need Dinosaur!

PRESENTATION

TO

GMISCA

COMPUTER TECHNOLOGY CONFERENCE

ON

"PERFORMANCE IMPROVEMENT:

TECHNIQUES AND EXPERIENCES"

INPUT

11 JUNE 1979

1. INTRODUCTION

2. CURRENT PERFORMANCE

3. TRENDS

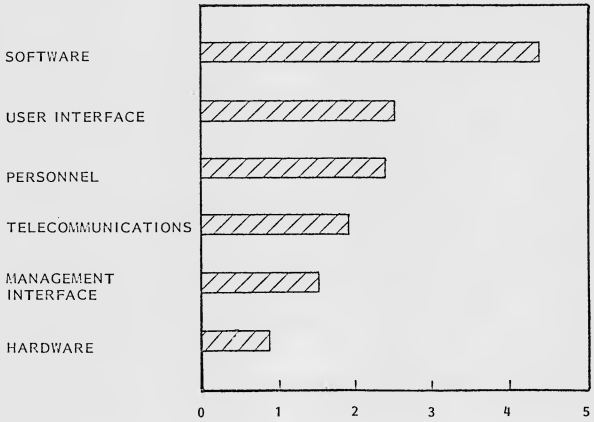
4. RECOMMENDATIONS

INTRODUCTION

SOFTWARE APPLICATIONS

- TYPES OF APPLICATION
  - MAIN-LINE
  - TACTICAL
  - STRATEGIC
- SOURCES
  - IN-HOUSE DEVELOPMENT
  - PACKAGES
  - CONTRACT DEVELOPMENT
  - SERVICE
- IN-HOUSE DEVELOPERS
  - EDP DEPARTMENT
  - USERS

RELATIVE IMPORTANCE OF IMPROVEMENTS
IN THE OVERALL EDP ENVIRONMENT*

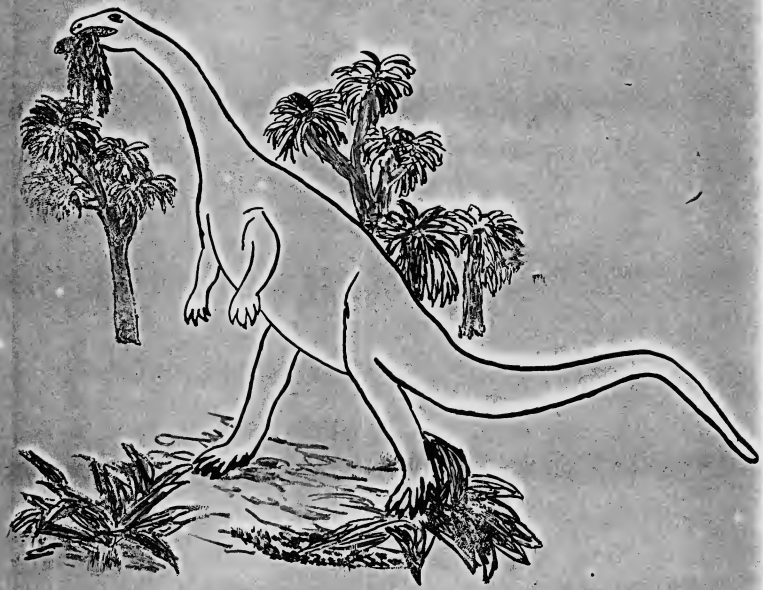


*BASED ON COMBINING RESPONDENT ESTIMATE OF NEED WITH RESPONDENT ESTIMATE OF ABILITY TO MAKE IMPROVEMENTS.

HOW MUCH CAN THE EDP AREA BE IMPROVED?

EDP AREA NEEDING IMPROVEMENT	POTENTIAL IMPROVEMENT RANGE (PERCENT)*
USER INTERFACE	20 - 1000%
MANAGEMENT INTERFACE	20 - 1000%
SOFTWARE	20 - 200%
TELECOMMUNICATIONS	60 - 100%
HARDWARE	25 - 50%
PERSONNEL	20 - 200%

* AS ESTIMATED BY RESPONDENTS



EDP ENVIRONMENT

- DEMAND FOR APPLICATION SOFTWARE GREATER THAN EVER
  - COMPLEXITY INCREASING
  - LIMITS HARDWARE SALES
  - BASED ON OUTDATED LANGUAGE

EDP ENVIRONMENT

- IN THE PAST
  - MAXIMIZE MACHINE EFFICIENCY
- IN THE FUTURE
  - MAXIMIZE PEOPLE EFFICIENCY

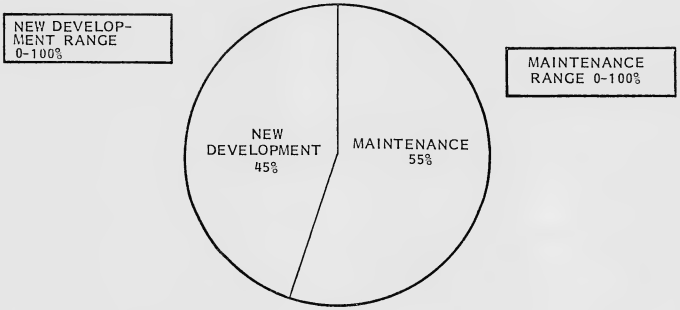
CURRENT PERFORMANCE

INPUT

RESPONDENTS' ALLOCATION OF EFFORT

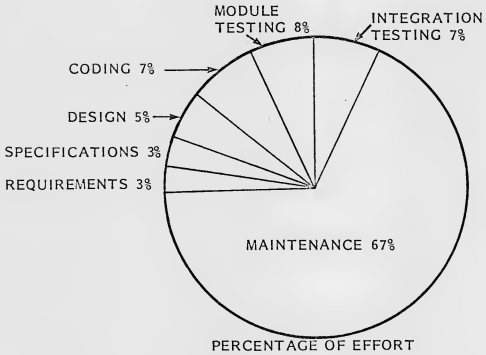
BETWEEN NEW DEVELOPMENT AND

MAINTENANCE OF SOFTWARE



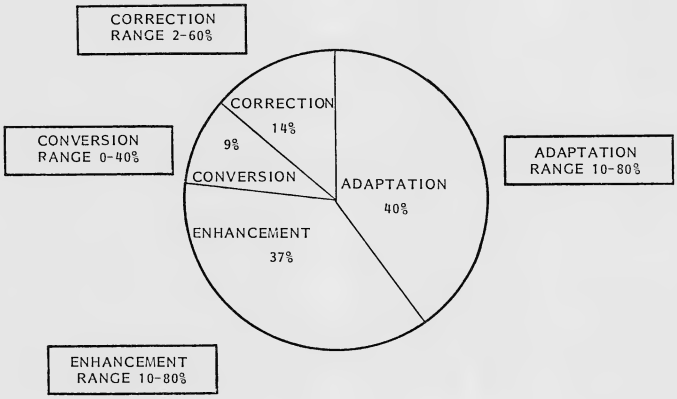
SOFTWARE COST ANALYSIS

FOR TYPICAL BUSINESS APPLICATION



SYSTEM LIFE CYCLE OF FIVE YEARS OR MORE

RESPONDENTS' ALLOCATION OF EFFORT WITHIN
SOFTWARE MAINTENANCE CYCLE



PERCENTAGE OF EFFORT

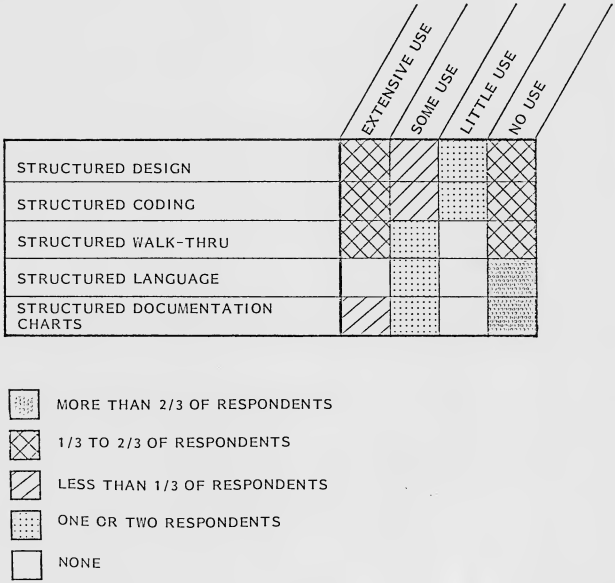
COMBINED RANKING OF NEED AND ABILITY TO

IMPROVE SOFTWARE DEVELOPMENT ASPECTS

ASPECT TO BE IMPROVED	RELATIVE PAYOFF OF IMPROVEMENTS*
DESIGN	VERY HIGH
PROGRAMMING	LOW-MEDIUM
TESTING	MEDIUM-HIGH
DOCUMENTATION	MEDIUM
USER INTERFACE	HIGH
PROJECT CONTROL	MEDIUM-HIGH
MANAGEMENT INVOLVEMENT	MEDIUM

*BASED ON RESPONDENTS' RATING OF NEED FOR IMPROVEMENTS COMBINED WITH THEIR RATING OF ABILITY TO MAKE IMPROVEMENTS.

RESPONDENTS' USE OF STRUCTURED TECHNIQUES



STRUCTURED DESIGN

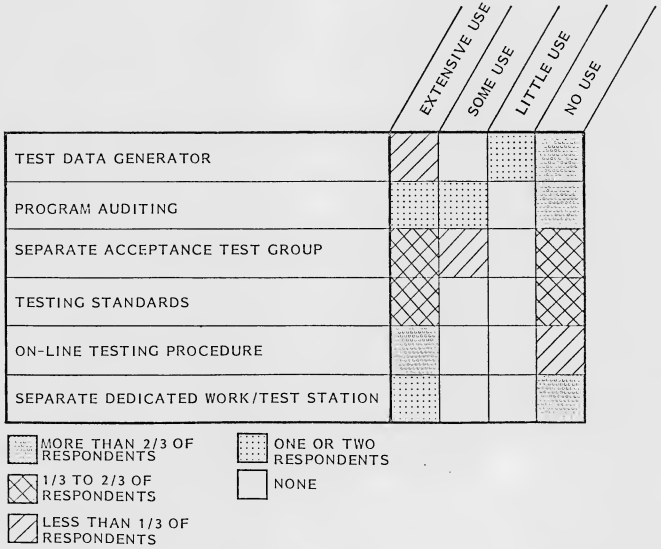
- MAJOR IMPROVEMENTS
  - PRODUCTIVITY
  - QUALITY
- IMPROVEMENTS
  - TESTING
  - FLEXIBILITY
  - DOCUMENTATION
- COSTS THE SAME OR MORE

PROGRAMMING IMPROVEMENT TECHNIQUES

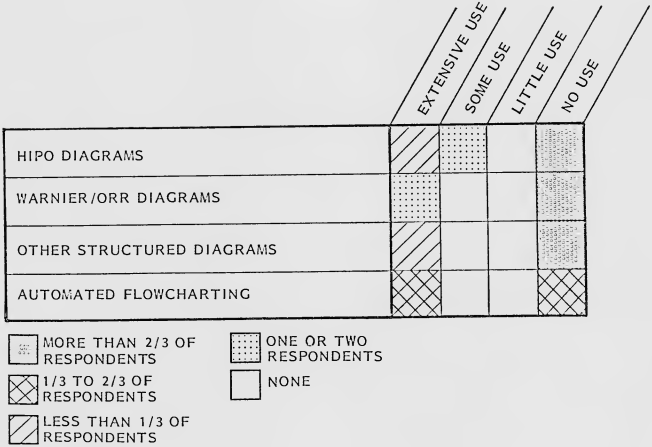
- HIGH USE
  - STANDARDS
  - ON-LINE DEVELOPMENT

- MEDIUM USE
  - AUTOMATED SOURCE PROGRAM CONTROL
  - STRUCTURED CODING
  - TOP-DOWN IMPLEMENTATION
  - STRUCTURED WALK-THRU
  - PROGRAM OPTIMIZERS/PRE-PROCESSORS

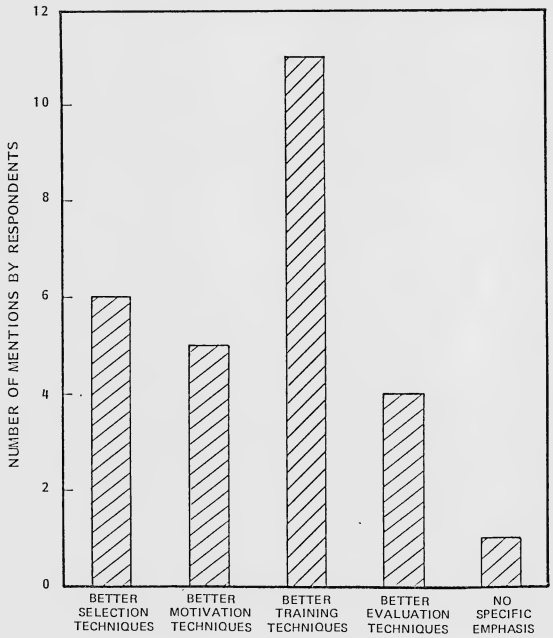
RESPONDENTS' USE OF TESTING
IMPROVEMENT TECHNIQUES



RESPONDENTS' USE OF DOCUMENTATION AIDS



RESPONDENTS' RANKING OF PERSONNEL
IMPROVEMENT STRATEGIES



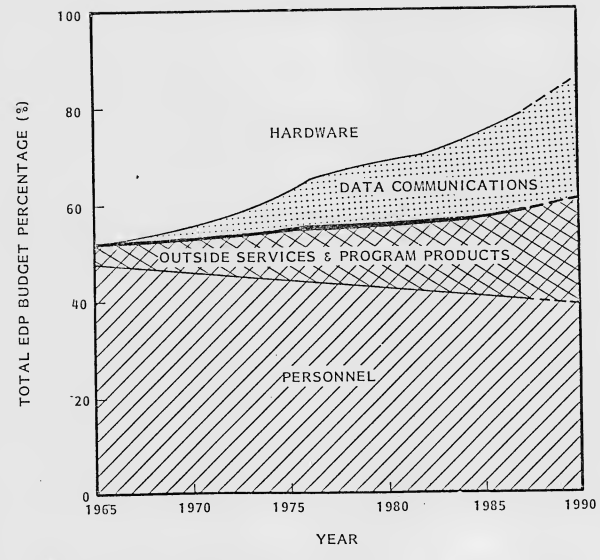
MEASUREMENT OF PROGRAMMING PRODUCTIVITY

- LINES OF CODE PER PROGRAMMER DAY
- SCHEDULED DUE DATE
- COMPARISON TO PREVIOUS PROJECT OF SIMILAR COMPLEXITY
- OPERATIONAL EFFICIENCY OF FINAL PRODUCT
- SUBJECTIVE EVALUATION
- COMBINATION OF STATISTICAL TECHNIQUES
- LENGTH OF TIME REQUIRED FOR PROGRAM CHANGES TO STABILIZE
- QUALITATIVE EVALUATION
- OVERALL INVESTMENT (PERSONNEL COSTS PLUS MACHINE TIME)
- EASE OF MAINTENANCE OF PRODUCT

TRENDS

SHIFT IN COST RATIO BETWEEN MAJOR

EDP BUDGET CATEGORIES



CONCLUSION

RECOMMENDATIONS

- VISIBLE, STRONG, MANAGEMENT SUPPORT
- INVOLVE THE USERS
- INVEST IN TRAINING
- INSTALL PROJECT MANAGEMENT MECHANISMS
- INSTALL PERFORMANCE EVALUATION MECHANISMS
- MOVE EMPHASIS TO FRONT-END ANALYSIS AND DESIGN

RECOMMENDATIONS

- CATEGORIZE SYSTEMS
- DESIGN SYSTEMS FOR CHANGE
- EMPHASIZE DESIGN IMPROVEMENTS
- DON'T DESIGN "PERFECT" SYSTEMS
- START TESTING IN DESIGN
- PUT DESIGN AND DEVELOPMENT IN PARALLEL

RECOMMENDATIONS

- USE DBMS AND IMPLEMENTATION LANGUAGES
- ON-LINE DEVELOPMENT (AND DESIGN)
- FIELD TEST AS ACCEPTANCE ONLY
- USE SOFTWARE PACKAGES
- TREAT SOFTWARE AS A PRODUCT NOT A UTILITY