

A MEASURE OF SOFTWARE COMPLEXITY

by

LLOYD DAVID BORCHERT

B.S., Oglethorpe University, Atlanta, Georgia, 1972

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree


MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

Approved by:


Major Professor

SPEC
COLL
LD
2668
R4
1981
B67
c.2

111200 068994

TABLE OF CONTENTS

Chapter 1. Introduction		1
A. Overview of Report	1	
B. Importance of complexity measurement	1	
C. Definition of program complexity	2	
D. Characteristics of measurement criteria	3	
Chapter 2. Some current methods for measuring program complexity		5
A. McCabe (Graph theory cyclomatic number)	5	
B. Chapin (Set theory index of module complexity)	5	
C. McClure (Control variable and module complexity)	12	
D. Halstead (Measure of program effort)	24	
Chapter 3. Basis of proposed measure		29
A. Characteristics measured	29	
B. Computation of measure	31	
C. Example of computation	31	
Chapter 4. Other examples and comparative analyses		33
Chapter 5. Conclusions		39
Chapter 6. Future work		40
Appendix A		41
References.		42

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

Chapter 1

Introduction

A. Overview of Report.

This report describes several complexity measures and illustrates how each can be used to quantify the complexity of a sample program. The purpose of the report is to present several different and unique approaches to program complexity measurement, each using a different program characteristic, and then to present an alternative approach to complexity measurement. The report is structured to first give the reader an introduction to program complexity, complexity measurement, and the types of characteristics used in measuring program complexity. Next, the reader is presented several current methods for measuring program complexity with the basis of each measure, the method of computation, and an example of computation, each using the same sample program. Then, a new complexity metric is presented with the basis of the proposed measure, the computation of the measure, and the example of computation. This is followed by additional examples, a comparative analysis of all the metrics presented, some conclusions, and the future work that is required on this proposal.

B. Importance of complexity measurement.

Anyone who has studied software engineering has had the

opportunity to recognize that there are many different ways to design and code a program that will accomplish the same basic function. The programming style used by most programmers is individualistic and generally based upon their own experiences and education. It is these diverse approaches to design and programming style that give us our traditional problems in developing, testing, and maintaining computer software. As demonstrated by many authors on this subject, often over half of the development time is spent in testing. Additionally, the total cost of a large software system is dominated by program maintenance. The increased cost of software development and software maintenance and the decreased cost of hardware has caused a shift in emphasis from concerns of machine efficiency to the production of clear and structured programs that can be easily maintained.

C. Definition of Program Complexity

Program complexity is an objective measure of how simple or how difficult a program is to understand when the coded instructions for that program are examined by an individual who is proficient in the language in which the coded instructions are written. Although related, the measure of program complexity is separate from the inherent complexity of the total problem. The inherent complexity of a problem is determined by the nature of the problem. Program complexity is a function of the process used to solve the problem. In the past, individuals have generally relied on intuition and common sense to measure what appeared to be simple and what appeared