

Debra Maschino, CFPS, PMP

Olga Makar-Limanov, PhD

Wave of the Future: Function Point Sizing & COTS Support

EDS

5401 Gateway Centre

Flint MI. 48507

USA



September 2006



Wave of the Future: Function Point Sizing and COTS Support

Presentation by:

Debra Maschino, EDS, CFPS, PMP

Olga Makar-Limanov, EDS, PhD

Agenda



- Problem
- Process Approach
- Function Point Counting
- Establish Scope
- Establish Size
- Determine Metrics
- Summary - Lessons Learned

Problem

- In today's world of rapid software development methods, it has become necessary to create, implement and support applications in non-traditional ways.
- This presentation will discuss an approach based on utilizing function point analysis techniques to establish a baseline for process improvement and contract requirements.
- To succeed in the approach, a repeatable and economically feasible method must be used.

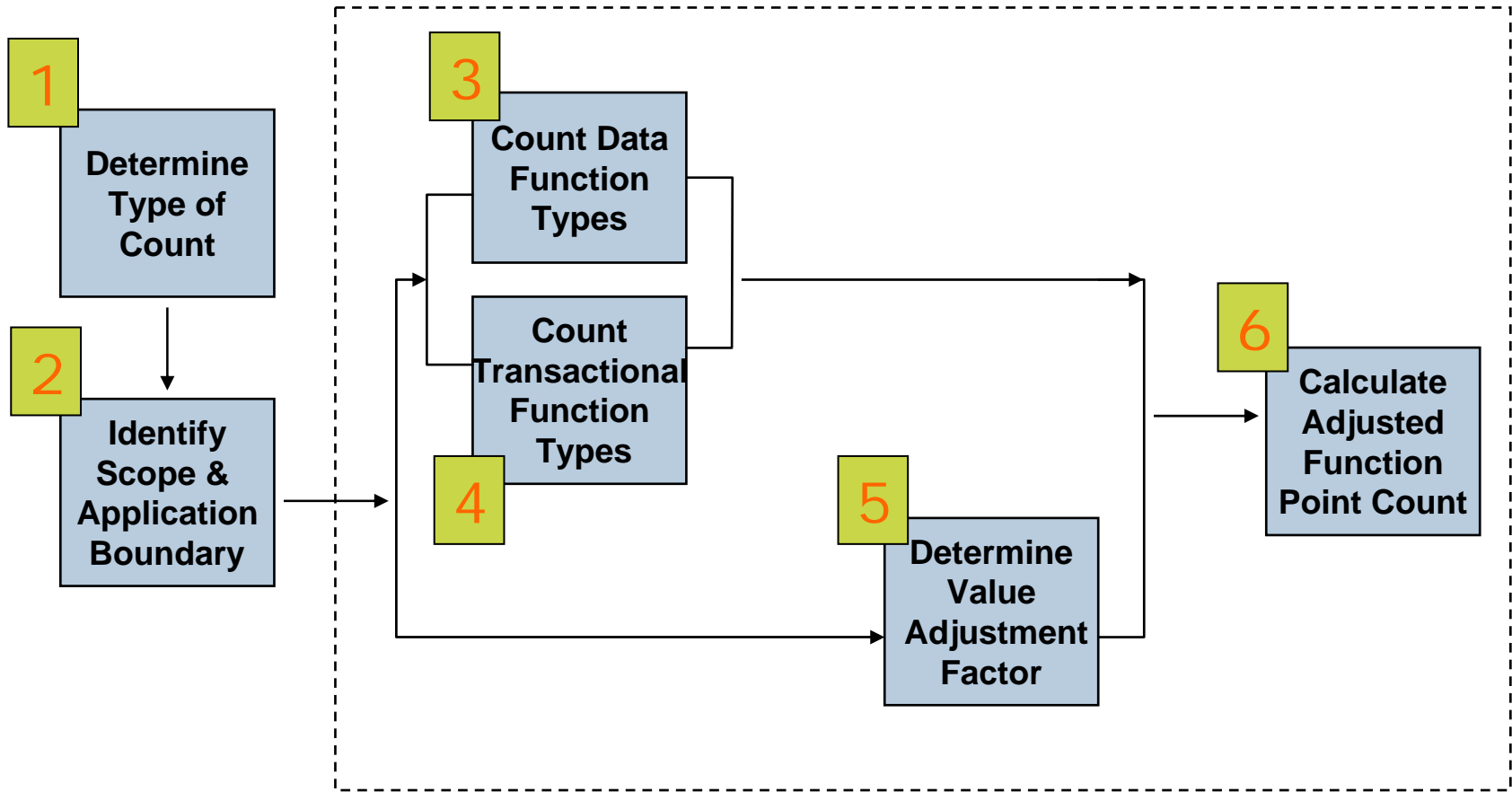


Process Approach

- Use IFPUG Function Point Counting Process
- Function Point Counting Practices Manual compliant
- Economically feasible (quick and inexpensive)
- Methods
 - Sampling
 - Quick Count

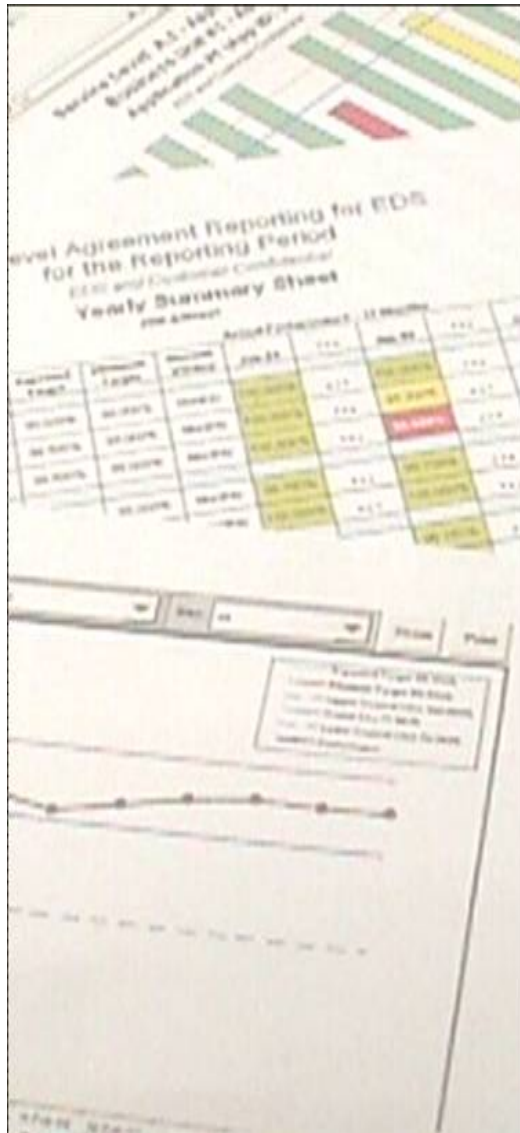


Function Point Counting Process



CPM v4.2

1 Determine Type of Count



- ***Application Count ****
 - ***Associated with the installed application***
 - ***Provides a measure of the current functions used***
 - ***Also referred to as Baseline count***

Purpose of the count is to determine scope basis for support rates and to identify the baseline for performance improvements

* IFPUG Function Point CPM 4.2

2 Identify Counting Scope

- **Counting Scope** *
 - *Defines a (sub) set of the software being sized*
 - *Is determined by the purpose for performing the count*
 - *Identifies functions included in the count*
 - *Could include more than one application*
- **An application count may include, depending on purpose:**
 - *Only the functions being used by the user*
 - *All functions delivered*

Scope of the count is to include all installed and utilized software

- **Counting only the functions being used by the user**
- **Count additional functions created that are not supplied by the package**

* IFPUG Function Point CPM 4.2



2 Identify Application Boundary



- **Application Boundary** *
 - Defines what is external to the application
 - Is the conceptual interface between 'internal application and 'external' user world
 - Encloses the logical data maintained by application
 - Assists in identifying logical data referenced but not maintained by this application
 - Is dependent on user's external business view of application; not based on technical/implementation

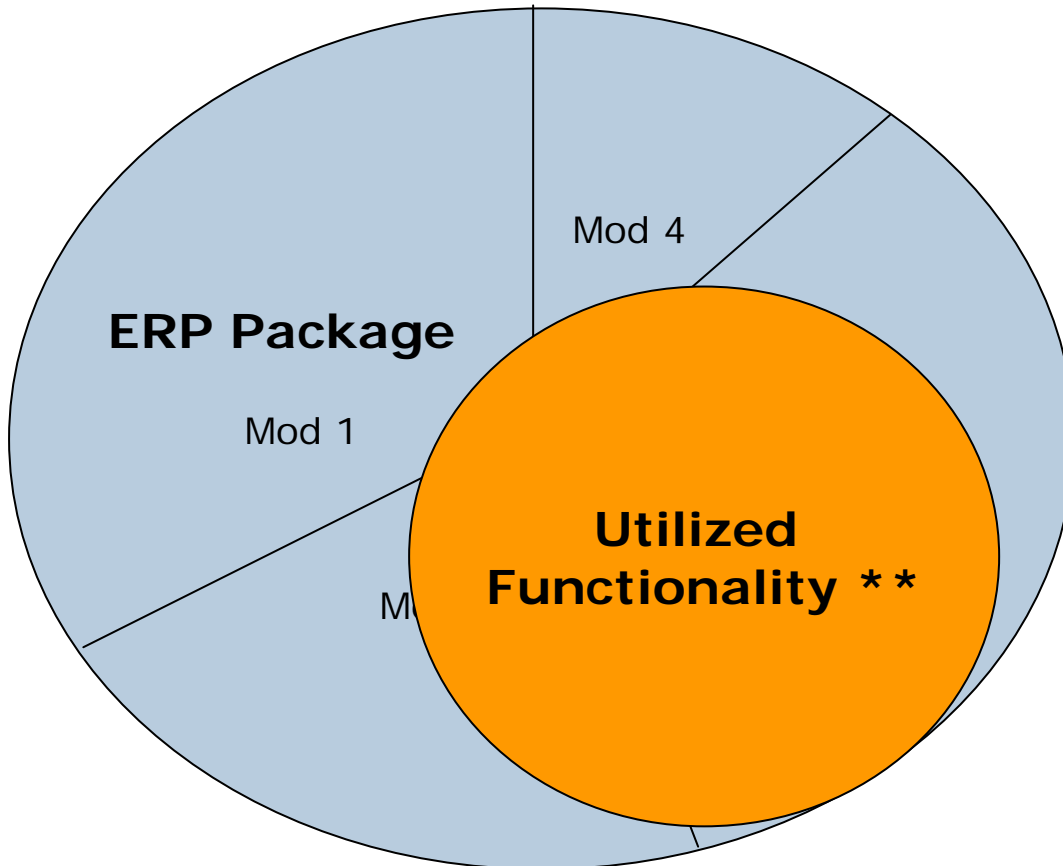
Depending on the modules utilized, the application boundary may be

- **One functional area**
- **Multiple functional areas**

* IFPUG Function Point CPM 4.2

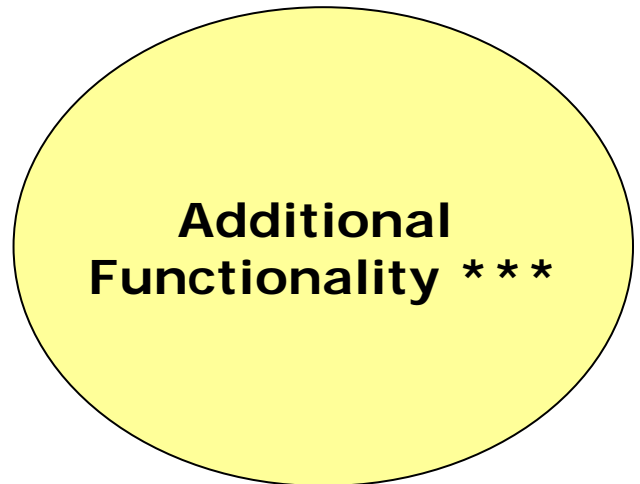
Establish Scope

- Establish size based on functions utilized



**Utilized Functionality +
Additional Functionality**

= MAINTAINED SIZE



3 Count Data Functions

- *Identify the ILFs and EIFs **
 - Most COTS/ERP packages include the capability to list data tables
 - Some may include data dictionary or other data relationship/data flow diagram
- *Determine the ILF or EIF complexity and contribution **
 - Use naming conventions to identify which files belong to which applications
 - Based on the identified fields, determine whether tables are separate internal logical files, shared data, record element types or data element types
 - Assign complexity based on RETs and DETs identified

* IFPUG Function Point CPM 4.2



4 Count Transactional Functions



- **Identify elementary processes ***

- Identify primary intent of identified elementary processes and classify as an EI, EO, or EQ
- Validate against the transaction identification rules

STEPS

- Use system monitoring tools to produce listing of active programs
- Use system security tools to produce listing of screens access
- Identify Non-active Processes & Screens
- Assign transactional function EI, EO, EQ based on description or naming convention
- Only identify unique elementary processes

* IFPUG Function Point CPM 4.2

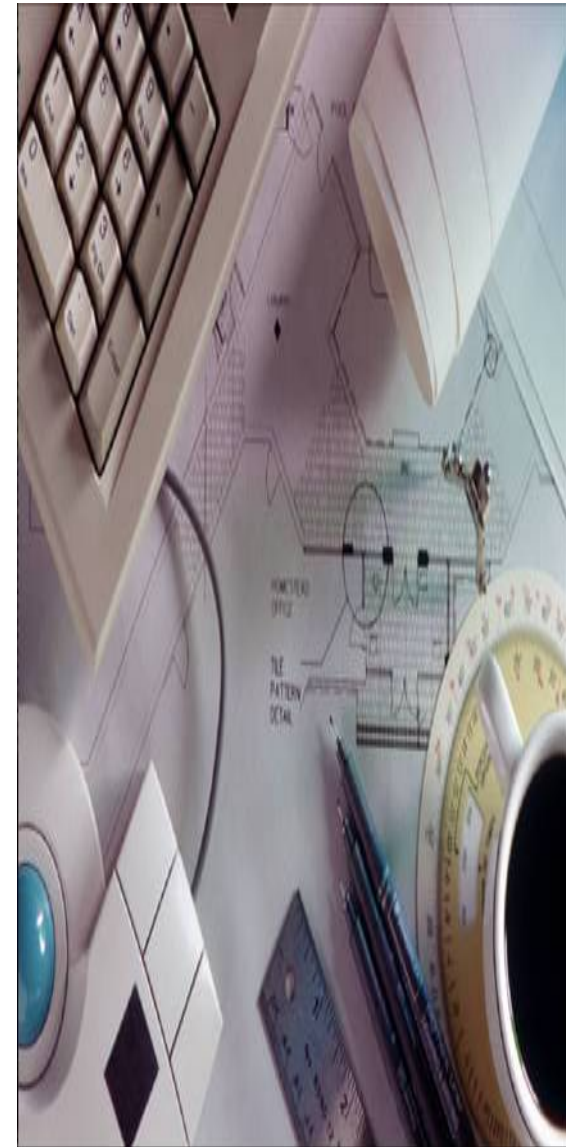
4 Count Transactional Functions (con't)

- *Determine the transaction complexity and contribution **

STEPS

- Use sampling technique to randomly sample from identified transactional functions
- Perform detailed FP count on random selection (Sampling)
- Determine ratio complexity for group
- Assign complexity percentages across remaining transactions based on ratios

* IFPUG Function Point CPM 4.2



Sampling in Practice

- ERP may contain multiple modules (boundaries)
- Sampling should be done within the modules
- Random sample size would be in the same ratio as the utilized pieces in the modules
- Group the EO, EQ, EI transactional functions and sample from each type
- Distribution complexity of low, average and high is the same as the sample

Example

APPLICATION_NAME	USER_CONCURRENT_PROGRAM_NAME	Description	FP Type	Complexity
Oracle Assets	Buildout Source Lines Upload	Update	EI	A
Oracle Payables	PRC: Process Mass Update Batches	Update	EI	A
Oracle Payables	1099 Forms - Comma Delimited Format	Report	EO	L
Oracle Payables	1099 Invoice Detail Report	Report	EO	L
Oracle Payables	1099 Invoice Exceptions Report	Report	EO	L
Oracle Payables	1099 Payments Report by Vendor Type	Report	EO	A
Oracle Payables	1099 Supplier Exceptions Report by Vendor Type	Report	EO	A
Oracle Human Resources	1099R Information Return	Not Active		
Oracle Human Resources	AAP - Job Group Analysis Report	Not Active		
Oracle Human Resources	AAP - Workforce Analysis Report	Not Active		
Oracle Inventory	ABC assignments report	Not Active		
Oracle Inventory	ABC descending value report	Not Active		
Oracle Human Resources	Absences Report	Not Active		
Oracle Receivables	Accept Consolidated Billing Invoices	Not Active		
Oracle Inventory	Account alias listing	Not Active		
Oracle General Ledger	Account Analysis - (132 Char)	Report	EO	L
Oracle General Ledger	Account Analysis - (180 Char)	Report	EO	L
Oracle General Ledger	Account Analysis - Average Balance Audit	Report	EO	A

* IFPUG Function Point CPM 4.2



5 Determine Value Adjustment Factor

- *Evaluate each of the 14 general system characteristics **
- *Add the degrees of influence for all 14 general system characteristics to produce a total degrees of influence (TDI) **
- *Determine Value Adjustment factor **

* IFPUG Function Point CPM 4.2



6 Calculate Adjusted Function Point Count

- *Use the formula to calculate the application function point count **

This becomes the size for calculated measures:

- **Function points supported per FTE**
- **Defects per function point**

The documented function point count is also used for:

- **Application inventory size**
- **Basis of performance improvements**

* IFPUG Function Point CPM 4.2



Determine Metrics: Economics of COTS

- **Inventory control**
 - How big is the package “in the box”?
 - How much is/will be used?
 - How much is/will be used “as is”?
 - How much of what is/will be used should be modified?
- **Support required performance measures**
 - Productivity baseline
 - Productivity tracking
- **Track improvements**
 - How much of the original system was replaced
 - How much of the original system was retained
- **Total cost of ownership**



Determine Metrics: Support Rates



- Support rates need to be based on more than the function point size
- The complexity of environment supported with the functions needs to be addressed
- Some factors to consider:
 - Geographies
 - Time zones
 - Languages
 - Political issues
 - Legal issues
 - Number of users
- Use a questionnaire based on COTS models such as the COCOTS or SLIM questionnaires

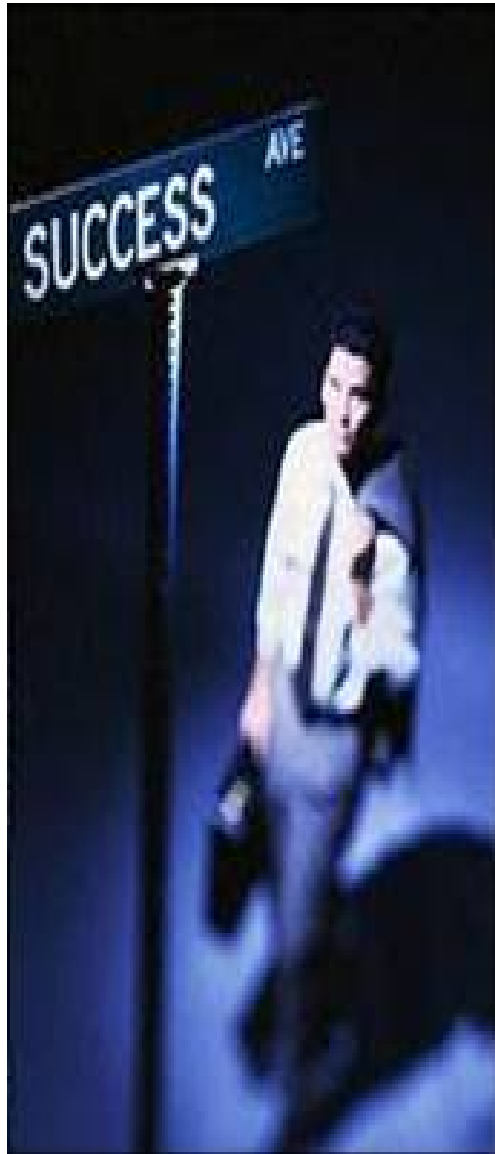
Other Considerations

Non-Functional factors impacting the support rate

- Count verification
- Configuration of COTS systems
- Maintenance or support process requirements
- Disaster Recovery
- User Training
- Help desk
- Types of effort
 - Adaptive
 - Corrective
 - Perfective
 - Enhancement



Summary - Lessons Learned



- Process must be repeatable and economically feasible
- Sampling and quick count methods can be used since package applications tend to be similar in look and presentation
- This approach does not facilitate enhancement counts since there is not much detailed documentation
- COTS enhancement and implementation project counts follow traditional function point analysis methods
- System experts are very important and extremely necessary in determining functionality utilized
- *DOCUMENT ASSUMPTIONS*

Questions?





September 2006



Presentation by:

Debra Maschino, EDS, CFPS, PMP

Olga Makar-Limanov, EDS, PhD

<mailto:debbie.maschino@eds.com>

<mailto:olga.makar-limanov@eds.com>

Appendix

- COTS - Commercial-Off-The-Shelf*
 - Short for *commercial off-the-shelf*, an adjective that describes software or hardware products that are ready-made and available for sale to the general public. For example, Microsoft Office is a COTS product that is a packaged software solution for businesses. COTS products are designed to be implemented easily into existing systems without the need for customization.
- ERP - Enterprise Resource Planning*
 - Short for *enterprise resource planning*, a business management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing. As the ERP methodology has become more popular, software applications have emerged to help business managers implement ERP in business activities such as inventory control, order tracking, customer service, finance and human resources.
- Bespoke - (pronounced bee-SPOHK)**
 - A term used in the United Kingdom and elsewhere for an individually- or custom-made product or service. Traditionally applied to custom-tailored clothing, the term has been extended to information technology, especially for software consulting services. Typically, software consulting companies offer packaged (already invented and generally applicable) software and bespoke software for client needs that can't be satisfied by packaged software. In the U.S., bespoke software is often called custom or custom-designed software.

* Webopedia.com

** Whatis.com

References

- **Evaluating COTS Using Function Fit Analysis:**
 - http://www.qpmg.com/evaluating_cots.htm
- **From IFPUG Discussion:**
 - Posted on Wednesday, May 18, 2005 - 10:07 am:
- **COCOTS - COCOTS (COnstructive COTS)**
 - Focuses on estimating the cost, effort, and schedule associated with using commercial off-the-shelf (COTS) components in a software development project. Though still experimental, COCOTS is a model complementary to COCOMO II, capturing costs that traditionally have been outside the scope of COCOMO. Ideally, once fully formulated and validated, COCOTS will be used in concert with COCOMO to provide a complete software development cost estimation solution.
 - <http://sunset.usc.edu/research/COCOTS/index.html>
- **SLIM-Estimate**
 - QSM's Software Lifecycle Management (SLIM) tools support better decision making at each stage of the software lifecycle: estimating, tracking, and benchmarking and metrics analysis. Each tool is designed to deliver results, whether used as a standalone application or as part of QSM's integrated suite of proven software metrics tools.
 - http://www.qsm.com/slim_estimate.htm