Monte Carlo Simulation for Software Cost Estimation

Pete MacDonald

Fatma Mili, PhD.

Definition

- Software Maintenance - The activities involved in implementing a set of relatively small changes to an existing program. Barry Boehm, *Software Engineering Economics*, Prentice-Hall, 1981

Quotes

- "Estimating maintenance is rather a tricky problem." Caper Jones, *Estimating Software Costs*, McGraw-Hill 1998
- "Inaccurate size estimates automatically translate into questionable budgets." Eduardo Miranda, *Improving Subjective Estimates Using Paired Comparisons*, IEEE Software, Jan./Feb. 2001

In the Beginning

- Before you can have Software Maintenance, there exists new development
- For new or substantially new development estimation is accomplished via COCOMO, etc.
- After several projects, REVIC estimates were within 10% of actuals if SLOC estimate was accurate

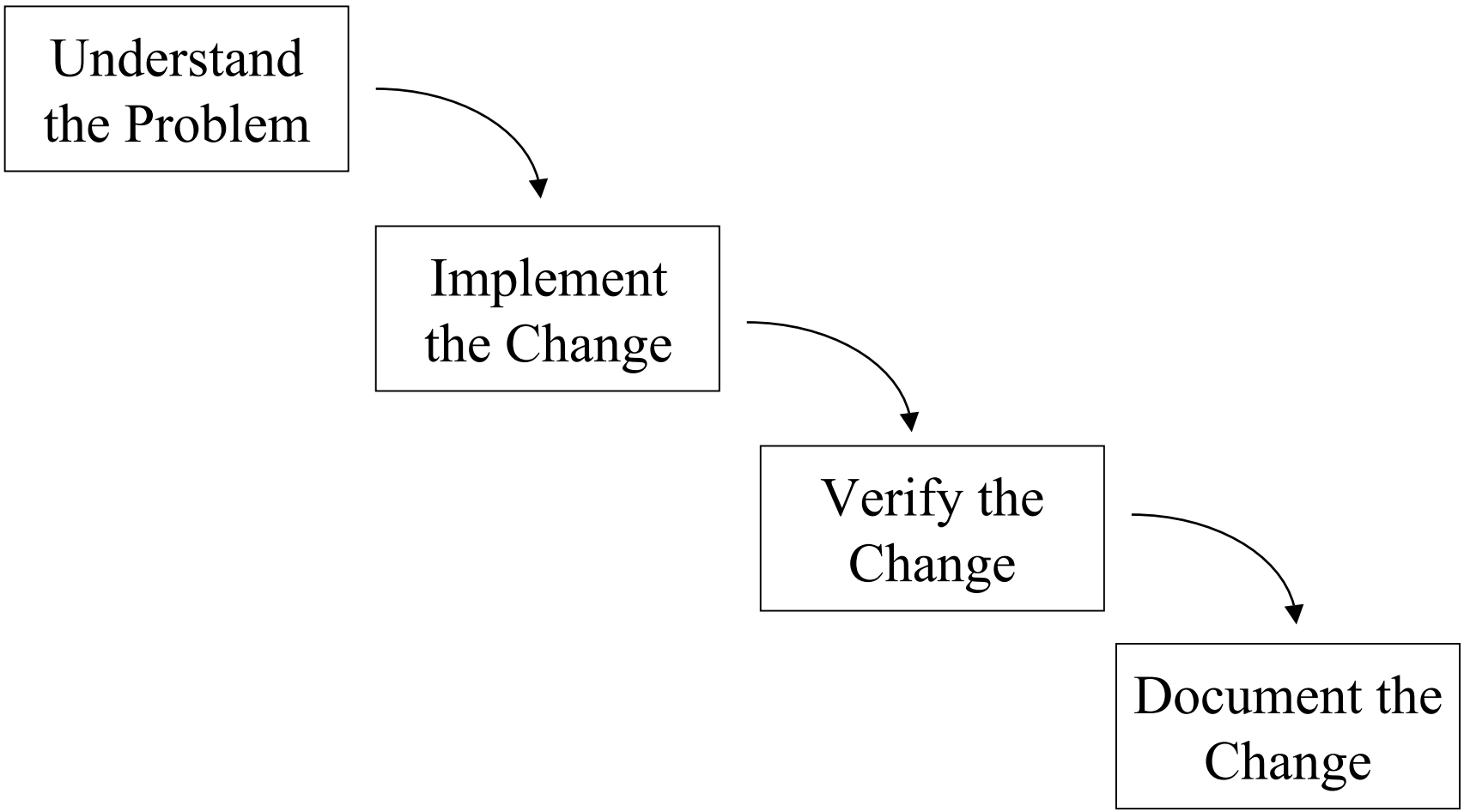
Typical New Development

- Starts with a set of user requirements
- Analyze these requirements to develop software requirements
- Produce a design
- Implement the design by coding and testing

Software Maintenance

- Frequently performed by “newer” employees
- Understand the problem
  - There is a known problem with the code
  - Typically requires instrumenting the code
- After the problem is identified
  - The code is modified and tested.
  - Changes made to the code are then reflected in the analysis and design documents

Maintenance Life-Cycle Model



Discussion

- Understand the problem - typically requires instrumenting the software to discover the problem location and/or potential solutions
- Implement the change - coding a potential correction into the software
- Verify the Change - test the change to verify the problem has been corrected
- Document the Change - Update the Software Requirements and/or Design Documents to reflect change in the software

SW estimating at GD

- Evolved via software process improvement activities
- Initially performed using REVIC
  - Once calibrated, estimates were within 10% if SLOC estimate was correct
- In ‘95 started software maintenance of existing software
- Attempted to apply REVIC

Initial Approach

- Difficult to estimate % of change
  - Design, Code & Test
- Looked at effort required
  - Used an average
  - Eliminated high and low outliers
- Went to engineering judgement
  - More accurate but difficult to quantify

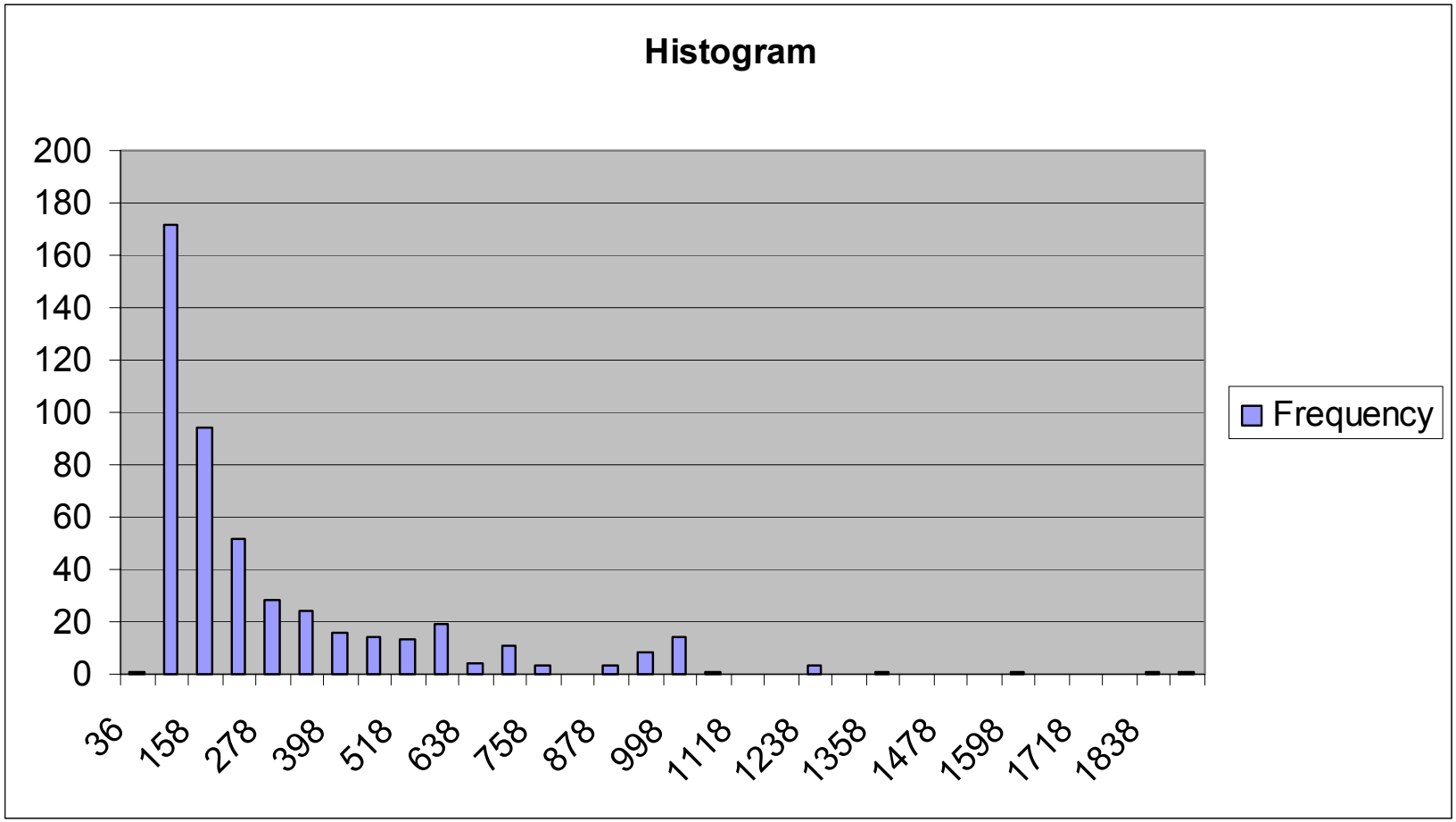
Initial Approach continued

- The way REVIC was applied led to inaccurate estimates
- Reverted to Engineering Estimates
- Example - Modification requires 25 SLOC
  - Is percentage of the Program or File?
  - Are Integration changes only the amount of tests changed or include the effort to execute the required test cases?

Second Approach

- Effort data for changes were available
- Discarded the outliers
- Take mean of remaining data and use as a "typical" Change
- Tended to cause an under estimation.
- Again Reverted to Engineering Estimate

Effort Distribution



Engineering Judgment

- Accomplished by senior developers
- May incorporate “Delphi technique”
- Difficult to quantify and/or justify
- Lacks the structure of a parametric model
- Historical data only in the memory of the engineer

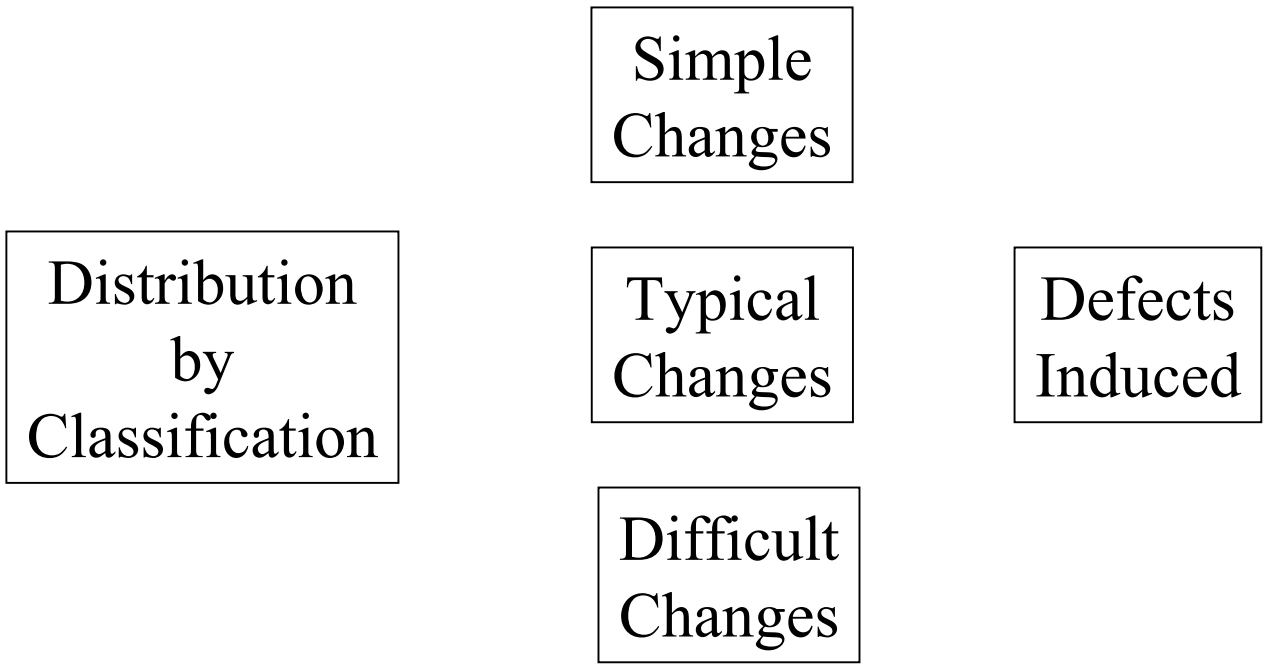
Monte Carlo Simulation

- Necessitates the use of multiple random distribution functions
  - Single distribution would like using the mean
- Distributions are slightly overlapping subsets of historical effort data
- Run multiple simulations to produce an estimate
  - Typically requires a minimum of 100 runs
  - Take the average of the runs.

Current Approach

- Apply Monte Carlo Simulation Model
- Senior Developers classify changes as
  - Simple, Typical or Difficult
- Each classification has its own distribution function
- Implementing a change into a baseline can induce the requirement to include additional changes

Model Overview



- Total effort = Sum of the hours estimated to all changes plus the effort to correct induced defects

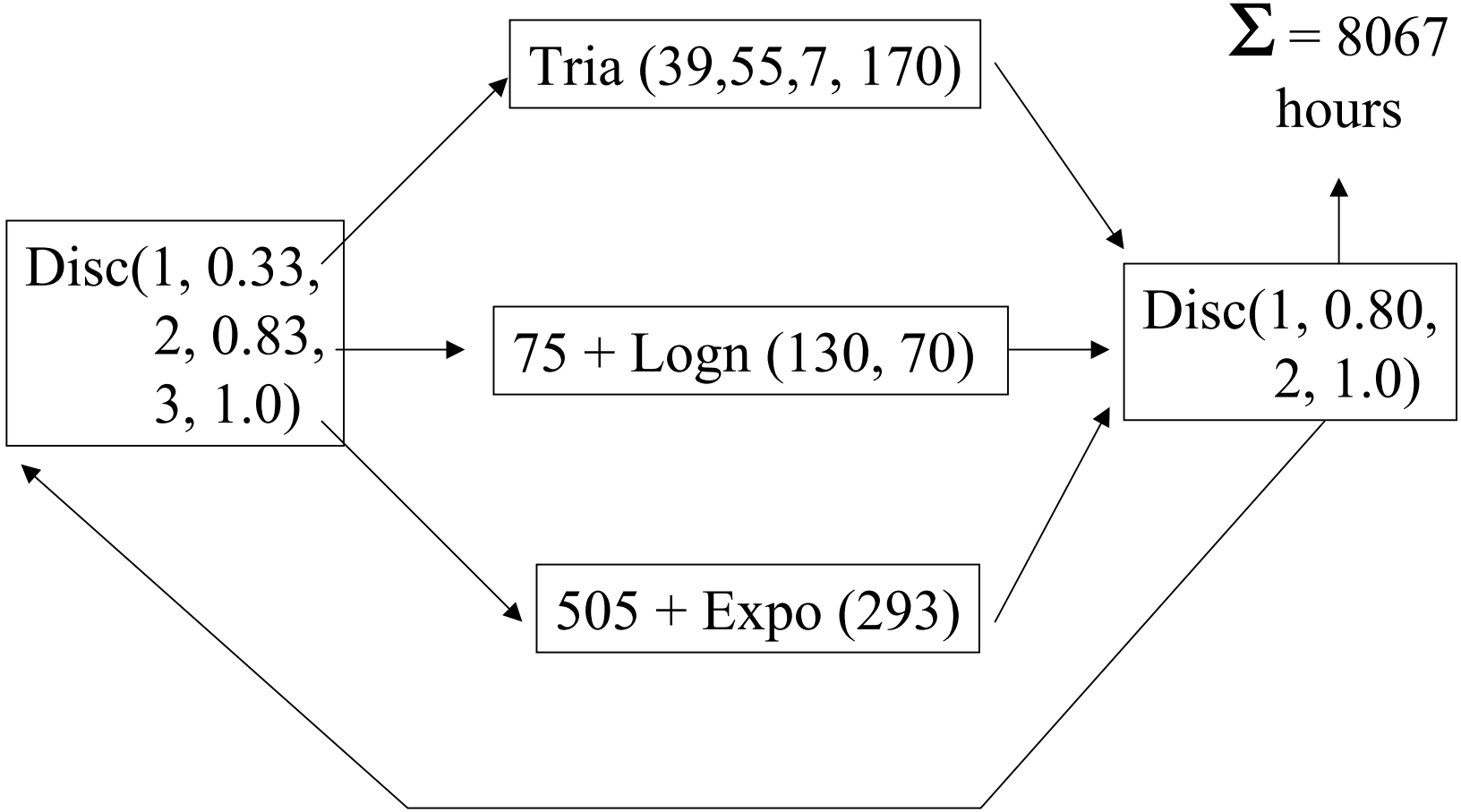
Defining the Distribution Functions

- Historical effort data is collected and maintained in ascending order
- Must have at least 40 data points to generate a distribution function
- Must validate each distribution
  - Mean square error < 0.005
  - Chi-Square and Kolmogorov- Smirnov tests > 0.05

An Example

- Given 30 changes to be implemented as follows
  - 10 easy
  - 15 typical
  - 5 difficult
- Produce an estimate to implement the changes.

Example Continued



Results utilizing Arena

Project: SW Estimator **Run execution date : 9/24/2002**
Analyst: Pete MacDonald **Model revision date: 9/24/2002**

OUTPUTS

Identifier	Average	Half-width	Min	Max	Replications
Duration	1976.9	26.61	1651.1	2405.2	100
Effort	8065.9	346.18	4526.6	14681.	100

Simulation run time: 0.08 minutes.

Comparison of Methods

No. of Changes	36	30	16
Actual	14710	9429	8468
Typical Fix	9000	7500	4000
% Error	38%	20%	53%
3rd Try	11465	7115	6936
% Error	22%	14%	18%
Current	14369	9848	8023
% Error	2.5%	4.5%	5%

Where do we go from here

- Automate generation of random distribution functions.
- Evaluate induced error data.
- Evaluate the application of Monte Carlo simulation as a Systems Engineering estimation tool.
- Include Engineer effectiveness factor.