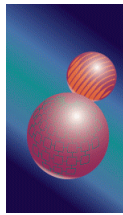


Myths, Mists and Other Cloaking
Devices –
How We Hide the Truth about Function
Point Analysis

Robyn Lawrie
CHARISMATEK Software Metrics
175 Dorcas Street
South Melbourne VIC 3205
Australia

Myths, Mists and Other Cloaking Devices – How We Hide the Truth about Function Point Analysis



Myths, Mists and Other Cloaking Devices – How We Hide the Truth about Function Point Analysis

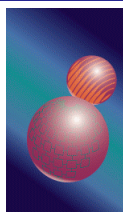
Robyn Lawrie
Robyn.Lawrie@charismatek.com
61 (0)3 9696-1255

CHARISMATEK Software Metrics
Publishers of Function Point WORKBENCH
www.charismatek.com
175 Dorcas Street, South Melbourne, VIC, 3205
Australia

**Aligning the Software Development Process
with Your Business Imperatives**

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 1



Agenda

1. Introduction
2. How We Make Function Point Analysis Inaccessible
3. Some Suggestions for the Future

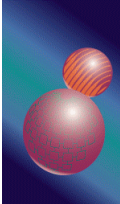


IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 2

Myths, Mists and Other Cloaking Devices

– How We Hide the Truth about Function Point Analysis




Functional Size What Is It?


Functional Size is a measure of software size that:

- ⇒ focuses on the **functionality** the software delivers to its **users**
- ⇒ is **independent** of how the software is built eg language, technology, methodology
- ⇒ is a **good predictor of effort** - and hence **cost** - to build and support the software
- ⇒ can be determined **early** in the development life cycle
- ⇒ is the only internationally used **objective** and **standardised** approach for software size measurement

Software Product

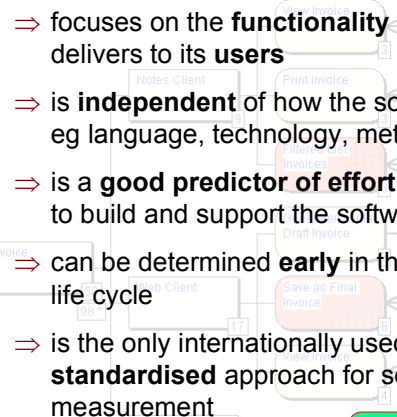


Software Project



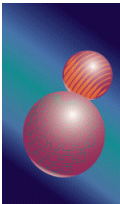
Function Point Analysis

Software Functional Size
650 Function Points



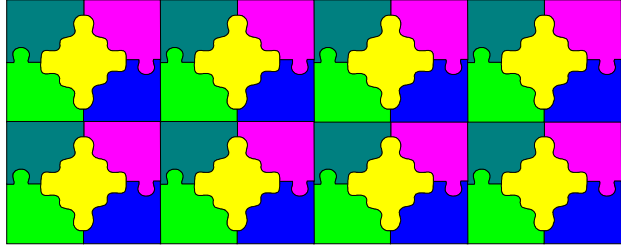
IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 3



Function Point Analysis Method for Sizing

* A software application is, in essence, a **set of interdependent functions**. When these functions are combined, they act together to provide a software application.




* **In order for the software to be properly sized, every unique function which contributes its functionality to the software must be identified and sized**

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 4

Myths, Mists and Other Cloaking Devices

– How We Hide the Truth about Function Point Analysis



There is a General Perception that Function Point Analysis is Hard

As a first step in our Project, we need to develop realistic estimates of cost and delivery schedule.


I suggest we use Function Points as our basis for estimation.

But Function Point Analysis needs an expert.

Let's use Use Case Points. They are easy!

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 5



There is a General Perception - Function Point Analysis is Hard

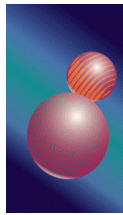
But Function Point Analysis is a standard method and there is extensive industry data to underpin our estimates.

But we already have Use Cases.

Let's use Use Case Points. They are easy!

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

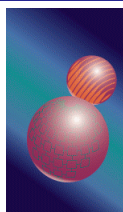
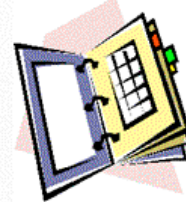
1 - 6



Function Point Analysis is “Not Hard”

I know it is “**not hard**” because:

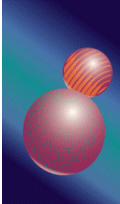
- * All the Rules can be summarised on a Reference Card
- * The technique is typically learned in a 2-3 day course
- * Compare this with the effort for learning other IT skills, e.g. a programming language or requirements analysis.



But We make Function Point Analysis Seem Hard

- * The ‘**weight**’ of the Counting Practices Manual and other supporting documents certainly adds to the perception that it is ‘**hard**’






We make Function Point Analysis the Province of the Specialist

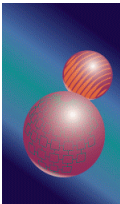
Should I count these tables as three different ILFs or one ILF with 3 RETs?

I would count one ILF for Employees with 2 RETs and one ILF for Contractors.

* The separation / isolation of counting from the Project Team reinforces the perception of 'the specialist'.



IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics 1 - 9




We are often 'Protectionist'

* The sizing specialist knows the sizing technique - but doesn't know the functionality to be delivered by the project or application.

* The analyst on the project team knows the functionality to be delivered - but doesn't know / has forgotten the sizing technique.

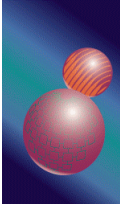
* We argue against the technique being used perhaps a little 'imperfectly' by the ad-hoc counter.



IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics 1 - 10


Myths, Mists and Other Cloaking Devices

– How We Hide the Truth about Function Point Analysis



We don't Share our Intelligence with those who could Benefit

We know Functional Sizing and related information can benefit a project in many ways, during the project e.g.

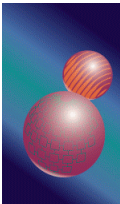


- * Feasibility Costing
- * Establishing a Business Case
- * Estimating Schedule and Cost based on Industry and/or In-House Benchmarks
- * Negotiating Scope
- * Establishing a Scope Baseline
- * Monitoring Change in Scope and Size
- * Measuring Delivery Progress
- * Benchmarking Achievements


WARNING!!
Function Point Counts
Do Not Open

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 11



We don't Share our Intelligence with those who could Benefit



Hooray!! We've got a PLAN!! The plan will save us!!

Huh! I don't think so. Given the size of software that you have to deliver, that would make you the most productive team ever!!!!

Project X-9 Plan of Action

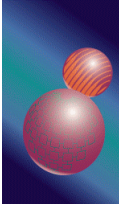
January 1997

The PLAN

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics


1 - 12

Myths, Mists and Other Cloaking Devices – How We Hide the Truth about Function Point Analysis



Our mysterious Language makes our counts Inaccessible

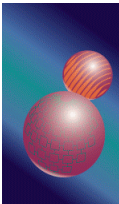
The percentage contribution of EI's is lower than Industry Average.



Type	Low	Avg	High	Total
EI	15 x 3 +	1 x 4 +	0 x 6 =	49
EO	1 x 4 +	12 x 5 +	1 x 7 =	71
EQ	2 x 3 +	13 x 4 +	0 x 6 =	58
ILF	6 x 7 +	1 x 10 +	1 x 15 =	67
EIF	2 x 5 +	1 x 7 +	0 x 10 =	17
Unadjusted FPs				262
VAF .96				
Adjusted FPs				252

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics


1 - 13



When we do share, the Language we use is often Inappropriate

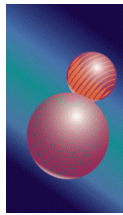
- * Put yourself in the position of your audience.
- * Forget the Function Types, the DETs, RETs, FTRs, etc.
E.g. For Establishing A Scope Baseline, just naming the functions contributing to the count will often do.

- * Feasibility Costing
- * Establishing a Business Case
- * Estimating Schedule and Cost Negotiating Scope
- * Establishing a Scope Baseline
- * Monitoring Change in Scope and Size
- * Measuring Delivery Progress
- * Benchmarking Achievements



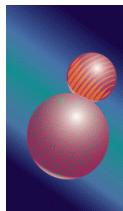
IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 14



When we do share, the Count is often in a format which makes it difficult to maximise Benefits from it

- * Can the completeness / coverage of the count be easily understood? Can the project team take ownership?
- * Can the count be easily used as a basis for discussion and agreement about project scope?
- * Can the count be sliced and diced?
- * Can solution strategies be explored against the count - e.g. purchased, developed externally, developed internally
- * Can the count be used to cross-validate estimate?

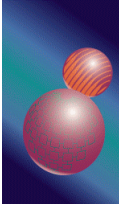


We sometimes Mislead our Audience about what we really Know

- * We often omit to indicate the Precision and Accuracy of our count - by providing the size as a range of Function Points or with confidence limits.

From early life cycle to software delivery, **Accuracy** will increase. **Precision** is limited by the information sources but is also a **choice** depending upon what the count is to be used for.

Precision - an indication of how finely something is measured
Accuracy - an indication of how close the measure is to reality

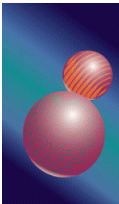



We sometimes Over-Engineer our Counts

- * We sometimes count to a finer granularity or level of precision than is appropriate for the purpose of the count and / or the information available.
- * This means a higher cost of counting.

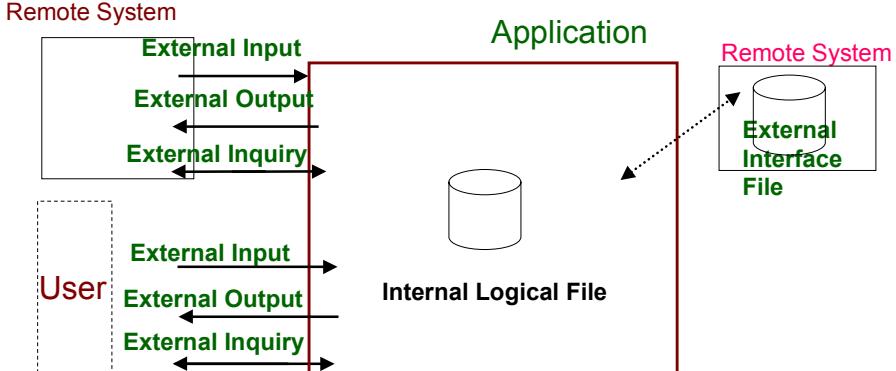
Because the project is very early in its life cycle, I have just captured the primary business view of the functionality.

Okay. So let's work on double that size for now and see how it fits in with our business case.



We don't have a very good 'Elevator' Description of the technique

- * 'Standard' description misleading and inadequate.



Remote System

Application

Remote System

User

External Input

External Output

External Inquiry

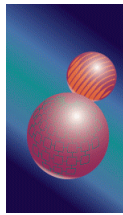
External Input

External Output

External Inquiry

Internal Logical File

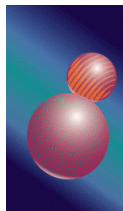
External Interface File



25 Years and Still Counting

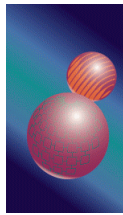
- * We must be doing something right
 - ◆ Function Point Analysis has come a long way and
 - ◆ Has withstood the test of time
- * But to consolidate our position and ensure more birthdays, as the software metrics professionals,

IT IS UP TO US



To become Mainstream, Function Point Analysis must be

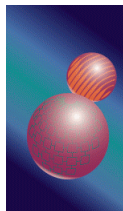
- * Perceived to be easy or, at least, “not hard”
- * Wanted and accessible - to benefit the project today
- * Propagated, or at least treated with respect, by Academia



Change the Perception of the Method to “Not Hard”

- * Review the body of knowledge and ruthlessly renovate, reduce, modernise
- * Make the method sensible and more intuitive
- * Do ‘peer reviews’ of count rather than ‘audits’ and ‘validations’
- * Allow ‘imperfect’ counts. The count may well be good enough for the purpose.

Perception
is
Everything



Make the Method and Results Accessible

- * **Socialise** the counts - with or without the Function Points showing
- * Encourage the project teams to take **ownership**
- * **Share** knowledge and intelligence



Allow the project to benefit Today

Hooray!! We've got agreement on the scope baseline for our project. We've agreed achievable delivery dates. We can now build a PLAN based in REALITY.

The realm of fact is distinct from fancy

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

1 - 23

Propagated, or at least treated with respect, by Academia

Focus :

- * The Future
- * Student Courses

IFPUG 2004 - © Copyright 2004 CHARISMATEK Software Metrics

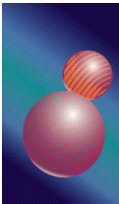
1 - 24



Summary

First, the Good News 

 Now, the Bad News



Thank You

Any Questions and Comments?