Practical Object-Oriented Measurement

Khaled El Emam

© 2000-2001 k sharp technology inc. All rights reserved.

Agenda

- Introduction to Object-Oriented Measurement
- Definition of Object-Oriented Measures
- Cost Estimation for Object-Oriented Projects
- Quality Management
  - Preventative action
  - Quality prediction
  - Design guidelines
- Methodological Considerations
  - Confounding effects
  - Optimal class size & faults
  - ROI calculations
- Conclusions

© 2000-2001 k sharp technology inc. All rights reserved.

Management Decisions

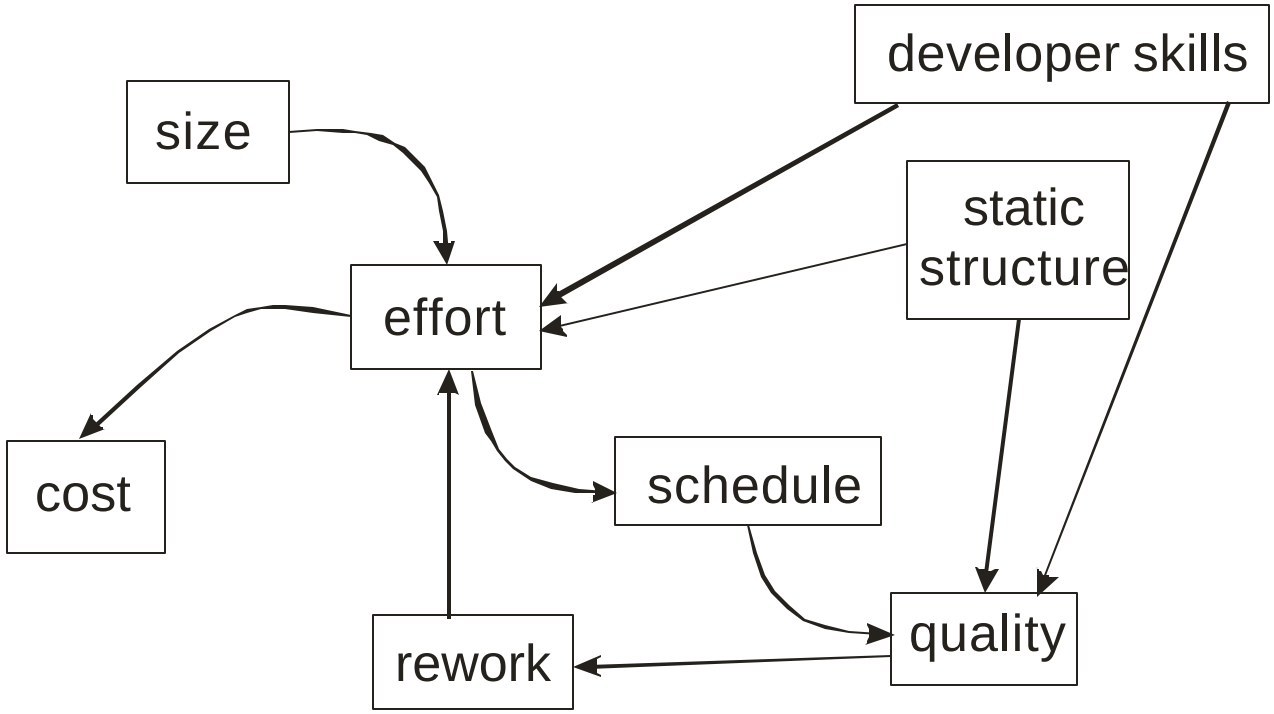
Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- The main purpose of collecting measures is to support management decisions, otherwise measurement is just a cost
- Measurement can be used for:
  - cost estimation
  - progress tracking
  - quality estimation
  - early warning of problems
  - optimization of project
  - benchmarking
- We will talk about measurement issues that are unique to object-oriented projects

© 2000-2001 k sharp technology inc. All rights reserved.

Performance Analysis Model

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Do We Really Need Measurement ?

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- For small projects (2-3 people) where the staff are very skilled and experienced, small number of known users, and the application is well understood, then probably not …
- But it is necessary for:
  - large projects (10's or 100's of developers)
  - average staff or wide variability in skills
  - many subcontractors
  - new technology
  - critical projects
  - competitive bidding for contracts
  - long lasting projects

© 2000-2001 k sharp technology inc. All rights reserved.

Types of Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Object-oriented measures quantitatively characterize the structure of an OO software system
  - you can also have qualitative characterizations of OO systems, such as “more complex than” or “elegant design”
  - qualitative characterizations are not practical except for small and well understood systems
- We are only concerned with *static measures* that can be collected from design documents or source code
  - Dynamic measures include test coverage and execution profile
  - Dynamic measures require an executing program in order to be collected

© 2000-2001 k sharp technology inc. All rights reserved.

Levels of OO Measurement

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- There are many levels at which measures can be collected:
  - **file:** not very useful
  - **method:** can be useful
  - **class:** most general utility
  - **system:** project estimation and tracking
- We will focus mostly on class and system level measures
- Class and system level measures can be collected early on in a project. The others require more detailed knowledge of the design and implementation.

© 2000-2001 k sharp technology inc. All rights reserved.

Stage of OO Measurement

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- OO Measures can be collected at different stages of a project
- Early measurement is useful for
  - cost and quality estimation
  - optimization of project
  - early warning of problems
  - benchmarking
- Measurement later in the project is useful for
  - tracking
  - updating estimates
  - benchmarking
- Ideally, measurement should start during the analysis phase of a project

© 2000-2001 k sharp technology inc. All rights reserved.

Categories of OO Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

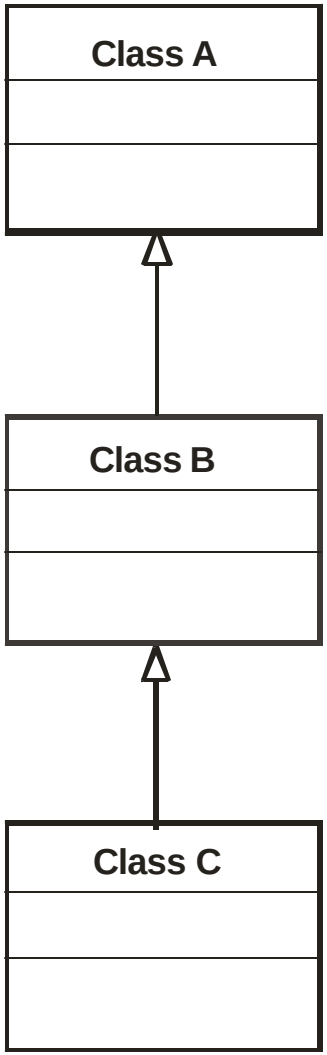
- Many OO measures have been developed, at least 200 at last count
- Most are very similar to each other
- There are five fundamental categories of OO measures:
  - size (measure how big a class or system is)
  - inheritance (characterize inheritance hierarchy)
  - coupling (characterize relations amongst classes)
  - cohesion (characterize relations within classes)
  - complexity (aggregates of method measures)
- Complexity is not a real OO measure, so we will not consider it further

© 2000-2001 k sharp technology inc. All rights reserved.

Example of Inheritance Measure: DIT

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



DIT=0

DIT=1

DIT=2

© 2000-2001 k sharp technology inc. All rights reserved.

Inheritance Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- The most common and useful inheritance measure is DIT
- At the class level, DIT measures depth of a class in an inheritance hierarchy
- At the system level, you still have to calculate the per class DIT and then compute the maximum DIT to determine the largest depth of the inheritance hierarchy
- There are strong cognitive arguments as to why DIT is an important static measure
- Other measures, such as NOC (number of children) have not been shown through repeated empirical study to be useful
- Typically, inheritance hierarchies tends to be shallow (except for GUIs and frameworks) and multiple inheritance is not used often

© 2000-2001 k sharp technology inc. All rights reserved.

Coupling Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

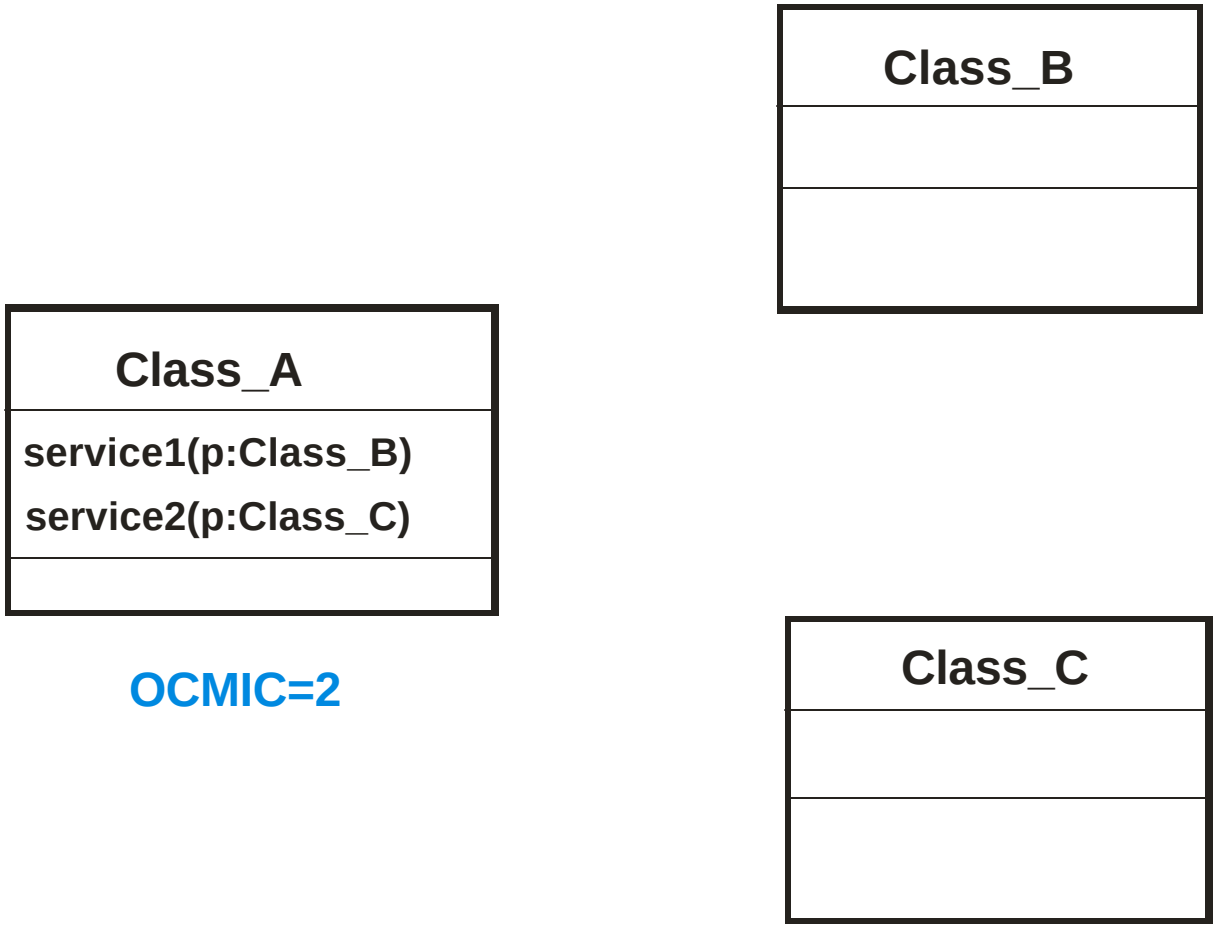
- Minimizing coupling among classes is an important design axiom:
  - makes the system much easier to understand so less faults are introduced and it is easier to detect faults (in inspections and testing)
  - reduces the ripple effect of making changes
- Coad and Yourdon emphasize:
  - reducing the complexity of message connections
  - reduce the number of messages that can be sent and received by any given class
- Coupling measures can be import or export, and can be among classes that do not have parent-child relationships or those that do
- Different aggregates for system-level coupling

© 2000-2001 k sharp technology inc. All rights reserved.

Example of a Coupling Measure: Class-Method Import Coupling

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

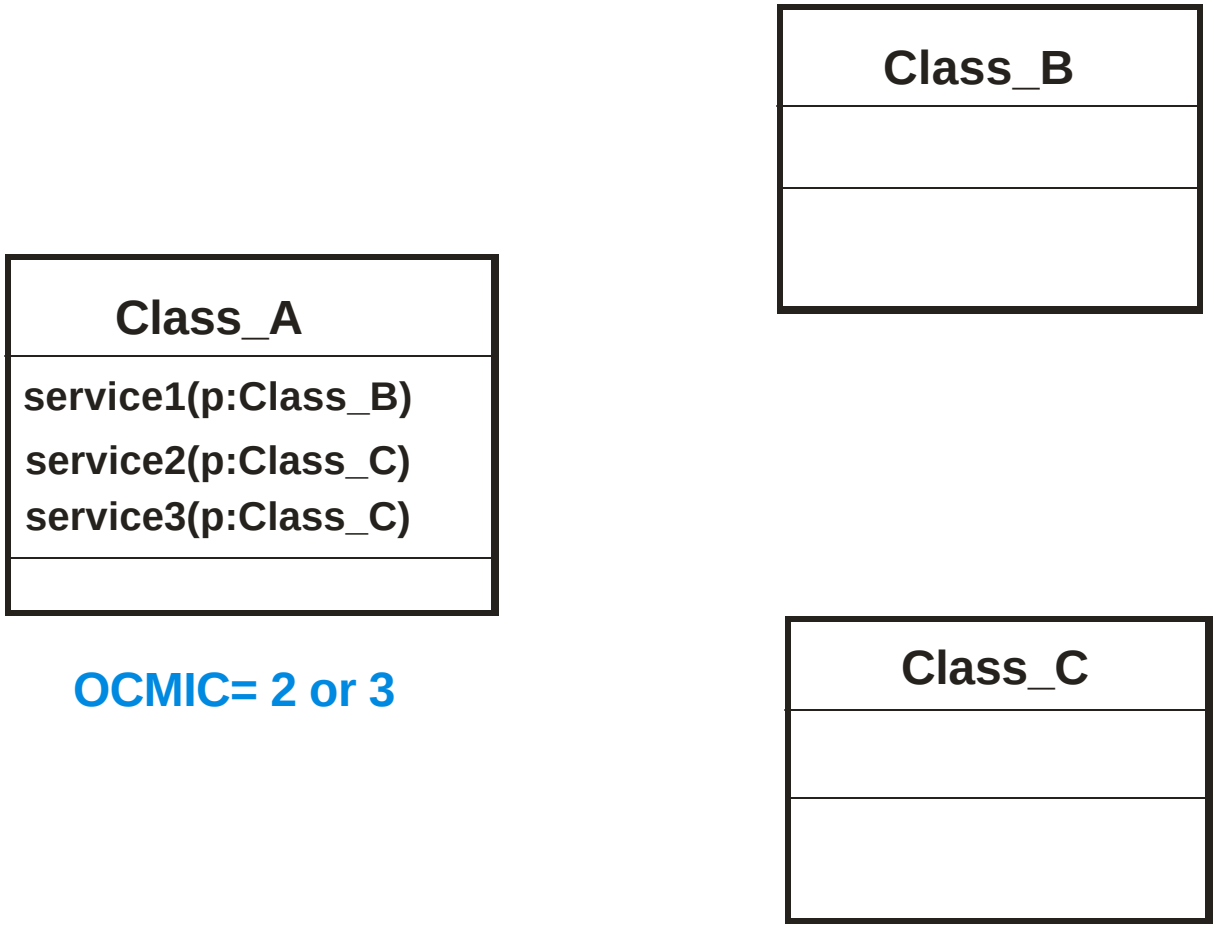


© 2000-2001 k sharp technology inc. All rights reserved.

Example of a Coupling Measure: Class-Method Import Coupling

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



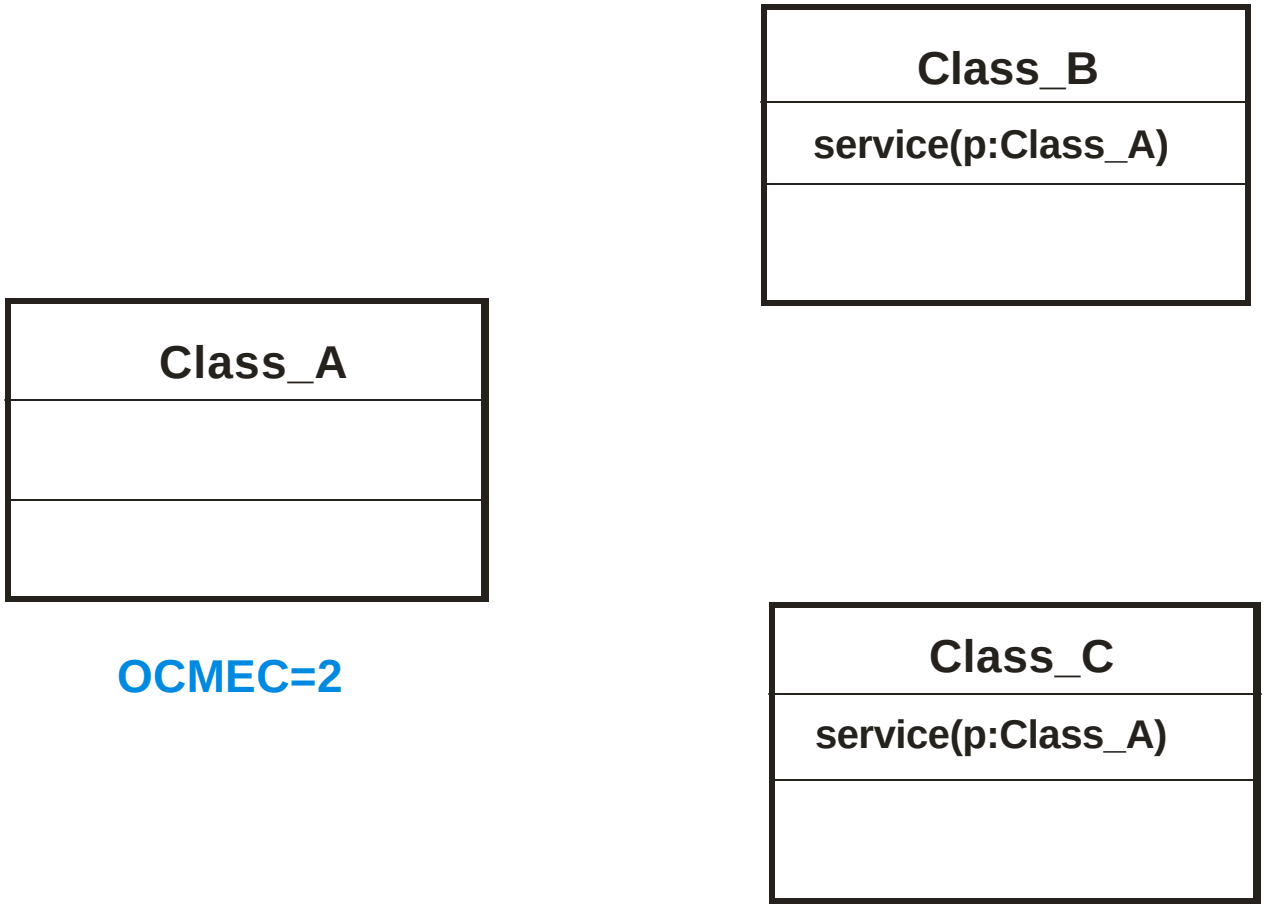
OCMIC= 2 or 3

© 2000-2001 k sharp technology inc. All rights reserved.

Example of a Coupling Measure: Class-Method Export Coupling

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

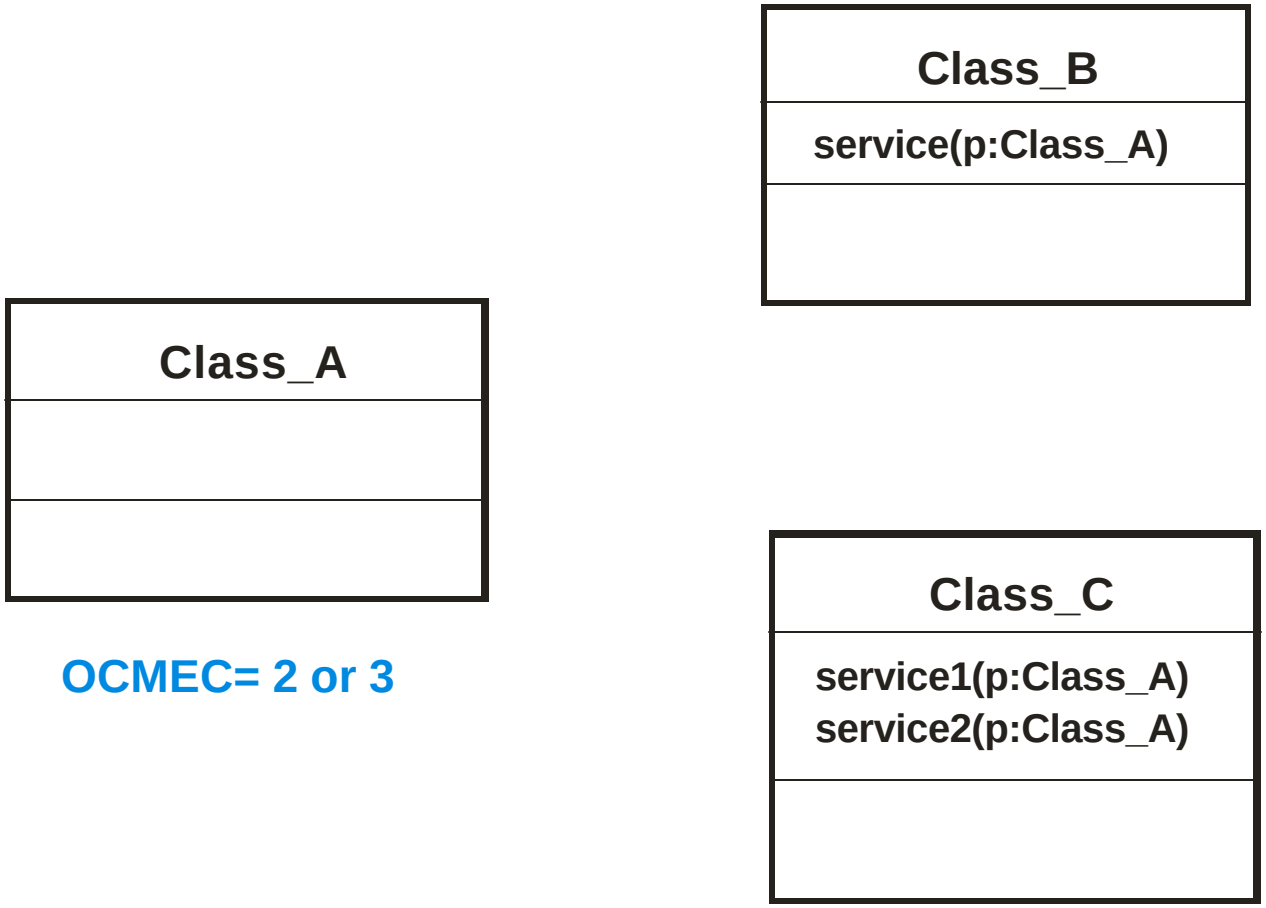


OCMEC=2

© 2000-2001 k sharp technology inc. All rights reserved.

Example of a Coupling Measure: Class-Method Export Coupling

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



OCMEC= 2 or 3

© 2000-2001 k sharp technology inc. All rights reserved.

Cohesion Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

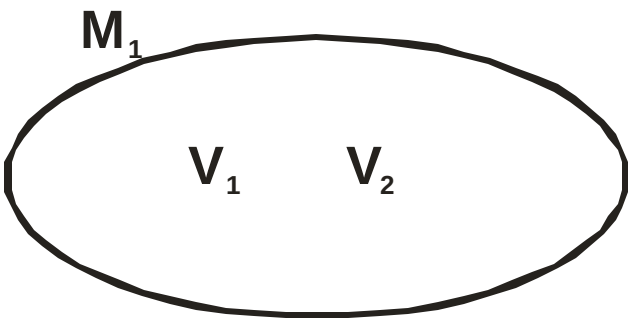
- Maximizing cohesion among classes is an important design axiom:
  - classes are easier to understand
  - reduces the ripple effect of making changes since they are likely to be localized
- Coad and Yourdon emphasize:
  - methods should carry out only one function
  - no unused attributes and methods
  - class functionality should be limited
- Different aggregations of class cohesion can be used for system-level cohesion

© 2000-2001 k sharp technology inc. All rights reserved.

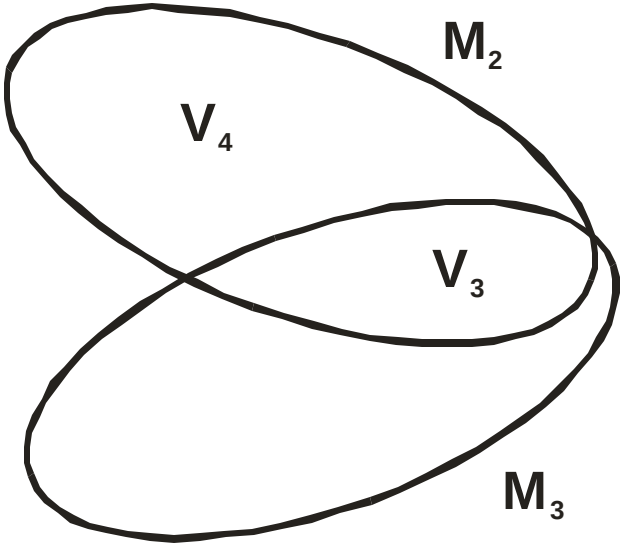
Example of a Cohesion Measure: Lack of Cohesion

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



LCOM=1



© 2000-2001 k sharp technology inc. All rights reserved.

Problem with Cohesion Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- For this type of cohesion measure, there are three important considerations:
  - access methods that provide access to a class' attributes (e.g., Set() and Get() methods) -> regular methods do not need to access the attributes at all
  - constructors that initialize all of the attributes in a class -> they will share attributes with almost all other methods in a class
  - how should one deal with accessing inherited attributes - are these counted as well, and if so, then a class and all its ancestors are a "semantic concept" rather than a class by itself ?

© 2000-2001 k sharp technology inc. All rights reserved.

Size Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Size is the most important object-oriented measure
- Different ways of computing class size:
  - LOC (this works well in practice)
  - number of methods (total)
  - number of attributes (total)
  - number of states/transitions
- System size can be computed by:
  - sum of class level measures
  - number of use cases per iteration (assuming iterative process)
- There are other measures, such as object points, but these are rarely used in practice

© 2000-2001 k sharp technology inc. All rights reserved.

Which is the Best Size Measure ?

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

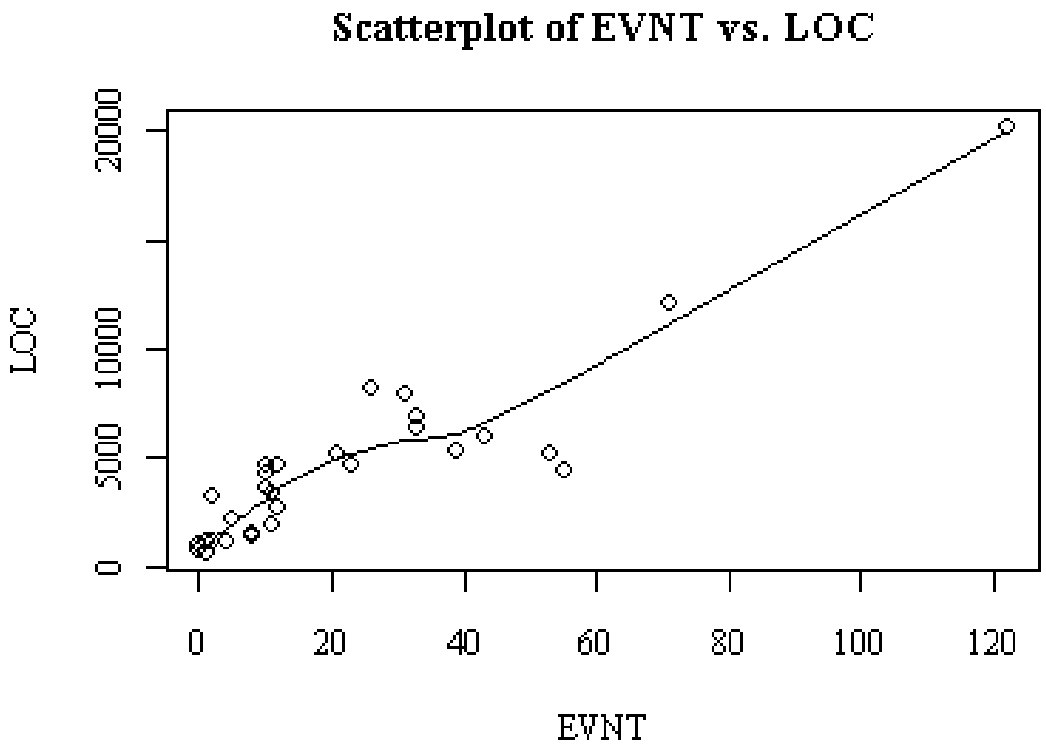
- Most size measures are strongly correlated with each other - so it does not really matter which one you choose
  - you can check that yourself by collecting different size measures on your system
- The criteria to use for selection:
  - which measure is your organization most comfortable with (you do not want to start a size measurement war)
  - which measure can you easily automate
  - select measures that are available at the stage in the project where you wish to make decisions

© 2000-2001 k sharp technology inc. All rights reserved.

Correlations Among Size Measures

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

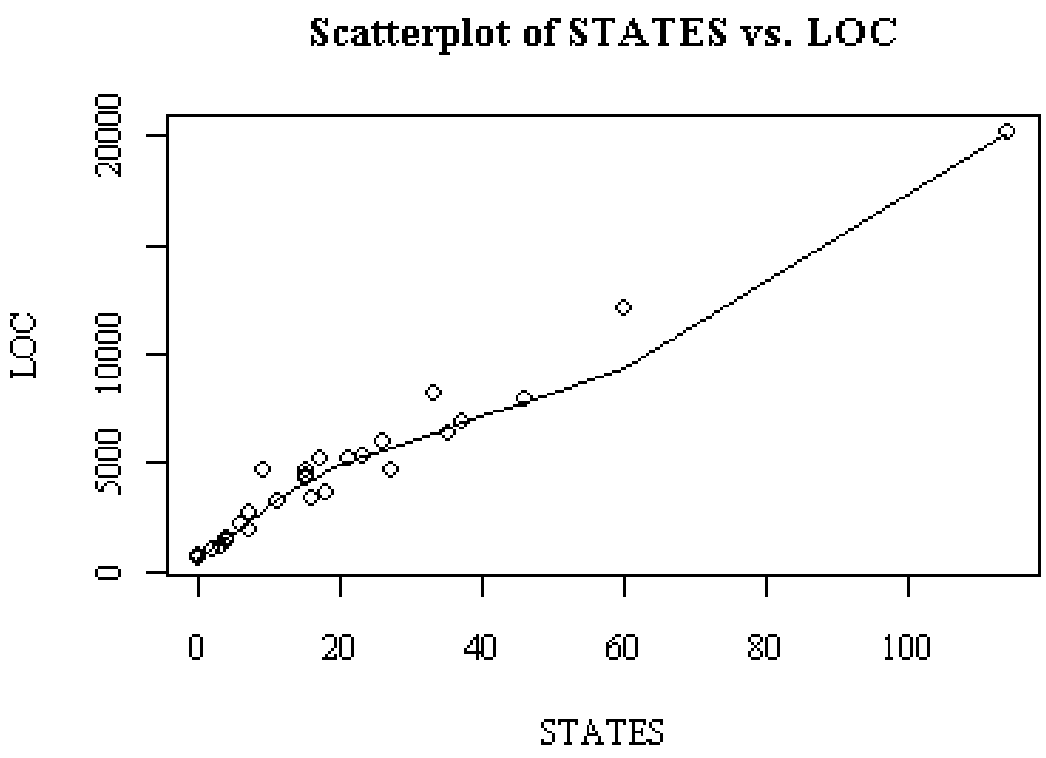


© 2000-2001 k sharp technology inc. All rights reserved.

Correlations Among Size Measures

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

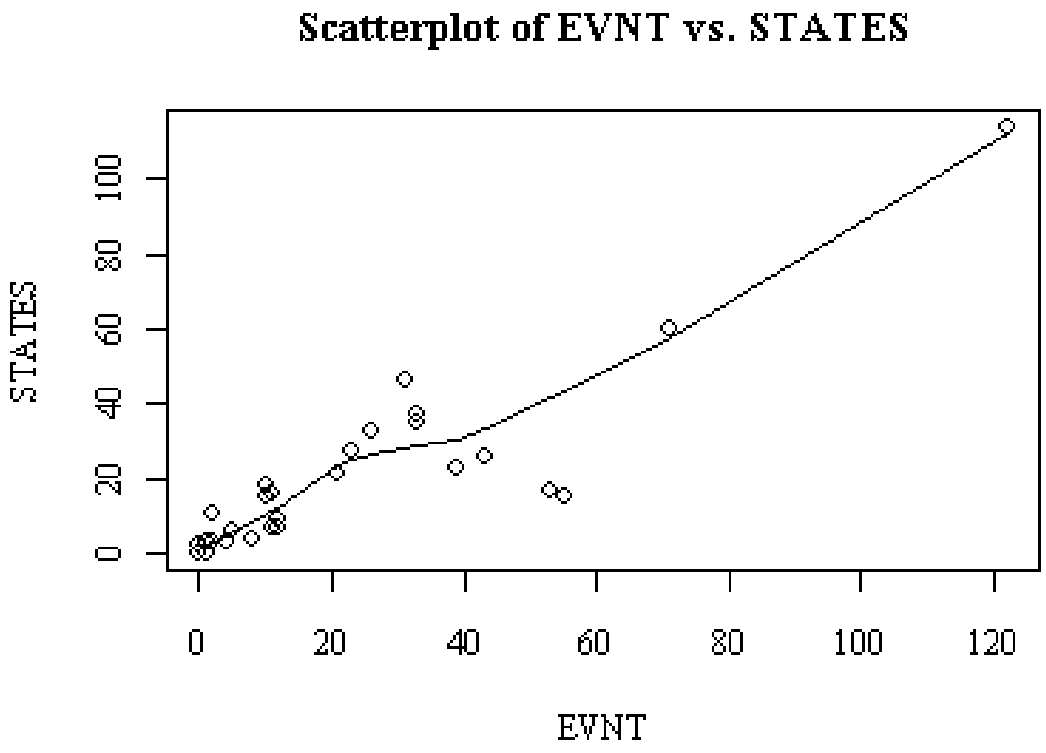


© 2000-2001 k sharp technology inc. All rights reserved.

Correlations Among Size Measures

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Size and Reuse - I

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- In most cases, reuse has to be taken into account when measuring size - if not then you can get very misleading numbers
- Reuse is an issue when:
  - external libraries are used
  - ad hoc reuse from previous projects
  - product line organization
- Usually, do not measure external libraries (e.g., MFC or ATL) except where:
  - there is import coupling from your application
  - there is inheritance from the libraries

© 2000-2001 k sharp technology inc. All rights reserved.

Size and Reuse - II

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Definitely an issue when there is ad hoc reuse or a product line organization
- Must define the unit of reuse, usually a class
- Three types of reuse levels:
  - reused verbatim
  - reused with modification
  - not reused
- Reused verbatim classes are not counted except:
  - for their impact on other classes, or
  - when the objective is to explicitly measure reuse
- Modified classes can be counted at a fraction of their size (say around 20% of their size)
- Not reused (i.e., new) do not require special treatment

© 2000-2001 k sharp technology inc. All rights reserved.

Size and Reuse - III

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- For example:
  - 10 KLOC reused verbatim
  - 40 KLOC reused with modification
  - 70 KLOC new code
- Total reuse-adjusted size of developed code would be:

$$Size = (40 \times 0.2) + 70 = 78\,KLOC$$

- Percentage of reuse:

$$Reuse = \frac{10 + (40 \times 0.8)}{(40 \times 0.2) + 70} \times 100 = 54\%$$

© 2000-2001 k sharp technology inc. All rights reserved.

Measurement Considerations

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Should we count instances of couplings to classes or the number of classes only ?
- Should we generally count external libraries as well or ignore these ?
- Should coupling measures be weighted by the “complexity” of the connection ?
- How to account for templates (especially coupling to templates) ?
- How to account for abstract classes ?
- What about exception classes - how do we measure their coupling ?
- Should indirect coupling be counted ?
- The stability of measures over time.

© 2000-2001 k sharp technology inc. All rights reserved.

What is Not Measured ? - I

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- OO measures do not capture everything that is important in your OO system
- Only looking at the interface, its “quality” is characterized by many things
- Naming consistency is important, especially for methods
- Signature consistency is important (e.g., two print() methods should have similar signatures with parameters in the same order)
- Following the rules of discourse in the interface is critical
- Quality of interface documentation (especially as a reference)

© 2000-2001 k sharp technology inc. All rights reserved.

What is Not Measured ? - II

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Examples of using classes is a tremendous help
- Architectural mismatch between interfacing components
- So there are many other things that will influence cost and defects
- Nevertheless, OO measures have proven, repeatedly, to be very valuable for helping make better decisions

© 2000-2001 k sharp technology inc. All rights reserved.

Other Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- There are other measures that have been defined and that may be useful
- Friendship measures are not considered because:
  - specific to C++
  - friendship is not used that often except for operator overloading
- Polymorphism measures are an interesting group of measures but they have not been used that extensively in practice, so judgement is reserved on their utility

© 2000-2001 k sharp technology inc. All rights reserved.

Most Useful OO Class Measures

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

Name	Description
Size	This can be the number of methods in a class or Lines of Code, depending on whether a design or source code are being evaluated.
DIT	Depth of Inheritance Tree indicates how deep in the inheritance hierarchy a class is.
OCMEC	Class-method export coupling to other classes indicates how many other non-descendant classes use a class type in their method signature.
OCMIC	Class-method import coupling to other classes indicates how many other class types are used in a class' method signatures.
ACMIC	Class-method import coupling to ancestor classes indicates how many of a class' method signatures use ancestor types.
DCMEC	Class-method export coupling to descendant classes indicates how many descendents use a class' type in their method signature.

© 2000-2001 k sharp technology inc. All rights reserved.

Collecting the OO Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- A measurement tool is required
- Most tools work on source code, but some are emerging that can collect the measures from design
- Some clients have taken advantage of the scripting capabilities in their design CASE tools to compute design measures
- We have a tool that is available freely under the GNU General Public License that collects these measures from C++ and Java programs
- You can download it from:

http://www.object-oriented.org

© 2000-2001 k sharp technology inc. All rights reserved.

Evaluating Measurement Tools

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

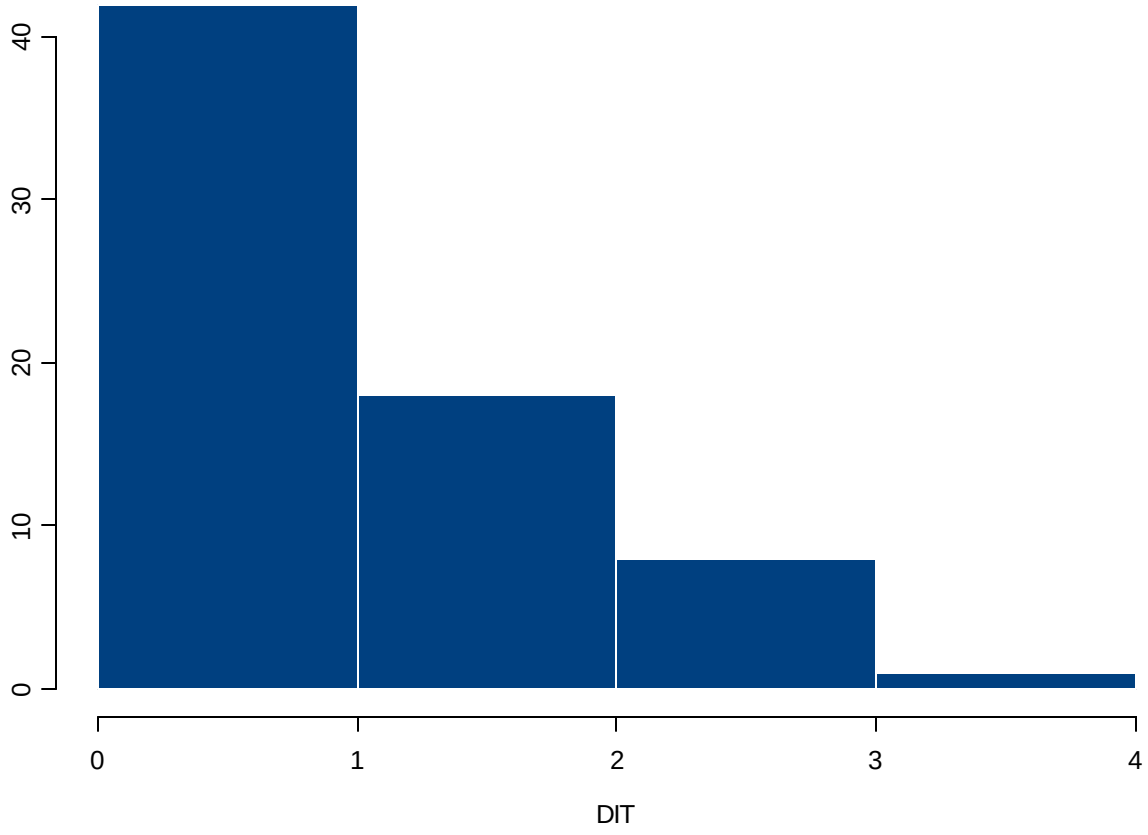
- The number of measures collected by a tool is not a good criterion because many OO measures are strongly correlated with each other
- Consider how easy it is to integrate the tool into your development environment
- Try the tool first on your particular system because:
  - some tools do not scale up - they cannot handle very large systems
  - there are many compiler and platform nuances that can consume huge amounts of effort to deal with in a measurement tool
- The tool should be able to export data easily for your particular reporting needs
- Ideally, the tool should be extensible so the user can define their own measures

© 2000-2001 k sharp technology inc. All rights reserved.

Distribution of OO Measures

Contents

Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

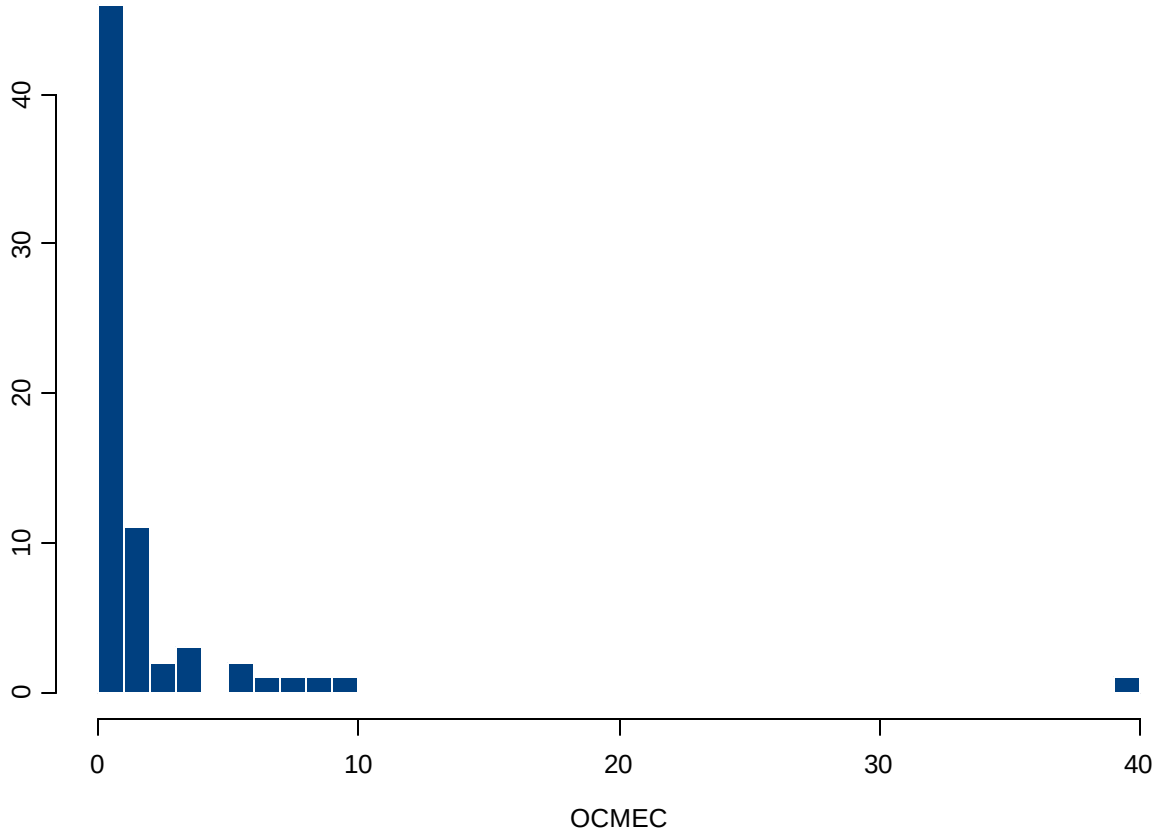


© 2000-2001 k sharp technology inc. All rights reserved.

Distribution of OO Measures

Contents

Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

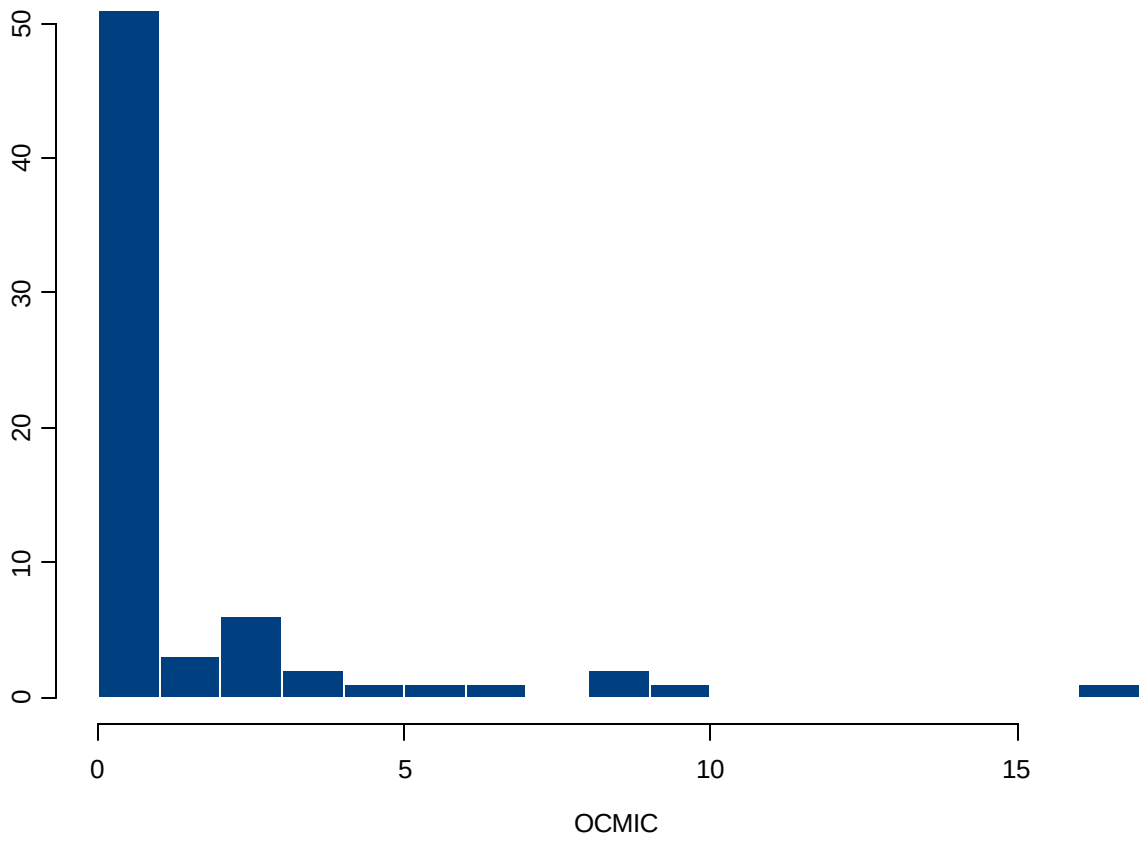


© 2000-2001 k sharp technology inc. All rights reserved.

Distribution of OO Measures

Contents

Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

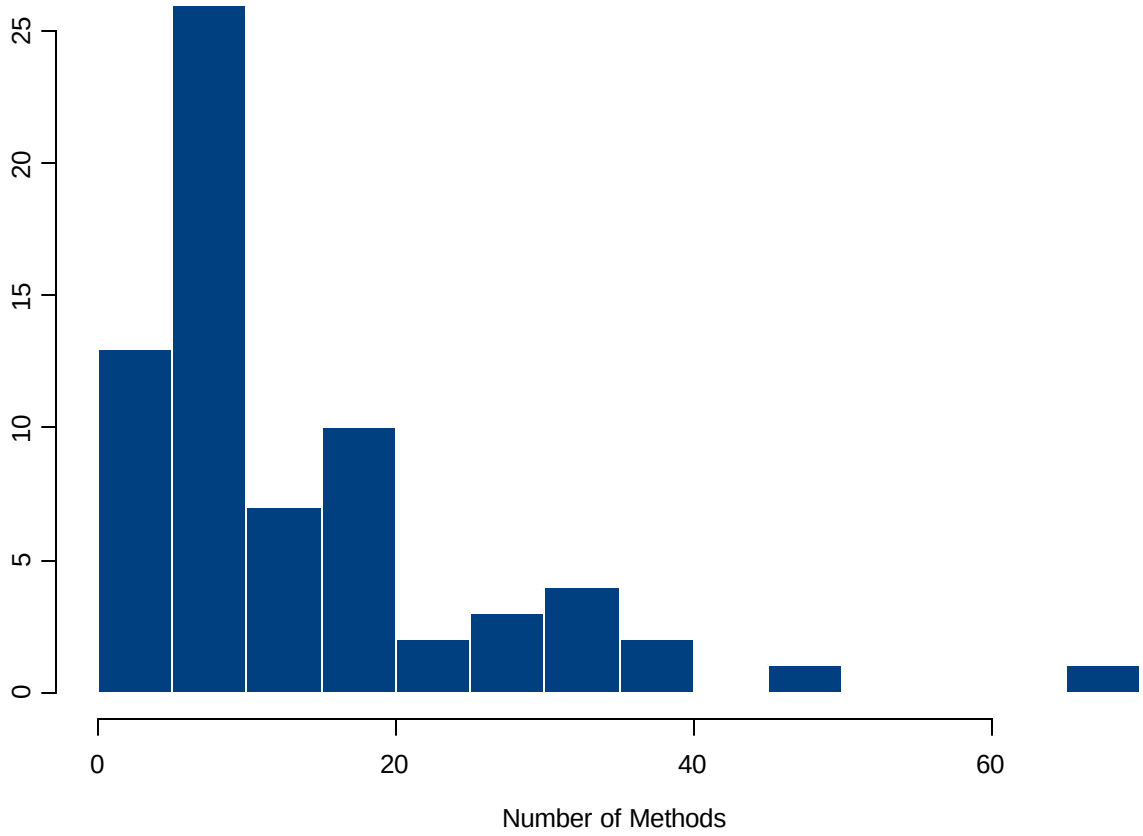


© 2000-2001 k sharp technology inc. All rights reserved.

Distribution of OO Measures

Contents

Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Overview of Cost Estimation

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Cost is typically evaluated as labor effort plus a fixed overhead - so we will focus on estimating the effort
- We are interested in overall project effort (or subsystem effort if that is a distinguishable project)
- Object-oriented projects are typically iterative therefore re-estimation throughout the project is necessary
- Two general estimation approaches:
  - estimate effort for each class separately and then aggregate to come up with an overall project estimate
  - estimate for the overall project to start off with

© 2000-2001 k sharp technology inc. All rights reserved.

Estimation Techniques

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Parametric models:
  - must be calibrated before they can be used
  - require historical data within the organization which takes time to collect (either to develop the parametric model or to calibrate it)
- Analogy estimates:
  - based mostly on experts
  - tends to work well in practice
  - can be automated with measurement on historical data
- Subjective estimates:
  - work well in practice
  - quick startup

© 2000-2001 k sharp technology inc. All rights reserved.

Estimation Practice

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Ideally, more than one technique should be used and if they do not concur then revisit the assumptions to see if anything has been missed
- Bad budget estimates have doomed many projects from the start - this is important and some serious consideration should be given to the estimates
- The solution is not to buy a tool that produces some kind of estimate - the estimation method must be understood by the project manager
- The focus will be on the subjective estimation approach because it also embodies some of the principles of analogy

© 2000-2001 k sharp technology inc. All rights reserved.

Subjective Estimation

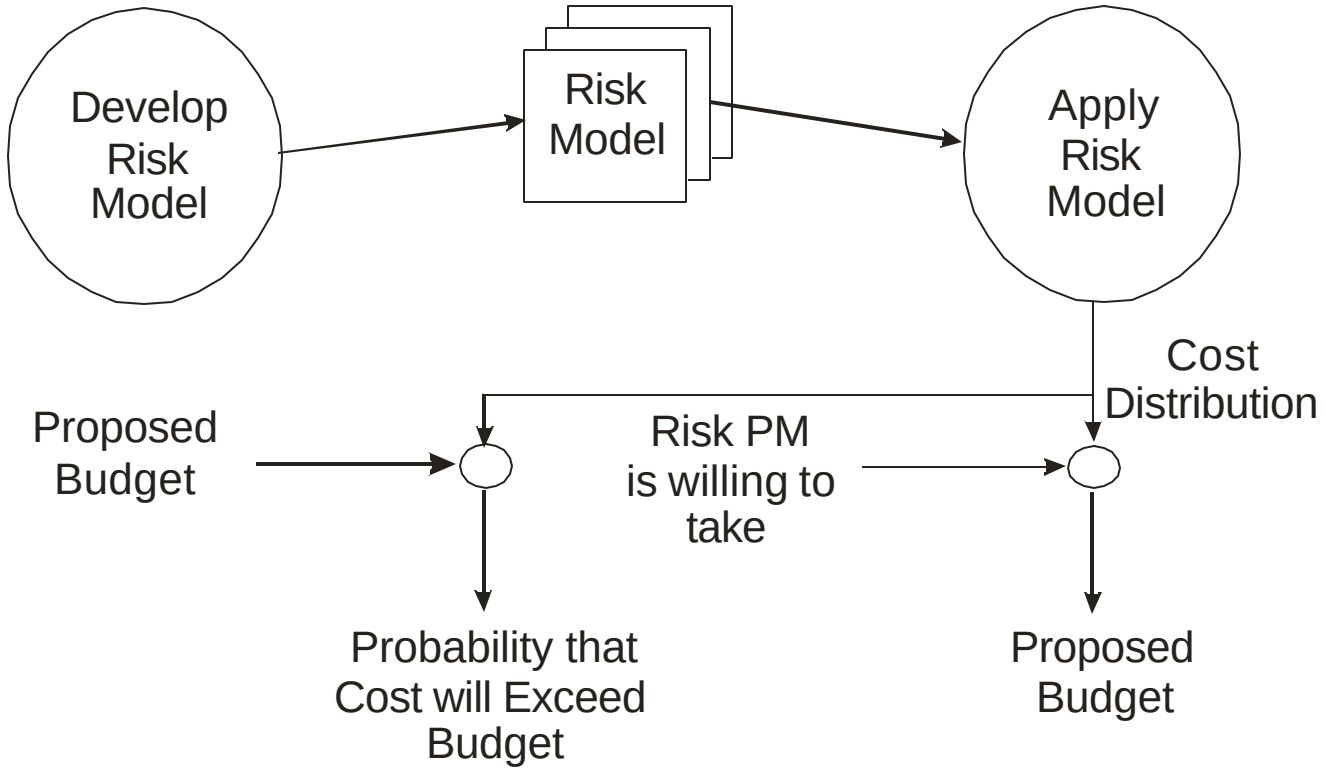
Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Can be used in two modes:
  - a budget has already been assigned (e.g., in a bidding situation) and the objective is to determine the probability that the cost of the project will exceed the budget
  - the objective is to come up with a budget that the project has a high probability of meeting
- It is a risk assessment approach because there is always uncertainty in estimates (i.e., predicting the future)
- Requires experts in OO technology and in the application domain
- From an estimation perspective, has been shown to have estimation errors as low as 9%

© 2000-2001 k sharp technology inc. All rights reserved.

Overall Process

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Principle of Risk Model

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- The cost estimate for a project is a distribution rather than a fixed value
- The basic model is:

$$Total\,Cost = NominalCost + RiskOverhead$$

- Nominal Cost is based largely on size and the organization's experience with this kind of system
- The Risk Overhead captures additional costs due to a number of *productivity factors*

© 2000-2001 k sharp technology inc. All rights reserved.

The Nominal Cost

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- The nominal cost is the effort under ideal conditions (e.g., where you have the right people and requirements do not change)
- The process has the following steps:
  - Produce generic productivity values for the organization for different types of OO classes
  - Estimate the size of the system in terms of the classes and the size of the classes
  - Compute the nominal cost
- The overall approach is quite simple but works well

© 2000-2001 k sharp technology inc. All rights reserved.

Productivity Table

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

	Key classes	Support classes
Low Complexity	8 LOC/ph	4 LOC/ph
Medium Complexity	6 LOC/ph	3 LOC/ph
High Complexity	3 LOC/ph	2 LOC/ph

$$NominalCost_i = \frac{Size}{Productivity}$$

© 2000-2001 k sharp technology inc. All rights reserved.

Economies of Scale

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- The relationship between size and effort is not linear; there can be economies or diseconomies of scale
- This means that productivity can increase as the overall project size increases or it can decrease
- Factors that influence this are, for example:
  - documentation costs
  - communication overhead
  - CASE tool costs
  - extent of interfaces
- Therefore, the productivity values should be based on projects of roughly a similar size

© 2000-2001 k sharp technology inc. All rights reserved.

Risk Overhead

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- To produce the Risk Overhead it is necessary to develop the Risk Model first. This involves:
  - identify relevant productivity factors
  - develop checklist of selected productivity factors
  - assign effort values for each of the productivity factors
- Now, assign values to your particular project on the checklist and compute the overhead costs
- Add this to the nominal cost

© 2000-2001 k sharp technology inc. All rights reserved.

Productivity Factors - I

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- There are five types of productivity factors that can be considered:
  - personnel
  - process
  - product
  - project
  - technology
- There are many of these, so only a subset should be used; this would be the subset most relevant to the organization
- A ranking exercise among a group of cost estimation experts can narrow down the productivity factors

© 2000-2001 k sharp technology inc. All rights reserved.

Productivity Factors - II

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

The requirements are clearly understood by the customer at the start of the project *(please select one)*.

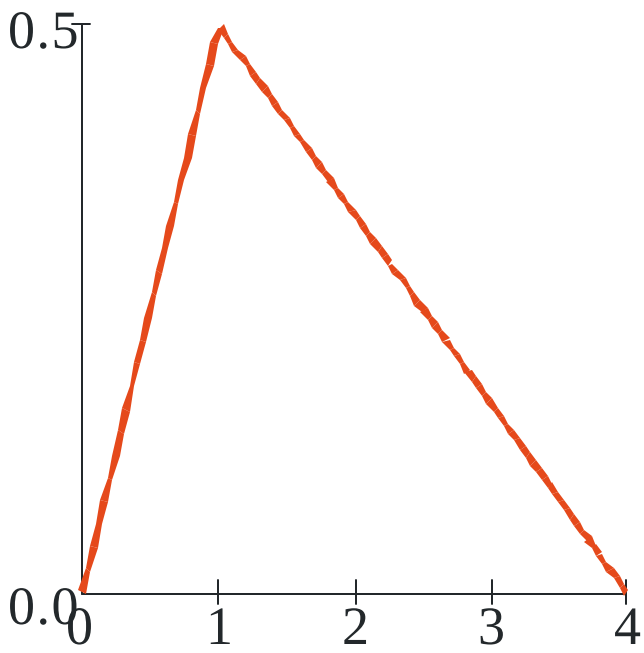
Nominal	❑	**Strongly Agree**
	❑	**Agree**
	❑	**Disagree**
Extreme	❑	**Strongly Disagree**

© 2000-2001 k sharp technology inc. All rights reserved.

Distribution of Effort per Productivity Factor

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

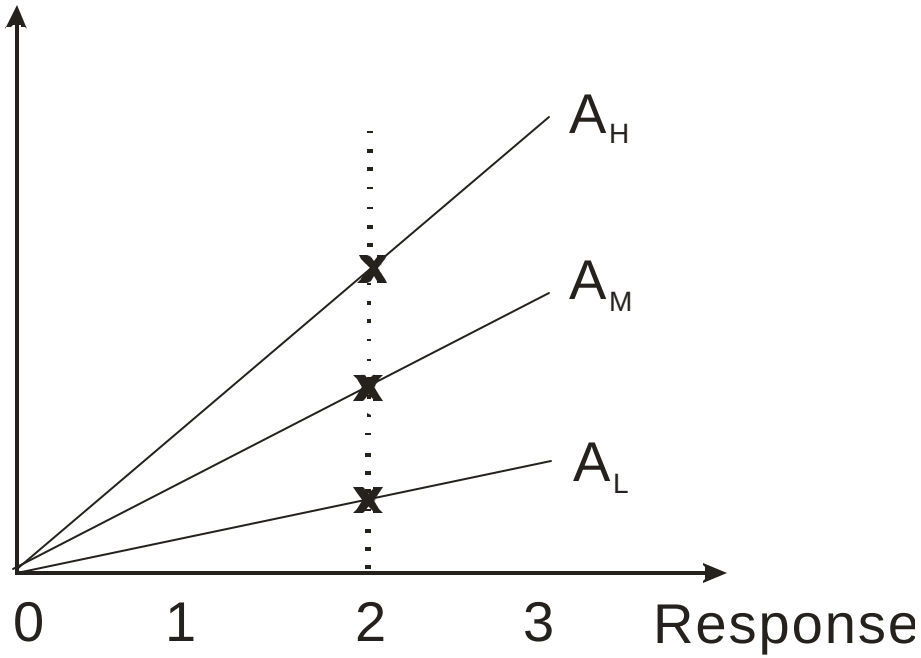


© 2000-2001 k sharp technology inc. All rights reserved.

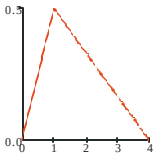
Computing Overhead for a Project

Contents

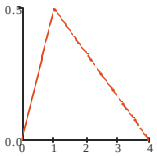
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



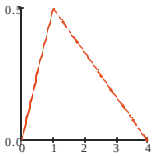
$$Risk\ Overhead = \sum$$



$+$



$+$

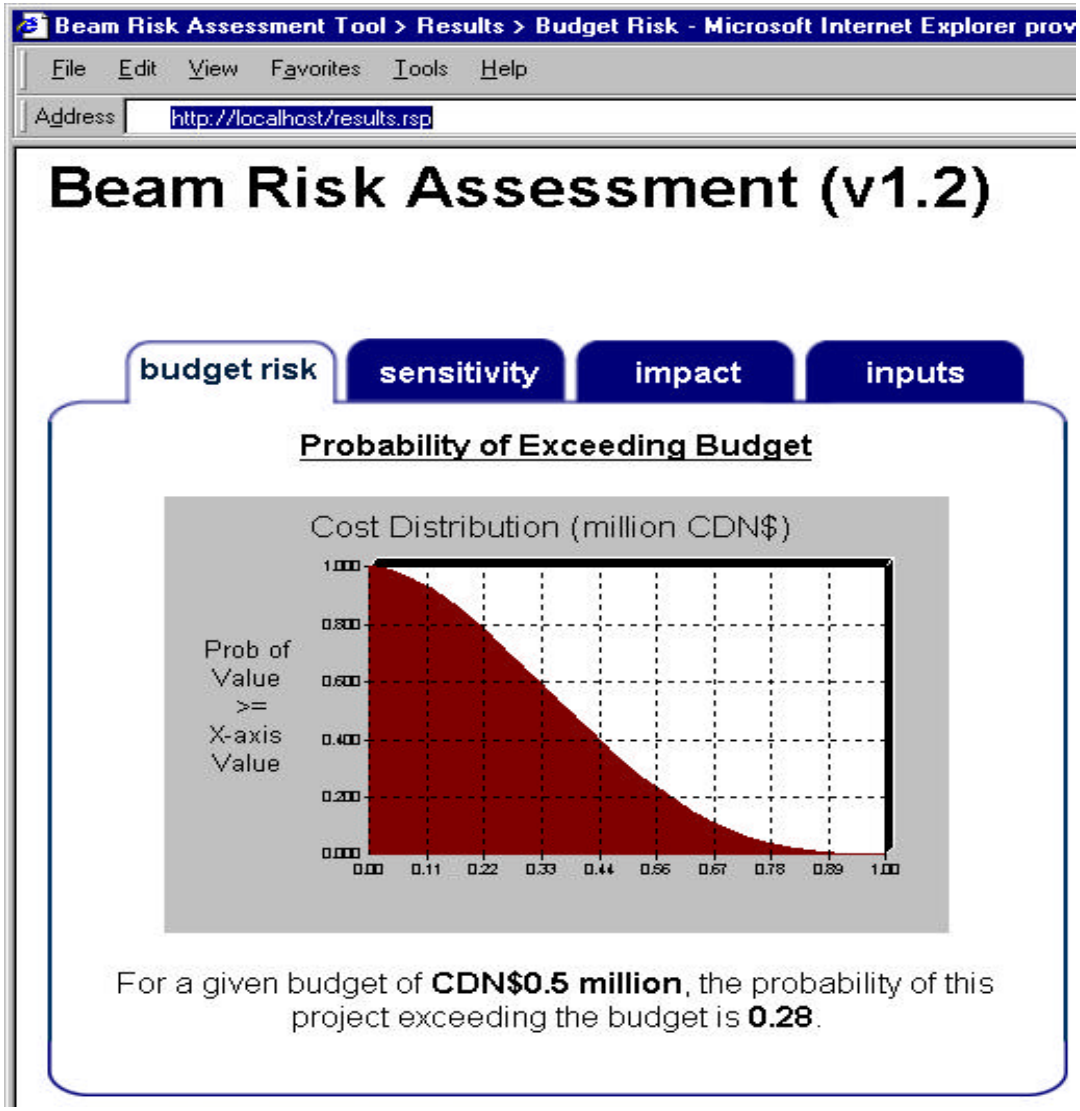


© 2000-2001 k sharp technology inc. All rights reserved.

Cost Distribution

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

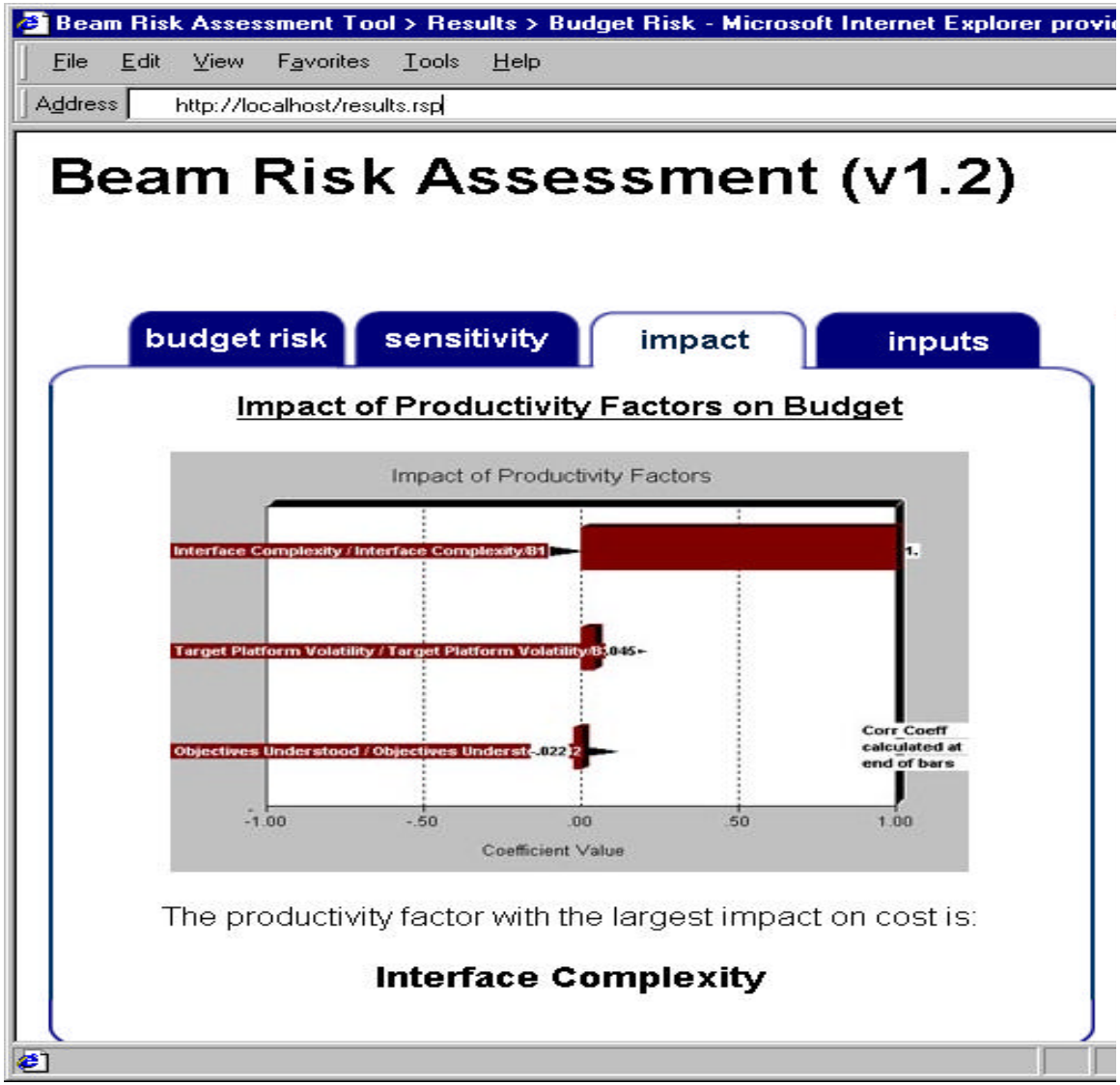


© 2000-2001 k sharp technology inc. All rights reserved.

Sensitivity Analysis

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Quality Management

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- In the context of quality management, object-oriented measures can potentially be used for:
  - Taking preventative action
  - Quality (and cost) estimation
  - Developing design guidelines
- You can get different results depending on which criterion you are talking about:
  - pre-release faults
  - post-release faults
  - development & maintenance effort

© 2000-2001 k sharp technology inc. All rights reserved.

Fault Measurement

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Although it seems straight forward, measuring the number of faults (pre- or post- release) can be quite inaccurate
- Usually the version control system is used to determine which classes have been modified to fix a fault
- However, multiple faults may have been fixed in the same delta to avoid paperwork, multiple cycles of regression testing, and multiple approvals
- This results in an undercount of the number of faults
- One can use an indicator of whether at least one fault has occurred instead of a count as the outcome variable

© 2000-2001 k sharp technology inc. All rights reserved.

Preventative Action - I

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- There exists evidence showing that most faults in a software system are detected in a small percentage of its components
- To improve defect detection effectiveness and efficiency it is desirable to identify these components early on
- Then one can take mitigating actions, such as more inspections, targeted testing, assignment of staff, or even a redesign

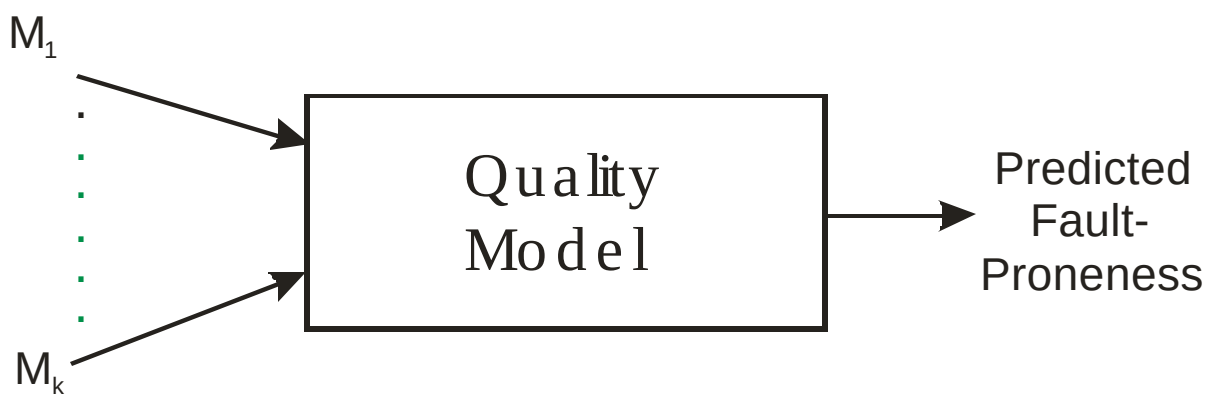
© 2000-2001 k sharp technology inc. All rights reserved.

Preventative Action - II

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

Quality Models

- Quantitative models are constructed to predict which components are likely to have a fault



© 2000-2001 k sharp technology inc. All rights reserved.

Preventative Action - III

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- The classes are ranked by their predicted fault-proneness (which is the probability of a fault)
- Then, the top x% are selected as high risk classes
- There are a number of criteria for selecting the value of x:
  - the project can only afford to take action on x% - for example, there are sufficient resources to perform a limited number of inspections
  - historically, x% of the classes had post-release faults (say for the previous release of the same product)

© 2000-2001 k sharp technology inc. All rights reserved.

Preventative Action - IV

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

class id	predicted probability of a Fault
101	0.95
102	0.92
103	0.904
104	0.88
105	0.87
106	0.83
107	0.83
108	0.78
109	0.77
110	0.74
111	0.74
112	0.69
...	...

highest risk
x% of classes

© 2000-2001 k sharp technology inc. All rights reserved.

Preventative Action - Example (1)

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

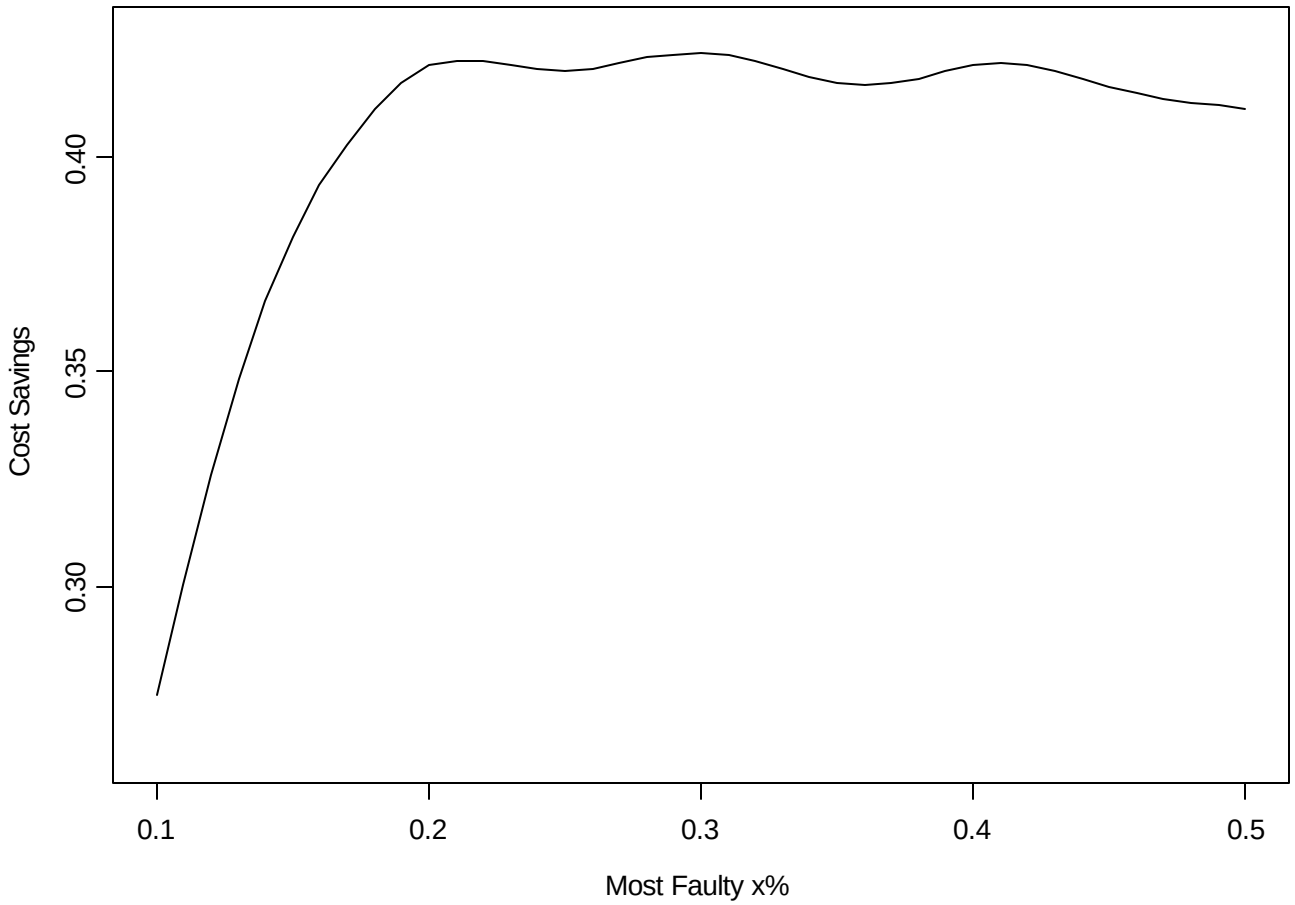
- A commercial Java application
- A quality model was constructed to predict the probability of a post-release fault
- Quality model was used to identify the classes that should be inspected
- Evaluated the cost savings from using the quality model (we had data on the cost of inspections and dealing with post-release faults)

© 2000-2001 k sharp technology inc. All rights reserved.

Preventative Action - Example (2)

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Quality Prediction

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Developed a quality model on one release of a commercial Java application
- Used that model to predict the proportion of classes in the next release that will have a fault in them
- The prediction has approx. 9-10% error (i.e., underestimated the proportion by 9-10%)
- This can be considered a good prediction accuracy

© 2000-2001 k sharp technology inc. All rights reserved.

Design Guidelines - I

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

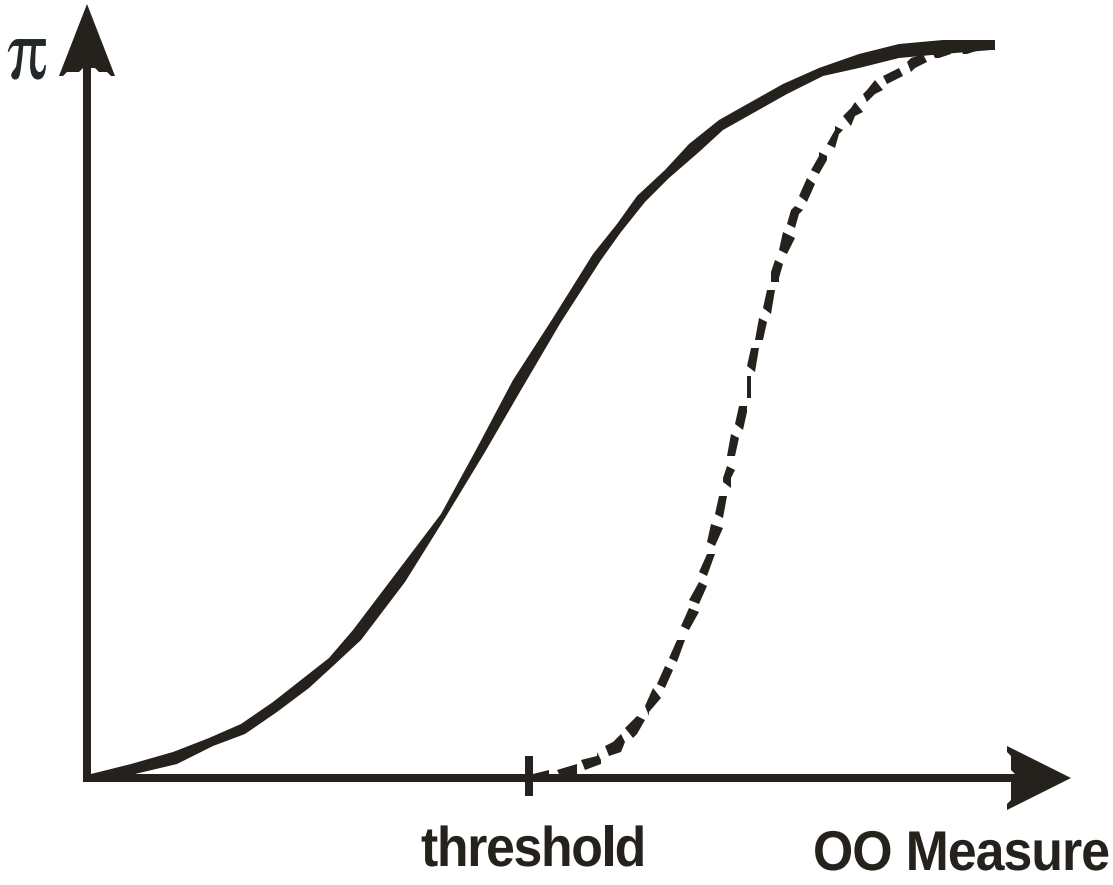
- It is also possible to empirically identify which structural properties of the software cause problems (e.g., which types of coupling)
- This information can then be used to construct proscriptive design guidelines to minimize problems in future development efforts
- Guidelines can be enforced automatically or through inspections
- Design guidelines expressed in the form of thresholds on the measures
- If rule is not satisfied then there is a greater risk of a fault

© 2000-2001 k sharp technology inc. All rights reserved.

Design Guidelines - II

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Design Guidelines - III

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

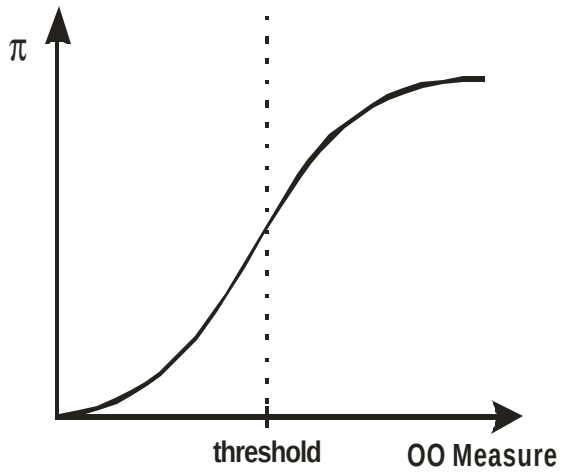
- Thus far, no evidence that thresholds exist
- We could not identify a specific value on any of the object-oriented measures where the probability of a fault suddenly increases
- Does this mean that thresholds are useless ?

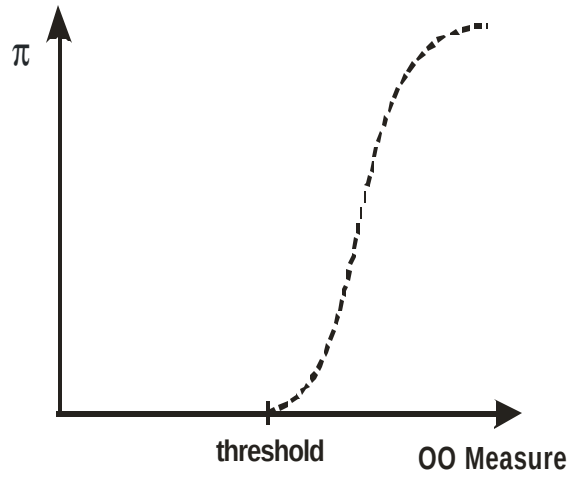
© 2000-2001 k sharp technology inc. All rights reserved.

Design Guidelines - IV

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing





© 2000-2001 k sharp technology inc. All rights reserved.

Methodological Considerations

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

- Confounding effects
- Optimal class size
- ROI calculations

© 2000-2001 k sharp technology inc. All rights reserved.

Common Quality Modeling Method

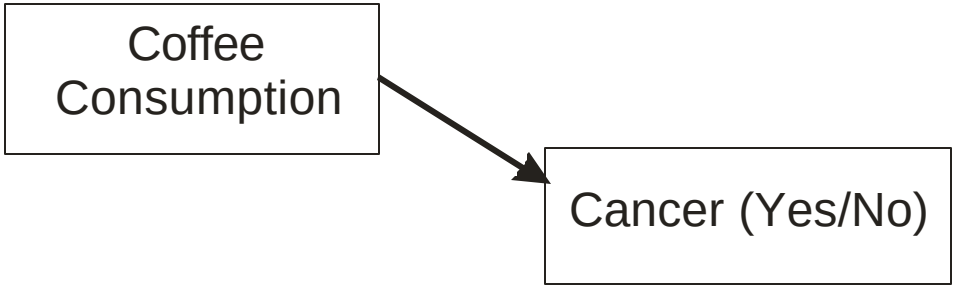
Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- You collect data on the object-oriented measure and on whether a fault occurs in a class
- If there is a positive relationship between the measure and fault-proneness, then the measure is said to be a good predictor
- A considerable number of studies do this and “successfully” demonstrate that a quality model with a given measure has good predictive power

© 2000-2001 k sharp technology inc. All rights reserved.

An Epidemiological Example

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



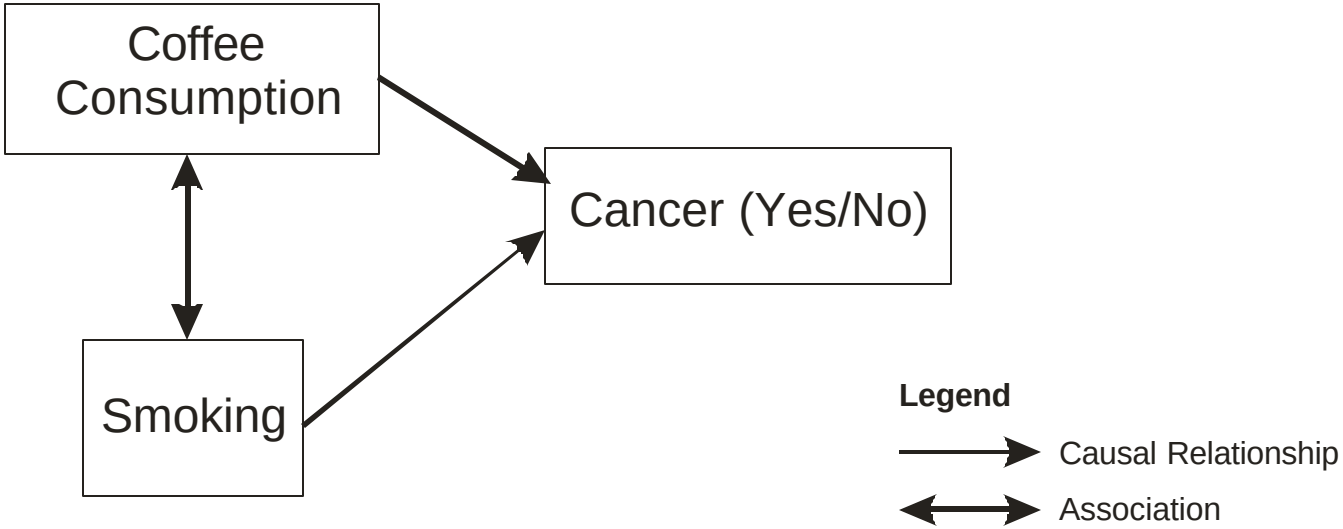
Legend

→ Causal Relationship

© 2000-2001 k sharp technology inc. All rights reserved.

An Epidemiological Example

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing



- If you look only at the coffee consumption → cancer relationship, you can get very misleading results
- Smoking is a confounder

© 2000-2001 k sharp technology inc. All rights reserved.

Confounding Effects

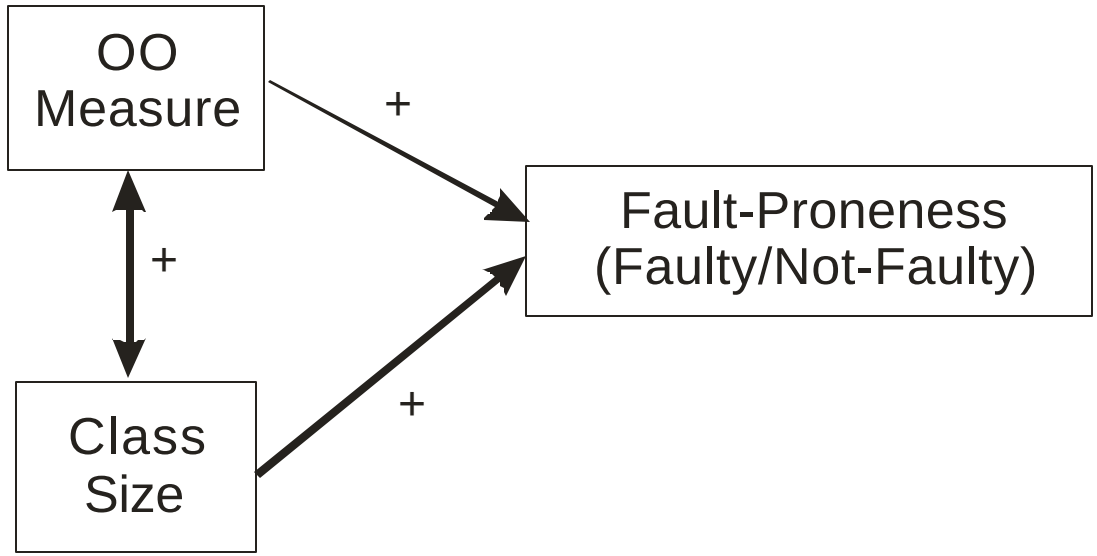
Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- It is known that smoking and coffee consumption are associated
- It is known that smoking causes cancer
- If smoking is not controlled during the study, then it is likely that an inflated coffee effect will be found
- This is a classic confounding effect
- Without proper controls, the study results can show a relationship where none really exists

© 2000-2001 k sharp technology inc. All rights reserved.

Confounding with Object-Oriented Measures

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing



- In this case, size is the confounder
- There is substantial evidence that most object-oriented measures are associated with class size

© 2000-2001 k sharp technology inc. All rights reserved.

Example Relationship with Size

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

OO Metric	*LOC*	
	Rho	**p-value**
WMC	0.88	<0.0001
DIT	0.098	0.19
CBO	0.46	<0.0001
RFC	0.88	<0.0001
LCOM	0.24	0.0011
NMA	0.86	<0.0001
NPAVG	0.27	0.000256

© 2000-2001 k sharp technology inc. All rights reserved.

The Confounding Effect of Size

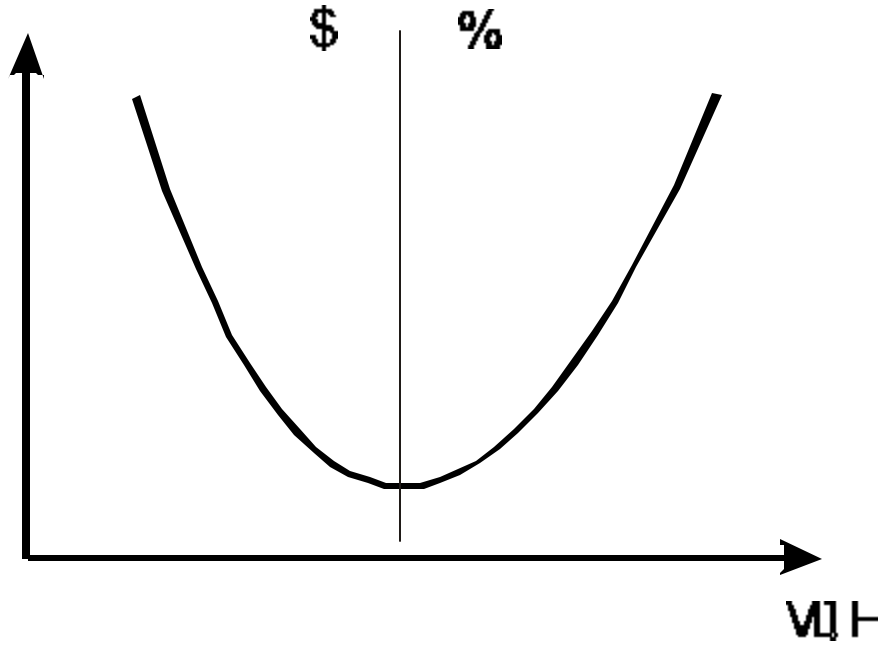
Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Just looking at the relationship between an object-oriented measure and faults does not make sense
- It is necessary to at least control for size

© 2000-2001 k sharp technology inc. All rights reserved.

Optimal Class Size & Faults - I

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing



- Smaller classes (or modules) are more faulty than larger ones (program decomposition is bad)
- There is an optimal class size

© 2000-2001 k sharp technology inc. All rights reserved.

Optimal Class Size & Faults - II

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

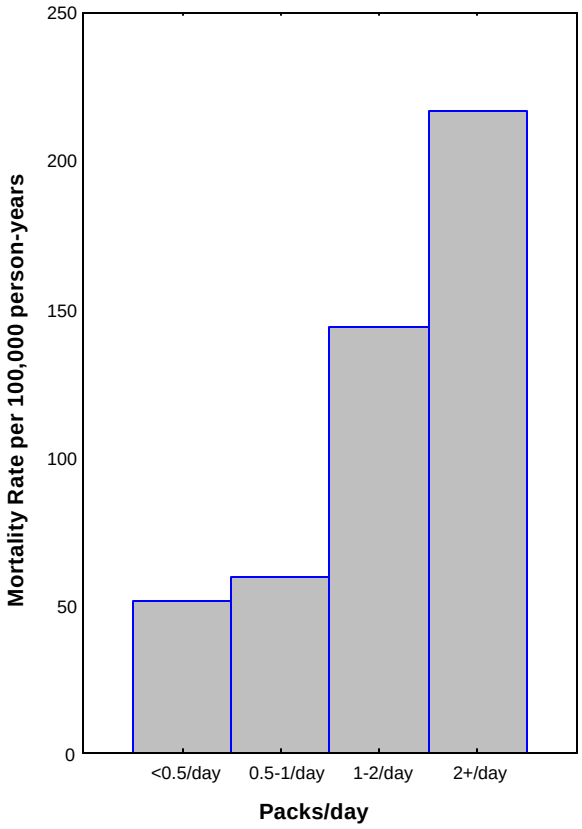
- The first conclusion is due to a mathematical artifact of plotting Y/X vs. X
- You will always get a negative association, even if there is no association between the raw variables
- It is easy to obtain absurd conclusions (turn a positive association to a negative one) this way
- Some examples ...

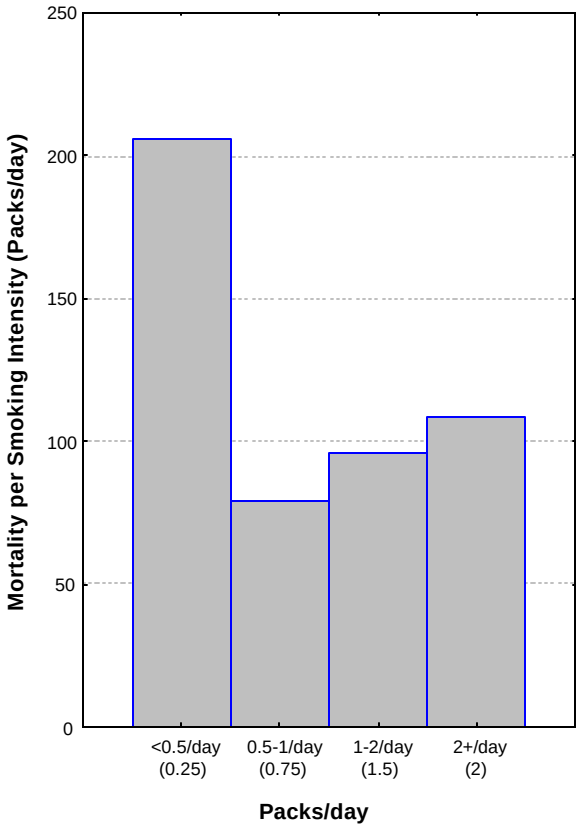
© 2000-2001 k sharp technology inc. All rights reserved.

Smoking and Mortality Rate

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing





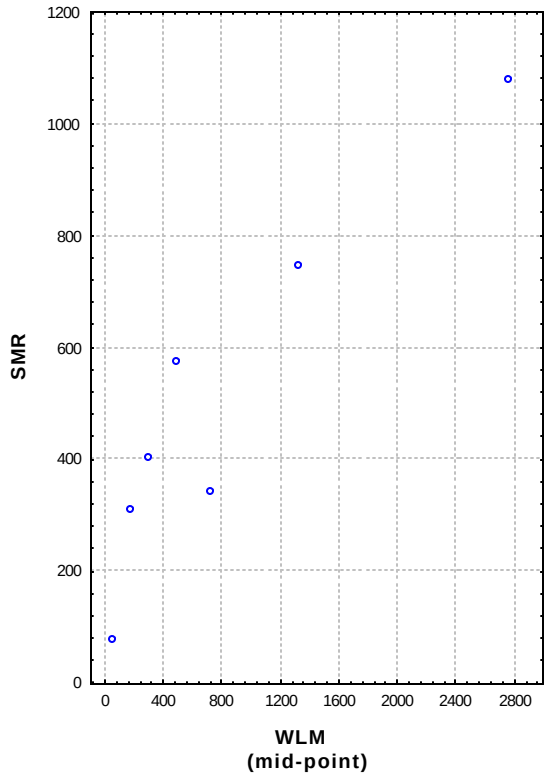
Optimal smoking: 0.75 packs/day

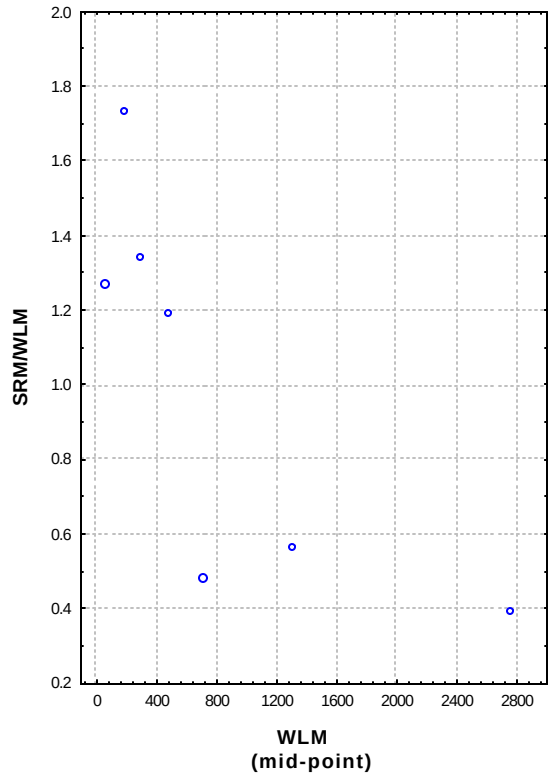
© 2000-2001 k sharp technology inc. All rights reserved.

Uranium Exposure & Mortality

Contents

- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing





Quickly, move closer to a uranium mine

© 2000-2001 k sharp technology inc. All rights reserved.

Calculating Return on Investment (ROI) - I

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

Principles

- Defect detection and correction in later phases of a project are much more expensive (some quoted numbers are 100:1 or even 200:1 for post-release defects)
- Effort spent now to detect and correct defects will lead to savings later on (so you invest now to save money later on)
- We can estimate how much is saved by finding and correcting defects now
- ROI uses that estimate as well as the cost of early defect detection and correction

© 2000-2001 k sharp technology inc. All rights reserved.

Calculating Return on Investment (ROI) - II

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

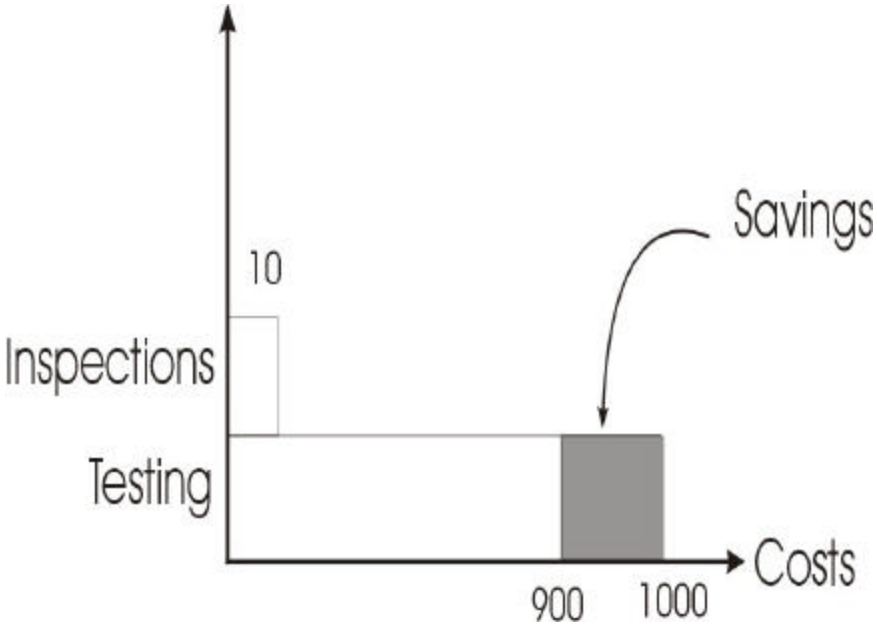
Basic Calculation

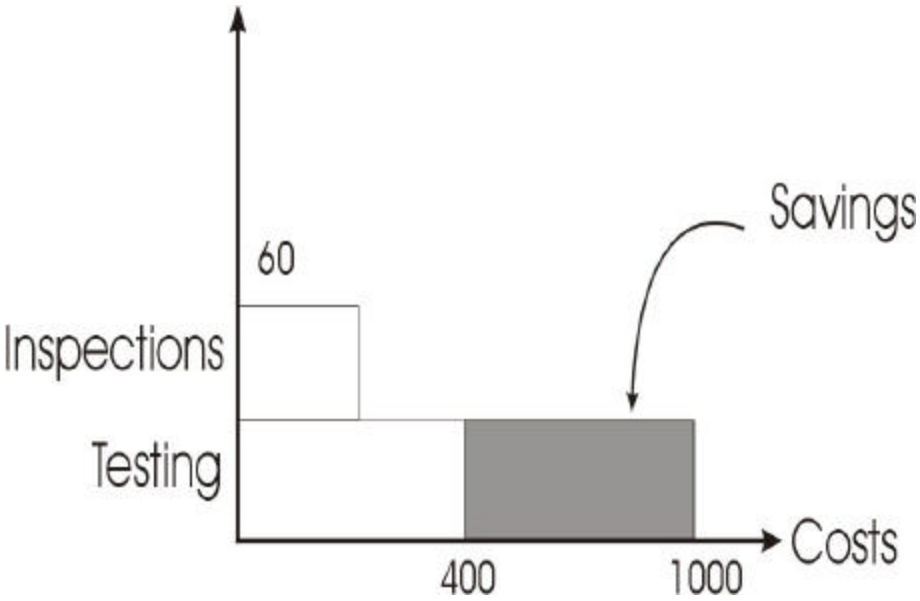
$$ROI = \frac{\text{cost saved}}{\text{cost consumed}}$$

© 2000-2001 k sharp technology inc. All rights reserved.

Calculating Return on Investment (ROI) - III

Problems with Basic ROI





Calculating Return on Investment (ROI) - IV

Contents
- Introduction
- Definition
- Cost Est.
- Quality Manag.
- Confounding
- Optimal Size
- ROI
- Closing

Improved Calculation

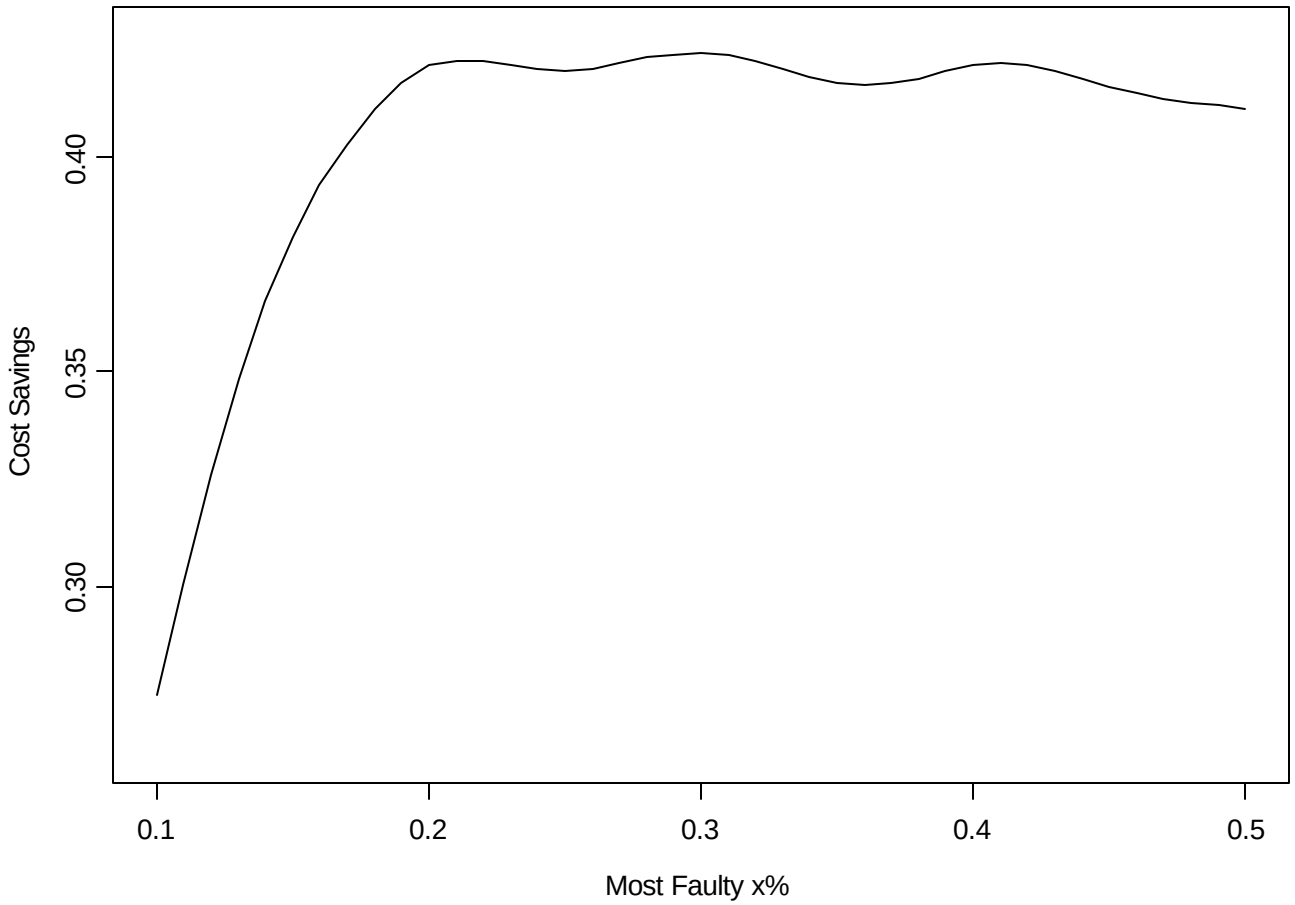
$$ROI = \frac{\text{cost saved}}{\text{virtual testing cost}}$$

© 2000-2001 k sharp technology inc. All rights reserved.

Calculating Return on Investment (ROI) - V

Contents

Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing



© 2000-2001 k sharp technology inc. All rights reserved.

Closing

Contents
Introduction
Definition
Cost Est.
Quality Manag.
Confounding
Optimal Size
ROI
Closing

- Object-oriented measures can play a very important role in managing projects:
  - cost estimation
  - quality management
- There are good tools available that make this process much easier to start
- There is considerable knowledge now about this - so it can be, and has been, applied on actual large projects (e.g., avionics, telecom, and MIS)

© 2000-2001 k sharp technology inc. All rights reserved.