



G A L O R A T H

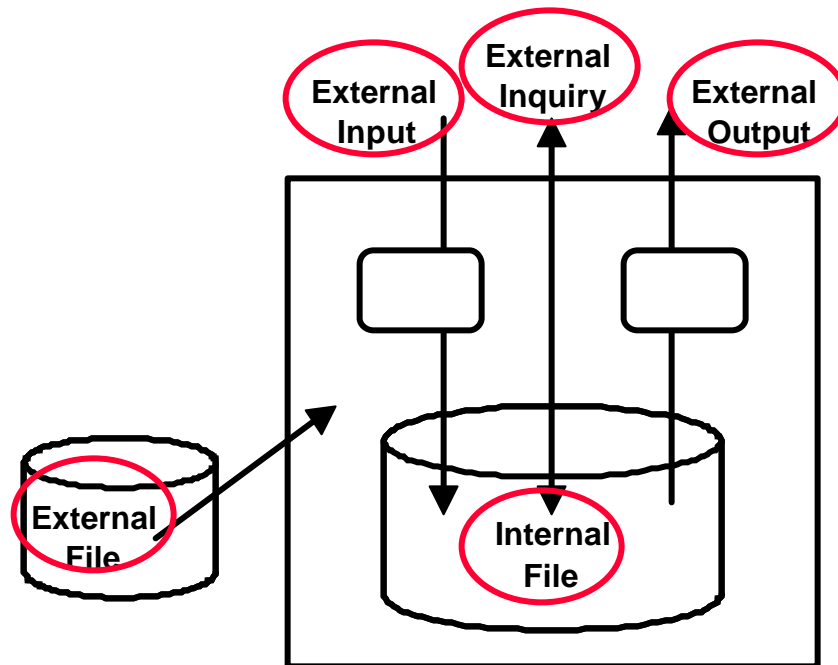
# EVOLVED

## Function Points

**Lee Fischman**  
**COCOMO/SCM 15 Forum**  
**Spring 2000**

# Traditional Function Points - A Great Start At Functional Scoping

## Identification



## Complexity Weighting

		DETs		
		Low	Low	Avg
RETs	Low	Low	Avg	High
	Avg	Avg	High	High
	High	High	High	High

$$\sum \text{Low} * l + \sum \text{Avg} * a + \sum \text{High} * h$$

## Final UFP Count

Unadjusted  
Function  
Points

# Orthodox Function Points...

## What Challenges Remain?

---

**Semantic obfuscations.** The terminology used can be confusing and may promote faulty practice: why call everything “external”? what is complexity? what are RETs, DETs, and FTRs?

**Too many steps.** The orthodox FP counting method is complex enough that it is not immediately obvious. Numerous transformations violate measurement theory and some steps may add no information.

**May not capture all aspects of functionality.** Standard function points may be incomplete; functionality that is substantially internal, without much manifestation at the user interface, may be missed.

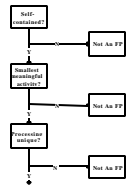
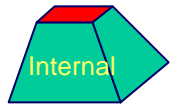
**Arbitrary weightings.** The low / average / high assignment process is arbitrary and discrete; it ideally should be continuous.

**Learning curve too steep.** Perhaps because the counting process is intimidating and semantically incompatible with modern developers, function points have not been widely accepted.



# Evolved Function Points

~~“External”~~



	DETs		
	Low	Low	Avg
RETs	Low	Avg	High
	Avg	High	High

**New names.** Remove “External” designation, deemphasize abbreviations, generalize where possible, clarify terminology.

**New point.** Captures functionality that may be missed with current pts.

**Flexible application boundary.** Recognize that not all “proper” functions may be identifiable at the outermost application interface.

**A backup rule set.** For when items are not clear-cut, a limited set of rules are available; these lie at the core of the metric’s measurement process.

**Adjustment to weighting method.** Simplify to an extent, recognize potential for changes in technology & environment, insure compatibility with current function point repositories.

Report → Output  
Screen → Input

**Discriminate from artifacts.** Tightly couple the metric to everyday software items, establishing a more intuitive connection and speeding up learning.



<New names, New point, Flexible boundary, Core rules, Weightings>

# Semantic Changes

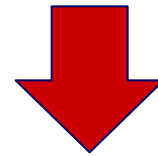
External  
Input  
(EI)

External  
Output  
(EO)

External  
Inquiry  
(EQ)

Internal  
Logical  
File  
(ILF)

External  
Interface  
File  
(EIF)



Input  
(I)

Output  
(O)

Simple  
Input/  
Output  
(IO)

Internal  
Data  
Grouping  
(iD)

External  
Data  
Grouping  
(eD)

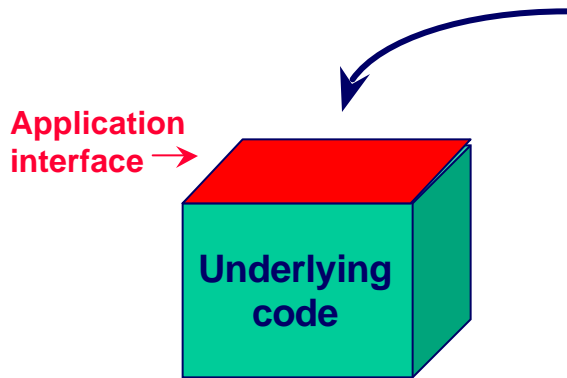


Internal  
Function  
(iF)

<New names, *New point*, Flexible boundary, Core rules, Weightings>

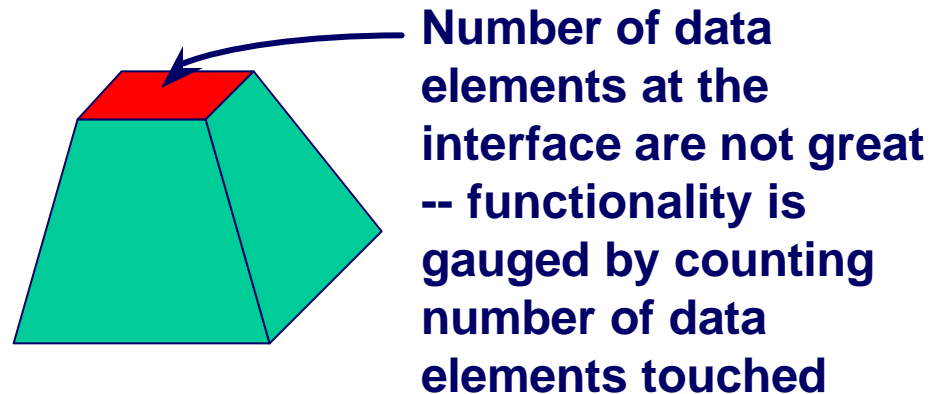
# New Point - Internal Function

## Standard Function Point



Functionality is gauged by counting the number of data elements at the interface

## “Internal Function”



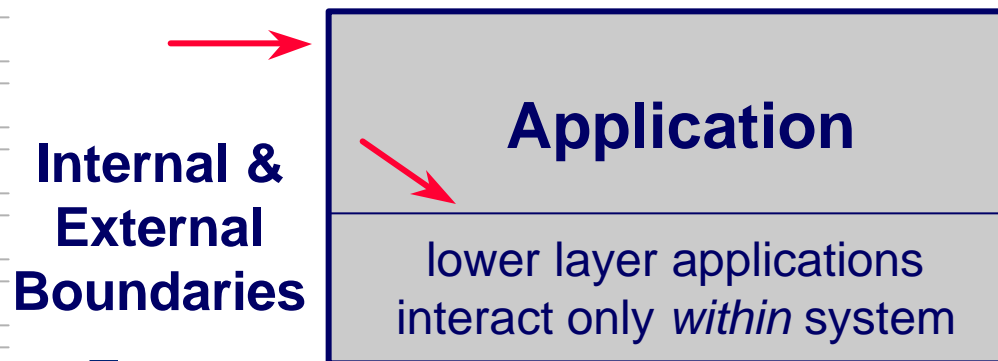
<New names, New point, *Flexible boundary*, Core rules, Weightings>

# Counting Boundary

**Orthodox definition.** *The border between the project or application being measured and the external applications or user domain.*



**Extended definition.** Same basic definition plus underlying boundary layers. This means that not every bonafied function point must be visible at the application's outermost boundary.



Less complex systems may only have a boundary at the user interface. More complex systems may have **self-contained** functions interacting with one another at lower "layers" (see FFP definition).

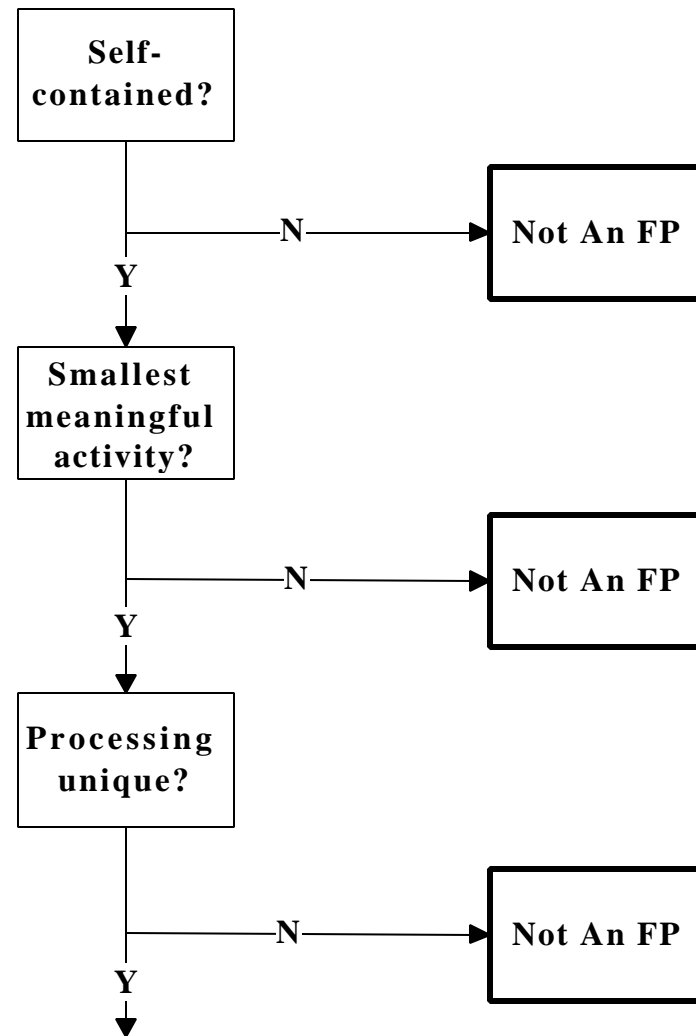
# Resolution Rules

**Three basic rules of FP counting define what is -- and isn't -- a "point"**

**Does this process leave the system in an equilibrium, consistent state?**

**Can this process act independently of something else?**

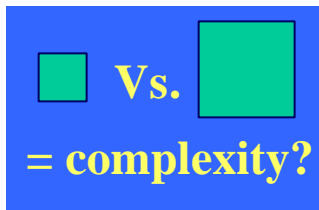
**Is the logic or the data being handled unique to this process?**



**(Two separate rules are used to classify data groupings.)**

<New names, New point, Flexible boundary, Core rules, *Weightings*>

# Critically Examining FP Weightings



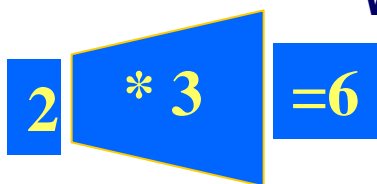
The function point weighting method refers to “complexity” but complexity is an algorithmic factor, hidden from this metric; what is really being gauged is size.



Original FP relative point weightings were determined from IBM applications in the late 70s; they cannot be generalized to other organizations and technologies.



Despite the lack of empirical support for specific weightings, there is some evidence that differentiating function points by type does add useful information.



Whatever weighting system is used, the magnitude of the linear transforms to UFPs is strictly arbitrary. However, it is useful to normalize with weighting magnitudes used to date -- this makes the count comparable with existing repositories of software project measurements.

<New names, New point, Flexible boundary, Core rules, *Weightings*>

## Modified Mappings To 'Size'

<i>Weights for Inputs, I/O, Outputs, Internal Files...</i>		Total Number of Fields		
		1 to 4	5 to 15	16 and over
Logical Groupings of Fields	1	Small	Small	Medium
	2	Small	Medium	Large
	3 and over	Medium	Large	Large

*Two scales are used, owing to different attribute magnitudes for transactions versus data structures.*

<i>Weights for Internal and External Data Groups...</i>		Total Number of Fields		
		1 to 19	20 to 50	50 and over
Logical Groupings of Fields	1	Small	Small	Medium
	2-5	Small	Medium	Large
	6 and over	Medium	Large	Large

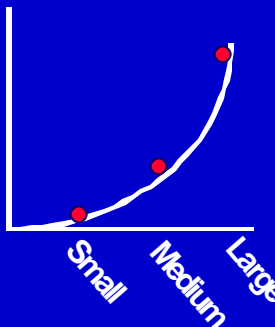


<New names, New point, Flexible boundary, Core rules, *Weightings*>

# How FP Weightings Are Done

**Purpose: derive a single scalar by combining the different types of function points.**

$$\sum \text{Low} * l + \sum \text{Avg} * a + \sum \text{High} * h$$

Basic Count		Multiplied By UFP Expansion Factors			Equals	Within types, some sort of qualitative / quantitative mapping is needed
		Small	Medium	Large		
3	ILF	7	10	15	30	
2	EIF	7	10	15	14	
10	EQ	3	4	6	40	
20	EI	3	4	6	120	
5	EO	4	5	7	20	

= 224 UFPs

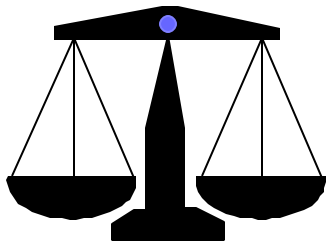
**Scalar value = “Unadjusted Function Points”**

<New names, New point, Flexible boundary, Core rules, *Weightings*>

## How FP Weightings *Could* Be Done

---

**Retain traditional weightings.** Under “naive” circumstances.



**Reweight.** When in-house “expert opinion” - via delphi, etc. - is deemed superior.

**Accept third-party reweightings.** When presented with superior research, i.e. regressions or other quantitative analyses.

*or*

**Keep function points classified separately.** Increasing estimating and descriptive power.



# Examples - Transactional

Target System Experience

Notes: Maximize Note...

Least Likely Most

Nom Nom Nom

OK Cancel Prev Next KBases More Help

Average software development team experience with the target operating system, architecture, compilers, and tools. Rate as of the beginning of development.

EXTRA HIGH	4+ Years Average
VERY HIGH	3 Years Average
HIGH	2 Years Average
NOMINAL	1 Year Average
LOW	4 Months Average
VERY LOW	Less Than 4 Months Average

## Input

4 fields in 2 logical groupings

Small

## Simple Input/Outputs

2 unlike IOs

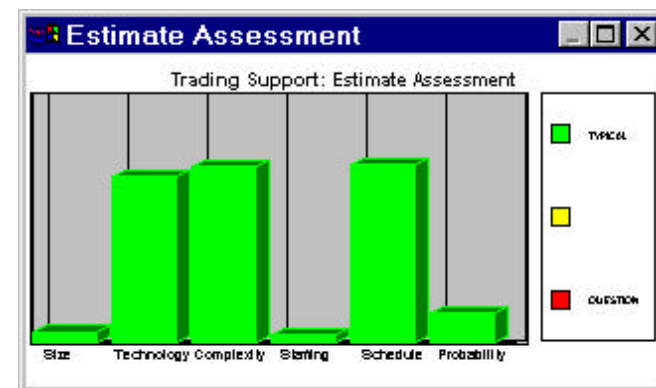
3 alike IOs

Small (3)

## Output

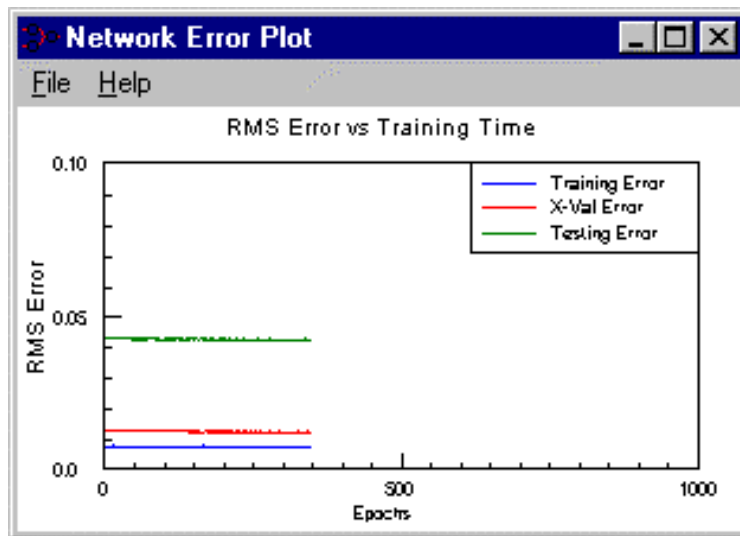
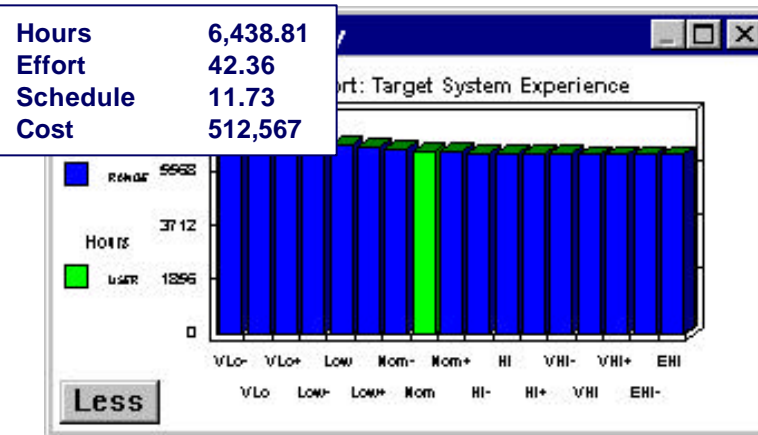
6 fields in 1 logical grouping

Small



# GUI Functional Counting

**Simple Input / Output**  
**4 fields in 1 logical grouping**  
**Small**

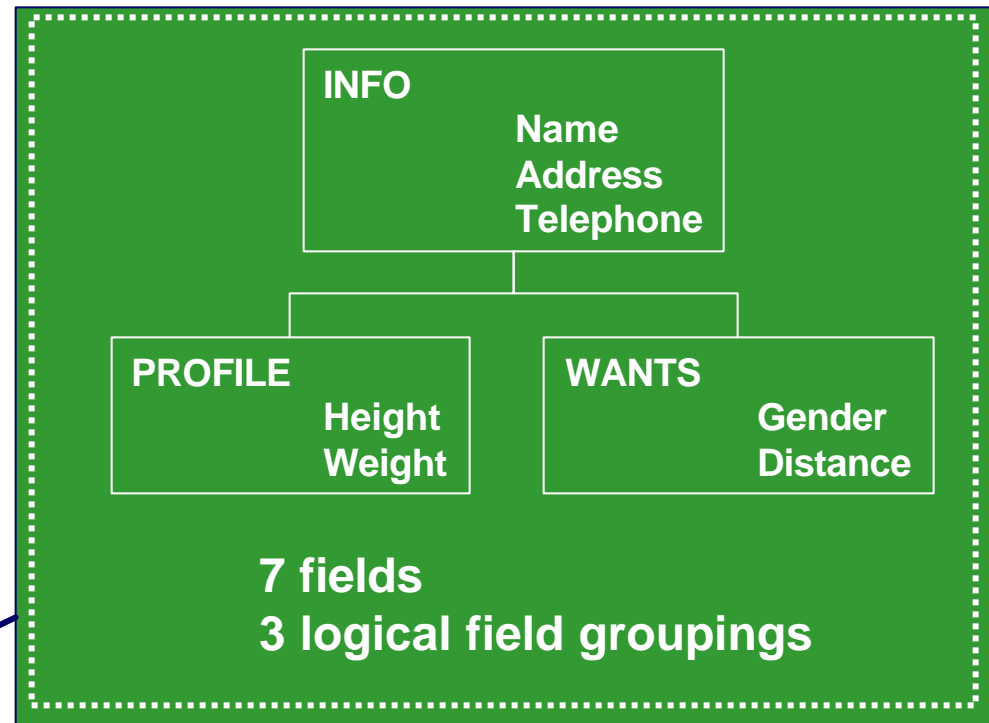
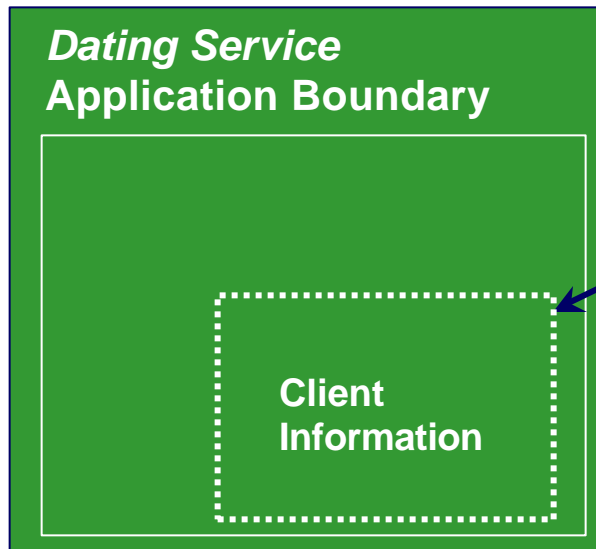


**Internal Function** (neural net)  
**3 fields at interface but much**  
**underlying data handling**  
**High**

or  
**Output and**  
**Lower Boundary Function**

# Functional Counting of Data Groupings

Internal Data Grouping  
7 fields in 3 logical groupings  
Small



# Discrimination From Artifacts

---

## Count Artifacts...

## Translate to points

Input screens; Batch, interactive & hardware inputs



**Input**

Reports; Media, software & hardware outputs



**Output**

Simple inquiries; Menus; Other request/response



**Simple  
Input / Output**

Functions with limited interaction  
but substantial underlying complexity



**Internal  
Function**

Database tables; Long-lived groupings of data



**Internal  
Datagroup**

Read-only files; Outside tables



**External  
Datagroup**



# How Evolved Function Points Are An Improvement On The Orthodox Version

---

## Simplifications

**Process that starts with intuitive software features.** Relate the metric's points to everyday software items, establishing a more intuitive connection and speeding up learning.

**A evolved, backup rule set.** When items are not clear cut, a limited set of rules, which lie at the core of the metric's measurement process.

**Simplified names.** Remove "External" designation, deemphasize abbreviation, generalize where possible, clarify terminology.

## Extensions

**"Internal" point.** For functionality that is missed with current method.

**Flexible application boundary.** Recognize that not all proper functions must be identifiable at a single, outermost application layer.

**Adjustments to weighting method.** Unify weighting, allow for continuous scale, insure compatibility with current function point repositories.

