

Matt Cox

4-14-15

## VBA Project: VitalSmarts Sales Forecasting Program

### I. Executive Summary

#### *Company and Problem Background*

VitalSmarts is a privately held corporate training company with headquarters in Provo, UT, which sells training programs, such as *Crucial Conversations Training*, that teach communication and leadership skills to a company's employees. VitalSmarts' executive leadership has recently asked that the Sales department provide monthly sales forecasts from each sales representative and client. This data is used by sales management and the company leadership to understand what revenue they should expect in a given month, depending on the percentage of confidence the sales rep has assigned to the given sales opportunity. In this way, management knows whether they need to push sales reps for more revenue, and they also know how to manage budgeting for the next month.

The Sales department had been creating sales forecasts for only 2 or 3 months, and the process was very manual. A sales rep would type their forecasted sales opportunities into an Excel workbook template, then send that workbook to their manager. The Sales Operations team would somehow aggregate the data from the 20+ sales reps' worksheets and present it to Sales management and the CEO each month.

#### *VBA Solution*

The solution I have created for this sales forecasting problem automates much of the reporting process in a clean user-friendly interface. The solution consists of two user-facing workbooks. One is a workbook for sales reps that allows them to enter, edit, delete, move, and track all their forecasted sales opportunities for the current month plus three months forward, as well as past opportunities. The second workbook serves as a "dashboard" for Sales management and the CEO, which lists all sales opportunities for all sales reps for the current month plus three months forward. The workbook allows management to filter the opportunities by month and by sales rep, and to see the total forecasted revenue plus the total adjusted forecasted revenue (adjusted for the percentage of expected close that the rep has assigned to the opportunity). Each sales rep will have their own workbook hosted on their computer, and each of those workbooks will send their data (the sales opportunities) to the managers' workbook through a third workbook that just acts as a sort of intermediate transfer database.

The new program created with VBA adds a lot of functionality for the sales rep without creating any more complexity. For management, the solution provides a clean, simple, automated dashboard where they can quickly understand the forecasted revenue for the month and can drill deeper if they want information for a specific sales rep's accounts.

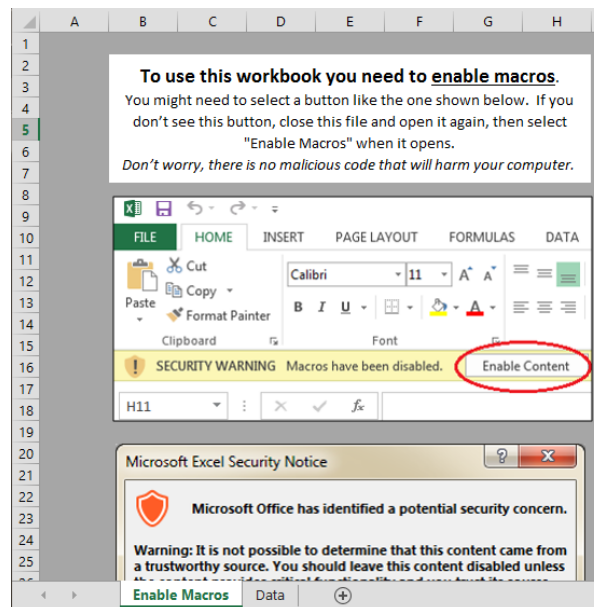
## II. Implementation Documentation

### a. Sales Rep Workbook

- i. The sales rep workbook consists of a single worksheet visible to the user that has a simple table listing the details of forecasted sales opportunities and three buttons that allow the rep to interact with the data (see below).

Sales Rep	Account	Product	Quantity	Price per Toolkit	Total Revenue	Month of Projected Close	% of Confidence
Matt Cox	MTN Cameroon	CC3	230	\$275.00	\$63,250.00	Apr-15	85%
Matt Cox	GSK	INF	50	\$275.00	\$13,750.00	Apr-15	95%
Matt Cox	Chevron	CC1	30	\$265.00	\$7,950.00	May-15	65%
Matt Cox	BYU	INF	20	\$200.00	\$4,000.00	May-15	100%
Matt Cox	BYU	INF	30	\$275.00	\$8,250.00	Jun-15	100%
Matt Cox	Voith	CC1	25	\$275.00	\$6,875.00	Jun-15	100%
Matt Cox	GSK	CC2	8	\$90.00	\$720.00	Jun-15	90%
Matt Cox	Voith	CC1	30	\$265.00	\$7,950.00	Jul-15	76%
Matt Cox	BYU	CC1	20	\$275.00	\$5,500.00	Jul-15	100%

- ii. *Enable Macros*: When the workbook is closed, code is run to hide the main “Data” worksheet and show a worksheet titled “Enable Macros” (see below). This way, if macros are disabled when the worksheet is opened again, this sheet will show and not let the user access their data without enabling macros. Since many users are not familiar with macros, this was a necessary step. Once they enable macros, code is run to make the “Enable Macros” sheet very hidden and show the “Data” sheet to the user.



- iii. *Sales Rep Name*: When a sales rep opens their workbook for the first time, it prompts them to type in their name, which is then stored in a cell on a very hidden sheet. The Sales Rep field then continues to populate from that cell every time the rep adds a new sales opportunity.
- iv. *Removing Old Data*: When the workbook is opened, a sub procedure will run to find all the listed sales opportunities in the table that are for a previous month

in time and then move that entire row to a very hidden sheet with old data, in order to keep the user's interface table up to date.

- v. *Add/Edit Userform*: The only way the sales rep can enter or edit sales opportunities is by using the userform, which is launched by pressing either the "ADD New Opportunity" or "EDIT Selected Opportunity" button. When adding a new sales opportunity the userform is empty (see below).

The screenshot shows a window titled "Add New Opportunity" with the VitalSmarts logo. The form contains the following fields and labels:

- Rep Name: Matt Cox
- Account: [dropdown menu]
- Product: [dropdown menu]
- Quantity: [text box]
- Price per toolkit: (without "\$") [text box]
- Month of projected close: [dropdown menu]
- Percent of confidence: (ex. 75) [text box]
- Total opportunity: [text box]
- Total expected opportunity: [text box]
- Buttons: Save, Cancel

1. The Account combobox allows the user to type a new account name, at which point the worksheet immediately and automatically saves the new account name to a very hidden "Accounts" worksheet and updates the options in the combobox to add the new account to the dropdown list. In this way, the userform will list any previous account name the user has entered in in the past.
2. The options for the Product combobox are also pulled from a very hidden "Products" sheet, but the options here are static – only the company's six products are listed. The user must select one of these options.
3. The Quantity, Price per toolkit, and Percent of confidence text boxes simply take in the numbers the user enters.
4. The Month of projected close combobox shows four options: the current month and three months forward. These options are only the name of the month (i.e. "April"). The combobox options are filled by a global 1x4 array that is filled when the workbook is first opened. The code identifies the current month, and the next three months then enters the month names into the array. This code accounts for when the four month options overlap a year so that it accurately produces "December, January, February, March", for example. The user then must select one of these month options in the combobox.
5. The userform dynamically shows the calculated total revenue and adjusted or expected total for the sales opportunity while the user enters and changes data on the form (see below)

**Edit Opportunity**

**VitalSmarts®**

Rep Name: Matt Cox

Account:  Total opportunity: **\$5500**

Product:  Total expected opportunity: **\$5500**

Quantity:

Price per toolkit:  (without "\$")

Month of projected close:

Percent of confidence:  (ex. 75)

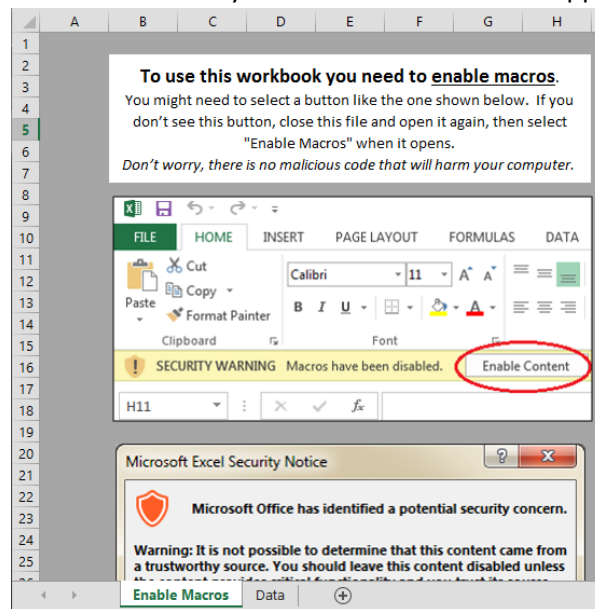
When the user changes either the Quantity or Price fields, both the total opportunity and total expected opportunity numbers automatically update. When they change the Percent of confidence field, the total expected opportunity number recalculates.

6. When the user clicks the “Save” button, the data from all the fields (including the revenue totals) is saved to the first empty row in the table, then the data in the table is immediately resorted by month then total revenue. If any field is left blank when the user clicks “Save”, a notification will pop up and tell them they must enter data in that field.
  7. *Edit Opportunity*: When the userform is activated by clicking on the “Edit Selected Opportunity” button, the code uses the data in the selected row of the table to populate the userform fields. If the selected cell is not within the range of the existing data, the user gets an error message reminding them they need to select a sales opportunity before pressing the “Edit” button.
- vi. *Transfer Data*: When the user closes the workbook the code takes all the current sales opportunities listed in the table and transfers them (without deleting them) to the first empty row in a similar table in a separate workbook called “transfer.xlsx”. This transfer workbook will be saved in a shared folder on the company’s network drive so that all the sales reps’ workbooks can access it. When the code starts to run it first checks whether the transfer workbook is open (i.e. someone else happened to be transferring the data at the same time) before it continues on with the code. If the transfer workbook is already open, the user gets a message asking them to try to transfer the data again. This entire transfer process happens quickly and in the background so the user doesn’t see any strange things happening.
- b. **Managers Workbook**
- i. The managers workbook consists of a single visible worksheet (see below) with a table showing all current sales opportunities for all reps, a couple comboboxes and a button that allow the manager to manipulate the view of the data, and a dynamically changing area that shows the full and the adjusted total revenue

forecasts for the opportunities listed on the page.

Select month: ALL MONTHS		View <input checked="" type="radio"/> See ALL opportunities <input type="radio"/> See only top 10 opportunities			
Select Sales rep: ALL REPS		Update Data			
<b>Total Forecast:</b>		<b>\$9,390,540</b>	<b>Total Weighted Forecast:</b>	<b>\$7,684,274</b>	
Sales Rep	Account	Total Revenue	Month of Projected Close	% of Confidence	Total Weighted Revenue
Matt Cox	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50
Matt Cox	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50
Matt Cox	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50
Tim Barlow	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50
Tim Barlow	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50
Kyle Moosmann	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50
Kyle Moosmann	MTN Cameroon	\$63,250.00	Apr-15	85%	\$53,762.50

- ii. *Enable Macros:* This workbook has the same “enable macros” safeguard as the rep workbook. When the workbook is closed, code is run to hide the main “Current Opportunities” worksheet and show a worksheet titled “Enable Macros” (see below). This way, if macros are disabled when the worksheet is opened again, this sheet will show and not let the user access their data without enabling macros. Once macros are enabled, code is run to make the “Enable Macros” sheet very hidden and the “Current Opportunities” sheet visible.

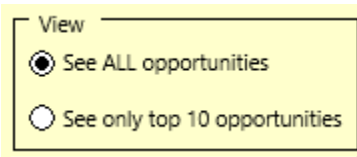


- iii. *Importing Data:* When this workbook is opened it automatically goes and gets all the data from the transfer workbook (accessible through the network drive), puts it in a hidden “all data” sheet in the managers workbook, and deletes all the data from the transfer workbook. All this is done out of sight of the user.
- iv. *Updating Data:* The workbook will then run code to delete any previous months’ sales opportunities from the aggregate data in the “all data” sheet, then it moves the remaining current data to the main worksheet where the user will be working. Since the “all data” sheet is not visible to the user, this move is the only way they can view the data.

1. *Preparing the workbook filters.* The code then goes through the range of the current data, identifies all the sales reps names, fills all the names into a very hidden worksheet, sorts them, and removes duplicates. This places the list of unique sales rep names from the data in the right place for the combobox on the sheet to pull in the names as options. The month names for the month combobox on the worksheet are also filled at this point by populating five cells on a very hidden “months” sheet based on a global array that stores the names of the current month plus three months forward. (This global array is designed the same as it is in the rep workbook.) The combobox on the worksheet then points to those cells for its list of data.
  2. *Calculating Forecasts.* The final step in updating the data in the worksheet is calculating the total forecasted and total adjusted forecasted revenue. At this point, code is run that totals the visible rows on the worksheet for both total forecast and total weighted forecast.
- v. *Using Filters to Analyze the Data.* The sheet is designed with two main filters the user can change to analyze the data.
1. *Months Filter.* The Months filter has five options: ALL MONTHS, which shows the data for all four months; the current month; and the next three months forward. This allows management to view total forecasted revenue for the current month or any of the coming months reps for which reps have entered data. As this combobox is changed, a code is run that applies an auto filter to the range of all the data and shows only those opportunities listed in the selected month. The ALL MONTHS option clears the filter so that all opportunities are visible. The total forecast and total weighted forecast are then recalculated based only on the visible rows. This way the user can see only the sales opportunities that are in the selected month and the forecasted revenue specific to that same month.
  2. *Sales Rep Filter.* The sales rep filter has an option to see ALL REPS, which works the same way as the ALL MONTHS filter, and also an option for each sales rep listed in the table of data. Just as the months filter, when the user selects a specific sales rep from the sales rep filter an auto filter is applied (or updated if it has already been applied through the months filter) to only show the rows with the selected rep’s name. The forecasted revenue totals are also then updated to show the sum of only the visible sales opportunities.  
Another feature used when a sales rep is selected un-hides three columns of data that have more specific information for the sales opportunities (product, quantity, price per unit). This allows sales managers to see specific details about their reps’ sales opportunities, while also allowing top level management to just see the most

important data when they're looking at the forecast for the entire company.

3. *Removing Filter Arrows.* When the month and sales rep filters are applied to the data and the auto filter is created, the auto filter arrows will show in the header row of the table. The worksheet is set up to make these arrows not visible to the user (so as not to distract them from the programmed filters). Because the filter is left applied to the sheet after it is used once, there is code that runs on the open workbook event to remove these filters.
4. *Top Ten Opportunities Option.* Some managers are only interested in seeing the top ten opportunities for a given month or a given sales rep, so the worksheet includes a toggle button to show either all opportunities or only the top 10 (see below). When the top 10 option is selected, the code makes only the first 10 rows in the table visible to the user and hides the rest. To see all opportunities, the code makes all rows in the table visible.



5. *Filters Working Together.* Because the months and sales rep filters use the Excel auto filter functionality, both of these filters can be applied at the same time very easily. This means the user can view opportunities for a single rep in a single month, or opportunities for a specific rep in all months, etc.  
The top 10 option also works when a month or sales rep filter is applied to the table. This is accomplished by selecting the first ten rows that are *visible* in the data and keeping those visible, then hiding all the rest. In essence, this applies another filter to the data without using the auto filter option (since auto filter does not allow you to see only a certain number of rows).
- vi. *Closing and Resetting the Workbook.* When the workbook is closed, the enable macros sheet is again shown and the current opportunities sheet is hidden, as described above. Also when the workbook is closed, a sub procedure is run to “reset” the current opportunities worksheet. A user will likely leave filters on when they close the workbook, which will make the data import not work properly when the workbook is opened again. Therefore, the “reset” code sets the months and sales rep filters to ALL MONTHS and ALL REPS and also turns off the top 10 toggle option.

### III. My Learning and Conceptual Difficulties Encountered

#### a. *What I Learned*

- i. *Referencing cells and ranges on inactive sheets and workbooks.* Both of my workbooks involve extensive referencing to cells and ranges on inactive sheets

and sometimes workbooks. I quickly learned that you can't refer to a range on an inactive sheet by just entering "Range(...)", because it will default to the range on the active sheet. This meant I had to create fairly long code to always include the sheet name, but I became very used to using codenames for sheets and variables for ranges.

- ii. *Auto filter.* The main functionality in my managers workbook relies heavily on filtering to be able to sort through the sales opportunities data in a logical way. I had first done this with a For Loop and If statement that went through each row in the data, but this became very slow as the number of rows increased to almost 600. I finally learned that I could use VBA to apply and manipulate Excel's auto filter functionality, which made for significantly faster code. I'm still not sure I understand all the syntax for the auto filter when filtering on two values, or especially how the auto filter itself turns on and off, but what I did learn was a perfect solution to my problem.
  - iii. *Getting feedback early.* When I first spoke with my client about the problem they were facing, I did a good job at asking question to understand what they were really looking for and what functionality would be necessary and what would be a nice-to-have. At the end, though, I ran out of time to get feedback from the client before presenting my "final" version to them, and they pointed out a lot of items that they would like to see in the program. I learned that when working with a client it's very important to get user feedback early on in the process to make sure all features address the needs.
  - iv. *Coding takes a lot longer than I thought.* When I started this project it seemed fairly simple to me: create two workbooks with tables, a basic userform, and some data filters, and send data from one workbook to the other. Very quickly, however, I discovered that the syntax and logic to make all that work was quite complicated. I learned that I need to be better at debugging and identifying bad code, because I spent a lot of time hunting down problems that I didn't understand in my code.  
Also, I realized I needed to make my code "fool proof" and include data validation and other measures to make sure any user could use it without breaking the code. Trying to do this took a lot of time and in the end I wasn't able to build all the data validation code I would have liked.
  - v. *Still, I like writing code.* Although this project took a lot of time and frustration, I also learned that I like writing code. I often preferred to work on this project over writing a paper or doing reading for other classes. There's something fulfilling about getting a small piece of code to work, even just to delete a row in a spreadsheet.
- b. *Conceptual Difficulties*
- i. *Counting filtered rows.* As described earlier in this report, my managers workbook has a feature where the user can select to see only the first 10 results. This was easy to do when all the rows were visible, but when another filter was first applied to the data I couldn't find any way to get the code to count the first ten *visible* rows and hide the rest. I finally talked with Dr. Allen to



find a different solution. He helped me figure out that a For Each loop with a variable would accurately count through visible rows and keep track of which rows to hide.

- ii. *Transferring data.* Another crucial feature of my program depends on transferring data from 20-25 rep workbooks to a single managers workbook. My client wanted to do this over their internal shared network drive, which I thought would be pretty easy. The problem I encountered, though, was that I didn't want the reps' workbooks to have to open the managers workbook to transfer the data because the managers workbook has code that is run on both the open and before close events that makes it slow to open and close. I discovered there's no way to transfer data to a closed workbook (besides using formulas within the excel worksheet, which wouldn't work in this case because of a moving range). Dr. Allen suggested using a CSV file as an intermediate transfer database between the workbooks, but I didn't know how to work with a CSV file and didn't have time to learn, so I ended up using just a .xlsx workbook and made it not visible to the user.

#### IV. Assistance

- a. *T.A.:* I got a little bit of help from one of the T.A.'s to help me think through the logic of calculating the forecast totals dynamically on the userform in the sales rep workbook. Because the text entered by the user for price and quantity are stored as strings, if one was deleted the forecasted total could no longer calculate and the code would break. The T.A. pointed out that I needed an and if statement on each on-update code to check if the text field was blank or not.
- b. *Dr. Allen:* I received significant help from Dr. Allen to make the top 10 option on the managers workbook run properly when a month or sales rep filter was applied. With the auto filter only showing certain rows of data, I couldn't figure out how to count through the first ten visible rows and show those, while hiding everything else.
- c. *Internet:* I search things on the internet a lot to figure out syntax of different methods, although usually my searches were in vain because the solutions didn't apply to my situation. One that I did use almost exactly was a line of code that allowed me to transfer data to another workbook while one workbook is closed or not visible.