Université du Québec à Montréal
Software Engineering Management Research Laboratory
and
Software Engineering Laboratory in Applied Metrics (SELAM)

Prepared by

Jean-Marc Desharnais
Denis St-Pierre
Marcela Maya
Alain Abran

In collaboration with

Nortel
Bell Canada
Hydro-Québec
JECS System Research

FULL FUNCTION POINTS:
COUNTING PRACTICES MANUAL
PROCEDURES AND COUNTING RULES

November 1997

.

Copyright 1997. All Rights Reserved. Software Engineering Management Research Laboratory of the Université du Québec à Montréal and Software Engineering Laboratory in Applied Metrics (SELAM). Permission to copy all or part of this document for commercial use is granted provided that the title of the publication, its date and its authors are cited. To copy in any other manner requires specific permission.

TABLE OF CONTENTS

Introduction

A previous report (Full Function Points: Counting Practices Manual, Technical Report 1997-04) outlined the origins of Full Function Points (FFP), explained the rules and procedures, point assignments and provided sample rules and procedures. This document includes a subset of the technical report in order to enable a wider dissemination of FFP with a minimum of legal and business constraints. For instance, the Copyright notice includes the following sentence: "Permission to copy all or part of this document for commercial use is granted…".

As in the Function Point Counting Practices Manual, we will not reiterate here the identification procedures and counting rules already addressed by IFPUG, and which also form part of the FFP technique, since the reader can easily find them in the Function Point Counting Practice Manual. The FFP Counting Practice Manual is available in PDF format at the following addresses:

http://www.lmagl.qc.ca/rtreport.pdf
or :
http://saturne.info.uqam.ca/Labo_Recherche/Lrgl/publi/treports/LRGL-1997-015.pdf

As previously indicated, you may copy all or part of these rules and procedures, even for commercial benefit, as long as the authors, the title and the date are cited in the following manner: Desharnais J.-M., St-Pierre D., Maya M., Abran A., Full Function Point: Counting Practices Manual , Procedures and Counting Rules, November, 1997.

1.0 FFP Counting procedure and rules

The following diagram outlines the current overall Function Point Analysis (FPA) counting procedure[1].

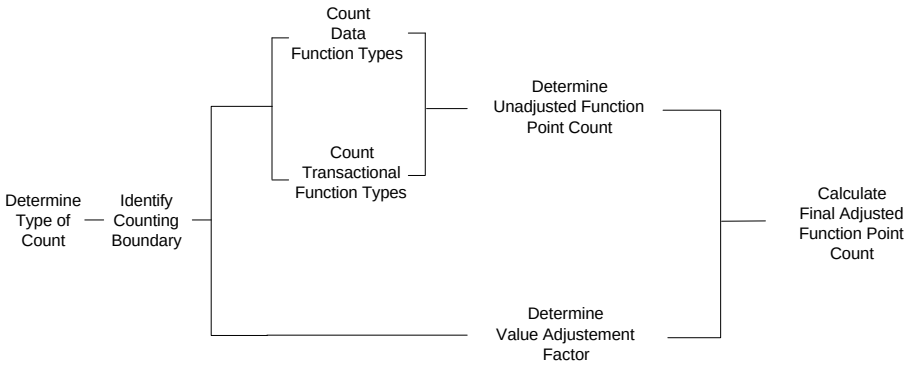

Figure 1 - FPA Counting Procedure Diagram

This procedure includes the following steps (IFPUG, 1994):

1. Determine the type of function point count
2. Identify the counting boundary
3. Determine the unadjusted function point count
   a. Count data function types
   b. Count transactional function types
4. Determine the value adjustment factor
5. Calculate the final adjusted function point calculation

For FFP, steps 1, 2, 4 and 5 are exactly the same as in FPA. Step 3 is divided into Management Function Types and Control Function Types, as shown in Figure 2.

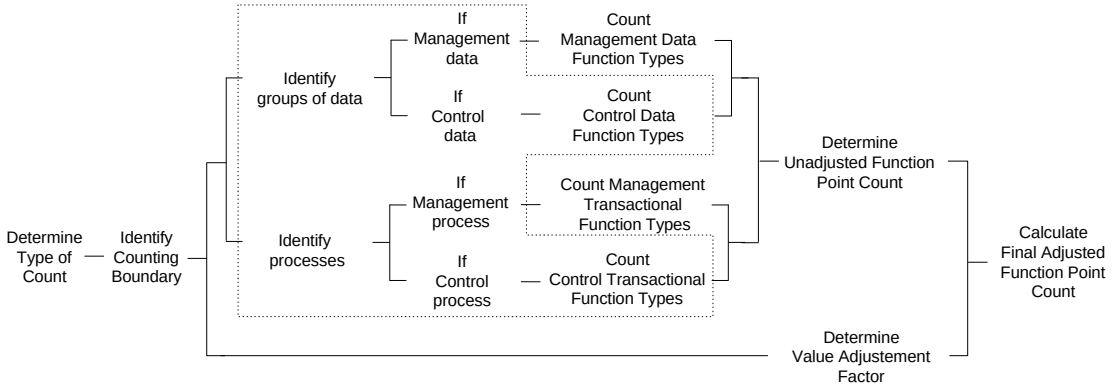

Figure 2 - FFP Counting Procedure Diagram

[1] Function Point Counting Practices Manual, release 4.0, IFPUG, 1994.

The portion of the procedure contained within the dotted line corresponds to the new FFP steps and is covered in this section. The remaining portion is covered by the Function Point Counting Practices Manual (IFPUG, 1994).

2.0 Identify groups of data

This step consists in identifying the groups of data that could represent the functionality provided to the user by the application being measured. Once the groups of data are identified, the definitions and rules associated with these function types are applied to determine whether the identified groups of data are counted as FFP function types.

2.1 Definitions

Group of data: Data identified and grouped together based on the functional perspective.

Management data: Data used by the application to support users in managing information, particularly business and administrative information.

Control data: Data used by the application to control, directly or indirectly, the behaviour of an application or a mechanical device.

2.2 Identification procedure

The procedure to identify group of data candidates is the following:

1. Look for groups of data identifiable from a functional perspective.
2. Determine if the group of data is a management group of data or a control group of data using the previous definitions. For management groups of data, existing ILF and EIF procedures and rules (IFPUG, 1994) should be applied. For Control Groups of Data, the following procedures and rules should be applied.

   Note: All single occurrence control data updated by the application are merged into one unique group of data. The same goes for read-only control data.

3.0 Count control data function types

3.1 Definitions

Updated Control Group (UCG): A UCG is a group of control data updated by the application being counted. It is identified from a functional perspective. The control data live for more than one transaction.

Read-only Control Group (RCG): An RCG is a group of control data used, but not updated, by the application being counted. It is identified from a functional perspective. The control data live for more than one transaction.

Functional perspective: Point of view of the functionality delivered by the application; it excludes technical and implementation considerations.

Transaction: All processing associated with an occurrence of an external trigger.

3.2 Counting procedure

For each group of data identified in the previous step as a group of control data:

1. Determine if the group of control data is a UCG or an RCG using the definitions and rules.
2. Determine the UCG or RCG contribution (point assignment) to the unadjusted function point count.

3.3 Identification rules

UCG identification rules:

- ❒ The group is either a logically related group of data or a single occurrence group of data.
- ❒ The group of data is **updated** within the application boundary.
- ❒ The group of data lives for more than one transaction.
- ❒ The group of data identified has not been counted as an RCG, ILF or EIF for the application.

All the previous counting rules must be applied from a functional perspective and they are all mandatory for the identification of a UCG

RCG identification rules:

- ❒ The group is either a logically related group of data or a single occurrence group of data.
- ❒ The group of data is **not updated** by the application being counted.
- ❒ The group of data is referenced by the application being counted.
- ❒ The group of data lives for more than one transaction.

- ❒ The group of data has not been counted as a UCG, ILF or EIF for the application.

All the previous counting rules must be applied from a functional perspective and they are all mandatory for the identification of an RCG.

3.4 Point Assignment

The number of points assigned to UCGs and RCGs depends on the kind of control group of data (single or multiple occurrences). Since multiple occurrence groups of data have the same structure as ILFs and EIFs in FPA, they are counted in exactly the same way as these two FPA function types, that is, using their number of DETs and RETs and the corresponding complexity matrix.

For single occurrence groups of data, the number of points depends only on DETs. Once the number of DETs is determined using the same rules as for ILFs and EIFs, the number of points is calculated using the following formulas:

UCG: ((number of DETs / 5) + 5)
RCG: (number of DETs /5)

These formulas are designed to keep the size of single occurrence groups of data as aligned as possible with the size of ILFs and EIFs of FPA.

A single occurrence UCG comprises all single control updated values (from a functional perspective) of the application being measured. Since it contains all single values of the application, there can be only one of them in an application. Consequently, an application can have more than one multiple occurrence UCG, but only one single occurrence UCG. The same goes for single occurrence RCGs.

In typical real-time applications, the number of such single values varies from a few up to hundreds. That is why a formula is used rather than a 3-level table like the standard FPA technique. It allows FFP to consider a large range of single occurrence groups of data.

4.0 Identify processes

Once the data function points (management and control) have been counted, the transactional function types are identified. Transactional function types represent the functionality provided to the user for the processing of data by an application. Therefore, to identify transactional function types we have to identify the processes of the application first.

4.1 Definition

Control process: Process that controls, directly or indirectly, the behaviour of an application or a mechanical device

4.2 Identification procedure

The procedure for identifying processes is the following:

1. Look for the different processes of the application from a functional perspective.
2. Determine if the process is a management process or a control process using the following definitions:

   Management process: Process the purpose of which is to support the user in managing information, particularly business and administrative information**.**

   Control process: Process that controls, directly or indirectly, the behaviour of an application or a mechanical device.

3. If the process is a control process, apply the definition and rules of the four new control transactional function types. If the process is a management process, apply the definition and rules of the current FPA transactional functions.

5.0 Count control transactional function types

5.1 Definitions

External Control Entry (ECE): An ECE is a unique sub-process. It is identified from a functional perspective. An ECE processes control data coming from outside the application's boundary. It is the lowest level of decomposition of a process acting on one group of data. Consequently, if a process enters two groups of data, there are at least 2 ECEs. ECEs exclude the updating of data, a functionality that is covered by another Control Function Type (Internal Control Write).

External Control Exit (ECX): An ECX is a unique sub-process. It is identified from a functional perspective. An ECX processes control data going outside the application boundary. It is the lowest level of decomposition of a process acting on one group of data. Consequently, if a process exits two groups of data, there are at least 2 ECXs. ECXs exclude the reading of data, a functionality that is covered by another Control Function Type (Internal Control Read).

Internal Control Read (ICR): An ICR is a unique sub-process. It is identified from a functional perspective. An ICR reads control data. It is the lowest level of decomposition of a process acting on one group of data. Consequently, if a process reads two groups of data, there are at least 2 ICRs.

Internal Control Write (ICW): An ICW is a unique sub-process. It is identified from a functional perspective. An ICW writes control data. It is the lowest level of decomposition of a process acting on one group of data. Consequently, if a process writes on two groups of data, there are at least 2 ICWs.

Functional perspective: Point of view of the functionality delivered by the application; it excludes technical and implementation considerations.

Control data: Data used by the application to control, directly or indirectly, the behaviour of an application or a mechanical device.

User: Human beings, applications or mechanical devices, which interact with the application, measured.

5.2 Counting procedure

Once a process has been identified as being a control process, the following steps must be performed:

- Identify all functional (not technical) sub-processes of the control process.
- Identify each sub-process as being an ECE, ECX, ICR or ICW according to the definitions and rules.
- Determine the ECE, ECX, ICR or ICW contribution (point assignment) to the unadjusted function point count.

Steps for identifying sub-processes:

1. According to the logical execution order of the sub-processes within the process, identify the first sub-process that receives, exits, reads or writes a group of control data.
2. Apply the relevant ECE, ECX, ICR or ICW set of rules.
3. Determine the ECE, ECX, ICR or ICW contribution (point assignment) to the unadjusted function point count.
4. Again according to execution order, identify the next sub-process that enters, exits, reads or writes a group of control data. There might be more than one "next sub-process" (e.g. an "IF" statement with two options). In this case, all paths have to be explored if there is potential for new sub-processes to be identified.
5. Repeat steps 2 to 4 until all sub-processes of the processes are identified.
6. At the end of the cycle, remove all the duplicated sub-processes (same processing[2] and same DETs).

   Note: If the same sub-process is associated with different control processes, it can be counted more than once.

5.3 Identification rules

ECE identification rules:

- ❒ The sub-process receives a group of control data from outside the application boundary.
- ❒ The sub-process receives only one group of data. If more than one different group of data is received, count one ECE for each group of data.
- ❒ The sub-process does not exit, read or write data.
- ❒ The sub-process is unique, that is, the processing and data elements identified are different from other ECEs associated with the same process.

Note 1: Clock triggers are considered external. Therefore, an event that takes place every 3 seconds is counted as an ECE with 1 DET, for example. However, the process that generates the event periodically is ignored.

[2] Processing includes not only the entry, exit, reading or writing of data, but other types of processing as well (calculation, filtering, comparisons, etc,) associated with the identified sub-process.

Note 2: Unless a special process is necessary, reading internal time is not counted. For example, when a process writes a time stamp, no ICR is counted for obtaining the internal clock value.

All the previous counting rules must be applied from a functional perspective and they are all mandatory for the identification of an ECE.

ECX identification rules:

- ❒ The sub-process sends control data external to the application's boundary.
- ❒ The sub-process sends only one group of data. If more than one different group of data is sent outside the application's boundary, count one ECX for each group of data.
- ❒ The sub-process does not receive, read or write data.
- ❒ The sub-process is unique, that is, the processing and data elements identified are different from other ECXs associated with the same process.

Note: All messages without user data (e.g. confirmation and error) are counted as one ECX. The number of DETs is the number of different types of messages.

All the previous counting rules must be applied from a functional perspective and they are all mandatory for the identification of an ECX.

ICR identification rules:

- ❒ The sub-process reads a group of control data.
- ❒ The sub-process reads only one group of data. If more than one different group of data is read, count one ICR for each group of data.
- ❒ The sub-process does not receive, exit or write data.
- ❒ The sub-process is unique, that is, the processing and data elements identified are different from other ICRs associated with the same process.

All the previous counting rules must be applied from a functional perspective and they are all mandatory for the identification of an ICR.

ICW identification rules:

- ❒ The sub-process writes a group of control data.
- ❒ The sub-process writes only one group of data. If more than one different group of data is written, count one ICW for each group of data.
- ❒ The sub-process does not receive, exit or read data.
- ❒ The sub-process is unique, that is, the processing and data elements identified are different from other ICWs associated with the same process.

All the previous counting rules must be applied from a functional perspective and they are all mandatory for the identification of an ICW.

5.4 Point Assignment

The number of points assigned to control transactional functions (ECE, ECX, ICW and ICR) depends on the number of DETs. The following rules apply when counting DETs:

For an **ECE and an ECX**:

- ❒ Count one DET for each unique user recognizable, nonrecursive field, that *crosses the boundary* of the application.

For an **ICR**:

- ❒ Count one DET for each unique user recognizable, nonrecursive field that *is read* from an ILF, EIF, UCG or RCG, including keys.

For an **ICW**:

- ❒ Count one DET for each unique user recognizable, nonrecursive field that *is written* in an ILF or UCG, including keys.

The number of points assigned to Control Transactional Functions Types (ECE, ECX, ICW and ICR) depends on the number of DETs[3]. Once the number of DETs is determined, the following table is used to translate DETs into points (Table 1):

DETs:	1 to 19 DETs	20 to 50 DETs	51 + DETs
Points:	1	2	3

Table 1: Control Transactional Function Type

These range boundaries (1 to 19, 20 to 50, 51+) were chosen in order to bring the size of Control Transactional Function Types in as close alignment as possible with FPA.

[3] DET: A unique user recognizable, nonrecursive field (IFPUG 94, glossary) that is either entered, exited, read or written depending the function type.

Conclusion

We are confident that this FFP measurement technique will expand the reach of applicability of FPA and increase its relevance to industry, mainly for real-time systems. The publication of this document, with a minimum of legal constraints, even for commercial use, will contribute to reach a larger public. We are open to collaborate with corporations and measurement associations like IFPUG to produce the next version of FFP

Glossary

The following definitions should be used for FFP, in addition to those contained in the IFPUG (IFPUG, 1994) glossary.

Control data: Data used by the application to control, directly or indirectly, the behaviour of an application or a mechanical device.

Control process: Process that controls, directly or indirectly[4], the behaviour of an application or a mechanical device.

Data function type: The functionality provided to the user to meet data requirements. Data function types are defined as UCG, RCG, ILF and EIF.

Functional perspective: Point of view of the functionality delivered by the application; it excludes technical and implementation considerations.

Group of data: Data identified and grouped together based on the functional perspective.

Management data: Data used by the application to support users in managing information, particularly business and administrative information.

Management process: Process the purpose of which is to support the user in managing information, particularly business and administrative information.

Multiple occurrence: Multiple occurrence groups of data can have more than one instance of the same type of record.

Process: A set of operations or activities which acts on inputs to produce a result.

Single occurrence: Single occurrence groups of data have one and only one instance of the record

Sub-process: In the context of this report, the smallest processing step identifiable from a functional perspective as either an entry, exit, read or write.

Transaction: All processing associated with an occurrence of an external trigger.

Transactional function type: The functionality provided to the user by an application to process data. Transactional function types are defined as ECE, ECX, ICR, ICW, EI, EO and EQ.

Trigger: An event that initiates a process from a functional perspective. The event comes from outside the application boundary. Clocks and timing events can be triggers.

Update: The ability to modify data.

User: Human beings, applications or mechanical devices which interact with the measured application.

[4] For example, a diagnostic process that reports to a human being is not in direct control of an application or mechanical device, but is a Control Process.