Kapitel 4

Vorschlag einer Komplexitätsmessung

In diesem Kapitel wird eine neue Komplexitätsmessung vorgeschlagen. Die Messung basiert auf den Meßverfahren von McCabe und Halstead. Die Programmier-Leistung erweitert die zyklomatische Zahl. Die Beschreibung der Voraussetzung für die Kombination beider Verfahren und die Eigenschaften der Messung erfolgt im Kapitel 4.1. Kann das vorgeschlagene Meßverfahren als Komplexitmessung eingesetzt werden. Diese Fragestellung wird im Kapitel 4.2 überprüft. Die Ermittlung der Programmier-Leistung nach Halstead erwies sich beim Messen der Beispiel-Programme als auswendig. Ein aus den Operatoren und Operanden extrahierter Faktor, vereinfacht die Messung. Kapitel 4.3 beschreibt die Vereinfachung von Halstead mittels der Faktorenanalyse. Die Gesamtergebnisse werden in Kapitel 4.4 diskutiert.

4.1 Erweiterung des Verfahrens von McCabe

Die zyklomatische Zahl von McCabe ist ein Komplexitätsmaß für ABAP/4-Programme, vgl. Kapitel 3.4.2. Das Verfahren beinhaltet jedoch einige Schwachpunkte. Eine umfassende Zusammenstellung aus der Literatur ist in Kapitel 2.2.2 zu finden. Bei den Messungen der ABAP/4-Beispiel-Programme konnten einige Mängel nachvollzogen werden: Die zyklomatische Zahl bildet die strukturelle Komplexität ab und ist nicht sensitiv auf sequentielle Anweisungen. Sequentielle Module und Programme sowie umfangreiche sequentielle Anteile im Programm fließen nicht zufriedenstellend in McCabes Komplexitätsmaß ein [Weyuker88] und [Zuse91]. [Weyuker88] stellt außerdem heraus, daß das Verfahren die Programme nur in einige wenige Komplexitätsklassen einteilt. Außerdem werden Mehrfachaufrufe von Modulen nicht berücksichtigt.

Zur Behebung der oben genannten Mängel gibt es einige Anregungen in der Literatur. Diese Empfehlungen bringen eine Erweiterung der zyklomatischen Zahl mit sich. [vanDoren97a] beschreibt, daß das Verfahren von McCabe gut in Kombination mit anderen Messungen einzusetzen ist. [Hansen78] formuliert z.B. eine neue

Komplexitätsmessung, die auf der zyklomatischen Zahl basiert. Er kombiniert die Messung von McCabe mit der Anzahl der Operatoren und schlägt eine Darstellung als 2-Tupel vor.

4.1.1 Zyklomatische Zahl und Programmier-Leistung

Zwischen den Meßergebnissen von McCabe und Halstead besteht ein hoher Zusammenhang. Die Korrelation ist mit einer Irrtumswahrscheinlichkeit unter 5% statistisch gesichert, Kapitel 3.4.4. Aufgrund diesem hohen Zusammenhang wird eine Komplexitätsmessung vorgeschlagen, die beide Verfahren kombiniert. Es erfolgt eine Erweiterung der zyklomatischen Zahl von McCabe um Halsteads Programmier-Leistung E. Die beiden Größen werden als 2-Tupel betrachtet. Dieser Vorschlag einer Komplexitätsmessung, McHa genannt, zielt darauf hin, Schwächen im Verfahren von McCabe zu beseitigen. Mit dieser Kombination ist die Korrektur der Nicht-Sensivität auf sequentielle Anweisungen und die Unterteilung der Programme in mehrere Komplexitätsklassen möglich.

Die Syntax der erweiterten Komplexität eines Programms wird im allgemeinen wie folgt beschrieben:

$$< ZZ >;< E >$$

$< ZZ >$ ist die zyklomatische Zahl und $< E >$ stellt die Programmier-Leistung dar. Die theoretischen Grundlagen zur Berechnung der zyklomatischen Zahl und der Programmier-Leistung basieren ohne Einschränkungen auf den Theorien von [McCabe76] und [Halstead77].

Reports aus der Beispiel-Auswahl zeigen nach dem vorgeschlagenen Verfahren folgende Komplexitäten:

Report	Komplexität nach McHa	Report	Komplexität nach McHa
S-UM-002	1;415	S-UM-008	1;1431
N-UM-007	4;4056	U-UM-014	5;2316
N-UM-020	7;14462	N-M-001	8;2431
N-UM-021	9;21466	U-M-004	23;28694

4.1.2 Eigenschaften der Komplexitätsmessung McHa

Die erweiterte Komplexitätsmessung McHa besitzt viele Charakteristiken der beiden Ursprungsverfahren:

- Die Komplexitätsmessung ist unabhängig von der Programmformatierung und kann auf eine große Anzahl von Programmiersprachen angewendet werden. Die Sprache muß fundamentale Entscheidungsstrukturen besitzen und darf nur aus Operatoren und Operanden bestehen.

- Das Verfahren mißt die strukturelle und die berechende Komplexität. Zum einen wird die Struktur eines Programms abgebildet. Zum anderen basiert das Verfahren auf algorithmische Charakteristiken, d.h. die Messung ist sensitiv auf sequentielle Anweisungen.
- Die Komplexität eines Programms kann direkt aus dem Programmtext ermittelt werden. Die Struktur wird dazu aus den Verzweigungen generiert. Die berechende Komplexität erfolgt über die Bestimmung der Operatoren und Operanden im Programmtext. Der strukturelle Anteil des Komplexitätsmaßes kann schnell und einfach ermittelt werden, der berechende Anteil ist relativ aufwendig.
- Die Messung klassifiziert die Programme in "komplexer als" und "gleich komplex". Die Programme werden in endlich viele Komplexitätsklassen unterteilt. Der strukturelle Anteil des Komplexitätsmaßes stellt eine grobe Klasseneinteilung dar, die durch den berechende Anteil differenziert wird.
- Die Abfolge von Anweisungen übt keinen Einfluß auf die Komplexität des Moduls aus. Die Komplexitätsmessung ist nicht sensitiv auf die Reihenfolge der Anweisungen.
- Es erfolgt keine Gewichtung von Verzweigungen wie bedingte Ausführungen und Iterationen. Es werden auch keine verschachtelten Fallunterscheidungen oder Iterationen abgebildet.

Neben den oben aufgeführten Eigenschaften genügt die vorgeschlagene Komplexitätsmessung folgenden Kriterien:

- Die Komplexitätsmessung kann mit anderen Meßverfahren kombiniert werden. Die Messungen sollten sich nur geringfügig beeinflussen, um sich zu ergänzen.
- Es besteht eine minimale Überschneidung zwischen der Messung von McCabe und von Halstead. Die zyklomatische Zahl wird aus den Verzweigungen im Modul und die Programmier-Leistung wird aus den Operatoren und Operanden im Modul ermittelt.
- Die Messung ist einfach zu interpretieren und beschreibt die psychologische Komplexität von Programmen.
  Sowohl die zyklomatische Zahl wie auch die Programmier-Leistung betrachten die psychologische Komplexität.
- Das Verfahren dient als Werkzeug, um die Komplexität von Programmen zu vergleichen.

4.1.3 Beispielmessungen

Die Komplexitätsmessung McHa wird anhand einiger Beispiele aus der ABAP/4-Auswahl erläutert. Die entnommenen Beispiele sind in sequentielle und nicht-sequentielle Programme untergliedert.

Report	Komplexität nach McHa	Intuitive Komplexität
S-UM-002	1;415	1
S-UM-008	1;1431	1
S-UM-011	1;3344	2
S-UM-003	1;8521	4
S-UM-003	1;9013	3

Tabelle 4.1: Komplexität nach McHa und intuitiven Komplexität von sequentiellen Programmen

Die erweiterte Komplexität unterteilt die Komplexitätsklasse 1, die bei McCabe sequentielle Programme umfaßt, in zusätzliche Klassen. Sequentielle Programme können durch die Erweiterung besser abgebildet werden. Dies veranschaulicht die intuitive Komplexität.

Auch bei Reports mit struktur-erzeugenden Anweisungen wird durch die erweiterte Komplexität eine Differenzierung der Programme innerhalb einer Komplexitätsklasse, die die zyklomatischen Zahl erzeugt, erreicht.

Report	Komplexität nach McHa	Intuitive Komplexität
N-UM-006	2;1088	2
U-UM-004	2;2091	3
N-UM-019	2;2515	4
U-UM-009	4;1085	3
N-UM-017	5;3206	3
U-UM-006	5;3505	5
U-UM-010	7;4250	4
U-UM-005	12;4909	6
N-UM-018	12;10526	5

Tabelle 4.2: Komplexität nach McHa und intuitiven Komplexität von nicht-sequentiellen Programmen

4.2 Validation der Komplexitätsmessung McHa

Im allgemeinen können Meßverfahren mit anderen Meßverfahren kombiniert werden. Die Verfahren sollten sich minimal überschneiden, um die gegenseitige Beeinflussung möglichst gering zu halten und um eine optimale Ergänzung zu erzielen.

Die zyklomatische Komplexität von McCabe ist um die Messung von Halstead erweitert worden, um eine verbesserte Komplexitätsmessung McHa zu erhalten. Zwei

Schwächen des Verfahrens von McCabe, die Nicht-Sensivität auf sequentielle Anweisungen und die nicht detaillierte Klasseneinteilung sollen behoben werden.

Die Validation der Komplexitätsmessung McHa erfolgt schrittweise: Zu Beginn wird im Kapitel 4.2.1 überprüft, ob die Komplexitätsmessung die Beispiel-Programme richtig bewertet. In Kapitel 4.2.2 erfolgt eine besondere Betrachtung der sequentiellen Programme. Ob im allgemeinen eine differenziertere Klasseneinteilung festzustellen ist, wird ebenfalls in dem Kapitel beurteilt. Abschließend findet im Kapitel 4.2.3 eine Untersuchung statt, ob McHa als Komplexitätsmaß die Komplexität von ABAP/4-Programme insgesamt besser charakterisiert. Eine Schlußbetrachtung der Ergebnisse stellt das Kapitel 4.2.4 dar.

4.2.1 Ist McHa ein Komplexitätsmaß?

Die Untersuchung stellt fest, ob die Komplexitätsmessung McHa als Komplexitätsmaß für ABAP/4-Programme anwendbar ist. Dieser Prozeß verläuft vergleichbar mit der Überprüfung der Verfahren von McCabe und Halstead, vgl. Kapitel 3.4.2 und 3.4.3.

Die drei Beispiel-Programme U-M-005, U-M-006 und U-M-007 bleiben bei der Bewertung der Komplexität nach McHa unberücksichtigt. Eine Klasseneinteilung der ABAP/4-Auswahl erfolgt auf Basis der zyklomatischen Zahl von McCabe. Die Berechnung der Mittelwerte der intuitiven Komplexität und des Komplexitätsmaßes Mcha wird in zwei Stufen vollzogen: Die zyklomatische Zahl ermöglicht eine erste Klassifizierung. Die weitere Untergliederung innerhalb jeder Klasse nimmt die Programmier-Leistung vor. Die Mittelwerte der intuitiven Komplexität wie auch der Durchschnitt des Komplexitätsmaßes McHa berechnen sich über diese zwei Stufen. Die nachfolgende Tabelle ist eine Zusammenfassung der ausführlichen Darstellung im Anhang E. Die nicht einbezogenen Programme sind durch Klammern gekennzeichnet.

Mittelwerte		Mittelwert	
Intu. Kompl.	Kompl. nach McHa	Intu. Kompl.	Kompl. nach McHa
2,07	2;968	4,00	11;4218
1,67	3;790	5,25	12;8237
2,50	4;1464	4,00	14;6341
3,50	5;3926	6,00	15;11287
4,00	6;3025	7,00	19;11770
3,88	7;5764	8,00	23;28694
3,67	8;3730	9,00	(55;44809)
5,50	9;12444	10,00	(77;24967)
5,00	10;5382	11,00	(89;30437)

Tabelle 4.3: Mittelwerte der intuitiven Komplexität und des Komplexitätsmaßes McHa

Eine statistische Auswertung erfolgt bei McHa mittels der multiplen Korrelationsrechnung. Sie wird verwendet, da die intuitive Komplexität von den beiden Einflußgrößen zyklomatische Zahl und Programmier-Leistung abhängt. Die Probenzahl n besteht aus 15 Komplexitätsklassen. Bei einem Korrelationskoeffizienten von $r = +0,940^{***}$ liegt die Irrtumswahrscheinlichkeit P unter 0,1%. Die Steigerung der intuitiven Komplexität ist über die untersuchten Komplexitätsklasse ist als hoch signifikant gesichert, anzusehen. Die Kompexlitätsmessung McHa kann zur Messung der Komplexität von ABAP/4-Programme benutzt werden. Dies bedeutet, daß das Komplexitätsmaß die intuitive Einschätzung bestätigt.

Die einfache multiple Regression zeigt die funktionale Beziehung zwischen der intuitiven Komplexität und dem Komplexitätsmaß nach McHa. Sie wird beschrieben durch: $\mathrm{IK} = 0,17687 \cdot \mathrm{ZZ} + 0,000092 \cdot \mathrm{PL} + 1,9895$. IK steht für die intuitive Komplexität, ZZ für die zyklomatische Zahl und PL für die Programmier-Leistung.

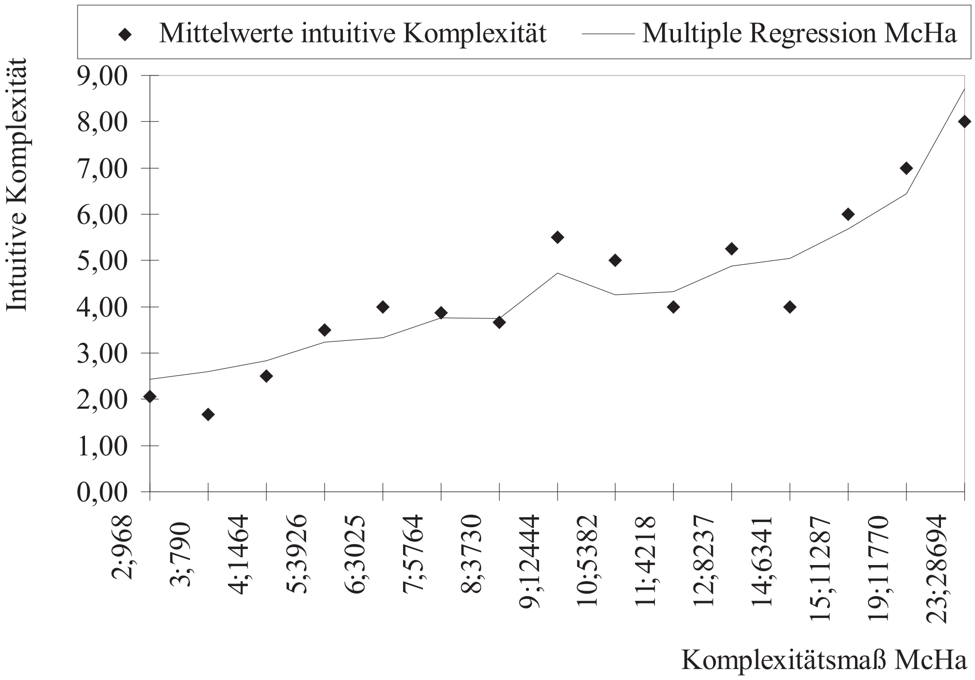

Abbildung 4.1: Graphische Darstellung der multiplen Korrelation zwischen McHa und der intuitiver Komplexität

4.2.2 Betrachtung einzelner Merkmale

In diesem Kapitel wird revidiert, ob durch die erweiterte Komplexitätsmessung McHa sequentielle Programme abgebildet werden und ob eine detaillierte Klasseneinteilung festzustellen ist.

Sequentielle Programme

Das Meßverfahren von McCabe bildet die Struktur eines Programms ab. Es ist nicht sensitiv auf sequentielle Anweisungen. Folglich können sequentielle Programme gar nicht und Programme mit einem hohen Anteil von sequentiellen Anweisungen nur unzureichend bewertet werden.

Die Korrelationsrechnung beschreibt wie stark der Zusammenhang zwischen der Komplexitätsmessung McHa und der intuitiven Komplexität ist. Dazu werden die sequentiellen Programme aus der ABAP/4-Beispiel-Auswahl verwendet.

Mittelwert Intuitive Komplexität	Komplexität nach McHa	Mittelwert Intuitive Komplexität	Komplexität nach McHa
1,00	1;366	2,50	1;3327
1,50	1;703	2,00	1;4724
1,00	1;1431	3,50	1;8767

Tabelle 4.4: Mittelwerte der intuitiven Komplexität und Komplexität nach McHa für sequentielle Programme

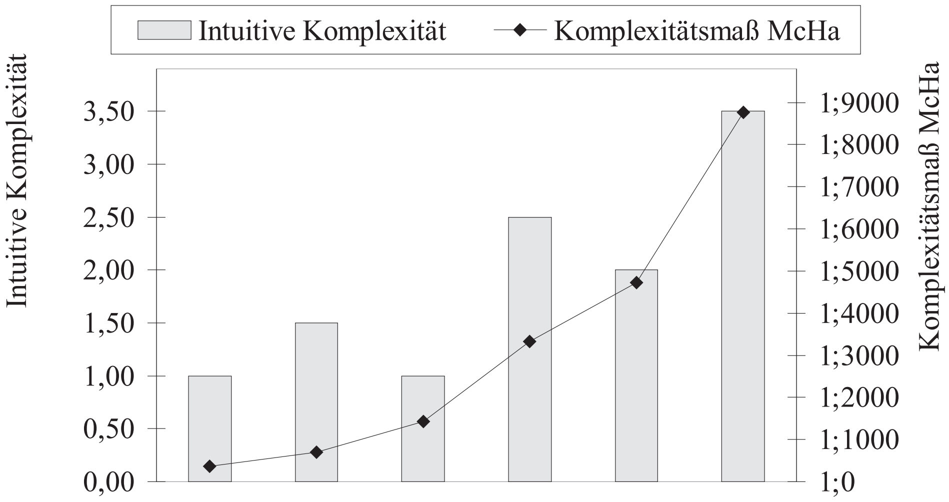

Abbildung 4.2: Abbildung von sequentiellen Programmen durch die Komplexität nach McHa

Ein Anstieg der intuitiven Komplexität in bezug auf die Komplexität nach McHa ist mit einer Irrtumswahrscheinlichkeit unter 1% statistisch gesichert. Der Korrelationskoeffizient r beträgt $+0,924^{**}$. Somit ist die intuitive Bewertung bestätigt.

Die Komplexitätsmessung McHa kann zur Messung der Komplexität von sequentiellen Programmen herangezogen werden. Die Tabelle 4.4 ist in der nachfolgenden Abbildung graphisch aufbereitet.

Die ABAP/4-Beispiel-Auswahl umfaßt keine Programme, die neben struktur-erzeugenden Schlüsselworten auch einen hohen Anteil von sequentiellen Anweisungen beinhalten. Eine Validation der Komplexitätsmessung McHa hinsichtlich entsprechender Programme, kann nicht vorgenommen werden. Aufgrund der hohen statistisch gesicherten Korrelation bei sequentiellen Programmen ist ein gutes Ergebnis zu erwarten. Die Abbildung derartiger Programme wird demnach durch die Komplexitätsmessung McHa verbessert.

Klasseinteilung

Die zyklomatische Zahl faßt viele Programme in eine Komplexitätsklasse zusammen. Innerhalb einer Klasse werden keine weiteren Differenzierungen vorgenommen. Eine genauere Betrachtung der Ergebnisse nach McCabe in bezug auf die intuitive Einschätzung zeigt diesen Mangel.

Report	Intu. Kompl.	Report	Intu. Kompl.	Report	Intu. Kompl.
S-M-003	2	U-UM-010	4	U-M-002	4
U-UM-007	5	N-UM-020	5		

Tabelle 4.5: Intuitive Komplexität verschiedenen Beispiel-Programme für die zyklomatische Zahl 7

Die Komplexitätsmessung McHa differenziert innerhalb eines Komplexitätsklasse. Dies ist in der folgenden Tabelle dargestellt.

Intu. Kompl.	Komplexität nach McHa	Intu. Kompl.	Komplexität nach McHa	Intu. Kompl.	Komplexität nach McHa
		2,00	7;1562		
4,00	5;2883	4,50	7;2783	3,00	8;2431
3,00	5;4969	4,00	7;4250	4,33	8;5029
		5,00	7;14462		

Tabelle 4.6: Darstellung der detaillierten Komplexitätsklassen von McHa

Ein Problem ist, daß nur innerhalb einer Klasse abgestuft wird. Eine Betrachtung über Klassengrenzen hinweg erfolgt nicht.

Dies heißt für die Klasseneinteilung, daß McHa zufriedenstellende Ergebnisse im Bereich einer Komplexitätsklasse nach McCabe liefert. Es können keine Aussagen getroffen werden, wenn zwei Programme aus verschiedenen Klassen betrachtet werden sollen.

4.2.3 Vergleich mit der zyklomatischen Komplexität

In diesem Abschnitt wird die Frage gestellt, ob die Komplexitätsmessung McHa die ABAP/4-Beispiel-Programme insgesamt besser charakterisiert. Die sequentiellen Programme bleiben in dieser Untersuchung unberücksichtig, da bereits eine genaue Überprüfung stattfand. Bei den betrachteten Programmen ist mit keine positiven Veränderung zu rechnen. Die Komplexitätsmessung McHa wirkt nur geringfügig auf die Einordnung der Programme der Beispiel-Auswahl. Die Programme besitzen keine großen Anteile von sequentiellen Anweisungen, um die Stärke des vorgeschlagenen Komplexitätsmäses McHa beurteilen zu können.

Bei der Analyse wird die Klasseneinteilung der zyklomatischen Komplexität nach McCabe verwendet. Die Mittelwerte der intuitiven Komplexität sind auf Basis der zyklomatischen Zahl berechnet.

Zyklomatische Zahl	Mittelwert Intuitive Komplexität	Zyklomatische Zahl	Mittelwert Intuitive Komplexität
2	2,27	10	5,00
3	1,75	11	4,00
4	2,57	12	5,33
5	3,80	14	4,00
6	4,00	15	6,00
7	4,00	19	7,00
8	4,00	23	8,00
9	5,50		

Tabelle 4.7: Mittelwerte der intuitiven Komplexität auf Basis der zyklomatischen Zahl

Der Zusammenhang zwischen der intuitiven Komplexität und der zyklomatischen Zahl ist mit einem Korrelationskoeffizienten von $r = +0,914^{***}$ bei einer Irrtumswahrscheinlichkeit unter 0,1% statistisch gesichert. Die entsprechende Regressionsgerade lautet: $\mathrm{IK} = 0,261 \cdot \mathrm{ZZ} + 1,910$, wobei IK die intuitive Komplexität und ZZ die zyklomatische Zahl repräsentiert.

Wie zu Beginn angedeutet, ist durch das Komplexitätsmaß McHa keine bedeutene Verbesserung hinsichtlich der Einordnung der betrachteten Programme eingetreten. Begründung: Kein Programm der Beispiel-Auswahl demonstriert die Stärke von McHa. Die Komplexitätsmessung McHa ist derart konzipiert, daß der Hauptfokus auf der zyklomatischen Zahl beruht.

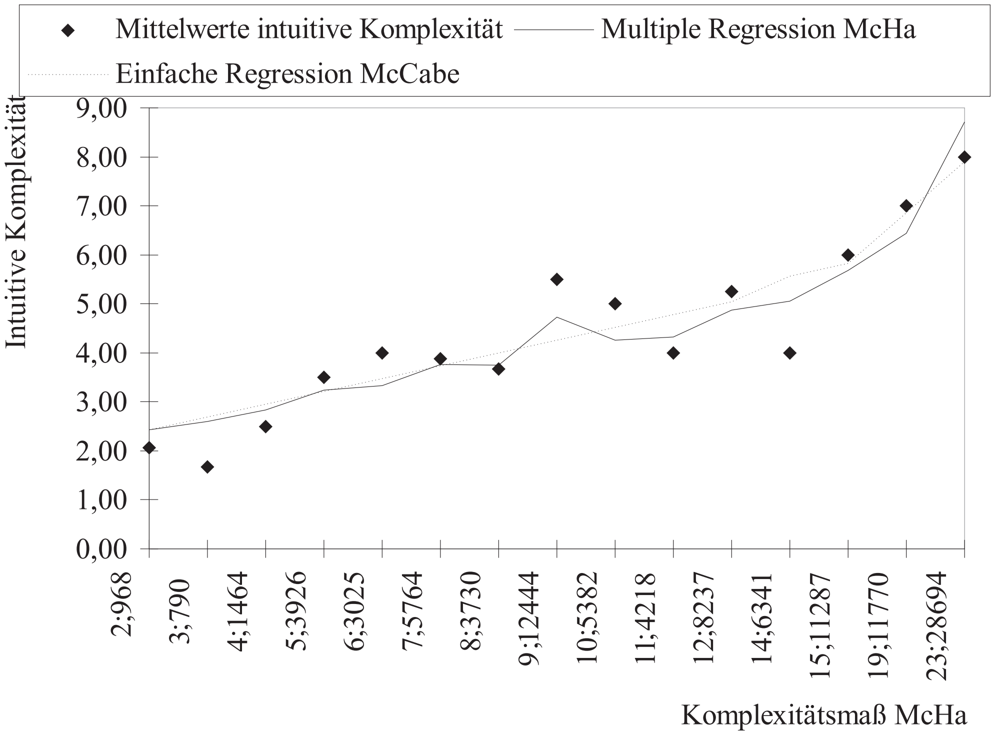

Abbildung 4.3: Vergleich der Regressionen zwischen der zyklomatischen Zahl und der Kompexlität nach McHa

4.2.4 Was kann als Fazit gezogen werden?

Die vorgeschlagene Komplexitätsmessung McHa kann als Komplexitätsmaß für ABAP/4-Programme eingesetzt werden. Zwei Schwächen der zyklomatischen Komplexität von McCabe sind durch die Erweiterung korrigierbar. Sequentielle Programme können von McHa abgebildet werden sowie eine Einteilung in detailliertere Komplexitätsklassen kann erfolgen.

Bei ABAP/4-Programmen, die aus struktur-erzeugenden und sequentiellen Anweisungen bestehen, ist keine nennenswerte verbesserte Abbildung festzustellen. Mit der intuitiven Komplexität korrelieren das Meßverfahren von McCabe und die Komplexitätsmessung McHa jeweils mit einer Irrtumswahrscheinlichkeit von unter 0,1%. Die Messung McHa besitzt einen höheren Korrelationskoeffizienten von $r = +0,940^{***}$ gegenüber den bei McCabe von $r = +0,914^{***}$.

Gründe sind in der Beispiel-Auswahl zu suchen: Die ABAP/4-Auswahl ist mit 62 Programmen relativ gering. Außerdem beinhaltet sie hauptsächlich nur kleine Programme mit LoC kleiner 100, die verhältnismäßig einfach sind. Durch die geringfügige Auswahl standen auch nicht ausreichend viele Programme mit entsprechenden

Eigenschaften zur Verfügung. Merkmale wären z.B. die Modularität, das Verhältnis von sequentiellen Anweisungen und struktur-erzeugenden Schlüsselworten und unstrukturiete Konstrukte. Aufbauend auf die Ergebnisse sind weitere Untersuchungen nötig, um McHa genau zu validieren.

4.3 Vereinfachung der Messung nach Halstead

In der Literatur herrschen viele konträre Meinungen über das Verfahren von Halstead. [Oman91, vanDoren97b] spricht von „einer starken Messung für die Wartbarkeit“. Viele Forscher kritisieren Halstead: [Weyuker88] und [Zuse91] sehen die Programmier-Leistung nicht als Maß für die Komplexität von Software an. Das Ergebnis der Auswertung in Kapitel 3.4.3 beseitigt diese Zweifel für die ABAP/4-Programme. In der vorliegenden Interpretation für die Programmiersprache ABAP/4 kann die Programmier-Leistung zur Komplexitätsmessung eingesetzt werden. Dies ist Voraussetzung, um die zyklomatische Zahl mit die Messung von Halstead erweitern zu können.

Ein schwieriges Problem stellt die eigentliche Messung der Basiseigenschaften von Halstead dar. Die Ermittlung der Anzahl von Operatoren η_1, N_1 und Operanden η_2, N_2 ist aufwendig. Weitere Messungen lassen sich aus den Basiseigenschaften einfach berechnen. Da die Programmier-Leistung E in die neue Komplexitätsmessung einfließt, soll der Aufwand für die Messung der Basiseigenschaften mit Hilfe der Faktorenanaylse reduziert werden. Es wird ein Faktor gesucht, in dem sich die Operatoren und Operanden ausdrücken. Dieser Faktor ersetzt die Programmier-Leistung E in der vorgeschlagenen Komplexitätsmessung. Die Komplexitätsmessung McHa wäre im Vergleich zu den beiden einzelnen Messungen effizienter.

4.3.1 Vorbereitungen zur Faktorenanalyse

Die Faktorenanaylse basiert zum einen auf der Grundannahme der linearen Kombination von Faktorenladungen und Faktorenwerten. Zum anderen beruht sie auf der Vorstellung, daß die Erklärung von Variablen durch aus ihnen gewonnene hypothetische Variablen möglich ist. Es wird davon ausgegangen, daß sich in den Variablen unter den angegebenen Bedingungen ein oder mehrere gemeinsame Faktoren ausdrücken. Eine direkte Beobachtung dieser Faktoren ist nicht möglich. Aufgrund hoher wechselseitiger Korrelationen von direkt quantifzierten Variablen wird auf das Vorhandensein der Faktoren geschlossen.

Die Faktoren stellen einen vereinfachten Interpretationsrahmen für die Ausgangsdaten dar. Die Korrelationskoeffizienten, die die Zusammenhänge der beobachteten Variablen beschreiben, werden auf einen oder mehrere Faktoren zurückgeführt. Die Faktoren bilden die Basis, aus der sich mit größtmöglicher Sicherheit die Ausgangsdaten der Untersuchungsobjekte wieder erzeugen lassen würden. Das Ziel ist, so wenige Faktoren wie nötig aus den Korrelationen der Variablen zu extrahieren, um diese möglichst einfach zu beschreiben und genau zu reproduzieren.

Klasse $2^i \leq E < 2^{i+1}$ i	Mittelwerte Eindeutige Operatoren η_1	Operanden η_2	Gesamtzahl Operatoren N_1	Operanden N_2
7	9,0	6,0	17,0	7,0
8	9,5	9,5	21,3	11,5
9	13,2	15,6	32,1	18,3
10	16,6	28,4	55,4	34,7
11	24,2	33,4	79,8	49,5
12	27,6	37,1	110,5	66,8
13	32,7	60,0	190,2	118,5
14	46,8	94,5	361,0	222,8
15	60,0	221,0	681,0	413,0

Tabelle 4.8: Datenmatrix als Ausgang der Faktorenanalyse

	Eindeutige Operatoren η_1	Operanden η_2	Gesamtzahl Operatoren N_1	Operanden N_2
η_1	0,9988	0,9325	0,9553	0,9586
η_2	0,9325	0,9995	0,9921	0,9909
N_1	0,9553	0,9921	1,0003	0,9999
N_2	0,9586	0,9909	0,9999	0,9997

Tabelle 4.9: Tabellendarstellung der Korrelationsmatrix r

Wie in der Einführung zu diesem Kapitel angedeutet wird ein Faktor gesucht, der die Basiseigenschaften abbildet. Aus den reproduzierten Operatoren und Operanden kann die Programmier-Leistung E berechnet werden. Eine Faktorenanalyse beginnt mit der Datenmatrix. Sie wird durch die Tabelle 4.8 beschrieben:

Die Datenmatrix umfaßt vier Variablen und neun Klassen. Die Klasseneinteilung erfolgt auf Basis der Programmier-Leistung E. Pro Klasse sind die Mittelwerte der Operatoren η_1, N_1 und Operanden η_2, N_2 dargestellt.

4.3.2 Extraktion des Faktors

Aus der Datenmatrix berechnet sich die Korrelationsmatrix r. In die Diagonale der Korrelationsmatrix werden die Schätzungen der Kommunalitäten eingesetzt. Diese sind nach dem Quadrat der multiplen Korrelationskoeffizienten berechnet.

Aus der erhaltenen reduzierten Korrlationsmatrix r_h erfolgt die Extraktion des Faktors a nach der Hauptachsenmethode. Dieser Faktor genügt, um die Korrelationsmatrix r_h möglichst genau zu reproduzieren. Die geschätzten Kommunalitäten sind in der Matrix r_h eingeklammert.

$$r_h = \begin{pmatrix} (0,9719) & 0,9325 & 0,9553 & 0,9586 \\ 0,9325 & (0,9872) & 0,9921 & 0,9909 \\ 0,9553 & 0,9921 & (0,9999) & 0,9999 \\ 0,9586 & 0,9909 & 0,9999 & (0,9999) \end{pmatrix}$$

Eine anschließende Rotation erfolgt nicht, da die Lösung nicht verbessert wird. Der Faktor a mit seinen Ladungen lautet:

$$a = \begin{pmatrix} 0,9660 \\ 0,9994 \\ 0,9989 \\ 0,9994 \end{pmatrix}$$

Den Bezug der Datenmatrix zu dem Faktor wird durch die Faktorenwerte $p^{'}$ hergestellt. Ihre Bestimmung findet durch die multiple Regressionsrechnung statt.

$$p^{'} = \begin{pmatrix} -0,8992 \\ -0,7577 \\ -0,6586 \\ -0,3683 \\ -0,2935 \\ -0,2989 \\ 0,1961 \\ 0,7378 \\ 2,3200 \end{pmatrix}$$

Im Anhang D sind die einzelnen Schritte zur Extraktion des Faktors ausführlich angegeben.

4.3.3 Interpretation des Ergebnis

Der extrahierte Faktor a beschreibt die Korrelationen der beobachteten Variablen. Seine Varianz beträgt 98,86% der Gesamtvarianz. Zusammen mit den Faktorenwerten p ist es möglich die Datenmatrix aus Tabelle 4.8 zu rekonstruieren.

Der Faktor a kann anstatt der Programmier-Leistung E in der vorgeschlagenen Komplexitätsmessung eingesetzt werden. Für die ABAP/4-Beispiel-Auswahl ist er als Vereinfachung der Messung von Halstead anzusehen.

4.3.4 Anwendung in der vorgeschlagenen Komplexitätsmessung

Die Vereinfachung der Messung nach Halstead in der Anwendung der Komplexitätsmessung McHa wird an zwei Beispielen demonstriert. Bei allen vier Komplexitätsmessungen ist das Programm U-UM-010 komplexer als N-UM-017.

	ABAP/4-Beispiele	
	N-UM-017	U-UM-010
Intuitive Komplexität	3	4
Zyklomatische Zahl	5	7
Programmier-Leistung	3206	4250
Komplexität nach McHa	5;3206	7;4250

Tabelle 4.10: Zwei Beispiele im Vergleich der vier Komplexitätsmessungen

Die beiden Programme lauten in der vereinfachten Variante der Komplexität nach McHa:

$$5; \begin{pmatrix} 0,9660 \\ 0,9994 \\ 0,9989 \\ 0,9994 \end{pmatrix} \quad \text{bzw.} \quad 7; \begin{pmatrix} 0,9660 \\ 0,9994 \\ 0,9989 \\ 0,9994 \end{pmatrix}$$

Aus dem Faktor können mit Hilfe der Faktorenwerte p die vier Basisgrößen berechnet werden. Die Tabelle stellt die eindeutigen Operatoren η_1 und Operanden η_2 und die Gesamtzahl N_1 und N_2 sowie die entsprechenden Programmier-Leistungen E gegenüber:

		Basisgrößen				
		η_1	η_2	η_1	η_2	E
N-UM-017	(gemessen)	26	23	74	43	3206
	(berechnet)	20	31	91	55	3685
U-UM-010	(gemessen)	30	32	95	57	4250
	(berechnet)	21	35	106	64	4548

Tabelle 4.11: Gegenüberstellung der Programmier-Leistung aus den gemessenen und reproduzierten Basisgrößen

Die aus den berechnten Basisgrößen gewonnene Programmier-Leistung ordnet die beiden Programme auf die gleiche Art und Weise ein.

4.4 Diskussion der Gesamtergebnisse

Die beiden in der Diplomarbeit betrachteten Meßverfahren spiegeln eine Tendenz in der Messung der Komplexität von Software wieder. Die Merkmale, die Einfluß auf die Komplexität von Programmen nehmen, können nicht alle durch ein einziges Verfahren beschrieben werden. Jede Methode weist Stärken wie auch Schwächen auf. Schwächen bedeuten, daß z.B. bestimmte Merkmale, die die Komplexität im

Programm hervorrufen, nicht abgebildet werden oder daß Verfahren für gewisse Programmiersprachen und in Programm-Entwicklungsstufen nicht eingesetzt werden können. Eine Konsequenz ist, daß die Komplexiät von Software besser durch ein n-Tupel als durch eine einzelne Größe ausgedrückt wird. Hinter einem n-Tupel stehen in der Regel mehrere Verfahren, die sensitiv auf unterschiedliche Programm-Merkmale sind.

Ein Ergebnis der Diplomarbeit besagt, daß die zyklomatische Zahl von McCabe und die Programmier-Leistung von Halstead zur Messung der Komplexität von ABAP/4-Programmen eingesetzt werden können. Eine Schwäche des Verfahrens von McCabe ist, daß es keine sequentiellen Programme und Programme mit einem hohen Anteil von sequentiellen Anweisungen unzureichend abbilden kann. Dieser Nachteil wird durch die Kombination mit der Programmier-Leistung von Halstead behoben. Die vorgeschlagene Komplexitätsmessung McHa setzt sich aus der zyklomatischen Zahl und der Programmier-Leistung zusammen und wird als 2-Tupel betrachtet. Die Validation der Messung zeigt, daß McHa sequentielle Programme einordnet. Eine Untersuchung von Programmen mit einem hohen sequentiellen Anteil erfolgte nicht in der Diplomarbeit. Im Vergleich zur zyklomatischen Zahl wird bei nicht-sequentiellen Programmen keine Verbesserung hinsichtlich der Einordnung erreicht.

Das Problem bei dem Meßverfahren von Halstead ist, daß die Ermittlung der Basiseigenschaften aufwendig ist. Die Extraktion eines Faktors mittls der Faktorenanalyse deutet darauf hin, daß sich hinter den Operatoren und Operanden ein Faktor befindet, der die Zusammenhänge der beobachteten Variablen beschreibt. Die Anwendung in der vorgeschlagenen Komplexitätsmessung McHa ist an zwei Beispielen demonstriert, daß der Einsatz des Faktors diskutabel ist.

Die vorgeschlagene Komplexitätsmessung McHa wurde an der ABAP/4-Beispiel-Auswahl validiert. Eine Überprüfung an großen Beispielen müßte erfolgen, um sie zu bestätigen. Dies gilt auch für die Vereinfachung von Halstead. Prinzipiell ist eine Reduzierung der Basisgrößen auf einen Faktor möglich. Die Reproduzierung der Größen aus dem Faktor und den Faktorenwerten muß ebenso an großen Beispielen untersucht werden. Insgesamt ist die Komplexitätsmessung McHa mit dem Faktor effizienter als die einzelne Ermittlung der Operatoren und Operanden.