes Evaluationsverfahren*. Die Evaluation der Software erfolgt durch eine oder mehrere Personen, die sich hierzu eines Fragenkatalogs bedienen. Dieser kann auf ein Modell des Wirklichkeitsausschnittes zurückgeführt werden, in dem die Software eingesetzt wird, und auf allgemeinere Modelle hinsichtlich des Umgangs mit Softwaresystemen.

Bei einer leitfadenorientierten Software-Evaluation erfolgt die Überprüfung und Bewertung des analysierten Sachverhalts subjektiv durch den Analytiker. Leitfadenorientierte Evaluationen haben aber auch einen objektiven Charakter [11, S. 329]. Mit Hilfe des Fragenkatalogs sind die analysierten Aspekte soweit operationalisiert, daß die Bewertung der Software entlang des zugrundegelegten Modells begründbar und nachvollziehbar ist.

Jedes Evaluationsergebnis der *qualitativen Evaluation* oder der *Evaluation der Benutzerschnittstelle* ist eine subjektive, aber anhand der zugrundeliegenden Kriterien begründbare Entscheidung der die Evaluation durchführenden Personen. Die Untersuchungsergebnisse der *quantitativen Evaluation* und die Ergebnisse der *Evaluation der Rahmenbedingungen* können als weitgehend objektiv angesehen werden, da hier lediglich Aussagen über das Vorhandensein von Unterstützungsmitteln gemacht, beziehungsweise konkrete Daten erhoben werden.

Die Verwendung des Frageleitfadens und Offenlegung der zugrundeliegenden Analyse-kriterien erlaubt eine von anderen Produkten unabhängige Einzelevaluation jeder Software. Die Untersuchung erfolgt nicht durch ein direktes Gegenüberstellen der evaluierten Produkte, sondern durch „Messen“ der relevanten Eigenschaften an einem für alle gleichen Bewertungsmaßstab. Hierdurch wird es möglich, die Evaluation auch auf weitere Produkte für einen ähnlichen Anwendungsbereich auszudehnen. Diese weiteren Untersuchungen erfolgen hierzu anhand der gleichen Kriterien, so daß die Vergleichbarkeit mit den Ergebnissen vorheriger Untersuchungen gewährleistet werden kann.

Durch die Trennung der Software-Evaluation in einen modellabhängigen und einen modellunabhängigen Evaluationsteil, kann die Untersuchung der Software auf das angestrebte Untersuchungsziel ausgerichtet werden. Während im *modellunabhängigen* Teil Aspekte überprüft werden, die unabhängig vom konkreten Evaluationszusammenhang existieren, wird die Software bei der *modellabhängigen Evaluation* im konkreten Anwendungszusammenhang bewertet.

Die Kriterien zur Durchführung der modellabhängigen Evaluation werden aus einem zuvor erstellten Modell des Wirklichkeitsausschnitts gewonnen, in dem die Software eingesetzt werden soll. Je nach Art der Zielsetzung der Evaluation kann die Beschreibung des Einsatzbereichs der Software nach unterschiedlichen Modellierungsparadigmen erfolgen.

Dieses Vorgehensmodell wurde im hier skizzierten Projekt bei der Evaluation von fünf Software-Produkten für die Unterstützung der Auftragsabwicklung im Heizung/Klima/Sanitär-Gewerk und von drei Produkten für den Einsatz in der Fertigungstechnik eingesetzt. Im vorliegenden Analysekontext stand die Unterstützung der Aufgaben innerhalb der Auftragsabwicklung und die Verbindung einzelner Unterstützungsmittel im Vordergrund, so daß die Modellierung nach dem Datenfluß-Paradigma erfolgte.

Die Untersuchung eines Produktes, mit Hilfe der entlang des vorgestellten Vorgehensmodells zur Software-Evaluation erstellten Frageleitfäden, konnte von zwei Personen in einem Zeitraum von jeweils etwas mehr als zwei Tagen durchgeführt werden. Dieser verteilte sich auf ca. zwei Stunden für eine erste Orientierung in der Software, acht bis zehn Stunden für die Durchführung der modellabhängigen Evaluation, ca. zwei Stunden für die Durchführung der modellunabhängigen Evaluation und ungefähr vier Stunden für die Erstellung der Dokumentation. Hierin ist der Aufwand zur Erstellung des Modells und der Ableitung der Untersuchungskriterien sowie Software-Demonstrationen durch Vertreter oder Hersteller nicht enthalten.

Anhand der so entstandenen Kurzbeschreibungen der analysierten Software sollen Produkte zur Unterstützung der Qualifikation von Handwerksmeistern ausgewählt werden.

## Literatur

- [1] Helmut Balzert: *Die Entwicklung von Software-Systemen, Prinzipien, Methoden, Sprachen, Werkzeuge*, Bibliographisches Institut, Zürich, 1988.
- [2] Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut Peschke, Gabriele Rohr, Norbert A. Streitz (Hrsg.): *Einführung in die Software-Ergonomie*, Walter de Gruyter, Berlin, New York, 1988.

- [3] DIN: *Bildschirmarbeitsplätze, Grundsätze der Dialoggestaltung (DIN 66.234, Teil 8)*, Beuth Verlag, Berlin, 1988.
- [4] Uwe Dumslaff, Jürgen Ebert, Markus Mertesacker, Andreas Winter: *Ein Vorgehensmodell zur Softwareanalyse am Beispiel von Branchenlösungen im Handwerk*, Projektbericht CASE 6/92, Universität Koblenz-Landau, Fachbereich Informatik, Rheinau 1, 5400 Koblenz, Oktober 1992.
- [5] Uwe Dumslaff, Markus Mertesacker, Andreas Winter: *Konzeptualisierung und Durchführung der Branchensoftwareanalyse*, in: Jürgen Ebert, Dieter Euler, Martin Twardy (Hrsg.): *Computerunterstützte Auftragsabwicklung im Handwerk – Untersuchung von Problemfeldern und Konzeptualisierung von Bildungsmaßnahmen für die Bereiche der Energie- und Fertigungstechnik*, Adalbert Carl, Laasphe 1992, S: 235–333.
- [6] Jürgen Ebert: *Beschreibungsmethoden für interaktive, transaktionsorientierte Systeme: Ein Überblick*, Softwaretechnik-Trends, 12(2), S. 4–19, Mai 1992.
- [7] Jürgen Ebert, Dieter Euler, Martin Twardy (Hrsg.): *Computerunterstützte Auftragsabwicklung im Handwerk – Untersuchung von Problemfeldern und Konzeptualisierung von Bildungsmaßnahmen für die Bereiche der Energie- und Fertigungstechnik*, Adalbert Carl, Laasphe, 1992.
- [8] Klaus-Peter Fähnrich (Hrsg.): *Software-Ergonomie*, Oldenbourg, München, Wien, 1987.
- [9] ISO: *Ergonomic requirements for office work with visual display terminals (VDT's) (ISO 9241), Part 10: Dialogue Principles, First Committee Draft*, 19. September 1991.
- [10] Glenford J. Myers: *Methodisches Testen von Programmen*, R. Oldenbourg Verlag, München, Wien, 1982.
- [11] Reinhard Oppermann: *Software-ergonomische Evaluationsverfahren* in: Helmut Balzert, Heinz U. Hoppe, Reinhard Oppermann, Helmut Peschke, Gabriele Rohr, Norbert A. Streitz (Hrsg.): *Einführung in die Software-Ergonomie*, S. 323–342, Walter de Gruyter, 1988.
- [12] Reinhard Oppermann, Bernd Murchner, Harald Reiterer, Manfred Hoch: *Software-ergonomische Evaluation, Der Leitfaden EVADIS II*, Walter de Gruyter, Berlin, 1992.
- [13] Douglas T. Ross: *Structured Analysis (SA): A Language for Communicating Ideas*, IEEE Transactions on Software Engineering, 3(1), S. 16–34, Jan. 1977.
- [14] Paul Schmitz, Heinz Bons, Rudolf van Mengen: *Software-Qualitätssicherung — Testen im Software-Lebenszyklus*, Programm Angewandte Informatik, Herausgeber: Paul Schmitz, Norbert Szyperski. Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 2. Auflage, 1983.
- [15] Ben Shneiderman: *Designing the User Interface: Strategies for Effective Human Interaction*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.