uTip	**Guidance from the Non-Functional Sizing Standards Committee on Topics Important to You**	SNAP Powered by IFPUG

Using Gantt Charts to Track SNAP Work Effort

uTip # 01 – (Version # 1.0, 06/18/2024)

uTips (Usage Tips) provide insight into potential uses of SNAP Points to support an organization's business needs. While uTips provide insight on usage opportunities, they do not provide detailed direction on the application of the IFPUG SNAP method in a particular situation. When necessary, the uTip maybe be followed by additional content on the topic providing specific how-to guidance. uTips are not rules, but interpretation and application of the rules, and provide guidance using a realistic example to explain the topic being covered.

This uTip is focused on describing the IFPUG SNAP method as it applies to compute non-functional productivities for estimation purposes. This uTip includes a series of examples but is not an exhaustive examination of the subject.

The initial issue for not using SNAP…

One of the main counterpoints that somebody could discuss when thinking to adopt SNAP [1] is: "*which are the SNAP productivities if I'd like to start using it at the estimation phase?*" Productivity is definable as the ratio between a quantity and its related effort to be produced. Thus, since values for "*nominal*' productivities [2] proposed also in external benchmarking repositories such as the ISBSG ones [3] consider the 100% of the "project" effort (thus including also the functional related one, not related of course to SNAP and NFRs), to start computing non-functional productivities, a split of the overall project effort is needed.

This iTip will propose a simple but effective way to do so, stimulating the usage of SNAP also for estimation purposes for the NFR-side of a software project.

Step #1: determine the NFR-related effort from your own projects.

A project is potentially composed by three types of requirements: A (*product* FURs, sizable with FPs), B (*product* NFRs, sizable with SNAP Points) and C (*project*-related activities), as in the "**ABC schema**" [2].

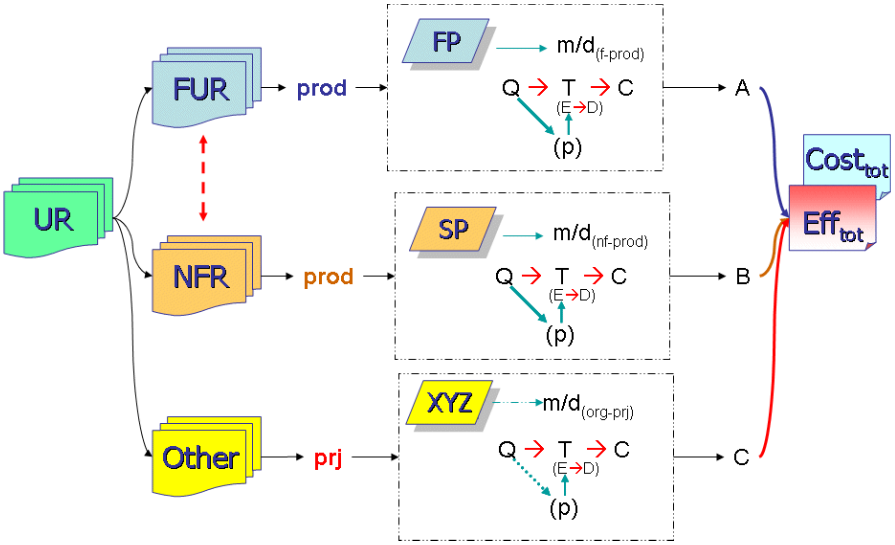
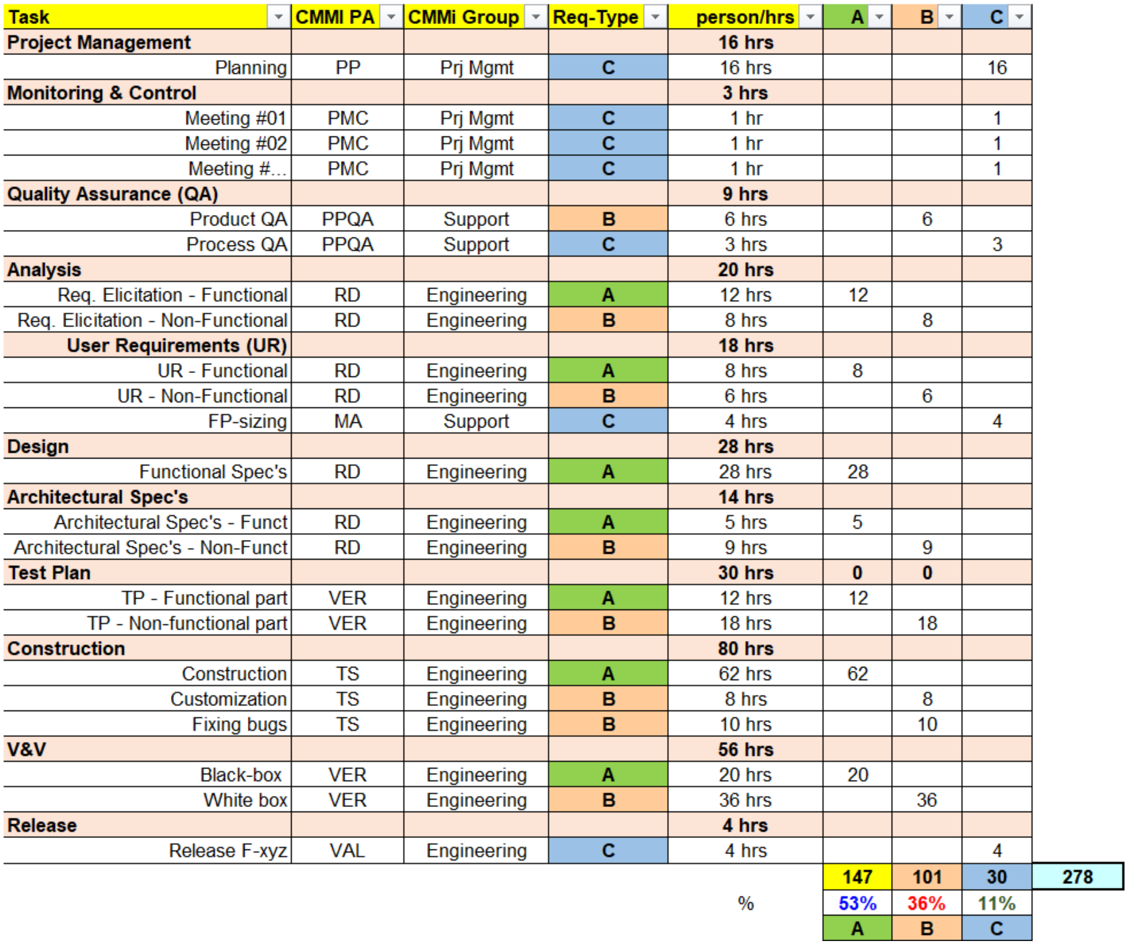


Figure 1 – The ABC schema

As in Figure 1, each stream must determine a quantity (Q), needed to determine the time (T) [to be applied as Effort (E) and Duration (D)] that will drive to determine the final costs (C) and prices. Productivity (P) is the ratio between Q/T. Aligned with basic metrological rules, quantities related to different entities and/or attributes cannot be summed up, while the related efforts and costs/prices can. The sum of the efforts coming from the three streams (A, B and C) delimits the "project scope," as well as from its related Gantt chart. In order to determine a "strict" non-functional productivity, we need to split the overall project effort into the three A/B/C parts.

For doing that, it's sufficient to take the Gantt chart for the project under investigation and assign to each leaf its related requirement type (A or B or C), as shown in Figure 2.



Task	CMMI PA	CMMi Group	Req-Type	person/hrs	A	B	C	
Project Management				**16 hrs**				
Planning	PP	Prj Mgmt	C	16 hrs			16	
Monitoring & Control				**3 hrs**				
Meeting #01	PMC	Prj Mgmt	C	1 hr			1	
Meeting #02	PMC	Prj Mgmt	C	1 hr			1	
Meeting #…	PMC	Prj Mgmt	C	1 hr			1	
Quality Assurance (QA)				**9 hrs**				
Product QA	PPQA	Support	B	6 hrs		6		
Process QA	PPQA	Support	C	3 hrs			3	
Analysis				**20 hrs**				
Req. Elicitation - Functional	RD	Engineering	A	12 hrs	12			
Req. Elicitation - Non-Functional	RD	Engineering	B	8 hrs		8		
User Requirements (UR)				**18 hrs**				
UR - Functional	RD	Engineering	A	8 hrs	8			
UR - Non-Functional	RD	Engineering	B	6 hrs		6		
FP-sizing	MA	Support	C	4 hrs			4	
Design				**28 hrs**				
Functional Spec's	RD	Engineering	A	28 hrs	28			
Architectural Spec's				**14 hrs**				
Architectural Spec's - Funct	RD	Engineering	A	5 hrs	5			
Architectural Spec's - Non-Funct	RD	Engineering	B	9 hrs		9		
Test Plan				**30 hrs**	**0**	**0**		
TP - Functional part	VER	Engineering	A	12 hrs	12			
TP - Non-functional part	VER	Engineering	B	18 hrs		18		
Construction				**80 hrs**				
Construction	TS	Engineering	A	62 hrs	62			
Customization	TS	Engineering	B	8 hrs		8		
Fixing bugs	TS	Engineering	B	10 hrs		10		
V&V				**56 hrs**				
Black-box	VER	Engineering	A	20 hrs	20			
White box	VER	Engineering	B	36 hrs		36		
Release				**4 hrs**				
Release F-xyz	VAL	Engineering	C	4 hrs			4	
					147	**101**	**30**	**278**
				%	**53%**	**36%**	**11%**	
					A	**B**	**C**	

Figure 2 – Splitting the project effort according to the ABC schema (and requirement-types)

In case you'd be in doubt because the leaf under investigation could include more than one requirement type, it'd mean that the leaf must still be decomposed into two or more parts. For instance: the task is about a generic writing of the test plan, but test cases could be about both functional and non-functional test cases. Thus, the next step will be to split the high-level task, not assignable to a unique ABC requirement type into two more detailed activities, assignable each to one of the ABC requirement-types.

In this way it's possible to compute the percentage of A (or B or C) related effort for a project. Since some ICT contracts could not apply SNAP, the calculation of the NFR-related effort is yet a first step for separating "apples and oranges" and allow to split the analysis by FURs and NFRs (not forgetting the third requirement type, the "C-type", related to each project-organizational related task, needed in the project scope, as the measurement activity itself).

Step #2: ok, I know the NFR-related effort, but what about SNAP productivities? Are they available or what should I do?

As well as it happened for the productivity values with FPs, you need to have two base measures (quantity, choosing a unit of measure; effort, choosing the time unit: person/hours, person/days, …) and then compute the derived measure (productivity). It needed time, gathering data from projects, and this is the story for several benchmarking initiatives (e.g. ISBSG, Leda, Galorath, QSM and other ones). The same needs to be done now with SNAP productivities. When the first SNAP APM was published, a bit later IFPUG and ISBSG created a specific SNAP Data Collection Questionnaire (DCQ) [6].

Thus, while waiting to have a sufficient number of projects for publishing the first SNAP repository by ISBSG, any organization can start to calculate its own number of SNAP Points (SPs) in a project and, knowing which is the NFR-related effort, deriving its own "strict" non-functional productivity, as in [2], formula (2).

Two levels of granularity can be achieved.

a) SNAP overall count: no matter the number of SNAP sub-categories applying to a SNAP count, the non-functional productivity would be derive dividing the overall number of SPs by the NFR-related effort.

b) SNAP sub-category level: in order to be more precise and compute a non-functional productivity for a specific topic (e.g. how much man/hours do we approximately need for reusing a component, according to SNAP §4.1 – Computer Based Software?), it'll be sufficient to divide the SP for that sub-category by its related, specific effort from the project Gantt chart. Of course, this (b) hypothesis shall be applicable only when an organization owns effort data specific to a certain SNAP sub-category.

Of course, the more granular the level of analysis, the better, with a reduced the "*cone of uncertainty*"' [5] for the next estimates, which will generate a lower relative error (RE) over time.

A practical example.

Let's assume to move from the effort data from Fig. 2 and to have measured 1100 SPs as the non-functional product size and 101 person/hours for the effort related to NFRs (the B-type, according to the ABC schema).

In this case the overall non-functional productivity for such project could be:

1100 SP / 101 p-hrs = 11 SP/p-hrs.

When a project team would go more in detail, mapping the single NFR-based activities to the SNAP sub-categories, it could be possible to derive a non-functional productivity level per each sub-category. For instance, the "customization" activity could be mapped with SNAP §1.5 (Data Configuration). In this case, assuming to configure 8 records with 5 attributes each, the number of SP would lead to compute for a low complexity range 30 SPs. Having a corresponding effort for 8 p-hrs, the non-functional productivity for SNAP §1.5 in this case would be:

30 SPs / 8 p-hrs = 3.75 SP / p-hrs.

In this second hypothesis, each project team could depict a sort of "*project non-functional profile'*," declaring which are the SNAP sub-categories applying (and which not), to allow a better comparability among projects of different types, as said before.

Productivities vary according to the project types…

When estimating effort and costs for a next project, it's logical to filter data and determine clusters of projects by similar characteristics It's the logical "divide-and-conquer" (from the Latin, *divide-et-impera*) principles for do not mix "*apples and oranges*" together into a unique, mixed, group of generic fruits.

It could seem trivial, but a web app project has different characteristics and A/B/C effort distribution than a DWH/BI project and so on, generating different nominal productivities. For achieving such classification by project types, the application of the **CHAR method** from the ISO/IEC 14143-5 standard [4] could be useful: it helps in clustering a set of projects by groups.

Of course, the ABC effort percentage triples of values shall be updated on a regular basis over time.

Some conclusions and next steps.

Estimation, as McConnel said in the title of his known book [5], is often a “black art” but measurement is the key for reducing the uncertainty and lowering the risks for facing “*scope creeps.*” Even if projects would not currently consider in detail NFR-based effort, it’s possible to derive it using a basic project management tool such as the Gantt chart. Whatever productivity level comes from the ratio of a quantity over a working effort, thus without one of the two basic measures, any productivity value could be computed. Consequently, not computing any productivity value, in the near future it couldn’t be possible to approximately estimate any project effort because the quantity would have been missed from the beginning…and it’s the main pitfall in Agile projects when assuming “velocity” and “story points” as absolute, quantitative units and values, while they are not. We need real, quantitative unit of measures. FPs and SPs can help in our journey to realize better estimates.

Next uTips will investigate more and more the added value that the NFR-side of the story (and SNAP) can bring into a valuable management of projects.

References

[1] IFPUG, SNAP APM v2.4, Non-Functional Standard Sizing Committee (NFSSC), September 2017

[2] Buglione L., The Next Frontier: Measuring and Evaluating the Non-Functional Productivity, IFPUG MetricViews, August 2012, URL: https://tinyurl.com/5feyurrv

[3] ISBSG, Development & Enhancement (D&E) repository, URL: https://www.isbsg.org/data-subscription-2/

[4] ISO/IEC, IS 14143-5:2004 (R2019), Information technology — Software measurement — Functional size measurement — Part 5: Determination of functional

domains for use with functional size measurement, 2019, URL: https://www.iso.org/standard/39986.html

[5] McConnell S., Software Estimation: Demystifying the Black Art, Microsoft Press, 2006, URL: https://tinyurl.com/muwyevx3

[6] ISBSG, SNAP Data Collection Questionnaire (DCQ), URL: https://www.isbsg.org/submit-data/

IFPUG offers uTips at no charge to the international Function Point community to stimulate the further promulgation and consistent application of the IFPUG SNAP Method. IFPUG would appreciate if you or your organization would support IFPUG in its mission by becoming a member. For further information please visit www.ifpug.org or send an email to ifpug@ifpug.org. IFPUG thanks you for your support.