NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DESIGN AND DEVELOPMENT OF AN EXPERT SYSTEM BASED QUALITY ASSURANCE MODULE FOR THE DYNAMO MODEL OF SOFTWARE PROJECT MANAGEMENT

by

Frank H. Leidy

March 1989

Thesis Advisor: Tarek K. Adbel-Hamid

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b RESTRICTIVE MARKINGS:

2a SECURITY CLASSIFICATION AUTHORITY:

2b DECLASSIFICATION / DOWNGRADING SCHEDULE:

3 DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S):

5 MONITORING ORGANIZATION REPORT NUMBER(S):

6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School

6b OFFICE SYMBOL *(If applicable)*: 54

7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*: Monterey, CA 93943-5000

7b ADDRESS *(City, State, and ZIP Code)*: Monterey, CA 93943-5000

8a NAME OF FUNDING / SPONSORING ORGANIZATION:

8b OFFICE SYMBOL *(If applicable)*:

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS *(City, State, and ZIP Code)*:

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE *(Include Security Classification)* DESIGN AND DEVELOPMENT OF AN EXPERT SYSTEM BASED QUALITY ASSURANCE MODULE FOR THE DYNAMO MODEL OF SOFTWARE PROJECT MANAGEMENT

12 PERSONAL AUTHOR(S) Leidy, Frank H.

13a TYPE OF REPORT: Master's Thesis

13b TIME COVERED: FROM ________ TO ________

14 DATE OF REPORT *(Year, Month, Day)*: 1989 March

15 PAGE COUNT: 88

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)* Artificial Intelligence, Expert Systems, Quality Assurance, Simulation

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)* Quality assurance is a crucial function to the successful development and maintenance of a software system. Because this activity has a significant impact on the cost of software development, the cost-effectiveness of quality assurance is a major concern to the software quality manager. There are tradeoffs between the economic benefits and costs of quality assurance.

Using the Dynamo model of software project management, an optimal quality assurance level and its distribution throughout a project's lifecycle can be identified. The focus of this thesis is to automated the process of identifying the optimum quality assurance level.

An expert system was developed that, when interfaced with the Dynamo model, will generate the optimum quality assurance distribution for a given set of parameters. The ability of the expert system to generated more cost-effective quality assurance levels than manually achievable was shown.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL: Tarek K. Abdel-Hamid

22b TELEPHONE *(Include Area Code)*: (408)646-2686

22c OFFICE SYMBOL: 54Ah

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office 1986—606-24.

Design and Development of an Expert System Based Quality Assurance Module for the Dynamo Model of Software Project Management.

by

Frank H. Leidy
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1989

ABSTRACT

Quality assurance is a crucial function to the successful development and maintenance of a software system. Because this activity has a significant impact on the cost of software development, the cost-effectiveness of quality assurance is a major concern to the software quality manager. There are tradeoffs between the economic benefits and costs of quality assurance.

Using the Dynamo model of software project management, an optimal quality assurance level and its distribution throughout a project´s lifecycle can be identified. The focus of this thesis is to automated the process of identifying the optimum quality assurance level.

An expert system was developed that, when interfaced with the Dynamo model, will generate the optimum quality assurance distribution for a given set of parameters. The ability of the expert system to generated more cost-effective quality assurance levels than manually achievable was shown.

TABLE OF CONTENTS

I. COUPLING EXPERT SYSTEMS AND SIMULATION

A. MOTIVATION

Management today is increasing in difficulty as the man-established systems of our society grow more complex. This complexity has as its basis the interrelations among the diverse elements of organizations and their interaction with other physical systems. Very often, changing one aspect of a system will produce changes or create the need for changes in other parts of the system. Since the arrival of electronic computers, one of the most useful tools for analyzing the design and operation of complex systems is simulation.

> Simulation is the process of designing a model of a real system for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system. [Ref. 1:pp. 1-2]

Effective computer simulation modeling requires the analysis of decisions that managers make along with the types of data and systematic methods that are most valuable in achieving those decisions. A mathematical model of a situation will often not be enough since qualitative reasoning is needed to understand the quantitative significance of the many problem variables. Therefore, in order to solve complicated problems, a solution which can continually shift between formal analysis and qualitative reasoning is required. [Ref. 2:p. 273]

Artificial intelligence (AI) is involved with designing computer systems that can imitate human thought for certain limited areas. Some of the human thought characteristics include the ability to learn, reason, solve problems, and understand ordinary human language. [Ref. 3:p. 276]

Advances in AI and expert systems have allowed for the design and implementation of coupled systems that can efficiently and effectively interleave the qualitative and quantitative components. Modules that relate to decision making processes can be modeled by an expert system, while modules that are associated with physical processes can be designed using a dynamic simulation model [Ref. 2:p. 273].

Simulation models accommodate many of the ideas being used in AI. Despite these similarities, there are important differences between the simulation modeling that is being done today and the AI-based (expert) simulation modeling approach. The primary difference is in the way the model is built and run. Simulation is an iterative process of designing the model, deciding on the inputs, running the experiment, analyzing the results, deciding on new inputs, running the experiment, and so on. Simulation modelers must convert the operation of a system into a set of numeric inputs and algorithms that are executed in sequential order. With an expert simulation system, the modeler asserts the knowledge about the system, defines the goals, and lets the computer find the solution. Figure 1-1 delineates some of the other differences between a simulation model and an expert simulation system. [Ref. 3:p. 278]

Simulation Model	Expert Simulation System
Primarily numeric	Have many symbolic processes
Algorithmic (solution steps explicit)	Use pattern invoked search (solution steps not explicit)
Integrated information and control	Command structure separate from knowledge domain
Several steps done outside of the model	All expertise possible built into the model to minimize user decisions
Model cannot do anything that is not preplanned	Model can learn from its own experience and modify itself as needed

Figure 1-1. Differences between Simulation Models and Expert Simulation Systems

B. EXPERT SYSTEM AND SIMULATION MODES

There are various applications of expert systems in simulation that can be particularly worthwhile. The first and most obvious way the two can be combined is by embedding an expert system into a simulation model, or vice versa. This is shown in figure 1-2. Many simulation models already use some knowledge, in addition to data. Embedding a simulation model with an expert system would allow the expert system to use time-dependent variables whose values are updated by the simulation. [Ref. 4:pp. 11-12]

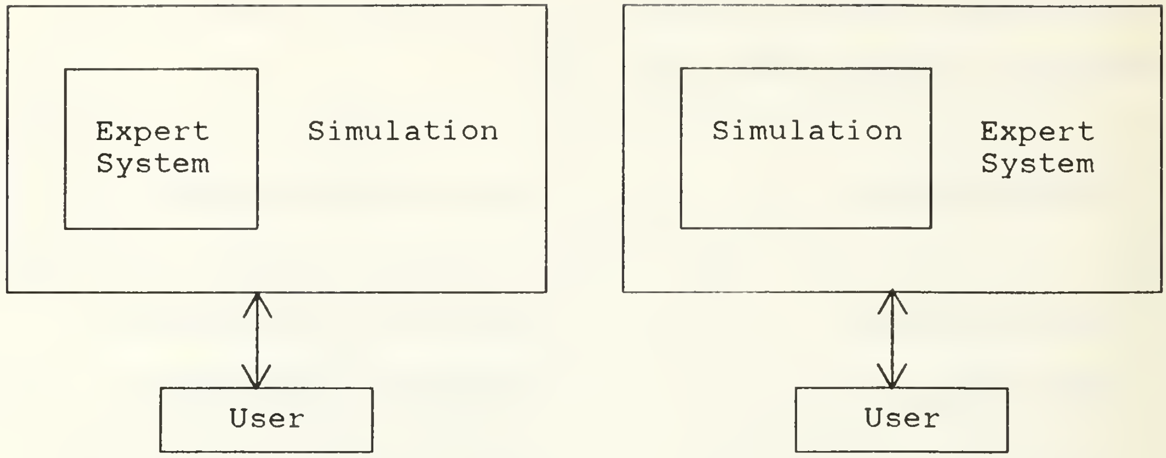


Figure 1-2. Embedded Expert System/Simulation Model

Expert systems and simulation models that are designed, developed, and implemented separately, may interact. An expert system can be used as an interface between the user and a simulation model, or the simulation model can serve as the interface between the user and an expert system. Figure 1-3 displays these two configurations. This thesis uses an expert system module to interrogate a project management simulation model iteratively until an optimal result is obtained. [Ref. 4:pp. 11-12]

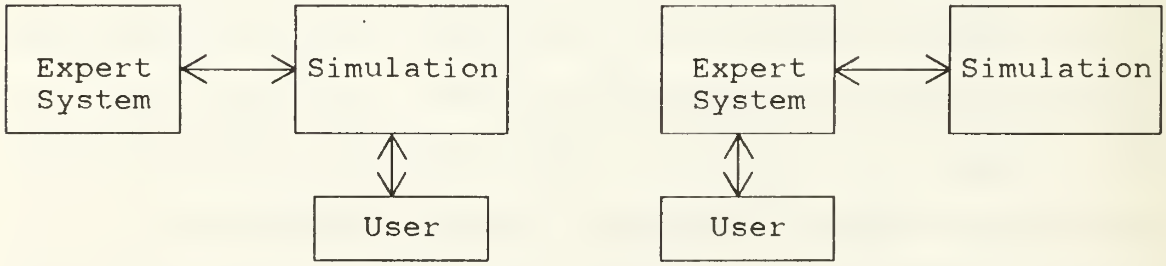


Figure 1-3. Parallel Expert System and Simulation Model

In some cases, the user will need to have access to both the expert system and the simulation model. Both will be used together, sharing data, and cooperating to accomplish some task. Figure 1-4 shows two examples of this configuration. This type of cooperation would be used when an expert system is used as an advice-giving tool to help the user with the operation or development of a simulation model. With the increasing trend of handing over simulation models to inexperienced end users, this would be especially advantageous to support the use, and guard against the misuse of such systems. [Ref. 4:p. 12]

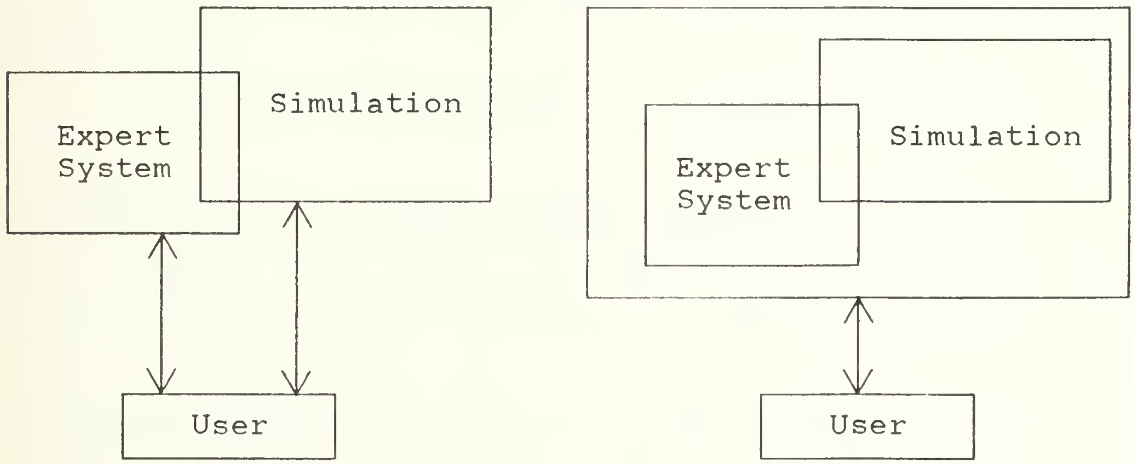


Figure 1-4. Cooperative Expert System/Simulation Model

The final application of the coupling of an expert system with a simulation model is that of using the expert system as an Intelligent Front End for an existing simulation model. This expert system would fit between a simulation model and its user, generating the necessary instructions or code to use the model after an exchange with the user. It would also interpret and explain any results from the model to the user. Figure 1-5 displays this architecture. [Ref. 4:p. 12]

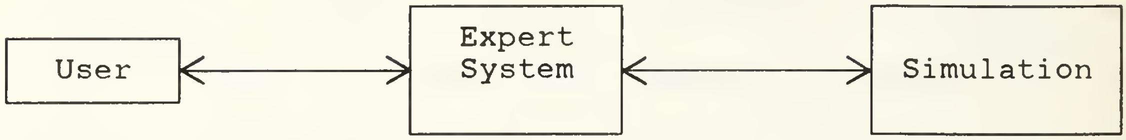


Figure 1-5. Intelligent Front End

C. OBJECTIVE OF THESIS

The objective of this thesis is to research the significance of implementing a decision making process with an expert system and a simulation model that were developed in parallel. The specific decision making process is that of choosing the optimal distribution of quality assurance effort to be expended throughout the lifecycle of a software project.

A comprehensive system dynamics model of the software development process has been developed that can serve as an experimentation vehicle for quality assurance policy. The model shows that the level of quality assurance expenditure has a significant impact on a project´s total cost. Using the model, an optimal quality assurance level and its distribution throughout a project´s lifecycle can be identified. This identification of the optimum level of quality assurance is presently done manually. This thesis will attempt to automate the process by interfacing an expert system with the software development model.

II. THE DYNAMO MODEL OF SOFTWARE PROJECT MANAGEMENT

A. OVERVIEW

Over the past 20 years there has been extraordinary growth in the demand for software systems. In recent years, rapid technological advancements in computer hardware, and the succeeding reduction in equipment cost, has increased the demand for hardware, resulting in an increase in the demand for software. Unfortunately, the software development process has become known for its problems of cost overruns, late deliveries, poor reliability and users' dissatisfaction. [Ref. 5:p. 1]

There are many variables that effect the software development process. These variables are not independent, but have complex relationships to one another. Understanding the operation of such a system is too complex for human intuition. [Ref. 5:pp. 6-7]

The Dynamo Model of Software Project Management is a comprehensive model of the software development process. The model is written in Professional Dynamo, a continuous simulation language developed in the late 1950's at the Sloan School of Management at M.I.T.. It can perform several important functions, including its main goal of aiding the project manager in understanding the software development process. The manager can use the model to perform "what if" experiments and develop a more complete understanding of the interrelationships of software development variables.

This thesis will specifically conduct "what if" experiments using the model to determine the impact of a project's quality assurance effort on the total cost of the project. The model can track, store, graph and plot large amounts of project data quickly and efficiently, allowing the manager to enrich and fine tune his basic understanding of the software development process through the use of a computer simulation model [Ref. 5:pp. 7-8].

The model can also be used to help the software manager in the management of an actual software project. For example, the model can be used to estimate the total project cost, the schedule completion time, and a number of other factors. Variables such as the Fraction of Manpower Devoted to Quality Assurance (TPFMQA) can be changed and simulations can be run in a matter of minutes to determine the effects of the change. This capability allows the manager to objectively evaluate different management strategies. [Ref. 5:pp. 7-8]

The model combines the multiple functions of the software development process. It includes both the management-type operations of planning, control and staffing, along with the software production-type activities of design, coding, reviewing, and testing. This integrative technique is effective as it prompts and assists in identifying the multiple, and conceivably scattered, set of factors that are compounding to cause some software project problems. [Ref. 5:p. 7]

Another characteristic of the Dynamo model is the use of System Dynamics feedback principles to organize and explain the complex structure of dynamically interacting variables involved in the development and management of software projects. Feedback is the mechanism in which an action taken by an entity will ultimately affect that entity. [Ref. 5:p. 7]

A third feature of the Dynamo model is its use of the computer simulation tools of System Dynamics to deal with the highly complex integrative feedback model. Even though the dynamic effects of single feedback loops may be quite obvious, the actions of systems with interconnected feedback loops will often confuse human intuition. Because of the complexity of the feedback structures present in many real problems, a problem's behavior over time may only be traceable through the use of simulation techniques. [Ref. 5:pp. 7-8]

The Dynamo model is made up of four subsystems, namely (1) the Human Resource Management Subsystem; (2) the Software Production Subsystem; (3) the Controlling Subsystem; and (4) the Planning Subsystem. Figure 2-1 shows some of the interrelationships between these subsystems. [Ref. 5:p. 9]

The Human Resource Management Subsystem deals with the hiring, training, assimilation, and transfer of a project´s human resources. The workforce is segregated into two categories of employees, "NEWLY HIRED WORKFORCE" and "EXPERIENCED WORKFORCE". This allows the model to compensate for the lower productivity of the less experienced workforce, as well as capturing the training processes involved in bringing new members into the project team. The Human Resource Management Subsystem provides workforce available to the Software Production Subsystem. [Ref. 5:pp. 9-11]

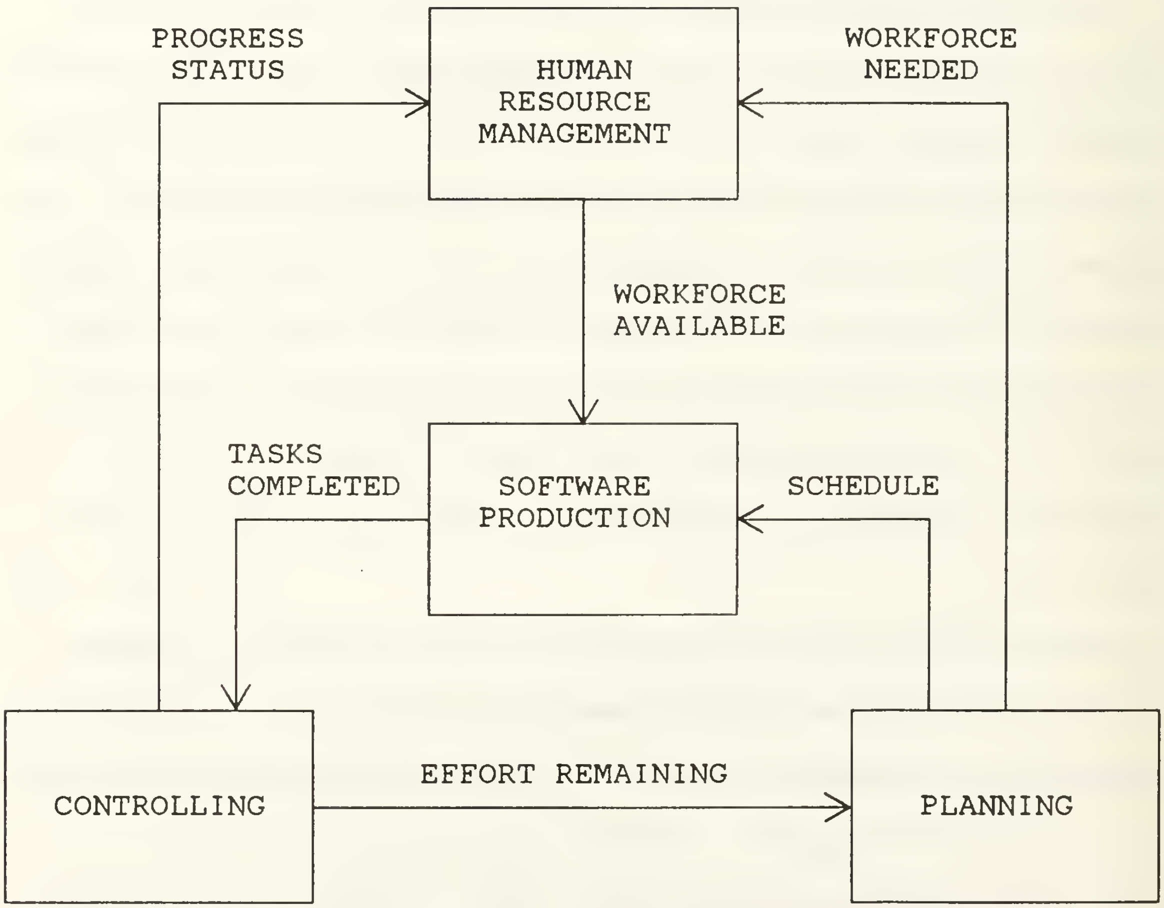


Figure 2-1. Four Subsystems of the Dynamo Model

The Software Production Subsystem deals with the development, quality assurance, rework, and testing activities. Development includes both the design and coding of software. As the software is developed, it is reviewed to detect errors. Errors that are detected through these quality assurance activities are then reworked. Some errors will not be detected and reworked until the testing phase. The Software Production Subsystem provides the Controlling Subsystem with the tasks completed. [Ref. 5:pp. 12-13]

The Planning Subsystem makes the initial project estimates for variables such as completion time, staffing load, and total man-days. As the project proceeds through its lifecycle, these estimates are revised. The Planning Subsystem provides the level of the workforce needed to the Human Resource Management Subsystem and the schedule to the Software Production Subsystem. [Ref. 5:p. 21]

Finally, the Controlling Subsystem measures the progress on the project. This subsystem provides effort remaining to the Planning Subsystem and progress status to the Human Resources Management Subsystem. [Ref. 5:p.17]

B. THE QUALITY ASSURANCE SECTOR

Quality assurance (QA) is handled by the Software Production Subsystem of the Dynamo Model of Software Project Management. The QA distribution is defined in the model by the FRACTION OF MANPOWER DEVOTED TO QUALITY ASSURANCE (TPFMQA). TPFMQA is the percent of development effort allocated in the project's plan for QA activity during the design and coding stages. Figure 2-2 shows how the eleven TPFMQA values correspond to the percent of tasks developed for a given project. A sample initial uniform distribution of quality assurance is also shown in Figure 2-2.

TPFMQA POINT DEFINITION											
% Tasks Dev't	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Point #	1	2	3	4	5	6	7	8	9	10	11
sample TPFMQA	.15	.15	.15	.15	.15	.15	.15	.15	.15	.15	0

Figure 2-2. Definition of TPFMQA points

Obviously the uniform distribution is not necessarily the most cost effective distribution.

> To identify a more cost effective distribution, we started with the 15% uniformly distributed policy and searched for areas in the project's lifecycle where such a level is not cost effective. This was done by conducting simulation runs to test the impact of negative impulses in the QA level, as the one shown in Figure 2-3. [Ref. 6:p. 407]

The magnitude of the pulse is a set percentage of the TPFMQA value (the percentage is entered by the user). If the result of this pulse is a decrease in the project's total cost, then it would indicate that the original QA value was too high. If the result of the pulse is an increase then a positive pulse at that point will be generated. If this results in a lower project cost then the original QA value was too low. If both a negative and positive pulse result in higher project costs then the original QA value at this point

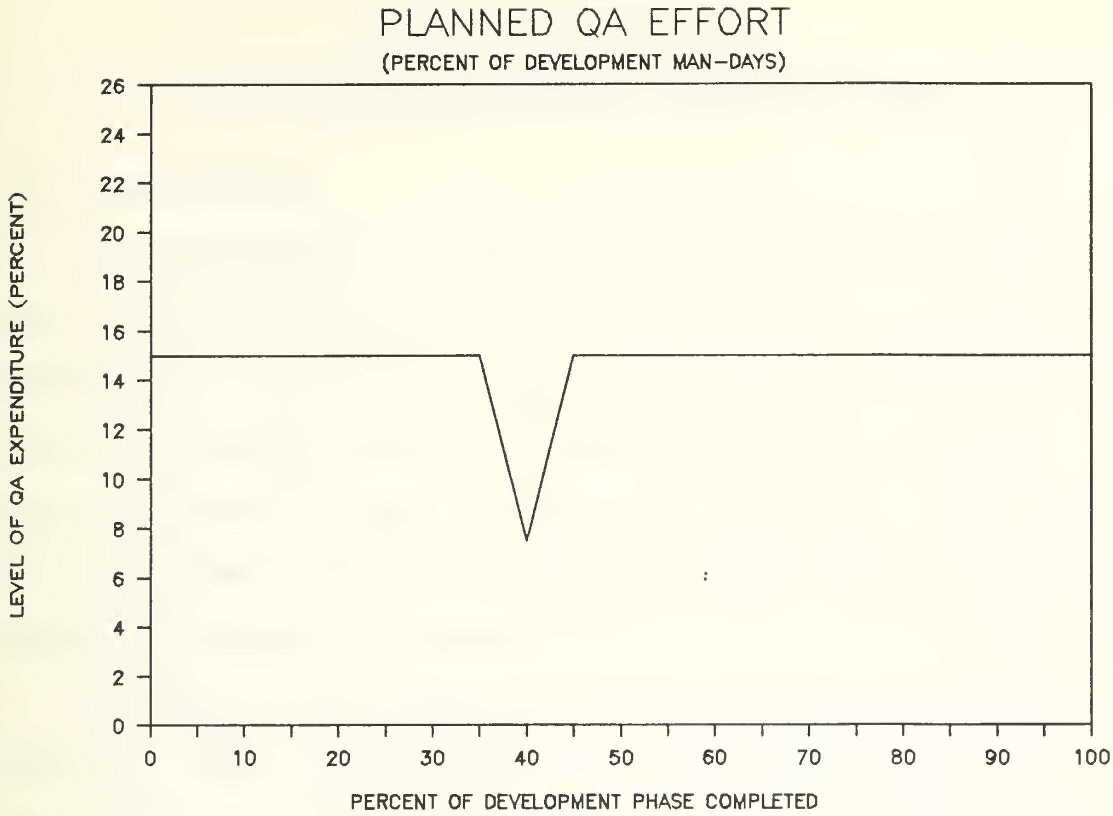


Figure 2-3. Example Negative Impulse.

is maintained. After the lowest cost alternative (negative pulse, positive pulse, or no pulse), at one point is found, a negative pulse is introduced to the next point. This procedure is repeated at each of the first ten TPFMQA points in one cycle.

> Figure 2-4 summarizes the results obtained from a series of simulation runs in which negative impulses were applied at different stages of the lifecycle. The results show that the simplistic uniform distribution policy under-spends in the early phases of the project and over-spends in the middle and final stages. By reiterating through the above experimentation strategy we can use the model to derive a more cost-effective QA distribution. [Ref. 6:p. 407]

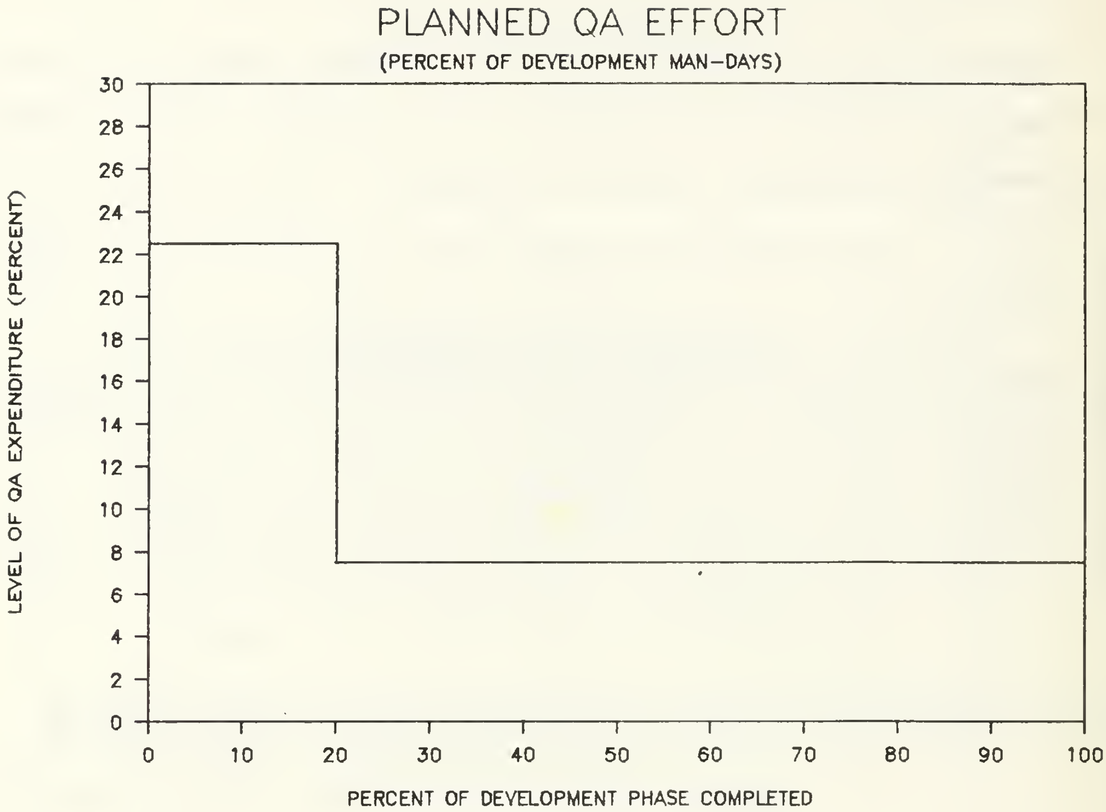


Figure 2-4. TPFMQA Distribution After One Cycle.

III. THE EXPERT SYSTEM SIMULATION MODEL

A. INTRODUCTION

This chapter will provide a detailed description of the expert system simulation model. Explanatory information is provided for each of the rules in the Prolog program as well as all of the files that are required for the Prolog to Dynamo interface. This chapter will serve as a users manual for the system, providing guidance on operating the system and explaining how it works.

The expert system simulation model consists of one Prolog program, a Dynamo model, and various other system files required for the interface between Prolog and Dynamo. First, the system architecture will be examined. Then each of the 15 Prolog rules of the expert system module will be explained. Finally, initialization procedures will be discussed.

Operation of the expert system simulation model requires an IBM or IBM compatible AT microcomputer with 640 K of RAM and a hard disk. Microsoft (MS) or Personal Computer (PC) DOS 2.xx or greater is also required.

B. THE EXPERT SYSTEM SIMULATION MODEL ARCHITECTURE

The main components of the expert system simulation model architecture and the execution flow through the system are shown in figure 3-1. The following files were created for this architecture:

PQA.ARI
SUMMARY.DAT
PROJECT.DYN
PROJECT.DNX
PROJECT.OUT
PROJECT.DRS
FRANK.BAT

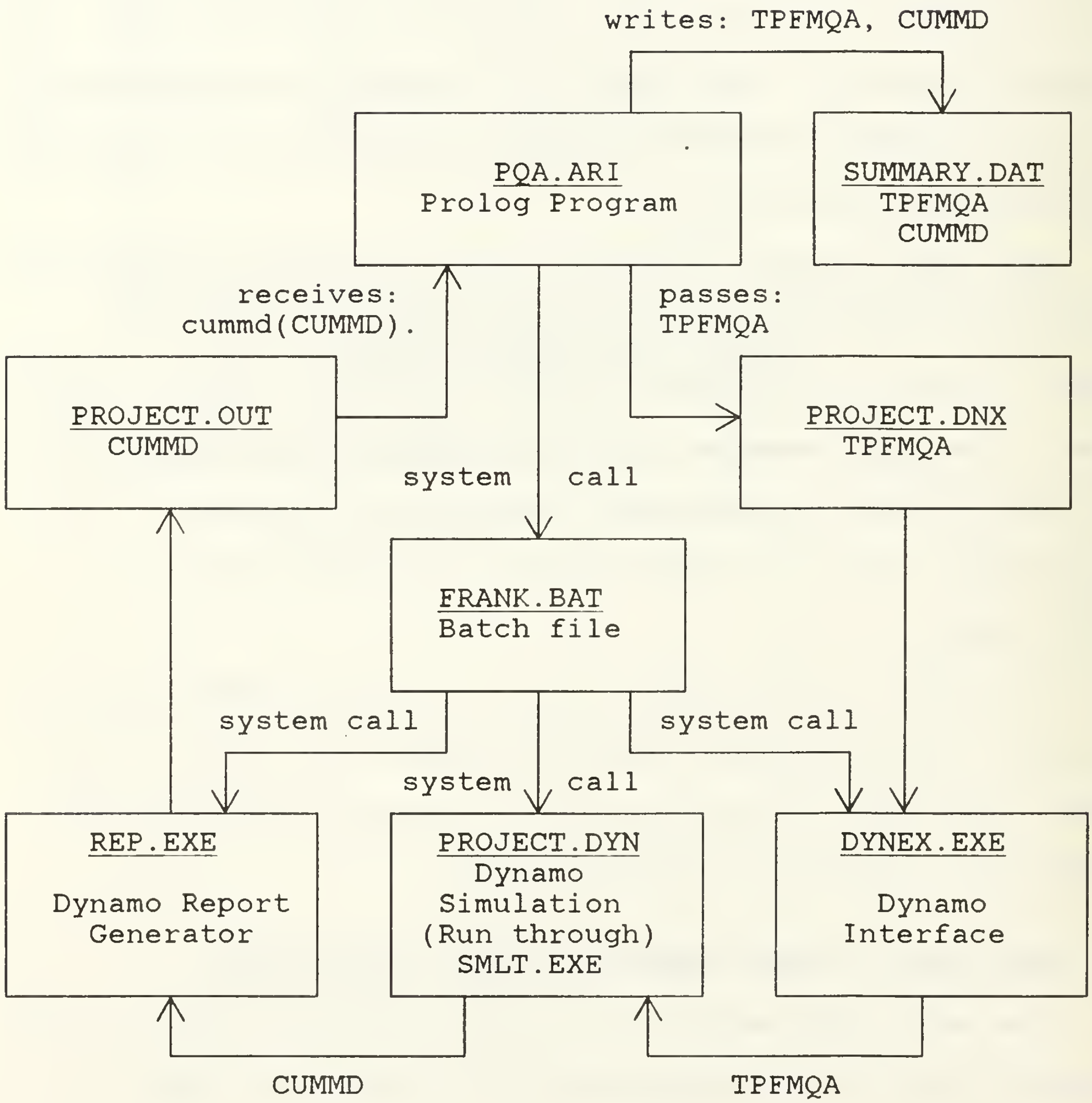


Figure 3-1. The Expert System Simulation Model Architecture

PQA.ARI, which is the Prolog program that controls the execution of the expert system simulation model, will be discussed in detail in Chapter III Section C. The rest of these files will be explained in this section.

1. SUMMARY.DAT

This file is used to collect a record of each of the TPFMQA tables used during each run of the expert system simulation model as well as the cumulative man day total associated with each set of TPFMQA values. The cumulative man day total appears directly below its TPFMQA table. There are also entries made in this file for the five initial input parameters, namely, pulse size factor, maximum number of cycles, exit condition, minimum QA value and the original TPFMQA table. Finally, the best cumulative man day total is repeated at the end. Figure 3-2 is an example of the contents of SUMMARY.DAT.

```
       Pulse size factor = 0.15
Maximum number of cycles = 30
          Exit condition = 0.0000001
        Minimum QA value = 0.03

TPFMQA=0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
1. CUMMD=1656.71

****************   Start of a new cycle   ****************

TPFMQA=0.128/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
1. CUMMD=1707.37
TPFMQA=0.173/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
2. CUMMD=1616.99

The best CUMMD is: 1616.99
```

Figure 3-2. Sample Contents of SUMMARY.DAT

2. PROJECT.DYN

This file is the dynamo simulation model. This model is discussed in Chapter II.

3. PROJECT.DNX

This file is recreated by PQA.ARI every time we run the simulation, i.e. with a different TPFMQA value. It contains one line in the following format:

```
T TPFMQA=0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0
```

The Dynex Model Interface uses this file to determine the applicable TPFMQA values for the simulation.

4. PROJECT.DRS

When the simulation for a particular TPFMQA value is finished, the Dynamo Report Generator uses this file to format its output. In this case the only output of interest is the value of the cumulative man days. This needs to be stored in a format suitable for the Prolog Program PQA.ARI to read it (i.e., as: cummd(1515.66).). PROJECT.DRS is listed below:

```
##############################################################
REPORT
TIME=MAXTIME,
FORMAT="1<,15>,16<",PICTURE="ZZZZZZV.99"
"cummd(",CUMMD,")."
##############################################################
```

5. PROJECT.OUT

This is the output file actually created by the Dynamo Report Generator (on the basis of the format specified by PROJECT.DRS). It contains the cumulative man days total for the last run of the simulation in the format of a Prolog fact (cummd(1515.66).).

6. FRANK.BAT

This is a DOS batch file that executes the Dynex Interface, the Dynamo Simulator, and the Dynamo Report Generator. It is called by PQA.ARI to initiate a simulation run every time a change is made to TPFMQA, i.e., a new pulse (either negative or positive) is generated. When the batch file is finished executing, control is automatically passed back to PQA.ARI. A listing of FRANK.BAT is shown below:

```
############################################################
DYNEX PROJECT -d project.drs
IF ERRORLEVEL 4 GOTO ERROR
SMLT PROJECT -GO = -DTM =
REP PROJECT -T
GOTO EXIT
:ERROR
ECHO *** ERROR 1 ****
:EXIT
############################################################
```

7. OTHER FILES

In addition to the above files, the following executable files are required to run the expert system simulation model:

API.EXE (Arity Prolog Interpreter)
DYNEX.EXE (Dynamo Model Interface)
SMLT.EXE (Dynamo simulator)
REP.EXE (Dynamo Report Generator)

C. THE EXPERT SYSTEM MODULE

The expert system module, i.e. PQA.ARI, is designed to yield the optimal quality assurance distribution for a given software project. The quality assurance distribution is defined in the Dynamo Model by the FRACTION OF MANPOWER DEVOTED TO QUALITY ASSURANCE (TPFMQA). TPFMQA is the percent of development effort allocated in the project´s plan for QA activity during the design and coding stages.

Though it is common practice, a uniformly distributed QA effort is not necessarily the most cost effective.

> To identify a more cost effective distribution, we started with the 15% uniformly distributed policy and searched for areas in the project´s lifecycle where such a level is not cost effective. This was done by conducting simulation runs to test the impact of negative impulses in the QA level. [Ref. 6:p. 407]

The magnitude of the pulse is a set percentage of the TPFMQA value (the percentage is entered by the user). If the result of this pulse is a decrease in the project´s total cost, then it would indicate that the original QA value was too high. If the result of the pulse is an increase then a positive pulse at that point will be generated. If this results in a lower project cost then the original QA value was too low. If both a negative and positive pulse result in higher project costs then the original QA value at this point is maintained. After the lowest cost alternative (negative pulse, positive pulse, or no pulse), at one point is found, a negative pulse is introduced to the next point. This procedure is repeated at each of the first ten TPFMQA points in one cycle. The user can select to put a ceiling on the total number of cycles to be evaluated.

PQA.ARI is written in Arity Prolog and contains 15 rules. The remainder of this section will explain each of the rules. A listing of the rule will follow the explanation. The full listing of the program is provided in Appendix A.

Several Arity Prolog predicates appear often in the code listings. These predicates are listed and explained below [Ref. 7:pp. 271-290]:

asserta(Clause)	Adds a clause to the beginning of a database
call(goal(X))	Looks for goal(X) in the database. X will be assigned the value of the first case of the goal predicate encountered.
create(F,´file´)	Creates ´file´ and opens it as file F to write.
ifthen(P,Q)	If P is true, then execute Q.
ifthenelse(P,Q,R)	If P is true, then execute Q, otherwise execute R.
open(F,´file´,r)	Opens ´file´ as file F to read.
open(F,´file´,a)	Opens ´file´ as file F to append.
read(Term)	Reads term from the keyboard.
read(F,Term)	Reads term from the file opened as F.
retract(Clause)	Removes the clause from the database.
shell(DOScommand)	Executes the DOS command and returns to Prolog.
write(Term)	Writes term to the screen.
write(F,Term)	Writes term to the file opened as F.

1. Rule - pqa

This rule starts the system. First it prompts the user to input the five parameters: pulse size factor, maximum number of cycles, exit condition, minimum QA value, and the initial distribution. Then it calls the rules calc_zero, initial_run, and dopqa.

```
################################################################
pqa:-
   asserta(number(1)),
   asserta(calc(0)),
   write('What is your desired pulse size factor? '),
   read(PU),
   asserta(size(PU)),
   write('What is the maximum number of cycles? '),
   read(MX),
   asserta(cycle(MX)),
   asserta(newcycle(1)),
   write('What is the exit condition? '),
   read(EX),
   asserta(stop(EX)),
   write('What is the minimum QA value? '),
   read(MN),
   asserta(min(MN)),
   write('Enter the initial QA distribution.  Point 1 '),
   read(QA1),
   write('                                    Point 2 '),
```

```
  read(QA2),
  write('                                    Point 3 '),
  read(QA3),
  write('                                    Point 4 '),
  read(QA4),
  write('                                    Point 5 '),
  read(QA5),
  write('                                    Point 6 '),
  read(QA6),
  write('                                    Point 7 '),
  read(QA7),
  write('                                    Point 8 '),
  read(QA8),
  write('                                    Point 9 '),
  read(QA9),
  write('                                    Point 10 '),
  read(QA10),
  write('                                    Point 11 '),
  read(QA11),
  calc_zero(PU,MX,EX,MN),
  initial_run(QA1,QA2,QA3,QA4,QA5,QA6,QA7,QA8,QA9,QA10,
              QA11),
  dopqa.
/* end of pqa */
##############################################################
```

2. Rule - dopqa

This rule is a repeat-fail loop. This loop is needed to allow the expert system simulation model to move from one TPFMQA value to the next. When the repeat clause is encountered, it is executed. This clause always succeeds. The program will continue through the clauses that follow until it comes to one that fails. Then the program will backtrack to the repeat predicate and execute the clauses again. In this rule, rule main is called and always succeeds. Then the clause "fail" is encountered. Because this clause will never succeed, the program will backtrack to the repeat predicate indefinitely. Program execution is halted through rules quit_test and no_calc.

```
##############################################################
dopqa:-
   repeat,
   main,
   fail.
/* end of dopqa */
#############################################################
```

3. Rule - main

This rule first reads the position for the next negative pulse (ITER) and whether the last pulse was positive or negative (TYPE = 0 for negative, TYPE = 1 for positive). Using ITER, the previous position (PREV) is calculated. Then the man days total for the previous position (CHECK), and for the current position in the last cycle (OLD) are read from the table of the last 10 cumulative man day totals (cummdold). The cumulative man days total for the current TPFMQA table is then read by rule read_cummd and its value is assigned to the variable NEW. Based on the relationship between NEW and CHECK, the rule will next call another rule to generate either a negative (calc_less) or a positive pulse (calc_more). Finally, the current man days total is removed from the database.

```
############################################################
main:-
/* get the x value for the pulse */
   call(number(ITER)),
   call(calc(TYPE)),
/* get the previous man days */
   PREV is [[[ITER + 8] mod 10] + 1],
   call(cummdold(PREV,CHECK)),
   call(cummdold(ITER,OLD)),
/* get the man days from the last QA numbers */
   read_cummd,
   call(cummd(NEW)),
/* calculate the new y value (QA) for the current x value */
   case([NEW =< CHECK -> calc_less(ITER,NEW,OLD,TYPE),
         NEW > CHECK -> calc_more(ITER,NEW,OLD,CHECK,TYPE)]),
   retract(cummd(NEW)),
   statistics,
```

```
   write(ITER).
/* end of main */
```

###

4. Rule - quit_test

This rule is called from rule calc_less whenever a new cycle begins. It checks to see if either of the exit criteria have been met. First, it compares OLD and NEW (described above) to see if the percent change is less than the number entered in response to the exit condition question at the start of the system. Next it checks to see if the maximum number of cycles has been reached. If either of these criteria succeeds then rule no_calc will be called, otherwise rule calc_less will proceed.

###

```
quit_test(NEW):-
/* get final man days of last cycle */
   call(cummdold(1,OLD)),
/* test if exit condition exits */
   call(stop(EXIT)),
   ifthenelse(OLD=:=0,TEST is 100,TEST is abs(OLD-NEW)/OLD),
   ifthen(TEST < EXIT,no_calc(NEW)),
   call(cycle(MAX)),
   call(newcycle(NOW)),
   NEXT is NOW + 1,
   ifthen(MAX < NOW,no_calc(NEW)),
   retract(newcycle(NOW)),
```

```
   asserta(newcycle(NEXT)).
/* end of quit_test */
################################################################
```

5. Rule - calc_zero

This rule creates the DOS text file SUMMARY.DAT which contains a complete listing of each TPFMQA table that has been generated, along with the associated cumulative man days cost for these TPFMQA values. After its creation the first four initialization parameters (pulse size factor (PU), maximum number of cycles (MX), exit condition (EX) and minimum QA (MN)) are written to the file. An example of the output is:

```
       Pulse size factor = 0.25
Maximum number of cycles = 30
          Exit condition = 0.00001
        Minimum QA value = 0.03
```

```
################################################################
calc_zero(PU,MX,EX,MN):-
   create(S,'summary.dat'),
   write(S,'       Pulse size factor = '),write(S,PU),nl(S),
   write(S,'Maximum number of cycles = '),write(S,MX),nl(S),
   write(S,'          Exit condition = '),write(S,EX),nl(S),
   write(S,'        Minimum QA value = '),
   write(S,MN),nl(S),nl(S),
   close(S).
/* end of calc_zero */
################################################################
```

6. Rule - calc_less

This rule is used to generate a negative pulse at the current position. First, at the beginning of each cycle (ITER = 1), it calls rule exit_test to see if the exit criteria have been met. Next the QA values for the current position (QA) and for position number 10 (DIV) are read along with the pulse size factor (PULSE). Then the negative pulse (NEWQA) is calculated. NEWQA is checked against the minimum QA value (MINQA) and the greater of MINQA and NEWQA replaces the current QA value. The table of old QA values is updated as is the position for the next negative pulse (number(ITER)). Finally the table of QA values (TPFMQA) is written to file PROJECT.DNX and the batch file (FRANK.BAT) to run the dynamo model is called.

```
##############################################################
calc_less(ITER,NEW,OLD,TYPE):-
   ifthen(ITER =:= 1,quit_test(NEW)),
/* read the current QA values */
   call(tpfmqa(ITER,QA)),
   call(tpfmqa(10,DIV)),
   call(size(PULSE)),
/* calculate the new QA value (NEWQA) */
   NEWQA is round([QA-[PULSE*DIV]],3),
/* check if the new QA is less than the minimum (MINQA) */
   call(min(MINQA)),
   retract(tpfmqa(ITER,QA)),
```

```
   ifthenelse(NEWQA < MINQA,asserta(tpfmqa(ITER,MINQA)),
               asserta(tpfmqa(ITER,NEWQA))),
/* add man days for this cycle */
   retract(cummdold(ITER,OLD)),
   asserta(cummdold(ITER,NEW)),
/* move to the next position */
   retract(number(ITER)),
   NEWITER is [ITER mod 10] + 1,
   asserta(number(NEWITER)),
   output_cummd(NEW,ITER),
   ifthen(ITER =:= 1,output_break),
   output_tpfmqa,
   shell(frank),
/* save fact that last pulse was negative */
   retract(calc(TYPE)),
   asserta(calc(0)).
/* end of calc_less */
#############################################################
```

7. Rule - calc_more

If the result of rule calc_less produces a more costly project, then calc_more will be called. This rule will generate a positive pulse of the same magnitude as the negative pulse which led to a higher cost. If this positive pulse also results in a worse case, then the QA value will be returned to its original value for this cycle. This rule is very similar to rule calc_less in the way it works.

```
###############################################################
calc_more(ITER,NEW,OLD,CHECK,TYPE):-
/* reset the position back 1 */
   NEWITER is [[[ITER + 8] mod 10] + 1],
   call(tpfmqa(NEWITER,QA)),
   call(tpfmqa(10,DIV)),
   call(size(PULSE)),
/* calculate new QA value depending on whether the last */
/* pulse was negative (TYPE = 0) or positive (TYPE = 1) */
   case([TYPE =:= 0 -> NEWQA is round(QA+[2*PULSE*DIV],3),
         TYPE =:= 1 -> NEWQA is round(QA-[PULSE*DIV],3)]),
/* check if the new QA is less than the minimum (MINQA) */
/* add the new QA value to the database */
   call(min(MINQA)),
   retract(tpfmqa(NEWITER,QA)),
   ifthenelse(NEWQA < MINQA,asserta(tpfmqa(NEWITER,MINQA)),
              asserta(tpfmqa(NEWITER,NEWQA))),
   retract(calc(TYPE)),
/* reset the type of calculation */
   case([TYPE=:=0-> calc_up(NEW,NEWITER),
         TYPE=:=1-> calc_orig(NEWITER,ITER,NEW,OLD,CHECK)]).
/* end of calc_more */
###############################################################
```

8. Rule - calc_up

Rule calc_up is called by calc_more to run the dynamo model after a positive pulse is made. First the fact that a positive pulse has been made is added to the database, then the TPFMQA values are output to file PROJECT.DNX. And finally, the batch file (frank) is called to run the dynamo simulation.

```
################################################################
calc_up(NEW,NEWITER):-
   asserta(calc(1)),
   output_cummd(NEW,NEWITER),
   output_tpfmqa,
   shell(frank).
/* end of calc_up */
################################################################
```

9. Rule - calc_orig

This rule is called by calc_more after a QA value has been returned to its original value for that cycle. It writes an entry in SUMMARY.DAT then calls calc_less to generate a negative pulse at the next position.

```
################################################################
calc_orig(NEWITER,ITER,NEW,OLD,CHECK):-
   asserta(calc(0)),
   output_cummd(NEW,NEWITER),
   output_tpfmqa,
   calc_less(ITER,CHECK,OLD,0).
/* end of calc_orig */
################################################################
```

10. Rule - no_calc

This rule is called by rule quit_test when either of the exit criteria is met. The exit criteria will be met when the maximum number of cycles has been reached or when the percent change in cumulative man days for the last cycle is less than the exit condition entered in response to the third initialization question. It writes the best cumulative man days total to file SUMMARY.DAT then halts the system, returning to DOS.

```
###############################################################
no_calc(NEW):-
   call(cummdold(10,BEST)),
   open(S,´summary.dat´,a),
   write(S,´10. CUMMD=´),write(S,NEW),nl(S),nl(S),
   write(S,´The best CUMMD is: ´),
   ifthenelse(BEST < NEW,write(S,BEST),write(S,NEW)),
   nl(S),
   close(S),
   halt.
/* end of no_calc */
###############################################################
```

11. Rule - output_tpfmqa

This rule is used to output the TPFMQA table to two DOS files, PROJECT.DNX and SUMMARY.DAT. First it reads the TPFMQA table, then it creates file PROJECT.DNX. Then it writes the current TPFMQA values into this file. Next it updates file SUMMARY.DAT with the cumulative man days total for the previous TPFMQA values in addition to the current TPFMQA values.

```
##############################################################
output_tpfmqa(NEW,ITER):-
   call(tpfmqa(1,QA1)),
   call(tpfmqa(2,QA2)),
   call(tpfmqa(3,QA3)),
   call(tpfmqa(4,QA4)),
   call(tpfmqa(5,QA5)),
   call(tpfmqa(6,QA6)),
   call(tpfmqa(7,QA7)),
   call(tpfmqa(8,QA8)),
   call(tpfmqa(9,QA9)),
   call(tpfmqa(10,QA10)),
   call(tpfmqa(11,QA11)),
   create(D,'project.dnx'),
   write(D,'T TPFMQA='),
   write(D,QA1),write(D,' '),
   write(D,QA2),write(D,' '),
   write(D,QA3),write(D,' '),
   write(D,QA4),write(D,' '),
   write(D,QA5),write(D,' '),
   write(D,QA6),write(D,' '),
   write(D,QA7),write(D,' '),
   write(D,QA8),write(D,' '),
```

```
   write(D,QA9),write(D,' '),
   write(D,QA10),write(D,' '),
   write(D,QA11),nl(D),
   close(D),
   open(S,'summary.dat',a),
   write(S,ITER),write(S,'. '),
   write(S,'CUMMD='),write(S,NEW),nl(S),
   write(S,'TPFMQA='),
   write(S,QA1),write(S,'/'),
   write(S,QA2),write(S,'/'),
   write(S,QA3),write(S,'/'),
   write(S,QA4),write(S,'/'),
   write(S,QA5),write(S,'/'),
   write(S,QA6),write(S,'/'),
   write(S,QA7),write(S,'/'),
   write(S,QA8),write(S,'/'),
   write(S,QA9),write(S,'/'),
   write(S,QA10),write(S,'/'),
   write(S,QA11),nl(S),
   close(S).
/* end of output_tpfmqa */
############################################################
```

12. Rule - output_cummd

This rule writes the current cumulative man days total to file SUMMARY.DAT.

```
############################################################
```

```
output_cummd(NEW,ITER):-
   open(S,'summary.dat',a),
   write(S,ITER),write(S,'. '),
   write(S,'CUMMD='),write(S,NEW),nl(S),
   close(S).
/* end of output_cummd */
################################################################
```

13. Rule - output_break

This rule places a statement in file SUMMARY.DAT denoting the beginning of each new cycle.

```
################################################################
output_break:-
   open(S,'summary.dat',a),
   nl(S),write(S,
   '**************   Start of a new cycle   **************'),
   nl(S),nl(S),
   close(S).
/* end of output_break */
################################################################
```

14. Rule - read_cummd

This rule reads the contents of file PROJECT.OUT (output file from the Dynamo model). The contents of this file will look like this:

```
cummd(1543.56).
```

This is in the format of a Prolog fact. Once read, the new fact is added to the current database by the asserta predicate.

```
####################################################################
read_cummd:-
   open(C,'project.out',r),
   read(C,CUMMD),
   asserta(CUMMD),
   close(C).
/* end of read_cummd */
####################################################################
```

15. Rule - initial_run

This rule provides the starting point for the system. First, it adds the uniformly distributed TPFMQA table to the database. The value for the uniform distribution was entered in response to the fifth initialization question. Next it writes the TPFMQA values to the file PROJECT.DNX and calls the batch file (frank) to run the Dynamo simulation with those values. Finally it reads the cumulative man days total from file PROJECT.OUT and adds it, and the rest of the last 10 cumulative man days totals (initially set to zero), to the database.

```
####################################################################
initial_run(QA1,QA2,QA3,QA4,QA5,QA6,QA7,QA8,QA9,QA10,QA11):-
   asserta(tpfmqa(1,QA1)),
   asserta(tpfmqa(2,QA2)),
   asserta(tpfmqa(3,QA3)),
```

```
   asserta(tpfmqa(4,QA4)),
   asserta(tpfmqa(5,QA5)),
   asserta(tpfmqa(6,QA6)),
   asserta(tpfmqa(7,QA7)),
   asserta(tpfmqa(8,QA8)),
   asserta(tpfmqa(9,QA9)),
   asserta(tpfmqa(10,QA10)),
   asserta(tpfmqa(11,QA11)),
   output_tpfmqa,
   shell(frank),
   read_cummd,
   call(cummd(INITIAL)),
   asserta(cummdold(1,0)),
   asserta(cummdold(2,0)),
   asserta(cummdold(3,0)),
   asserta(cummdold(4,0)),
   asserta(cummdold(5,0)),
   asserta(cummdold(6,0)),
   asserta(cummdold(7,0)),
   asserta(cummdold(8,0)),
   asserta(cummdold(9,0)),
   asserta(cummdold(10,INITIAL)),
   retract(cummd(INITIAL)).
/* end of initial_run */
##############################################################
```

D. GETTING STARTED

To initiate the expert system simulation model first make sure you are in the directory containing the files for the expert system simulation model. Then start the Arity Prolog Interpreter by typing (all required user input will appear in boldface type):

C> **API**

Once you get the Arity Prolog prompt (?-) type:

?- **consult(pqa).**

The system will respond with "yes" and leave another prompt. Type:

?- **pqa.**

Now the system will ask a series of five questions (shown below with sample responses). All of the answers that are input <u>must</u> end with a period (.). All answers must be positive and if they are less than 1, they must have a 0 before the decimal point. The first question will be:

What is your desired pulse size factor? **0.25.**

This input will determine the size of each spike generated by the expert system. Enter a number between 0 and 1.

The second question will be:

What is the maximum number of cycles? **10.**

This will determine how many times the system will go through all of the TPFMQA values unless the exit condition is met. Enter a positive integer.

The third question will be:

What is the exit condition? 0.0001.

This is the minimum percent change in cumulative man days between consecutive cycles in order for the system to continue. Enter 0 if you want all of the cycles to be executed. This is not recommended if more than 30 cycles has been requested. Numbers greater than 0.001 are not recommended as they will cause the program to terminate after only a few cycles.

The fourth question will be:

What is the minimum QA value? 0.025.

This is the least amount of quality assurance that will be allowed at any of the TPFMQA points.

Finally the user will be prompted to enter the initial QA distribution, one point at a time, as follows:

Enter the initial QA distribution. Point 1 0.15.

Point 2 0.15.

Point 3 0.15.

Point 4 0.15.

Point 5 0.15.

Point 6 0.15.

Point 7 0.15.

Point 8 0.15.

Point 9 0.15.

Point 10 0.15.

Point 11 0.

This will establish the initial distribution of TPFMQA values (the last TPFMQA value is usually zero).

After the five questions have been answered the model will begin to run. This will be very time intensive so be patient. The results of each iteration will be stored in the file SUMMARY.DAT in the current directory.

IV. EXPERIMENT: NASA´S DE-A PROJECT

A. THE DE-A SOFTWARE PROJECT

To test the effectiveness of the expert system simulation model, the model results can be compared to the results of an actual software project. The real project that will be used was conducted at the Systems Development Section of NASA´s Goddard Space Flight Center (GSFC) at Greenbelt, Maryland. The key requirements for the project were to design, implement, and test a software system for processing telemetry data and providing attitude determination and control for the DE-A satellite.

It was estimated that the size of the system would be 16,000 delivered source instructions (DSI), that it would need 1,100 man-days to develop and test, and that it would be completed in 320 working days. Quality assurance (QA) resources were apportioned in proportion to the project´s total development effort. In the case of the DE-A project, 30% of the project´s development resources were allotted to QA. The final statistics for the project were as follows:

project size	24,000 DSI
development cost	2,200 man-days
completion time	380 working days

Figure 4-1 shows the actual QA expenditure as a percentage of total effort for this project. [Ref. 6:p. 396]

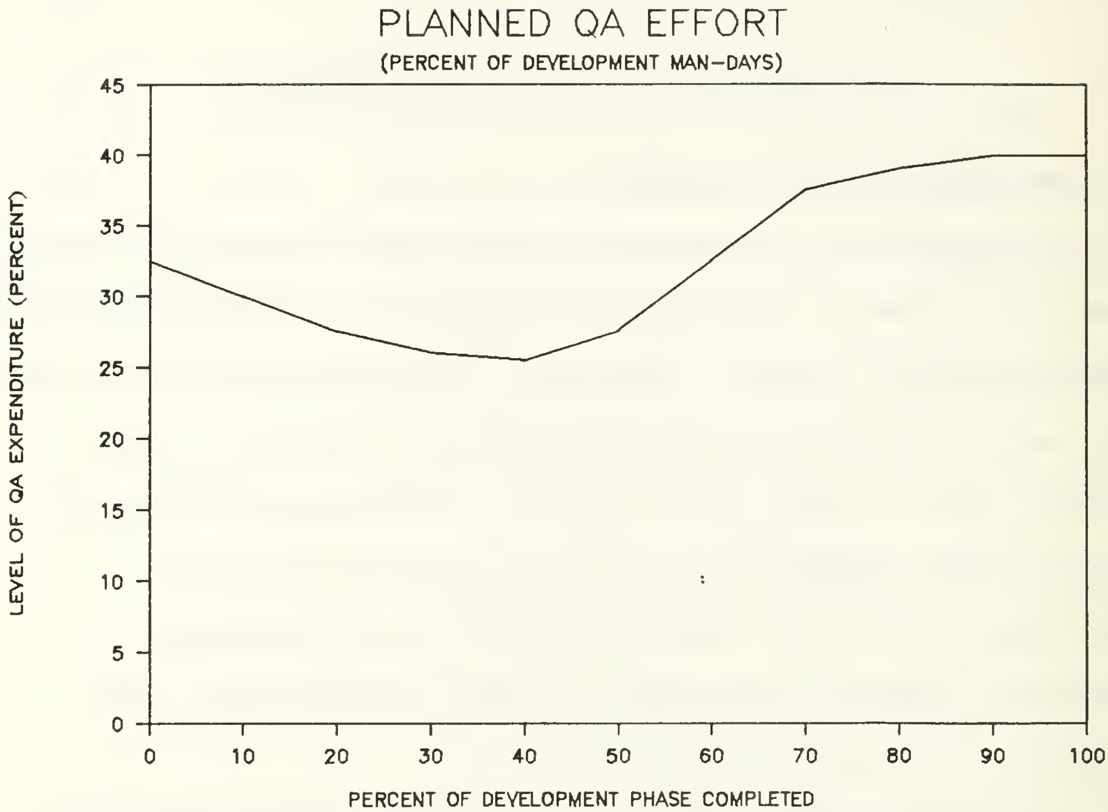


Figure 4-1. Actual DE-A QA distribution

B. THE MANUAL EXPERIMENT

To determine the effectiveness of NASA´s QA policy for the DE-A project an experiment to find the optimal level of QA effort was performed [Ref. 6:p. 407]. This experiment used the Dynamo model of software development with changes being manually made to the TPFMQA table. The initial QA effort was distributed uniformly at 15%. Then a search was made for the places in the project´s lifecycle where the 15% level was not cost effective. This was done by adjusting the QA level with a negative pulse one point at a time, as shown in Figure 4-2, then running the simulation to check the

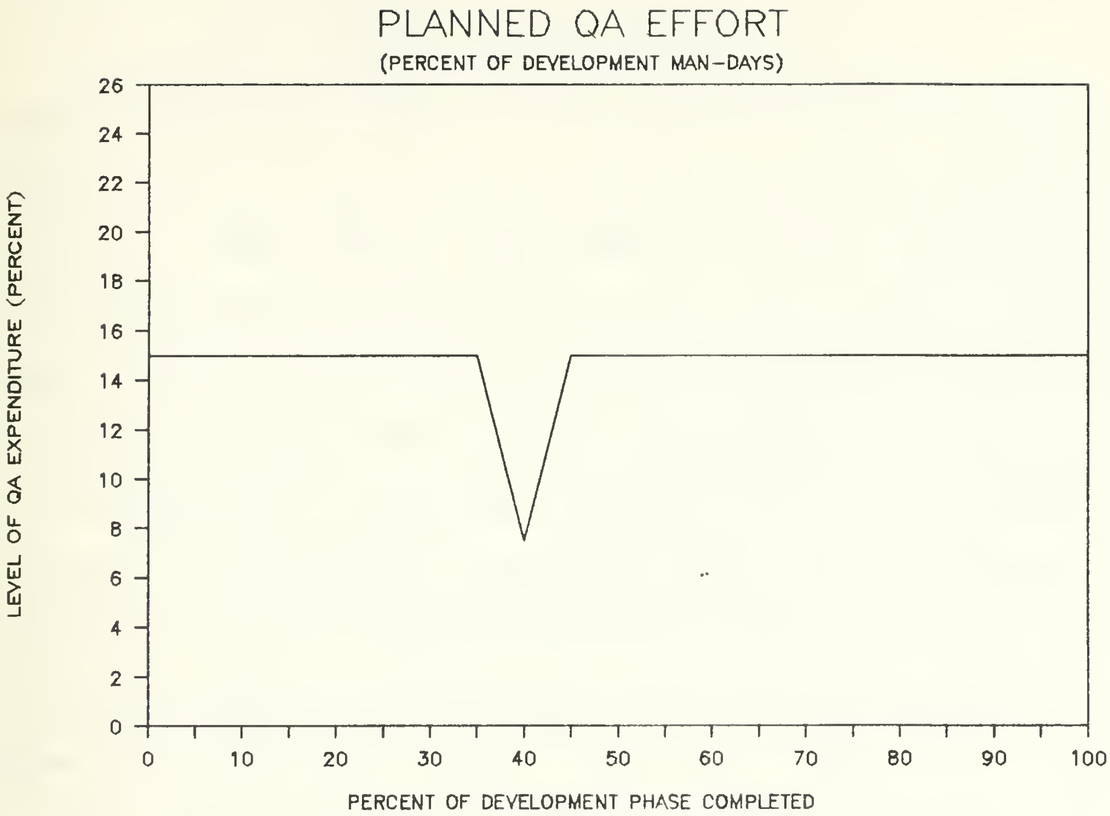


Figure 4-2. Example Negative Impulse

effect of the pulse on the total cost in man days. If the negative pulse led to a decrease in the project's total cost, then the QA value remained at the lower level. If the negative pulse resulted in a higher total cost for the project then the QA value for that point was adjusted by a positive pulse. If the positive pulse led to a decrease in the project's total cost, then the QA value remained at the higher level. This was done at each of the 10 TPFMQA values. Then the technique was repeated until the change between consecutive cycles was insignificant. The magnitude of every pulse was set at 50% of the TPFMQA value. This method

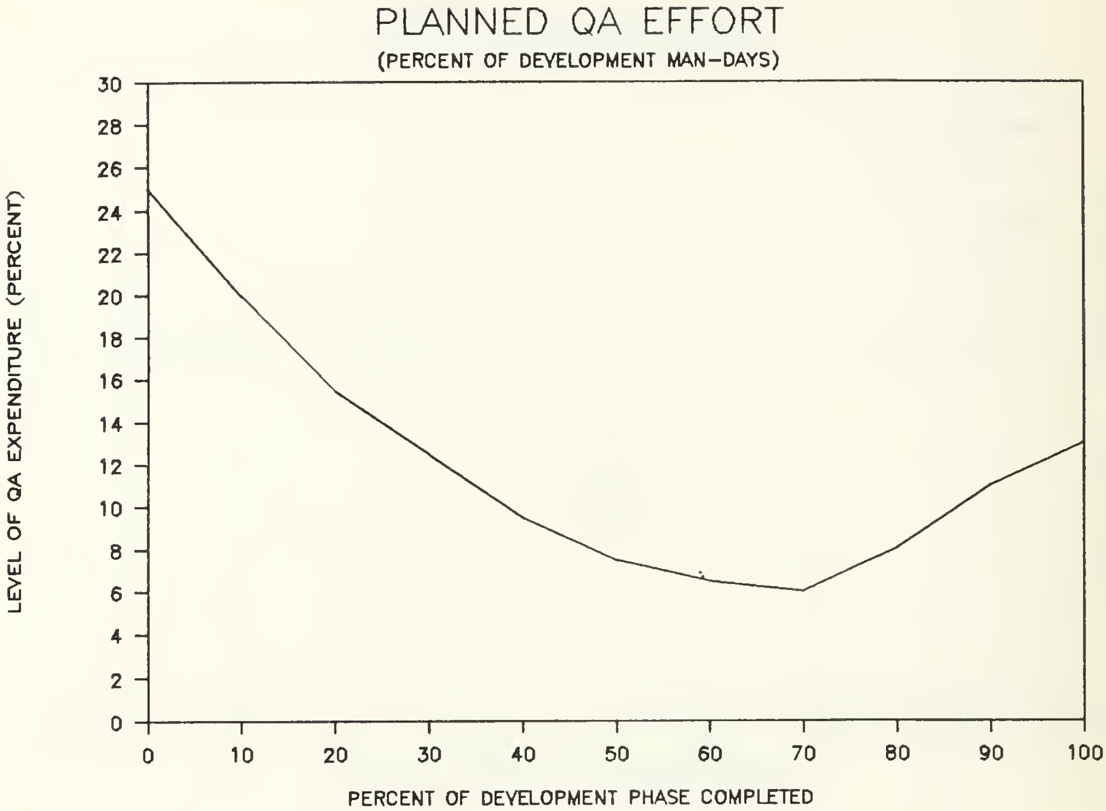


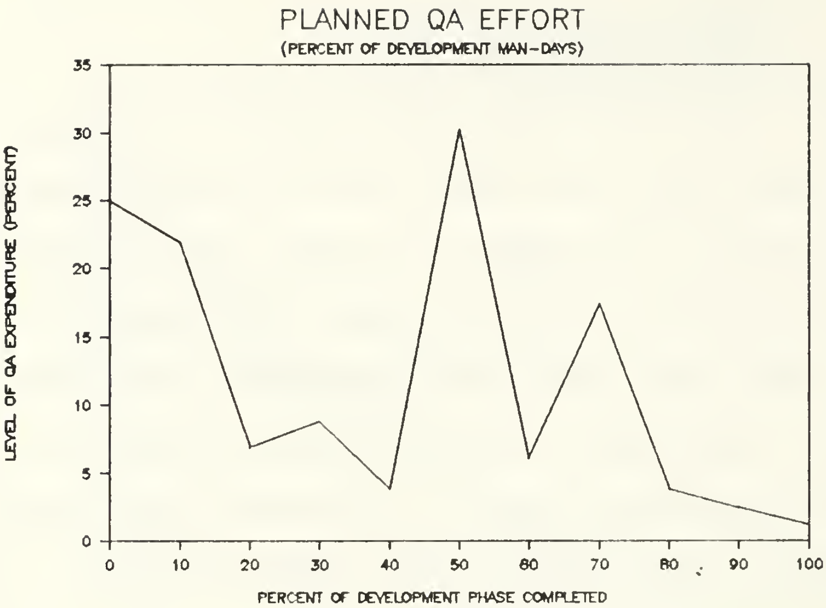
Figure 4-3. QA Distribution For Manual Experiment

resulted in a total project cost of 1524.5 man-days with 161.9 man-days spent on the QA effort as compared with an actual total project cost of 2093.0 man-days with 524.0 man-days of QA effort. Figure 4-3 shows the QA expenditure as a percentage of total effort for the manual experiment. [Ref. 6:pp. 407-408]

C. EXPERIMENT WITH THE EXPERT SYSTEM SIMULATION MODEL

Four runs of the expert system simulation model were made for the DE-A project. The results are summarized in figures 4-4 through 4-9. Figures 4-4 through 4-7 graphically display the level of QA expenditure, in percent, versus the percent of the development phase completed for all of the experimental runs. Figure 4-8 shows the initial five input values plus the total cost in man days that was achieved with each run. Additionally, the number of cycles column shows the maximum number that was originally input along with the actual number of cycles that were performed before the exit condition was met. The first run used the same pulse size and minimum quality assurance values as the manual experiment discussed in Section B. The results were quite different between the manual experiment and the expert system simulation model, with the manual experiment producing a better conclusion.

The second experimental run was made with the same 0.5 pulse size factor, but the minimum QA was doubled from Run #1 to 0.050. The results showed some sensitivity to increases in the minimum QA in the form of a decrease in the total cost. But this total cost was still greater than with the manual experiment. The last two experimental runs were made with a pulse size of 0.15, much smaller than that used in the manual experiment. Run #3 used 0.050 as the minimum QA as did Run #2. The total cost in man days improved significantly with this change. This indicates that the expert system simulation model has a great deal of sensitivity to the pulse size. Run #4 kept the 0.15 pulse size factor but decreased the minimum QA to 0.030. This



Pulse Size: 0.5

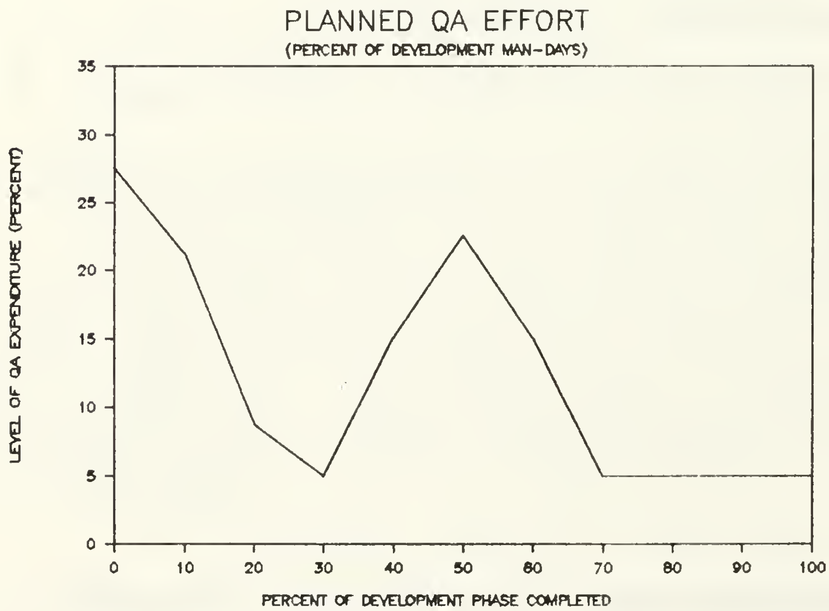
\# of Cycles: 30

Exit Condition: 0.0000001

Minimum QA: 0.025

Total Man-days: 1,534.07

Figure 4-4. QA Distribution For Experimental Run #1



Pulse Size: 0.5

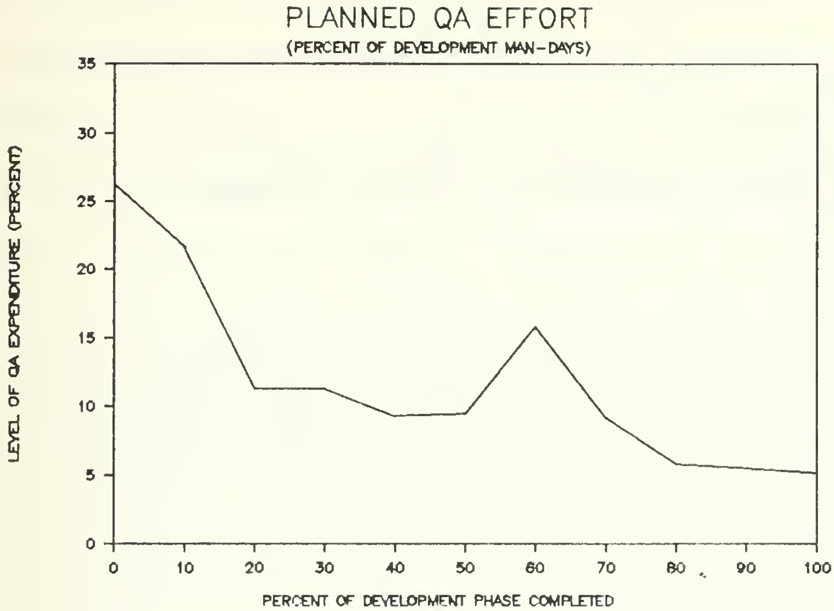
\# of Cycles: 30

Exit Condition: 0.0000001

Minimum QA: 0.05

Total Man-days: 1,531.54

Figure 4-5. QA Distribution For Experimental Run #2



Pulse Size: 0.15

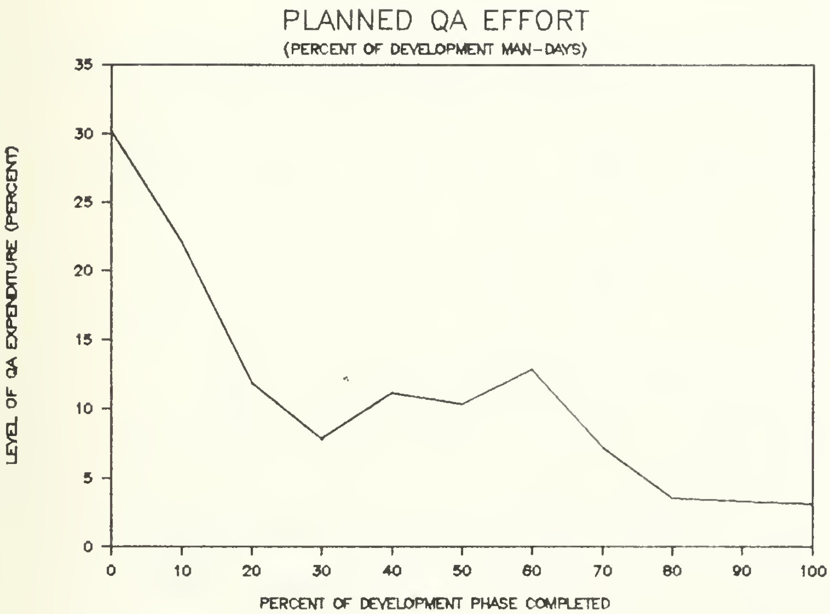
# of Cycles: 30

Exit Condition: 0.0000001

Minimum QA: 0.05

Total Man-days: 1,522.96

Figure 4-6. QA Distribution For Experimental Run #3



Pulse Size: 0.15

# of Cycles: 30

Exit Condition: 0.0000001

Minimum QA: 0.03

Total Man-days: 1,515.66

Figure 4-7. QA Distribution For Experimental Run #4 provided the best results of any of the four experimental runs. The complete SUMMARY.DAT file listing for the best run is enclosed as APPENDIX B to this thesis. Figure 4-9 shows the final TPFMQA values for each of the four experimental runs.

	Pulse Size	# of Cycles	Exit Condition	Min QA	Init QA	Cost in man-days
Run #1	0.5	30/15	0.0000001	0.025	0.15	1,534.07
Run #2	0.5	30/7	0.0000001	0.050	0.15	1,531.54
Run #3	0.15	30/12	0.0000001	0.050	0.15	1,522.96
Run #4	0.15	30/20	0.0000001	0.030	0.15	1,515.66

Figure 4-8. Input Parameters For Experimental Runs.

The results of the four experimental runs indicate some amount of sensitivity to the pulse size factor and the minimum QA values that are used. The exit condition will also have some effect on the outcome, although not demonstrated by any of the experiment runs since the value

TPFMQA VALUES FOR FOUR EXPERIMENTAL RUNS											
Stage	1	2	3	4	5	6	7	8	9	10	11
Run #1	.250	.219	.069	.088	.038	.302	.161	.174	.038	.025	0
Run #2	.276	.213	.088	.050	.151	.226	.151	.050	.050	.050	0
Run #3	.263	.216	.113	.113	.093	.095	.158	.092	.058	.055	0
Run #4	.302	.222	.119	.078	.111	.103	.128	.072	.035	.033	0

Figure 4-9. Final TPFMQA Values.

used for all four experiment runs was the same (10^{-7}). Even with this small value for the exit condition, the maximum number of cycles was never reached. Larger values for the exit condition will cause the system to terminate earlier, with a higher total cost. The model is quite insensitive to the maximum number of cycles unless zero has been entered as the exit condition or if a small number of cycles is desired. With an exit condition of zero, the model will continue to run until the maximum number of cycles has been reached. In the case of a small number of cycles, the model may terminate before reaching an optimal solution.

V. CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

A. ACCOMPLISHMENTS

The primary objective of this thesis was the development of an expert system to generate the optimal level and distribution of quality assurance for a software development project throughout its lifecycle. An Arity Prolog program was created that meets this objective when it is interfaced to the Dynamo Model of Software Project Management discussed in Chapter II.

The expert system simulation model accepts user input of several parameters (pulse size, maximum number of cycles, exit condition, minimum QA, and the initial QA distribution) then proceeds to determine the optimal mix of quality assurance for those specific parameters. The system was found to be most sensitive to different values for the pulse size and the minimum QA parameters.

B. SUGGESTIONS FOR FUTURE RESEARCH

There are many areas available for future research. These areas fall under two categories, namely (1) refining the current expert system simulation model and (2) expanding the capabilities of the expert system simulation model.

1. Refining the Current Expert System

The current version of the expert system simulation model assumes that the user has a starting point for the

TPFMQA table. A rule could be added to the current program that would first calculate the optimal uniform distribution of QA for a given project.

Adjustments to the QA distribution are presently based on a constant pulse size factor. This factor could be reduced as the improvements in cost after each cycle decrease.

The final QA distribution generated by the current system may be organizationally unacceptable as it would require a great deal of adding to and subtracting from the QA effort throughout the lifecycle. A possible refinement would be to ensure a "smoother" QA distribution curve to make the outcome more politically sound.

2. Expanding the Capabilities of the Expert System

This system only considers the effect of changing the QA effort on the total project cost. Changes in other variables, e.g. testing, will also have an effect on the cost of a project. The capability of considering the effect of changing other variables on the total project cost could be added.

Finally, adding the capability to keep an eye on the time while trying to optimize the cost, and vice versa, would be a helpful expansion of the current system's capabilities.

APPENDIX A

PROGRAM LISTING OF PQA.ARI

```
/****************************************************************/
/*                                                              */
/* PROGRAM - pqa.ari                                            */
/*                                                              */
/* This program will accept a value for cost in man days to    */
/* complete a software project and output values for the      */
/* desired level of quality assurance in order to optimize the */
/* cost in man days.                                            */
/*                                                              */
/****************************************************************/

/****************************************************************/
/*                                                              */
/* RULE - pqa                                                   */
/*                                                              */
/* This rule initializes the system parameters and gets the    */
/* system started.                                              */
/*                                                              */
/****************************************************************/

pqa:-

   asserta(number(1)),
   asserta(calc(0)),

   write('What is your desired pulse size factor? '),
   read(PU),
   asserta(size(PU)),

   write('What is the maximum number of cycles? '),
   read(MX),
   asserta(cycle(MX)),
   asserta(newcycle(1)),

   write('What is the exit condition? '),
   read(EX),
   asserta(stop(EX)),

   write('What is the minimum QA value? '),
   read(MN),
   asserta(min(MN)),

   write('Enter the initial QA distribution.  Point 1 '),
   read(QA1),

   write('                                     Point 2 '),
   read(QA2),
```

```
   write('                                  Point 3 '),
   read(QA3),

   write('                                  Point 4 '),
   read(QA4),

   write('                                  Point 5 '),
   read(QA5),

   write('                                  Point 6 '),
   read(QA6),

   write('                                  Point 7 '),
   read(QA7),

   write('                                  Point 8 '),
   read(QA8),

   write('                                  Point 9 '),
   read(QA9),

   write('                                  Point 10 '),
   read(QA10),

   write('                                  Point 11 '),
   read(QA11),

   calc_zero(PU,MX,EX,MN),

   initial_run(QA1,QA2,QA3,QA4,QA5,QA6,QA7,QA8,QA9,QA10,QA11),

   dopqa.

/* end of pqa */

/***************************************************************/
/*                                                             */
/* RULE - dopqa                                                */
/*                                                             */
/* Creates a repeat-fail loop to execute rule main until the  */
/* maximum number of cycles or the exit condition is met      */
/*                                                             */
/***************************************************************/

dopqa:-

   repeat,

   main,

   fail.
```

```
/* end of dopqa */

/*************************************************************/
/*                                                           */
/* RULE - main                                               */
/*                                                           */
/* The main module.  Does most of the testing to determine   */
/* the next course of action.                                */
/*                                                           */
/*************************************************************/

main:-

/* get the x value for the pulse */
   call(number(ITER)),
   call(calc(TYPE)),

/* get the previous man days */
   PREV is [[[ITER + 8] mod 10] + 1],
   call(cummdold(PREV,CHECK)),
   call(cummdold(ITER,OLD)),

/* get the man days from the last QA numbers */
   read_cummd,
   call(cummd(NEW)),

/* calculate the new y value (QA) for the current x value */
   case([NEW =< CHECK -> calc_less(ITER,NEW,OLD,TYPE),
         NEW > CHECK -> calc_more(ITER,NEW,OLD,CHECK,TYPE)]),

   retract(cummd(NEW)),

   statistics,

   write(' Iteration = '),write(ITER).

/* end of main */

/*************************************************************/
/*                                                           */
/* RULE - quit_test                                          */
/*                                                           */
/* Tests current man days value (NEW) with the final man days */
/* value of the last cycle (OLD).                            */
/*                                                           */
/*************************************************************/

quit_test(NEW):-

/* get final man days of last cycle */
   call(cummdold(1,OLD)),
```

```
/* test if exit condition exits */
   call(stop(EXIT)),

   ifthenelse(OLD =:= 0,TEST is 100,TEST is abs(OLD-NEW)/OLD),
   ifthen(TEST < EXIT,no_calc(NEW)),

   call(cycle(MAX)),
   call(newcycle(NOW)),

   NEXT is NOW + 1,
   ifthen(MAX < NOW,no_calc(NEW)),

   write(' Cycle number = '),write(NOW),

   retract(newcycle(NOW)),
   asserta(newcycle(NEXT)).

/* end of quit_test */

/*****************************************************************/
/*                                                               */
/* RULE - calc_zero                                              */
/*                                                               */
/* Creates summary data files for the man days value received   */
/* from the dynamo module (sumcummd.dat) and for the TPFMQA     */
/* values sent to the dynamo module (sumqa.dat).  Then          */
/* calc_less is invoked to generate the first pulse.            */
/*                                                               */
/*****************************************************************/

calc_zero(PU,MX,EX,MN):-

   create(S,'summary.dat'),

   write(S,'       Pulse size factor = '),write(S,PU),nl(S),
   write(S,'Maximum number of cycles = '),write(S,MX),nl(S),
   write(S,'          Exit condition = '),write(S,EX),nl(S),
   write(S,'        Minimum QA value = '),
   write(S,MN),nl(S),nl(S),

   close(S).

/* end of calc_zero */

/*****************************************************************/
/*                                                               */
/* RULE - calc_less                                              */
/*                                                               */
/* Calculates a negative pulse at the current position equal    */
/* to one half of the QA value at position number 10.           */
/*                                                               */
/*****************************************************************/
```

```
calc_less(ITER,NEW,OLD,TYPE):-

   ifthen(ITER =:= 1,quit_test(NEW)),

/* read the current QA values */
   call(tpfmqa(ITER,QA)),
   call(tpfmqa(10,DIV)),

   call(size(PULSE)),

/* calculate the new QA value (NEWQA) */
   NEWQA is round([QA-[PULSE*DIV]],3),

/* check if the new QA is less than the minimum (MINQA) */
   call(min(MINQA)),
   retract(tpfmqa(ITER,QA)),
   ifthenelse(NEWQA < MINQA,asserta(tpfmqa(ITER,MINQA)),
              asserta(tpfmqa(ITER,NEWQA))), .

/* add man days for this cycle */
   retract(cummdold(ITER,OLD)),
   asserta(cummdold(ITER,NEW)),

/* move to the next position */
   retract(number(ITER)),

   NEWITER is [ITER mod 10] + 1,
   asserta(number(NEWITER)),

   output_cummd(NEW,ITER),
   ifthen(ITER =:= 1,output_break),
   output_tpfmqa,

   shell(frank),

/* save fact that last pulse was negative */
   retract(calc(TYPE)),
   asserta(calc(0)).

/* end of calc_less */

/****************************************************************/
/*                                                              */
/* RULE - calc_more                                             */
/*                                                              */
/* Calculates a positive pulse if the previous negative pulse   */
/* resulted in a higher man days value.  If the man days        */
/* resulting from the positive pulse also results in a higher   */
/* man days value then the QA value will be returned to its     */
/* original value for this cycle.                               */
/*                                                              */
```

```
/*****************************************************************/

calc_more(ITER,NEW,OLD,CHECK,TYPE):-

/* reset the position back 1 */
   NEWITER is [[[ITER + 8] mod 10] + 1],

   call(tpfmqa(NEWITER,QA)),
   call(tpfmqa(10,DIV)),

   call(size(PULSE)),

/* calculate the new QA value depending on whether the last */
/* pulse was negative (TYPE = 0) or positive (TYPE = 1) */
   case([TYPE =:= 0 -> NEWQA is round(QA+[2 * PULSE * DIV],3),
         TYPE =:= 1 -> NEWQA is round(QA-[PULSE * DIV],3)]),

/* check if the new QA is less than the minimum (MINQA) */
/* add the new QA value to the database */
   call(min(MINQA)),
   retract(tpfmqa(NEWITER,QA)),
   ifthenelse(NEWQA < MINQA,asserta(tpfmqa(NEWITER,MINQA)),
              asserta(tpfmqa(NEWITER,NEWQA))),

   retract(calc(TYPE)),

/* reset the type of calculation */
   case([TYPE =:= 0 -> calc_up(NEW,NEWITER),
         TYPE =:= 1 -> calc_orig(NEWITER,ITER,NEW,OLD,CHECK)]).

/* end of calc_more */

/*****************************************************************/
/*                                                               */
/* RULE - calc_up                                                */
/*                                                               */
/* Runs the dynamo model after an increase in the current        */
/* QA position.                                                  */
/*                                                               */
/*****************************************************************/

calc_up(NEW,NEWITER):-

   asserta(calc(1)),

   output_cummd(NEW,NEWITER),
   output_tpfmqa,

   shell(frank).

/* end of calc_up */
```

```
/*****************************************************************/
/*                                                               */
/* RULE - calc_orig                                              */
/*                                                               */
/* After the QA value at a point has been returned to its       */
/* original value, this rule continues with the next point.      */
/*                                                               */
/*****************************************************************/

calc_orig(NEWITER,ITER,NEW,OLD,CHECK):-

   asserta(calc(0)),

   output_cummd(NEW,NEWITER),
   output_tpfmqa,
   calc_less(ITER,CHECK,OLD,0).

/* end of calc_orig */

/*****************************************************************/
/*                                                               */
/* RULE - no_calc                                                */
/*                                                               */
/* Halts the program when an exit condition is met.             */
/*                                                               */
/*****************************************************************/

no_calc(NEW):-

   call(cummdold(10,BEST)),

   open(S,'summary.dat',a),
   write(S,'10. CUMMD='),write(S,NEW),nl(S),nl(S),

   write(S,'The best CUMMD is: '),

   ifthenelse(BEST < NEW,write(S,BEST),write(S,NEW)),

   nl(S),
   close(S),

   halt.

/* end of no_calc */

/*****************************************************************/
/*                                                               */
/* RULE - output_tpfmqa                                          */
/*                                                               */
/* Creates one file and updates one file:                        */
/*                                                               */
/*    1. project.dnx - created with the format required by the */
```

```
/*      dynamo simulation model.                                */
/*                                                              */
/*   2. summary.dat - updated by adding the current tpfmqa      */
/*      values to the end of the list of all previous values.   */
/*                                                              */
/****************************************************************/

output_tpfmqa:-

   call(tpfmqa(1,QA1)),
   call(tpfmqa(2,QA2)),
   call(tpfmqa(3,QA3)),
   call(tpfmqa(4,QA4)),
   call(tpfmqa(5,QA5)),
   call(tpfmqa(6,QA6)),
   call(tpfmqa(7,QA7)),
   call(tpfmqa(8,QA8)),
   call(tpfmqa(9,QA9)),
   call(tpfmqa(10,QA10)),
   call(tpfmqa(11,QA11)),

   create(D,'project.dnx'),
   write(D,'T TPFMQA='),
   write(D,QA1),write(D,' '),
   write(D,QA2),write(D,' '),
   write(D,QA3),write(D,' '),
   write(D,QA4),write(D,' '),
   write(D,QA5),write(D,' '),
   write(D,QA6),write(D,' '),
   write(D,QA7),write(D,' '),
   write(D,QA8),write(D,' '),
   write(D,QA9),write(D,' '),
   write(D,QA10),write(D,' '),
   write(D,QA11),nl(D),
   close(D),

   open(S,'summary.dat',a),
   write(S,'TPFMQA='),
   write(S,QA1),write(S,'/'),
   write(S,QA2),write(S,'/'),
   write(S,QA3),write(S,'/'),
   write(S,QA4),write(S,'/'),
   write(S,QA5),write(S,'/'),
   write(S,QA6),write(S,'/'),
   write(S,QA7),write(S,'/'),
   write(S,QA8),write(S,'/'),
   write(S,QA9),write(S,'/'),
   write(S,QA10),write(S,'/'),
   write(S,QA11),nl(S),
   close(S).

/* end of output_tpfmqa */
```

```
/****************************************************************/
/*                                                              */
/* RULE - output_cummd                                          */
/*                                                              */
/* Outputs cumulative man days total to file SUMMARY.DAT        */
/*                                                              */
/****************************************************************/

output_cummd(NEW,ITER):-

   open(S,'summary.dat',a),

   write(S,ITER),write(S,'. '),
   write(S,'CUMMD='),write(S,NEW),nl(S),

   close(S).

/* end of output_cummd */

/****************************************************************/
/*                                                              */
/* RULE - output_break                                          */
/*                                                              */
/* Outputs a line denoting the start of a new cycle to          */
/* SUMMARY.DAT                                                  */
/*                                                              */
/****************************************************************/

output_break:-

   open(S,'summary.dat',a),

   nl(S),write(S,
   '****************   Start of a new cycle   ****************'),
   nl(S),nl(S),

   close(S).

/* end of output_break */

/****************************************************************/
/*                                                              */
/* RULE - read_cummd                                            */
/*                                                              */
/* Reads from file cummd.dat the value of man days output by    */
/* the dynamo simulation model.                                 */
/*                                                              */
/****************************************************************/

read_cummd:-
```

```
   open(C,'project.out',r),

   read(C,CUMMD),
   asserta(CUMMD),

   close(C).

/* end of read_cummd */

/***************************************************************/
/*                                                             */
/* RULE - initial_run                                          */
/*                                                             */
/* Runs the dynamo model with the initial TPFMQA values.       */
/*                                                             */
/***************************************************************/

initial_run(QA1,QA2,QA3,QA4,QA5,QA6,QA7,QA8,QA9,QA10,QA11):-

   asserta(tpfmqa(1,QA1)),
   asserta(tpfmqa(2,QA2)),
   asserta(tpfmqa(3,QA3)),
   asserta(tpfmqa(4,QA4)),
   asserta(tpfmqa(5,QA5)),
   asserta(tpfmqa(6,QA6)),
   asserta(tpfmqa(7,QA7)),
   asserta(tpfmqa(8,QA8)),
   asserta(tpfmqa(9,QA9)),
   asserta(tpfmqa(10,QA10)),
   asserta(tpfmqa(11,QA11)),

   output_tpfmqa,
   shell(frank),

   read_cummd,
   call(cummd(INITIAL)),

   asserta(cummdold(1,0)),
   asserta(cummdold(2,0)),
   asserta(cummdold(3,0)),
   asserta(cummdold(4,0)),
   asserta(cummdold(5,0)),
   asserta(cummdold(6,0)),
   asserta(cummdold(7,0)),
   asserta(cummdold(8,0)),
   asserta(cummdold(9,0)),
   asserta(cummdold(10,INITIAL)),

   retract(cummd(INITIAL)).

/* end of initial_run */
/* end of program pqa.ari */
```

APPENDIX B

RESULTS OF EXPERIMENT RUN NUMBER FOUR

```
       Pulse size factor = 0.15
Maximum number of cycles = 30
          Exit condition = 0.0000001
        Minimum QA value = 0.03

TPFMQA=0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
1. CUMMD=1656.71

****************   Start of a new cycle   ****************

TPFMQA=0.128/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
1. CUMMD=1707.37
TPFMQA=0.173/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
2. CUMMD=1616.99
TPFMQA=0.173/0.128/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
2. CUMMD=1625.36
TPFMQA=0.173/0.173/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
3. CUMMD=1614.21
TPFMQA=0.173/0.173/0.128/0.15/0.15/0.15/0.15/0.15/0.15/0.15/0
4. CUMMD=1607.39
TPFMQA=0.173/0.173/0.128/0.128/0.15/0.15/0.15/0.15/0.15/0.15/0
5. CUMMD=1603.66
TPFMQA=0.173/0.173/0.128/0.128/0.128/0.15/0.15/0.15/0.15/0.15/0
6. CUMMD=1597.89
TPFMQA=0.173/0.173/0.128/0.128/0.128/0.128/0.15/0.15/0.15/0.15/0
7. CUMMD=1591.42
TPFMQA=0.173/0.173/0.128/0.128/0.128/0.128/0.128/0.15/0.15/0.15/0
8. CUMMD=1591.42
TPFMQA=0.173/0.173/0.128/0.128/0.128/0.128/0.128/0.128/0.15/0.15/0
9. CUMMD=1591.36
TPFMQA=0.173/0.173/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0.15/0
10. CUMMD=1584.65
TPFMQA=0.173/0.173/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0
1. CUMMD=1584.57

****************   Start of a new cycle   ****************

TPFMQA=0.154/0.173/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0
1. CUMMD=1610.64
TPFMQA=0.192/0.173/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0
2. CUMMD=1567.04
TPFMQA=0.192/0.154/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0
2. CUMMD=1568.45
TPFMQA=0.192/0.192/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0
3. CUMMD=1566.8
TPFMQA=0.192/0.192/0.109/0.128/0.128/0.128/0.128/0.128/0.128/0.128/0
4. CUMMD=1565.99
```

TPFMQA=0.192/0.192/0.109/0.109/0.128/0.128/0.128/0.128/0.128/0.128/0
5. CUMMD=1562.12
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.128/0.128/0.128/0.128/0.128/0
6. CUMMD=1556.55
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.128/0.128/0.128/0.128/0
7. CUMMD=1550.25
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.109/0.128/0.128/0.128/0
7. CUMMD=1550.29
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.147/0.128/0.128/0.128/0
7. CUMMD=1556.78
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.128/0.128/0.128/0.128/0
8. CUMMD=1550.25
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.128/0.109/0.128/0.128/0
9. CUMMD=1550.22
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.128/0.109/0.109/0.128/0
10. CUMMD=1550.18
TPFMQA=0.192/0.192/0.109/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
1. CUMMD=1543.56

**************** Start of a new cycle ****************

TPFMQA=0.176/0.192/0.109/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
1. CUMMD=1553.51
TPFMQA=0.209/0.192/0.109/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
2. CUMMD=1540.58
TPFMQA=0.209/0.176/0.109/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
3. CUMMD=1539.92
TPFMQA=0.209/0.176/0.093/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
3. CUMMD=1544.96
TPFMQA=0.209/0.176/0.126/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
3. CUMMD=1540.91
TPFMQA=0.209/0.176/0.11/0.109/0.109/0.109/0.128/0.109/0.109/0.109/0
4. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.093/0.109/0.109/0.128/0.109/0.109/0.109/0
4. CUMMD=1541.91
TPFMQA=0.209/0.176/0.11/0.126/0.109/0.109/0.128/0.109/0.109/0.109/0
4. CUMMD=1543.59
TPFMQA=0.209/0.176/0.11/0.11/0.109/0.109/0.128/0.109/0.109/0.109/0
5. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.11/0.093/0.109/0.128/0.109/0.109/0.109/0
5. CUMMD=1540.39
TPFMQA=0.209/0.176/0.11/0.11/0.126/0.109/0.128/0.109/0.109/0.109/0
5. CUMMD=1544.98
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.109/0.128/0.109/0.109/0.109/0
6. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.093/0.128/0.109/0.109/0.109/0
6. CUMMD=1546.29
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.126/0.128/0.109/0.109/0.109/0
6. CUMMD=1545.62
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.128/0.109/0.109/0.109/0
7. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.112/0.109/0.109/0.109/0

```
7. CUMMD=1546.03
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.145/0.109/0.109/0.109/0
7. CUMMD=1545.84
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.109/0.109/0.109/0
8. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.093/0.109/0.109/0
8. CUMMD=1545.95
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.126/0.109/0.109/0
8. CUMMD=1545.91
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.109/0.109/0
9. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.093/0.109/0
9. CUMMD=1545.93
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.126/0.109/0
9. CUMMD=1545.92
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.109/0
10. CUMMD=1539.92
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.093/0
10. CUMMD=1545.93
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.121/0
10. CUMMD=1545.93
TPFMQA=0.209/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
1. CUMMD=1539.92

****************   Start of a new cycle   ****************

TPFMQA=0.194/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
1. CUMMD=1548.64
TPFMQA=0.225/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
1. CUMMD=1543.24
TPFMQA=0.21/0.176/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
2. CUMMD=1539.92
TPFMQA=0.21/0.161/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
2. CUMMD=1545.3
TPFMQA=0.21/0.192/0.11/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
3. CUMMD=1539.86
TPFMQA=0.21/0.192/0.095/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
3. CUMMD=1544.51
TPFMQA=0.21/0.192/0.126/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
3. CUMMD=1541.48
TPFMQA=0.21/0.192/0.111/0.11/0.11/0.11/0.129/0.11/0.11/0.103/0
4. CUMMD=1539.86
TPFMQA=0.21/0.192/0.111/0.095/0.11/0.11/0.129/0.11/0.11/0.103/0
4. CUMMD=1541.75
TPFMQA=0.21/0.192/0.111/0.126/0.11/0.11/0.129/0.11/0.11/0.103/0
4. CUMMD=1543.78
TPFMQA=0.21/0.192/0.111/0.111/0.11/0.11/0.129/0.11/0.11/0.103/0
5. CUMMD=1539.86
TPFMQA=0.21/0.192/0.111/0.111/0.095/0.11/0.129/0.11/0.11/0.103/0
5. CUMMD=1540.28
TPFMQA=0.21/0.192/0.111/0.111/0.126/0.11/0.129/0.11/0.11/0.103/0
5. CUMMD=1545.09
```

```
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.11/0.129/0.11/0.11/0.103/0
6. CUMMD=1539.86
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.129/0.11/0.11/0.103/0
7. CUMMD=1539.63
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.114/0.11/0.11/0.103/0
7. CUMMD=1539.71
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.145/0.11/0.11/0.103/0
8. CUMMD=1539.55
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.145/0.095/0.11/0.103/0
8. CUMMD=1539.57
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.145/0.126/0.11/0.103/0
8. CUMMD=1539.57
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.145/0.111/0.11/0.103/0
9. CUMMD=1539.55
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.103/0
10. CUMMD=1539.55
TPFMQA=0.21/0.192/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
1. CUMMD=1539.55

****************   Start of a new cycle   ****************

TPFMQA=0.197/0.192/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
1. CUMMD=1541.78
TPFMQA=0.223/0.192/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
2. CUMMD=1537.36
TPFMQA=0.223/0.179/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
2. CUMMD=1542.12
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
3. CUMMD=1532.81
TPFMQA=0.223/0.205/0.098/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
3. CUMMD=1536.77
TPFMQA=0.223/0.205/0.124/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
3. CUMMD=1535.49
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
4. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.098/0.111/0.095/0.145/0.111/0.095/0.088/0
4. CUMMD=1534.72
TPFMQA=0.223/0.205/0.111/0.124/0.111/0.095/0.145/0.111/0.095/0.088/0
4. CUMMD=1537.48
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
5. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.111/0.098/0.095/0.145/0.111/0.095/0.088/0
5. CUMMD=1533.56
TPFMQA=0.223/0.205/0.111/0.111/0.124/0.095/0.145/0.111/0.095/0.088/0
5. CUMMD=1538.66
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
6. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.082/0.145/0.111/0.095/0.088/0
6. CUMMD=1533.05
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.108/0.145/0.111/0.095/0.088/0
6. CUMMD=1539.17
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
```

```
7. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.132/0.111/0.095/0.088/0
7. CUMMD=1532.86
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.158/0.111/0.095/0.088/0
7. CUMMD=1539.32
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.111/0.095/0.088/0
8. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.098/0.095/0.088/0
9. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.098/0.082/0.088/0
10. CUMMD=1532.81
TPFMQA=0.223/0.205/0.111/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
1. CUMMD=1532.81

****************   Start of a new cycle   ****************

TPFMQA=0.212/0.205/0.111/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
1. CUMMD=1534.69
TPFMQA=0.235/0.205/0.111/0.111/0.111/0.095/0,145/0.098/0.082/0.075/0
2. CUMMD=1530.74
TPFMQA=0.235/0.194/0.111/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
2. CUMMD=1534.61
TPFMQA=0.235/0.217/0.111/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
2. CUMMD=1533.08
TPFMQA=0.235/0.206/0.111/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
3. CUMMD=1530.74
TPFMQA=0.235/0.206/0.1/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
3. CUMMD=1533.74
TPFMQA=0.235/0.206/0.123/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
3. CUMMD=1533.27
TPFMQA=0.235/0.206/0.112/0.111/0.111/0.095/0.145/0.098/0.082/0.075/0
4. CUMMD=1530.74
TPFMQA=0.235/0.206/0.112/0.1/0.111/0.095/0.145/0.098/0.082/0.075/0
4. CUMMD=1531.71
TPFMQA=0.235/0.206/0.112/0.123/0.111/0.095/0.145/0.098/0.082/0.075/0
4. CUMMD=1534.82
TPFMQA=0.235/0.206/0.112/0.112/0.111/0.095/0.145/0.098/0.082/0.075/0
5. CUMMD=1530.74
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.095/0.145/0.098/0.082/0.075/0
6. CUMMD=1530.57
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.084/0.145/0.098/0.082/0.075/0
6. CUMMD=1530.78
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.107/0.145/0.098/0.082/0.075/0
6. CUMMD=1536.87
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.145/0.098/0.082/0.075/0
7. CUMMD=1530.57
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.134/0.098/0.082/0.075/0
7. CUMMD=1530.6
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.157/0.098/0.082/0.075/0
8. CUMMD=1530.5
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.157/0.087/0.082/0.075/0
8. CUMMD=1530.51
```

TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.157/0.11/0.082/0.075/0
8. CUMMD=1537.01
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.157/0.099/0.082/0.075/0
9. CUMMD=1530.5
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.075/0
10. CUMMD=1530.5
TPFMQA=0.235/0.206/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
1. CUMMD=1530.5

**************** Start of a new cycle ****************

TPFMQA=0.225/0.206/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
1. CUMMD=1532.23
TPFMQA=0.244/0.206/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
2. CUMMD=1528.91
TPFMQA=0.244/0.196/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
2. CUMMD=1532.43
TPFMQA=0.244/0.215/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
2. CUMMD=1532.27
TPFMQA=0.244/0.205/0.112/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
3. CUMMD=1528.91
TPFMQA=0.244/0.205/0.102/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
3. CUMMD=1532.33
TPFMQA=0.244/0.205/0.121/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
3. CUMMD=1533.0
TPFMQA=0.244/0.205/0.111/0.112/0.1/0.096/0.157/0.099/0.071/0.064/0
4. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.102/0.1/0.096/0.157/0.099/0.071/0.064/0
4. CUMMD=1531.05
TPFMQA=0.244/0.205/0.111/0.121/0.1/0.096/0.157/0.099/0.071/0.064/0
4. CUMMD=1534.72
TPFMQA=0.244/0.205/0.111/0.111/0.1/0.096/0.157/0.099/0.071/0.064/0
5. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.111/0.09/0.096/0.157/0.099/0.071/0.064/0
5. CUMMD=1530.28
TPFMQA=0.244/0.205/0.111/0.111/0.109/0.096/0.157/0.099/0.071/0.064/0
5. CUMMD=1529.2
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.096/0.157/0.099/0.071/0.064/0
6. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.086/0.157/0.099/0.071/0.064/0
6. CUMMD=1529.96
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.105/0.157/0.099/0.071/0.064/0
6. CUMMD=1529.61
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.157/0.099/0.071/0.064/0
7. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.147/0.099/0.071/0.064/0
7. CUMMD=1529.84
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.166/0.099/0.071/0.064/0
7. CUMMD=1529.75
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.099/0.071/0.064/0
8. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.089/0.071/0.064/0

```
8. CUMMD=1529.81
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.108/0.071/0.064/0
8. CUMMD=1529.79
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.071/0.064/0
9. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.064/0
9. CUMMD=1529.8
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.08/0.064/0
9. CUMMD=1529.8
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.064/0
10. CUMMD=1528.91
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.054/0
10. CUMMD=1529.8
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.07/0
10. CUMMD=1529.8
TPFMQA=0.244/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
1. CUMMD=1528.91

****************   Start of a new cycle   ****************

TPFMQA=0.235/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
1. CUMMD=1531.38
TPFMQA=0.253/0.205/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
2. CUMMD=1528.19
TPFMQA=0.253/0.196/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
2. CUMMD=1531.38
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
3. CUMMD=1525.01
TPFMQA=0.253/0.214/0.102/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
3. CUMMD=1527.72
TPFMQA=0.253/0.214/0.12/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
3. CUMMD=1528.79
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
4. CUMMD=1525.01
TPFMQA=0.253/0.214/0.111/0.102/0.099/0.095/0.156/0.098/0.07/0.06/0
4. CUMMD=1526.34
TPFMQA=0.253/0.214/0.111/0.12/0.099/0.095/0.156/0.098/0.07/0.06/0
4. CUMMD=1530.15
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
5. CUMMD=1525.01
TPFMQA=0.253/0.214/0.111/0.111/0.09/0.095/0.156/0.098/0.07/0.06/0
5. CUMMD=1525.52
TPFMQA=0.253/0.214/0.111/0.111/0.108/0.095/0.156/0.098/0.07/0.06/0
5. CUMMD=1531.0
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
6. CUMMD=1525.01
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.086/0.156/0.098/0.07/0.06/0
6. CUMMD=1525.17
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.104/0.156/0.098/0.07/0.06/0
6. CUMMD=1531.35
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
7. CUMMD=1525.01
```

```
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.147/0.098/0.07/0.06/0
7. CUMMD=1525.05
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.165/0.098/0.07/0.06/0
7. CUMMD=1531.47
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
8. CUMMD=1525.01
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.089/0.07/0.06/0
8. CUMMD=1525.02
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.107/0.07/0.06/0
8. CUMMD=1531.51
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.07/0.06/0
9. CUMMD=1525.01
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.06/0
10. CUMMD=1525.01
TPFMQA=0.253/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
1. CUMMD=1525.01

****************   Start of a new cycle   ****************

TPFMQA=0.245/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
1. CUMMD=1526.47
TPFMQA=0.26/0.214/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
2. CUMMD=1523.73
TPFMQA=0.26/0.206/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
2. CUMMD=1526.55
TPFMQA=0.26/0.221/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
2. CUMMD=1527.74
TPFMQA=0.26/0.213/0.111/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
3. CUMMD=1523.73
TPFMQA=0.26/0.213/0.103/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
3. CUMMD=1526.49
TPFMQA=0.26/0.213/0.118/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
3. CUMMD=1528.46
TPFMQA=0.26/0.213/0.11/0.111/0.099/0.095/0.156/0.098/0.061/0.051/0
4. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.103/0.099/0.095/0.156/0.098/0.061/0.051/0
4. CUMMD=1525.56
TPFMQA=0.26/0.213/0.11/0.118/0.099/0.095/0.156/0.098/0.061/0.051/0
4. CUMMD=1529.83
TPFMQA=0.26/0.213/0.11/0.11/0.099/0.095/0.156/0.098/0.061/0.051/0
5. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.11/0.091/0.095/0.156/0.098/0.061/0.051/0
5. CUMMD=1524.98
TPFMQA=0.26/0.213/0.11/0.11/0.106/0.095/0.156/0.098/0.061/0.051/0
5. CUMMD=1530.63
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.095/0.156/0.098/0.061/0.051/0
6. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.087/0.156/0.098/0.061/0.051/0
6. CUMMD=1524.73
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.102/0.156/0.098/0.061/0.051/0
6. CUMMD=1524.47
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.156/0.098/0.061/0.051/0
```

```
7. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.148/0.098/0.061/0.051/0
7. CUMMD=1524.64
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.163/0.098/0.061/0.051/0
7. CUMMD=1524.58
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.098/0.061/0.051/0
8. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.061/0.051/0
8. CUMMD=1524.62
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.105/0.061/0.051/0
8. CUMMD=1524.61
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.061/0.051/0
9. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.053/0.051/0
9. CUMMD=1524.61
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.068/0.051/0
9. CUMMD=1524.61
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.051/0
10. CUMMD=1523.73
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.043/0
10. CUMMD=1524.61
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.056/0
10. CUMMD=1524.61
TPFMQA=0.26/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
1. CUMMD=1523.73

****************   Start of a new cycle   ****************

TPFMQA=0.253/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
1. CUMMD=1525.89
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
2. CUMMD=1523.31
TPFMQA=0.267/0.206/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
2. CUMMD=1525.79
TPFMQA=0.267/0.22/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
2. CUMMD=1527.3
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
3. CUMMD=1523.31
TPFMQA=0.267/0.213/0.103/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
3. CUMMD=1525.42
TPFMQA=0.267/0.213/0.117/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
3. CUMMD=1527.67
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
4. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.103/0.098/0.094/0.155/0.097/0.06/0.048/0
4. CUMMD=1524.33
TPFMQA=0.267/0.213/0.11/0.117/0.098/0.094/0.155/0.097/0.06/0.048/0
4. CUMMD=1528.74
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
5. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.11/0.091/0.094/0.155/0.097/0.06/0.048/0
5. CUMMD=1523.7
```

```
TPFMQA=0.267/0.213/0.11/0.11/0.105/0.094/0.155/0.097/0.06/0.048/0
5. CUMMD=1529.4
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
6. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.087/0.155/0.097/0.06/0.048/0
6. CUMMD=1523.43
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.101/0.155/0.097/0.06/0.048/0
6. CUMMD=1529.67
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
7. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.148/0.097/0.06/0.048/0
7. CUMMD=1523.33
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.162/0.097/0.06/0.048/0
7. CUMMD=1529.77
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.097/0.06/0.048/0
8. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.06/0.048/0
9. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.048/0
10. CUMMD=1523.31
TPFMQA=0.267/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
1. CUMMD=1523.31

****************   Start of a new cycle   ****************

TPFMQA=0.261/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
1. CUMMD=1524.43
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
2. CUMMD=1522.17
TPFMQA=0.273/0.207/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
2. CUMMD=1524.32
TPFMQA=0.273/0.219/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
2. CUMMD=1526.51
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
3. CUMMD=1522.17
TPFMQA=0.273/0.213/0.104/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
3. CUMMD=1523.99
TPFMQA=0.273/0.213/0.116/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
3. CUMMD=1526.84
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
4. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.104/0.098/0.094/0.155/0.09/0.053/0.041/0
4. CUMMD=1523.05
TPFMQA=0.273/0.213/0.11/0.116/0.098/0.094/0.155/0.09/0.053/0.041/0
4. CUMMD=1527.76
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
5. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.11/0.092/0.094/0.155/0.09/0.053/0.041/0
5. CUMMD=1522.51
TPFMQA=0.273/0.213/0.11/0.11/0.104/0.094/0.155/0.09/0.053/0.041/0
5. CUMMD=1528.32
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
```

```
6. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.088/0.155/0.09/0.053/0.041/0
6. CUMMD=1522.28
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.1/0.155/0.09/0.053/0.041/0
6. CUMMD=1528.55
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
7. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.149/0.09/0.053/0.041/0
7. CUMMD=1522.2
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.161/0.09/0.053/0.041/0
7. CUMMD=1528.63
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
8. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.084/0.053/0.041/0
8. CUMMD=1522.18
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.096/0.053/0.041/0
8. CUMMD=1528.65
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.053/0.041/0
9. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.041/0
10. CUMMD=1522.17
TPFMQA=0.273/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
1. CUMMD=1522.17

****************   Start of a new cycle   ****************

TPFMQA=0.268/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
1. CUMMD=1523.12
TPFMQA=0.279/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
1. CUMMD=1527.52
TPFMQA=0.274/0.213/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
2. CUMMD=1522.17
TPFMQA=0.274/0.208/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
2. CUMMD=1523.77
TPFMQA=0.274/0.219/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
2. CUMMD=1526.32
TPFMQA=0.274/0.214/0.11/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
3. CUMMD=1522.17
TPFMQA=0.274/0.214/0.105/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
3. CUMMD=1523.14
TPFMQA=0.274/0.214/0.116/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
3. CUMMD=1526.3
TPFMQA=0.274/0.214/0.111/0.11/0.098/0.094/0.155/0.09/0.047/0.035/0
4. CUMMD=1522.17
TPFMQA=0.274/0.214/0.111/0.105/0.098/0.094/0.155/0.09/0.047/0.035/0
5. CUMMD=1522.06
TPFMQA=0.274/0.214/0.111/0.105/0.093/0.094/0.155/0.09/0.047/0.035/0
5. CUMMD=1522.34
TPFMQA=0.274/0.214/0.111/0.105/0.104/0.094/0.155/0.09/0.047/0.035/0
5. CUMMD=1528.21
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.094/0.155/0.09/0.047/0.035/0
6. CUMMD=1522.06
```

```
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.089/0.155/0.09/0.047/0.035/0
6. CUMMD=1522.09
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.1/0.155/0.09/0.047/0.035/0
6. CUMMD=1528.38
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.095/0.155/0.09/0.047/0.035/0
7. CUMMD=1522.06
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.095/0.15/0.09/0.047/0.035/0
8. CUMMD=1522.01
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.095/0.15/0.085/0.047/0.035/0
9. CUMMD=1522.01
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.035/0
10. CUMMD=1522.01
TPFMQA=0.274/0.214/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
1. CUMMD=1522.01

****************   Start of a new cycle   ****************

TPFMQA=0.27/0.214/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
1. CUMMD=1522.77
TPFMQA=0.279/0.214/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
2. CUMMD=1521.06
TPFMQA=0.279/0.21/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
2. CUMMD=1522.49
TPFMQA=0.279/0.219/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
2. CUMMD=1525.77
TPFMQA=0.279/0.215/0.111/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
3. CUMMD=1521.06
TPFMQA=0.279/0.215/0.107/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
3. CUMMD=1521.91
TPFMQA=0.279/0.215/0.116/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
3. CUMMD=1525.68
TPFMQA=0.279/0.215/0.112/0.105/0.099/0.095/0.15/0.085/0.042/0.03/0
4. CUMMD=1521.06
TPFMQA=0.279/0.215/0.112/0.101/0.099/0.095/0.15/0.085/0.042/0.03/0
5. CUMMD=1520.99
TPFMQA=0.279/0.215/0.112/0.101/0.095/0.095/0.15/0.085/0.042/0.03/0
5. CUMMD=1521.21
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.095/0.15/0.085/0.042/0.03/0
6. CUMMD=1520.71
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.091/0.15/0.085/0.042/0.03/0
6. CUMMD=1520.78
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.1/0.15/0.085/0.042/0.03/0
6. CUMMD=1527.1
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.096/0.15/0.085/0.042/0.03/0
7. CUMMD=1520.71
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.096/0.146/0.085/0.042/0.03/0
8. CUMMD=1520.71
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.096/0.146/0.081/0.042/0.03/0
9. CUMMD=1520.71
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
10. CUMMD=1520.71
TPFMQA=0.279/0.215/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
```

```
1. CUMMD=1520.71

****************   Start of a new cycle   ****************

TPFMQA=0.275/0.215/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
1. CUMMD=1521.47
TPFMQA=0.284/0.215/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
2. CUMMD=1519.77
TPFMQA=0.284/0.211/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
2. CUMMD=1521.19
TPFMQA=0.284/0.22/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
2. CUMMD=1524.47
TPFMQA=0.284/0.216/0.112/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
3. CUMMD=1519.77
TPFMQA=0.284/0.216/0.108/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
3. CUMMD=1520.62
TPFMQA=0.284/0.216/0.117/0.101/0.104/0.096/0.146/0.081/0.038/0.03/0
3. CUMMD=1524.39
TPFMQA=0.284/0.216/0.113/0.101/0.104/0.096/0,146/0.081/0.038/0.03/0
4. CUMMD=1519.77
TPFMQA=0.284/0.216/0.113/0.097/0.104/0.096/0.146/0.081/0.038/0.03/0
5. CUMMD=1519.69
TPFMQA=0.284/0.216/0.113/0.097/0.1/0.096/0.146/0.081/0.038/0.03/0
5. CUMMD=1519.92
TPFMQA=0.284/0.216/0.113/0.097/0.109/0.096/0.146/0.081/0.038/0.03/0
5. CUMMD=1525.89
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.096/0.146/0.081/0.038/0.03/0
6. CUMMD=1519.69
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.092/0.146/0.081/0.038/0.03/0
6. CUMMD=1519.71
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.101/0.146/0.081/0.038/0.03/0
6. CUMMD=1526.03
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.146/0.081/0.038/0.03/0
7. CUMMD=1519.69
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.142/0.081/0.038/0.03/0
8. CUMMD=1519.63
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.142/0.077/0.038/0.03/0
8. CUMMD=1519.64
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.142/0.086/0.038/0.03/0
8. CUMMD=1526.11
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.142/0.082/0.038/0.03/0
9. CUMMD=1519.63
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
10. CUMMD=1519.63
TPFMQA=0.284/0.216/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
1. CUMMD=1519.63

****************   Start of a new cycle   ****************

TPFMQA=0.28/0.216/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
1. CUMMD=1520.38
TPFMQA=0.289/0.216/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
```

```
1. CUMMD=1525.16
TPFMQA=0.285/0.216/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
2. CUMMD=1519.63
TPFMQA=0.285/0.212/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
2. CUMMD=1520.87
TPFMQA=0.285/0.221/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
2. CUMMD=1524.14
TPFMQA=0.285/0.217/0.113/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
3. CUMMD=1519.63
TPFMQA=0.285/0.217/0.109/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
3. CUMMD=1520.29
TPFMQA=0.285/0.217/0.118/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
3. CUMMD=1524.07
TPFMQA=0.285/0.217/0.114/0.097/0.105/0.097/0.142/0.082/0.034/0.03/0
4. CUMMD=1519.63
TPFMQA=0.285/0.217/0.114/0.093/0.105/0.097/0.142/0.082/0.034/0.03/0
5. CUMMD=1519.36
TPFMQA=0.285/0.217/0.114/0.093/0.101/0.097/0.142/0.082/0.034/0.03/0
5. CUMMD=1519.59
TPFMQA=0.285/0.217/0.114/0.093/0.11/0.097/0.142/0.082/0.034/0.03/0
5. CUMMD=1525.56
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.097/0.142/0.082/0.034/0.03/0
6. CUMMD=1519.36
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.093/0.142/0.082/0.034/0.03/0
6. CUMMD=1519.38
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.102/0.142/0.082/0.034/0.03/0
6. CUMMD=1525.7
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.098/0.142/0.082/0.034/0.03/0
7. CUMMD=1519.36
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.098/0.138/0.082/0.034/0.03/0
8. CUMMD=1519.31
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.098/0.138/0.078/0.034/0.03/0
9. CUMMD=1519.31
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
10. CUMMD=1519.31
TPFMQA=0.285/0.217/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
1. CUMMD=1519.31

****************   Start of a new cycle   ****************

TPFMQA=0.281/0.217/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
1. CUMMD=1520.06
TPFMQA=0.29/0.217/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
2. CUMMD=1518.36
TPFMQA=0.29/0.213/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
2. CUMMD=1519.78
TPFMQA=0.29/0.222/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
2. CUMMD=1523.06
TPFMQA=0.29/0.218/0.114/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
3. CUMMD=1518.36
TPFMQA=0.29/0.218/0.11/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
3. CUMMD=1519.2
```

TPFMQA=0.29/0.218/0.119/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
3. CUMMD=1522.99
TPFMQA=0.29/0.218/0.115/0.093/0.106/0.098/0.138/0.078/0.03/0.03/0
4. CUMMD=1518.36
TPFMQA=0.29/0.218/0.115/0.089/0.106/0.098/0.138/0.078/0.03/0.03/0
5. CUMMD=1518.28
TPFMQA=0.29/0.218/0.115/0.089/0.102/0.098/0.138/0.078/0.03/0.03/0
5. CUMMD=1518.5
TPFMQA=0.29/0.218/0.115/0.089/0.111/0.098/0.138/0.078/0.03/0.03/0
5. CUMMD=1524.47
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.098/0.138/0.078/0.03/0.03/0
6. CUMMD=1518.28
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.094/0.138/0.078/0.03/0.03/0
6. CUMMD=1518.29
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.103/0.138/0.078/0.03/0.03/0
6. CUMMD=1524.61
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.099/0.138/0.078/0.03/0.03/0
7. CUMMD=1518.28
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.099/0.134/0.078/0.03/0.03/0
8. CUMMD=1518.22
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
9. CUMMD=1518.22
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
10. CUMMD=1518.22
TPFMQA=0.29/0.218/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
1. CUMMD=1518.22

**************** Start of a new cycle ****************

TPFMQA=0.286/0.218/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
1. CUMMD=1518.97
TPFMQA=0.295/0.218/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
2. CUMMD=1517.27
TPFMQA=0.295/0.214/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
2. CUMMD=1518.69
TPFMQA=0.295/0.223/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
2. CUMMD=1521.97
TPFMQA=0.295/0.219/0.115/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
3. CUMMD=1517.27
TPFMQA=0.295/0.219/0.111/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
3. CUMMD=1518.1
TPFMQA=0.295/0.219/0.12/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
3. CUMMD=1521.91
TPFMQA=0.295/0.219/0.116/0.089/0.107/0.099/0.134/0.074/0.03/0.03/0
4. CUMMD=1517.27
TPFMQA=0.295/0.219/0.116/0.085/0.107/0.099/0.134/0.074/0.03/0.03/0
5. CUMMD=1517.18
TPFMQA=0.295/0.219/0.116/0.085/0.103/0.099/0.134/0.074/0.03/0.03/0
5. CUMMD=1517.41
TPFMQA=0.295/0.219/0.116/0.085/0.112/0.099/0.134/0.074/0.03/0.03/0
5. CUMMD=1523.37
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.099/0.134/0.074/0.03/0.03/0

```
6. CUMMD=1517.18
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.095/0.134/0.074/0.03/0.03/0
6. CUMMD=1517.2
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.104/0.134/0.074/0.03/0.03/0
6. CUMMD=1523.51
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.1/0.134/0.074/0.03/0.03/0
7. CUMMD=1517.18
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.1/0.13/0.074/0.03/0.03/0
8. CUMMD=1517.12
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.1/0.13/0.07/0.03/0.03/0
8. CUMMD=1517.13
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
9. CUMMD=1517.12
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
10. CUMMD=1517.12
TPFMQA=0.295/0.219/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
1. CUMMD=1517.12

****************   Start of a new cycle   ****************

TPFMQA=0.291/0.219/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
1. CUMMD=1517.88
TPFMQA=0.3/0.219/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
1. CUMMD=1522.64
TPFMQA=0.296/0.219/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
2. CUMMD=1517.12
TPFMQA=0.296/0.215/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
2. CUMMD=1518.35
TPFMQA=0.296/0.224/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
2. CUMMD=1521.63
TPFMQA=0.296/0.22/0.116/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
3. CUMMD=1517.12
TPFMQA=0.296/0.22/0.112/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
3. CUMMD=1517.76
TPFMQA=0.296/0.22/0.121/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
3. CUMMD=1521.57
TPFMQA=0.296/0.22/0.117/0.085/0.108/0.1/0.13/0.079/0.03/0.03/0
4. CUMMD=1517.12
TPFMQA=0.296/0.22/0.117/0.081/0.108/0.1/0.13/0.079/0.03/0.03/0
5. CUMMD=1516.84
TPFMQA=0.296/0.22/0.117/0.081/0.104/0.1/0.13/0.079/0.03/0.03/0
5. CUMMD=1517.06
TPFMQA=0.296/0.22/0.117/0.081/0.113/0.1/0.13/0.079/0.03/0.03/0
5. CUMMD=1523.03
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.1/0.13/0.079/0.03/0.03/0
6. CUMMD=1516.84
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.096/0.13/0.079/0.03/0.03/0
6. CUMMD=1516.85
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.105/0.13/0.079/0.03/0.03/0
6. CUMMD=1523.16
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.101/0.13/0.079/0.03/0.03/0
7. CUMMD=1516.84
```

```
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.101/0.126/0.079/0.03/0.03/0
8. CUMMD=1516.78
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
9. CUMMD=1516.78
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
10. CUMMD=1516.78
TPFMQA=0.296/0.22/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
1. CUMMD=1516.78

****************   Start of a new cycle   ****************

TPFMQA=0.292/0.22/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
1. CUMMD=1517.54
TPFMQA=0.301/0.22/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
2. CUMMD=1515.83
TPFMQA=0.301/0.216/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
2. CUMMD=1517.24
TPFMQA=0.301/0.225/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
2. CUMMD=1520.52
TPFMQA=0.301/0.221/0.117/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
3. CUMMD=1515.83
TPFMQA=0.301/0.221/0.113/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
3. CUMMD=1516.65
TPFMQA=0.301/0.221/0.122/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
3. CUMMD=1520.47
TPFMQA=0.301/0.221/0.118/0.081/0.109/0.101/0.126/0.075/0.03/0.03/0
4. CUMMD=1515.83
TPFMQA=0.301/0.221/0.118/0.077/0.109/0.101/0.126/0.075/0.03/0.03/0
5. CUMMD=1515.73
TPFMQA=0.301/0.221/0.118/0.077/0.105/0.101/0.126/0.075/0.03/0.03/0
5. CUMMD=1522.43
TPFMQA=0.301/0.221/0.118/0.077/0.114/0.101/0.126/0.075/0.03/0.03/0
5. CUMMD=1521.92
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.101/0.126/0.075/0.03/0.03/0
6. CUMMD=1515.73
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.097/0.126/0.075/0.03/0.03/0
6. CUMMD=1522.22
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.106/0.126/0.075/0.03/0.03/0
6. CUMMD=1522.06
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.126/0.075/0.03/0.03/0
7. CUMMD=1515.73
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.122/0.075/0.03/0.03/0
7. CUMMD=1522.14
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.131/0.075/0.03/0.03/0
7. CUMMD=1522.11
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.127/0.075/0.03/0.03/0
8. CUMMD=1515.73
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
9. CUMMD=1515.66
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
10. CUMMD=1515.66
TPFMQA=0.301/0.221/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
```

```
1. CUMMD=1515.66

****************   Start of a new cycle   ****************

TPFMQA=0.297/0.221/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
1. CUMMD=1516.42
TPFMQA=0.306/0.221/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
1. CUMMD=1521.16
TPFMQA=0.302/0.221/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
2. CUMMD=1515.66
TPFMQA=0.302/0.217/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
2. CUMMD=1516.88
TPFMQA=0.302/0.226/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
2. CUMMD=1520.16
TPFMQA=0.302/0.222/0.118/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
3. CUMMD=1515.66
TPFMQA=0.302/0.222/0.114/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
3. CUMMD=1516.28
TPFMQA=0.302/0.222/0.123/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
3. CUMMD=1520.12
TPFMQA=0.302/0.222/0.119/0.077/0.11/0.102/0.127/0.071/0.03/0.03/0
4. CUMMD=1515.66
TPFMQA=0.302/0.222/0.119/0.073/0.11/0.102/0.127/0.071/0.03/0.03/0
4. CUMMD=1521.84
TPFMQA=0.302/0.222/0.119/0.082/0.11/0.102/0.127/0.071/0.03/0.03/0
4. CUMMD=1520.59
TPFMQA=0.302/0.222/0.119/0.078/0.11/0.102/0.127/0.071/0.03/0.03/0
5. CUMMD=1515.66
TPFMQA=0.302/0.222/0.119/0.078/0.106/0.102/0.127/0.071/0.03/0.03/0
5. CUMMD=1521.37
TPFMQA=0.302/0.222/0.119/0.078/0.115/0.102/0.127/0.071/0.03/0.03/0
5. CUMMD=1520.87
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.102/0.127/0.071/0.03/0.03/0
6. CUMMD=1515.66
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.098/0.127/0.071/0.03/0.03/0
6. CUMMD=1521.17
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.107/0.127/0.071/0.03/0.03/0
6. CUMMD=1521.01
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.127/0.071/0.03/0.03/0
7. CUMMD=1515.66
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.123/0.071/0.03/0.03/0
7. CUMMD=1521.09
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.132/0.071/0.03/0.03/0
7. CUMMD=1521.06
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.071/0.03/0.03/0
8. CUMMD=1515.66
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.067/0.03/0.03/0
8. CUMMD=1521.08
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.076/0.03/0.03/0
8. CUMMD=1521.07
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.03/0.03/0
9. CUMMD=1515.66
```

```
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.03/0.03/0
9. CUMMD=1521.07
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.039/0.03/0
9. CUMMD=1521.07
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.035/0.03/0
10. CUMMD=1515.66
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.035/0.03/0
10. CUMMD=1521.07
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.035/0.039/0
10. CUMMD=1521.07
TPFMQA=0.302/0.222/0.119/0.078/0.111/0.103/0.128/0.072/0.035/0.033/0
10. CUMMD=1515.66

The best CUMMD is: 1515.66
```

LIST OF REFERENCES

1. Shannon, R. E., *Systems Simulation The Art And Science*, Prentice-Hall, Inc., 1975.

2. Abdel-Hamid, T. K., and Sivasankaran, T. R., "Incorporating expert system technology into simulation modeling: An expert-simulator for project management," *AI and Simulation 1988*, 1988.

3. Shannon, R. E., Mayer, R., and Adelsberger, H. H., "Expert systems and simulation," *Simulation*, v.44, June 1985.

4. O´Keefe, R., "Simulation and Expert Systems - A Taxonomy and some Examples," *Simulation*, v.46, January 1986.

5. Abdel-Hamid, T. K., and Madnick, S. E., "Modeling the Dynamics of Software Project Management", CISR WP No. 163, MIT, Cambridge, MA, September 1987.

6. Abdel-Hamid, T. K., "The Economics of Software Quality Assurance: A Simulation-Based Case Study", *MIS Quarterly*, v.12, September 1988.

7. Marcus, C., *Prolog Programming, Applications for Database Systems, Expert Systems and Natural Language Systems*, Addison Wesley, Inc., 1986.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Dr. Tarek K. Abdel-Hamid, Code 54Ah Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943-5000	5
4.	Curricular Officer, Code 37 Computer Technology Programs Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Commandant (G-PTE-1) U. S. Coast Guard 2100 2nd Avenue, S.W. Washington, DC 20593	2
6.	Dr. Magdi Kamel, Code 54Ka Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943-5000	1
7.	LT Frank H. Leidy P.O. Box 3502 Alexandria, VA 22302	2

Thesis
L4757 Leidy
c.1 Design and development of an expert system based quality assurance module for the Dynamo Model of software project management.