

“Function points are good at measuring functional requirements but what can we use to measure non-functional requirements?”

May 2014

Scope of this Report

Most if not all CFPS (Certified Function Point Specialists) have encountered projects that sit outside the organizational norm with significant development requirements for non-functional work.

Function points by their very name are focused on delivered functionality so do not provide for non-functional development.

SNAP (Software Non-functional Assessment Process) has been developed to supplement the FP sizing methodology and provide a sizing technique for the non- functional component.

This report considers the following:

- When do I need to use a non-functional measure?
- Where did SNAP come from?
- What does it capture?
- How do I implement it?
- Do estimating products cover SNAP?
- Can SNAP productivity be compared against the rest of the Industry?
- Are there alternatives?

When do I need to use a non-functional measure?

Where a significant part of the project effort (and atypical for the organization) relates to non-functional development, the project’s productivity measured using function points will appear to be lower than the norm and this can lead to animated discussions with the project team.

The following are just some of the common software development activities excluded by IFPUG counting rules but covered under the SNAP framework. They deliver no change to the functional process or are done for purely technical reasons and are not user-recognizable.

- Creation of static code tables or parameter files
- Addition of new reference data entries to an existing table
- Database changes for performance reasons
- User Interface (UI) changes for cosmetic reasons

These activities can significantly increase effort and cost and therefore impact a project’s productivity or perceived performance.

This isn't a failing or a gap in the Function point methodology but a shift in perspective. We should consider it in conjunction with function points to

- provide insight into the delivery of projects and maintenance of applications,
- assist in project estimating, and
- provide insights for the analysis of quality and productivity performance.

Where did SNAP come from?

Our recommended approach to non-functional sizing is SNAP - a methodology developed by the International Function Point Users Group (IFPUG) to meet the needs of the industry to have a generic sizing process for non-functional requirements or functional requirements not captured by function points.

IFPUG recognized the issue and began development of the SNAP counting process and SNAP points in 2008.

- Project set up to create a framework how to size non-functional requirements in 2008
- APM (Assessment Process Manual) 1.0 – Released for Beta test June 2011 with updates to August 2012
- APM 2.0 – January 2013
- APM 2.1 – April 2013

A virtual team from around the globe representing many different companies had and continues to have input into the production of the SNAP method via the IFPUG SNAP forum.

The SNAP counting process uses many definitions in line with the IFPUG counting practice manual for example, application boundaries are consistent, the view of an elementary process is the same and it recognizes a process may have both functional and non-functional characteristics.

What does SNAP capture?

SNAP is used to determine the size of the non-functional components of software. Non-functional characteristics include things such as compatibility, maintainability, usability, portability and reliability. In addition, other important characteristics that SNAP considers are security, performance efficiency and functional stability. In simple terms, Function Point Analysis (FPA) describes what the software will do, while SNAP indicates how the software will do it.

SNAP is divided into 4 major categories and a number of sub-categories to reflect the variety and nature of non-functional requirements.

1 Data Operations

The Data Operations Category relates to how data is processed within the Snap Counting Unit to meet the Non-Functional requirements in the Application.

- 1.1 Data Entry Validation** – complexity of the validation nesting level per elementary process
- 1.2 Logical and Mathematical Operations** – extensive logical or mathematical decisions applied to the elementary process
- 1.3 Data Formatting** – Data transformed during display to a user, e.g. replacing part of a credit card number with **** or decoding code data for display.
- 1.4 Internal Data Movements** – Data moving purely within the application boundary, e.g. data backup within application boundary.
- 1.5 Delivering Added Value to Users by Data Configuration** - by adding, changing or deleting reference data/ code data information from the database with no change in code or database structure.

2 Interface Design

Interface Design assesses the design of UI processes and methods that allow the user to interface with the application.

- 2.1 User Interfaces** - Unique, user identifiable, independent graphical user interface elements added or configured on the user interface that do not change the functionality of the system but affect non-functional characteristics.
- 2.2 Help Methods** – considers the approach to help on the application.
- 2.3 Multiple Input Methods** – where multiple methods have not been considered for FP but there are multiple media inputs.
- 2.4 Multiple Output Methods** - where multiple methods have not been considered for FP but there are multiple media outputs, e.g. web screen, phone, pdf, csv and xls of the same report.

3 Technical Environment

Technical Environment assesses technology as well as changes to internal data and configuration that do not provide added or changed functionality from a Function Points perspective.

- 3.1 Multiple Platforms** - Operations that are provided to support the ability of the software to work on more than one hardware or software platform, for example multiple browsers IE9, Firefox, Safari, chrome or platforms, IOs , Android.
- 3.2 Database Technology** - Features and operations that are added to the database or to the statements to read / write data to and from the database to deliver NFR without affecting the functionality that is provided , for example, Indexes, partitioning, clustering.
- 3.3 Batch Processes** - processes which are triggered within the boundary of the application, fetch data from the database, do certain transformations and then update the database without crossing the boundary.

4 Architecture

Architecture relates to the design and coding techniques utilized to build and enhance the application. SNAP assesses the complexities of modular and/or component based development.

4.1 Component based software - Pieces of software used to integrate with previously-existing software or to build components in the system

4.2 Multiple Input / Output Interfaces - Additional input and output interfaces due to growing number of users and volume of data over a period of time are covered in this subcategory.

How is SNAP implemented?

Although the process looks a little daunting, the good news is that - unlike FPA analysis - you can pick and choose which categories you want to implement and when you will implement them. You should consider the basic SNAP fundamental course as a first step or approach authorized providers of SNAP training and consulting for assistance.

Once you determine the scope of the metrics capture for size and estimation, then consider the following implementation activities.

- Determine if organizational historical data is available or can be reasonably derived.
- If not, identify future projects where you can capture the size metrics by category and build a historical repository for those types.
- When planning projects separate the non-functional development tasks from the functional tasks in the work breakdown structure. This gives a much more balanced view of a project's productivity and ensures you are comparing similar activities.
- Unlike FPA, multiple categories can apply to the same process. In that situation, you count each SNAP category that applies to that process.
- Decide if you want to implement on all projects or projects which are atypical with a large non-functional component.
- Gather data across different technologies to set benchmarks for future estimating.
- Use the excellent SNAP counting spreadsheet available from IFPUG – details to download can be found at <http://www.ifpug.org/about-ifpug/about-snap/>

Do estimating products cover SNAP?

Common tools such as SEER-SIM and SLIM¹ don't formally have SNAP points as a size metric as there is very limited industry data available due to the newness of the technique. It is possible to use custom alternate size metrics so you can adapt the tools to use your own organizational SNAP data.

Can SNAP productivity be compared against the rest of the Industry?

It is not possible to benchmark your SNAP productivity against the rest of the industry at present as the methodology is new and there is very limited data available. In time it is likely that benchmark data will

¹ SEER-SIM is a registered trademark of Galorath, Inc. SLIM is a registered trademark of Quantitative Software Management, Inc.

become available and the value of the new measure when applied to industry as a whole will be realized.

By gathering your own data you can

- assist your company with benchmarks for FP based productivity,
- build your own historical database for future reference, and
- use the technique positively as a project management tool.

Is there an alternative?

It is possible to use your own internal size measures in place of the SNAPs but you lose the ability to compare your performance against industry in the future, and our experience of in-house generated measures is that the quality of the results gets worse over time, particularly when the inventor of the method changes jobs.

However, if you find any situations not considered by SNAPs, determine a custom size metric and follow similar processes.

- Determine if organizational historical data is available.
- Identify future projects where you can capture the size metric and build a historical repository for that type.
- Ensure that the method is fully documented and under quality and change control.

Conclusion

With SNAPs there is now consistent size metric that we can use to measure non-functional changes.

In many projects it is unnecessary to use SNAP, as the amount of non-functional change impacted is not out of the ordinary (or typical for those projects used as a benchmark/ comparison).

The benefit is found when the degree of non-functional change is likely to skew the total effort needed to deliver the project and by extension have an untoward impact of productivity. What % of effort that represents will vary from organization to organization but both the project manager and FP counter are likely to highlight projects that fall into this category.

SNAP enhances project metrics to give a more complete view of project size which greatly benefits the understanding of project performance and ultimately effort estimation.

Sources:

- IFPUG Software Non-functional Assessment Process SNAP 2.1 Workshop Presentation materials

- SNAP APM V2.1.