Introduction to SNAP Assessment Practices Manual Release 2.0

Non-Functional Sizing Standards Committee
ISMA7 October 2012
How SNAP came into being

2007
- IFPUG approval to ITPC for project ‘Technical Sizing Framework’
- Goal: Define a framework covering technical aspects of software not covered by Function Point

2009
- First draft version of Software Non-functional Assessment Process (SNAP) out for review
- IFPUG SNAP Release 0.1 (Oct 2009) released

2010
- First Beta version released for pilot in industry
- Post industry feedback SNAP APM Release 1.0 BETA released

2011
- Further beta test in May 2011 across globe
- SNAP APM Release 1.0 launched at ISMA 6 workshop

2012
- ISMA6 workshop feedback reviewed
- APM updated with feedback areas
- Beta test in 10 countries and 18 organizations
- APM updated post beta test findings
- SNAP APM Release 2.0 launched at ISMA 7 workshop
Non-functional characteristics (ISO/IEEE 9126-1, 25010)

ISO standard 25010

- Portability
- Security
- Usability
- Performance efficiency
- Reliability
- Maintainability
- Compatibility
- Functional Suitability

Quality Model
Non-functional characteristics (ISO/IEEE 25010)

<table>
<thead>
<tr>
<th>Functional Suitability</th>
<th>Usability</th>
<th>Security</th>
<th>Maintainability</th>
<th>Performance Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completeness</td>
<td>Recognizability</td>
<td>Confidentiality</td>
<td>Modularity</td>
<td>Time Behavior</td>
</tr>
<tr>
<td>Correctness</td>
<td>Learnability</td>
<td>Integrity</td>
<td>Reusability</td>
<td>Resource Utilization</td>
</tr>
<tr>
<td>Appropriateness</td>
<td>Operability</td>
<td>Non-Repudiation</td>
<td>Analyzability</td>
<td>Capacity</td>
</tr>
<tr>
<td></td>
<td>User Error Protection</td>
<td>Accountability</td>
<td>Modifiability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User Interface Aesthetics</td>
<td>Authenticity</td>
<td>Testability</td>
<td></td>
</tr>
<tr>
<td>Portability</td>
<td>Accessibility</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Install ability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace ability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Do we measure the NFRs?

A requirement to improve performance can be met by:

- Adding or upgrading the hardware: No added size
- Improve database capabilities (adding indexes, views…): Size the database changes
- Improve the code: Size the code changes
- All of the above: Size the database changes, Size the code changes
How does it fit with SNAP?
Categories and Sub-categories

Definition:

- A Category is a group of components, processes or activities that are used in order to meet the Non-Functional requirement.
- 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.

Categories do not replace or redefine non functional requirements; they define how these requirements are met.
Categories and subcategories

**Data Operations**
- 1.1 Data Entry Validation
- 1.2 Logical and Mathematical Operations
- 1.3 Data Formatting
- 1.4 Internal Data Movements
- 1.5 Delivering Added Value to Users by Data Configuration

**Interface Design**
- 2.1 UI Changes
- 2.2 Help Methods
- 2.3 Multiple Input Methods
- 2.4 Multiple Output Methods

**Technical Environment**
- 3.1 Multiple Platforms
- 3.2 Database Technology
- 3.3 Batch Processes

**Architecture**
- 4.1 Component based software
- 4.2 Multiple Input / Output Interfaces
What's different in Release 2.0?

✓ Improved Concepts, Processes, and Rules for assessing Software Non Functional Requirements

✓ Enhanced definitions were added to clarify the terms

✓ Some Sub-categories re-defined based on user’s experience and comments

✓ Guidelines added for link between SNAP and FPA

✓ Guidelines for sizing requirements involving both functional and non-functional requirements
Software Non-functional Assessment Process

1. Determine Assessment Purpose, Scope and Boundary
2. Associate Non-Functional Requirements to Categories and Sub-Categories
3. Identify the SCUs
4. Determine the complexity of the SCU
5. Calculate the SNAP Point of the SCU
6. Calculate the non-functional size

SCU: SNAP Counting Unit
SNAP Calculation Approach

For each non-functional requirement, it is possible to determine the non-functional size in three steps:

1. For each of the sub-categories, identify the SCUs
2. Determine the non-functional size (SP) for each SCU within the sub category, by using the equations or the tables for the sub-categories
3. Determine the SP for a specific project or application by using the formula for the project type in question

The SCU is a component or activity, in which complexity and size is assessed. The SCU can be a component, a process or an activity identified according to the nature of the sub-category/sub-categories.
Non-functional Size of a Sub-category

The non-functional size of each sub-category shall be determined using its complexity parameters.

There is one definition of the SCU for each of the sub-categories. These assessment criteria are defined in the sub-category definition.

The SP for each sub-category is determined by using the defined equations or tables for each sub-category.
SNAP and FPA

Non-functional size should be used in conjunction with Functional size
To provide overall view of the software application

Assessing the effort impact on projects as a result of the non-functional is not in scope of SNAP

Organizations should collect and analyze their own data to determine productivity
A requirement may contain both functional and non-functional aspects

Functional size measured in Function Points; Non-functional size, measured in SNAP Points

Requirement should be broken into its functional and non-functional components
The segregation should be agreed by both the users and developers

Use FP for FR related tasks and SP for NFR related tasks
Beta Test across the globe
Thank You Very Much
Thank You