INCOSE Enchantment Chapter

January 11, 2012

A Function Point Overview with Potential Application in Systems Engineering

Albuquerque, N.M.

Joe Schofield
CFPS, CSMS, CQA, LSS BB, CMMI® Certified Instructor
joescho@joejr.com
President, International Function Point Users Group

(Somewhat) Related Presentations & Publications

Function Points & Estimating	*Function Point Analysis – A Cornerstone to Estimating*; ISMA Cinco!, Sao Paulo, Brazil; September 14, 2010 *Why You Need a Certified Function Point Specialist –and lingering questions you can only pretend to answer; ISMA paper; September 2010* The Use of Function Points for Software Measurement & Estimation; Measurement Workshop; Ft. Worth, TX., 2007
Lines of Code	*Counting Lines of Code: Virtually Worthless for Estimating and Software Sizing*, IT Metrics and Productivity Journal; December, 2009 *Is There a Weakest Link After All?,* IT Metrics and Productivity Journal; December, 2009 *Is There Value to using Lines of Code for Measuring People After All?,* IT Metrics and Productivity Journal; December, 2009 *Lines of Code - Statistically Unreliable for Software Sizing*?; Computer Aid, Inc.; Webinar; October 14, 2008 *The Statistical Case Against the Case for using Lines of Code in Software Estimation;* 4th World Congress on Software Quality; Bethesda, MD.; September 17, 2008 *The Statistically Unreliable Nature of Lines of Code;* CrossTalk, April 2005 (Cited by NIST *Metrics and Measures* http://samate.nist.gov/index.php/Metrics_and_Measures)
Defect-icide	*Estimating Latent Defects Using Capture-Recapture: Lessons from Biology; Arlington, VA.; 2008 International Software Measurement and Analysis (ISMA) Conference; September 18, 2008* *Beyond Defect Removal: Latent Defect Estimation with Capture Recapture Method*; CrossTalk, August 2007 (reprinted in IFPUG's MetricViews, Winter 2008) *Latent Defect Estimation - Maturing Beyond Defect Removal using Capture-Recapture Method*; QAI QAAM Conference; September 10, 2008 *Defect Collection & Analysis – The Basis of Software Quality Improvement*; ISMA Conference, September, 2006 *Defect Management through the Personal Software Process℠; CrossTalk, September 2003*
Lean Six Sigma	*Leaning Lean Six Sigma for Results;* ISMA; September, 2009 *When Did Six Sigma Stop Being a Statistical Measure?; CrossTalk, April 2006* *Lean Six Sigma - Real Stories from Real Practitioners*; Albuquerque, N.M.; N.M. SPIN; August 2005 *Six Sigma & Software Engineering: Complement or Collision*; Albuquerque, N.M.; N.M. SPIN; August, 2004 *Applying Lean Six Sigma to Software Engineering;* IFPUG Conference; September, 2004
Process Improve-ment	*'Manda, Panda, and the CMMI(R)*; Las Vegas, NV.; 2007; ISMA Conference; September 14, 2007 *Amplified Lessons from the Ant Hill – What Ants and Software Engineers Have in Common*; IFPUG Conference, Sept., 2003 *Lessons from the Ant Hill - What Ants and Software Engineers Have in Common*; Information Systems Management, Winter 2003 *The Team Software Process℠ – Experiences from the Front Line*; Software Quality Forum; Arlington, Virginia, March; 2003 *Measuring Software Process Improvement - How to Avoid the Orange Barrels*; System Development, December 2001 *Usable Metrics for Software Improvement within the CMM*; Software Quality Forum 2000; Santa Fe, N.M.; April, 2000

Quick Overview . . .

- What are Function Points?
- When are Function Points useful in the product life cycle (hint: from planning to retirement)?
- What is IFPUG?
- Five function point types and their values
- An example, of course
- Function Points and estimating

How Function Point Analysis Helps . . .

- As an ISO standard (ISO/IEC 20926 SOFTWARE ENGINEERING) Function Point Analysis (FPA) provides a basis for repeatable and consistent sizing
- Supported by IFPUG and its membership community, FPA remains viable as new technologies and approaches to software development evolve (case studies, books, conferences, workshops, certifications, and, the "standard")
- Functional sizing is not influenced by programming language, in-house or COTS development
- Functional sizing is not impacted by development approach: outsourcing, in-sourcing, iterative, incremental, scrum, or agility
- Functional sizing can be approximated at the first sighting of customer requirements, estimated with a design, and counted upon delivery
- FPA can be used to track requirements volatility over the life of a project (FPs added, changed, deleted) to size *requirements creep*

About IFPUG (http://www.ifpug.org/). . .

IFPUG is the International Function Point Users Group

IFPUG is a volunteer non-profit organization

IFPUG maintains the standard(s)

- Counting Practices Manual 4.3.1 (2010)
- Certification Process and automated exam in several languages

Provides conferences, workshops, white papers

Supported by numerous service providers for training, consulting, counting

Has a voting membership across six continents

Has a fulltime "home office"

Offers individual and organizational memberships

Why Projects Stumble

Standish Chaos Report
Challenged projects suffer from:

1. Lack of User Input
2. Incomplete Requirements and Specifications
3. Changing Requirements and Specifications
4. Lack of Executive Support
5. Technology Incompetence (DTRA, XML?)
6. Lack of Resources
7. Unrealistic Expectations
8. Unclear Objectives
9. Unrealistic Time Frames
10. New Technology

Impaired (cancelled) projects suffer from:

1. Incomplete Requirements
2. Lack of User Involvement
3. Lack of Resources
4. Unrealistic Expectations
5. Lack of Executive Support
6. Changing Requirements and Specifications
7. Lack of Planning
8. Didn't Need it Any Longer
9. Lack of IT Management
10. Technology Illiteracy

IEEE Spectrum, Robert N. Charette, September, 2005
Why Software Fails

1. Unrealistic or unarticulated project goals
2. Inaccurate estimates of needed resources
3. Badly defined system requirements
4. Poor reporting of the project's status
5. Unmanaged risk
6. Poor communication among customer, developers, and users
7. Use of immature technology
8. Inability to handle the project's complexity
9. Sloppy development practices
10. Poor project management

Function Points come from two sources:

Data Functions

- Internal Logical Files (ILFs)
- External Interface Files (EIFs)

Transactional Functions

- External Input (EIs)
- External Output (EOs)
- External Inquiry (EQs)

Ref: Function Point Counting Practices Manual 4.3.1; January, 2010

Definitions of Function Point Data Functions (two types):

External Interface File (EIF) – user recognizable group of logically related data or control information, which is referenced by the application being measured, but which is maintained within the boundary of another application (Joe's abbreviated description – a data structure which is used to access or retrieve data updated by the system)

Internal Logical File (ILF) – user recognizable group of logically related data or control information maintained within the boundary of the application being measured (Joe's abbreviated description – a data structure which is used to hold data updated by the system)

Ref: Function Point Counting Practices Manual 4.3.1; January, 2010

Definitions of Functional Components are of three types:

"CRUD" – create, read, update, delete

External Input (EI) – elementary process that processes data or control information sent from outside the boundary (Joe's abbreviated description – CUD)

External Inquiry (EQ) – elementary process that sends data or control information outside the boundary (Joe's abbreviated description – R)

External Output (EO) – elementary process that sends data or control information outside the boundary and includes additional processing logic beyond that of an External Inquiry (Joe's abbreviated description – C or U or D, R)

Ref: Function Point Counting Practices Manual 4.3.1; January, 2010

Take a Crack at these Apps (using previous definitions)

1st Crack (but need a systems engineer!)

Application	ILF	EIF	EI	EO	EQ
On/ Off			3		
Mute			3		
Volume +	1		3		
Volume -					
CH +	1		3		
CH -					

1st Crack (but need a systems engineer to validate!)

	LOW	AVG.	HIGH
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Sum of the functional size – (4 (low) EIs x 3) + (2 (low ILFs) x 7)

= (4 x 3) + (2 x 7)

= 12 + 14

= 26

We have a number (26) – so what?

Consider metrics (assuming you collect them) for:

- **Function Points per Person Month**
- **$ per Function Point**
- **Defects per Function Point**
- **Delivery time per Function Point**
- **# of Function Points supported per person (operations & support)**

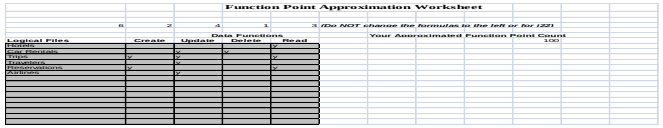
Examples of the diverse usage of FPA
(from Capers Jones)

Products	Circa 2009 Available	Circa 2018 Available	Daily usage (hours)
Home computer	1,000,000	2,000,000	2.5
Automobile	300,000	750,000	3.0
Smart appliances	100,000	750,000	24.0
Televisions	25,000	125,000	4.0
Home alarms	5,000	15,000	24.0
Home music	7,500	20,000	2.5
I-Phone	20,000	30,000	3.0
Digital camera	2,000	5,000	0.5
Electronic books	10,000	20,000	2.5
Social networks	25,000	75,000	2.5
TOTAL	1,494,500	3,790,000	20.5

Using Function Point Metrics For Software Economic Studies, Capers Jones, January 2010

(now that we know what function points are) Estimating and measuring throughout the product lifecycle

1. **Approximate** the size as soon as requirements are discovered (not merely when baselined, accepted, or approved). We now have an idea of the size of the product.



Function Point Approximation Worksheet

	6	2	4	1	3

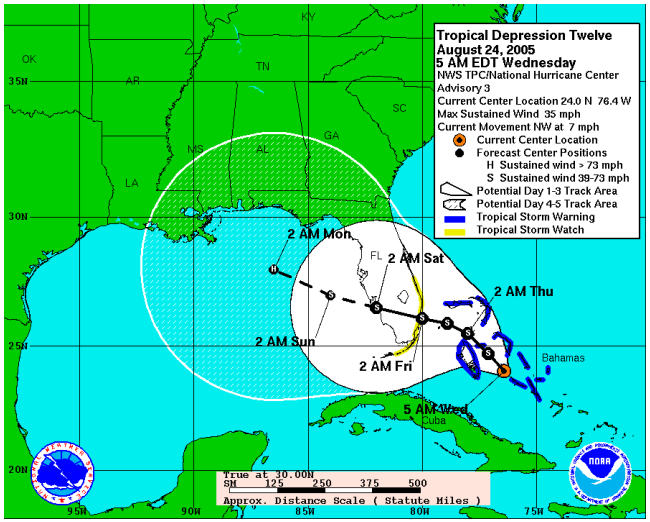
(Do NOT change the formulas to the left or for i221

Logical Files	Data Functions Create	Update	Delete	Read	Your Approximated Function Point Count 135
Hurricane					
Car Rentals		L			
Trips	L	H		L	
Travelers		L			
Reservations	L	H		L	
Airlines		H			

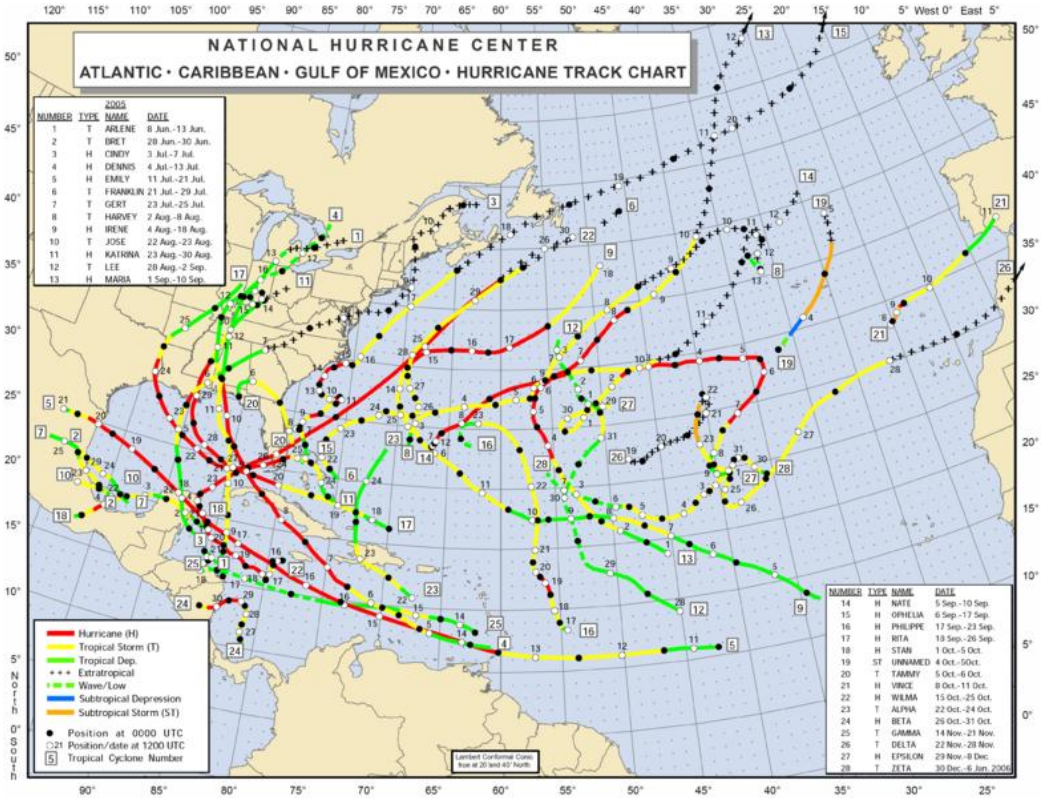
More approximations are preferred – statistical *cone of uncertainty*.

Hurricane Katrina

Provides a likely path (outcome) and margin of error



Use multiple models (QDE) and historic data



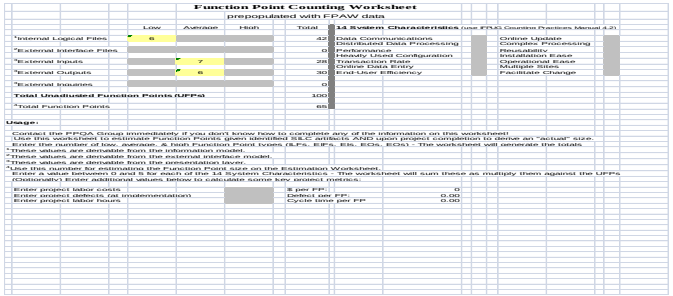
Approximate based on historical performance data (this assumes that such data is collected, stored, and analyzed).

See also CMMI-DEV® v1.3:

- **Measurement and Analysis, SG2**
- **Organizational Process Performance, SG1**

Estimating and measuring throughout the product lifecycle (continued)

2. **Estimate** when size is understood and resources are made available to the project



Function Point Counting Worksheet

prepopulated with FPAW data

	Low	Average	High	Total	14 System Characteristics (use FPA's Counting Practices Manual 4.2)			
*Internal Logical Files	6			42	Data Communications		Online Update	
					Distributed Data Processing		Complex Processing	
*External Interface Files				0	Performance		Reusability	
					Heavily Used Configuration		Installation Ease	
*External Inputs		7		28	Transaction Rate		Operational Ease	
					End-User Efficiency		Multiple Sites	
*External Outputs		6		30	End-User Efficiency		Facilitate Change	
*External Inquiries				0				
Total Unadjusted Function Points (UFPs)				100				
*Total Function Points				85				

Usage:

Contact the PPQA Group immediately if you don't know how to complete any of the information on this worksheet!
Enter the number of low, average, & high Function Point Units (ILFs, EIFs, EIs, EOs, EQs) - The worksheet will generate the totals.
*These values are derivable from the information model.
**These values are derivable from the external interface model.
***These values are derivable from the presentation layer.
*Use this number for estimating the Function Point size on the Estimation Worksheet.
Enter a value between 0 and 5 for each of the 14 System Characteristics - The worksheet will sum these as multiply them against the UFPs (Optionally) Enter additional values below to calculate some key project metrics.

Enter project labor costs		$ per FP:	0
Enter project defects (at implementation)		Defect per FP:	0.00
Enter project labor hours		Cycle time per FP	0.00

3. **Count, record, and analyze** the size, cost, and schedule of the project. (can be used for future estimations)

Closing thoughts . . .

Others working on the use of Function Point-like measures for systems engineering and INCOSE include Ricardo Valerdi, University of Arizona and Mauricio Aguiar, President of TI Metricas, Brazil. http://www.lit.inpe.br/seday

Use Function Points to get closer to the right size of the product; improve estimates . . .

Elicitation with the customer is a discussion you will have anyhow (the spreadsheet is merely a way to record it)

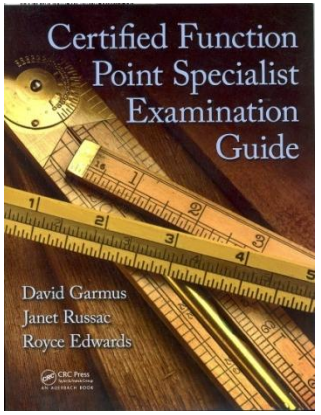
Do you have organizational measures on which to predict cost and hours / schedule once you have a size?

Do you have multiple ways of estimating that might show you the overlapping space and raise confidence?

Do you contribute your measures to an organizational repository for your benefit and that of others?

Defect re-work is already in your organizational productivity data; what happens if you eliminate much of that re-work?

Additional References



[1] Certified Function Point Specialist Examination Guide; Garmus, et. al.; 2010; ISBN 978-1-4200-7637-0
[2] http://en.wikipedia.org/wiki/Function_point
[3] http:www.ifpug.org
[4] *Beyond Defect Removal: Latent Defect Estimation with Capture Recapture Method*; CrossTalk; August, 2007
[5] Capers Jones has reported on a survey of organizations that used lines of code as a size for software; one-third of the participants counted comments as lines of code, one-third did not include lines of code in their counts, and the other one-third didn't know if they counted comments or not
[6] *The Statistically Unreliable Nature of Lines of Code;* CrossTalk, April, 2005
[7] ISO / IEC 20926:2009
[8] http://www.ifpug.org/certification/cfps.htm
[9] *Chaos Summary 2009*; Standish Group, 2009
[10] *A Discipline for Software Engineering*; Watts Humphrey; Addison-Wesley; 1995 pg. 84
[11] *Why Software Fails*; Robert N. Charette; IEEE Spectrum; September, 2005
[12] *Counting Lines of Code: Virtually Worthless for Estimating and Software Sizing*, IT Metrics and Productivity Journal; December, 2009
[13] *Is There a Weakest Link After All?,* IT Metrics and Productivity Journal; December, 2009
[14] *Is There Value to using Lines of Code for Measuring People After All?,* IT Metrics and Productivity Journal; December, 2009
[15] *2010 DCG Survey Results Performance Measurement;* David Consulting Group; 2010