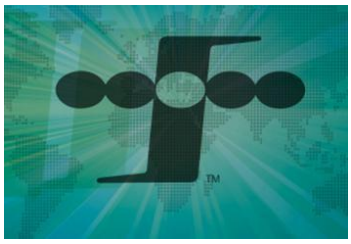

iTip	**Guidance from the Non-functional Sizing Standards Committee on topics important to you.**	
# 04	December 2014	

Counting Algorithms Using SNAP

This iTip focuses on SNAP sub-category 1.2 Logical and Mathematical Operations, specifically counting algorithms.

Background

Historically, a perceived gap in the software metric community's functional metric methodologies has been the general inattention to algorithms. A few proposed specific algorithm counting methods have arisen in the community from time to time, but none have gained permanent traction. Some software project managers have avoided any brand of function points because their applications can be algorithmic-intensive and require a significant level of work effort for their programming and testing; these managers have instead chosen other sizing metrics. There are algorithms in engineering and military applications. There are also many algorithms in business, as experienced by graduate school students who typically take an entire semester course studying algorithms in their MBA programs. So when counting an EO using function points, it is now important to also consider whether that EO was generated with the help of an algorithm.

IFPUG has made a software metrics technology improvement by recognizing the algorithm as a specific, non-functional characteristic of software. An algorithm is defined in the SNAP Assessment Practices Manual (APM) as "a series of mathematical equations and calculations executed in conjunction with, or according to, logical operators to produce results identifiable to the user." The APM has a simple method for sizing them – based on the number of DETs "required to operate the extensive mathematical operation, such as values for an algorithm's variables and settings maintained by the algorithm's control information." Also included in the algorithm's sizing is the count of the ILFs containing the algorithm's DETs. The details can be found in the APM in Sub-category 1.2 Logical and Mathematical Operations.

Counting Example

Here is a simple example of using an algorithm, adapted from a business textbook. [1]

A company wants to build an application to forecast its television sales. The EO component of the application will contain a table with three DETs: the first column being for the forecast year, the second being for the forecast year's forecast quarter (Winter, Spring, Summer, and Fall), and the third being for the forecast quarter's forecast sales. Assuming the data for this EO is from one ILF, this will be counted in function points as one low EO, at four function points.

However, this forecast is not the result of just a simple calculation. Behind the scenes, a statistical regression analysis is completed on the data to determine the forecast trend component. Then, a series of calculations are completed to compute seasonal indices. Then, the regression trend line is mathematically extended into the next year's four forecast quarters and subsequent data points for those quarters are seasonally adjusted. Finally, those seasonally adjusted data points are placed into the EO table. All of the extra algorithmic work is quantified through SNAP. Using the Subcategory 1.2 methodology, first consult Table 1-8. Since there is one ILF as the FTR, this is a low complexity logical or mathematical operation. Going to Table 1-9, further classify this as a mathematical operation. Finally, going to Table 1-10, one low complexity mathematical operation is counted under SNAP at "3*#DETs" – in this case, nine SNAP points. The total size of this component of the application is therefore sized at four function points and nine SNAP points.

Algorithm Examples

Here are some examples of mathematical operations which, depending on the situation, could be likely candidates for using algorithms and therefore countable under SNAP.

- Project scheduling using PERT (Program Evaluation and Review Technique).
- Simplex linear programming.
- Military applications.
- Analytic Hierarchy Process decision making.
- Decision tree decision making.
- Shortest route through a network calculations.
- Calculus integration formulas.
- Methods for solving differential equations.
- Complex tax calculations.
- Financial return on investment calculations for a large industrial machine.
- Statistical Analysis of Variance calculations.
- Monte Carlo simulations of financial forecasts.
- Financial payment schedules.
- Retirement pension calculations.

Summary

The IFPUG SNAP method for sizing algorithms is simple, easy to apply, and does not require either "digging" into software code or extensive knowledge of the algorithm's technology by the SNAP counter as some other approaches might require – it requires just identifying and counting DETs and ILFs. And the counted SNAP points will help justify the work effort required to build those algorithms.

Further Reading

Software Non-functional Assessment Process (SNAP) Assessment Practices Manual Release 2.2, Sub-category 1.2 Logical and Mathematical Operations.

SNAP Case Study 1: Assessing the Size of Extensive Mathematical Operations Using SNAP.

Reference

[1] Anderson, David R., Sweeney, Dennis J, and Williams, Thomas A, "Quantitative Methods for Business, Seventh Edition," South-Western College Publishing, Cincinnati, OH, USA, 1998, pgs. 180-188.

We hope that you found this iTip helpful. Additional comments and questions can be sent to the NFSSC mailbox: nfssc@ifpug.org