

THE APPLICATION OF COMPLEXITY METRICS
TO A SOFTWARE REQUIREMENTS SPECIFICATION

by

ALICE JOSEPHINE LEMIEUX

B. A., University of Maine, 1979

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

1984

Approved by:


Major Professor

LD
2668
.R4
1984
L45
c. 2

111202 662231

TABLE OF CONTENTS

1. INTRODUCTION.....	1
1.1 Overview.....	1
1.2 Software Life Cycle.....	1
1.3 Software Requirements Specifications.....	2
1.4 Application of Metrics Early in Life Cycle.....	3
1.5 Software Metrics and Their Uses.....	4
1.6 Software Complexity Metrics.....	6
1.7 McCabe's Control Flow Metric.....	8
1.8 Halstead's Software Science Metrics.....	9
1.9 Henry and Kafura's Information Flow Metrics.....	12
1.10 Scope of the Implementation.....	14
2. IMPLEMENTATION REQUIREMENTS AND DESIGN DECISIONS.....	17
2.1 Overview.....	17
2.2 Application of Halstead's Complexity Metrics.....	19
2.3 Application of Procedure Complexity Metric.....	21
2.4 Design Decisions Influenced by the Requirements.....	22
3. DESIGN OF COMPLEXITY METRICS APPLICATION.....	23
3.1 Overview.....	23
3.2 Scanning the Specification Input.....	25
3.3 Calculation of the Complexity Metrics.....	26
3.4 Output From The Analyzer.....	27
4. IMPLEMENTATION OF A COMPLEXITY METRIC ANALYZER.....	28
4.1 Overview.....	28
4.2 Scanning the Requirements Specification.....	28
4.3 Calculation of the Complexity Metrics.....	30
4.4 Testing the Analyzer.....	32
5. CONCLUSIONS AND EXTENSIONS.....	33
BNF For The ERA Requirements Specification.....	37
Example of an ERA Requirements Specification.....	41
Requirement Specification for the Implementation.....	47
Header File Of Tables For Implementation.....	56
Source Code For Implementation.....	58

LIST OF FIGURES

Figure 1 - Complexity Analyzer Data Flow Diagram.....	24
Figure 2 - Afferent Flow.....	24
Figure 3 - Efferent Flow.....	24

CHAPTER 1

1. INTRODUCTION

1.1 Overview

This report describes the application of complexity metrics to the software requirements specification. A complexity analyzer is developed which applies the complexity metrics of Halstead [8] and Henry and Kafura [9] to the requirements specification. The application of complexity metrics to the software requirements specification is an attempt to predict the complexity of a software project early in its development.

1.2 Software Life Cycle

The software life cycle is an important concept of software engineering. The software life cycle is a macro view of the process through which software is developed. Requirements analysis, requirements specification, preliminary design, detailed design, coding, testing, and operations and maintenance are the phases which make up the software life cycle[15].

Each phase in the software life cycle is supported by the documentation which it uses and produces. The documentation produced during the software life cycle phases are an important part of the software product. This documentation is the means by which the software product

moves from one software life cycle phase to another during its development and maintenance.

1.3 Software Requirements Specifications

The software requirements specification is a product of the requirements specification phase of the software life cycle. The software requirements specification provides the foundation for the remainder of the software development process. The software requirements specification provides an important means of communication among the user or requester, the analyst, the software developers, and the management.

The software requirements specification describes the meaning of the software to be developed and the constraints which must be adhered to by the software developers. The flow and structure of information, descriptions of the functions of the software, interfaces between functions, constraints on the design, and validation criteria are provided in the software requirements specification. The software requirements specification should provide all of the information required for the design of the software and the validation of that design.

Many formats for the software requirements specification have been proposed and, in some cases, marketed. The requirements specification may describe the software through the flow of information, the structure of

information, or the functions performed on the information. An information flow approach is of particular interest for this report.

The software requirements specification which will be analyzed by this implementation is in an entity-relationship-attribute form. The entity-relationship-attribute form is most closely aligned to the information flow approach to software requirements specification. The information flows between entities which have attributes and relationship to other entities.

The entity-relationship-attribute form of the software requirements specification is a text description of the software to be developed. The flow of the information through the entities is described by the attributes and relationships which are associated with the entities. There is no graphical representation of the flow of information through the entities as there is in some other formats of information flow software requirements specifications.

1.4 Application of Metrics Early in Life Cycle

Since the software requirements specification guides the further refinement and detail of the software being developed, it is important for the specification to be understandable and maintainable. The terms understandable and maintainable are often considered to be synonyms for complexity[5]. Complexity introduced in the requirements