IFPUG SNAP
v2.1
(Software Non-functional Assessment Process)
Quick Guide
© IFPUG 2013

Author: Luigi Buglione, Engineering.IT

SNAP Objectives

- Measure the non-functional size of the software that the user requests and receives
- Measure software development and maintenance based on the non-functional requirements (such as technology used for implementation)

Possible Types of Assessments

- Development Project SNAP Points (**DSP**): an assessment of the NFRs provided to the users with the first release of the software, as measured by the development project SNAP assessment by the activity of applying the SNAP Method.
- Enhancement Project SNAP Points (**ESP**): a measure of the non-functional characteristics added, changed or deleted at the completion of an enhancement project, as measured by the enhancement project SNAP assessment.
- Application SP after the Enhancement project (**ASPA**): a measure of the non-functional characteristics that an application provides to the user, determined by conducting the application SNAP assessment. It is also referred to as the baseline or installed non-functional size.

Counting procedure - Steps

1. Determine assessment purpose, scope, boundaries & partition
2. Associate non-functional requirement (NFRs) with Categories and Sub-Categories
3. Identify The SNAP Counting Units (SCU's)
4. Determine the complexity of each SCU
5. Calculate the SNAP size of each SCU
6. Calculate the non-functional size

SNAP & FPA

- 'FP + SNAP Points' are **not** equal to the overall project size.
- The size of a software application is considered to have two distinct parts: the size of the functional requirements and the size of the non-functional requirements.

Elementary Process (EP)

- It is the smallest unit of activity that is meaningful to the user
- It must constitute a complete transaction
- It must be self-contained
- It must leave the business of the application being counted in a consistent state

Purpose, Scope, Boundary, Partition

- Purpose of the count: to provide an answer to a business question
- Counting Scope: it defines the set of NFR to be included the SNAP count
- Boundary: it is a conceptual interface between the software under study and its users. Boundaries are common to FPA and SNAP
- Boundary Rules:
  - Defines what is external to the application
  - Indicates the border between the software being measured and the user
  - Acts as a "membrane" through which data processed by transactions pass into and out of the application
  - Is dependent on the user"s external business view of the application; it is independent of non-functional and/or implementation considerations
  - The Logical Application Boundaries need to be consistent between the FPA and SNAP processes
- Partition: a set of software functions within an application boundary that share homogeneous assessment criteria and values. A partition requires development effort, that may not be reflected when sizing the functional aspect of the project/product, using FPA

Categories & Sub-Categories

- Category: a group of components, processes or activities that are used in order to meet the non-functional requirement. Each category are divided into sub-categories
- Sub-Category: a component, a process or an activity executed within the project, to meet the non-functional requirement

Note: A non-functional process may have to execute more than one sub-category to meet the non-functional requirement.

List of categories and subcategories

1. DATA OPERATIONS (5)
1.1. Data Entry Validations
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

2. INTERFACE DESIGN (4)
2.1. User Interfaces
2.2. Help Methods
2.3. Multiple Input Methods
2.4. Multiple Output Methods

3. TECHNICAL ENVIRONMENT (3)
3.1. Multiple Platforms
3.2. Database Technology
3.3. Batch Processes

4. ARCHITECTURE (2)
4.1. Component based software
4.2. Multiple Input / Output interfaces

- **The SNAP 'magic number'**: **5-4-3-2**

SNAP Counting Unit (SCU)

- 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.
- An SCU may contain both functional and non-functional characteristics. In these cases, sizing of the elementary process will be performed for its functional sizing using function point analysis, and for its non-functional sizing, using SNAP.

TIP: Use the following order: 1. Identify the non-functional requirement. 2. Identify the SNAP subcategory/ies. 3. Within the subcategory, identify the SCU

Category 1: Data Operations

The **Data Operations** Category relates to how data is processed within the SCU to meet the non-functional requirements in the Application

1.1 Data Entry Validation

- Definition: Operations that are taken either to allow only certified (predefined) data or to prevent the acceptance of uncertified data
- SCU: The Elementary process
  - Nesting Level(s) The number of conditional validations (IF-Else combo/"While" loop/"For" loop or any other validation blocks) in the longest chain of validation
- Complexity Parameters:
  - Nesting level complexity (# of nesting levels in the longest chain of validation)
  - Number of DETs used for validation
- SP Calculation:

	Nesting Level Complexity - # of nesting levels in the longest chain of validation		
	Low	**Average**	**High**
DETs	**1-2**	**3-5**	**6+**
SP =	2* #DETs	3* #DETs	4* #DETs

- Examples: validations using code data or logical checks

1.2 Logical and Mathematical Operations

- Definition: Extensive logical decisions, Boolean operations, and extensive mathematical operations applied on the process
- SCU: The Elementary process
- Complexity Parameters:
  - FTR complexity of the logical table to be accessed
  - Processing logic type of EPs (logical / mathematical)
  - Number of DETs
- SP Calculation:

	FTR Complexity Level		
FTRs	**0-3 FTRs**	**4-9 FTRs**	**10+ FTRs**
Complexity	**Low**	Average	**High**

EP Type	Main EP's purpose
Logical	Decision making or evaluating a condition using data that exist in one or more logical files (internal and / or external) Example: Exception processing
Mathematical	Transformation of data and / or use of control information that exist in one or more logical files (internal and / or external), which is used for an extensive mathematical operation. Example: Complex tax calculation

Note: When the main purpose cannot be clearly identified, select "Logical"

		Complexity Level		
		Low	**Average**	**High**
EP Type: Logical	SP=	4* #DETs	6* #DETs	10* #DETs
EP Type: Mathematical	SP=	3* #DETs	4* #DETs	7* #DETs

1.3 Data Formatting

- Definition: any change in a transaction that deals with structure, format, or administrative information not directly relevant to functionality that is seen by the user
- SCU: The Elementary Process
- Complexity Parameters:
  - Transformation complexity (Low / Average / High)
    Low: Data type conversions or simple formatting such as byte padding, or data substitution, using a maximum of 2 operators (Celsius to Fahrenheit, Single Integer to Double Integer)
    Average: Involves encryption / decryption which is a characteristic of the application and applies to almost all processes, which is provided through a library -API interface
    High: Involves local Encryption/Decryption.
  - Number of DETs transformed
- SP Calculation:

	Transformation complexity		
	Low	**Average**	**High**
SP =	2* #DETs	3* #DETs	5* #DETs

- Examples (Complex): Enabling Multi-Lingual support for an application by using Code Data

1.4 Internal Data Movements

- Definition: Data Movement process from one partition to another within application Boundary with specific data handling. Data handling may include data formatting, logical /mathematical operations or Reference Data Maintenance
- SCU: The elementary process within the application boundary, which crosses partition.
  - Note: If an elementary process crosses more than one partition, use the formula below per each partition crossing.
- Complexity Parameters:
  - Number of DETs transferred into and out of the partition, in which data is processed and / or maintained
  - Number of FTRs either read or updated by the elementary process
- SP Calculation:

	Complexity Level		
	Low	**Average**	**High**
FTRs	**0-3 FTRs**	**4-9 FTRs**	**10+ FTRs**
SP =	4* #DETs	6* #DETs	10* #DETs

1.5 Delivering Added Value to Users by Data Configuration

- Definition: additional unique business value to users that is provided by adding, changing or deleting reference data/ code data information from the database or data storage with no change in software code or the database structure.
- SCU: The Elementary Process per logical file
- Complexity Parameters:
  - Number of unique attributes involved in the elementary process, that are added / modified / deleted
  - Number of Records configured
  - Attribute: An independent parameter that has a unique business meaning and contains a set of different values
  - Record: One row in a logical file
  - A Logical File: A user recognizable group of logically related data or control information
- SP Calculation:

	Complexity Level		
	Low	**Average**	**High**
	1-10 records	**11-29 records**	**30+ records**
SP =	6* #attributes	8* #attributes	12* #attributes

Category 2: Interface Design

The **Interface Design** Category relates to the end user experience. This category assesses the design of UI processes and methods that allow the user to interface with the application

2.1 User Interface

- Definition: 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 (such as usability, ease of learning, attractiveness, accessibility)
- SCU: the Set of screens as defined by the elementary process
- Terms: UI element, UI element properties (www.w3.org/TR/CSS2/ui.html), UI element set
- Complexity Parameters:
  - The sum of the # of unique properties configured for each UI element in the SCU.
  - Number of unique UI elements impacted
- SP Calculation:

	UI Type Complexity		
	Low	**Average**	**High**
# of Properties added or configured:	**<10**	**10-15**	**16+**
SP =	2* #unique UI elements	3* #unique UI elements	4* #unique UI elements

2.2 Help Methods

- Definition: Information provided to the users that explains how the software provides its functionality or other supportive information provided to users
- SCU: The Assessed Application
- Terms: Help Item, Context Help, Static webpage
- Complexity Parameters:
  - Help Type (a. User Manual; b. Online text; c. Context; d. Context + Online)
  - Number of Help items impacted
- SP Calculation:

Help Type	SP =
a. User Manual	1*(#help items)
b. Online Text	2*(#help items)
c. Context Help	2*(#help items)
d. Context + Online	3*(#help items)

2.3 Multiple Input Methods

- Definition: The ability of the application to provide its functionality while accepting multiple input methods
- SCU: The Elementary Process
- Terms: Input Methods: A technique or media type, which is used to deliver data into the assessed application, such as Bar Code reader, Fax, PDF, Office document, screen, voice message, SMS, Smart mobile device etc
- Complexity Parameters:
  - Number of DETs in the SCU
  - The number of additional input methods
- SP Calculation:

	Input Methods complexity		
	Low	**Average**	**High**
	1-4 DET's	**5-15 DET's**	**16+ DET's**
SP =	3* # additional input methods	4* # additional input methods	6* # additional input methods

2.4 Multiple Output Methods

- Definition: The ability of the application to provide its functionality while using multiple output methods
- SCU: The Elementary Process
- Terms: Output Methods: A technique or media type, which is used to deliver data from the assessed application, such as Fax, PDF, Office document, screen, voice message, SMS etc.
- Complexity Parameters:
  - Number of DETs in the SCU
  - The number of additional output methods
- SP Calculation:

	Output Methods complexity		
	Low	**Average**	**High**
	1-5 DET's	**6-19 DET's**	**20+ DET's**
SP =	3* # additional output methods	4* # additional output methods	6* # additional output methods

Category 3: Technical Environment

The **Technical Environment** Category relates to aspects of the environment where the application resides. It 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

- Definition: Operations that are provided to support the ability of the software to work on more than one platform (computing; software; hardware)
- SCU: The Elementary Process
- Complexity Parameters:
  - Nature of the platform(s)
  - Number of platforms to operate
- SP Calculation:

	2 platforms	**3 platforms**	**4+ platforms**
Category 1 – Software platforms (same family)	SP=20	SP=30	SP=40
Category 2 – Software Platforms (different family)	SP=40	SP=60	SP=80
Category 3 – Software Platforms (different browsers)	SP=10	SP=20	SP=30
Category 4 – H/W platforms (Real Time embedded systems)	SP=TBD	SP=TBD	SP=TBD
Category 5 – H/W platforms (Non Real Time embedded systems)	SP=TBD	SP=TBD	SP=TBD
Category 6 – Combination of H/W and S/W; Non-real time embedded systems	SP=TBD	SP=TBD	SP=TBD

3.2 Database Technology

- Definition: Features and operations that are added to the database or to the statements to read / write data to and from the database to deliver non-functional requirements without affecting the functionality that is provided
- SCU: The Elementary Process
- Terms: Database Changes
- Complexity Parameters:
  - Logical File Complexity
  - Number of database-related changes
- SP Calculation:

	Logical File complexity:		
	1-19 DET's	**20-50 DET's**	**51+ DET's**
1 RET	Low	Low	Average
2-5 RET's	Low	Average	High
6+ RET's	Average	High	High

	FTR Complexity Factor		
	Low	**Average**	**High**
SP=	6* #changes	9* #changes	12* #changes

3.3 Batch Processes

- Definition: Batch jobs that are not considered as functional requirements (they do not qualify as a transactional function) can be considered in SNAP. This sub-category allows for the sizing of batch processes which are triggered within the boundary of the application, not resulting in any data crossing the boundary.
- SCU: User identified batch job
  - Note: When several batch jobs are automated (run always as a whole) and only the end result is user identifiable, count these batch jobs as an individual SCU
- Complexity Parameters:
  - Number of DETs processed by the job
  - Number of FTRs either read or updated by the job
- SP Calculation:

	Complexity Level		
	Low	**Average**	**High**
	1-3 FTR's	**4-9 FTR's**	**10+ FTR's**
SP=	4* #DETs	6* #DETs	10* #DETs

Category 4: Architecture

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

4.1 Component Based Software

- Definition: Pieces of software used within the boundary of the assessed application to integrate with previously-existing software or to build components in the system
- SCU: The Elementary Process
- Terms: a software component
- Complexity Parameters:
  - 3rd-partycomponent or in-house reuse
  - Number of unique components involved in the EP
- SP Calculation:

Type	SP Calculation
In-house components	SP=3*(#unique component)
3rd-party components	SP=4*(# unique component)

4.2 Multiple Input / Output Interfaces

Definition: Applications required supporting multiple input and output interfaces (user files with the same format) are covered in this subcategory. For example: due to a growing number of users and volume of data over a period of time.

Adding more input/output interfaces without changing the functionality is not considered functional change and hence such changes are not sized by FP. This sub-category should be used to size such changes in an application.

- SCU: The Elementary Process
- Complexity Parameters:
  - Number of DETs in the SCU
  - Number of additional input and output interface
- SP Calculation:
  - Count the number of additional input and output interfaces.
  - When an interface is used for both input and output, count it once as an input and once as an output

	Complexity Level		
	Low	**Average**	**High**
DETs	**1-5**	6-20	**21+**
SP=	3* Additional # of Interfaces	4* Additional # of Interfaces	6* Additional # of Interfaces

SNAP Materials (IFPUG website)

- APM (Assessment Process Manual): IFPUG Online store
- SNAP Counting Template: http://goo.gl/qXlpV (IFPUG Members)

Formula Approach - Steps

1. For each requirement, identify the categories and sub-categories that are associated with the requirement
2. For each of the sub-categories, identify the SCUs
3. Determine the non-functional size (SNAP Points -SP) for each SCU within the sub-category, by using the equation or the table for the sub-categories
4. Determine the SP for a specific project or application by using the formula for the project type in question.

Formulas – DSP, ESP, ASPA

- Development: *DSP = ADD*
- Enhancement: *ESP = ADD + CHGA + DEL*
- Application after Enhancement:
  ASPA = ASPB + (ADD + CHGA) – (CHGB + DEL)

Legend

- **ADD**: size of the NFRs being delivered by the development project or added by the enhancement project
- **ESP**: The enhancement project's SNAP size
- **ASPA**: Application SNAP Points count *after* the enhancement project
- **ASPB**: Application SNAP Points count *before* the enhancement project
- **CHGA**: size of the NFRs being changed by the enhancement project – as they are / will be after implementation
- **CHGB**: size of the NFRs being changed by the enhancement project – as they are / were before the project commenced
- **DEL**: size of the NFRs being deleted by the enhancement project