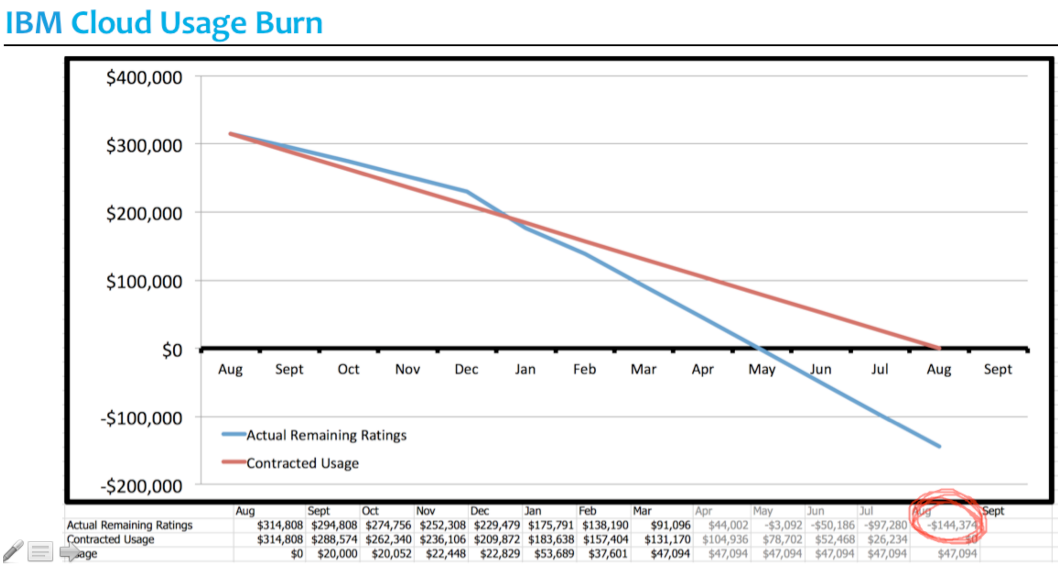
Projecting Usage for Complex Watson Use Cases

Backstory

During onboarding I noticed that a large financial customer, due to a Watson usage, was trending to overspend their cloud subscription by almost $150,000.



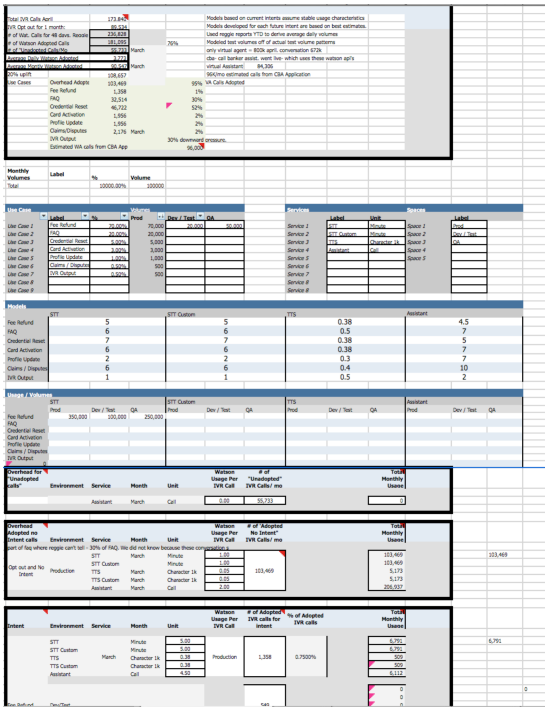
	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept
Actual Remaining Ratings	$314,808	$294,808	$274,756	$252,308	$229,479	$175,791	$138,190	$91,096	$44,002	-$3,092	-$50,186	-$97,280	-$144,374	
Contracted Usage	$314,808	$288,574	$262,340	$236,106	$209,872	$183,638	$157,404	$131,170	$104,936	$78,702	$52,468	$26,234		
age	$0	$20,000	$20,052	$22,448	$22,829	$53,689	$37,601	$47,094	$47,094	$47,094	$47,094	$47,094	$47,094	

Projected overspend calculation: -$143,000

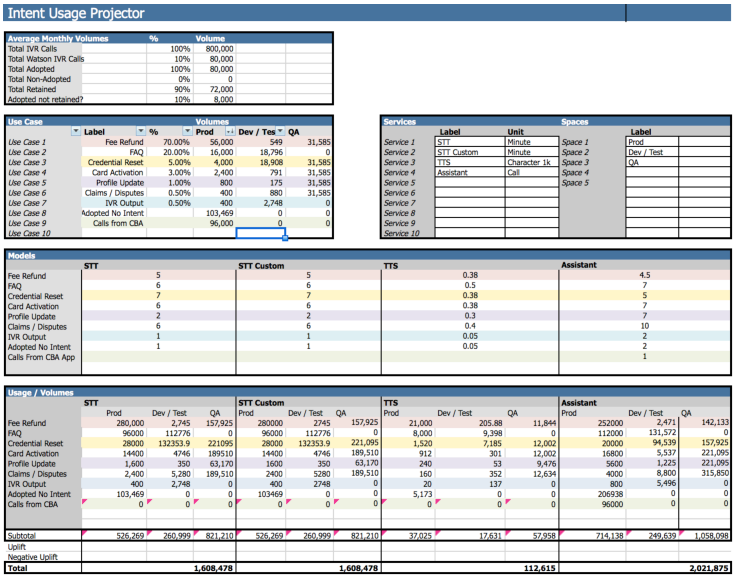
The cloud seller and I sat down with the client and walked them through their usage. It became clear that they were unaware that their aggressive testing requirements were having such a large impact on their consumption. We also learned that an IBM services team had proposed building the second phase of their virtual phone attendant and friction had arisen because the client claimed that the Watson usage projection they gave was crude, inaccurate and unsound. The cloud team was called upon to use a solid methodology for predicting usage volumes.

The CSM toolkit: Magic 8 ball get's little use on this one.

We ultimately developed an approach and packed it into a spreadsheet, which is **here**, https://ibm.box.com/s/wx2m1v7ejm530rpenguumn1kifx0pd33 and at the bottom. Because it was built under the gun of a pitch deadline, the first spreadsheet was a 200+ line unwieldy hackjob. In the end, I rebuilt it with 50 lines and 5 sections to make an easy to use and re-use template.



Version 1: The first few sections of a 200+ line sprawling hack job.



Final version: concise and usable by a monkey… and me.

The methodology we followed centers on isolating the component parts of the virtual agent solution:

- Watson services = 3 (TTS, SST and WA)
- Use Cases = 9, and…
- Environments = 4 (Dev, Test, QA, Prod)

To organize the calculations as much as possible we defined:

- the unit of measurement for each service,
- the dynamics of each use case and
- the input volume for each environment

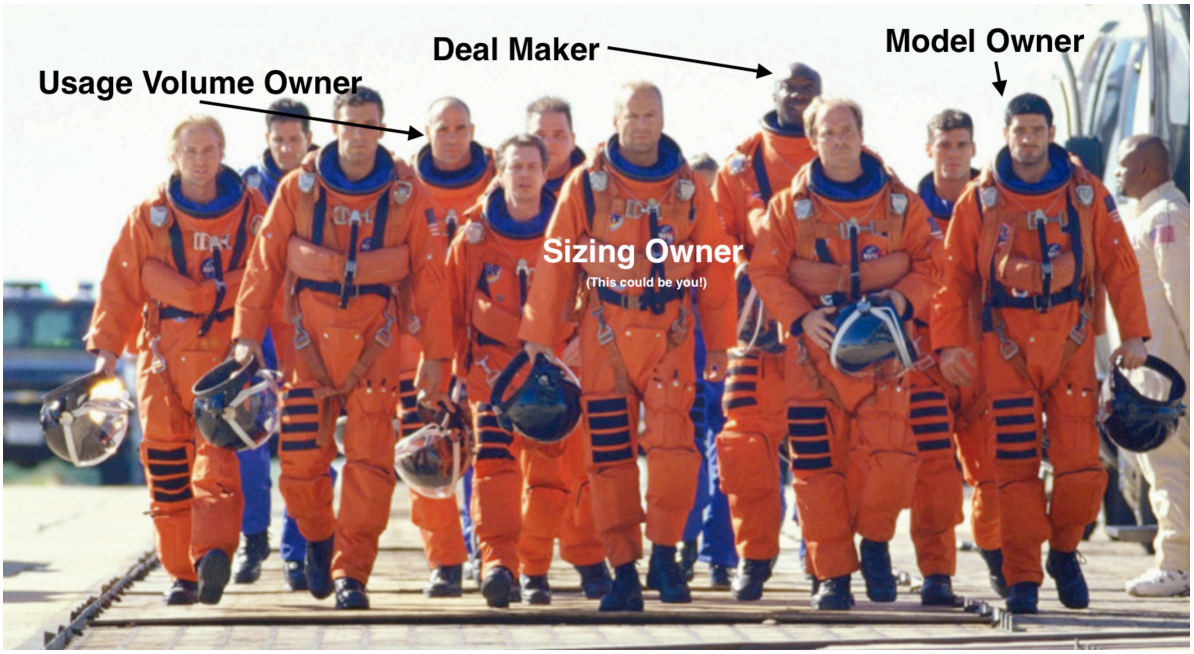
This allowed us to derive all 108 (3*9*4) usage drivers. This exercise took 5 people 2 weeks (part-time). I made the mistake of trying to take it on myself without completely understanding the complexity involved to do it right. As the scope stretched wider and wider I was forced to bring in co-owners to keep from downing.

Lessons Learned

Sizing is a team sport; don't try to own all the data

Accept from the outset that to accurately project usage for a broad set of use cases, you will want to identify multiple individual owners including but not limited to:

- **Model Owner:** Someone who helped architect the solution and who understands how usage, in our case phone call flows, have been built, tested and intended to be used. Will callers be on the phone 2 minutes or 20? Will they be using speech to text every time or is online chat another enabled channel? Etc.
- **Usage/Volume Owner**: Someone, most likely from the client's side, who understands current phone call volumes and can help make predictions about future volumes for your use cases. In our case we had volumes and volumes of reports that were difficult to interpret. We needed someone who understood the current call center dynamics and how to translate those to our intended use case volumes.
- **Deal Maker:** The output from this exercise will be raw service usage volumes and so your seller can work with a deal maker and offering management to convert the volumes into customer cost. They will need to include things like premium instance charges, premium support, expanded concurrency charges, and any other cloud services charges so the client sees a complete package.
- **Sizing Owner**: Someone to put all the pieces together, enter the data into the template and execute the sizing. If you are a CSM, this could be you.



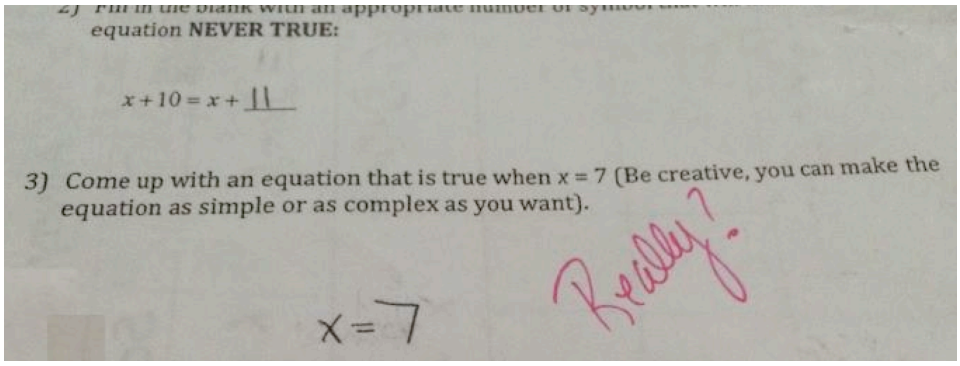
Astronauts train for years. You have 3 days.

- **Just follow the process**
    - You can be 100% certain that your projection will be wrong. This process is partially about standing in front of a client and backing up a proposal with solid logic.
    - Don't get bogged down due to lack of perfect data, call out your assumptions and move on. I slowed down our progress by several days by seeking certainty, which didn't exist. Embrace imperfection, be the imperfection!
    - Understand the implications of the volumes and tiered pricing. Once we had established roughly what the volumes were we then referenced the associated service pricing schedules. It became very clear that obsessing over 5000 calls what a complete waste of time as their tiers were priced at 1M call levels. This was eye opening.
    - When we did present this to the client we didn't drag them into the spreadsheet weeds but we were able to walk them high level though how we built discreet models, service-by-service, environment-by-environment, use-case-by-use-case. They just nodded…and then signed the expansion!


Strive for perfection, but accept best effort

Use a model/template

- If sizing is a team sport, the spreadsheet is your playing field. It requires granularity but it gives structure and will downsize a complicated problem to a simple systematic process of fill-in-the-blank. We developed a template that captures the methodology



Reduce the challenge to a fill in the blank exercise!

The Methodology

Let's walk though the 5 sizing steps using the modeling spreadsheet we built as the backdrop. In this example we have 9 use cases that each use 3 Watson services (TTS, SST and WA) across 4 environments (Dev, Test, QA, Prod). As mentioned before, we realized that this meant there was the need to calculate output for at least 108 discreet uses (3x4x9=108).

Let's get into the weeds

Step 1. Start at the top of the funnel, capture your volume assumptions

Here we input the total number of actual IVR calls that come into the bank and then break it down from there. You can see below, 800K calls come into the bank every month via the bank's IVR system (IVR=Interactive Voice Response, as in "Press 1 for hours, 2 for loans and mortgages..."). We also defined that 80% of them are expected to route to Watson services. Based on phase one we know that 90% of those should be successful and that the customer won't transfer out (that's pretty good!). You will notice that we use words like "adopted" and "retained", this is because we wanted to mirror the terminology used by the customer. Each sizing will have its own funnel entry dynamics and client specific lexicon but the point is that you need to capture your fundamental volume assumptions. We chose to project monthly volumes because that matched the customer's call logs, which were the basis of our understanding.

Average Monthly Volumes	%	Volume
Total IVR Calls	100%	800,000
Total Watson IVR Calls	80%	640,000
Total Adopted	100%	640,000
Total Non-Adopted	0%	0
Total Retained	90%	576,000
Adopted not retained?	10%	64,000

Capture your fundamental volume assumptions

Step 2. Divide the funnel into use cases

The expected production volumes from step 1 will now need to be separated by use case because each use case has different dynamics (Which we will define/model in step 4). Your Volume Owner will need to determine the percentages of calls attributed to each use case. So below you can see we labeled each use case in the first column and then assigned it a percentage of the total inflow of calls. We then let the spreadsheet do the heavy lifting to calculate the volumes in column 3 "Prod". For the dev/test and QA columns we actually had testing governance requirements and historic use case test volumes to guide us on what to input.

Use Case	Label	%	Volumes: Prod	Dev / Tes	QA
Use Case 1	Fee Refund	70.00%	448,000	549	31,585
Use Case 2	FAQ	20.00%	128,000	18,796	0
Use Case 3	Credential Reset	5.00%	32,000	18,908	31,585
Use Case 4	Card Activation	3.00%	19,200	791	31,585
Use Case 5	Profile Update	1.00%	6,400	175	31,585
Use Case 6	Claims / Disputes	0.50%	3,200	880	31,585
Use Case 7	IVR Output	0.50%	3,200	2,748	0
Use Case 8	Adopted No Intent		103,469	0	0
Use Case 9	Calls from CBA		96,000	0	0
Use Case 10					

Define your use cases and attribute percentages to them

Step 3. Define the Services and Spaces

Label the Services and associated unit of consumption as well as the spaces/environments that need to be projected. Here we ended up collapsing Dev/Test into one line because the volumes were low.

Services	Label	Unit	Spaces	Label	
Service 1	STT	Minute	*Space 1*	Prod	
Service 2	STT Custom	Minute	*Space 2*	Dev / Test	
Service 3	TTS	Character 1k	*Space 3*	QA	
Service 4	Assistant	Call	*Space 4*		
Service 5			*Space 5*		
Service 6					
Service 7					
Service 8					
Service 9					
Service 10					

Each space/environment that consumes services should be captured

Step 4. Define a model for each use case

These models are the core of what drives the usage projection, this is where your Model Owner is invaluable. They should understand how the call flows are designed, what the expected utterances are etc. Below we see that for the Fee Refund intent (the first red/pink line), a single phone call on average results in:

- 5 SST minutes as well as…
- 5 custom model minutes (which have a different price and so must be calculated separately)
- 380 characters of text to speech and..
- 4.5 Assistant calls.

We built up a model for each intent and loaded them in this section. The assumption is that the usage dynamics will be the same across all environments so by building one model it can calculate volumes for all of your environments. The colors just make it easy to follow a particular use case through the spreadsheet.

Models				
	STT	STT Custom	TTS	Assistant
Fee Refund	5	5	0.38	4.5
FAQ	6	6	0.5	7
Credential Reset	7	7	0.38	5
Card Activation	6	6	0.38	7
Profile Update	2	2	0.3	7
Claims / Disputes	6	6	0.4	10
IVR Output	1	1	0.05	2
Adopted No Intent	1	1	0.05	2
Calls From CBA App				1

Model each intent: For the Fee Refund intent (red line), a single phone call results in 5 SST minutes, 380 characters of TTS and 4.5 WA calls.

Step 5. Let the data flow!

The final section for our work pushed out 112 calculations. These calculations are simply the product of the use case volumes from step 2, and the models from step 4. These calculations are then summed into into 12 subtotals and finally the 4 Watson service usage projections (in bold) that we needed to give to the deal maker!

Usage / Volumes												
	STT			STT Custom			TTS			Assistant		
	Prod	Dev / Test	QA	Prod	Dev / Test	QA	Prod	Dev / Test	QA	Prod	Dev / Test	QA
Fee Refund	2,240,000	2,745	157,925	2240000	2745	157,925	168,000	205.88	11,844	2016000	2,471	142,133
FAQ	768000	112776	0	768000	112776	0	64,000	9,398	0	896000	131,572	0
Credential Reset	224000	132353.9	221095	224000	132353.9	221,095	12,160	7,185	12,002	160000	94,539	157,925
Card Activation	115200	4746	189510	115200	4746	189,510	7,296	301	12,002	134400	5,537	221,095
Profile Update	12,800	350	63,170	12800	350	63,170	1,920	53	9,476	44800	1,225	221,095
Claims / Disputes	19,200	5,280	189,510	19200	5280	189,510	1,280	352	12,634	32000	8,800	315,850
IVR Output	3,200	2,748	0	3200	2748	0	160	137	0	6400	5,496	0
Adopted No Intent	103,469	0	0	103469	0	0	5,173	0	0	206938	0	0
Calls from CBA	0	0	0	0	0	0	0	0	0	96000	0	0
Subtotal	3,485,869	260,999	821,210	3,485,869	260,999	821,210	259,989	17,631	57,958	3,592,538	249,639	1,058,098
Uplift												
Negative Uplift												
Total			**4,568,078**			**4,568,078**			**335,579**			**4,900,275**

Building this modeler took some time, but allowed us to focus on the higher value efforts of identifying owners, sourcing data, and interpreting that data. I hope it is helpful and would love to hear feedback or ideas on how to improve it!

To recap:

1. **Define a team**
    a. Model Owner
    b. Usage/Volume Owner
    c. Deal Maker
    d. Sizing Owner
2. **Follow the Process**
    a. Call out your assumptions
    b. Accept imperfection.
    c. Our methodology
        i. Capture your volume assumptions
        ii. Divide your funnel into 1. use cases and 2. environments
        iii. Define the services in question
        iv. Develop a model for each use case
        v. Calculate volumes for each use case in each environment using your models.
3. **Use a template**
    a. Here take mine➔ abcd.box.com