Software Sizing

Experiences in the Classroom

Presented at the 18th International Forum on COCOMO and Software Cost Modeling

Jo Ann Lane **21 October 2003**

Academic Goals with Respect to Software Sizing

- Emphasize the importance of software sizing with respect to typical business goals
- Overview current techniques
- Encourage students to try one or more current techniques
- Challenge students to find a better approach

Teaching Approach

- Early in the course
  - Overview various techniques and counting rules
  - Provide guidelines for selecting an approach
- As part of team projects, students must
  - Develop initial estimate
  - Monitor estimates and update if necessary
  - Perform final count to compare to estimates
  - Identify probable causes for any significant estimation inaccuracies

Quantitative Comparison of Results Obtained to Date

- Team project characteristics
- Preferred sizing techniques
- Estimation accuracy
- Probable reasons for estimation inaccuracies

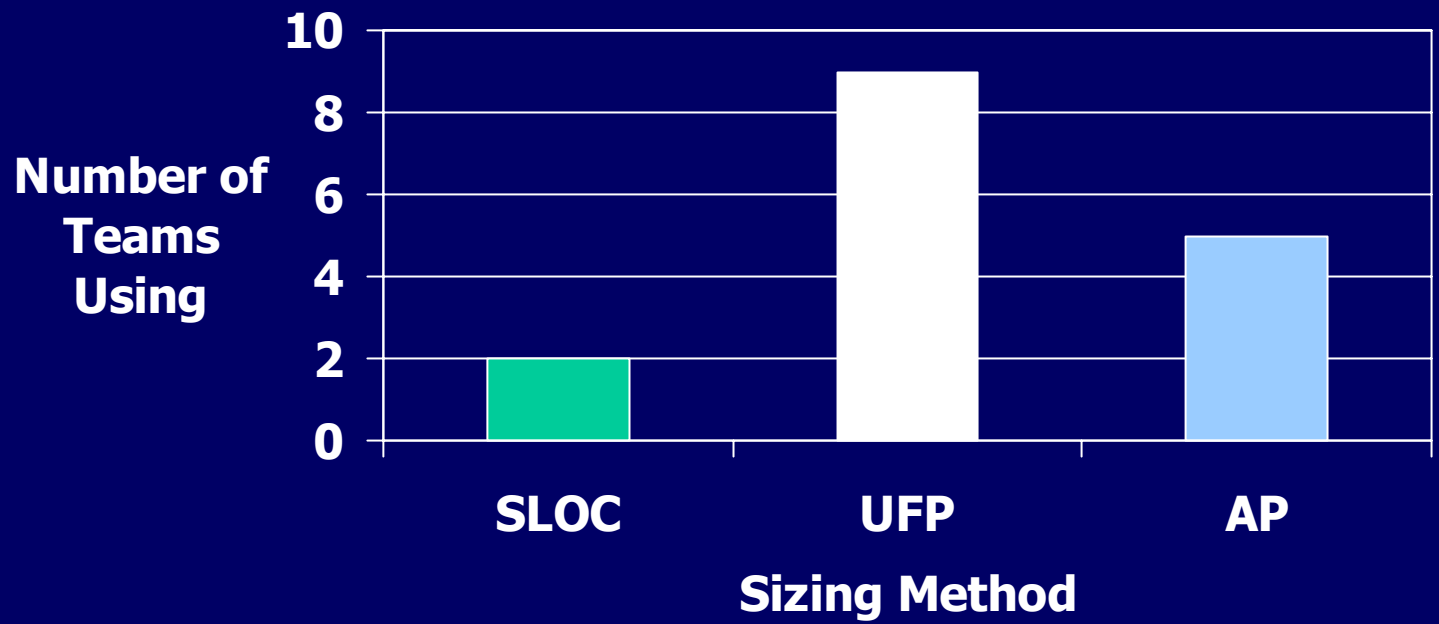
Team Project Characteristics

- Fixed schedule (1 semester/15 weeks)
- New Access database application that requires development of
  - Database schema
  - Screens
  - Reports

Team Project Characteristics
(continued)

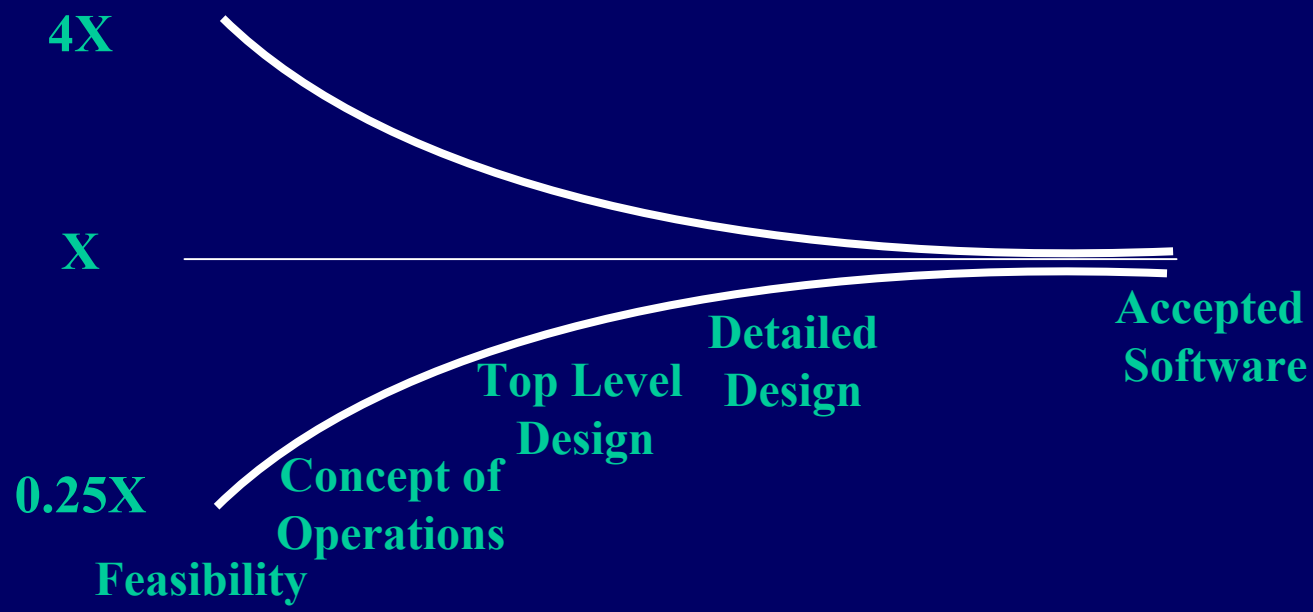
- Students provided with performance and environment requirements as well as functional requirements
- Must be developed for the NT environment
- Team size: 5-6 people
- Data collected from 16 teams over 2 years

Student's Selected Sizing Methods for Development Activities*

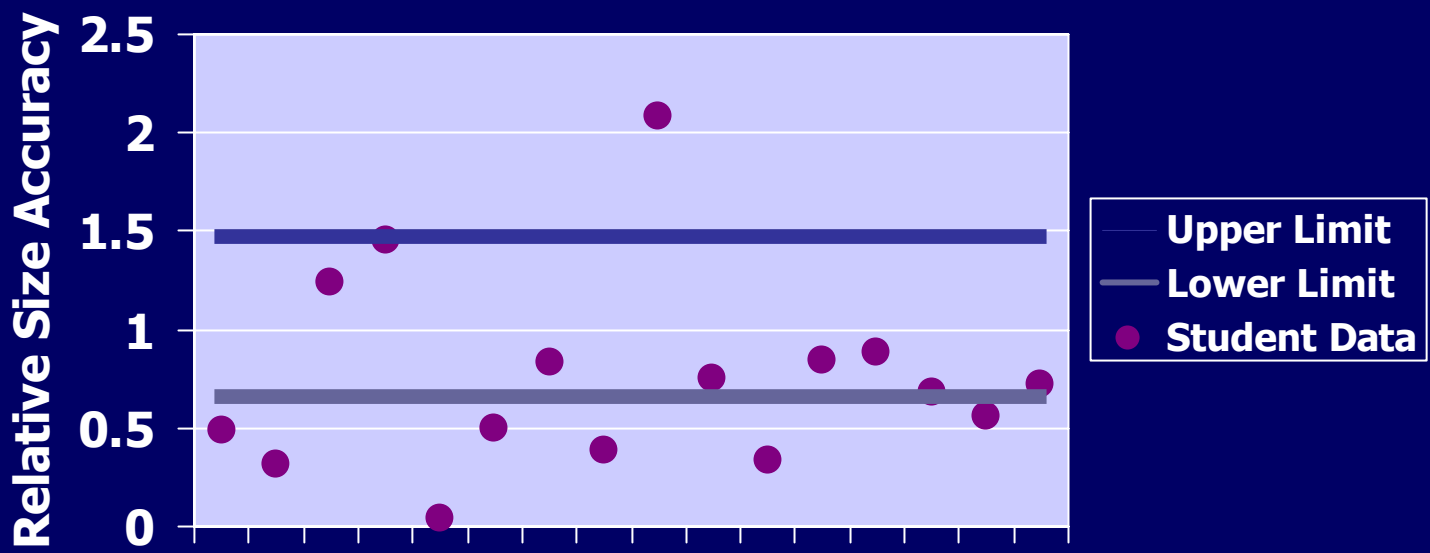


**** In addition, some testers have used use cases to estimate testing activities with some success…***

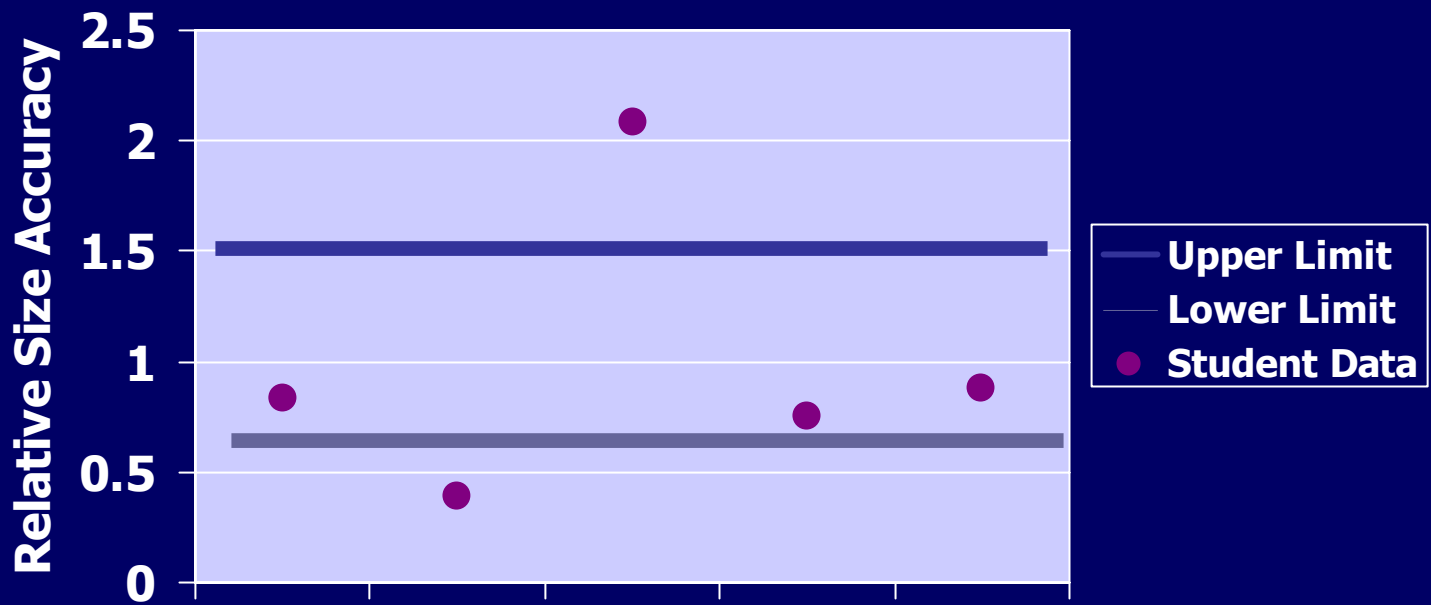
Software Sizing Accuracy Versus Phase



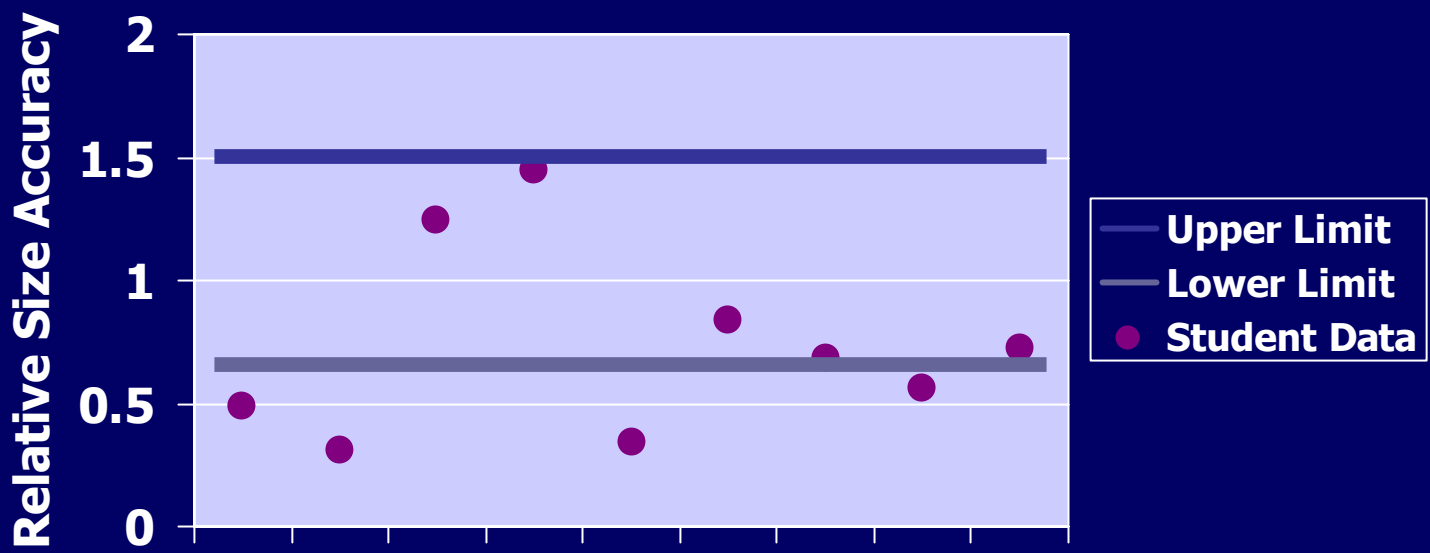
Student Data Comparison with Industry Data Accuracy at Requirements Phase: All Sizing Methods



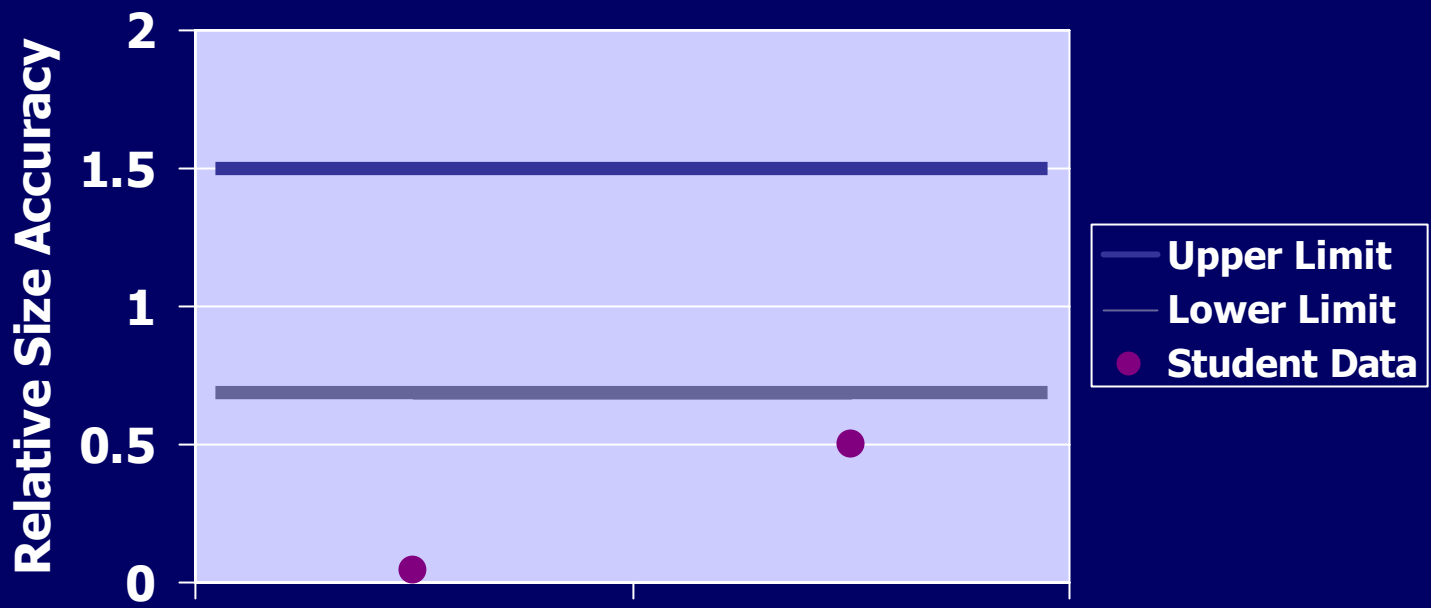
Student Data Comparison with Industry Data Accuracy at Requirements Phase: Application Points



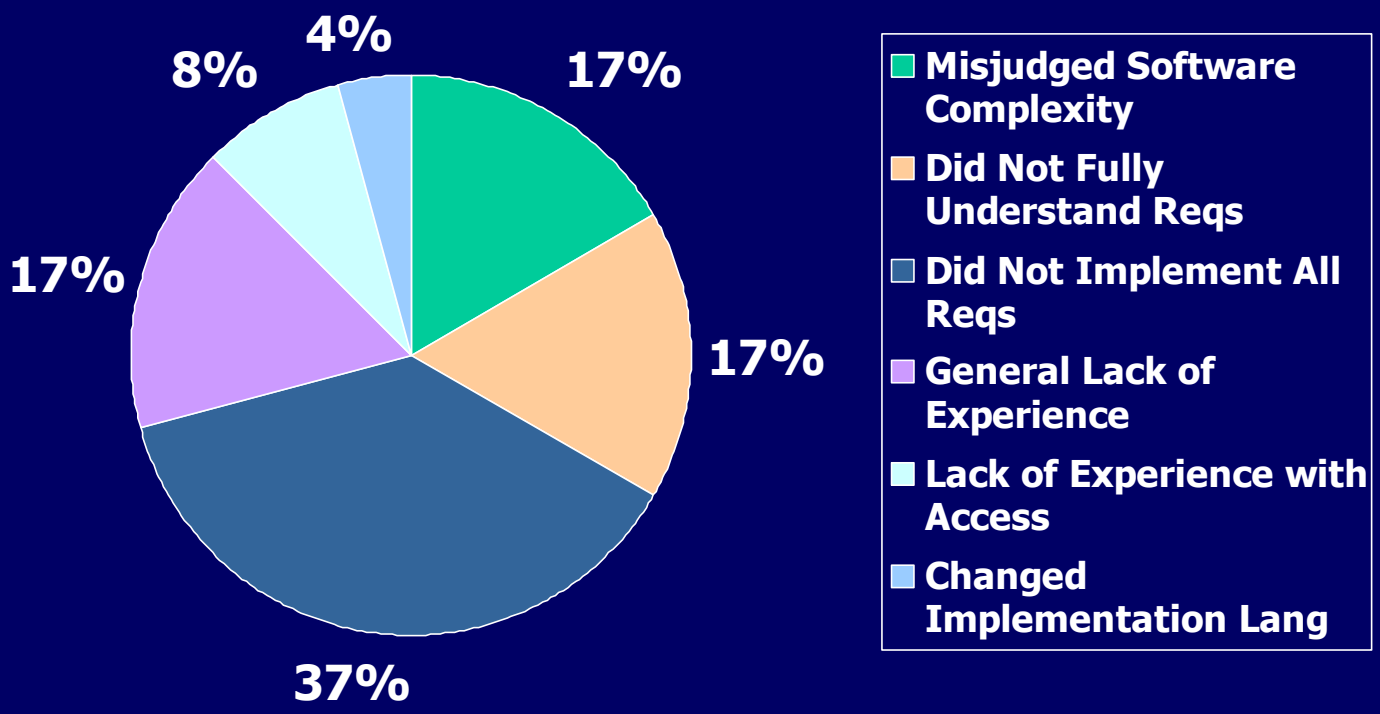
Student Data Comparison with Industry Data Accuracy at Requirements Phase: Unadjusted Function Points (UFP)



Student Data Comparison With Industry Data Accuracy at Requirements Phase: SLOC



Student Reasons for Software Size Estimation Inaccuracies



Feedback from Students

- It's HARD….
- … especially trying to estimate using SLOC
- Complexity from the student's point of view is probably very different from what was intended in sizing rules – in hindsight, students report that they probably overestimated complexity

Feedback from Prospective Employers

- Excited to hear that estimation processes are being taught
- Ask students if any new/better techniques are on the horizon (translation: hiring managers are still looking for a better way)

Instructor Assessment of Student Sizing Efforts

- To the casual observer, sizing inaccuracies for students using UFPs or APs is not much different than research data from industry
- Reasons that initial software sizes are off—not much different from reasons found on real projects

Instructor Assessment of Student Sizing Efforts (continued)

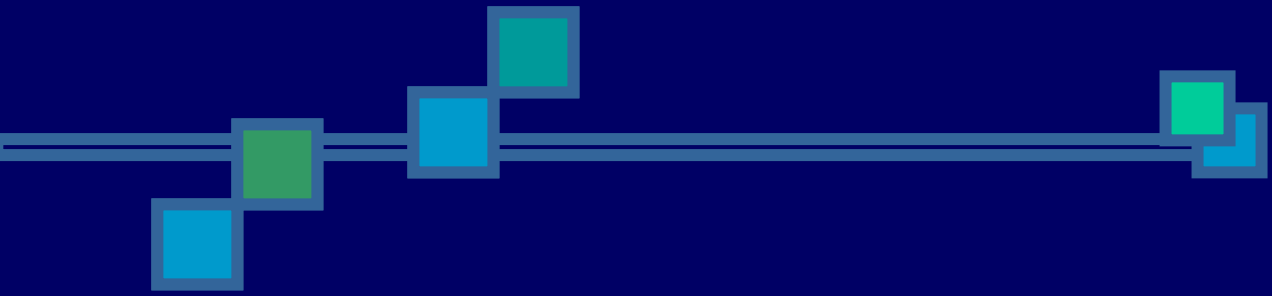
- Correlation between functional requirements and software size is fairly stable
- Correlation between performance/environment requirements and software size is not as predictable
- Current complexity factors may not adequately address some performance issues

Instructor Assessment of Student Sizing Efforts (continued)

- Some of the "hard" or time-consuming requirements that did not get completed probably had little or no impact on student's software size estimates
  - Web-enable application
  - Set up security user groups to limit system access
  - Provide help text for forms and fields

Instructor Assessment of Student Sizing Efforts (continued)

- UFP counts are probably extremely high due to lack of experience with logical file concepts… however, actual counts use the same counting rules, so counting errors are probably not significant



Questions?