

# Automated Software Sizing From Use Case Points And Requirements Repositories

Daniel D. Galorath, Galorath, Inc.

310-414-3222

[Galorath@galorath.com](mailto:Galorath@galorath.com)

Daniel V. Ferens, Air Force Research Laboratory (AFRL)

315-330-4098

[Daniel.ferens@rl.af.mil](mailto:Daniel.ferens@rl.af.mil)

Lee Fischman, Galorath, Inc.

310-414-3222

## Introduction

Software size is the key input to all software cost estimating models. There are several popular methods for estimating software size, including source lines of code, objects and function points; however, estimating these inputs early in a program can be difficult. A method or methods to estimate software size early in a program from high-level information such as requirements specifications and logical system models would be useful, particularly if the methods are automated.

This paper discusses current research in automatic sizing tools in two areas: use case points and requirements repositories. Both areas show promise in being able to accurately estimate program size early in a program.

## History of Software Sizing<sup>1</sup>

“Size”, a measure of software “volume”, mass or functionality, has always been an input to most, if not all software cost models. When PRICE-S was released in 1977, number of (machine-level) instructions was the key input. The SLIM model, first released in 1979, also had size in lines of code as a key input. The 1981 edition of COCOMO had thousands of delivered source instructions (KDSI) as the key input to the effort equation for the model. Despite the importance of size, there were no methods for estimating size that were widely accepted. A software cost model user could often only guess what this parameter would be.

During the mid-1980s, several size estimation tools were developed. The Aerospace Corporation developed a sophisticated database analogy model for estimating the size of space software programs. Dr. George Bozoki developed the Software Sizing Model (SSM), an expert judgment model which estimated software size based on pair-wise comparisons and PERT size estimates. PRICE Systems developed the PRICE Sizer (PRICE-SZ) model to estimate the size of military and commercial programs based on parametric inputs. Function points, another size measure, became increasingly popular during this period. None of these measures, however, provided consistently accurate measures of size that were applicable to a wide variety of programs and useful early in a program.

The 1990s witnessed the development of more size estimating methods and refinements of existing methods. Several variants of function points arose such as Capers Jones’ Feature Points, Reifer’s ASSET-R Function Points, and Simon’s Mark II Function Points. Bozoki’s ideas were incorporated into Galorath Inc.’s SEER-SSM, a commercially available software sizing model. With the increasing use of object-oriented design for software, a new size measure, object points, was proposed. There are now several variants of object points formulated, including Banker’s object points which form the basis for the Application Composition model for COCOMO II, Predictive Object Points used in PRICE-S, and object-based sizing available in SEER-SEM. Still, there was no measure that was applicable to a wide variety programs and useful early in a program.

---

<sup>1</sup> For more information about the history of software sizing, see Mr. Ferens’ article *Software Size Estimation: Quo Vadis?* listed in the references at the end of this paper.

## **The Research Program**

In 2001, the Air Force Research Laboratory awarded a Phase I Small Business Innovative Research (SBIR) contract to Galorath Inc. to study software size estimation methods that would be useful early in a program based on the architecture of the software being developed. Galorath researchers hypothesized that three types of documents could be used. They were use cases, structured requirements documents, and textual specifications. The opportunity for each method is now briefly explained; use cases and requirements repositories will be explained later in more depth.

### ***Use Cases***

Use cases comprise a notional description of a system, frequently used at the earliest stages in a project. Use cases are one type of graphically oriented notation in the Unified Modeling Language (UML), a family of notational methods (diagrams) used to describe various aspects of software and its underlying structures. Some of these, for example use cases and class diagrams, can be used as part of the early analysis process.

At least two descriptive characteristics of use cases can be exploited to provide sizing information. The first is the number of use cases. A small application might have only one use case while very large applications may have hundreds. The other major characteristic is the number of “actors” per use case. Actors are the “agents” that interact with the software and so use cases must have at least one actor. A commonly accepted standard is one actor per use case though this is by no means the rule. Use cases and actors can be weighted for complexity to provide a refined “use case points” estimate. Use case points are the leading metric for quantifying system size based solely on use cases.

### ***Structured Requirements Documents***

It may be possible to estimate size from a requirements document or specification by investigating the number, type and complexity of requirements compared with similar documents for which size is known. Some of the techniques now found in SEER-SSM, such as pair-wise comparison, can be useful here. What is needed, of course, is a repository where past requirements documents and corresponding software sizes are available.

### ***Textual Analysis***

This is related to structured requirements in that it may be possible to estimate size based on technical artifacts, such as word count and phrases. Information extraction and other capabilities may be able to derive size from text documents describing software. Information extraction tools have been developed that can automatically extract machine-readable information from documents. The field of Information Extraction has left the purely academic arena and is making inroads in commercial applications such as web searching and data mining. Perhaps it can be used to correlate textual characteristics with the size of a software product that the text indirectly describes.

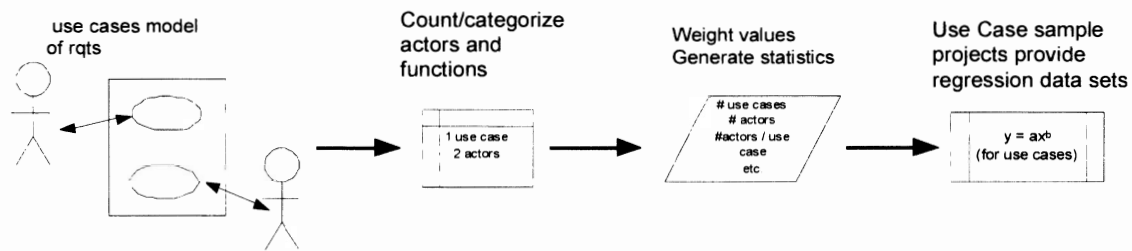
## **Phase I Results<sup>2</sup>**

The Phase I effort concentrated on use case points. For this work, five software programs were chosen. These were existing, commercial quality programs, enabling direct comparisons against

---

<sup>2</sup> The Phase I results have been published in a Technical Report, “Automated Software Project Size Estimation Via Use Case Points” available as Defense Technical Information Center (DTIC) Report AD-B-284575.

actual values. Each program was ‘specified’ into use cases by an experienced UML analyst, and then counted by an experienced, certified function point analyst. Using experts in each of these areas helped to ensure that use cases and function point counts were consistent and correct. Figure 1 shows the general process used.



**Figure 1. Method of analyzing use cases**

The sample of five programs was then evaluated by modeling and estimating specialists to determine their use cases’ potential to indicate a corresponding function point count.

The goal was to estimate function points from use case data, using the sample of five systems. For this analysis ordinary least squares regressions were used on a set of specially transformed variables. Later stages of this project will use more data and more sophisticated estimating methodologies.

The equations estimated through regression analysis were intended to explain function points in terms of UML use case characteristics. For the use case data (dependent variables, those being used to estimate) we used these variables:

- “adjusted use cases”
- “adjusted actors”
- scenarios

Developing “adjusted use cases” and “adjusted actors” was a two-part process. The first step was to assign impartial ratings of “low complexity”, “average complexity” or “high complexity” depending on the number of transactions (use cases) or depending on the type of interaction with the system (actors). In the second step, the use cases and actors were weighted according to the complexity levels.

Based on a graphical analysis of the data, as well as *a priori* theoretical considerations, several different function forms were considered. Variations included whether the dependent variable and one or more independent variables should be logged, whether the independent variables used should be adjusted or unadjusted, and the overall shape of the function. Several of the more promising functional forms were estimated. The result indicated the potential for good fits between function points and use cases, although due to the limited nature of the data and consequent lack of verifiability, results are not reported here. Use of additional data and more sophisticated methodologies, including simultaneous estimation of a system, should provide good estimates for the other function point measures.

Additionally, this methodology was applied to a large system by both a contractor and Galorath researchers. The result was an estimate that was within 12 percent of actual effort, with input provided by knowledgeable engineers, and within 20% when inputs were provided by cost analysts without program knowledge.

## Phase II Progress

The success of Phase I resulted in the Missile Defense Agency awarding a Phase II SBIR to Galorath to further study software sizing and to commercialize their product. The Phase II effort was begun in June, 2002 and is scheduled for completion in June, 2004. A kick-off meeting was held at Galorath in July, 2002 and technical reviews were held in January and July 2003. The progress now discussed is as of September, 2003.

Galorath has named the planned product of the Phase II effort, "CriticalMass". It is envisioned that CriticalMass will be a standalone tool and will be commercialized as part of the SEER family of commercial models. This is in line with the purpose of the SBIR program, which is to assist small businesses (like Galorath) in research and development of value to the Department of Defense.

Figure 2 illustrates the basic concept behind CriticalMass. The inputs will either be use case point or requirements specifications. The outputs will be size estimates in SLOC, function points, or whatever size input is appropriate for the cost model being used.

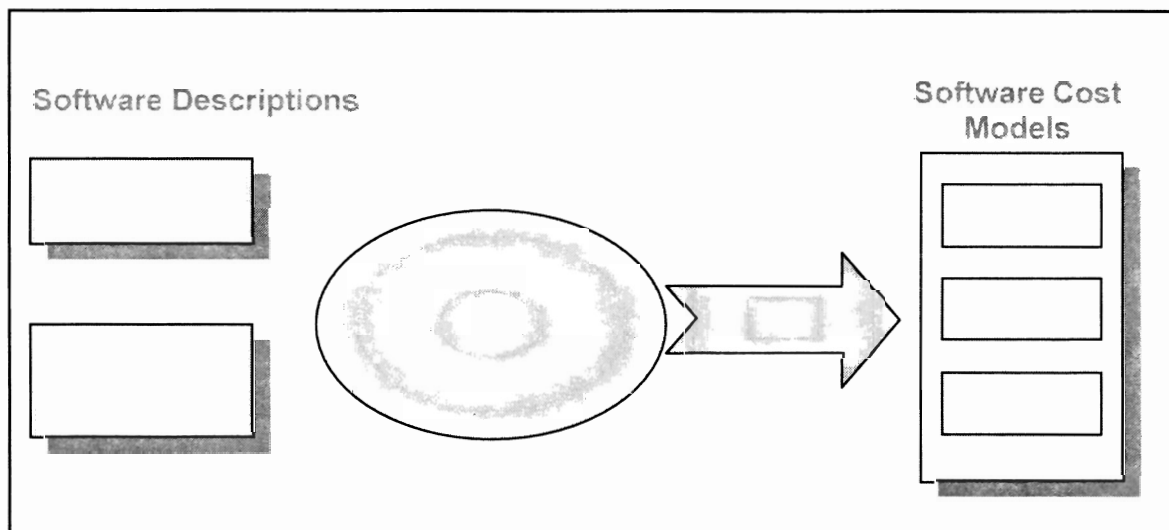


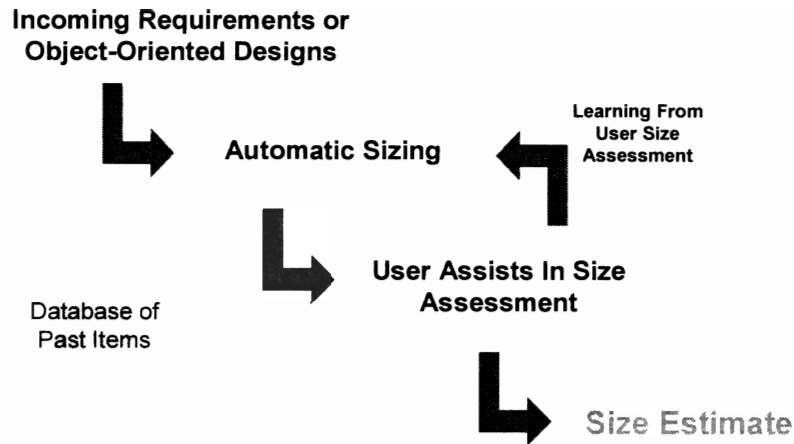
Figure 2: CriticalMass Concept

In Figure 2, it is evident that the number of software descriptors has been reduced from three to two. Textual analysis has been subsumed into requirements documents. Progress for each of the two software descriptors is now addressed.

### **Common Features**

CriticalMass is implemented in two versions, for DOORS and Rational Rose. The former product is a requirements repository and the latter is used for specifying use cases. Both CriticalMass versions are tightly coupled with their respective products, extracting the information from each that is necessary to obtain size estimates. Sizing is dynamic; there is not a set of static coefficients that can be applied under all circumstances.

Both tools are further enabled with a user-assisted pair-wise comparisons feature. This method (accurate in itself) comes into play when the user has a particularly good feel for certain items' relative size, and wishes to override automatic sizing.



### ***CriticalMass For Rational Rose: Sizing From Use Cases***

CriticalMass for IBM/Rational's Rose tool will permit translation of use case diagrams into size estimates, letting planners achieve very early, efficient estimates of system size based on little more than notional designs.

Galorath performed further research on use cases, the area they were successful with in the Phase I effort, and developed a modified metric called "normalized use cases". This metric has promise to be more robust and accurate than use case points, which to date has been the only widespread, though imperfect, metric for use case sizing.

At this time, CriticalMass is linked with the Rational Rose tool, from which it extracts use cases and against which preliminary estimates are obtained. The estimation algorithms still must be refined and calibrated before reliable estimates can be produced.

### ***CriticalMass for DOORS: Sizing From Requirements Repositories***

CriticalMass for Telelogic's DOORS requirements repository - and potentially other repositories - will provide an efficient opportunity to compare new requirements with those of known size so that rapid assessments of future size can be made.

Galorath has at this time integrated CriticalMass with the DOORS requirements repository and is linking the data that is extracted (alongside numerous derived metrics) with algorithms intended to estimate the size of extracted requirements.

### **Conclusion**

Critical Mass is expected to be completed by June, 2004 with earlier beta versions. The system will no doubt evolve as different development shops gain experience with it and offer their feedback. In fact, with its ability to dynamically calibrate and incorporate past results, evolution is built into its framework.

### **Bibliography**

Abran, Alain and Pierre N. Robillard, *Function Points Analysis: An Empirical Study of Its Measurement Processes*, IEEE Transactions on Software Engineering, 22(12), pp. 895-910, December 1996

Albrecht, Allan J., Gaffney, John E., *Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation*, IEEE Transactions on Software Engineering, Vol. SE-9, no. 6, pp. 639-648, Nov. 1983

Bock, Douglas B., Klepper, Robert, *FP-S: a simplified function point count method*, The Journal of Systems and Software, July 1992 vol 18 no 3 p245 (10)

Briand, Lionel and Khaled El Emam and Sandro Morasca, *Theoretical and Empirical Validation of Software Product Measures*, Technical Report, Centre de Recherche Informatique de Montréal, Number ISERN-95-03, 1995

Ferens, Daniel V., *Software Size Estimation: Quo Vadis?*, National Estimator, Winter, 1999, pp. 43-54.

Fischman, Lee, *et al*, "Automated Software Project Size Estimation Via Use Case Points" (DTIC Technical Report AD-B-284575), 2002.

Fischman, Lee, *The Place of Function Points In An Underlying Model of Software Content*, presented at the 1999 IFPUG National Conference, available at [www.galorath.com](http://www.galorath.com)

Fischman, Lee, "Analysis of Function Points In A Tree-Based Framework", presented at the International Workshop on Software Metrics, Lac-Superieur, Quebec, fall 1999.

Fischman, Lee, "UML-Based Software Sizing", presented at the International Society of Parametric Analysts / Society for Cost Estimating and Analysis national conference, 1999 and at COCOMO 1999.

Fischman, Lee, *Function Point Counting For Mere Mortals*, presented at the 1999 Applications of Software Metrics conference, available at [www.galorath.com](http://www.galorath.com)

Galorath, Daniel, "Managed Software Development", book in progress; McGraw Hill, 2002.

2000: Galorath, D., Stump, E., "The Challenge and Opportunity of Low Level Parametric Estimating," presented at the 2000 ISPA Conference, Netherlands.

Galorath, D., Fischman, L., "Revisiting Size Expansion Factors," Discussion of the Galorath function point to SLOC expansion table. Presented at the International Society of Parametric Analysts national conference, spring 2000.

Galorath, D., "Determining Quality Through Parametrics," describes the many ways in which modeling can be applied to improve the quality of delivered software. Published in Crosstalk: The Journal of Defense Software Engineering, December 1998.

Galorath, D., Fischman, L, Et Al, "OSD Software Estimation Guide," for the largest software developer in the United States, a 300-page guide to the economic evaluation of software projects., 1997.

GartnerGroup, "Why Galorath Matters", 1998.

Harrison, Warren and Gene Miluk, The Impact of Within Size Variability on Software Sizing Models.

Haseltine. Bill, "Applying Unified Modeling Language Use Cases to Software Estimating", presented at the Third Joint Annual ISPA/SCEA Conference, Vienna, VA, June, 2001.

Ho, V.T., Abran, A., Oligny, S., Using COSMIC-FFP to Quantify Functional Reuse in Software Development, Escom-Scope 2000.

International Function Points User Group, *IFPUG Counting Practices Manual, Version 4.1*, [www.ifpug.org](http://www.ifpug.org)

Jones, Capers "Feature Points (Function Point Logic for Real Time and System Software)", presented at the Fall 1988 IFPUG National Conference

Kemerer, Chris F., *Reliability of Function Points Measurement: A Field Experiment*, MIT Sloan School of Management WP#3193-90-MSA

Kitchenham, Barbara and Shari L. Pfleeger and Norman Fenton, *Towards a Framework for Software Measurement Validation*, IEEE Transactions on Software Engineering, 21(12), pp. 929-943, December 1995

## **Biographies**

### ***Daniel D. Galorath***

During his 31 years in the computer industry, Mr. Galorath has been solving a variety of management, costing, systems, and software problems for both information technology and embedded systems. He has performed all aspects of software development and software management. One of his strengths has been reorganizing troubled software projects, assessing their progress applying methodology and plans for completion and estimated cost to complete. He has personally managed some of these projects to successful completion. He has created and implemented software management policies, and reorganized (as well as designed and managed) development projects. His company, Galorath Incorporated, has developed tools, methods, and training for software cost, schedule, risk analysis, and management decision support. He is one of the principal developers of the SEER-SEM™ software evaluation model. His teaching experience includes development and presentation of courses in Software Cost, Schedule, and Risk Analysis; Software Management; Software Engineering; and Weapons Systems Architecture. Mr. Galorath has lectured internationally. Among Mr. Galorath's published works are papers encompassing software cost modeling, testing theory, software life cycle error prediction and reduction, and software and systems requirements definition. Most recently, Mr. Galorath was named winner of the 2001 International Society of Parametric Analysts (ISPA) Freiman Award. Named after ISPA's founder, Frank Freiman, the prestigious honor is awarded to those individuals who have made outstanding contributions to the theoretical or applied aspects of parametric modeling.

### ***Daniel V. Ferens***

Daniel V. Ferens is currently a program manager at the Air Force Research Laboratory's Information Directorate in Rome, New York. Here, he manages two Phase II Small Business Innovative Research (SBIR) efforts with Galorath, Inc. in the area of affordability. Prior to his Rome assignment, he served two years as the Information Directorate's Corporate Affordability Officer at Wright-Patterson AFB in Dayton, Ohio, where he directed cost analysis efforts for selected laboratory programs. He also directed several affordability-related SBIR efforts and consulted with several major programs regarding software affordability issues. He was also a frequent guest lecturer at the Air Force Institute of Technology in the areas of software estimation and software management in general.

Mr. Ferens has a Master's Degree in Electrical Engineering from Rensselaer Polytechnic Institute and a Master's Degree in Business from the University of Northern Colorado. Mr. Ferens is a lifetime member of the International Society of Parametric Analysts (ISPA) and has been an active member of the Society of Cost Estimating and Analysis (SCEA) for ten years. He was the ISPA Freiman Award winner in 1999, and the ISPA Parametrician of the Year in 1990. He has presented papers or chaired panels at numerous national and international conferences. He has had several articles published in refereed professional journals, and wrote two books on software management in the Department of Defense. He is listed in the 2003 editions of Marquis *Who's Who in America* and *Who's Who in the World*.

### ***Lee Fischman***

Lee Fischman is Director of Special Projects at Galorath Incorporated, where he is engaged in the conception, research and development of new product lines. Mr. Fischman studied economics at the University of Chicago (BA) and UCLA. He has been developing software for the academic, publishing, financial and estimation industries since the mid-1980s.