

Benjamin Bailey  
Brigham Young University  
MBA614 Spreadsheet Automation  
14APR2015

## **FINAL PROJECT**

### **EXECUTIVE SUMMARY**

The relevant business for this project is one with continuous, repetitive operations - such as a manufacturing line or distribution center - without frequent, large variation. This type of business is one where the current state value stream can be measured as few as one to three times and still be sufficiently accurate as a baseline for making significant improvements in the process (a.k.a. system or product line).

The tool built for this project is called a Value Stream Compiler. It enables an observer to use a touchscreen tablet device running Microsoft Excel to quickly and efficiently capture the sequence of steps that constitute the business process of interest as well as the average time per unit for each step. Understanding this sequence and the time spent performing each step facilitates identifying process constraints (i.e. bottlenecks and cycle time), imbalances in labor standard work, and opportunities to apply Lean principles to reduce non-value added work (i.e. waste). In essence, quickly understanding the current state of a process value stream is the crucial foundation to making improvements that benefit the system as a whole. This helps to avoid making supposed improvements to a portion of the process that have negligible or even detrimental net effects on the larger system.

### **SYSTEM IMPLEMENTATION**

This compiler tool consists of a Microsoft Excel workbook containing three types of sheets: Instructions, Data Gathering, and Totals & Chart.

#### *A. Instructions*

- \*Provides step by step guidelines for using the tools
- \*Defines abbreviations used in the tool
- \*Provides tips to ensure various elements of the tool are used efficiently and consistently

VALUE STREAM COMPILER INSTRUCTIONS		ABBREVIATIONS
<b>PURPOSE</b> The purpose of this tool is to capture the time it currently takes to perform the steps of a supply chain process (i.e. involving the flow of material or information to meet customer needs).		Sec.: Seconds TPT: Throughput - CT: Cycle Time - th WC: Work Conten
<b>SCOPE</b> Identify the product line or boundaries of the overall process you want to analyze Identify what constitutes 1 unit of production for your purposes		
<b>PROCESS HIERARCHY TERMINOLOGY</b> Process (a.k.a. System or Product Line): the overall process you're analyzing Operation (a.k.a. Station): the sub-processes with typically clear delineation between each other Step (a.k.a. Activity or Task): the actions required to complete a sub-process		
<b>OPERATIONS</b> For consistency, use delineations between different operations based on the model router (aka work order). In other words, distinguish your sub-processes of your system the same way as the operators. This is important so others recognize the operations you are referring to.		
<b>ACTIVITY TYPES</b> Info: any activity where the operator is documenting the process or communicating information (e.g. writing on paper, typing on a computer, signing the work order, scanning component batch info, tracking today's output, etc) Setup: any activity where the operator is setting up the workstation (e.g. equioment start-up/settings, setting out the appropriate tools/templates)	DOES BATCH SIZE AFFECT THE TOTAL TIME PER BATCH FOR THIS ACTIVITY? No No	

**Figure 1 – Instructions Sheet**

**B. [Data Gathering] (e.g. ProductLine1, ProductLine2, ProductLine3, etc.)**

\*Where process steps & times are recorded by buttons embedded at the top of the sheet  
-The buttons are located on frozen pane rows so they stay visible when scrolling between process steps and times recorded below. I could alternatively put buttons like this among the Excel header tabs instead on frozen pane rows, to avoid messing up the width of the buttons when I resize column widths.

10	Operation Description	Opn	Activity Type	Start Date & Time	Sec. Elapsed	Step Description	Units	Sec./Unit	Operator	Sec./Unit/ Machine	Sec./Unit/ Machine
1	2	3	+	*	Opn	Info	Manual	New Row	10 Batch Size 915 TPT 114 CT 624 WC 47% Idle-Setup/WC 20% Dwell/TPT 12% Idle/TPT		
4	5	6	-	/	Idle	Setup	Dwell	Undo			
7	8	9	=								

**Figure 2 – Frozen Pane Macro Buttons**

-These buttons are linked to VBA Subroutines (macros). When the observer judges a new step has begun, they decide what type of step it is (e.g. Manual, Dwell, etc.) and press the corresponding button to record the step type as well as the current date & time (i.e. a time stamp) on the next available process step row regardless of which cell is currently active

-There is a separate button for each **activity type**. Each button records the current date and time in order to calculate time elapsed. However, each button records its distinct activity type as well as the most likely quantity of units processed, the time per unit per operator, and the time per unit per machine (dwell device).

~**Info**: operator documents the process or communicates information

~**Setup**: operator sets up the workstation so ready to process units

~**Manual**: operator is actively engaged in processing a unit

~**Dwell**: the unit is processed automatically without operator engagement

~**Idle**: the unit neither moving to next station nor having value added to it

-**It does not matter which cell is selected** on the sheet, when you press an activity type button, it will automatically record the current date and time (time stamp) in the next available process step row, automatically calculating the time elapsed since the start of the previous step

-There are also **numeric and operator buttons** for making manual edits, typically in the Time Elapsed field, to override inaccuracies due to:

- 1) missing entry of an intended time stamp,
- 2) picking up where leaving off the process from a previous day, or
- 3) a need to adjust how many units, operators, or machines were involved for the time period elapsed

The reason we have these buttons is that it's faster, with fewer touches, than using the Windows virtual keyboard. I think this idea of squeezing out seconds out of computer interactions certainly adds up over time and more importantly, it makes things easier on the user. Very small increments of time can be the difference between an enjoyable user experience and a deterrent that discourages using the item at all.

-For any given process step, it is important that the Value Stream Compiler captures the number of **units, operators, and machines** (of 1 type) so that the true capacity of the process can be compared to Takt time

-There is a "**New Row**" **button** for easily inserting a new row for a process step that was missed earlier or for breaking process steps into smaller segments. This was coded so as only to work below the frozen pane and to automatically populate several of the fields in a convenient manner.

-A good question to consider is why not **video record** the process of interest instead of using this tool? Video recordings of an operation are useful. They allow you to take your time in measuring a value stream precisely. Videos can serve as a common, unchanging lens if there are disagreements about standard times. On the other hand, if you video record a process, you still need to transfer your observations from the video into a spreadsheet to organize the data. If there are many steps in the process, the video can take a long time to watch and record your observations. The Value Stream Compiler can be used that way if high detail is desired. Usually, however, using the Value Stream Compiler to record the process sequence and times directly from observation, instead of from a video recording, should be sufficiently accurate. Using the Value Stream Compiler in this manner also enables speed and flexibility in moving back and forth through many processes in a much shorter time. Plus, tablets are less intimidating for operators than video recordings.

C. *[Totals & Chart] (e.g. ProductLine1Chart, ProductLine2Chart, etc.)*

\*Where process times are summarized into operations and totals for building a chart

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
39	TIMES PER UNIT								TIMES PER UNIT ADJUSTED FOR MULTIPLE OPERATORS & MACHINES						
40	Station	Info	Setup	Manual	Dwell	Idle	TOTAL		Operators	Machines	Per Operator	Per Machine	with Multi Operators /Machines TAKT		
41	1	1.20	2.20	26.00	0.00	0.00	29.40		1.00	0.00	29.40	0.00	29.40	57.6	
42	2	2.20	0.40	13.00	5.00	20.00	40.60		1.00	1.00	15.60	5.00	40.60	57.6	
43	3	0.50	0.00	31.00	0.00	0.00	31.50		2.97	0.00	10.83	0.00	10.83	57.6	
44	4	0.90	1.10	15.00	27.00	0.00	44.00		1.00	1.00	17.00	27.00	44.00	57.6	
45	5	0.60	0.50	18.00	12.00	7.00	38.10		1.00	2.00	19.10	6.00	32.10	57.6	
46	6	0.90	0.80	53.00	11.50	0.00	66.20		1.00	1.00	54.70	11.50	66.20	57.6	
47	7	0.80	0.60	38.00	12.00	0.00	51.40		1.00	1.00	39.40	12.00	51.40	57.6	
48	8	0.00	0.00	0.00	9.00	79.00	88.00		0.00	1.00	0.00	9.00	88.00	57.6	
49	9	11.40	1.10	48.00	3.20	0.00	63.70		1.40	1.00	39.50	3.20	42.70	57.6	
50	10	0.70	1.70	63.00	22.00	0.00	87.40		1.00	1.00	65.40	22.00	87.40	57.6	
51	11	1.00	0.50	28.00	31.00	0.00	60.50		1.00	2.00	29.50	15.50	45.00	57.6	
52	TOTAL	20.20	8.90	333.00	132.70	106.00	600.80		12.36	11.00	320.43	111.20	537.63		

Figure 3 – Processing Times by Operation

\*Visualization of the process times into a chart organized by operation (x-axis)

The general sequence for using the tool is as follows: start with a *Data Gathering* sheet, populate the time elapsed and activity type for the process steps, manually adjust values for Units, Operators, and Machines (1 Type), create a *Totals & Chart* sheet, analyze the Chart and the summary data. The following illustrative example measured the process times from a video about how contact lenses are made: <https://www.youtube.com/watch?v=62tha1Kxa2c>.

1. Create a new *Data Gathering* sheet

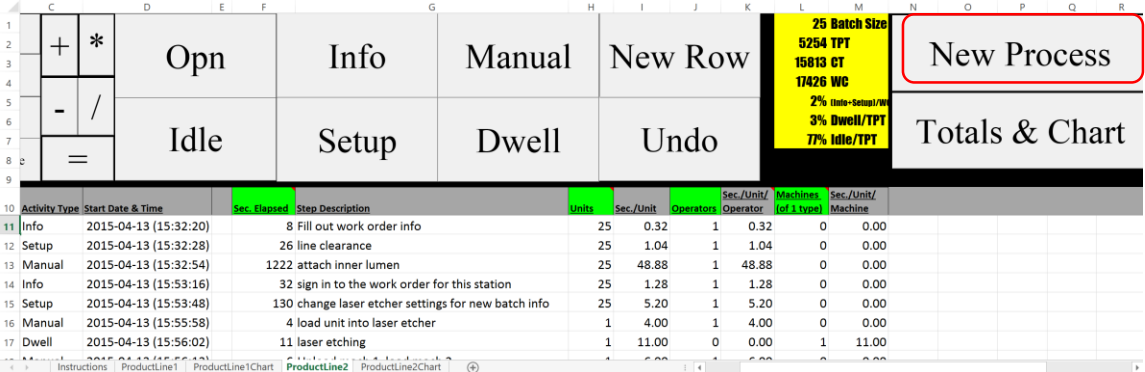


Figure 4 – The New Process Button

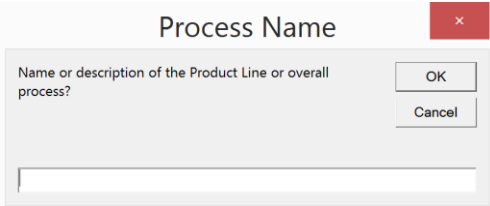


Figure 5 – Input Box to Name the New Process

2. Use the Activity Type buttons to record times and enter description of process steps

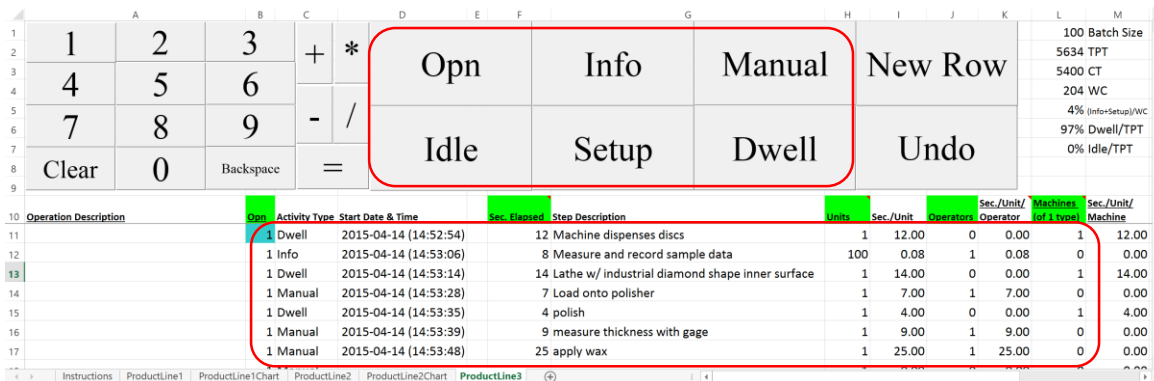


Figure 5 – The Activity Type Buttons and Process Steps

3. Create a new Totals & Chart sheet (enter appropriate values for the prompt pop-ups)

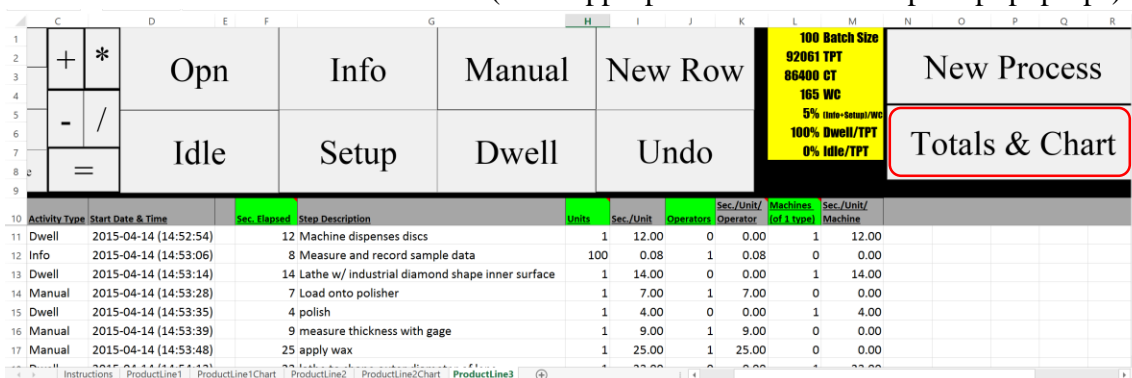


Figure 6 – The Totals & Chart Button

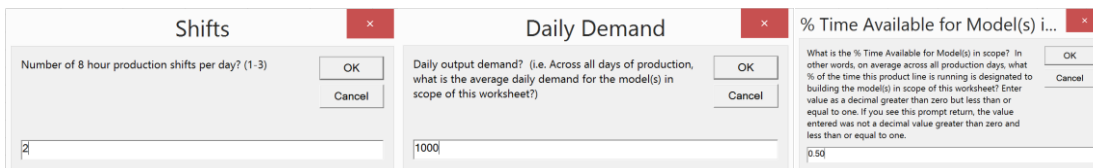


Figure 7 – Input Boxes Required to Calculate Takt Time

- Analyze the chart by identifying opportunities for improvement
  - Any operations with times that exceed Takt time (bar higher than the flat line)
  - High idle times (indicative of excess Work-In-Process inventory)
  - Poor Takt Balance (not building to demand) or Constraint Balance (choppy flow)

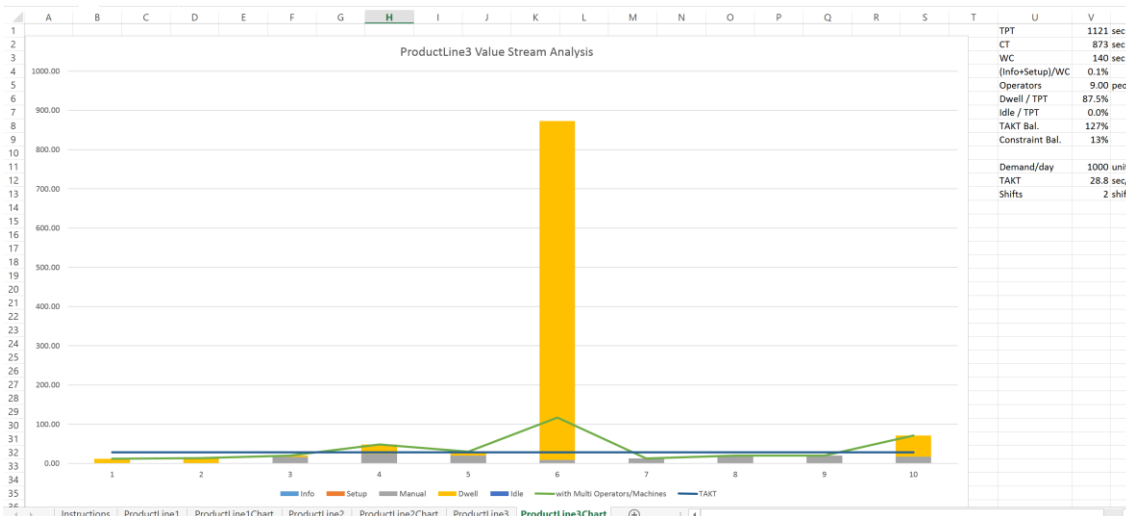


Figure 8 – Process Chart & Summary Info

TIMES PER UNIT							TIMES PER UNIT ADJUSTED FOR MULTIPLE OPERATORS & MACHINES								
Station	Info	Setup	Manual	Dwell	Idle	TOTAL	Operator	Machine	Operator	Machine	Operator	Machine	TAKT		
1	0.08	0.00	0.00	12.00	0.00	12.08	1.00	1.00	0.08	12.00	12.08	28.8		TPT	1121 sec
2	0.00	0.00	0.00	14.00	0.00	14.00	0.00	1.00	0.00	14.00	14.00	28.8		CT	873 sec
3	0.00	0.00	16.00	4.00	0.00	20.00	1.00	1.00	16.00	4.00	20.00	28.8		WC	140 sec
4	0.00	0.00	25.00	23.00	0.00	48.00	1.00	1.00	25.00	23.00	48.00	28.8		(Info+Setup)/WC	0.1%
5	0.00	0.00	20.00	10.00	0.00	30.00	1.00	1.00	20.00	10.00	30.00	28.8		Operators	9.00 peop
6	0.00	0.00	9.00	864.00	0.00	873.00	1.00	8.00	9.00	108.00	117.00	28.8		Dwell / TPT	87.5%
7	0.00	0.00	13.00	0.00	0.00	13.00	1.00	0.00	13.00	0.00	13.00	28.8		Idle / TPT	0.0%
8	0.00	0.00	20.00	0.00	0.00	20.00	1.00	0.00	20.00	0.00	20.00	28.8		TAKT Bal.	127%
9	0.00	0.00	19.80	0.00	0.00	19.80	1.00	0.00	19.80	0.00	19.80	28.8		Constraint Bal.	13%
10	0.00	0.00	17.00	54.00	0.00	71.00	1.00	1.00	17.00	54.00	71.00	28.8		Demand/day	1000 units
TOTAL	0.08	0.00	139.80	981.00	0.00	1120.88	9.00	14.00	139.88	225.00	364.88			TAKT	28.8 sec/
														Shifts	2 shift

Figure 9 – Processing Times by Operation & Summary Info

- For the example used above, some possible observations might be:
  - ~operation 6 is exceeding the Takt time, so demand is not being fully met (it is possible demand was being met with overtime or that there might have been more than 8 saline dwell machines available on the line... average overtime can be adjusted in this tool by using decimals when identifying the number of shifts such as 1.5 or 2.2 shifts)
  - ~Constraint Balance is only 13% so perhaps the work content could be adjusted between operations to improve flow
  - ~There is no idle time observed, so either the process is very efficient, or we just didn't see any due to editing of the YouTube video

## LEARNING POINTS & CONCEPTUAL DIFFICULTIES

### Time Data

\*Capturing **time elapsed** for a process step, I originally thought I should capture date & time when a step starts and when it stops. However, it is redundant to do both when I could simply capture the date & time stamp for when a step starts. To get the time

elapsed I just find the difference between the time stamp of when the first process step started and the time stamp of when the second process step started.

\*For the “Start Date & Time” field, I originally used VBA code to insert the “=NOW()” **formula** into the cells, but I realized this didn’t work because all the data updated to the current date & time whenever the formulas recalculated (e.g. when opening or saving the workbook). So I simply inserted the values of the “Now” VBA built-in function hardcoded into the cells instead.

\*I like displaying the values for reference, but a drawback is that a user might infer that they can change the “Sec. Elapsed” field by changing the values in the “Start Date & Time” field. Although the “Sec. Elapsed” field values are derived from the “Start Date & Time” field, it is that way only initially as the resulting values are hardcoded too. This ensured consistency in case the user needs to **manually overwrite values in the “Sec. Elapsed” field**, so that we don’t have a mix of some cells with a formula and some hardcoded. It’s just simpler to overwrite the “Sec. Elapsed” field values than to modify the “Start Date & Time” field values.

\*A particularly confusing obstacle in developing this tool was figuring out how to **format time data**.

- The “**Start Date & Time**” field I created custom formatting to show the full date as well as hours, minutes, and seconds in a 24-hour military time format.

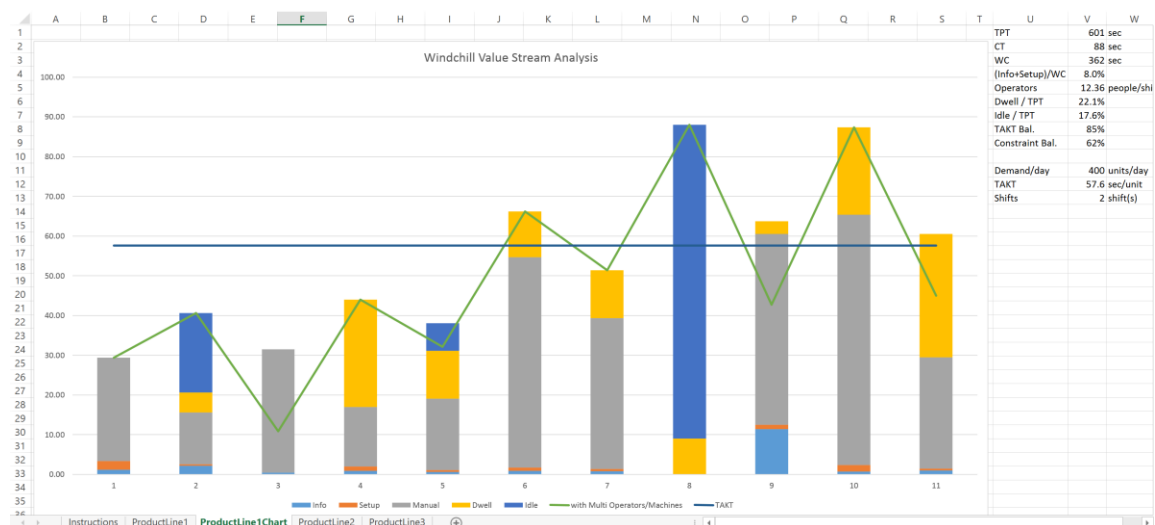
- The “**Sec. Elapsed**” field I originally created a custom format to show hours, minutes, and seconds (i.e. hh:mm:ss). However, this became cumbersome because when you click on the cell to edit the value, its true form is in time of day rather than a chronometer style. For example, a value one hour and thirty minutes would show up as 1:30AM. This did not inhibit use of mathematical formulas in connection with these cells. However, it did leave a blind spot. If a value were greater than 24 hours (e.g. a process step requires oven curing for 72 hours), that value still is displayed like a time of day (i.e. between 0001 and 2400 hours). Furthermore, it was cumbersome to manually overwrite these type of formatted cells; it could not be done with the numeric and operator buttons I had created. Thus, even though I like to see time data displayed with days, hours, minutes, and seconds, I decided it was clearer and easier to use by keeping all the time data in the tool in seconds format. Future improvements might be to have another column as reference that converts the time data from seconds into days, hours, minutes, and seconds. It is hard to quickly recognize how much time we’re talking about when values in seconds get above five hundred or so.

\***Significant digits** was another concern. I’ve experienced in the past how it can be frustrating when numbers don’t add up, especially with lots of decimals involved. For simplicity, I chose to make the “Sec. Elapsed” field round to the ones place. First, the real detailed time studies can be performed by industrial engineers. For this model, rounding primary measurements to the nearest 1 second helps keep it clearer and less in the weeds. This was also a necessity so my numeric and operator buttons at the top of the page could still work: they don’t work well with trying to incorporate decimals. However, for the calculated times that are derived out of this primary time measurement,

I've rounded to the hundredths place. This is a level of precision that most people can still wrap their heads around, yet also offers a fair level of precision for finding process step times per unit.

## Charts

\*I learned how to utilize InputBox better. When I create a *Data Gathering* sheet for a new Process (a.k.a. System or Product Line), I used InputBox to get the name of the Process so I could change the Name property of the new sheet (which is displayed on the tab at the bottom of the Excel window). As I started using InputBox to gather other important information from the user, I realized that a UserForm may be more desirable than InputBox. This is a potential improvement to consider.



**Figure 3 – Value Stream Analysis Chart with Summary Data**

\*When creating a *Totals & Chart* sheet I also used InputBox to collect crucial pieces of information: the daily demand quantity and the time available each day to build the model(s) in scope. I discovered a nice balance between detail and simplicity to find time available: I ask the user for how many shifts are used (assume 8 hour shifts) and I seek a % estimate for how much of the time the model(s) of interest are being produced. These are crucial pieces of information to make the Chart useful. It enables creation of a **Takt line** which tells us how much time we should have between finished units coming out at the end of the Process. As long as all the bars on the chart are below the Takt line, we can meet demand. If the bars are above the Takt line, we're still okay as long as the line graph – which accounts for multiple operators or duplicate equipment at the same work station. The Takt line lets us see at a glance whether we're able to meet customer demand without overtime, or whether we're overproducing, perhaps to increase inventories.

\*I learned the importance of using String and Object type variables when having to track different objects carefully. When the tool creates a *Totals & Chart* sheet, I originally had the VBA code build the chart using the data from my original *Data Gathering* sheet (e.g.



the *Sheet1* codename). However, as I created new sheets for different Product Lines, I had to be sure that the charts I created **pulled the data from the new Product Line sheet** and not from *Sheet1*. I learned to do this by concatenating a string variable in place of the references to *Sheet1*. I again learned the importance of timing with defining variables. I learned that when I defined a variable by saying it is equal to the name of the active sheet, that original name is kept in that variable's memory, even after I change the name of the Active Sheet. In addition to needing a string variable for these purposes, I found I also needed an object variable to replace instances where *Sheet1* appears as an object instead of as the prefix to a range reference.

\*When **recording macros** for creating charts, I learned several things:

- Macro recording is crucial, but don't clean up the code right away. There's fewer useless elements of code to remove by going back and recording the macro again until it's pretty smooth with as few unnecessary clicks as possible.
- Using absolute references is usually easier than relative references
- If the dimensions of the chart's data region can change (i.e. you might have more rows of data than columns of data or vice versa), we need to create an IF statement so that **if the number of rows are less than or equal to the number of columns** we must build out the data series manually in the code. Otherwise, the chart's x-axis will show the Activity Types instead of the Operation numbers.
- Since the **number of rows of data** (i.e. Operations) can change based on how many the user enters, we need a variable with a short 1 or 2-letter name to represent the number of rows. That short variable must be concatenated in to replace any cell absolute or relative cell references to the "last row" generated by the macro recorder. It sounded simple to do this, but there was a significant amount of troubleshooting and code revision necessary to adapt the tool to create charts with varying numbers of data rows (i.e. Operations).

### **Errors & Troubleshooting**

\*I built an Undo button into the tool, for the touchscreen convenience on a tablet, but I learned that the VBA code gave error messages when I tried to undo something that doesn't work, such as trying to undo a Subroutine macro that just ran. I added "On Error Resume Next" code, and I imagine this can be helpful in many other situations.

\*Due to the complexity of the tool, there are a higher amount of opportunities for the code to fail based on unanticipated user actions. Further time could be spent going through the tool to mistake proof the tool and avoid code errors. For example, the Input Boxes should have rules included into the code if a user selects Cancel instead of OK.

### **Expanding Capabilities Beyond VBA**

\*I considered trying to use DLL (Dynamic Link Library) in order add the following functionality to the tool, but I was not able to figure out how to do so. With more research I may be able to find ways to add this functionality in the future.

- Using VBA to capture my voice and convert it to text to avoid having to pull up the Windows virtual keyboard to type out process step descriptions

-Using VBA to use the camera on my table to take pictures and video that could be linked to a desired process step entered onto this Value Stream Compiler

**Other**

\*I learned the difference between the Activate method and the Select method is that you can Select multiple cells, but you can only have one Activated cell.

**ASSISTANCE RECEIVED**

\*I did not receive any significant assistance on this assignment. The code was developed by referring to class notes and examples, the course textbook, and the macro recorder.