Brevium Insurance Code VBA Script

Grant Buehner

IS 520 Spreadsheet Automation

December 10, 2014

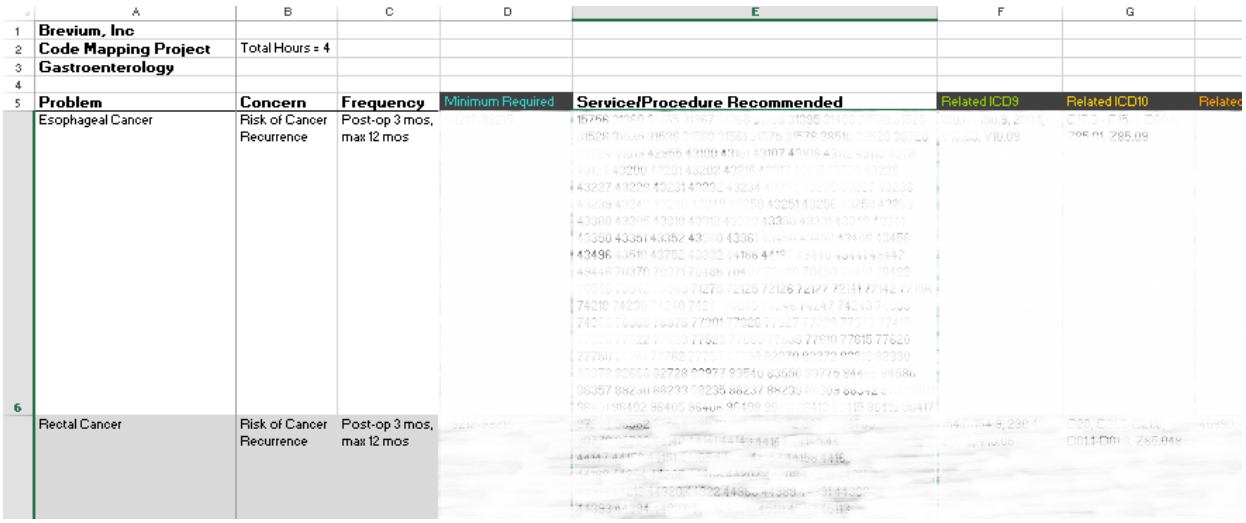
Executive Summary

Brevium Inc. is a software company that provides doctors with software to reactivate lost patients, to keep track of patients due for appointments, etc. Currently Brevium only serves a small number of specialized doctors, including ophthalmologists and dermatologists, but management would like to add the ability for Brevium software to be able to serve more types of specialties. Part of this expansion includes adding more insurance codes to our software. A consultant has been hired to provide us insurance codes. The codes are changed into regex form and put into our database. Now the test team has been given the responsibility to test to make sure that all the codes in the database are in the spreadsheet and vice versa. Also all codes need to be verified to be sure they apply to the specialty we are working with and any missing codes need to be added. The VBA script I created has been helping use look through all the codes to see which are missing, which are extra, and which have errors in them.

Problem

Brevium Inc. is looking to expand its customer base. As of right now it serves a small number of doctor specialties, such ophthalmology and dermatology. It would like to serve more specialties to increase the number of potential customers. To do so requires us (Brevium employees) to add specialty insurance codes to our database.

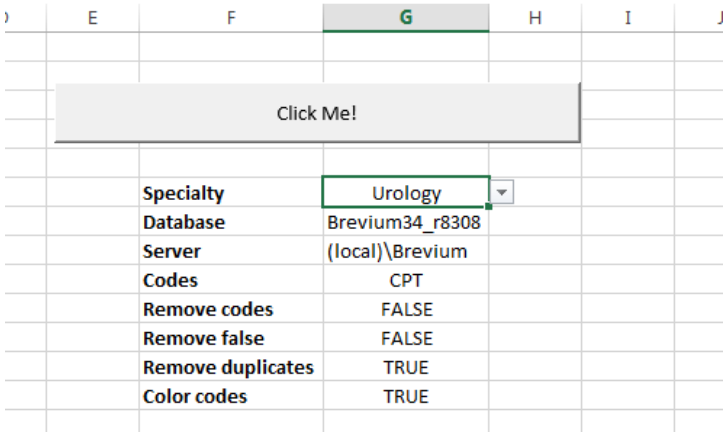
To do this a consultant was hired to look into each specialty and to give us the kinds of insurance codes our product needs. The image below is a sample of a spreadsheet he gave us with insurance code information on it (the codes have been blurred). Once all the codes were given to us they were entered into the database in regex form. The responsibility was then given to the testers to verify each code given by the consultant was valid, each valid code was correctly covered by the regex codes in the database, and each code we found that may also have applied was reported to the consultant.



	A	B	C	D	E	F	G
1	**Brevium, Inc**						
2	**Code Mapping Project**	Total Hours = 4					
3	**Gastroenterology**						
4							
5	**Problem**	**Concern**	**Frequency**	Minimum Required	**Service/Procedure Recommended**	Related ICD9	Related ICD10
6	Esophageal Cancer	Risk of Cancer Recurrence	Post-op 3 mos, max 12 mos				
	Rectal Cancer	Risk of Cancer Recurrence	Post-op 3 mos, max 12 mos				

Solution

To solve this problem I took the spreadsheet the consultant sent to us and added a VBA script to it. The VBA script runs at the click of one simple button. Some settings can be set by changing the values in some fields below the button (see the image below).



From the image above you can see that the user simply needs to select a specialty; in this case Urology was selected. The database and database server where the insurance codes can be found must be identified. Below those fields is the Codes field, which allows the user to select from a drop-down what type of insurance code they would like to look at (CPT is the current insurance code selected). The user can then choose from number of other settings including “Remove codes”, “Remove false”, “Remove duplicates”, and “Color codes”. Each of these fields served a different purpose that will be discussed below.

Project Details

Once the button has been clicked a number of processes take place, depending on the settings that were selected. The first settings we use is the remove code and remove false settings. If “Remove code” is set to true then all the codes found in the database and all the codes found in the spreadsheet are lined up in two columns side by side for easy comparison (see the image below). At the top of this spreadsheet the specialty name, database id, and type of code looked at are listed, along with some basic explanations about the sheet.

	A	B	C
1	**Urology 23 : CPT**		
2	Column A tells us which values are in our DB, but not found in this spreadsheet.		
3	Column B tells us which values are not in our DB, but are listed in this spreadsheet.		
4	Column C tells us what the codes in column B relate to.		
5	Column D (when included) tells us what problem the codes relate to.		
6			
7	**DB codes**	**SS codes**	**Long name**
8	5022	99212	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: PROBLEM FOCUSED HISTORY; EXAM; MEDICA
9	5023	99213	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: EXPANDED PROBLEM HISTORY; EXAM; MEDIC
10	5024	99214	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: DETAILED HISTORY; EXAM; MEDICAL DECISIO
11	5025	99215	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: COMPREHENSIVE HISTORY; EXAM; MEDICAL
12	50360	99212	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: PROBLEM FOCUSED HISTORY; EXAM; MEDICA
13	50365	99213	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: EXPANDED PROBLEM HISTORY; EXAM; MEDIC
14	50780	99214	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: DETAILED HISTORY; EXAM; MEDICAL DECISIO
15	50782	99215	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: COMPREHENSIVE HISTORY; EXAM; MEDICAL
16	50783	99212	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: PROBLEM FOCUSED HISTORY; EXAM; MEDICA
17	50785	99213	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: EXPANDED PROBLEM HISTORY; EXAM; MEDIC
18	50787	99214	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: DETAILED HISTORY; EXAM; MEDICAL DECISIO
19	50948	99215	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: COMPREHENSIVE HISTORY; EXAM; MEDICAL
20	51530	99212	OFFICE OR OTHER OUTPATIENT VISIT FOR EVAL & MGMT OF ESTABLISHED PATIENT, REQUIRES 2 OF 3: PROBLEM FOCUSED HISTORY; EXAM; MEDICA

Getting the codes from the database wasn’t easy as we have to convert each regex expression into plain text. For example, if the regex code looks like the following value 1[23][1-3], the possible codes would include 121, 122, 123, 131, 132, 133. We use the VBA script to “de-Regex” all these possibilities.

It is also hard to go through all the codes in the spreadsheet. Some are single codes, like 12345, and others were in ranges such as A2342-B2355. We have to go through each cell and pull out each individual code and put that code by itself in a column.

The script is also used to automatically color code the spreadsheet codes so that they can be easily found in the consultants spreadsheets (as shown above). All the names of these codes are found using a vlookup on a list of insurance codes found in another spreadsheet. Next to the name is the problem field, which simply shows what diagnosis will use the code we are using.

The main use of this is to get a list of which codes we are currently using, both in the database and in the spreadsheet. Sometimes the vlookup of the name returns a false, which means we are

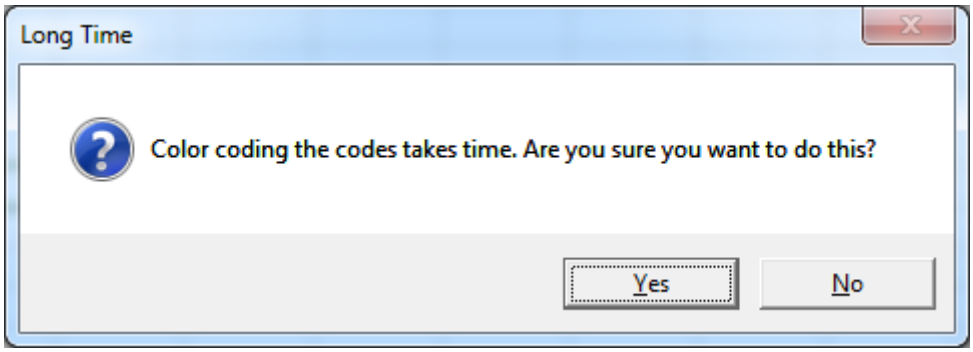
using a code that doesn't exist. Other times we can easily see the name doesn't apply to the type of problem we are dealing with. Sometime a name that returns false is supposed to be in the code. When this happens we simply use the "Remove false" setting in the script to hide any codes that returned a false.

Once we do an overview of which codes we are using we can set "Remove code" to false and rerun the script. This setting compares all the codes in the database to the codes found in the spreadsheet (see image below).

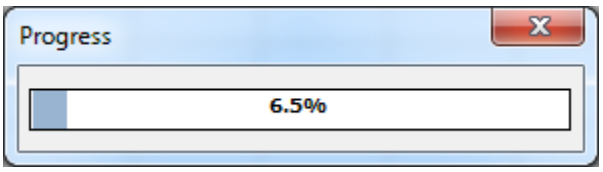
	A	B	C	D
1	**Urology 23 : CPT**			
2	Column A tells us which values are in our DB, but not found in this spreadsheet.			
3	Column B tells us which values are not in our DB, but are listed in this spreadsheet.			
4	Column C tells us what the codes in column B relate to.			
5	Column D (when included) tells us what problem the codes relate to.			
6				
7	**DB codes**	**SS codes**	**Long name**	**Problem**
8	50787	50947	LAPAROSCOPY, SURGICAL; URETERONEOCYSTOSTOMY WITH CYSTOSCOPY AND URETERAL STENT PLACEMENT	Chronic Pyelonephritis/Vesicoureteral Reflux
9	81001			

The image above shows two codes found in the database were not found in the spreadsheet. This means either we added the codes and need to let the consultant know about them, or the codes were added by mistake. The image also shows one code that was found in the spreadsheet given by the consultant that was not entered into the database. This code will need to be looked at to determine if it should be in the database.

The last setting, "Color code", can be set to true or false. Setting "Color code" to true and then running the script results in a message box appearing asking the user if they are sure they want to set this to true, because this step will take some time (see image below).



If the user selects “Yes” then a progress bar (see below) appears showing the progress of the VBA code looking at all the codes found in the database and marking them in a complete list of all insurance codes.



The color coded results enable us to analyze which codes we are using and to look at the surrounding codes to see if we are missing codes that obviously apply to the specialty we are working with (see the image below for an example of the color coding).

	A	B	
1549	1875	Malig neo epididymis	Malignant neoplasm of epididymis
1550	1876	Mal neo spermatic cord	Malignant neoplasm of spermatic cord
1551	1877	Malign neopl scrotum	Malignant neoplasm of scrotum
1552	1878	Mal neo male genital NEC	Malignant neoplasm of other specified sites of male genit
1553	1879	Mal neo male genital NOS	Malignant neoplasm of male genital organ, site unspecifie
1554	1880	Mal neo bladder-trigone	Malignant neoplasm of trigone of urinary bladder
1555	1881	Mal neo bladder-dome	Malignant neoplasm of dome of urinary bladder
1556	1882	Mal neo bladder-lateral	Malignant neoplasm of lateral wall of urinary bladder
1557	1883	Mal neo bladder-anterior	Malignant neoplasm of anterior wall of urinary bladder
1558	1884	Mal neo bladder-post	Malignant neoplasm of posterior wall of urinary bladder
1559	1885	Mal neo bladder neck	Malignant neoplasm of bladder neck
1560	1886	Mal neo ureteric orifice	Malignant neoplasm of ureteric orifice
1561	1887	Malig neo urachus	Malignant neoplasm of urachus
1562	1888	Malig neo bladder NEC	Malignant neoplasm of other specified sites of bladder
1563	1889	Malig neo bladder NOS	Malignant neoplasm of bladder, part unspecified
1564	1890	Malig neopl kidney	Malignant neoplasm of kidney, except pelvis
1565	1891	Malig neo renal pelvis	Malignant neoplasm of renal pelvis
1566	1892	Malign neopl ureter	Malignant neoplasm of ureter
1567	1893	Malign neopl urethra	Malignant neoplasm of urethra
1568	1894	Mal neo paraurethral	Malignant neoplasm of paraurethral glands
1569	1898	Mal neo urinary NEC	Malignant neoplasm of other specified sites of urinary org
1570	1899	Mal neo urinary NOS	Malignant neoplasm of urinary organ, site unspecified
1571	1900	Malign neopl eyeball	Malignant neoplasm of eyeball, except conjunctiva, corne
1572	1901	Malign neopl orbit	Malignant neoplasm of orbit
1573	1902	Mal neo lacrimal gland	Malignant neoplasm of lacrimal gland

Learning Opportunities

One of the major difficulties for writing this code was optimizing it so that it would work quickly. One of the big problems found was that when we were adjusting rows the vlookups would all recalculate, which would take a lot of time. To solve this problem we ran the vlookup once for all the codes and then copied the vlookup values as plain text. That way they no longer recalculate.

Another optimization problem is the amount of codes we are working with. Originally the script compared every type of code against all the other types of code. This was creating both false filtering of the codes, as well as taking way too long. Once I made the VBA code look at each type of code individually the code ran much faster and was more accurate. Now the code works just how we want it to and it does so very quickly.

Assistance

I did end up receiving some assistance from a co-worker on this project. He and I are working on this insurance code issue together so it made sense for us to both look at and write the script together to be how we wanted it to be. We worked side by side for the start of the project. My co-worker doesn’t know VBA so I had to teach it to him as we went along. Once he got the hang of it he wrote around a 10% of the code on his own. I went over everything he wrote to make sure it was done correctly and made changes where necessary.