Effective Use of Function Points for Analogous Software Estimation

Dan French, PMP, CFPS, CSM
Principal Consultant
dfrench@cobec.com
202-827-1316
www.cobec.com
- Introduction
- Definition of Analogous Estimation
- Advantages and Disadvantages of Analogous Estimates
- Best and Worst Times to Use the Analogous Estimation Technique
- Common Mistakes Made
- Proper Application of the Analogues Estimation Technique
- Challenges of Using Software Lines of Code (SLOC)
- What are Function Points (FP)
- Background of Function Points
- Why Function Points are preferable to SLOC when performing Analogous Estimating
- Organizational Infrastructure Needed to Support Good Analogous Estimating
- Sources of Information
- Conclusion
Introduction

• B.S. in Economics from Virginia Tech
• Graduate of the Chubb Institute Top Gun Program
• Over 15 years experience in software cost estimation
• Counting function points for 17 years and been a Certified Function Point Specialist (CFPS) for 15 years
• Experience in a number of estimation techniques and tools including SEER-SEM, COCOMO, SLiM, Delphi, and Estimating by Analogy
• Chairman of the International Function Point Users Group (IFPUG) Functional Software Sizing Committee (FSSC)
• Former member of the IFPUG Conference Committee for 5 years
• GAO Cost Guide expert team member
• Project Management Institute (PMI) Project Management Professional (PMP)
• Agile Alliance Certified SCRUM Master (CSM)
What is an analogous estimate

Analogous estimating is a technique which uses the values of parameters from historical data as the basis for estimating the same or similar parameter for a future activity. Example of parameters include scope, cost, duration or they could be measures of scale like size, weight or complexity.¹

¹ http://www.projectmanagementlexicon.com/analogous-estimating/
Advantages of Analogous Estimates

- It can be used early in the software development life cycle, even when detailed program requirements are unknown
- Can be developed quickly and economically
- Does not require significant amount of skill to create
- Easily understood by decision makers and other stakeholders
- If properly done, the estimate is defensible
Disadvantages of Analogous Estimates

- Appropriate analogous program may be difficult to find
- Lack of or inaccurate historical data
- High degree of subjectivity
- Higher degree of uncertainty and risk than more rigorous estimation methodologies
- Over confidence in similarity of selected program and resulting estimate
- Difficulty in assessing key factors that influence adjustments to project
Best and Worst Times to Use

• Best:
  • Early in the project (Initiation analysis phase)
  • When accurate and appropriate project actuals data is available, preferably organizational
  • Good understanding of the desired high-level system functionality
  • Clear understanding of differences between project being estimated and selected analogy

• Worst: Anytime none of the above apply
Most Common Mistakes Using Analogous Estimates

- Use of aged actuals data (>5 years)
- Lack of technically or functionally appropriate program
- Selecting the wrong analogous program
- Improperly, or not bothering to, adjusting the estimate based on differences between the selected program and the estimated program
- Not using software size as the key comparative factor
- Using SLOC as the software size comparative factor
- Using estimate data from another project estimate
What is required to do a good, defensible and reliable analogous estimate?

1. Accurate project actuals data in an accessible project database
2. Requirements to a sufficient level of detail that the project can be appropriately matched
3. Proper adjustment of analogous project data
Typical project metrics captured for use in estimation include:

- Size (FP, Story Points, Cosmic FP, Use Case Points, SLOC)
- Effort (hours, person-months, FTEs)
- Duration (day, months)
- Cost
- Staff (headcount)
- Computer Resources (CPU, Workstations, Servers, bandwidth)
- Date
• Requirements don’t have to be highly detailed to do an analogous estimate but they should be of sufficient detail and quality that:
  • They are verifiable and testable
  • Provide sufficient understanding of desired system functionality
  • Are good quality (no negative requirements)
Adjustments

• Adjustment are the critical and most challenging part of analogous estimating:
  • They should reflect the differences between the estimated project and the target analogy
  • Include factors such as platform, software language, team experience, development tools, development environment, and development methodology
  • Don’t rely on one person, make the adjustments using inputs from multiple SMEs
Challenges of Using Software Lines of Code (SLOC)

- No defined counting rules or standards organization
- Language and platform dependent
- Inconsistent rules mean there is no reliable and verifiable industry data
- Penalizes efficient software writing, incentivizes poor coding
- Heavily dependent upon developer skill and style
- Difficult to estimate early in lifecycle

Function Points address and overcome these challenges
What are Function Points (FP)?

- Function Points are a unit of software size measure
- Measure the work product of software development
- Work product is measured in terms of functionality from user perspective
- Functions points do not measure internal architecture, effort, or technological complexity of an application
Function Point History

- Developed by Allan Albrecht of IBM in 1979
- Created as an alternative to Source Lines of Code (SLOC) for measuring software size
- Counting Rules are established by the International Function Point Users Group (IFPUG)
- Current version is 4.3.1, Released in January 2010
- International Standards Organization (ISO) Standard for software functional sizing (ISO/IEC 20926 SOFTWARE ENGINEERING - FUNCTION POINT COUNTING PRACTICES MANUAL)
Why Function Points are Preferable to SLOC When Performing Analogous Estimating

• Oldest and most utilized functional size metric
• Codified set of rules
• Platform and language independent
• Functional vs. technical viewpoint
• Can be applied to all software applications
• More accurate estimation
• Not dependent upon software developer skill level
• More consistent and accurate metrics
How FP Are More Effective When Used for Analogous Estimates?

• Size, based on requirements, is the most important factor for cost in software projects. Correlation ($r^2$) ranges between .7 and .8

• SLOC does not allow for good comparisons between projects due to:
  1. lack of consistent counting rules and standards
  2. dependence upon developer skills
  3. language and platform variances

These factors all add significant uncertainty to any analogous effort. Function points minimize these risks
Organizational Infrastructure Needed to Support Good Analogous Estimating

- Senior management sponsorship and support
- Identified Key metrics and processes and procedure for data collection
- Time Tracking at project and preferably task level and accurate time recording
- Detailed and accurate project cost accounting
- Dedicated and properly trained Metrics Team responsible for:
  - Regular (typically monthly) data collection
  - Metrics database development and maintenance
  - Process and procedure ownership, enforcement and maintenance
Sources of Information

• These organizations can assist in establishing a metrics program or providing industry data for use until a metrics program is established:

  • International Function Point Users Group (IFPUG) (www.ifpug.org)
  • International Software Benchmark Standards Group (www.isbsg.org)
  • International Cost Estimating and Analysis Association (http://www.iceaaonline.com/)
  • Systems and Software Consortium, Inc. (www.software.org)
  • Software Engineering Institute (SEI) (www.sei.cmu.edu)
  • Vendors: Q/P Management, David Consulting Group, QSM, Longstreet Consulting
Conclusions

• Analogous estimation is a good software estimation technique to use early in the software development lifecycle
• Must have good data upon which to base estimates
• Must properly adjust data to develop a good estimate
• Using function points as the size metric helps reduce risk and uncertainty
Additional Information
Types of Function Point Counts

- Function points are used to count both projects and applications.
- There are 3 types of function point counts:
  - Development Project
  - Count of new software (including conversion functionality)
  - Enhancement Project
  - Count of enhancements to existing software functionality (added, changed, or deleted)
  - Application
  - Count of an application installed in production
Function Points Transaction Definitions

• Five Functional Components, 3 Transactional and 2 Data
  • Transaction Functions
    • External Inputs (EI) – Batch transaction file, input screen, control information
    • External Outputs (EO) – Reports with calculations, output files with derived data
    • External Inquiries (EQ) – On-line query screen, interface file with no calculations or derived data
  • Data Functions
    • Internal Logical Files (ILF) – Application file, internal database
    • External Interface File (EIF) – Reference