Producing Production Quality Software

Lecture 14:
Group Programming Practices
Prof. Arthur P. Goldberg
Fall, 2005
Topics

• Logistics
• MIS ‘Best Practices’: Capers Jones’s view
• No Final Assignment
• Improving Software Management
• Effort Estimation
• Literature
• Productivity
Oral Exam Logistics

Out noon on Thursday Dec. 15, with 2 example answers

Last time I answer questions about the final is midnight Dec. 15

Exam session
• 20 minutes, my office
• Be on time, or be rescheduled
• 5 questions, 4 minutes each
  2 you pick
  1 chosen at random
  2 I pick for you
• Closed book
• While answering, you may talk or write on paper or a white board

Results will be released on about 12/21, after everyone has taken the exam
Best Practices

• As influences
  – Quality
  – Productivity
Software Development Quality

• Capers Jones and Software Productivity Research’s quantitative approach


  – Measure 9000 development projects from 600 organizations
    • In 150 of the Fortune 500
    • In 100 small companies
    • In many government and military organizations
Classification

• Classify organizations and projects by
  – Client country
  – Industry
  – Project ‘nature’
  – Project scope
  – By software type
  – Project class

• 37,400 permutations

• Calibrate and benchmark measures of productivity and defect rate
Project ‘Nature’

1. New Development
2. Enhancement (new functions added to existing software)
3. Mandatory change (updates for new statutes or regulations)
4. Maintenance (defect repairs to existing software)
5. Performance updates (revisions needed to improve throughput)
6. Conversion or adaptation (migration to a new platform)
7. Nationalization (migration to a new national language)
8. Reengineering (re-implementing a legacy application)
9. Mass update (modification for the Euro or Y2K)
10. Hybrid (concurrent repairs and functional additions)
Project Scope

1. Subroutine or sub-element of a program
2. Module of a program
3. Reusable module or object
4. Disposable prototype
5. Evolutionary prototype
6. Stand-alone program
7. Component of a system
8. Release of a system
9. New system or application
10. Compound system (multiple linked systems)
Software types

Non procedural (spreadsheet, query, generators, and so forth)
Web Applet
Batch application
Interactive application
Batch database application
Interactive database application
Pen-based application
Client/server application (two tier)
Client/server application (three tier)
Enterprise resource planning (ERP) application
Scientific or mathematical application

Systems or hardware control application
Communications or telecommunications application
Process control application
Embedded or real-time application
Trusted system with stringent security
Graphics, animation, or image-processing application
Robotic or manufacturing control application
Expert system with substantial knowledge acquisition
Artificial intelligence application
Neural net application
Hybrid project (multiple types)
Project class

Personal Application for private use
Personal application to be shared by others
Academic program developed in an academic environment
*Internal application to be installed at one location*
*Internal application to be accessed via an intranet or time-sharing*
*Internal application to be installed at many location*
*Internal application developed by contract personnel*
Internal application developed using military standards

External application, to be freeware or shareware
External application to be placed on the World Wide Web
External application leased to users
External application embedded in hardware
External applications bundled with hardware
External application marketed commercially
External application developed under outsource contract
External application developed under government contract
External application developed under military contract
General project classes

End-user applications - developed privately for personal use

Information systems - developed in-house for corporate use (MIS)

Outsource or contract projects - developed under legally binding contract

Commercial software - developed to be marketed to external customers

Systems software - developed to control physical devices

Military software - developed using military standards
Review
‘Group Programming Practices Handout’
• ‘Best Technical Practices for MIS Software’
• Distribution of SPR Project Benchmarks circa 1999
Software Effort Estimation

• **Issues**
  – When will the bleeping system be done?
  – How much will it cost?
  – If I give you Joe’s team, when will it be ready?

• **Approaches**
  – Little or no estimation: XP
  – Function Point Analysis
  – Lister’s “Estimating Quality Factor”
  – Wideband Delphi
Software Effort Estimation

Function Point Analysis
Function points

• A conceptual measure of complexity
• A linear sum, approximately
  – Simplistically,
    
    \[
    \text{FPC} = \text{External}\_\text{inputs} \times 4 + \text{External}\_\text{outputs} \times 7 + \text{External}\_\text{inquiries} \times 5 + \text{Internal}\_\text{logical}\_\text{files} \times 4 + \text{External}\_\text{interface}\_\text{files} \times 10
    \]
• Accuracy
  – To +/- 10%
  – IFPUG study by Kemerer at MIT in 1993
• Rarely used
Functional complexity modifier

- The multipliers vary with complexity
- Data element type (DET) count
- File type reference (FTR) count
- Review tables in ‘Group Programming Practices Handout’
FP ‘Priesthood’: International Function Point User’s Group [www.ifpug.org]

• Rules Version 4.1 1999
• Certified Function Point Specialist
  – Examination
Backfiring

• Convert SLOC to FPs
• Driven by tables in ‘Group Programming Practices Data’
• About +/- 20% accurate
Software Effort Estimation: EQF

• Tim Lister’s “Estimating Quality Factor”
• Lister
  – The Atlantic Systems Guild, Inc.
  – Books
  – Seminars
    • Risk Management for Software
    • Leading Successful Projects
Software Effort Estimation

Wideband Delphi
Delphi History

• Consensus without conflict
• Invented by RAND
  – Santa Monica ‘thinktank’
  – Used to estimate nuclear bombing damage
• Many other applications
• Wideband Delphi
  – Developed by Barry Boehm in 70s; see his *Software Engineering Economics*
Wideband Delphi Overview

• Input
  – Specification

• Outputs
  – Detailed task list
  – Estimation assumptions
  – Effort estimates from each participant
Wideband Delphi Process Overview

- Form group of experts
- *Estimate*: Each group member provides an anonymous estimate
- Coordinator assesses the estimates
  - Asks for reassessment of crazy estimates
- Combines estimates
- Presents combined results to experts
- If combined results do not converge then goto *Estimate*
Wideband Delphi

• **Benefits**
  – **Practical**
    • Employ multiple minds; promote reasoning
    • Accurate estimate avoids the underestimate dilemma of “skip & compromise quality vs. blow schedule”
    • Build comprehensive task list
  – **Psycho-social**
    • Reduce bias by influential people, or those with divergent agendas
    • May resolve disagreement among hostile parties
    • ‘Buy-in’ of ‘stakeholders’
  – **Other**
    • Acknowledges uncertainty

• **Drawbacks / costs**
  – **Time**
  – Takes power away from manager or ‘guru’ (their drawback)
Wideband Delphi - 1

• Each individual develops an estimate which is list of
  – Tasks
  – Assumption(s)
  – Effort

• Rules
  – Assume you’ll do all the work
  – Assume uninterrupted effort
Wideband Delphi – 2

• Moderator presents
  – Distribution of effort (anonymous) estimates
  – All task lists (or merged task lists)

• All participants modify estimates concurrently and secretly
Wideband Delphi

- **Termination**—at earliest of
  - 4 rounds
  - Acceptable convergence
  - Meeting time over
  - Nobody willing to change

- **Completion**
  - Moderator assembles the tasks into single list
  - Moderator merges assumptions
  - Moderator combines estimates
    - **Possibilities**
      - Ave
      - Min, ave, max
      - Ave and std-dev
Wideband Delphi Practice

• Lets try it
• First on the teller systems with the whole class
• Then on the DBMS
• Use the forms in the handout
  – Break into groups of 3 or 4
• [Record your effort if you build it]
• Compare
Improving Software Management

• Improve an organization
  – Involves
    • Technology
    • Management
    • Psychology
  – Approaches
    • CMM
    • SPR
    • Weinberg
Software Process Assessment Approaches

• SEI CMM – focus: mandated activities, and change processes
  – Review ‘Group Programming Practices Handout’
    • Five Levels of the SEI CMM
    • Five Levels of the SPR Excellence Scale
    • Approximate Conversion Between SPR and SEI Software Scores
  – Trained assessors

• SPR – focus: project
Literature
Au revoir et merci