Kapitel 6

Zusammenfassung und Ausblick

Für die von der SAP AG entwickelte Programmiersprache ABAP/4 sind in der Literatur keine Verfahren beschrieben, die die Komplexität von ABAP/4-Programmen messen. Das Erfordernis, die Komplexität von Programme zu messen, geht aus dem Releaseprojekt „juDit“ hervor. Die Entwickler, die eigenentwicklete ABAP/4-Programme auf den neuen Release umsetzten, bewerteten intuitiv eine Reihe von Programme als kompliziert bzw. als zu komplex. Daraus entwickelte sich die Idee, die intuitive Einschätzung von ABAP/4-Programmen durch geeignete Verfahren zu überprüfen.

Wie zu Beginn beschrieben, existieren keine Methoden, um die Komplexität von ABAP/4-Programmen betrachten zu können. Zwei anerkannte Meßverfahren, die zyklomatische Zahl von [McCabe76] und die Programmier-Leistung von [Halstead77] werden zur Messung der Komplexität von ABAP/4-Programmen herangezogen. In ihrer vorliegenden Interpretation für die Sprache ABAP/4, stellen sie die ersten beschriebenen Methoden zur Messung derartiger Programme dar.

Drei Ziele wurden in der Diplomarbeit behandelt. Zusammengefaßt sind die Ergebnisse:

1. Die Messung der Komplexität von ABAP/4-Programmen kann durch die Verfahren von McCabe und Halstead erfolgen.

2. Die vorgeschlagene Komplexitätsmessung McHa kann im Gegensatz zur zyklomatischen Zahl von McCabe sequentielle ABAP/4-Programme messen. Außerdem ist die Messung in bezug auf die Programmier-Leistung von Halstead effizienter.

3. Die neuer Version der Lieferscheinerstellung hat eine geringere Komplexität im Vergleich zur alten Version.

Die beiden folgenden Kapitel 6.1 und 6.2 geben einen Überblick der einzelnen Ergebnisse der Diplomarbeit wieder und beschreiben einen Ausblick aufbauend auf den Resultaten.

6.1 Zusammenfassung

Die Meßverfahren von McCabe und Halstead stammen aus Mitte der siebziger Jahre. Sie stellen eine der ersten vollständigen Software-Messungen dar und haben weitere Entwicklungen beeinflußt. Obwohl sie eine Vielzahl von Schwächen aufweisen und nicht immer unumstritten sind, gehören sie zu den anerkannteren Komplexitätsmessungen, vgl. Kapitel 2.1. Sie werden noch in vielen Bereichen zur Komplexitätsmessung von Programmen eingesetzt. Die Verwendung kann als alleinige Messung oder in Kombination mit anderen Verfahren erfolgen. Durch die Verknüpfung mehrerer Methoden können die verschiedenen Merkmale, die Einfluß auf die Komplexität von Software haben, besser abgebildet werden. Außerdem gleichen die einzelne Verfahren ihre Schwächen gegenseitig aus.

Die Untersuchungen in der Diplomarbeit haben gezeigt, daß die Meßverfahren von McCabe und Halstead zur Messung der Komplexität von ABAP/4-Programmen einsetzbar sind. Dies gilt unter den Voraussetzungen, daß erstens, die beschriebenen ABAP/4-Interpretationen für die Verfahren eingehalten werden. Zweitens muß die vereinfachte ABAP/4-Ablaufsteuerung gelten, d.h. die externe Steuerung durch ABAP/4-Ereignisse wird vernachlässigt. Der Vorschlag der Komplexitätsmessung McHa zielt auf den Ausgleich der Schwäche der zyklomatischen Zahl, keine sequentielle Programme abzubilden. Die Messung besteht aus einem 2-Tupel $< ZZ >; < E >$. ZZ stellt die zyklomatische Zahl von McCabe und E die Programmier-Leistung nach Halstead dar. Die Validation der Komplexitätsmessung McHa zeigt, daß McHa sequentielle Programme besser abbildet. Nur bei nicht-sequentiellen Programmen ist keine Verbesserung feststellbar. Die erwartete detailliertere Klasseneinteilung durch McHa kann eingeschränkt bestätigt werden. Innerhalb einer Komplexitätsklasse auf Basis der zyklomatischen Zahl differenziert McHa zwischen den Programmen. Ungeklärt bleibt der Übergang von einer Klasse zur anderen.

Bei der Messung nach Halstead ist die Ermittlung der Anzahl der Operatoren und Operanden aufwendig. Alle Messungen von Halstead, wie die Programm-Länge N, die Programm-Größe V und die Programmier-Leistung E werden aus vier Basisgrößen, die Operatoren und Operanden errechnet. Mit Hilfe der Faktorenanalyse lassen sich diese Größen durch einen Faktor beschreiben. Der Faktor und den Faktorenwerten erlauben die Berechnung der Basisgrößen und die Bestimmung der erwünschten Messungen. Diese Vereinfachung gilt im Rahmen der Diplomarbeit nur für die Beispiel-Programme. Angewendet in der Komplexitätsmessung McHa wird diese effizienter. Die Operatoren und Operanden müssen nicht mehr gemessen werden, sondern können direkt berechnet werden.

Die Betrachtung der Komplexität der alten und neuen Version der Lieferscheinerstellung hat das Resultat, daß die neue Version eine geringere Komplexität aufweist. Das gilt sowohl für die strukturelle wie auch für die berechnende Komplexität. Eine genauere Analyse der Programme zeigt eine überarbeitete Modulstruktur. Weder das Verfahren von McCabe, noch das von Halstead sind sensitiv auf Modulbeziehungen. Eine Erklärung der Verringerung der Komplexität bezieht sich auf die Struktur, die von Modulen durch ihre Anzahl und ihre Aufrufe erzeugt wird. Eine Verringerung der Anzahl der Module und das Ausschließen von überflüssigen Aufrufen wirkt sich positiv auf die strukturelle Komplexität aus.

6.2 Ausblick

Zwei anerkannte Komplexitätsmessungen sind in der Diplomarbeit auf die Programmiersprache ABAP/4 übertragen und angewendet worden. Die zyklomatische Zahl von McCabe und die Programmier-Leistung von Halstead stellen in der vorliegenden Interpretation erste Verfahren zur Messung der Komplexität für AbAP/4-Programme dar.

Die vereinfachte Betrachtung der ABAP/4-Ereignissteuerung ermöglicht die Anwendung der beiden Verfahren. ABAP/4 ist eine ereignisorientierte Sprache und besitzt neben einer internen auch eine externe Steuerung. Der Einbezug der externen Steuerung, die durch Ereignisse hervorgerufen wird, wäre eine potentielle Fortsetztung der Komplexitätsanalyse. Ereignisse rufen in ABAP/4 strukturelle Komplexität hervor. Die zu stellende Frage ist: Wie hoch ist ihr Einfluß auf die Komplexität im Programm?

Es existieren weitere Komplexitätsmessungen, die in dieser Arbeit nicht betrachtet werden konnten. Dazu zählen z.B. die Function-Point-Methode von [Albrecht83], die Informationsfluß-Messung von [Henry81] und die Datenfluß-Messung von [Oviedo80]. Zu untersuchen wäre, ob und wie die Verfahren auch auf die Sprache ABAP/4 anzuwenden sind. Können die Messungen als Komplexitätsmaß für ABAP/4-Programme angesehen werden?

Direkt aufbauend auf die Diplomarbeit und die Komplexitätsmessungen wäre eine Ausweitung der Betrachtung auf Dialogprogramme. Die Validation bei Reports hat gezeigt, daß die Verfahren als Komplexitätsmaß herangezogen werden können. Wie sieht es bei Dialogprogrammen aus?

Die Validation der Meßverfahren fand auf einer kleinen Auswahl von Schulungsprogrammen statt. Zu überprüfen wäre, ob die Messungen bei großen Beispielen noch zuverlässige Ergebnisse liefern. Das heißt, bestätigen mögliche Meßergebnisse bei umfangreichen ABAP/4-Programmen die Schlußfolgerungen aus der Beispiel-Auswahl in dieser Arbeit?