uTip

Guidance from the Functional Sizing Standards Committee on topics important to you

Project Testing

uTip # 2 – (version # 1.0 *7/6/2014)*

uTips provide insight into potential uses of function 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 FPA 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 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 FPA method as it applies to testing functional elements within a project. This uTip includes functional sizing of testing elements but is not an exhaustive examination of the subject. Please refer to the SNAP manual for non-functional user testing requirements.

Introduction

The Counting Practices Manual (CPM) does not directly provide a type of count specifically addressing project testing. The types of counts specifically addressed in the CPM are the development function point count, application function point count, and enhancement function point count. However, when the purpose and scope are applied properly, the functional size of a testing engagement can be discretely determined for any engagement requiring testing of functional components of an application or project.

Sizing Functionality to be Tested within a Development Project

For a development project, the size of the functionality to be tested will be a subset of the overall development project functionality (up to the entire size of the development project). Functionality to be tested can include development and conversion functionality. The key to sizing testing is to provide a purpose and scope to the count that uniquely identifies the functionality to be tested.

Example:

A functional user requirement is "*As a user of the HR system, I want to add employee data so that I can register a new hire.*" The project needs to test this functionality.

Purpose: Identify the size of the development project components to be tested to demonstrate that the functional user requirement *"As a user of the HR system, I want to add employee data so that I can register a new hire."* has been met.

Scope: Only size the components related to the functional user requirement: *"As a user of the HR system, I want to add employee data so that I can register a new hire."*

The size of the functionality directly related to the functional user requirement would be identified and sized to provide the size of the testing. (Of course this is a simple application of the technique for illustration purposes.)

From a practical standpoint, it is likely that the entire project has been sized based on functional user requirements and identifying the components to be tested is largely an exercise in selecting which functionality should be included in the test size.

Sizing Functionality to be Tested within an Enhancement Project

With an enhancement project, it is often necessary to test existing functionality of an application that was not directly part of the enhancement project. Functionality to be tested can include functionality to be developed by the project, functionality to be changed by the project, functionality within the application which is unchanged by the project and conversion functionality. The key to sizing testing is to provide a purpose and scope to the count that uniquely identifies the functionality to be tested (from both the enhancement project and the application prior to enhancement).

Example:

A functional user requirement for an enhancement project is "As a user of the HR system, I want to include an optional secondary phone number when employee data is added so that I can contact the person via multiple numbers."

Enhancement Count Purpose: Identify the size of the enhancement project components to be tested to demonstrate that the functional user requirement "*As a user of the HR system, I want to include an optional secondary phone number when employee data is added so that I can contact the person via multiple numbers*" has been met.

Enhancement Count Scope: Only size the components related to the functional user requirement *"As a user of the HR system, I want to include an optional secondary phone number when employee data is added so that I can contact the person via multiple numbers."*

Application Count Purpose: Identify the size of the existing components that are either necessary to set up conditions to test the project functionality, produce work products that consume data downstream from functionality related to the requirement or are needed to demonstrate that existing functionality was not adversely affected by the project changes for the functional user requirement *"As a user of the HR system, I want to include an optional secondary phone number when employee data is added so that I can contact the person via multiple numbers."*

Application Count Scope: Only size the components necessary to demonstrate that the functional user requirement "*As a user of the HR system, I want to include an optional secondary phone number when employee data is added so that I can contact the person via multiple numbers*" has been met without adversely affecting other application components.

From a practical standpoint, it is likely that the application and entire project have been sized based on functional user requirements (often within a common counting mechanism such as a spreadsheet or counting tool) and identifying the components to be tested is largely an exercise in selecting which functionality should be included in the test.

Sizing Functionality to be Tested in a Project that is Not a Development Project nor an Enhancement Project.

Currently the official IFPUG function point terminology only addresses new or changed functional requirements for an application. However, any testing of functional components can be sized based on the application function point count by using an appropriate purpose and scope – even if the project is not a development project or an enhancement project.

Example:

The user is performing an upgrade to the web server software and needs to test an application running on that web server to ensure that user functions continue to function the same way they did prior to the upgrade.

Application Count Purpose: Identify the size of the existing components to be tested to demonstrate that functional user requirements continue to behave in the same manner after the upgrade to the server software.

Application Count Scope: Only size the components necessary to demonstrate that the application continues to operate the same before and after the server software upgrade.

From a practical standpoint, it is likely that the application has been sized based on functional user requirements (often within a common counting mechanism such as a spreadsheet or counting tool) and identifying the components to be tested is largely an exercise in selecting which functionality should be included in the test.

Summary

Function points are an effective way of identifying the functional components to be tested for development projects, enhancement projects, or any situation where it is useful to identify the testing of functional components of an application or project. While this paper addresses testing components, the concepts are transferable to other situations where aggregate functionality does not meet the criteria for a development function point count, application function point count, or enhancement function point count. Make sure you document your purpose and scope wisely.

Further Reading

IFPUG Counting Practices Manual, Part 1, Section 5.3 – Determine the counting scope and boundary and identify Functional User Requirements.

IFPUG Counting Practices Manual, Part 2, Chapter 5 – Determine the Counting Scope and Boundary and Identify Functional User Requirements.

Appendix: The Rules and definitions

Section 5.3 of the CPM addresses the rules for setting up the function point count:

5.3 Determine the counting scope and boundary and identify Functional User Requirements

a) *identify the purpose of the count,*
b) *b) identify the type of count, based on the purpose, as one of:*

   1) a development project function point count;

   2) an application function point count;

   3) an enhancement project function point count,

c) *determine the counting scope, based on the purpose and type of count,*
d) *determine the boundary of each application within the counting scope, based on the user view, not on technical considerations*
e) *the user requirements may include a mixture of both functional and non-functional requirements; identify which requirements are functional, and exclude the non-functional requirements.*

The CPM also provides definitions of key terminology related to the various types of projects:

development project: *project to develop and deliver the first release of a software application.*

development project function point count: *activity of applying this International Standard to measure the functional size of a development project.*

development project functional size: *measure of the functionality provided to the users with the first release of the software, as measured by the development project function point count. NOTE The functional size of a development project may include the size of conversion functionality.*

enhancement project: *project to develop and deliver adaptive maintenance.*

enhancement project function point count: *activity of applying this International Standard to measure of the functional size of an enhancement project.*

enhancement project functional size: *measure of the functionality added, changed or deleted at the completion of an enhancement project, as measured by the enhancement project function point count. NOTE The functional size of an enhancement project can include the size of conversion functionality.*

application function point count: *activity of applying this International Standard to measure the functional size of an application.*

conversion functionality: *transactional or data functions provided to convert data and/or provide other user specified conversion requirements.*

IFPUG offers uTips at no charge to the international function point community to stimulate the further promulgation and consistent application of the IFPUG FPA 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.