Function Point Analysis

Introduction and Basic Overview as an Alternative to SLOC-based Estimation

November 17, 2010

Tucker Moore
TASC - ASOU
Software Cost Estimation

- Two Main Types of Developed SW Cost Estimation
  - SLOC Based Estimation
  - Function Point Analysis

- What’s the Difference?
  - SLOC deals specifically with counting and estimating the Lines of Code for a program. It is explicitly code length-based, usually to apply a $/LOC productivity rate to an estimate.
  - Function Point Analysis quantifies and assigns a value to the actual uses, interfaces, and purposes of a piece of SW. It also adjusts these values depending on the complexity of the program.

- This presentation focuses on Function Point Analysis as an alternative to SLOC – based estimations.

Robert Cringely - "If automobiles had followed the same development cycle as the computer, a Rolls-Royce would today cost $100, get a million miles per gallon, and explode once a year, killing everyone inside."
What is a Function Point?

IFPUG (International Function Point Users Group):
- Function Point Analysis (FPA) is a sizing measure of clear business significance. The FPA technique quantifies the functions contained within software in terms that are meaningful to the software users.

SCEA:
- Function points are a size measure that, as the name indicates, considers the number of functions being developed based on the requirements specification.

So...What does that mean?
- Simply Speaking: Function Points are the aspects of a SW application that a User recognizes as important to the SW program’s actual use.
For the Visual Learner: Cable Company Billing Example

Customer Bill Information Input Screen

- Name
- Phone Number
- SSN#
- Address
- Zip Code
- IP Address
- Services Used

Customer Bill in Mail

Customer Service Database

IP Address Database

Services and Price List

Advertising Messages

Financial Record System

© 2010 TASC, Inc.
Quick History of FP Analysis

- Allan Albrecht, of IBM, developed the method of Function Point Counting in 1979 in *A New Way of Looking at Tools*

- In 1986, the IFPUG, or International Function Point Users Group, was set up to develop and apply standards to the practice of function point analysis
  - IFPUG has numerous international partners in Europe, Australia, and Asia

- Since 1986, several versions of the Function Point Counting Practices Manual have been published by IFPUG. However, a new version is published only out of necessity in order to keep the standards from changing.

Getting Started: What do you need?

The Right Resources

- The Program’s Primary Users
- Program Developers / People who are familiar with the program (logically)
- Customers
- System Analysts
- Project Managers
- Function Point Specialists
- Measurement Analysts

Picture borrowed from the Audi website. They looked like they were working well together.
What else do you need?

The Right Documentation

- Helps give a visual look into the program being counted
  - High-level application architecture
  - A logical data model
  - Detailed design specifications and requirements, including functionality requirements
  - Business function/process models
  - User manuals
  - Screen prints
  - Printed report layouts
  - Function Point Counting Practices Manual
Function Point Analysis can be performed with as many/few of these documents as are available

- Documents are only necessary for assisting the analyst to facilitate the visual mapping process for the program with a manager or engineer
- A high level architecture, design specifications, and function/process models are all sufficient if the analyst can understand them and the manager can explain them
  - This ability to work with preliminary documents is beneficial especially because this is all the cost analyst has to work with in many situations
Where do I get this data?

- ICBD (for the Intelligence Community)
- CARD (for DoD Programs)
- User Interviews
- Customer Interviews
- Programmer Interviews
- Past Similar Systems
  - Like in SLOC-based estimation
  - Gives a great comparison metric
- Common Sense
Why do you need this data?

- Historical Data and Pre-Established Parametric Data
  - Similar programs can be used to establish relationships or to see possible trends in the function growth and development time frame

- Must be able to visualize the logical progression
  - Visual Maps are essential to understanding the flow of the program

- Insight into the program complexity

- Identify important, easily-forgotten features
So, How do you count Function Points?

- Disclaimer: This is just the Basic Idea
- Let’s go back to the “ComCable Company” Example
  - Estimate for New Customer Billing System
  - Assuming we’re starting from scratch
  - Customer Services maintains Customer Billing Info, enters into the system
  - The information going onto Bill comes from multiple, externally maintained systems

<table>
<thead>
<tr>
<th>SX4 &amp; Associates</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Harnett</td>
</tr>
<tr>
<td>123 King Street,</td>
</tr>
<tr>
<td>Waterloo, Ontario</td>
</tr>
<tr>
<td>NL 3J1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Minutes</th>
<th>Dialed Minutes</th>
<th>Rate ($/min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu, Oct 11</td>
<td>7:00am</td>
<td>32</td>
<td>123</td>
<td>$0.35</td>
</tr>
<tr>
<td>Sun, Nov 1</td>
<td>7:00am</td>
<td>22</td>
<td>0</td>
<td>$0.35</td>
</tr>
<tr>
<td>Sun, Nov 2</td>
<td>1:00pm</td>
<td>20</td>
<td>20</td>
<td>$0.35</td>
</tr>
<tr>
<td>Wed, Nov 5</td>
<td>7:00am</td>
<td>22</td>
<td>0</td>
<td>$0.35</td>
</tr>
<tr>
<td>Sat, Nov 7</td>
<td>5:00pm</td>
<td>45</td>
<td>123</td>
<td>$0.25</td>
</tr>
<tr>
<td>Mon, Nov 16</td>
<td>7:00am</td>
<td>20</td>
<td>20</td>
<td>$0.35</td>
</tr>
<tr>
<td>Sat, Nov 23</td>
<td>7:30am</td>
<td>39</td>
<td>123</td>
<td>$0.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Long Distance Call</td>
<td>$30</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Current Bill</td>
<td>$57.15</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Billing Summary
  - Discount Amount: $0.00
  - Total Current Bill: $57.15
  - Total Amount Due: $57.15

© 2010 TASC, Inc.
Again, For the Visual Learner

Customer Service Database
IP Address Database
Services and Price List
Tax Tables

Advertising Messages
Financial Record System

Customer Bill
Information Input Screen
Name
Phone Number
SSN#
Address
Zip Code
IP Address
Services Used

Customer Bill in Mail
Function Point Counting Process

1. Determine the **Type** of FP Count (New Development, Enhancement, Existing Application)
2. Identify the **Scope** and the **Boundaries** of the Application
3. Count the **Data Function** Types
4. Count the **Transaction Function** Types
5. Calculate the **Unadjusted Function Points (UFP)**
6. Determine the **Value Adjustment Factor (VAF)**
7. Calculate the **Adjusted Function Points (AFP)**
Where are we in the process?

- The **Type** of count that we’re performing is a “New Development Count”
  - We assumed that this is the first time a billing system was created
  - No existing code or structure was introduced

- We’ve already identified the **Scope and Application Boundaries**
  - We know the purpose
  - We know what data goes in / comes out through interfaces and user transactions
  - We know what the User wants

```
Determine the Type of FP Count
(New Development, Enhancement, Existing Application)
```

```
Identify the Scope and the Boundaries of the Application
```
Now We Count the Functions

- **Two Types of Functions**
  - Data Functions
  - Transaction Functions

- **Data Functions**
  - Made up of the Internal and External “resources” that affect the system
  - Internal Logical Files (ILF) and External Interface Files (EIF)

- **Transaction Files**
  - Made up of the processes that are exchanged between the user, the internal files, and the external files
  - External Inputs (EI), External Outputs (EO), and External Inquiries (EQ)

- More detail on these on next slide
Remember, we have two types of Data Functions
  – ILFs and EIFs

**ILFs**
  – Internal Logical Files are those that are User identifiable groups of data and are maintained by the User
  – Let’s assume we have one ILF: “ComCable” Customers

**EIFs**
  – External Interface Files are User identifiable groups of data that are maintained by someone Other Than the user.
  – EIF’s hold information that is referenced to by an ILF
  – Assume we have six
Transaction Functions are the inputs, outputs, and data retrievals through logical processing.

Types: External Inputs, External Outputs, External Inquiries

External Inputs (EI)
- Unique process, data goes INTO application from outside the boundary
- Intent is to maintain / alter the system

External Output (EO)
- Data comes OUT of the system
- Intent is to present information to a user
- Performs Calculation, Derives Data, or Updates ILF

External Inquiries (EQ)
- Data comes OUT of the system
- Intent is to present information to a user
- Performs NO calculations, Derives NO data, Updates NO ILFs
Transaction Functions in the Example

- **External Inputs**
  - (on INPUT screen)
    - Add Record Feature
    - Change Record Feature
    - Delete Record Feature

- **External Outputs**
  - The Customer Bill Report
  - Print Report Feature

- **External Inquiries**
  - (on INPUT screen)
    - Report Look-Up Feature
Customer Information

Name:__________________
Phone Number:__________
SSN#:_________________
Address:_______________

Bill Output

ComCable Customer Bill

Name
Phone Number
SSN#
Address
Zip Code
IP Address
Services Used
Taxes
Hidden Fees
Total
Advertisement Info

Add  Change  Delete  Print  Look-Up

Print Bill
Great, so how many Function Points?

- Here is where **Complexity** comes into play
- EIFs and ILFs are broken up into two parts
  - Record Element Types (RET)
  - Data Element Types (DET)
- EI, EO, and EQs are broken into two parts
  - File Types Referenced (FTR)
  - Data Element Types (DET)
In ILFs and EIFs, Record Element Types (RET) are the largest user-identifiable subgroup of elements
- Our ILF has 3 examples: Cable, Phone, and Internet Customers WITHIN ComCable Customers
- EIF Example: Customer’s Current Balance Due within the Financial Record System

Data Element Types (DETs) are the different elements within each RET
- The Cable Customer RET has Name, Number, SSN, etc. as DETs
- The Customer’s Current Balance Due has “Balance Due” as a DET
FTRs and DETs

- Counted for EI, EO, and EQ
- Same basic definitions as RETs and DETs for ILF/EIF
- File Types Referenced (FTRs) are the larger, user-identifiable subgroups within the EI, EO, EQ that are *Referenced To*
- Data Element Type (DET) is the data subgroup within an FTR
  - These DETs are only counted ONCE for the same logical process: if already counted by an earlier process, then they can’t be counted again
### Example of RET, FTR, DET Counts

<table>
<thead>
<tr>
<th>ILF/EIF</th>
<th>RET</th>
<th>DET</th>
<th>EI/EO/EQ</th>
<th>FTR</th>
<th>DET</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ILF- ComCable Customers</strong></td>
<td>Cable Customers</td>
<td>Name Number SSN Address Zip Code Service Used</td>
<td>EI – Cable Customer - Add Record</td>
<td>ILF – ComCable Customers</td>
<td>Name Number SSN Address Zip Code IP Address Service Used</td>
</tr>
<tr>
<td><strong>ILF- ComCable Customers</strong></td>
<td>Phone Customers</td>
<td>Name Number SSN Address Zip Code Service Used</td>
<td>EI – Cable Customer – Change Record</td>
<td>ILF – ComCable Customers</td>
<td>Name Number SSN Address Zip Code IP Address Service Used</td>
</tr>
<tr>
<td><strong>ILF- ComCable Customers</strong></td>
<td>Internet Customers</td>
<td>Name Number SSN Address Zip Code Service Used</td>
<td>EO – Customer Bill</td>
<td>ILF – ComCable Customer</td>
<td>ALL OF ABOVE Total Due Taxes Bar Code</td>
</tr>
<tr>
<td><strong>EIF – Zip Code</strong></td>
<td>Zip Code Table</td>
<td>Zip Code</td>
<td></td>
<td>EIF – Services/ Price</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EIF – Zip Code</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EIF – Financial Records</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EIF – Advertisements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EIF – Tax Table</td>
<td></td>
</tr>
</tbody>
</table>

© 2010 TASC, Inc.
These tables give function point values to the different RET/FTR DET combinations.

Each ILF, EIF, EI, EO, EQ is counted separately, then added up.

Ex. The Customer Bill EO has >3 FTRs, >6 DETs, therefore HIGH complexity, 7 Function Points.

The total of these Function Points = Unadjusted Function Point (UFP) count.

### ILF / EIF

<table>
<thead>
<tr>
<th>RET’s</th>
<th>DATA ELEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Low Low Ave</td>
</tr>
<tr>
<td>2-5</td>
<td>Low Ave High</td>
</tr>
<tr>
<td>&gt; 5</td>
<td>Ave High High</td>
</tr>
</tbody>
</table>

### EI

<table>
<thead>
<tr>
<th>FTR’s</th>
<th>DATA ELEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>Low Low Ave</td>
</tr>
<tr>
<td>5-15</td>
<td>Low Ave High</td>
</tr>
<tr>
<td>&gt; 15</td>
<td>Ave High High</td>
</tr>
</tbody>
</table>

### EO and EQ

<table>
<thead>
<tr>
<th>FTR’s</th>
<th>DATA ELEMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>Low Low Ave</td>
</tr>
<tr>
<td>6-19</td>
<td>Low Ave High</td>
</tr>
<tr>
<td>&gt; 19</td>
<td>Ave High High</td>
</tr>
</tbody>
</table>

### Ratings and Values

<table>
<thead>
<tr>
<th>Rating</th>
<th>ILF</th>
<th>EIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Average</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>High</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rating</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>EQ</td>
</tr>
<tr>
<td>Low</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>5</td>
</tr>
<tr>
<td>High</td>
<td>7</td>
</tr>
</tbody>
</table>
Next Step in the Process

1. Determine the **Type** of FP Count (New Development, Enhancement, Existing Application)
2. Identify the **Scope** and the **Boundaries** of the Application
3. Count the **Data Function** Types
4. Count the **Transaction Function** Types
5. Calculate the **Unadjusted Function Points (UFP)**
6. Calculate the **Adjusted Function Points (AFP)**
7. Determine the **Value Adjustment Factor (VAF)**
The factor that normalizes the Unadjusted Function Point count

Calculated by asking the 14 General System Characteristic Questions
- Purpose is to apply further valuation to system complexity
- Sums up “Degrees of Influence” for each GSC

VAF calculation can be performed at Any point in the FP counting process
- Any Added / Changed / Deleted functionality of a system results in VAF recalculation

\[ VAF = 0.65 + \left( \frac{\sum \text{Deg. Of Influence}}{100} \right) \]
General System Characteristic Questions

These questions help to describe the complexity of a program.

The analyst assigns a value of 1 – 5 Degrees of Influence for most questions.

<table>
<thead>
<tr>
<th></th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data Communications: Describes the degree to which the application communicates directly with the processor.</td>
</tr>
<tr>
<td>2</td>
<td>Distributed Data Processing: Describes the degree to which the application transfers data among physical components of the application.</td>
</tr>
<tr>
<td>3</td>
<td>Performance: Describes the degree to which response time and throughput performance considerations influenced the application development.</td>
</tr>
<tr>
<td>4</td>
<td>Heavily Used Configuration: Describes the degree to which computer resource restrictions influenced the development of the application. Heavily used operational configurations may require special considerations when designing the application.</td>
</tr>
<tr>
<td>5</td>
<td>Transaction Rate: Describes the degree to which the rate of business transactions influenced the development of the application.</td>
</tr>
<tr>
<td>6</td>
<td>On-Line Data Entry: On-line User Interface describes the degree to which data is entered or retrieved through interactive transactions. On-line User Interface for data entry, control functions, reports, and queries are provided in the application.</td>
</tr>
<tr>
<td>7</td>
<td>End-User Efficiency: Describes the degree of consideration for human factors and ease of use for the user of the application measured. The on-line functions provided emphasize a design for user efficiency.</td>
</tr>
<tr>
<td></td>
<td>General System Characteristic Questions</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>8</td>
<td><strong>On-Line Update</strong>: Describes the degree to which internal logical files (ILF) are updated on-line. The application provides on-line updates for the ILF’s.</td>
</tr>
<tr>
<td>9</td>
<td><strong>Complex Processing</strong>: Describes the degree to which processing logic influenced the development of the application.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Reusability</strong>: Describes the degree to which the application and the code in the application have been specifically designed, developed, and supported to be usable in other applications.</td>
</tr>
<tr>
<td>11</td>
<td><strong>Installation Ease</strong>: Describes the degree to which conversion from previous environments influenced the development of the application. A conversion/installation plan and/or tools were provided and tested during the system test phase.</td>
</tr>
<tr>
<td>12</td>
<td><strong>Operational Ease</strong>: Describes the degree to which the application attends to operational aspects, such as start-up, back-up, and recovery processes. The application minimizes the need for manual activities, such as tape mounts, paper handling, and direct, on-location manual intervention.</td>
</tr>
<tr>
<td>13</td>
<td><strong>Multiple Sites</strong>: Describes the degree to which the application has been developed for different hardware and software environments.</td>
</tr>
<tr>
<td>14</td>
<td><strong>Facilitate Change</strong>: Describes the degree to which the application has been developed for easy modification of processing logic or data structure. Made up of two parts: Flexible Query and Business Data Control Data.</td>
</tr>
</tbody>
</table>
The “Final” Function Point count

Applies the Value Adjustment Factor (VAF) to the Unadjusted Function Point (UFP)

\[
\text{AFP} = \text{UFP} \times \text{VAF}
\]

Some certain situations, such as an Enhancement Function Point Count, require additional math
Function Points in Cost Estimation

- Major metric is $ / Function Point  (Cost)
- Function Point / Person-Month  (Productivity)

- For Very Similar Systems: SLOC / Function Point  (Cost)

- Like all cost estimation, ALL of these metrics require GOOD historical data
Pros and Cons of Function Points

General Benefits (not necessarily benefits over SLOC):
- Independent of Technology
- Independent of Programming Languages
- Requirements are the only thing needed for a FP Count
- SLOC can grow but Functionality usually stays the same
- Provides a method of easier communication with business groups
- Clear view of size, cost, and productivity
- Keeps all parties involved in estimate
- Provides a naturally strong base of documentation

Cons:
- Can be very time-consuming
- Requires a good base of historical data and past function point counts
- Requires a trained function point counter
- Counting techniques can vary from counter to counter
- There are no COTS packages available for FP Counting that are recognized by IFPUG
- Suffers some of the same pitfalls as the Build-Up methodology
- Incurs the inherent risk when using analogies
Recommendations

- Begin counting Function Points alongside counting SLOC
  - Need historical data before relying on FP’s completely
  - Strengthens FP knowledge and ability within group

- Count Function Points for past programs
  - Again, need to build a firm base of historical data
  - FP counting training and practice

- Compare Results
  - How long it takes to produce function point-based estimates
  - How Accurate / Precise (margin of error)
  - Customer preference
Function Point Analysis quantifies a system or application’s functional uses

Function Points are a solid alternative to SLOC counting for developed SW estimation
- Independent of Technology / Programming Languages
- Relatively simple
- Great communication device

Can be completed at all stages of development

Should Test and Practice
- Gain a base of historical data
- Compare to SLOC
- Pick up where SLOC leaves off

To Reiterate: This presentation is not trying to assert Function Point Analysis as dominant over SLOC-based estimation
- FPA is presented as an oft-overlooked alternative to SLOC
Thank You

- For more information, see the references page and visit some of the sites given
- Contact me with questions, comments, concerns, etc.
- Tucker Moore – TASC
  - tucker.moore@tasc.com
  - (703) 449-3646
  - (703) 785-8650
Resources


- Q/P Management Group – Lori Holmes

