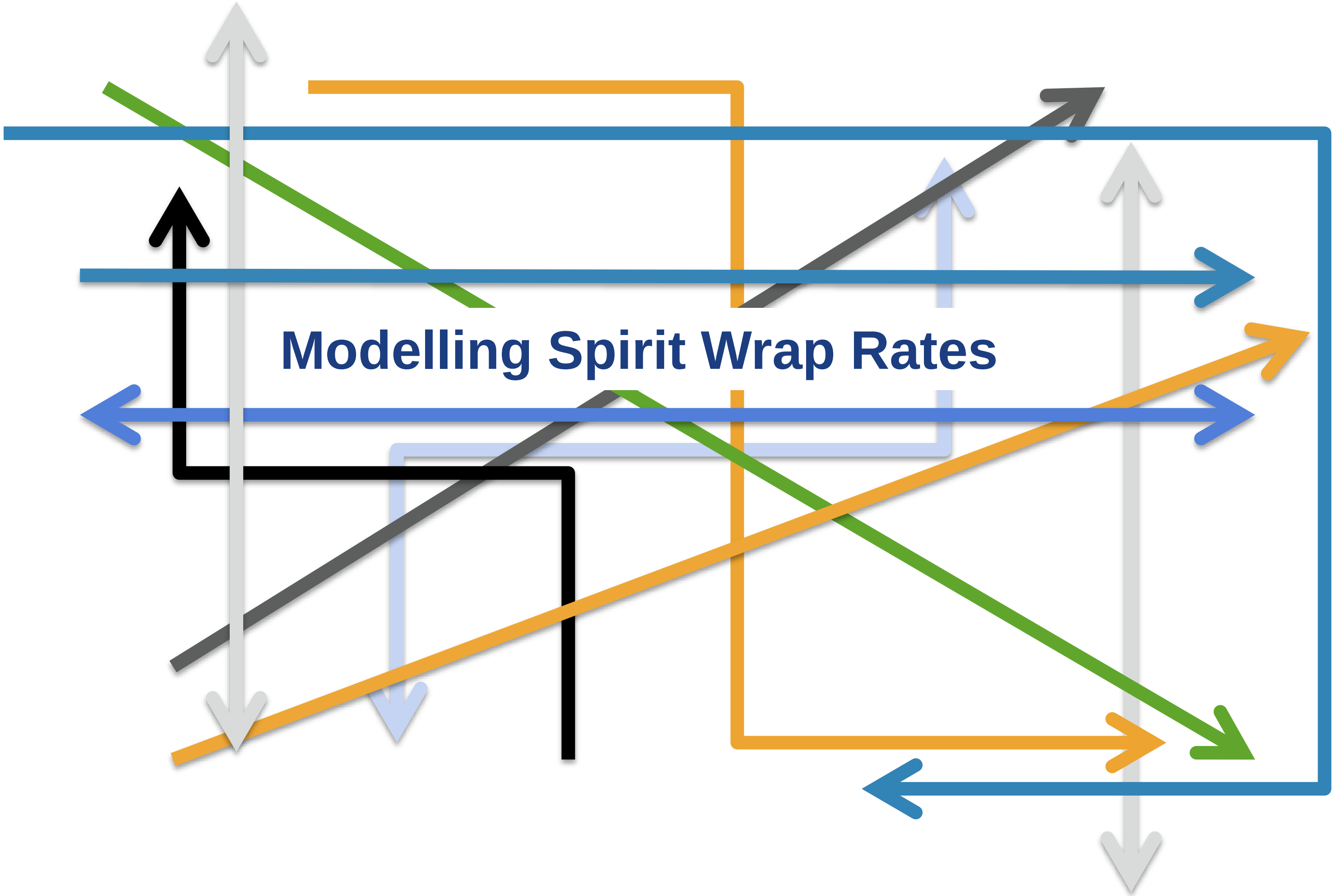
Building Dynamic Cost Estimating Models

Miranda Jones, Spirit AeroSystems

June 2018

SPIRIT AEROSYSTEMS®

Purpose



Modelling Spirit Wrap Rates

SPIRIT AEROSYSTEMS

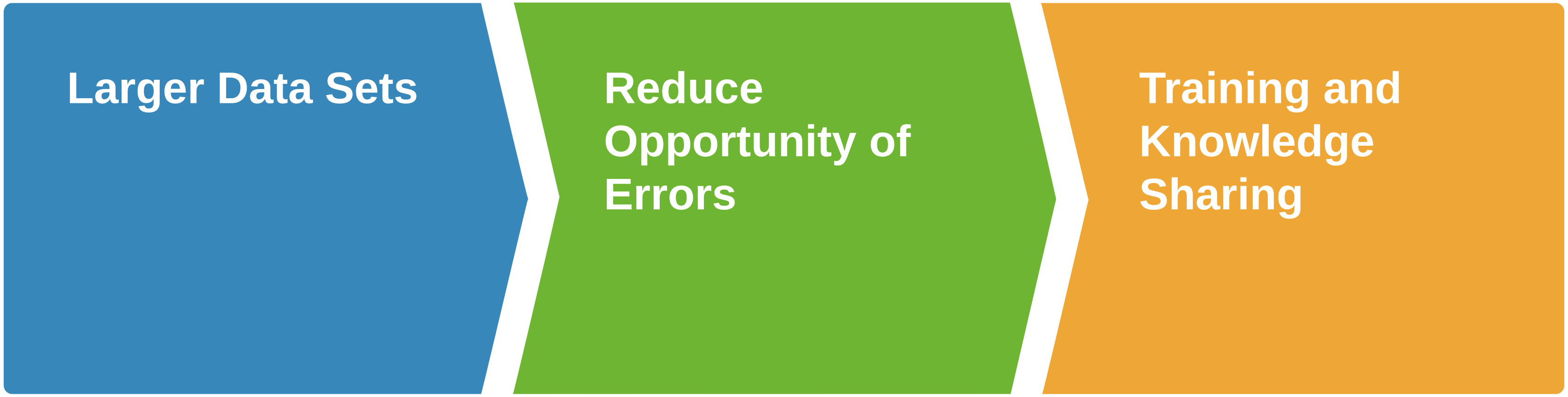
What are Dynamic Cost Estimating Models?

- **Dynamic**: *adjective* (a process or system) characterized by constant change, activity or progress

- Dynamic Cost Estimating Models can
  - Adapt to a variety of inputs
    - Quantity and Arrangement
  - Accommodate changes to model assumptions
    - End user input
    - Fundamental formulas used
  - Provide flexibility and transparency to user
    - End user understands how data is transformed or summarized
    - End user acknowledges data form and organization

Source: Definition of Dynamic from http://www.Oxforddictionaries.com

Why Develop Dynamic Models?



Larger Data Sets

Reduce Opportunity of Errors

Training and Knowledge Sharing

Reduce time and effort to create more detailed estimates or analysis reports

Determine Model Requirements

- Think through some questions before creating model
  - Who is the end user of the model?
    - What aids do they need?
    - What background do they have?
  - What does the input data look like?
    - How does the format vary?
    - Is the data coming from a software package or free-form files?
  - What type of output is required?
    - Does the information need to be viewed in various ways?
    - Who is viewing the output?
    - Will the output of this model need to interact with any other models?

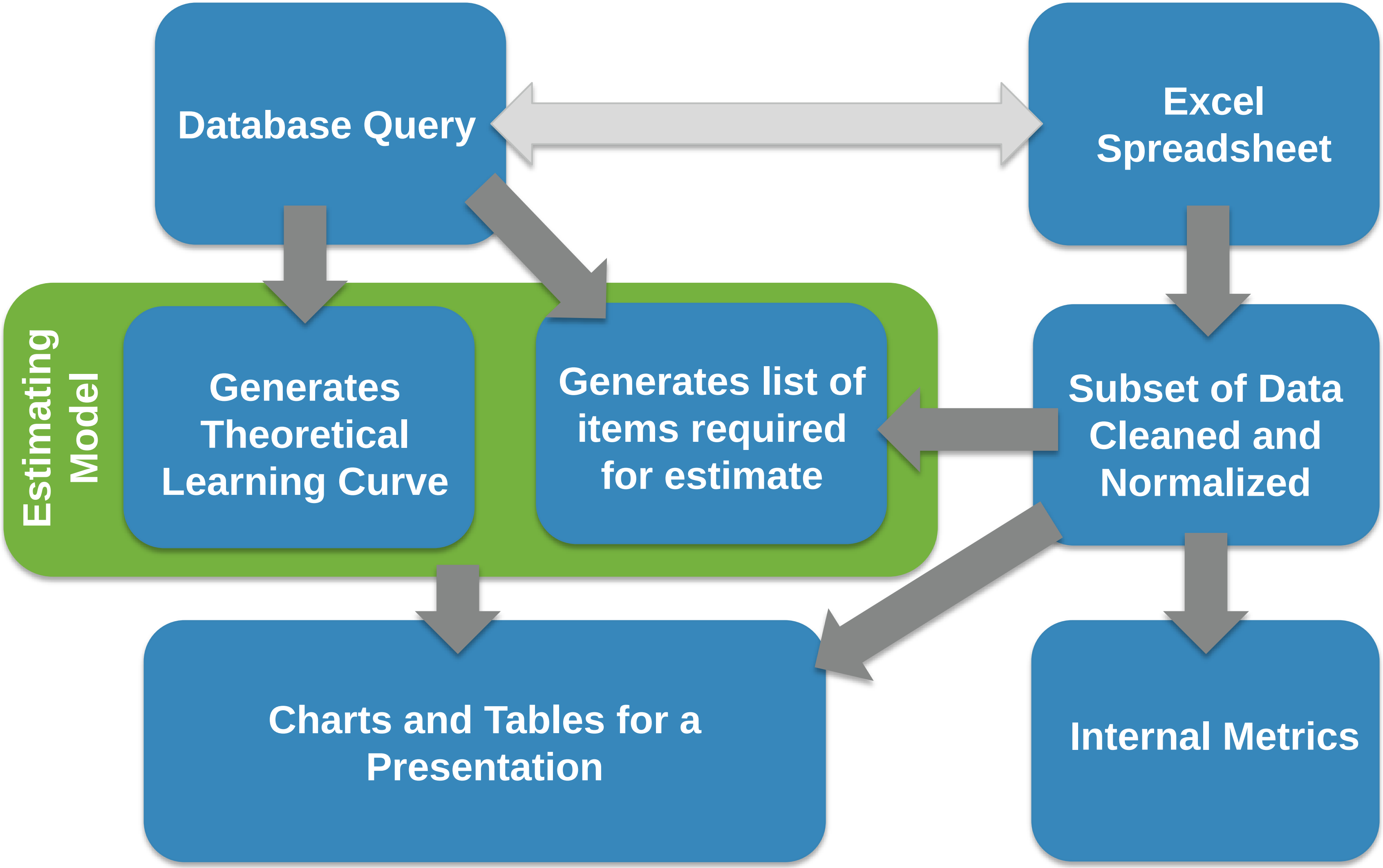
Where did the data come from and where is it going?

Determine Model Requirements

- How is the data being transformed?
  - Analogy
  - Parametric Model
  - Build-up
  - Extrapolation from Actuals
  - Learning Curves
  - Slicing/Dicing of Data

- **How can the manipulation of the data be generalized?**

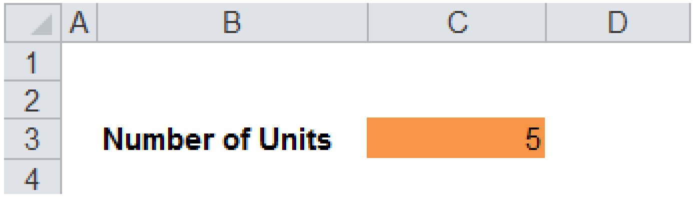
Determine Model Requirements


Database Query
Excel Spreadsheet
Estimating Model
Generates Theoretical Learning Curve
Generates list of items required for estimate
Subset of Data Cleaned and Normalized
Charts and Tables for a Presentation
Internal Metrics

Methods and Techniques

Adapting to a Variety of Inputs – Quantity

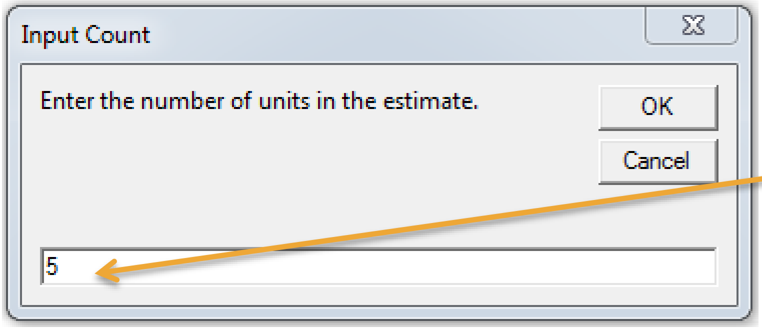
Excel Spreadsheet:



	A	B	C	D
1				
2				
3		**Number of Units**	5	
4				

Excel VBA:

```
input_count = InputBox("Enter the number of units in the estimate.", "Input Count")
```



Input Count

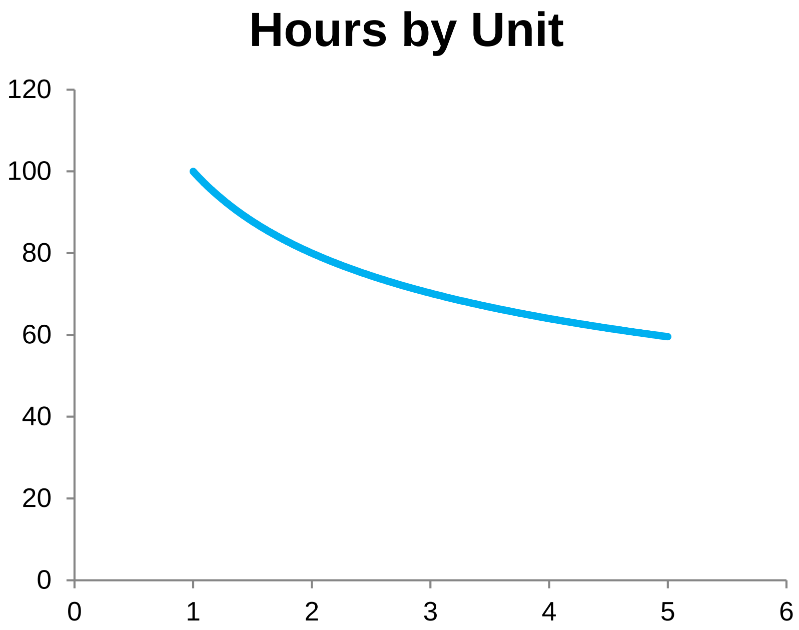
Enter the number of units in the estimate.

5

Input box could be pre-populated based on a counting criteria, but user interaction may be more desirable.

Methods and Techniques

Adapting to a Variety of Inputs – Quantity



Unit	Hours
1	100.00
2	80.00
3	70.21
4	64.00
5	59.56
6	56.17
7	53.45
8	51.20
9	49.29
10	47.65
11	46.21
12	44.93
13	43.79
14	42.76
15	41.82

Excel Spreadsheet:

Create a Named Range – Unit_Count

=OFFSET('Learning Curve Data'!$B$4,0,0,Inputs!$C$3,1)

Reference Sheet Name and Named Range in Chart Series X Values

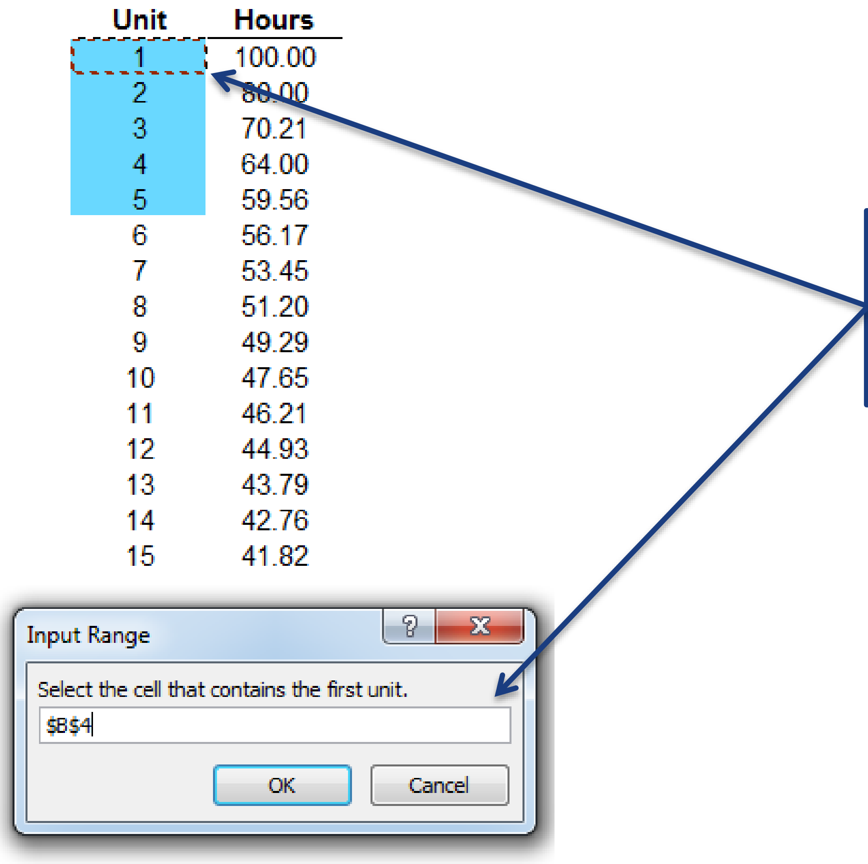
='ICEAA 2018 Presentation.xlsm'!Unit_Count

Methods and Techniques

Adapting to a Variety of Inputs – Arrangement

Excel VBA:

```
data_range = Application.InputBox(“Select the cell that contains the first unit.", “Input Range“, Type:=8)
```



Unit	Hours
1	100.00
2	80.00
3	70.21
4	64.00
5	59.56
6	56.17
7	53.45
8	51.20
9	49.29
10	47.65
11	46.21
12	44.93
13	43.79
14	42.76
15	41.82

Input Range

Select the cell that contains the first unit.

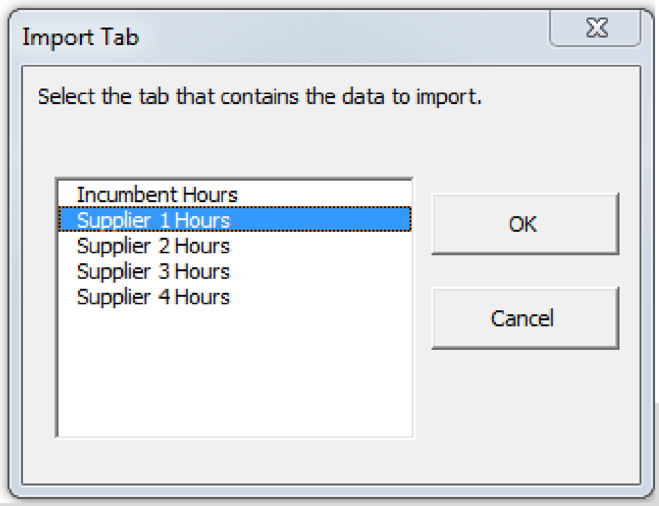
$B$4

The user selects cell B4.
The cell reference shows up in the input box, and the cell is outlined.

Methods and Techniques

Adapting to a Variety of Inputs – Arrangement

- Create a dynamic list of tab names



Import Tab

Select the tab that contains the data to import.

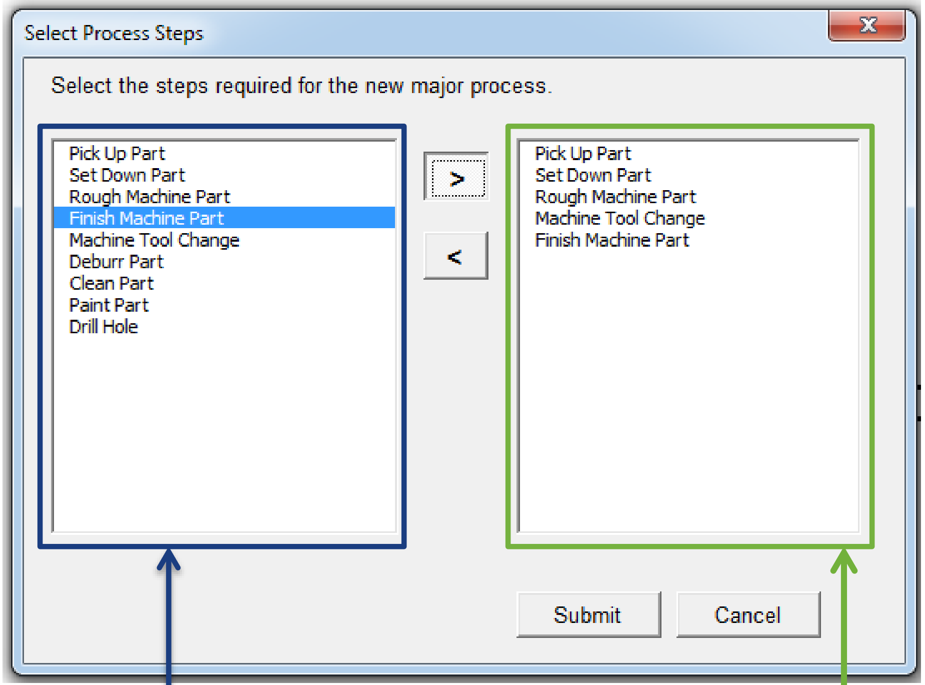
Incumbent Hours
Supplier 1 Hours
Supplier 2 Hours
Supplier 3 Hours
Supplier 4 Hours

Excel VBA:

```
Private Sub UserForm_Initialize()
Dim i As Integer
    For i = 1 To ActiveWorkbook.Sheets.Count 'counts number of sheets in the active workbook
        If ActiveWorkbook.Sheets(i).Tab.Color = RGB(247, 150, 70) Then
         'adds sheet name to list if it is colored this shade of orange
            LB_Tabs.AddItem ActiveWorkbook.Sheets(i).Name
        Else
            'do nothing (don't add it to the list)
        End If
    Next i
End Sub
```

Methods and Techniques

Adapting to Changes in Model Assumptions – End User Input



Select Process Steps

Select the steps required for the new major process.

Pick Up Part
Set Down Part
Rough Machine Part
Finish Machine Part
Machine Tool Change
Deburr Part
Clean Part
Paint Part
Drill Hole

Pick Up Part
Set Down Part
Rough Machine Part
Machine Tool Change
Finish Machine Part

Excel VBA:

```
Private Sub Add_Button_Click()
Dim i As Long
For i = 0 To LB_Master_List.ListCount - 1
If LB_Master_List.Selected(i) Then
        LB_Sub_List.AddItem LB_Master_List.List(i, 0)
Else
        'do nothing
End If
Next
End Sub
```

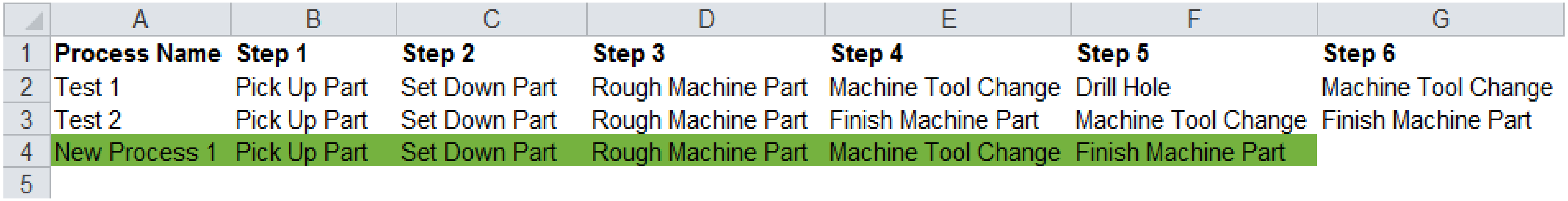
List box is populated by pre-determined list of options in spreadsheet.

List box is populated by user selections from the list box on the left of the form.

Methods and Techniques

Adapting to Changes in Model Assumptions – End User Input

- List box returns selected values to the estimating template in Excel



	A	B	C	D	E	F	G
1	**Process Name**	**Step 1**	**Step 2**	**Step 3**	**Step 4**	**Step 5**	**Step 6**
2	Test 1	Pick Up Part	Set Down Part	Rough Machine Part	Machine Tool Change	Drill Hole	Machine Tool Change
3	Test 2	Pick Up Part	Set Down Part	Rough Machine Part	Finish Machine Part	Machine Tool Change	Finish Machine Part
4	New Process 1	Pick Up Part	Set Down Part	Rough Machine Part	Machine Tool Change	Finish Machine Part	
5							

- New Process 1 is now a process type that the user can utilize in future estimates

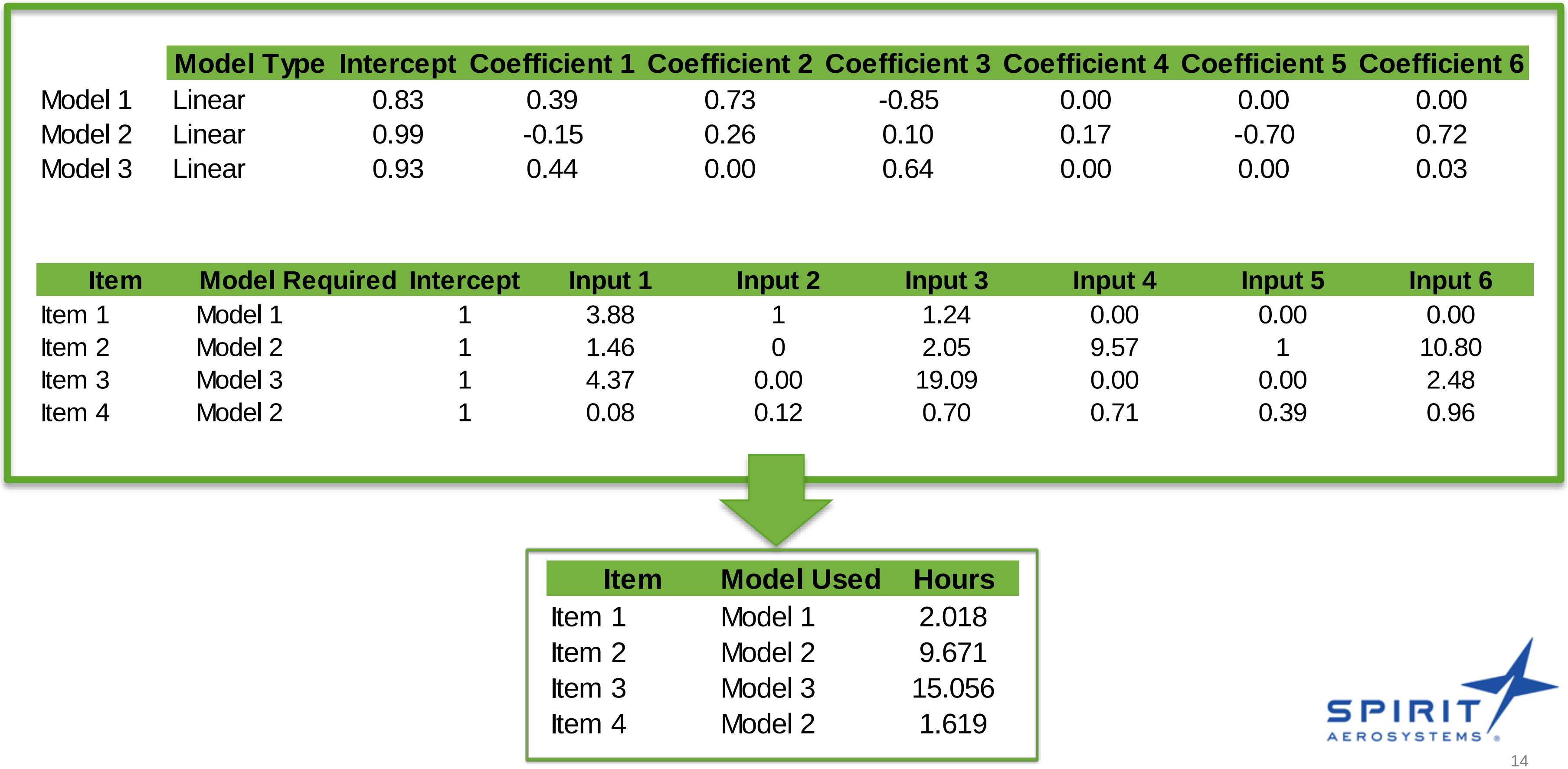
Methods and Techniques

Adapting to Changes in Model Assumptions – Fundamental Formulas Used

- Suite of Multivariate Parametric Models

$$\hat{y} = \hat{a} + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \cdots + \hat{b}_n x_n$$

- Generalize application of models using the Excel SUMPRODUCT function



	Model Type	Intercept	Coefficient 1	Coefficient 2	Coefficient 3	Coefficient 4	Coefficient 5	Coefficient 6
Model 1	Linear	0.83	0.39	0.73	-0.85	0.00	0.00	0.00
Model 2	Linear	0.99	-0.15	0.26	0.10	0.17	-0.70	0.72
Model 3	Linear	0.93	0.44	0.00	0.64	0.00	0.00	0.03

Item	Model Required	Intercept	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6
Item 1	Model 1	1	3.88	1	1.24	0.00	0.00	0.00
Item 2	Model 2	1	1.46	0	2.05	9.57	1	10.80
Item 3	Model 3	1	4.37	0.00	19.09	0.00	0.00	2.48
Item 4	Model 2	1	0.08	0.12	0.70	0.71	0.39	0.96

Item	Model Used	Hours
Item 1	Model 1	2.018
Item 2	Model 2	9.671
Item 3	Model 3	15.056
Item 4	Model 2	1.619

Methods and Techniques

Adapting to Changes in Model Assumptions – Fundamental Formulas Used

- Parametric Models

Excel VBA:

```
Dim j as Integer
Dim Inputs() As Double ‘Creates inputs array, but does not specify dimension
j=1
While Sheets(“Model Inputs”).Cells(2,j) <>””
        If Sheets(“Model Inputs”).Cells(2,j)  <> “”
                Redim Inputs(j-1) As Double ‘adds another dimension to array
                j = j+1
        Else
                ‘do nothing
        End If
Wend
```

Dynamically build array so the VBA script does not have to be adjusted as more models are added to the template.

```
Dim Coefficients(j-1) As Double ‘sets dimension to array j-1
For A = 1 To j-1
Coefficients(A) = Application.WorksheetFunction.VLookup(Model, Sheets("Model Coefficients").Range("ModCoef"), A + 1, False)
Next A

For A = 1 To j-1
Inputs(A) = Application.WorksheetFunction.VLookup(Model, Sheets("Model Inputs").Range("ModInputs"), A + 1, False)
Next A

Sheets(“Sheet1”).Range(“A1”) = Application.WorksheetFunction.SumProduct(Coefficients,Inputs)
```

Methods and Techniques

Adapting to Changes in Model Assumptions – Fundamental Formulas Used

- Parametric Models

R:

```
#This function applies a parametric model to the input data and varies a specified input to show the sensitivity of the input on the estimate

#a = An m x n matrix that contains input data for the parametric model, 1st column contains item name
#b = An m x (n-1) matrix that contains the coefficients of the parametric model to be applied in each rows

Apply_Models <- function(a,b){
        #change a into a data frame
        Model_Inputs_DF <-data.frame(a)

        #change b into a data frame
        Model_Coefficients_DF <-data.frame(b)

        #first column in the input data
        Model_Ouptuts <- data.frame(a[,1])
        for(j in 2:e)
        {Model_Outputs[,j] <- a[,j]*b[,j-1]}
}
```

$$\text{Model Inputs} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

$$\text{Model Coefficients} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n-1} \\ x_{21} & x_{22} & \cdots & x_{2n-1} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn-1} \end{bmatrix}$$

R data frame entries can be indexed by row and/or column
Model_Outputs[i,j]

SPIRIT AEROSYSTEMS

Methods and Techniques

Adapting to Changes in Model Assumptions – Fundamental Formulas Used

- Parametric Models

R:

```
#Read in external data and create variables
Model_Inputs <- read.xlsx(“ICEAA 2018 Conference”, sheetName = “Model Inputs”)

#Read in external data and create variables
Model_Coefficients <- read.xlsx(“ICEAA 2018 Conference”, sheetName = “Model Coefficients”)

#Apply function
Apply_Model(Model_Inputs,Model_Coefficients)

#Create an excel file with the output
Write.xlsx(Model_Outputs, file = “ICEAA 2018 Conference – Output”, sheetName=“Model Outputs”,row.names = FALSE)
```

Results

- Significant reduction in flow time to create internal wrap rate reports
  - Before – 1 week to create tables for Wichita site
  - After – 2 days to create tables for all Spirit sites
- Development of new estimated wrap rates streamlined and consistently documented
  - Before – set up scenarios to run overnight
  - After – run scenarios real-time in reviews, if necessary
- Timely development of pricing and business cases for internal review
  - Inputs, outputs, and transformation of the data documented in a consistent manner
  - Internal leaders have high level of trust in models
    - Fewer reviews or go-backs due to miscommunication of assumptions and methodology

Questions