

1-1-1999

## A WWW Based Software Metrics Environment for Software Process Management and Software Product Quality Improvement

Xiaoqing Frank Liu

Missouri University of Science and Technology, [fliu@mst.edu](mailto:fliu@mst.edu)

Ravi Viswanathan

Follow this and additional works at: [https://scholarsmine.mst.edu/comsci\\_facwork](https://scholarsmine.mst.edu/comsci_facwork)



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

X. F. Liu and R. Viswanathan, "A WWW Based Software Metrics Environment for Software Process Management and Software Product Quality Improvement," *Proceedings of the 23rd Annual International Computer Software and Applications Conference, 1999*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1999.

The definitive version is available at <https://doi.org/10.1109/CMPSAC.1999.812721>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# A WWW Based Software Metrics Environment for Software Process Management and Software Product Quality Improvement

Xiaoqing (Frank) Liu  
Dept. of Computer Science  
University of Missouri-Rolla  
[fliu@cs.UMR.edu](mailto:fliu@cs.UMR.edu)

Ravi Viswanathan  
Dept. of Computer Science  
University of Missouri-Rolla  
[viswa@UMR.edu](mailto:viswa@UMR.edu)

## Abstract

*The software process needs to be continuously improved to develop high quality software. However, with increasing specialization in the workforce and decentralization in the workplace, software process planning, monitoring, analysis and dynamic tuning in a heterogeneous distributed environment becomes a challenge. In this paper we describe a tool which takes advantage of emerging internet technology to implement a software metrics environment for software process management and software quality improvement. The tool uses a dimensional analytic model to visualize the software development process. The system offers facilities to monitor the status and quality attributes of projects being developed at multiple sites and on multiple platforms based on internet, Java, JDBC and a database system.*

## Keywords

World Wide Web, Software Metrics, Software Project Management, Software Quality Improvement

## 1. Introduction

The quality of a software product is as good as the process that creates it. Several software process improvement methodologies have been developed, such as the Capability Maturity Model (CMM) [9] of the Software Engineering Institute and Total Quality Management (TQM) [10]. Many software metrics need to be collected and analyzed if processes are to be improved [8]. However, with the increasing specialization in the workforce and decentralization in the workplace, the collection and analysis of software metrics in a distributed environment becomes a challenge. Therefore, a WWW (World Wide Web) based toolkit for software process improvement and product quality management is developed.

### 1.1. Approach

The tool has been designed to work on the internet. This has meant the use of Java as the programming language and the use of packages such

as JDBC (Java Data Base Connectivity) [7] in making the tool. The most significant impact of this move is that the tool is accessible globally. Using the internet has also relieved the tool of being tied down to just one hardware platform and one operating system. It enables the tool to work in heterogeneous platforms. The tool has been implemented using Java, JDBC, and Oracle on a SUN Solaris 2.x platform.

### 1.2. Relevant Work

Much work already exists in the field of software metrics relating to software process management and software quality improvement [1, 2, 3]. Software metrics have been used in industry for software project management and software quality improvement [1, 2, 3]. Several software metrics systems have been developed recently although they are not developed on the internet [5, 6]. Recently, NASA's WISE Project Management System [4] uses the internet to provide the framework for issue management in software development, although its scope in applications of software metrics in software development is limited. Java has been used by a research project team at the University of Southern California as part of the COCOMO-II project[2]. One of the aims of this project is the development of a cross platform remotely accessible tool based on the COCOMO model for software development management.

## 2. Software Development Visualization

Software metrics can be collected and organized in many ways depending on its use. In this case, the primary use is to create a software metrics environment for an organization involved in writing software and developing software products. An object model to map the relationships between the entities in the software metrics environment has been developed. Software metrics and software metric attributes are classified into three broad categories: metrics relating to the organization, metrics relating to processes and metrics relating to products.

The organization metrics hierarchy is based on the structure of a typical company that is divided into divisions, groups and individuals. Each of them is a separate entity in the object model. Another hierarchical thread deals with projects. It is subdivided in to phases and tasks. Each task gives rise to many issues, which is also an entity. This thread tracks the metrics that are related to the process that creates the products. Software metrics associated with projects include cost, schedule, effort and defects in the process. The last hierarchical thread represents the structure of a software product. Each product is composed of multiple subsystems, each of which is made up of one or many modules.

This object oriented model is used to construct the actual database which is used to store the companies' software metrics and software quality data.

### **2.1. Dimensional Analytic Model**

It is important to visualize the project in order to be able to properly manage it. Software development can be usually visualized from multiple perspectives. It leads to the development of a dimensional analytic model for visualizing software development, as shown in Figure 1. There are two cubes in the figure, each of which contain three dimensions. The first cube, which is at the left, describes three dimensions associated with any software development: 1) project which describes the process of producing software, 2) product that describes the structure of a product and 3) organization which describes the structure of the company that produces the product. Each of these dimensions are broken into smaller hierarchical component structures. The second cube to the right describes the attributes that apply to all the dimensions through their hierarchies. Using an attribute from each of the dimensions, it is possible to describe the state of a particular artifact at a particular point of time. These attributes, such as cost or schedule, apply to all the hierarchical elements in all the three dimensions. By relating time to each of the dimensions and their related attributes, it is possible to see the trend in a particular attribute for an artifact from a particular dimension over a given period of time.

### **3. Architecture for Software Quality and Metrics Environment**

Being implemented on a distributed environment, the architecture of the system is basically a server - client type of system. The clients in this case are any non-specific remote machines that also communicate with the internet. The server is the central machine where the tool and the database that houses the software metrics data is stored. The numerous clients

and the server are connected through the WWW. Figure 4 shows the basic architecture of the entire system. Using the internet has the obvious advantage that the system is universally accessible and yet the cost is low. Most computers already have an internet connection and have Java enabled browsers.

## **4. Software Process Management Applications**

In this section we present a few examples that demonstrate the use of important aspects of this tool. Software metrics data is contained in the database that relates to the various projects that are in progress in a company. The projects and products are further broken down into logical sub-entities that relate to smaller aspects of the software development process. Only the main features of the tool will be explored in this section, by traversing through the object hierarchy. They show how a manager sitting at one location can track down problematic areas by using the reports available in the tool

### **4.1. Project Planning**

The first stage in any project involves planning for resources such as cost, schedule and man-power. Planning for the project cost, effort and schedule can be done with this tool. This model uses the COCOMO Early Design Model, and is expected to be used in the first stages of planning the software project, when many of the parameters are vaguely defined. The estimations of effort, cost and schedule are calculated based on user defined constants and cost drivers. If the value of these cost drivers is not known, the buttons next to the respective text box will lead to a lower level form to calculate the constant

### **4.2. Software Process Monitoring**

Monitoring the process provides data about problems in the process and clues that point to the possible cause of the problem. In this example, we visualize the software process through reports and queries provided by the tool and investigate problems with the process. This tool uses Report Generators to allow the user to define the type of report to create. In a typical project, a manager is faced with a choice of attributes such as cost, schedule, effort, or defects. These attributes can be examined with respect to almost any area of the project or any stage of the process. Figure 3 shows the Report Generator for the projects in the system. On the left hand side, one can notice the project hierarchy that reflects the one in the object model. On the right are the attributes that apply to a project such as cost and schedule. The

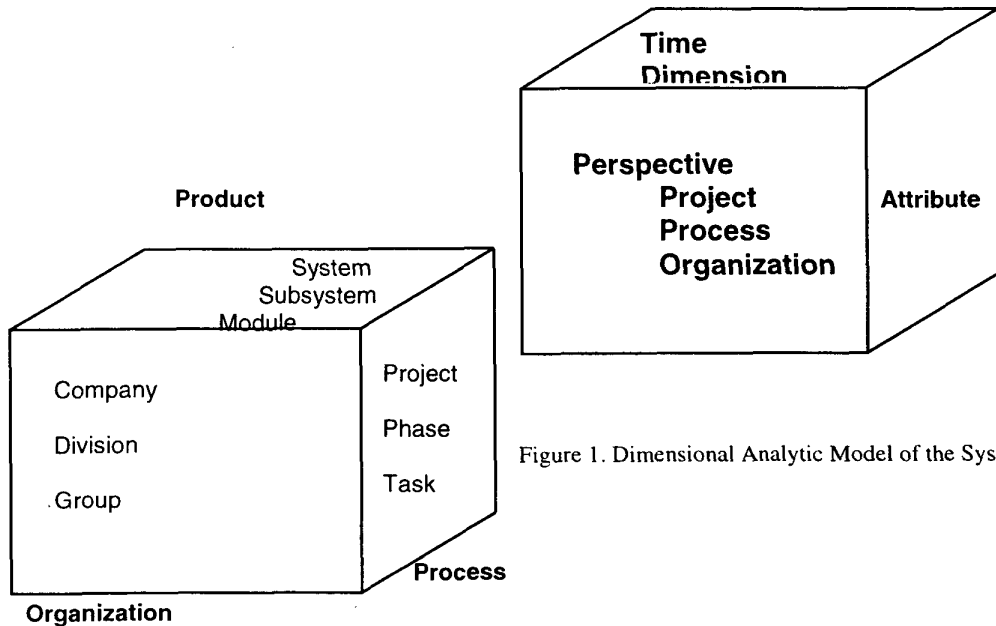


Figure 1. Dimensional Analytic Model of the System

three report generators are similar in structure and reflect the hierarchy detailed in the Object Model.

Let us assume that projects have a problem of delayed schedules and the manager wants to localize the problem to an area as small as possible. Suppose that the item 'All Projects' is selected for the 'Project' element and 'Schedule' is selected for the attribute. This implies that the schedule for all projects will be displayed. The resulting chart is shown in Figure 4.

We can see that the project P151 has the most amount of delay, of 12 days.

We can "zoom" further into a particular area of interest using the report generator. Therefore, starting out with all the projects in the company, a manager is able to narrow the cause of delayed schedules to a single task and a group of individuals responsible for that task.

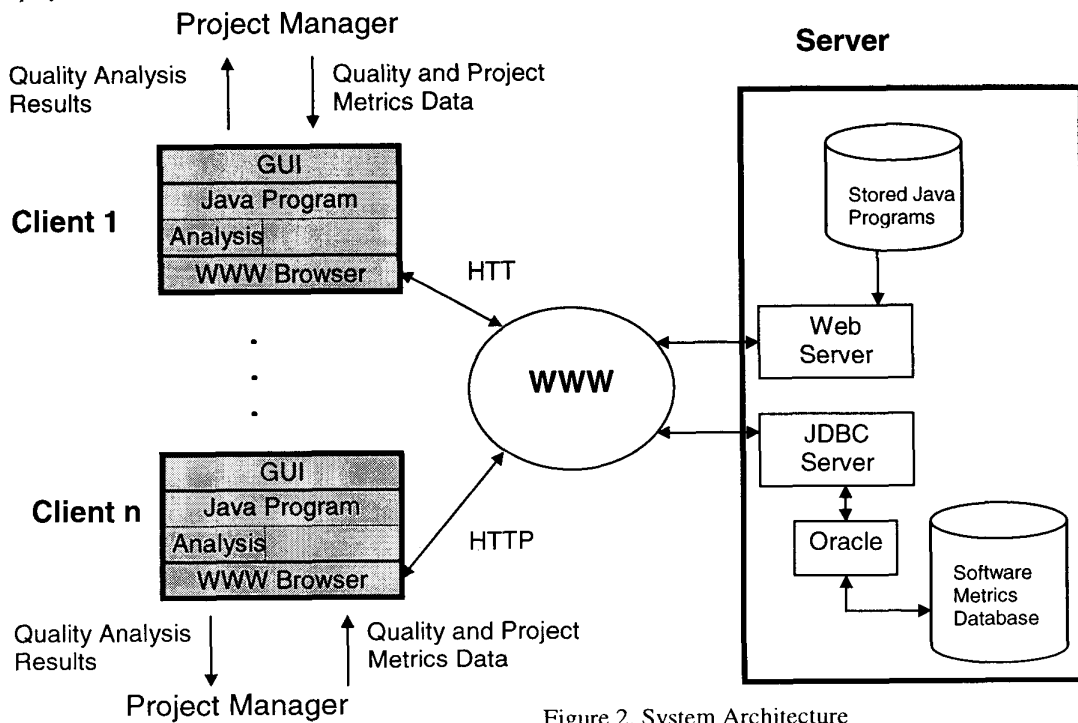


Figure 2. System Architecture

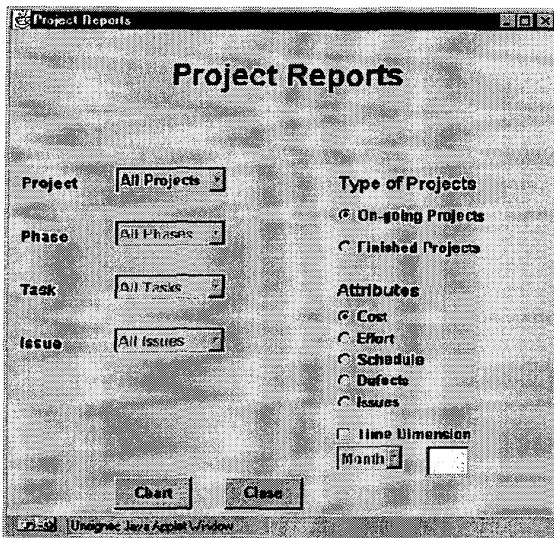


Figure 3. Report Generators for Projects

#### 4.5. Using Queries for Problem Identification

Reports provide a useful method of monitoring software development. Often times, it is required to get to the problem areas quickly without having to go through all the levels of reports. This is important when dealing with a large software metrics database where the user or the manager has to discover important information from the reports. Queries are very similar to reports and they work with Query Generators that parallel with Report Generators in structure. The important difference is that results generated by queries contain only aberrant information. For example, it could show projects that are delayed beyond a threshold level or products that have defects above the control line. This functionality allows the manager to focus only on the projects that need urgent attention.

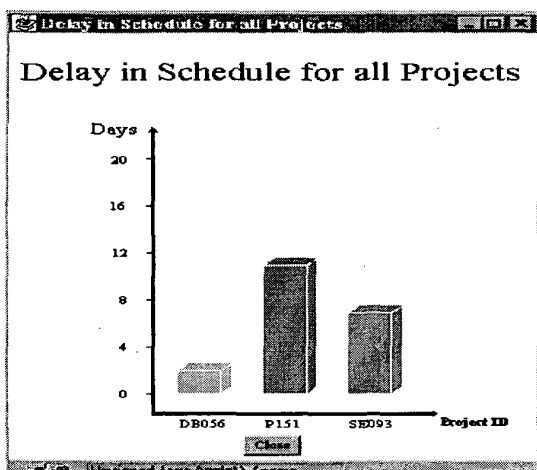


Figure 4. Chart generated for all Projects

## 5. Conclusion

In this paper, we have described in detail the design and implementation of a WWW based software metrics environment for software project management and software quality improvement based on Java, JDBC, and Oracle database system. It enables acquisition and analysis of software metrics through the internet. It can facilitate the identification of software development problems as early as possible by visualizing software development from multiple perspectives based on its dimensional analytic model and monitoring its progress in a timely fashion. Several examples have been given to demonstrate its benefit for software process management and quality improvement in a heterogeneous distributed environment.

## References

- [1] Barry W. Boehm, "Software Engineering Economics", Prentice Hall Inc., 1981.
- [2] Barry W. Boehm, COCOMO - 2 Research Project, University of Southern California, 1996. Available at the site <http://sunset.usc.edu/COCOMOII/Cocomo.html>
- [3] Robert B. Grady and Deborah L. Caswell, "Software Metrics: Establishing a Company-Wide Program", PTR Prentice Hall Inc., 1987.
- [4] NASA Software IV&V Facility, "The WWW Integrated Software Metrics Environment". Available at <http://research.ivv.nasa.gov/projects/WISE/wise.html>, 1995.
- [5] Dick B. Simmons, Newton C. Ellis, and Way Kuo, "Software Process Agents", *Proc. of the SEKE-96*, Lake Tahoe, Nevada, 1996.
- [6] C. L. Chee, S. Jarzabek, and C. V. Ramamoorthy, "An Intelligent Process for Formulating and Answering Project Queries", *Proc. of the SEKE-96*, Lake Tahoe, Nevada, 1996.
- [7] Jerrid Hamann, "Analysis of Java Database Connectivity and its Application in a Multi-Platform, Multi-DBMS Environment", CPSC485 Report, Texas A&M University, College Station, Texas, 1996.
- [8] Dick B. Simmons, "A Win-Win Metric Based Software Management", *IEEE Transaction on Engineering Management*, Feb. 1992.
- [9] Pualk M. et al., "Capability Maturity Model for Software", SEI, Carnegie Mellon University, Pittsburgh, PA, 1993.
- [10] Lowell Jay Arthur, "Improving Software Quality: An Insiders Guide to TQM", John Wiley & Sons Inc., New York, 1995.