# Size Measurement and Estimation

*An approach to convert collected effort data into something that can meaningfully be used for estimation and comparison purposes*

# Introduction

- This document was prepared because assessments have shown that by and large, organizations misunderstand what is meant by size and how size information is used in process engineering
- Size becomes critical when an organization wishes to implement quantitative control

# Exploiting effort data

- How can effort data be exploited to define a unit that reflects the size of what is being produced?
- Once this unit is known, it can be refined with the collection of additional data, and used to derive ratios that will facilitate estimation, monitoring and control of future work

# Example

- Assume development of an application comprised of 4 features made up of various components
- Components categorized as follows
  - *Data processing (DP)*
  - *Internal transactions (IT)*
  - *External transactions (ET)*
  - *Reports or read-only screens (ROS)*
  - *Refreshable screens (RS)*

# Example (Cont'd)

- Following parameters initially defined based on experience

| Component type | Estimated # of work-hours |
|---|---|
| DP | 10 |
| IT | 8 |
| ET | 11 |
| ROS | 5 |
| RS | 16 |

| Value | Complexity |
|---|---|
| 0.5 | Simple |
| 1.0 | Medium |
| 1.5 | Complex |

# Example (Cont'd)

- Initially, the size of DP, IT, ET, ROS and RS are defined in terms of effort, on the basis of experience
  - *For instance, a nominal Data Processing component requires 10 w-h to develop, whereas a simple Data Processing component requires 5 w-h*
  - *A table is defined that takes into account complexity, which then becomes a representation of the size of a component*
- By multiplying the nominal effort by the complexity factor, it is possible to obtain the effort required to develop a given component

# Example (Cont'd)

- Feature 1
  - *3 DP (2 Simple; 1 Complex)*
  - *5 IT (1 Simple; 3 Medium; 1 Complex)*
- Feature 2
  - *1 DP (Medium)*
  - *3 IT (1 Simple; 2 Complex)*
  - *2 ROS (1 Simple; 1 Complex)*
  - *1 RS (Medium)*

# Example (Cont'd)

- Feature 3
  - *2 IT (1 Simple; 1 Medium)*
  - *3 ET (2 Simple; 1 Complex)*
  - *3 RS (2 Simple; 1 Complex)*
- Feature 4
  - *5 DP (2 Simple; 2 Medium; 1 Complex)*
  - *2 ET (2 Medium)*
  - *3 ROS (2 Medium; 1 Complex)*
  - *1 RS (Medium)*

# Size and Effort Estimation

- Application size expressed as a function of Component Types = # of Components of a given Type x Complexity

| Component Type | Simple (0.5) | Medium (1.0) | Complex (1.5) | Total Size | Effort (w-h) | Total Effort (w-h) |
|---|---|---|---|---|---|---|
| DP | 4 | 3 | 2 | 8 | 10 | 80 |
| IT | 3 | 4 | 3 | 10 | 8 | 80 |
| ET | 2 | 2 | 1 | 4.5 | 11 | 49.5 |
| ROS | 1 | 2 | 2 | 5.5 | 5 | 27.5 |
| RS | 2 | 2 | 1 | 4.5 | 16 | 72 |

# Size and Effort Estimation (Cont'd)

- The number 8 under Total size is obtained by multiplying 4*0.5, 3*1.0 and 2*1.5, and adding the three values
- In this case, it does not make much sense to add the total size for each component type to get the total system size
  - *The effort for each component type has not been normalized (this effort depends on the type), and adding total sizes for component types is like adding apples and oranges*
- However, that effort can be added to obtain total system effort, which amounts to 309 w-h

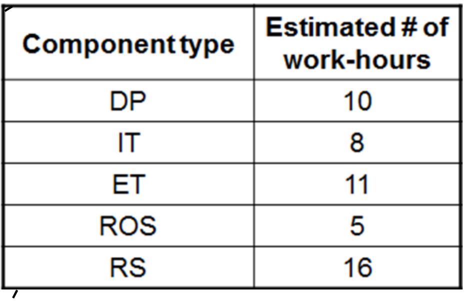
# Improving Size Estimation

- Limitations of this approach
  - *No standard component unit size*
  - *Number of defects per feature and effort per feature cannot easily be compared*
    - Depend on component type (e.g. DP, IT)

# Improving Size Estimation (Cont'd)

- **Define a standard component size**
  - *Arbitrarily define Component Unit size (CU)*
  - *Define Effort Ratio (ER) as 10 work-hours/CU*
    - Work-hours averaged over Component Types

| Component type | Weight |
|---|---|
| DP | 1 |
| IT | 0.8 |
| ET | 1.1 |
| ROS | 0.5 |
| RS | 1.6 |



| Component type | Estimated # of work-hours |
|---|---|
| DP | 10 |
| IT | 8 |
| ET | 11 |
| ROS | 5 |
| RS | 16 |

As data is collected, weights can be refined with time

# Improving Size Estimation (Cont'd)

- Size in CU = $\Sigma$types (# of Components of a given Type x Complexity x Weight)
- Effort = Size (CU) x ER (w-h/CU)

| Value | Complexity |
|---|---|
| 0.5 | Simple |
| 1.0 | Medium |
| 1.5 | Complex |

Feature 1
3 DP (2 Simple; 1 Complex)
5 IT (1 Simple; 3 Medium; 1 Complex)

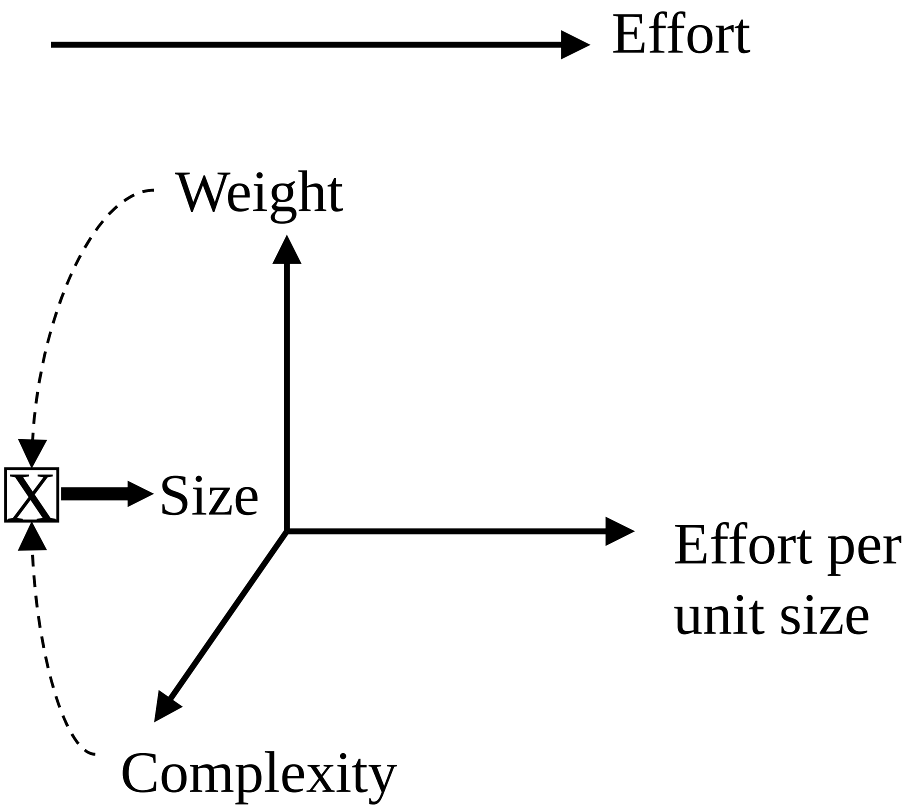
| Feature | DP<br>$W_1$=1<br>S-M-C | IT<br>$W_2$=.8<br>S-M-C | ET<br>$W_3$=1.1<br>S-M-C | ROS<br>$W_4$=.5<br>S-M-C | RS<br>$W_5$=1.6<br>S-M-C | Size (CUs) | Est. Effort (w-h) |
|---|---|---|---|---|---|---|---|
| 1 | 2-0-1 | 1-3-1 | 0-0-0 | 0-0-0 | 0-0-0 | 6.5 | 65 |
| 2 | 0-1-0 | 1-0-2 | 0-0-0 | 1-0-1 | 0-1-0 | 6.4 | 64 |
| 3 | 0-0-0 | 1-1-0 | 2-0-1 | 0-0-0 | 2-0-1 | 7.95 | 79.5 |
| 4 | 2-2-1 | 0-0-0 | 0-2-0 | 0-2-1 | 0-1-0 | 10.05 | 100.5 |

# Improving Size Estimation (Cont'd)

- The size of Feature 1 is equal to 2*0.5 *1+ 1*1.5*1 (for Data Processing components) + 1*0.5*0.8 + 3*1.0*0.8 + 1*1.5*0.8 (for Internal Transactions components) = 6.5 CUs
  - *Total system development effort adds up to 309 w-h, the same value obtained previously*
- In this case, because the effort has been normalized by defining it for each CU, adding the size of each feature translates into total application size
  - *Total application size amounts to 30.9 CUs, whereas in the previous case, total application size was 32.5, but 32.5 what?*

# Conceptualization

- From one dimension, namely effort, along which measurements were collected…
- Two additional dimensions are being added, namely weight and complexity, which even though still based on effort, will allow deriving size as actual data is collected to calibrate the values arbitrarily assigned to these dimensions



# Comparing with Actuals

- With data collected for 3 or more features, we can solve for Complexity factors Simple (S), Medium (M), and Complex (C)
  - $W_2 = 0.8W_1$
  - $W_3 = 1.1W_1$
  - $W_4 = 0.5W_1$
  - $W_5 = 1.6W_1$
- 4 Equations of the following type
  - *With N varying from 1 to 4, Effort Ratio = 10*

$$[W_1*(S*DP_S+M*DP_M+C*DP_C) + .8W_1*(S*IT_S+M*IT_M+C*IT_C) +$$

$$1.1W_1*(S*ET_S+M*ET_M+C*ET_C) + .5W_1*(S*ROS_S+M*ROS_M+C*ROS_C) +$$

$$1.6W_1*(S*RS_S+M*RS_M+C*RS_C)]*10 = \text{Actual Effort for Feature N}$$

# Comparing with Actuals (Cont'd)

- Initially, Simple (S), Medium (M) and Complex (C) were defined from experience and intuitively from the amount of effort invested in the development of these components
    - *The next step is to collect actuals in order to determine what the real values of Simple (S), Medium (M) and Complex (C) are*
- There are three unknowns: S, M and C
- At least three equations are needed in order to find solution for each, but more equations are desirable to remove random fluctuations

# Comparing with Actuals (Cont'd)

- This table shows 4 data sets that have been collected resulting in 4 equations to solve 3 unknowns

| Feature | DP<br>$W_1$=1<br>S-M-C | IT<br>$W_2$=.8<br>S-M-C | ET<br>$W_3$=1.1<br>S-M-C | ROS<br>$W_4$=.5<br>S-M-C | RS<br>$W_5$=1.6<br>S-M-C | Size<br>(CUs) | Actual<br>Effort<br>(w-h) |
|---|---|---|---|---|---|---|---|
| 1 | 2-0-1 | 1-3-1 | 0-0-0 | 0-0-0 | 0-0-0 | 6.5 | 60 |
| 2 | 0-1-0 | 1-0-2 | 0-0-0 | 1-0-1 | 0-1-0 | 6.4 | 70 |
| 3 | 0-0-0 | 1-1-0 | 2-0-1 | 0-0-0 | 2-0-1 | 7.95 | 85 |
| 4 | 2-2-1 | 0-0-0 | 0-2-0 | 0-2-1 | 0-1-0 | 10.05 | 112 |

# Determine Complexity Factors from Actuals

- **Solve for all 4 Features**
  1. *$[W_1*(2S+C) + W_2*(S+3M+C)]*10 = 60$*
  2. *$[W_1*(M) + W_2*(S+2C) + W_4*(S+C) + W_5*(M)]*10 = 70$*
  3. *$[W_2*(S+M) + W_3*(2S+C) + W_5*(2S+C)]*10 = 85$*
  4. *$[W_1*(2S+2M+C) + W_3*(2M) + W_4*(2M+C) + W_5*(M)]*10 = 112$*
- **Solution (linear approximation)**

| Value | Complexity |
|---|---|
| 0.54 | Simple |
| 1.18 | Medium |
| 1.32 | Complex |

# Determine Complexity Factors from Actuals (Cont'd)

- The value 10 shown for equation 1 is the effort ratio defined for a CU
  - *The value of 60 is the actual effort that has been obtained from time recorded by developers working on feature 1*
- The obvious solution is to calculate the average for each of the three values obtained for S, M, and C
- Based on these calculations, S is not 0.5 as previously estimated but 0.54; likewise, M and C are 1.18 and 1.32, respectively, not 1 and 1.5

# Determine Complexity Factors from Actuals (Cont'd)

- These calculations will typically be performed by the personnel having the responsibility of analyzing data
  - *Development personnel will provide raw data*
- Doing this exercise with collected data provides insight on its reliability
  - *If, after resolving these equations, it is found for example that Medium has a value larger than Complex, then it can be concluded that collected data was unreliable, and something has to be done to remedy the situation*

# Linear Approximation

- Solve for x and y
  - 2 unknowns, 3 equations

    $a\mathbf{x} + b\mathbf{y} = K$, where a, b, and K are constants

    $c\mathbf{x} + d\mathbf{y} = L$, where c, d, and L are constants

    $e\mathbf{x} + f\mathbf{y} = M$, where e, f, and M are constants

- Can also be expressed as

$$\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} K \\ L \\ M \end{pmatrix} \Rightarrow \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} \begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} \begin{pmatrix} K \\ L \\ M \end{pmatrix}$$

- Which reduces to

    $\mathbf{x}\,(a^2 + c^2 + e^2) + \mathbf{y}\,(ab + cd + ef) = aK + cL + eM$

    $\mathbf{x}\,(ab + cd + ef) + \mathbf{y}\,(b^2 + d^2 + f^2) = bK + dL + fM$

# Linear Approximation (Cont'd)

- **Solving 3 unknowns with four equations can easily be done manually**
  - ***However, if data for 50 features had been collected, this would result in having to find solutions for three unknowns with 50 equations, a tedious task at best***
- **The preceding slide shows how to use matrix algebra to solve for 2 unknowns (namely x and y) with three equations**
  - ***It can be generalized for any number of equations and solved with the help of Excel***

# Linear Approximation (Cont'd)

- The left-hand side of the second bullet is just a matrix representation of the three equations shown in bullet 1
- In the right-hand side of bullet 2, each side of the equation has been multiplied by the transpose of the leftmost matrix
  - *The transpose of a matrix consists of interchanging rows and columns, which is a standard Excel function*
- The third bullet is the result of the matrix operation, which is easily solved for x and y

# Determine Weight Factors from Actuals

- **Then, with data collected for 5 or more features, we can solve for weight factors $W_1$, $W_2$, $W_3$, $W_4$, and $W_5$**
  - ***Using values previously calculated for Complexity Factors S, M, and C, and Effort Ratio = 10***
- **L equations of the following type**
  - ***With L varying from 1 to a value $\geq$ 5***

$$[W_1*(S*DP_S+M*DP_M+C*DP_C) +$$
$$W_2*(S*IT_S+M*IT_M+C*IT_C) +$$
$$W_3*(S*ET_S+M*ET_M+C*ET_C) +$$
$$W_4*(S*ROS_S+M*ROS_M+C*ROS_C) +$$
$$W_5*(S*RS_S+M*RS_M+C*RS_C)]*\text{Effort Ratio} = \text{Actual Effort for Feature L}$$

# Determine Effort Ratio from Actuals

- Finally, with data collected for 1 or more feature, we can solve for Effort Ratio
  - *Using values previously calculated for Complexity Factors S, M and C, and for Weight Factors $W_1$, $W_2$, $W_3$, $W_4$, and $W_5$*
- P equations of the following type
  - *With P being a value $\geq$ 1*

$$
\begin{aligned}
&[W_1*(S*DP_S+M*DP_M+C*DP_C) + \\
&W_2*(S*IT_S+M*IT_M+C*IT_C) + \\
&W_3*(S*ET_S+M*ET_M+C*ET_C) + \\
&W_4*(S*ROS_S+M*ROS_M+C*ROS_C) + \\
&W_5*(S*RS_S+M*RS_M+C*RS_C)]*\text{Effort Ratio} = \text{Actual Effort for Feature P}
\end{aligned}
$$

# Recalibration

- **It is recommended to periodically recalibrate with new data the parameters used for deriving size**
  - ***To reflect, among other things, the improvement in know-how as applications are being developed***
- **Benchmarking is accomplished by measuring size for a sample of projects using one of the “standard” measurement units, and converting it into CUs**
- **Agile methods provide an opportunity to derive a unit size quickly as a result of frequent deliveries**