OOPSLA 1993

Addendum to the Proceedings

Washington, DC—26 September–1 October, 1993

Workshop

# Processes and Metrics for Object-Oriented Software Development

**Organized By:**

Steven C. Bilow, Tektronix, Inc.
Doug Lea, SUNY Oswego & NY CASE Center
Karl Freburger, Knowledge Systems Corporation
Dennis de Champeaux, Rational

**Report By:**

Steven C. Bilow, Tektronix, Inc.
Doug Lea, SUNY Oswego & NY CASE Center

## Introduction & Workshop Overview

On Sunday, 26 September, 1993 the OOPSLA '93 Workshop on Processes and Metrics for Object Oriented Software Development was held in Washington DC. The workshop was structured in 3 sections consisting of specific, focused activities. Those sections were; (1) A series of formal paper presentations. (2) Four smaller special topic discussion groups facilitated by the workshop organizers. (3) A consensus building process, group discussion, and construction of a position statement. Summaries of these activities will constitute the body of this report.

## Formal Presentations

The first session of the day began with a brief introduction by Steve Bilow, an overview of the days activities by Doug Lea, and 1 minute participant introductions. This was followed by the presentation of 8 position papers. Those papers were selected by the workshop organizers, from the body of 24 papers submitted, as the most interesting, thought provoking, or universally relevant. The following were the papers selected.

### Shyam Chidamber and Chris Kemerer, MIT Sloan School

#### *"MOOSE: Metrics for Object Oriented Software Engineering"*

Shyam Chidamber represented himself and Chris Kemerer in a presentation of the current status of the MOOSE metrics. The Chidamber and Kemerer metrics have generated a significant amount of interest and are currently the most well known suite of measurements for O-O software. In Shyam's quest to validate his metrics he has spent 3 months interviewing software designers and several month collecting empirical data from both C++ and Smalltalk projects. His principle points are that metrics must be theoretically rigorous and practically relevant. Toward those goals, the MOOSE metrics are beginning to show strong empirical validity.

In response to Kenny Rubin's question about how the numbers resulting from the metrics should/can be used, Shyam proposed calibrating the data against a large body of "typical" cases and then using them to predict maintainability, effort, reuse, etc. One member of the group asked whether Shyam had a feel for the relative "quality" of Smalltalk versus C++ and he provided the only rational answer that you can't compare apples and oranges.

One of the most important things we can learn from the work of Chris and Shyam is that it is extremely difficult to collect a body of empirical data large enough to rationally validate a set of measures.

### Sallie Henry and Mark Lettanzi, Virginia Tech.

#### *"Measurement of Software Maintenance and Reliability in the Object Oriented Paradigm"*

Among the most interesting work in the field of O-O metrics is that of Dr. Sallie Henry and her students. For this workshop, doctoral student Mark Lettanzi presented the status of his work on the prediction of maintenance and reliability. The paper describes the results of three studies. One, a study of maintenance difficulty in procedural versus O-O software. Two, a relationship between reuse, productivity, and O-O techniques. Three, a modification and application of the MOOSE (C&K) metrics for the purpose of predicting maintainability. Mark openly shared his work and proposed that there is far too little evidence to support the O-O industries claim of increased maintainability. His research effort, however, does support that claim even if not through empirically validated metrics.

**Mark Lorenz and Jeff Kidd,**
**Hatteras Software**

***"O-O Metrics Position Paper"***

Mark Lorenz and Jeff Kidd came to the workshop with metrics taken from their up and coming book on the subject. They made two very important points. First, they make a large distinction between what they term "Project Metrics" and "Design Metrics." Project metrics include such items as schedule, staffing estimating, and nearness to completion. Design metrics, on the other hand include such items as method "size," class size, use of inheritance, cohesion, etc. He showed a number of charts that illustrated distributions of some of his measures across several projects. He did not speak to the relation of his proposed metrics to formal mathematics, but, his body of data is most interesting because it come from empirical data rather than theory.

More important than the presentation of Mark's metrics was his statement that metrics should not "drive design" but, rather, should be used to pinpoint anomalies. Some may disagree with this premise but it does remind us of how easy it is to misuse measures.

**Kathy Reinold,**
**Bull HN Information Systems**

***"Processes and Metrics for Object-Oriented Software Development"***

Kathy Reinold brought to us the perspective of a software manager tasked with moving from a procedural world to an object oriented one. Her paper dealt primarily with our old familiar friend—the "line of code." While many of us shy away from KLOC as a measure of software, Kathy brought use the perspective of one forced to "use what is available." She spoke of KLOC and function points throughout her talk as well as what she called "object counting" which the group decided was actually "class counting." She presented charts illustrating her contention that Function Point Analysis and "object counting" have the ability to product reliable KLOC estimates, even in O-O projects.

**Linda Rising,**
**Honeywell, Inc.**

***"An Information Hiding Metric"***

Another quite interesting bit of work came from Linda Rising at Honeywell. She has been working on measures of information hiding in Object-Based languages and brought the ADA perspective to the group. Unlike the work of Lorenz and Kidd or Reinold, Linda's metric has its foundation in measurement theory, not practical application. She proposed two basic premises. One, that a module having more that one design decision is "poorer" at information hiding than one containing only one design decision ( She did not, unfortunately, have time to define "design decision"). Two, the more entities that exist but are not required to implement the design decision the "poorer" the information hiding. Linda's work has been validated theoretically, subjectively, experimentally, and in a case study of a large Ada program.

**Irene Brooks,**
**Texas Instruments**

***"Object Oriented Metrics Collection and Evaluation with a Software Process"***

Irene Brooks presented a paper based on practical experience at Texas Instruments. She explained how TI has found measures of size, defect density, and defect intensity to be very useful in the schedule management and defect control. She commented that some traditional metrics like McCabe complexity measure do not appear to be useful for O-O development. She also noted that it seems necessary to find adequate measures for polymorphism, inheritance, and the cohesion between class attributes. Her primary goals are repeatability of process, quality measurement, project management and configuration control techniques.

**Jerry Hamilton,**
**McDonnell Douglas**

***"Metrics: Where Things Went Wrong"***

Perhaps the most deeply revealing story of the day came from Jerry Hamilton who summed up the misuse of metrics through practical example. Jerry's is the story of "last minute metrics" and the result that unsubstantiated metrics can have on a project.

The project began with no metrics and only later did the management and customer decide that they need some measurements of progress. The project team decided to report, to the customer four categories of information. (1) the number of classes defined, coded, and integrated, (2) the number of services designed, coded, and integrated, (3) the number of classes remaining, and (4) the number of services remaining. As the project progressed the class count climbed. This may have been due to many causes such as refactoring and re-architecture but the customers perception was that the project was "out of control." This resulted in management declaring "NO MORE CLASSES." The rest of the story is painfully obvious.

**Xiping Song,**
**Siemens**

***"Engineering OO Design Methods into Repeatable Design Processes"***

The final formal presentation was by Xiping Song. His was the most "process oriented" paper to be accepted for presentation. Song's work is devoted to process "repeatability." His goal is to quantify existing design methodologies and generalize them into a set of processes, each of which is specific to a certain class of project, and which may be used consistently, and repeatably across these projects. The work is rooted in what Leon Osterweil calls "Process Programming." In other words, a software process should be designed, "coded," and executed in a manner quite similar to software itself. To apply this technique to OO methods, Song began with an architecture based on the work of Booch. He derived, partially from Booch's design examples, 7 variables according to which an OO method could be tailored for a given project. He tailored the Booch process into 2 different processes and developed a supporting tool that allows the user to select the process according to a set of variables.

## Discussion Groups

Following the formal presentation of papers the group broke into 3 smaller groups each with a specific focus. Steve Bilow led the group on "Metrics and Management," Doug Lea led the "Quality Metrics" group, and Dennis de Champeaux's group discussed "Processes and Micro-Processes." Due to space considerations these sessions will not be described in detail. Summaries of these sessions may be obtained from the respective workshop co-ordinator.

## Group Position Statement

The final session was devoted to construction of lists summarizing the state of affairs in OO process and metrics research and practice. The goal of this "list making" process was to obtain some measure of consensus from a very diverse group of 30 people. This was probably the most important part of the workshop and, though the resulting lists are a bit overly general, they do portray a group consensus relatively well. This process was facilitated by Doug Lea.

The first list is a set of metrics that are, without question or objection, valuable in OO software development. Unsurprisingly, given the constraints of uniform applicability, they mainly consist of metrics useful in just about any engineering effort:

- Problem size and complexity
- Fraction of budget expended
- Fraction of functionality completed
- Amount of effort expended
- Schedule progress
- Number of iterations

However, three categories of OO-specific metrics were also listed:

- Interdependence and coupling among object (e.g., counts of static and dynamic references)
- Class cohesion (e.g., measures of relatedness of methods within a class)
- Effective use of inheritance (e.g., measures of relatedness and reuse in subclasses)

In contrast, the following list contains metrics that are of such dubious value and interpretation in OOSE, that they should not be used, or used only with extreme caution:

- Lines of code written
- Lines of code reused
- Number of classes
- McCabe and Halsted complexity metrics

In the middle ground lie categories of metrics that are considered important, but are in need of further development to be useful. (This list was composed by the Quality Metrics breakout group in a separate session.)

- Modularity
- Reusability
- Interference
- Operationalizations of customer requirements
- Customer views of quality
- Human factors
- Productivity and quality in non-linear models, participatory design methods, and other novel processes

The group identified the following issues as the most pressing problems in current OO development practices.

- Lack of common lifecycle process definition
- Lack of corresponding management models and practices
- Inaccurate estimation
- Instability and unscalability of typical development processes
- Misunderstanding and misuse of OO concepts and constructs by developers
- Lack of sufficient training and PR
- Lack of reusability, especially at the domain level
- Failure to use proven procedures such as design reviews

The final list describes issues that are in most immediate need of further research to help solve practical development problems.

- The relationship between easily measured quantities and desired results
- The relationship between measures of the size and complexity of the problem space and those of the solution
- Accounts of the essential differences between OO and non-OO development
- Methods for accurately estimating development time and effort from initial requirements
- Development of better macro- and micro process models
- Better "theory of illness" of OO programs that underlies quality metrics
- Better accounts of essential differences between OO and non-OO metrics
- Better ways to use metrics to control complexity
- Development of metrics and instrumentation that programmers find informative, not threatening
- Collection and evaluation of empirical data of all sorts, especially for metrics validation, development of norms, and assessment of the impact of reuse on productivity and quality

## Conclusion

In general, we believe that our workshop was both productive and interesting. The combination of process and metrics was a mixed blessing. It increased the number of participants to the point where the effectiveness of participant interaction was strained. At the same time, it is very important for "process people" and metricians to actively communicate since these 2 areas will continue to contribute much to each other. It appears that this workshop did quite a good job of promoting that interaction.

## Participants

Frank Armour
*Frank_Armour@mail.amsinc.com*
John Baker N/A
Imran Bashir *ibashir@cat.syr.edu*
Steve Bilow *steven.bilow@tek.com*
Irene Brooks *elise@tasmania.works.ti.com*
Shyam Chidamber *shyam@athena.mit.edu*
Dennis de Champeaux *ddc@rational.com*
Martin Fowler
*100031.3311@compuserve.com*
Karl Freburger freb@goldfnch.mv.com
Jerry Hamilton N/A
Rachel Harrison rh@ecs.soton.ac.uk
John Hogg hogg@bnr.ca
Fred Jewell N/A
Cynthia Jones
*Cynthia_Palladino_Jones@mail.amsinc.com*
Roger Kelly *kelly@sde.mdso.vf.ge.com*
Jeff Kidd *71214.3120@compuserve.com*
Ralph Kolewe *rkolewe@eridani.com*
Mark Lattanzi *lattanzi@cs.vt.edu*
Doug Lea *dl@g.oswego.edu*
Mark Lorenz *71214.3120@compuserve.com*
Marco Mulazzani *M.Mulazzani@rcvie.co.at*
Jorgen Norberg *norberg@vnet.ibm.com*
Carl Ponder *oakhill!ponder@cs.utexas.edu*
Linda Rising
*rising%elhx.dnet@esu36.cfsat.honeywell.com*
Kenny Rubin *krubin@parcplace.com*
Radhika Seshan *rads@houxa.att.com*
Xiping Song *song@scr.siemens.com*
Todd Stevens *todds@cs.vt.edu*
Geree Streun *streun@fischer.com*
David Tegarden *tegarden@galllium.csusb.edu*
Laurel Von Gerichten *laurel@goofy.att.com*
Kurt Welker *wdk@inel.gov*
John Williams *jwilliam@stsci.edu*
Russel Winder *R.Winder@cs.ucl.ac.uk*