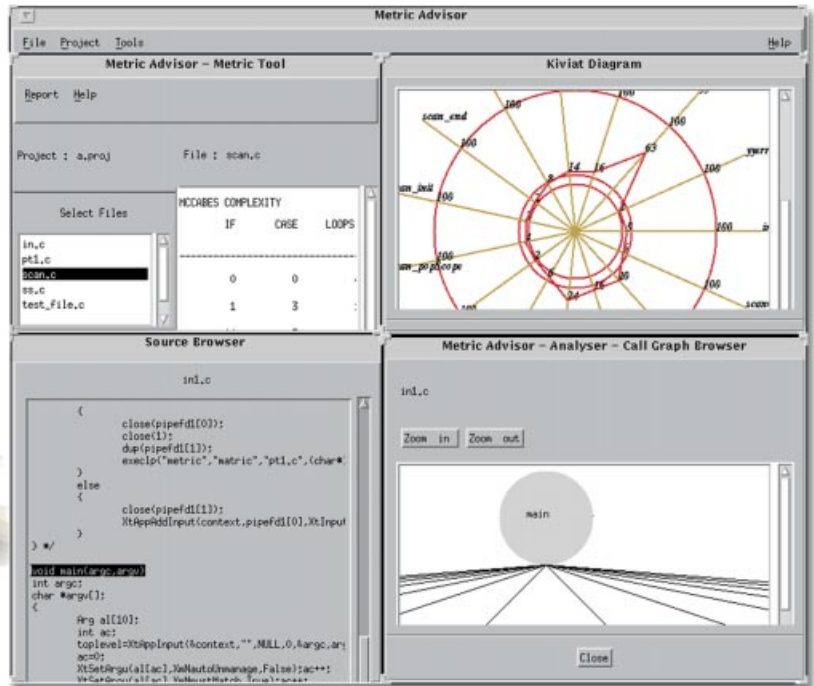


## Enabling Developers to build-in Quality



MetricAdvisor

### INTRODUCTION

One of the major areas of concern in Software Engineering is the maintenance process, as it is estimated to consume 40-75% of the software effort. Finding the code that is error prone and expensive to maintain is the first step in reducing the maintenance effort. After identifying such code, it can be subjected to more rigorous tests or re-written if necessary. The use of metrics helps the developer in identifying source code that is error prone and difficult to test. Metrics also provide an insight into the software, as well as the processes used to develop and maintain it. The MetricAdvisor\* is a tool that helps in developing a highly maintainable product by evaluating key software metrics and graphically presenting the structure of the program.

### DESCRIPTION

The MetricAdvisor calculates the following metrics: McCabe metrics which helps in estimating code complexity for identifying the most risky code, Halstead metrics which helps in calculating the programming effort in man-months, Complexity

### HIGHLIGHTS

- Halstead metrics for man-months calculation.
- Cyclomatic Complexity for identifying the most complex modules.
- Complexity Density Metrics for maintenance productivity.
- Fan-in and Fan-out metrics for estimating maintenance complexity.
- Commentedness and LOC metrics for maintenance and productivity estimation.
- Call graph for source code analysis.
- Graphical display using Kiviatic and X-Y diagrams.
- User configurable threshold values for the metrics.
- Context sensitive help.

Density Metrics that helps in predicting the maintenance productivity and Fan-in, Fan-out metrics which help in locating modules that contribute the highest maintenance effort. Besides these, it also helps the user to understand the structure of the code.

For a detailed analysis of software quality it presents the following graphic results:

- The architecture of the application through call graphs which display the relationship between functions.
- Compliance with the quality model through Kiviat and X-Y graphs which help to visualize the level of compliance with the thresholds.

The Metric Advisor has two components: the Metric Tool and the Analyser. The Metric Tool which is the major component, calculates a variety of source code metrics by parsing the source code. It also compares the software against the configured quality threshold. It prints a report of the quality level of the software to help the developer focus on the portions of code that needs to be reworked. The user can take corrective actions to make the code meet the expected standards. The Analyser helps the developer in locating the various identifiers and understanding the call graph of the code. The call graph shows the relationship between the different functions in the program. This helps in easy identification of functions that contribute to maximum complexity and maintenance effort.

## METRIC TOOL

The Metric Tool calculates the following metrics.

- McCabe's Cyclomatic complexity
- Fan-in and Fan-out
- KLOC
- Commentedness
- Complexity Density
- Halstead Software Science Metrics.

### McCabe's Cyclomatic Complexity

Cyclomatic complexity is based on program control structure and can be calculated from the number of conditional statements in the source code. If a function has a large number of decision statements, it has a tendency to be more difficult to comprehend and more difficult to test. As a result, functions with a high cyclomatic complexity are more error prone. Thus,

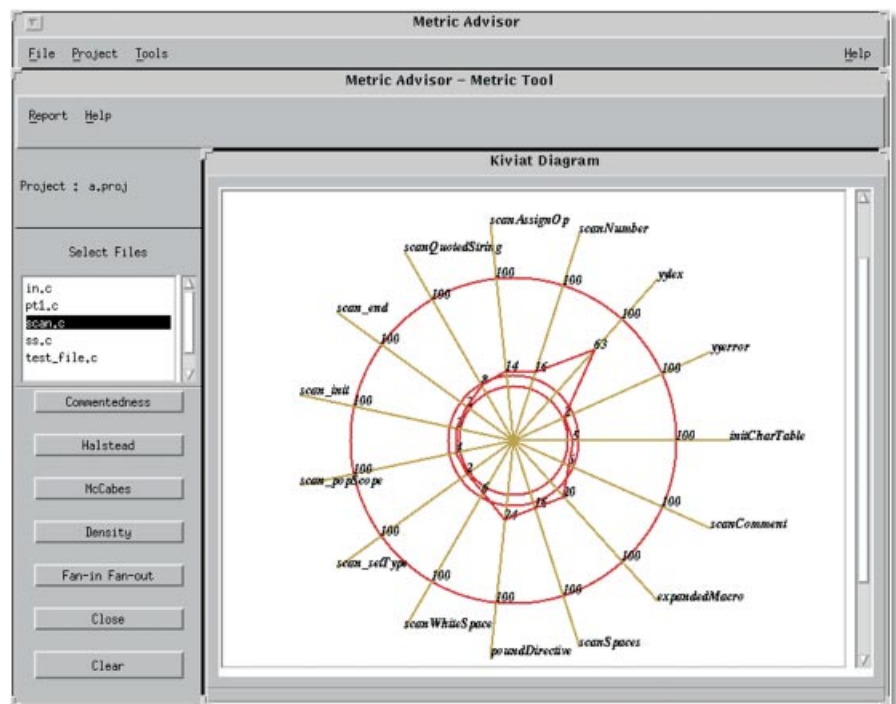
this metric allows one to focus on the most complex code, which will result in higher rate of defect removal.

Complexity metrics are plotted on Kiviat and X-Y diagrams making it easy to spot the most complex and error prone functions.

### Fan-in and Fan-out

The Fan-in and Fan-out metrics are used in estimating the complexity of maintaining the software.

Fan-out indicates the number of functions a function calls. Modifying a function can cause side effects in the functions that are called by the modified function. A maintainer of this module has to understand many other functions making maintenance harder and time consuming. Therefore, functions with a large Fan-out will be more expensive to maintain.



Metric Advisor - Kiviat Graph

Fan-in is the count of the number of functions that call a function. A function with a high Fan-in means that many functions use this function. It could also mean that the function is implementing a number of functionalities. If the specifications of a function with large Fan-in is changed then all the other functions that use this have to be modified.

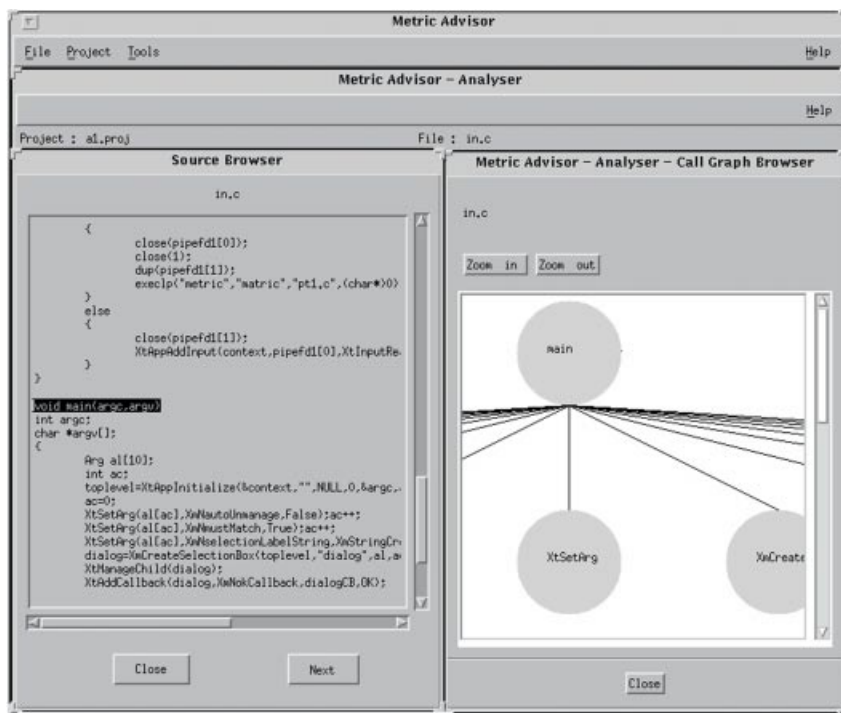
### Commentedness and KLOC

Commentedness is an indication of the readability of a program. This is very important for maintenance and re-engineering.

Program development and maintenance effort is primarily a function of program size in kilo lines of code (KLOC). The length hypothesis states that the number of bugs is proportional to the number of machine language statements. KLOC can be used for cost estimation of future projects, comparison of products and for determining productivity. Most of the standard cost estimation models depend on program size.

### Complexity Density

Complexity Density is inversely proportional to the maintenance productivity. Maintenance productivity can be defined as the total number of lines added divided by the time in hours spent in coding and testing upgrades to the software. Thus, Complexity Density also can be used as a measure of the difficulty of maintaining a program.



Metric Advisor - Analyser

### Halstead Software Science Metrics

Halstead metrics is based on the fact that the complexity of a program is related to the number of operators and operands in the program. Operators are syntactic elements such as +, -, <, > and operands are quantities that receive the operators namely variables and constants. Halstead metrics calculates the Program Volume, Difficulty and Programming Effort. The Effort measured in the Halstead metrics is a measure of the effort invested in man-months programming the module.

### Other Features


Besides calculating the above metrics, the tool allows the user to change the default threshold values and also

indicates the extent to which the measured value exceeds the threshold value. The on-line help includes documentation that helps the user to interpret the metric values.

### ANALYSER

The Analyser parses the source code and lists out the functions, structures, macros, global variables and externals in the programs. It also includes a source browser that allows the user to click on an identifier and view the corresponding source code. The local variables in a function can also be listed.

The call graph supports zoom in and zoom out facilities. The user can view the corresponding source code and the complexity metrics of a function by

clicking on any node in the call-graph.  **ON-LINE HELP**

It also draws the call-graph with any arbitrary function as the root node.

The call graph helps the developer to understand the code better which helps in re-engineering and maintenance.

The MetricAdvisor environment has context sensitive on-line help. Users can select help for overall product features or for a specific topic. Within the on-

line help, users can navigate by following hyperlinks to related topics. The help gives information to the users that allow them to interpret the metrics calculated.

## AVAILABILITY

Supported Hardware	:	UNIX Workstations
Supported Operating System	:	Solaris 2.x*
User Interfaces	:	GUI
Supported Languages	:	C
Prerequisite Software	:	X/Motif*



A Scientific Society of the  
Department of Electronics  
Government of India

### Additional Information

For more information on CDAC HPCC software, contact your CDAC marketing representative, access the CDAC Home Page on the internet World-Wide Web ( [www.cdac.org.in](http://www.cdac.org.in) ), or send an e-mail over the internet to : [ssg@cdacb.ernet.in](mailto:ssg@cdacb.ernet.in)

C-DAC reserves the right to change or modify any of the product or service specifications or features described herein without notice. The product summary is for information only. C-DAC makes no express or implied representations or warranties in this summary

\*All trademarks and brand names are owned by their respective owners.

#### Headquarters

University of Poona Campus,  
Ganesh Khind, Pune - 411 007, INDIA  
Tel : 352461 Fax : 91-212-357551  
Tlx : 0145-7615 CDAC IN  
email : [business@cdac.ernet.in](mailto:business@cdac.ernet.in)

#### Business Division

Ramanashree Plaza, 2/1 Brunton Road,  
Bangalore - 560 025, INDIA  
Tel : 5584271 Fax : 91-80-5584893  
Tlx : 0845-8413 CDAC IN,  
email : [bdm@cdacb.ernet.in](mailto:bdm@cdacb.ernet.in)

#### Delhi Centre

E-13, 2nd Floor, Hauz Khas,  
New Delhi - 110 016, INDIA  
Tel : 6863428 Fax : 91-11-6863428