

**CS 350: Introduction to Software Engineering**

---

Slide Set 2  
Process Measurement  
C. M. Overstreet  
Old Dominion University  
Spring 2006

---

---

---

---

---

---

---

---

**Reading assignment**

---

- Chapters 3 & 4, PSP text

Fall 2005 CS 350/ODU 2

---

---

---

---

---

---

---

---

**Lecture Topics**

---

- Process measurement
- Planning overview
- Software size
  - why measure size
  - size measurement criteria
  - the SEI size measurement framework
- Counting program size
  - size counters
  - coding standards

---

---

---

---

---

---

---

---

## Process Measurement Principles

---

- To be useful, measurements should be
  - gathered for a specific purpose
  - explicitly defined
  - properly managed
  - properly used
- Measuring your process will not improve it. You must make process changes to achieve lasting improvement.

---

---

---

---

---

---

---

---

## Process Measurement Purposes

---

- We measure to:
  - understand and manage change
  - predict or plan for the future
  - compare one product, process, or organization with another
  - determine adherence to standards
  - provide a basis for control

---

---

---

---

---

---

---

---

## Measurement Objectives

---

- Measurements only produce numbers.
- To be useful, they must:
  - relate to business objectives
  - be properly interpreted
  - lead to appropriate action
- If the business purposes for the measurements are not understood
  - the wrong data may be gathered
  - data may not be properly used

---

---

---

---

---

---

---

---

## Types of Measurements

- We generally need objective and explicit measures.
- To be useful, we need relationships that correlate.
  - program size versus development hours
  - cost distributions
  - defect densities
- We also seek a controlling or predictive capability.
  - actions to reduce test defects
  - steps to improve review quality
  - means to improve productivity

---

---

---

---

---

---

---

---

## The PSP Measurements

- The basic PSP data are
  - program size
  - time spent by phase
  - defects found and injected by phase
- Both actual and estimated data are gathered on every item.
- Measures derived from these data
  - support planning
  - characterize process quality

---

---

---

---

---

---

---

---

## PSP Size Measures

- The goals of the PSP size measures are to
  - define a consistent size measure
  - establish a basis for normalizing time and defect data
  - help make better size estimates
- Some of the questions these data can help to answer are
  - What size program did I plan to develop?
  - How good was my size estimate?
  - What was the completed size of the finished program?

---

---

---

---

---

---

---

---

## PSP Time Measures

- The goals of the PSP time measures are to
  - determine how much time you spend in each PSP phase
  - help you to make better time estimates
- Typical questions these data can help answer are
  - How much time did I spend by PSP phase?
  - How much time did I plan to spend by PSP phase?

---

---

---

---

---

---

---

---

## PSP Defect Measures

- The goals of the PSP defect measures are to
  - provide a historical baseline of defect data
  - understand the numbers and types of defects injected
  - understand the relative costs of removing defects in each PSP phase
- Some questions these data can help answer are
  - How many defects did I make in each phase?
  - How many defects did I remove in each phase?
  - How much time did it take to find and fix each defect?

---

---

---

---

---

---

---

---

## PSP Derived Measures

- Some PSP derived measures are
  - To Date and To Date %
  - Product size developed or reviewed per hour
  - CPI
  - % Reuse and % New Reusable
  - A/FR
  - PQI
- You will learn about these measures in the rest of the PSP course.

---

---

---

---

---

---

---

---

## Size Measurement Criteria

- Size measurements must be
  - related to development effort
  - precise
  - machine countable
  - suitable for early planning

---

---

---

---

---

---

---

---

## Size Versus Development Effort

- The principal requirement: If the size measure is not directly related to development cost, it is not worth using.
- There are many possible measures.
  - database elements
  - lines of code (LOC)
  - function points
  - pages, screens, scripts, reports
- The size measure should be sensitive to language, design, and development practice.

---

---

---

---

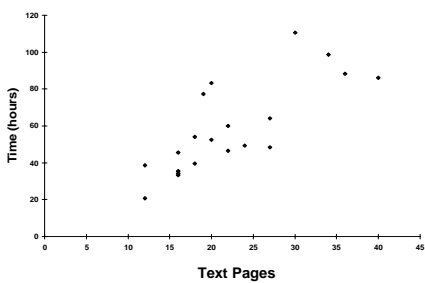
---

---

---

---

## Text Pages Versus Time



---

---

---

---

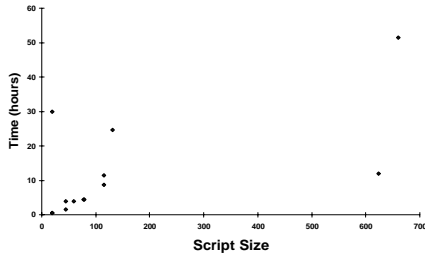
---

---

---

---

## Script Size Versus Time



---

---

---

---

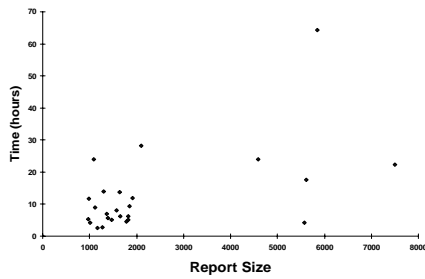
---

---

---

---

## Report Size Versus Time



---

---

---

---

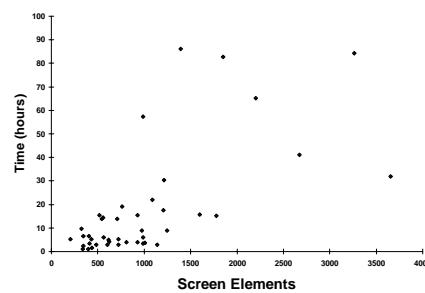
---

---

---

---

## Screen Elements Versus Time



---

---

---

---

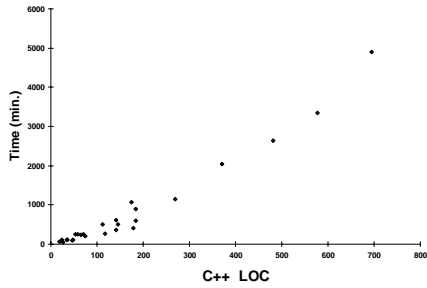
---

---

---

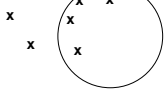
---

## C++ LOC Versus Time



## Precision and Accuracy

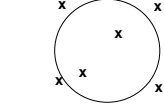
Imprecise and inaccurate



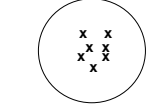
Precise and inaccurate



Imprecise and accurate



Precise and accurate



---

---

---

---

---

---

---

---

## Measurement Precision

- When two people measure the same thing, will they get the same result?
- To do so requires a precise measurement definition.
- The measure must also be properly applied.
  - Different people will likely have different definitions of database elements.
  - Pascal LOC do not equate to assembler LOC.
  - New LOC are not the same as modified LOC.
  - Logical LOC do not equate to physical LOC.
  - One person's C++ LOC may not relate to someone else's C++ LOC.

---

---

---

---

---

---

---

---

## Machine Countable

- Manual size counting is time-consuming and inaccurate.
- Automated counters will only work for defined product characteristics.
- Counters can be complex, depending on the
  - size definition selected
  - counting method used

---

---

---

---

---

---

---

---

## Suitable for Early Planning

- 1

- For making initial project plans, measures are needed that you can visualize at the beginning of the job.
  - For a house, square feet predicts cost.
  - Few people can visualize a house in terms of square feet of living space.
  - Numbers of rooms is more intuitive.
- Intuitive size measures are usually needed for initial plans.

---

---

---

---

---

---

---

---

## Suitable for Early Planning

- 2

- Unfortunately, popular intuitive measures are not often measurable, and popular measurable measures are often not intuitive.
- Function points
  - intuitive
  - not directly measurable
- LOC
  - not intuitive
  - directly measurable

---

---

---

---

---

---

---

---

## Selecting a Size Measure

- 1

- Start with product development data.
  - resources required
  - product characteristic measures
  - any special development conditions
- Rank products by resources required.
- See what characteristics distinguish those products that took the greatest effort from those that took the least.

---

---

---

---

---

---

---

---

## Selecting a Size Measure

- 2

- See if these differences are measurable.
  - Correlate a selected measure for the product set.
  - If there is no correlation, try again.
- There may be no single best measure.
  - A combination of measures could be needed.
  - Methods for handling multiple measures are discussed later.

---

---

---

---

---

---

---

---

## Selecting a Size Measure

- 3

- If you are better at estimating resources than program size, size estimation will not improve your planning.
- If you estimate resources directly, you must
  - keep accurate records
  - build a large database
  - use an estimating guru

---

---

---

---

---

---

---

---

## Counting Program Size

- Logical lines
  - invariant to editing changes
  - correlate with development effort
  - uniquely definable
  - complex to count
- Physical lines
  - are easy to count
  - are not invariant
  - must be precisely defined for each case

---

---

---

---

---

---

---

---

## CS 350 LOC Measurement

- The CS 350 LOC measure uses logical (versus physical) lines of code.
- Key advantage: it's easy to apply
- What matters is consistent measures
- Count the number of lines containing at least one semi-colon
  - Can use UNIX command:  
`grep ";" *.h *.cpp | wc -l`
  - Assumes all source code for a module is in a single directory

---

---

---

---

---

---

---

---

## Size Accounting

- For small products, size tracking can be done manually, but it requires care.
- For larger products, size tracking requires an accounting system.
- Size accounting provides an orderly and precise way of tracking size changes through multiple product versions.

---

---

---

---

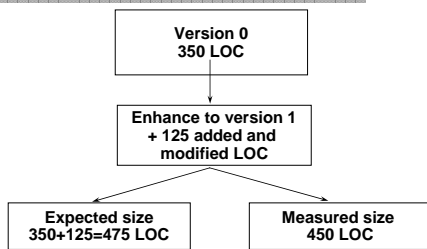
---

---

---

---

## Example of Size Accounting - 1



What happened?

---

---

---

---

---

---

---

---

## Example of Size Accounting - 2

	Added	Subtracted	Base
Base V0			0
Deleted		0	
Modified	0	0	
Added	350		
Base V1	350	-0	350
Deleted		0	
Modified	25	25	
Added	100		
V1 Product	125	-25	450
Total Added and Modified LOC			475

---

---

---

---

---

---

---

---

## Messages to Remember

- To effectively plan and manage your work, you must measure product size.
- For different types of work, use different size measures.
- For each measure, size must correlate with development time.
- If the size measure does not correlate or is not automatically countable, it will not be very useful.
- Every size measure should be precisely defined and automatically countable.

---

---

---

---

---

---

---

---

## Personal Software Process

Using PSP0.1

---

---

---

---

---

---

---

---

## Tutorial Objectives

---

- After this tutorial, you will
  - understand the PSP0.1 process
  - know how to use the PSP0.1 process scripts and forms
  - be prepared to use PSP0.1 for program 2

---

---

---

---

---

---

---

---

## PSP0.1 Objectives

---

- The objectives of PSP0.1 are to help you to
  - measure the size of the programs that you produce
  - perform size accounting for the programs that you produce
  - make accurate and precise size measurements

---

---

---

---

---

---

---

---

## New Process Elements

---

- There are two new process elements.
  - process improvement proposal (PIP) form
  - size counting and coding standards
- The project plan summary has been expanded.
  - a Program Size Summary section has been added
  - planned time in phase is calculated based on historical time in phase percentage

---

---

---

---

---

---

---

---

## PSP0.1 Process Script Additions

- The additions to the PSP0.1 process scripts include new steps for
  - estimating and reporting software size
  - distributing development time over planned project phases
  - using a size counting and coding standard
  - recording process problems and improvement ideas

---

---

---

---

---

---

---

---

## Process Improvement Proposal - 1

- To improve your process, you will need to capture process problems and propose improvements for future reference.
- You will need to know
  - any problems you have encountered in using the process
  - any suggestions you have for process improvements
  - your observations and findings from doing the assignments

---

---

---

---

---

---

---

---

## Process Improvement Proposal - 2

- You should complete a PIP form for each assignment.
- The PIP holds process improvement information.
  - date
  - problem description
  - proposed solution
  - notes and comments

---

---

---

---

---

---

---

---

## Coding and Counting Standards

- Coding and size counting standards are needed to write the PSP programs.
- These standards are
  - tailored to your language and needs
  - designed to make size counting easier
- The coding standard defines quality-oriented exit criteria for the code phase.

---

---

---

---

---

---

---

---

## PSP Software Size Measures

- In the PSP, software size measures are used to
  - relate the amount of product produced with effort expended to project total effort
  - calculate productivity
  - normalize defects
  - project the total defects
- Software size is measured in LOC.
- To accurately relate size to effort, the different types of LOC in your program are counted separately.

---

---

---

---

---

---

---

---

## PSP0.1 Project Plan Summary

- PSP0.1 adds the Program Size Summary for estimated and actual size data.
- The types of size include
  - base [B]
  - deleted [D]
  - modified [M]
  - added [A]
  - reused [R]
  - added and modified [A+M]
  - new reusable

---

---

---

---

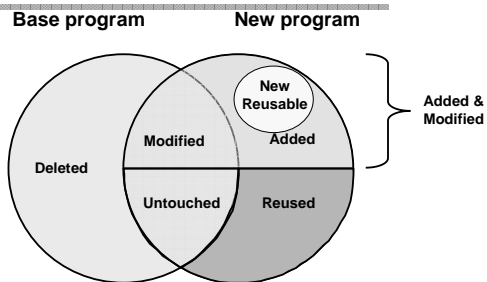
---

---

---

---

## Program Size Type Relationships



Fall 2005

CS 350/ODU

46

---

---

---

---

---

---

---

---

## Estimating Size

- During planning
  1. If this is an enhancement, measure the size of the base program and enter this in the Base (B) space under Actual.
  2. Estimate the added and modified size and enter this in the Total Added and Modified (A+M) space under Plan.

---

---

---

---

---

---

---

---

## Estimating Development Time

- During planning
  1. Enter estimated development time
  2. Planned time in phase is then calculated based on the percentage of time in phase for all completed projects

---

---

---

---

---

---

---

---

## Recording Size - 1

- During postmortem
  1. Measure total program size and enter this in the Total Size (T) space under Actual.
  2. Count the deleted size and enter this in the Deleted (D) space under Actual.
  3. Count the modified size and enter this in the Modified (M) space under Actual.

---

---

---

---

---

---

---

---

## Recording Size - 2

- During postmortem
  1. Count the reused size and enter this in the Reused (R) space under Actual.
  2. Count or estimate the number of new and changed size that will be added to the reuse library and in the New Reusable space und Actual

---

---

---

---

---

---

---

---

## Message to Remember

- Your principal objective is to measure and estimate the size of the programs that you produce so that you can effectively plan and manage your work.

---

---

---

---

---

---

---

---