Reporting AT&T Quality Assurance Results using UNIX® Tools

John A. Conte

AT&T Technologies, Inc.

ABSTRACT

A new method of generating and distributing AT&T Quality Assurance Reports has been developed. From 1980 through 1986 AT&T Quality Assurance Report masters were generated at AT&T Bell Labs, Holmdel, New Jersey using a mainframe method of graphics production, reproduced in Atlanta, Georgia, and distributed using the U.S. Postal Service. This method was replaced in January 1987, using a combination of several UNIX® software tools including:

- AT&T DOCUMENTER'S WORKBENCH™ (DWB) Software (grap, tbl, and troff) to produce the reporting graphics.
- AT&T 5620 DOT-MAPPED DISPLAY (DMD) Software (proof) and a Teletype® 5620 terminal to preview and develop improved graphical displays of data.
- The INFORMIX™ relational database system to manage the large amounts of data and generate the source for processing by the AT&T DOCUMENTER'S WORKBENCH™ (DWB) Software.
- UNIX® networking facilities (uuto) to distribute the DWB source code to multiple locations for printing on laser printers.

The new method is the product of several UNIX® tools each of which is productive in its own right. The combination was extremely successful is solving a real problem. Using these four UNIX® tools it was possible to produce AT&T Quality Assurance Reports with higher quality graphics and drastically reduced production time.

About the Author

John A. Conte is a Senior Engineer in the AT&T Network Systems Quality Department in Morristown, New Jersey. Mr. Conte is currently responsible for development of a UNIX replacement for the Quality Assurance Information System and has a special interest in the continued development of AT&T Documenter's Workbench™ software. He joined AT&T at the Kansas City Works of Western Electric in 1968 and has B.S. and M.S. in Industrial Engineering from the University of Missouri at Columbia.

As reprinted from the proceedings
of the
1987 APEX Technical Seminar
February, 1987

On December 30, 1989 John A. Conte retired from AT&T after twenty-one years of service. He is currently employed by DSC Communications Corporation as Manager of Quality Reporting Systems. His mailing address at DSC is MS-316, 1000 Coit Road, Plano, Texas, 75075. His telephone number is (214) 519-6917. He accepts electronic mail at attmail!jconte.

Reporting AT&T Quality Assurance Results using UNIX® Tools

John A. Conte

AT&T Technologies, Inc.

1. Introduction

AT&T Quality Assurance checks samples of completed products for conformance to quality standards. The results of those quality audits are reported to upper management in AT&T Quality Assurance Reports.

The reports contain charts of statistical confidence intervals about the true quality level computed using empirical Bayes statistics. These charts are referred to as QMP (Quality Measurement Plan) charts[1]. Quality Assurance audit results are collected from over fifty facilities and entered into a database at AT&T Bell Labs, Holmdel, New Jersey. At the end of each rating period the data is summarized and report masters complete with graphics are generated. The report masters are generated using the 1980 technology of "mainframe" graphics generation. The report masters are then sent to a printing facility in Atlanta, Georgia, for reproduction. Distribution is made using the U.S. Postal Service.

A new method of generating and distributing AT&T Quality Assurance Reports has been developed. The new method uses the following UNIX® tools*.

- AT&T DOCUMENTER'S WORKBENCH™ (DWB) Software** (grap, tbl, and troff) to produce reporting graphics of higher quality than the present method.
- AT&T 5620 DOT-MAPPED DISPLAY (DMD) Software (proof) and a Teletype® 5620 terminal*** to preview and develop improved graphical displays of data.
- The INFORMIX™ relational database system**** to manage the large amounts of data and generate the source for processing by AT&T DOCUMENTER'S WORKBENCH™ (DWB).
- UNIX® networking facilities (uuto) to distribute the DWB source code to multiple locations for printing on laser printers.

The present method of collecting the audit data and computing the statistics will not be changed by this new reporting method. Data will still be entered into the database located in Holmdel and at the end of each rating period the summary statistics will be copied to the INFORMIX™ database. Investigations into using a UNIX® database for the data entry and data computations are currently in progress.

The new method is the product of several UNIX® tools each of which is productive in its own right. The combination was extremely successful is solving a real problem. This paper makes an effort to describe each of the tools as they were used to solve the real problem of timely reporting of AT&T Quality Assurance results. Readers will possibly find a solution to one of their own problems with the use of one or more of these tools.

2. AT&T DOCUMENTER'S WORKBENCH™ (DWB) Software

AT&T DOCUMENTER'S WORKBENCH™ Software provides tools of unusual precision for sophisticated text processing. The software package is the result of unbundling and repackaging the **nroff** and **troff** commands of the System V release of UNIX®. Version 2.0 of AT&T

---

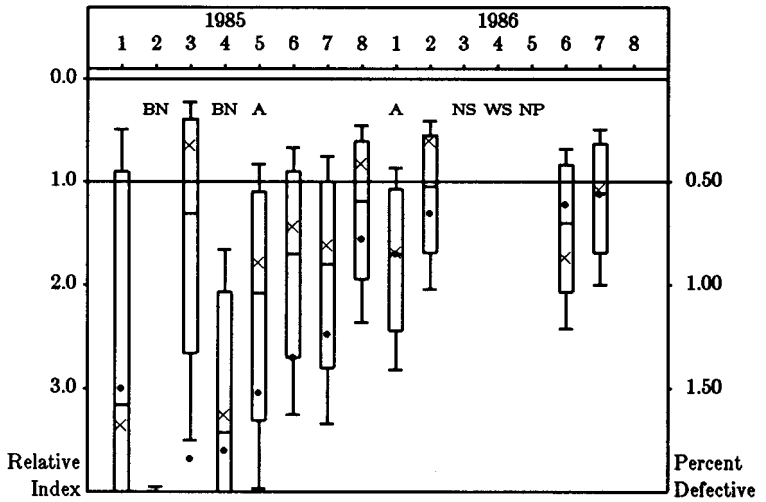
* UNIX is a registered trademark of AT&T.

** DOCUMENTER'S WORKBENCH is a trademark of AT&T.

*** Teletype is a registered trademark of AT&T.

**** INFORMIX is a trademark of Informix, Inc.

DOCUMENTER'S WORKBENCH™ Software was released in May 1986 with the significant addition of the **grap** command. **Grap** is a preprocessor language that produces input for the **troff** text processor. Most of the code processed by **grap** is passed through unchanged, but statements between **.G1** and **.G2** are translated into commands that draw graphs. A complete description of AT&T DOCUMENTER'S WORKBENCH™ Software is available in the User's Guide[2] and the Reference Manual[3]. I first learned about **grap** from a technical memorandum[4] in August 1985. Within three days I was able to generate a hypothetical Quality Assurance QMP graph similar to the following.



Class XXXTT

Title of Hypothetical Class

Period	8607
Output	49114
Sample	927
Equivalent Expectancy	4.65
Assessed	5
Expected	4.63
Current Index	1.08
Best Measure	1.11

Figure 1. QMP Graph Drawn Using **grap, tbl**, and **troff** (DWB) Code

The complete **DWB** source code for this graph is contained in Appendix A.

2.1 troff Format Control Lines

Lines of code that begin with a "." (dot) are common **troff** format control lines. A short introduction to a few of these format control lines follows:

- **.ne 3.0i** - (line 101) need 3.0 inches of vertical space. If 3.0 inches of vertical space does not exist on the current page then begin a new page.
- **.mk R** - (line 102) mark the current vertical place in register R.
- **.ps 10** - set the point size of the text. A point is equal to 1/72 of an inch. The class identifier and title are set with a point size of 10 (line 103) and the scale labels "Relative Index" and "Percent Defective" are set with a point size of 8 (line 112).
- **.vs 12** - (line 104) set the size of the vertical spacing to 12 points (12/72 of an inch).
- **.sp** - (line 105) vertically space one line in the down direction.
- **.in 4.2i** - (line 106) indent the following text (lines 108 and 110) 4.2 inches from the left margin.
- **.ce** - (line 107) center the next line of text (line 108) between the current indentation and the right margin.
- **.rt** - (line 111) return upward to the vertical place marked in register R in line 102.

2.2 grap Code

Beginning with the **.G1** (line 116) you will find the following examples of **grap** code:

- **define PLOT Z** - (line 119) defines the macro **PLOT** between the letter **Z** and the next **Z** (line 136).
- **frame ht 2.2 wid 3.0** - (line 122) defines the basic dimensions of the graph at 2.2 inches high and 3.0 inches wide.

- **coord x 0,17 y 0.1,-4** - (line 123) defines a graph with a coordinate system from 0 to 17 in the x direction and 0.1 to -4 in the y direction. Note that the rating periods (x-coordinate) are transformed from a notation of 8501, 8502, ..., 8508, 8601, 8602, ..., 8608 to 1 to 16. This was done using the following code on the $2 parameter:

  ```
  if $2 > 8600 then K xc =( $2 - 8600 + 8 ) K else K xc = ($2 - 8500) K
  ```

  Setting the x coordinate system from 0 to 17 and plotting only values 1 to 16 allows a margin at the beginning and end. Note also that quality charts are typically plotted in the negative y direction.

- **ticks left out at 0 "0.0", -1 "1.0", -2 "2.0", -3 "3.0"** - (line 127) create tick marks on the left side of the chart in the out direction at positions 0, -1, -2, and -3 and label them 0.0, 1.0, 2.0, and 3.0 respectively.

- **"Index" rjust at -0.2,-4.00** - (line 129) place the text "Index" right justified at the coordinate position of -0.2 (x direction) and -4.00 (y direction). This equates to the lower left hand corner of the graph when the graph was created with a coordinate system of **x 0,17 y 0.1,-4**.

- **line from 0,-1 to 17,-1** - (line 135) draw a line from the coordinate pair 0,-1 to 17,-1. This is the horizontal line on the graph that identifies the standard quality level.

- **bm=-$3** - (line 160) assign the negative value of the third parameter passed with the `QMPBOX($1,$2,$3,$4,$5,$6,$7,$8,$9)` data line to the variable **bm**. Note that line 160 falls within the definition of the **QMPBOX** macro (between lines 139 and 178).

- **plot "$3" size -2 at xc,-0.3 Z** - (line 184) print the value of the third parameter passed with the `LABEL4($1,$2,$3)` data line two points smaller than the current text point size at coordinate position xc (as determined by $2 period transformation) and -0.3 y direction. Note that line 184 falls within the definition of the **LABEL4** macro (lines 181 to 184).

2.3 tbl Code

A table is a systematic arrangement of data organized in rows and columns. Tables can be specified in a document by using **tbl** another **troff** preprocessor. **Tbl** provides a high level declarative language for specifying tables in a simple and straightforward way[5]. The **tbl** code used included:

- **.TS** - (line 216) **T**able **S**tart macro.

- **tab(:);** - (line 217) definition of the colon (:) character that will separate the data items. The semicolon character is used to define the end of the global attributes of the table.

- **lw(21) rw(8).** - (line 218) define a table with two columns of data. The left hand column is to be 21 characters wide and the data is to be left justified. The right hand column is to be 8 characters wide and the data will be right justified. The period at the end of the format line defines the end of the table specification.

- **Period:8607** - (line 219) data that will be left and right justified in the table.

- **.TE** - (line 227) **T**able **E**nd macro.

3. AT&T 5620 Dot-Mapped Display (DMD) Software

The Teletype® 5620 (DMD) terminal is an intelligent high resolution graphics terminal designed especially for use with the UNIX® System V[6]. The DMD Text/Graphics Package[7] **proof** command that allows **troff** output to be viewed on the DMD screen is of particular value to me. The **proof** command allows for a minimal turn around time (about 15 seconds for the graph shown above) when checking for errors, format details, and developing graphics. Trying various configurations of charting is my style and the improvement of the graphics was dependent on repeated trials. Because no printer was available locally the DMD terminal was invaluable since hard copy turnaround time was 24 to 72 hours. When a hard copy was wanted it was generated by electronically sending the source file from Aurora, Colorado to either Chicago, Illinois, or Morristown, New Jersey, for printing.

4. QA Reports without INFORMIX™ Software

I produced reports with forty to sixty charts using only DWB and DMD Software. The basis for the charts was the QMPBOX data line. I was able to generate **grap** data lines easily from the database in Holmdel and transfer them to a file on the UNIX® machine in Aurora, Colorado.

```
QMPBOX(MVXXXTT,8607,1.11,1.12,1.08,0.49,0.63,1.69,2.00)
```

Figure 2. Data Line used by **grap**

Note that the first parameter in the sample data line is the chart identifier. I was able to **grep** on the chart identifier to produce three charts, MVXXXTT, MVXXXT1, and MVXXXT2, on a single page using the same chart macro file (chart.macro) and a single data file (chart.data).

```
.bp
.G1
sh ? cat chart.macros > tmp ?
sh ? grep "MVXXXTT" chart.data >> tmp ?
copy "tmp"
.G2
.in 0
.G1
sh ? cat chart.macros > tmp ?
sh ? grep "MVXXXT1" chart.data >> tmp ?
copy "tmp"
.G2
.in 0
.G1
sh ? cat chart.macros > tmp ?
sh ? grep "MVXXXT2" chart.data >> tmp ?
copy "tmp"
.G2
```

Figure 3. grap Code Used Before Introduction of Database Software

Two other examples of **grap** code shown in this figure are as follows:

- **sh ? ... ?** - This code allows an escape to the UNIX® shell to execute a command, such as **grep** or **cat**, between the delimiter **?** character.
- **copy "tmp"** - This code processes whatever is in the file **tmp** as **grap** code.

It was found that the generation of similar source for a complete AT&T Quality Assurance report with 600 charts was a tedious job.

5. QA Reports with INFORMIX™ Software

Database management systems (DBMS) are general purpose programs that dramatically reduce the time necessary to computerize an application. They allow you to enter, store, manipulate, and retrieve information in a database. A DBMS like INFORMIX™ provides an interactive access to your database and provides easy ways to create printed reports[8].

INFORMIX™ type software is designed for storing large amounts of data. But storage of data did not seem to be a problem. The problem seemed to be how to generate the DWB source code for hundreds of charts when not all charts had identical format. To understand the solution it is necessary to understand the design of the database.

5.1 Database Design

INFORMIX™ is a relational DBMS. As such it has a database defined with two or more files each related to one another with a common field. The name of my database was **qmpfile** and it was composed of three files.

The **loc_titles** file contained information about each location including:

- **loc_code** the two-character location code.
- **loc_name** the name of each AT&T manufacturing facility.

The **qmptitles** file contained information about each chart to be printed in the reports including:

- **lc_code** the two-character location code related to the **loc_code** in the **loc_titles** file.
- **chartid1** the seven-character identifier of the chart.
- **rateunit** is a single digit used to identify the type of chart as either demerit, percent defective, or parts per million.
- **rateshop** a five-character organization number used to separate charts within a given location code.
- **chart_title** the title of the chart.

The **qmprates** file contained the QMP data for each period including:

- **chartid2** the seven-character identifier related to the **chartid1** in the **qmptitles** file.
- **period** the four-character period identifier of the record.
- **universe, sample** data values about the product category described by the individual record.
- **assessed, expected, variance** data values used to compute the individual confidence level plot.
- **curr_index, bestmeas, long_runa, pctl99, pctl95, pctl5, pctl1** the seven data values that define each individual confidence interval plot.
- **adj** a single character flag that generates a two-character notation instead of an individual confidence interval plot indicating either No Production, Work Stoppage, or No Sample, if one of these conditions exist.

The value of each field in the database was output as either a parameter in a **grap** data line, text, data for a table, or a test parameter for the ACE™ report writer.

5.2 Introduction to ACE™

ACE™****** is the INFORMIX™ "report writer" program for producing reports from the database. It is a general purpose report writer.

The ACE™ code for generating the **DWB** source code for three QMP charts per page for a single location code is given in Appendix B. The ACE™ code contains sections as follows:

- **define** section (lines 302 to 309) where I defined parameters and variables. The parameter **loc_id** will be passed from the UNIX® shell and will contain the two-character location code. The variables **ee, e1, e2, pdef, pdef2,** and **pdef3** will be used to compute the equivalent expectancy and absolute scale values. The variable **quote** was defined to allow the syntax of the ACE™ **print** statement to print a "double quote."
- **output** section (lines 310 to 313) where I decided to define my output as a file and not a report by setting the **top margin**, **bottom margin**, and **left margin** to zero length and directed the output to a file named **data.tmp** using the **report to** statement.
- **read** section (lines 314 to 317) that selects data from the database by relating the three files with a **joining** clause **joining loc_code = lc_code and chartid1 = chartid2** and restricting the search to a passed location parameter **loc_id**.

****** ACE is a trademark of Informix, Inc.

- **sort** section (line 318) that sorts the retrieved data by a defined order of **loc_code, rateshop, chartid2,** and **period**.
- **format** section (lines 319 to 438) that normally produces printed output, but in this application, **DWB** source code.

Within the **format** section the following types of statements are found:

- **before group of chartid2** - (line 345) Before the reading of the next record with a new **chartid2** value do the following (lines 346 to 365).
- **print ".mk R"** - (line 347) Print the literal **.mk R** in the output stream. Since each **print** statement creates a newline and the left margin has been defined as zero lenght, this statement put **.mk R** on the next line beginning in column one.
- **print "Class ", rateclas** - (line 353) Print in the output stream the literal **Class**, a space character, and the value of the variable **rateclas**.
- **let pdef = (expected/sample) * 100** - (last half of line 386) Assign the value of the variable **expected** divided by the variable **sample** times **100** to the variable **pdef**.
- **if (rateunit="1") then begin ... end** - (lines 390 to 396) If the condition **rateunit="1"** is true then execute the statements between the word **begin** and **end**.
- **print " Best Measure:", bestmeas using "#######.##"** - (line 433) Print in the output stream the literal **Best Measure:** and the value of the **bestmeas** for the most recent record (after group of chartid2) with a numeric format of two decimals to the right.

Thus the **print** statements produced the **DWB** source code from a test of values in the database, and the values of various database fields, and literals.

5.3 Passed Report Parameters

The ACE™ reporting program is invoked from the shell as follows:

```
$ acego -q combo "MV"
```

Acego is the ACE™ report program command. The **-q** option tells the program to proceed in the quiet mode, by not processing progress messages. **Combo** is the program name and **"MV"** is a positional parameter being passed to the ACE™ program to restrict the report to a specific location using the **loc_code** field.

5.4 Computing Equivalent Expectancy

The UNIX® database was an identical copy of the original database in Holmdel and did not contain the value of the **equivalent expectancy**. The equivalent expectancy is the value of the **expectancy** squared divided by the **variance**. It was desired that the **equivalent expectancy** for the most recent period be printed next to each chart. The variables **e1, e2**, and **ee** and ACE™ code were used as follows to generate the **equivalent expectancy**.

```
if variance > 0.01 then begin
    let e1 = expected
    let e2 = variance
    let ee = (e1 * e1) / e2
    print "Equivalent Expectancy:", ee using "##########.##"
end else print "Equivalent Expectancy:", "-"
```

Figure 4. ACE™ Code Used to Compute Equivalent Expectancy

5.5 Computing an Absolute Scale

During the development process I received a request to publish the absolute scale on each chart in addition to the relative scale of 0.0, 1.0, 2.0, and 3.0. For more than seven years people had accepted the relative scale of assessed to expected without always knowing what the quality expectancy was per

unit. Since that value could now be computed and displayed on the QMP chart a survey was made to the desirability and the response was positive. The variables **pdef**, **pdef2**, and **pdef3** were used as follows to generate the absolute scale labels.

```
if ( sample > 0 ) then begin
    if ( rateunit="1" ) then let pdef = (expected/sample)
    if ( rateunit="2" ) then let pdef = (expected/sample) * 100
    if ( rateunit="3" ) then let pdef = (expected/sample) * 1000000
    let pdef2 = pdef * 2.0
    let pdef3 = pdef * 3.0
```

Figure 5. ACE™ Code Used to Compute Absolute Scale

QMP chart are not all labeled alike. Some charts should be labeled as "Demerits per Unit," others "Percent Defectives" and still others "Parts per Million." The database contained a field **rateunit** that was tested to determined this label and appropriate **grap** code was generated to label the graph on the right side with the absolute scale values. The absolute scale labels for a "Parts per Million" graph were created as follows:

```
if ( rateunit="3" ) then begin
    print "ticks right out at",
    " -1 ", quote, pdef  using "######", " PPM", quote, ",",
    " -2 ", quote, pdef2 using "######", " PPM", quote, ",",
    " -3 ", quote, pdef3 using "######", " PPM", quote
    print quote, "Parts per", quote, " ljust at 17.2, -3.74"
    print quote, "Million ", quote, " ljust at 17.2, -4.00"   end
```

Figure 6. ACE™ Code for Creating **grap** Code for Absolute Scale Labels

The addition of the absolute scale labels to the right of each chart significantly increased the perception of the true quality of each product.

6. UNIX® uuto command

The fourth significant UNIX® tool used in this new method of reporting AT&T Quality Assurance results was the **uuto** command. The **uuto** command was used to send the **DWB** source files to various locations for printing. Used as follows the uuto command will send a UNIX® file (QA3.MV.s) to another UNIX® machine (mvgpa) and user (mvesp) for printing.

```
$ uuto QA3.MV.s mvgpa!mvesp
```

As soon at the UNIX® database is loaded a UNIX® shell script is executed that creates files of **DWB** code and sends those files to more than a dozen locations. When the prospective recipients login the morning after the data entry deadline they have mail messages indicating that their files are ready for retrieval with the **uupick** command and printing on local printers.

6.1 AT&T Facilities with troff print capability

Electronic distribution of the AT&T Quality Assurance reports was a primary goal in the development of this new method. But when the project began in January, 1986 only two locations interested in receiving AT&T Quality Assurance reports were capable of printing **troff** documents. They were AT&T Bell Labs in Holmdel, New Jersey and AT&T Resource Management, Hickory Ridge Conference Center, Lisle, Illinois. During 1986 I promoted the idea of printing **troff** documents at some thirty AT&T locations. As a result the following locations installed either hardware or software to provide for the printing of **troff** documents.

- AT&T Network Systems - Morristown, New Jersey - machine **mduxf**.

- AT&T Quality Assurance - Omaha, Nebraska - machine **ohgua**.
- AT&T Technology Systems - Lee's Summit, Missouri - machine **kced**.
- AT&T Technology Systems - Allentown, Pennsylvania - machine **aluxz**.
- AT&T Network Systems - Atlanta, Georgia - machine **akgua**.
- AT&T Quality Assurance - Oklahoma City, Oklahoma - machine **qaokis**.
- AT&T Network Systems - Columbus, Ohio - machine **cbuxc**.
- AT&T Network Systems - Newark, New Jersey - machine **neuxf**.
- AT&T Training - Aurora, Colorado - machine **lobo**.
- AT&T Network System - Merrimack Valley, Massachusetts - machine **mvgpa**.
- AT&T Network Systems - Phoenix, Arizona - machine **phuxa**.
- AT&T Network Systems - Winston Salem, North Carolina - machine **we22b**.
- AT&T Technology Systems - Berkeley Heights, New Jersey - machine **bkuxf**.
- AT&T Computer Systems - Lisle, Illinois - machine **cscqaa**.
- AT&T Consumer Products - Montgomery, Illinois - machine **mgweed**.
- AT&T Network Systems Software Center - Lisle, Illinois - machine **nwuxc**.

I expect the number of AT&T locations capable of printing **troff** documents to triple in 1987.

6.2 AT&T Facilities without the grap Software

Two types of locations were found with the ability to print **troff** documents, those that had installed Version 2.0 of AT&T DOCUMENTER'S WORKBENCH™ (DWB) Software and those that had installed Version 1.0. (Remember that the significant difference between version 1.0 and 2.0 was the addition of the **grap** command.) Locations with version 1.0 were sent source files preprocessed by the **grap** command. Locations with version 2.0 were sent the source files as generated by the ACE™ report writer program.

7. Conclusion

The overall result of this UNIX® approach to reporting AT&T Quality Assurance results will be a reduction in time between summarizing the audit data to distributing the reports from four weeks to 12 hours. The quality of the graphical displays was vastly improved and additional metrics and commentaries will be displayed within the new reports.

For more information on this method of reporting quality results contact:

John A. Conte, Senior Engineer
AT&T Network Systems, Department 11MR007220
Aurora, Colorado, 80010, C-363-5761 or (303) 363-5761
ihnp4!dc003b!qahqjac

8. Acknowledgements

I am grateful to Ray Peterson, Manager Network System Quality, for encouraging and supporting me to find a better way to report AT&T Quality Assurance results.

I am grateful to Steve Richards and Michele Waltmire at the AT&T Training Center at Hickory Ridge, Lisle, Illinois for providing me with new skills that allowed me to create this new method.

I am also grateful to the many AT&T Quality Assurance personnel who assisted in the many trials of this new method including Tom Scurlock, Jim Smick, Joe Gibbs, Bob Wolkins, Hiram Johnson, Joan Deisler, Jerry Edmondson, Scott Kilgore, John Pidd, Maki Parsons, Gary Stolz, Arif Rehman, Kelly Shipman and Scott Prentice.

I must also thank the two great UNIX® system administrators of the dc003b machine, Pat Snider and Kay Bissantz, for installing and maintaining the software required.

REFERENCES

1. Wadsworth, Harrison M., Stephen, Kenneth S., and Godfrey, A. *Modern Methods for Quality Control and Improvement.* New York: John Wiley & Sons, Inc., 1986.
2. UNIX® System V AT&T DOCUMENTER'S WORKBENCH™ Software User Guide, AT&T Select Code 310-004.
3. UNIX® System V AT&T DOCUMENTER'S WORKBENCH™ Software Technical Discussion and Reference Manual, AT&T Select Code 310-005.
4. Bentley, J. L. and B. W. Kernighan 1984. **grap** - A Language for Typesetting Graphs: Tutorial and User Manual. Computing Science Technical Report No. 114, AT&T Bell Laboratories, Murray Hill, New Jersey.
5. Gehani, Narain. *Document Formatting and Typesetting on the UNIX® System.* Silicon Press, Summit, New Jersey, 1986.
6. 5620 DOT-MAPPED DISPLAY User Guide, Release 2.0, AT&T Select Code 306-140.
7. 5620 DOT-MAPPED DISPLAY Text/Graphics Guide, Release 2.0, AT&T Select Code 306-143.
8. INFORMIX Relational Database Management System USER'S MANUAL, AT&T Select Code 983-015.

Appendix A - troff, tbl and grap Source for Example Graph

```
.ne 3.0i
.mk R
.ps 10
.vs 12
.sp
.in 4.2i
.ce
Class XXXTT
.sp
Title of Hypothetical Class
.rt
.ps 8
.vs 10
.in 0
.in 0.4i
.G1
#begin macro definitions
#
define PLOT z
# PLOT- defines the frame size, user unit scaling, x-axis labeling
#  and left y-axis labeling.
frame ht 2.2 wid 3.0
coord x 0,17 y 0.1,-4
ticks top out  0.02 at 1 "1", 2 "2", 3 "3", 4 "4", 5 "5", 6 "6",
    7 "7", 8 "8", 9 "1", 10 "2", 11 "3", 12 "4", 13 "5", 14 "6",
    15 "7", 16 "8"
ticks left out at 0 "0.0", -1 "1.0", -2 "2.0", -3 "3.0"
"Relative" rjust at -0.2,-3.74
"Index" rjust at -0.2,-4.00
"1985" at 4,.55
```

```
"1986" at 12,0.55
line from 0,.7 to 17,.7
line from 0,0 to 0,.7
line from 17,0 to 17,.7
line from 0,-1 to 17,-1
line from 0,0 to 17,0 Z
# end PLOT macro
#
define QMPBOX Z
#QMPBOX macro  - defines the box plot algorithm
#  arguments:
#  $1  - chart identifier
#  $2  - rating period
#  $3  - best measure
#  $4  - long term average
#  $5  - current index
#  $6  - 99th percentile
#  $7  - 95th percentile
#  $8  -  5th percentile
#  $9  -  1st percentile
#
boxwidth=.4
#  compute the x-axis position from the rating period number
if $2 > 8600 then K xc =( $2 - 8600 + 8 ) K else K xc = ($2 - 8500) K
#
x1=xc-boxwidth/2; x2=xc+boxwidth/2
#  set internal variables to reflect the upside-down nature of index plots
p99=-$6; p95=-$7; p5=-$8; p1=-$9
ci=-$5; lt=-$4
bm=-$3
#  clip percentile values at bottom of chart (-4.0)
if $8 > 4.0 then T p5 = -4.0 T
if $9 > 4.0 then T p1 = -4.0 T
if $5 <=4.0 then T times at (xc,-$5) T
if $4 <=4.0 then T bullet at (xc,-$4) T
if $7 > 4.0 then T p95 = -4.0 T
if $6 > 4.0 then T p99 = -4.0 T
if $3 > 4.0 then T bm = -4.0 T
#  draw the box and whiskers
line from (x1,p1) to (x2,p1)
line from (x1,p5) to (x2,p5)
line from (x1,p95) to (x2,p95)
line from (x1,p99) to (x2,p99)
line from (x1,bm) to (x2,bm)
line from (x1,p5) to (x1,p95)
line from (x2,p5) to (x2,p95)
line from (xc,p5) to (xc,p1)
line from (xc,p95) to (xc,p99) Z
# end QMPBOX macro
#
define LABEL4 Z
#  LABEL4  - prints special symbols
if $2 > 8600 then K xc =( $2 - 8600 + 8 ) K else K xc = ($2 - 8500) K
plot "$3" size -2 at xc,-0.3 Z
# end LABEL4 macro
#
```

```
187  PLOT
188  QMPBOX(MVXXXTT,8501,3.16,3.00,3.36,0.49,0.90,6.53,8.68)
189  QMPBOX(MVXXXTT,8502,7.10,5.30,8.09,3.95,4.72,9.88,11.27)
190  LABEL4(MVXXXTT,8502,BN)
191  QMPBOX(MVXXXTT,8503,1.31,3.68,0.65,0.22,0.39,2.66,3.50)
192  QMPBOX(MVXXXTT,8504,3.43,3.60,3.26,1.66,2.07,5.06,5.92)
193  LABEL4(MVXXXTT,8504,BN)
194  QMPBOX(MVXXXTT,8505,2.08,3.04,1.79,0.83,1.10,3.31,3.97)
195  LABEL4(MVXXXTT,8505,A)
196  QMPBOX(MVXXXTT,8506,1.70,2.70,1.44,0.67,0.90,2.70,3.25)
197  QMPBOX(MVXXXTT,8507,1.80,2.48,1.62,0.76,1.00,2.80,3.34)
198  QMPBOX(MVXXXTT,8508,1.19,1.56,0.83,0.45,0.61,1.94,2.36)
199  QMPBOX(MVXXXTT,8601,1.70,1.70,1.69,0.87,1.07,2.44,2.82)
200  LABEL4(MVXXXTT,8601,A)
201  QMPBOX(MVXXXTT,8602,1.05,1.31,0.61,0.41,0.55,1.69,2.04)
202  LABEL4(MVXXXTT,8603,NS)
203  LABEL4(MVXXXTT,8604,WS)
204  LABEL4(MVXXXTT,8605,NP)
205  QMPBOX(MVXXXTT,8606,1.40,1.22,1.74,0.68,0.84,2.07,2.42)
206  QMPBOX(MVXXXTT,8607,1.11,1.12,1.08,0.49,0.63,1.69,2.00)
207  ticks right out at -1 " 0.50 %", -2 " 1.00 %", -3 " 1.50 %"
208  "Percent" ljust at 17.2, -3.74
209  "Defective " ljust at 17.2, -4.00
210  .G2
211  .in 4.2i
212  .ps 10
213  .vs 12
214  .rt
215  .sp 1.0i
216  .TS
217  tab(:);
218  lw(21) rw(8).
219  Period:8607
220  Output:49114
221  Sample:927
222  Equivalent Expectancy:4.65
223  Assessed:5
224  Expected:4.63
225  Current Index:1.08
226  Best Measure:1.11
227  .TE
228  .in 0
229  .rt
230  .sp 2.8i
```

Appendix B - ACE™ Code for Generating DWB Source Code

```
301 database qmpfile end

302 define param [1] loc_id type character length 2
303         variable pdef type double
304         variable pdef2 type double
305         variable pdef3 type double
306         variable ee type double
307         variable e1 type double
308         variable e2 type double
```

```
309        variable quote type character length 1    end

310 output top margin 0
311        bottom margin 0
312        left margin 0
313        report to "data.tmp"    end

314 read  into temp loc_titles qmptitles qmprates
315       joining loc_code = lc_code
316           and chartid1 = chartid2
317       where lc_code = loc_id     end

318 sort by loc_code rateshop chartid2 period ascending end

319 format

320 before group of loc_code
321 let quote="""
322 print ".pl 66"
323 print ".ll 7.0i"
324 print ".po 1.0i"
325 print ".de HD"          { define HEADER macro }
326 print "'sp"
327 print "'ft R"
328 print "'ps 10"
329 print "'vs 12"
330 print "'tl 'AT&T Technologies'Quality Assurance'",
331           loc_name without trailing spaces, "'"
332 print "'sp"
333 print ".."
334 print ".de FT"          { define FOOTER macro }
335 print "'in 0"
336 print "'ft B"
337 print ".ps 8"
338 print ".vs 10"
339 print "'ce 2"
340 print "AT&T TECHNOLOGIES, INC. - PROPRIETARY "
341 print "Use Pursuant to Company Instructions"
342 print ".."
343 print ".wh 0 HD"
344 print ".wh -0.7i FT"

345 before group of chartid2
346 print ".ne 3.0i"
347 print ".mk R"             {print classid and title}
348 print ".ps 10"
349 print ".vs 12"
350 print ".sp"
351 print ".in 4.2i"
352 print ".ce"
353 print "Class ", rateclas
354 print ".sp"
355 print chart_title without trailing spaces
356 print ".rt"
357 print ".ps 8"
358 print ".vs 10"
```

```
359 print ".in 0"
360 print ".in 0.4i"
361 print ".G1"                {read in three macro definitions}
362 print "copy PLOT.macro"
363 print "copy QMPBOX.macro"
364 print "copy LABEL4.macro"
365 print "PLOT"

366 on every record            {create QMPBOX data lines}
367 if ( adj!="H" ) then begin
368 if ( adj!="U" ) then begin
369 if ( adj!="R" ) then begin
370 print "QMPBOX(", chartid2, ",",
371       period, ",", bestmeas   using "##.##",  ",",
372       long_runa  using "##.##",  ",", curr_index using "##.##",  ",",
373       pctl99     using "##.##",  ",", pctl95     using "##.##",  ",",
374       pctl5      using "##.##",  ",", pctl1      using "##.##",  ")"
375 end end end
376                             {create LABEL4 data lines}
377 if ( pctl99 > 1.00 ) then print "LABEL4(", chartid2, ",", period, ",BN)"
378 if ( pctl95 > 1.00 and pctl99 < 1.00 )
379            then print "LABEL4(", chartid2, ",", period, ",A)"
380 if ( adj="U" ) then print "LABEL4(", chartid2, ",", period, ",NP)"
381 if ( adj="H" ) then print "LABEL4(", chartid2, ",", period, ",WS)"
382 if ( adj="R" ) then print "LABEL4(", chartid2, ",", period, ",NS)"

383 after group of chartid2
384 if ( sample > 0 ) then begin   {compute absolute scale values}
385     if ( rateunit="1" ) then let pdef = (expected/sample)
386     if ( rateunit="2" ) then let pdef = (expected/sample) * 100
387     if ( rateunit="3" ) then let pdef = (expected/sample) * 1000000
388     let pdef2 = pdef * 2.0
389     let pdef3 = pdef * 3.0
390 if ( rateunit="1" ) then begin  {create Demerits scale}
391     print "ticks right out at",
392     " -1 ", quote, pdef  using "###.###", quote, ",",
393     " -2 ", quote, pdef2 using "###.###", quote, ",",
394     " -3 ", quote, pdef3 using "###.###", quote
395     print quote, "Demerits", quote, " ljust at 17.2, -3.74"
396     print quote, "per unit", quote, " ljust at 17.2, -4.00" end
397 if ( rateunit="2" ) then begin    {create Percent Defective scale}
398     print "ticks right out at",
399     " -1 ", quote, pdef  using "##.####", " %", quote, ",",
400     " -2 ", quote, pdef2 using "##.####", " %", quote, ",",
401     " -3 ", quote, pdef3 using "##.####", " %", quote
402     print quote, "Percent", quote, " ljust at 17.2, -3.74"
403     print quote, "Defective ", quote, " ljust at 17.2, -4.00" end
404 if ( rateunit="3" ) then begin    {create Parts Per Million scale}
405     print "ticks right out at",
406     " -1 ", quote, pdef  using "######", " PPM", quote, ",",
407     " -2 ", quote, pdef2 using "######", " PPM", quote, ",",
408     " -3 ", quote, pdef3 using "######", " PPM", quote
409     print quote, "Parts per", quote, " ljust at 17.2, -3.74"
410     print quote, "Million ", quote, " ljust at 17.2, -4.00" end
411 end
412 print ".G2"
```

```
413 print ".in 4.2i"
414 print ".ps 10"
415 print ".vs 12"
416 print ".rt"                        {print table data}
417 print ".sp 1.0i"
418 print ".TS"
419 print "tab(:);"
420 print "lw(21) rw(8)."
421 print "PERIOD:", period
422 print "Output:", universe   using "##########&"
423 print "Sample:", sample     using "#########&"
424 if variance > 0.01 then begin
425     let e1 = expected
426     let e2 = variance
427     let ee = (e1 * e1) / e2
428     print "Equivalent Expectancy:", ee using "##########.##"
429 end else print "Equivalent Expectancy:-"
430 print "Assessed:", assessed   using "#########&"
431 print "Expected:", expected   using "#######.##"
432 print "Current Index:", curr_index using "#######.##"
433 print "Best Measure:", bestmeas   using "#######.##"
434 print ".TE"
435 print ".in 0"
436 print ".rt"
437 print ".sp 2.8i"
438 end
```