Introducing Software Non-Functional Assessment Process (SNAP)

For years, Function Point Analysis (FPA), a method for the functional sizing of software, has been used worldwide by companies of all sizes and industries. FPA involves sizing the functional requirements of a project or application for more effective management of the software deliverable. Typically, FPA is used in conjunction with other standard measures to estimate the size of software projects, measure performance and evaluate software quality.

The International Function Point Users Group (IFPUG) is the governing body that oversees the FPA sizing practices and guidelines and certifies sizing specialists and training materials. Recently, IFPUG released a new, non-functional sizing method, the Software Non-Functional Assessment Process (SNAP), the guidelines for which can be found in IFPUG’s SNAP Assessment Practice Manual (APM). Newly introduced to the software industry, SNAP is quickly being accepted into the sizing community by practitioners and service providers, including DCG Software Value.

Function Point Analysis Versus SNAP

FPA involves the identification of logical functional components of software that are user recognizable. These components include all unique inputs, outputs, inquiries, logical groups of data and interfaces. Each unique component is evaluated based on pre-defined complexity criteria and is assessed a function point value. When all components are accounted for and assessed, the values are added together for a computed total size. The resulting size is then used in conjunction with other metrics to determine such industry standard measures as productivity (effort/size) and quality (defects/size).

Alternatively, 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, FPA describes what the software will do, while SNAP indicates how the software will do it.

The SNAP framework is best used in conjunction with function points to provide insight into the delivery of projects and maintenance of applications, to assist in project estimating and to provide insights for the analysis of quality and productivity performance. Other benefits include the ability to:

- Better plan and estimate projects.
- Identify areas of process improvement.
- Quantify the impacts of the current non-functional strategies.
- Provide specific data when communicating non-functional issues.

How to Use SNAP

The SNAP Assessment Practice Manual (APM) outlines the sizing process as follows:

1. Determine the assessment purpose, scope, boundary and partition
2. Associate non-functional requirements to categories and sub-categories
3. Identify the SNAP Counting Units
4. Determine the complexity of the SNAP Counting Units
5. Calculate the SNAP size of the SNAP Counting Units
6. Calculate the non-functional size
1. **Determine the assessment purpose, scope, boundary and partition**

These guidelines are similar to the FPA guidelines for purpose, scope and boundary. With SNAP, the purpose determines the type of non-functional assessment (application, development or enhancement) and the scope of the required assessment. While FPA uses a boundary to determine entities, SNAP uses a similar concept called a “partition,” which is described in the APM as follows:

“A partition is a set of software functions within an application boundary that share homogeneous assessment criteria and values. A partition requires separate development effort, that may not be reflected when sizing the functional aspect of the project/product, using FPA.”

Partitions are positioned within the application boundary and are used to gauge the SNAP entities to evaluate. The intent is to cover all of the non-functional areas of the application, and there are specific guidelines to set the partitions. As in FPA, the purpose and the scope will influence the positioning of the boundary and the partitions within the boundary. Examples of partitions are:

- The client functions of a client-server application
- The server functions of a client-server application
- The entire application (a single partition)
- The internal batch processes of an application

2. **Associate non-functional requirements to categories and sub-categories**

A category is a group of components, processes or activities that are used in order to meet the non-functional requirement. There are four categories (Data Operations, Interface Design, Technical Environment and Architecture), which consist of 14 subcategories. A sub-category is defined as a component, a process or an activity executed within the SNAP Counting Unit (SCU), to meet the project’s non-functional requirement. The subcategories generally cover the technical items that cannot be counted in FPA. For example, internal data processing cannot be counted under FPA because no data crosses the boundary; however, internal data processing can be evaluated and considered under Data Operations in SNAP. Code data can also be counted within the SNAP framework. The table below itemizes the categories and their subcategories.

<table>
<thead>
<tr>
<th>Data Operations</th>
<th>Interface Design</th>
<th>Technical Environment</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Data Entry Validation</td>
<td>2.1 UI Changes</td>
<td>3.1 Multiple Platforms</td>
<td>4.1 Component based software</td>
</tr>
<tr>
<td>1.2 Logical and Mathematical Operations</td>
<td>2.2 Help Methods</td>
<td>3.2 Database Technology</td>
<td>4.2 Multiple Input / Output Interfaces</td>
</tr>
<tr>
<td>1.3 Data Formatting</td>
<td>2.3 Multiple Input Methods</td>
<td>3.3 Batch Processes</td>
<td></td>
</tr>
<tr>
<td>1.4 Internal Data Movements</td>
<td>2.4 Multiple Output Methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5 Delivering Added Value to Users by Data Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Operations</th>
<th>Interface Design</th>
<th>Technical Environment</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Data Entry Validation</td>
<td>2.1 UI Changes</td>
<td>3.1 Multiple Platforms</td>
<td>4.1 Component based software</td>
</tr>
<tr>
<td>1.2 Logical and Mathematical Operations</td>
<td>2.2 Help Methods</td>
<td>3.2 Database Technology</td>
<td>4.2 Multiple Input / Output Interfaces</td>
</tr>
<tr>
<td>1.3 Data Formatting</td>
<td>2.3 Multiple Input Methods</td>
<td>3.3 Batch Processes</td>
<td></td>
</tr>
<tr>
<td>1.4 Internal Data Movements</td>
<td>2.4 Multiple Output Methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5 Delivering Added Value to Users by Data Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. Identify the SNAP Counting Units

The Snap Counting Unit (SCU) is the component or activity in which complexity and SNAP size is assessed. The SCU can be a component, a process or an activity identified according to the nature of the sub-category. There is one definition of the SCU for each of the sub-categories.

4. Determine the complexity of the SNAP Counting Units
5. Calculate the SNAP size of the SNAP Counting Units
6. Calculate the non-functional size

The non-functional size of each of the 14 subcategories is determined using its complexity parameters. These assessment criteria are defined in the sub-category definition. The SNAP Points (SP) for each sub-category is determined by using the defined equations or tables for each sub-category. Once all relevant complexity parameters have been assessed, the size of each SCU is calculated and the SP of all SCUs are added together to obtain the calculated SP for the sub-category.

Conclusion

Ideally, SNAP should be used in tandem with FPA. A project or application would have the unadjusted function point count for functional requirements and a SNAP count for the non-functional requirements. These numbers are not added together, but considered separately when used in estimation modeling or other metrics. As with FPA, historical data can be collected to determine SNAP productivity and other non-functional metrics. Using the two complimentary methods, organizations now have a comprehensive sizing tool for portfolio management.

DCG is an authorized provider of FPA and SNAP consulting and training Advisory Services. The basic SNAP fundamentals course, offered by DCG, is the best path for those interested in getting started with the method. As the use of SNAP continues to expand throughout the IT world, we look forward to supporting organizations in implementing both SNAP and FPA programs for increased insight into their software.

Attribution:
This article was largely based on information from the IFPUG Software Non-functional Assessment Process SNAP 2.1 Workshop Presentation materials.