EVALUATING SOFTWARE DEVELOPMENT CHARACTERISTICS: ASSESSMENT OF SOFTWARE MEASURES IN THE SOFTWARE ENGINEERING LABORATORY

Victor R. Basili
University of Maryland
College Park, MD 20742

The purpose of this presentation is to discuss some of the work done on metrics in the Software Engineering Laboratory. To put things in perspective, there are many factors that affect software quality and each of these factors has several criteria which define it. Metrics represent some sort of measurement as to whether or not we have achieved a particular criteria. For example, one factor that we would like the software to possess is reliability. One of the many criteria that goes to make up this generalized factor of reliability might be fault tolerance. One of the metrics that can be used to evaluate fault tolerance might be the number of crashes of the system.

There are many views of metrics. We can think of metrics as being subjective or objective. Subjective metrics normally do not involve any exact measurement; they tend to be an estimate of extent to a degree in the application of some technique or a classification or qualification of a problem or experience. Subjective metrics are usually done on a relative scale; e.g., they may be binary (yes or no), or discrete numbers (zero, 1, 2, 3). Examples of subjective metrics would be a qualitative judgment on the use of Process Design Language or an evaluation of the experience of programmers in a particular application.

Objective metrics, on the other hand, tend to be absolute measures taken on the product or process. For example, the time of development, the number of lines of code delivered, the productivity in lines of code per staff month, the number of errors or changes associated with the project. The distinction between subjective and objective metrics is typically a little bit fuzzy. Very often we make a metric subjective because we don't know how to quantify it.

Another characterization of metrics is as product or process metrics. Product metrics measure the developed product, such as the source code, the object code, or the documentation. Such metrics might be lines of code (objective metric) or readability of the source code (subjective metric). Process metrics tend to measure the process model used for developing the product. Metrics such as use of methodology (subjective metric) and effort and staff months (objective metric) are two metrics that measure the process.

Another characterization is to think of metrics as being cost or quality metrics. It is clear that cost can be a quality metric. However, typically a goal in software development is to minimize cost and maximize quality. So for that reason we will consider these as separate views. Cost normally involves the expenditure of resources in dollars, which might include some capital investment, and this metric is usually normalized according to some value component. For example, we measure staff months or productivity in terms of dollars received for dollars spent, or output for dollars spent, or size per time slice. Quality metrics, on the other hand, measure some form of the value of the product. For example, trying to measure the mean time to failure of the product, the ease of change, the correctness, or the number of errors remaining are all quality measures.

Use of Metrics

We use metrics in varying ways. We can use them to characterize, evaluate, or predict. Almost all metrics fit in the characterizing category. In that sense, the metric helps to distinguish the product and process or environment. For example, we may categorize an environment by the use of a methodology, the number of externally-generated changes, or the size. This allows us to compare environments or products or processes.

Not all characterizing metrics are evaluative. Metrics are considered evaluative if the metric correlates with or shows directly the quality of the process or the product. For example, the number of errors recorded during acceptance testing or the productivity involved in the development of a software project give us some way of evaluating whether the product has some reasonable reliability or the development is cost effective.

The most powerful capability a metric can have is prediction; that is, the measure is estimable or calculable and is used to predict another measure. For example, estimating size as a predictor of effort is a way to use an estimable metric to predict some desired information.

To demonstrate that a particular metric evaluates or predicts, requires some validation. Too often metrics are proposed in the literature which are meant to be evaluative or predicted, but that capability is not established by experiment or case study.

Analyzing Objective Metrics in the Software Engineering Laboratory

In a paper presented at the Sigmetrics Workshop (Basili/Phillips), we tried to use the laboratory project data to study the relationship between various metrics of size and complexity. One of the questions raised was could we predict effort, which was a cost measure, and the number of errors, a quality metric, using the various size and complexity metrics that appear in the literature. A second question was to be able to check the internal consistency of several of those size and complexity metrics. The metrics used are given in Table 1. The relationship between the various complexity metrics appears in Table 2, which gives the Pearson correlation coefficient. As can be seen from this table, several of the complexity and size metrics

OBJECTIVE SIZE AND COMPLEXITY MEASURES STUDIED

SRC : SOURCE LINES OF CODE INCLUDING COMMENTS

XQT : EXECUTABLE STATEMENTS

SOFTWARE SCIENCE METRICS

- N : LENGTH IN OPERATORS AND OPERANDS
- V : VOLUME
- V* : POTENTIAL VOLUME
- L : LEVEL
- E : EFFORT

CYC : CYCLOMATIC COMPLEXITY

CLS : NUMBER OF CALL STATEMENTS

CAJ : CALLS AND JUMPS

CHG : CHANGES TO THE SOURCE CODE

REV : NUMBER OF REVISIONS (VERSIONS) IN THE LIBRARY

EFF : NUMBER OF HOURS EXPENDED IN DEVELOPMENT

ERR : NUMBER OF ERRORS ASSOCIATED WITH COMPONENT

Table 1

RELATIONSHIP BETWEEN SIZE AND COMPLEXITY METRICS

	REV	CHG	XQT	SRC	CAJ	CYC	CLS
E	.6750	.2407	.8390	.8706	.8742	.8906	.7966
CLS	.6427	.3579	.7594	.8186	.9648	.8651	
CYC	.7921	.2534	.9253	.9519	.9666		
CAJ	.7439	.3158	.8734	.9176			
SRC	.8415	.2942	.9896				
XQT	.8560	.2920					
CHG	.4229						

Table 2

correlate well with one another. On the other hand, the change metrics do not correlate well. In trying to use combinations of these metrics to predict effort and errors, we see by Table 3 that there is some success in accounting for effort with some of the metrics, but less success in accounting for errors.

PREDICTING EFFORT AND ERRORS USING

SIZE AND COMPLEXITY METRICS

	EFF	ERR
EFF		.6346
CLS	.7977	.5704
CYC	.7399	.5592
CAJ	.7957	.5848
SRC	.7583	.5576
XQT	.7400	.5485
REV	.7122	.6734
E	.6612	.5432
CHG	.4799	

Table 3

Another study was to look at the internal validation of some of the metrics. Specifically, the software science metrics were examined to see whether predicted values for some of the metrics and actual values related in some way. Again, Pearson's correlation was used; the results are given in Table 4. One can see from this table that metrics like length, that is, N and $\hat{N}$, do correlate. There is not a bad relationship between V and V*, although in the group of metrics, that relationship is probably the worst. It should be noted that projects are broken up into two groups--those of small components which were 50 lines or less, and large components which were more than 50 lines.

Based on this study, we made the following conclusions: First of all, there does exist some relationship between complexity metrics and effort and errors. However, most of the complexity metrics do not do much better at estimation than lines of code or executable statements. On the other hand, many of the metrics related very well with each other, which seems to imply that they really are measuring the same thing. The goal, therefore, should be concentrated on looking at orthogonal metrics. We are currently investigating data metrics in the SEL.

Using Subjective and Objective Metrics to Predict Cost

In a paper presented at the 5th International Conference on Software Engineering (Bailey/Basili), we inverted that experiment by examining the relationship between productivity and various factors. Basically, we used nonparametric statistics. The results were as follows: We found no significant relationship between productivity and size. However, there was a large set of methodology factors that showed varying degrees of positive correlation with productivity. A combined methodology factor that was used to pre-

INTERNAL VALIDATION

SMALL COMPONENTS	50 LINES	(280)
LARGE COMPONENTS	50 LINES	(285)

	LARGE	SMALL
$N \sim \hat{N}$	.79	.83
$V \sim V^*$	.52	.50
$L \sim \hat{L}$	.71	.62
$E \sim \hat{E}$	.61	.42

PEARSON CORRELATION

Table 4

dict cost or effort in the cost model showed a significant positive correlation with productivity as might have been expected. In this study, projects with high methodology rating were shown to have come from a different population than those with a low methodology rating. No other factor showed a significant positive correlation with productivity and we were able to show, at least in the SEL environment, that methodology does correlate with productivity and therefore has been an effective approach to software development.

Using Subjective Metrics to Predict Quality

Based on the study to predict productivity but changing the statistical approach to factor analysis, we compressed three sets of metrics into three factors--quality, methodology, and complexity. Methodology ana complexity were not significantly correlated in the study. However, quality was significantly correlated with methodology with a correlation (R) of .67 and quality was also significantly correlated with complexity with a correlation (R) of -.64. In both cases, the correlation was less than a .001 significance level.

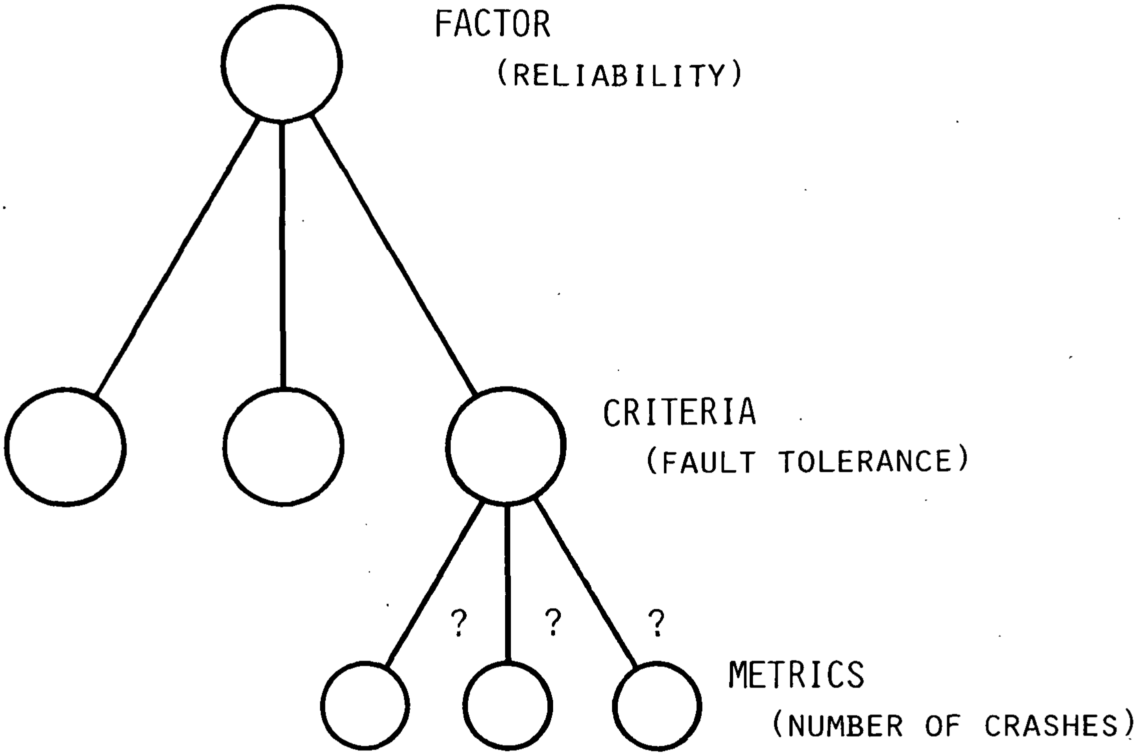
Using methodology alone to predict quality, the coefficient of determination (R^2) is equal to .45. This means that methodology accounted for essentially 45% of the quality rating. Using methodology and complexity both, we got an R^2 of .65. This implies that there is some evidence that we can predict quality from methodology and complexity and that methodology is again highly correlated, not with just productivity as we saw in the previous study, but also with quality. Work in this particular area is just beginning and we plan to make tremendous use of the subjective metrics, not just for evaluation, but also for prediction.

REFERENCES

(Basili/Phillips) - Basili, V. and Phillips, T.,"Validating Metrics on Project Data" - Submitted to special issue of Software Metrics, Transactions on Software Engineering.

(Bailey/Basili) - Bailey, J. and Basili, V., "A Meta-Model for Software Development Resource Expenditures," Proceedings of the Fifth International Conference on Software Engineering, March 1981, pp. 107-116

THE VIEWGRAPH MATERIALS
for the
V. BASILI PRESENTATION FOLLOW



FACTOR
(RELIABILITY)

CRITERIA
(FAULT TOLERANCE)

? ? ?

METRICS
(NUMBER OF CRASHES)

VIEWS OF METRICS

SUBJECTIVE VS. OBJECTIVE

SUBJECTIVE:

- NO EXACT MEASUREMENT
- AN ESTIMATE OF EXTENT OR DEGREE IN THE APPLICATION OF SOME TECHNIQUE
- A CLASSIFICATION OR QUALIFICATION OF PROBLEM OR EXPERIENCE
- USUALLY DONE ON A RELATIVE SCALE
- E.G., USE OF A PDL
- EXPERIENCE OF THE PROGRAMMERS IN THE APPLICATION

OBJECTIVE:

- AN ABSOLUTE MEASURE TAKEN ON THE PRODUCT OR PROCESS
- E.G., TIME FOR DEVELOPMENT
- NUMBER OF LINES OF CODE
- PRODUCTIVITY
- NUMBER OF ERRORS OR CHANGES

VIEWS OF METRICS

PRODUCT VS. PROCESS

PRODUCT:

MEASURE OF THE ACTUAL DEVELOPED PRODUCT

I.E., SOURCE CODE, OBJECT CODE, DOCUMENTATION

E.G., LINES OF CODE, READABILITY OF THE SOURCE CODE

PROCESS:

MEASURE OF THE PROCESS MODEL USED FOR DEVELOPING THE PRODUCT

E.G., USE OF METHODOLOGY, EFFORT IN STAFF MONTHS

- - - - - - - - - - - -

COST VS. QUALITY

COST:

EXPENDITURE OF RESOURCES IN DOLLARS INCLUDING CAPITAL INVESTMENT USUALLY NORMALIZED ACCORDING TO SOME VALUE COMPONENT

E.G., STAFF MONTHS, PRODUCTIVITY, SIZE/TIME SLICE

QUALITY:

SOME FORM OF VALUE OF THE PRODUCT

E.G., RELIABILITY, EASE OF CHANGE, CORRECTNESS, NUMBER OF ERRORS REMAINING

USE OF METRICS

PREDICTIVE VS. EVALUATIVE VS. CHARACTERIZING

CHARACTERIZING:

MEASURE HELPS DISTINGUISH THE PRODUCT OR PROCESS OR ENVIRONMENT

E.G., USE OF A METHODOLOGY, NUMBER OF EXTERNALLY GENERATED CHANGES, SIZE

EVALUATIVE:

MEASURE CORRELATES WITH OR SHOWS DIRECTLY THE QUALITY OF THE PROCESS OR PRODUCT

E.G., NUMBER OF ERRORS REPORTED DURING ACCEPTANCE TESTING, PRODUCTIVITY

PREDICTIVE:

MEASURE IS ESTIMATABLE OR CALCULABLE AND IS USED TO PREDICT ANOTHER MEASURE

E.G., ESTIMATING SIZE AS A PREDICTOR OF EFFORT

USE REQUIRES VALIDATION

ANALYZING OBJECTIVE MEASURES IN THE SEL

USING SEL PROJECT DATA TO STUDY THE RELATIONSHIP BETWEEN VARIOUS METRICS OF SIZE AND COMPLEXITY

PREDICTING EFFORT (A COST MEASURE) AND NUMBER OF ERRORS (A QUALITY METRIC) USING SIZE AND COMPLEXITY METRICS

CHECKING THE INTERNAL CONSISTENCY OF SEVERAL SIZE AND COMPLEXITY METRICS

OBJECTIVE SIZE AND COMPLEXITY MEASURES STUDIED

SRC : SOURCE LINES OF CODE INCLUDING COMMENTS

XQT : EXECUTABLE STATEMENTS

SOFTWARE SCIENCE METRICS

- N : LENGTH IN OPERATORS AND OPERANDS
- V : VOLUME
- V* : POTENTIAL VOLUME
- L : LEVEL
- E : EFFORT

CYC : CYCLOMATIC COMPLEXITY

CLS : NUMBER OF CALL STATEMENTS

CAJ : CALLS AND JUMPS

CHG : CHANGES TO THE SOURCE CODE

REV : NUMBER OF REVISIONS (VERSIONS) IN THE LIBRARY

EFF : NUMBER OF HOURS EXPENDED IN DEVELOPMENT

ERR : NUMBER OF ERRORS ASSOCIATED WITH COMPONENT

PREDICTING EFFORT AND ERRORS USING SIZE AND COMPLEXITY METRICS

	EFF	ERR
EFF		.6346
CLS	.7977	.5704
CYC	.7399	.5592
CAJ	.7957	.5848
SRC	.7583	.5576
XQT	.7400	.5485
REV	.7122	.6734
E	.6612	.5432
CHG	.4799	

RELATIONSHIP BETWEEN SIZE AND COMPLEXITY METRICS

	REV	CHG	XQT	SRC	CAJ	CYC	CLS
E	.6750	.2407	.8390	.8706	.8742	.8906	.7966
CLS	.6427	.3579	.7594	.8186	.9648	.8651	
CYC	.7921	.2534	.9253	.9519	.9666		
CAJ	.7439	.3158	.8734	.9176			
SRC	.8415	.2942	.9896				
XQT	.8560	.2920					
CHG	.4229						

INTERNAL VALIDATION

SMALL COMPONENTS	50 LINES	(280)
LARGE COMPONENTS	50 LINES	(285)

	LARGE	SMALL
$N \sim \hat{N}$	.79	.83
$V \sim V^*$	.52	.50
$L \sim \hat{L}$	.71	.62
$E \sim \hat{E}$	.61	.42

PEARSON CORRELATION

CONCLUSION

- CAN USE COMMERCIALLY-OBTAINED DATA TO VALIDATE COMPLEXITY METRICS
- VALIDITY CHECKS AND ACCURACY RATINGS ARE VITAL
- THERE EXIST RELATIONSHIPS BETWEEN COMPLEXITY METRICS AND EFFORT AND ERROR COUNTS
- THE BETTER THE DATA, THE BETTER THE RESULTS
- DON'T DO MUCH BETTER THAN LINES OF CODE ON EXECUTABLE STATEMENTS
- METRICS RELATE WELL WITH EACH OTHER (MEASURING THE SAME THING)

USING SUBJECTIVE AND OBJECTIVE METRICS TO PREDICT COST (EFFORT)

A META-MODEL WAS DEVELOPED FOR DERIVING AN INDIVIDUALIZED COST MODEL FOR THE LOCAL ENVIRONMENT

IT ASSUMES EACH ENVIRONMENT IS DIFFERENT AND IS CLASSIFIABLE BY A SET OF FACTORS (CAPTURED USING SUBJECTIVE METRICS)

SOME FACTORS ARE CONSTANT ACROSS THE ENVIRONMENT AND ARE HIDDEN IN A BASIC SIZE/EFFORT EQUATION BASED UPON PAST HISTORY WITHIN THE ENVIRONMENT

OTHER FACTORS CAUSE DIFFERENCES BETWEEN PROJECTS AND CAN BE USED TO EXPLAIN THE DIFFERENCE BETWEEN ACTUAL EFFORT AND EFFORT AS PREDICTED BY THE BASIC SIZE/EFFORT EQUATION

CAN PREDICT COST (EFFORT) WITH THE USE OF SUBJECTIVE METRICS

EVALUATING THE EFFECT OF VARIOUS FACTORS ON PRODUCTIVITY

WE EXAMINED THE RELATIONSHIP BETWEEN PRODUCTIVITY AND VARIOUS FACTORS

FOUND NO SIGNIFICANT RELATIONSHIP BETWEEN PRODUCTIVITY AND SIZE

A LARGE SET OF METHODOLOGY FACTORS SHOWED VARYING DEGREES OF POSITIVE CORRELATION WITH PRODUCTIVITY

A COMBINED METHODOLOGY FACTOR SHOWED A SIGNIFICANT POSITIVE CORRELATION WITH PRODUCTIVITY

[PROJECTS WITH HIGH METHODOLOGY RATING CAME FROM A DIFFERENT POPULATION THAN THOSE WITH A LOW METHODOLOGY RATING]

NO OTHER FACTORS SHOWED A SIGNIFICANT POSITIVE CORRELATION WITH PRODUCTIVITY

METHODOLOGY IS CORRELATED WITH PRODUCTIVITY

USING SUBJECTIVE METRICS TO PREDICT QUALITY

WE COMPRESSED THREE SETS OF METRICS INTO THREE FACTORS: QUALITY, METHODOLOGY, AND COMPLEXITY

METHODOLOGY AND COMPLEXITY WERE NOT SIGNIFICANTLY CORRELATED

QUALITY WAS SIGNIFICANTLY CORRELATED WITH METHODOLOGY ($R = .67$) AND COMPLEXITY ($R = -.64$) AT LESS THAN .001 SIGNIFICANCE LEVEL

USING METHODOLOGY ALONE TO PREDICT QUALITY, $R^2 = .45$

USING METHODOLOGY AND COMPLEXITY WE GET $R^2 = .65$

THERE IS EVIDENCE WE CAN PREDICT QUALITY FROM METHODOLOGY AND COMPLEXITY

METHODOLOGY IS CORRELATED WITH QUALITY