Ripples from the Flint Water Crisis: The Road to Safe Drinking Water in Michigan

Randy Winowiecki

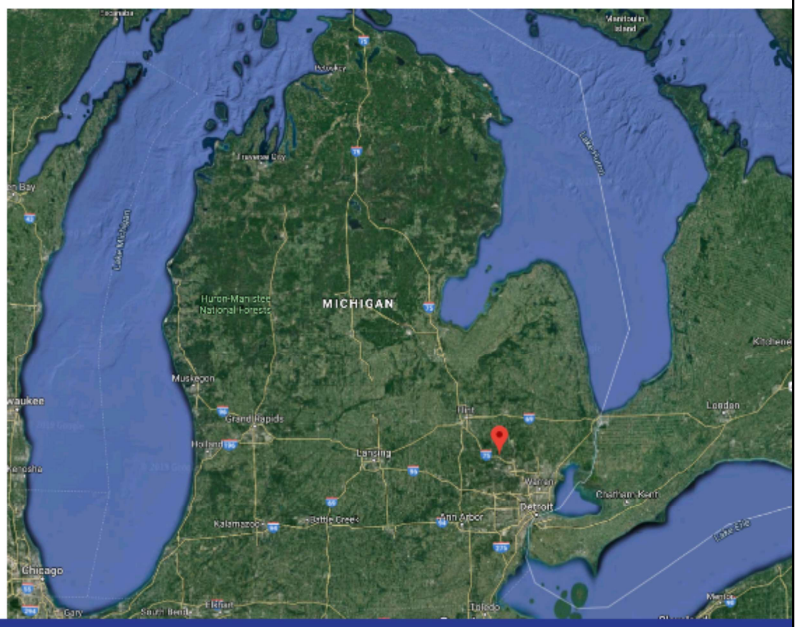
Where the Road Began

- Our water treatment crew collects thousands of data points weekly
- Critical to record data efficiently and without errors
- Flint Water Crisis led to regulation
- Created a report using Survey123 and Python to more efficiently adhere to those regulations

Hi, my name's Randy Winowiecki. I am a GIS Technician for Independence Township's Department of Public Works in Clarkston, Michigan. I've been working very closely with the water treatment crew for over a year. The crew consists of 3 members that run the 10 water treatment plants in Independence Township to provide the residents with the best quality water possible. One of their main objectives is to collect data on a daily basis. This includes water samples, residuals of substances in the water, and amount of water pumped. I'm going to go over the basics of the Flint Water Crisis. I'm also going to explain the process I implemented using Survey123 in our department to more efficiently adhere to water quality legislation that resulted from the crisis.

Where is Independence Township?

- North Central Oakland County Michigan
- 30 Miles from Flint
- 40 Miles from Detroit
- Community of about 35,000 residents
- 137 miles of water main and no lead service lines



Google Maps, Google, www.google.com/maps.

The red pin on the map is Independence Township. We fall in between Flint, Michigan and Detroit, Michigan. We have a relatively large population, and municipalities around us either do private wells or are hooked up to Detroit water. We also have a large water infrastructure compared to those around us.

Flint Water Crisis: The Basics

- In April 2014, Flint switched water sources to the Flint River until a pipeline could be completed in 2017 to bring in Lake Huron water
- This was a cost cutting measure for a struggling city

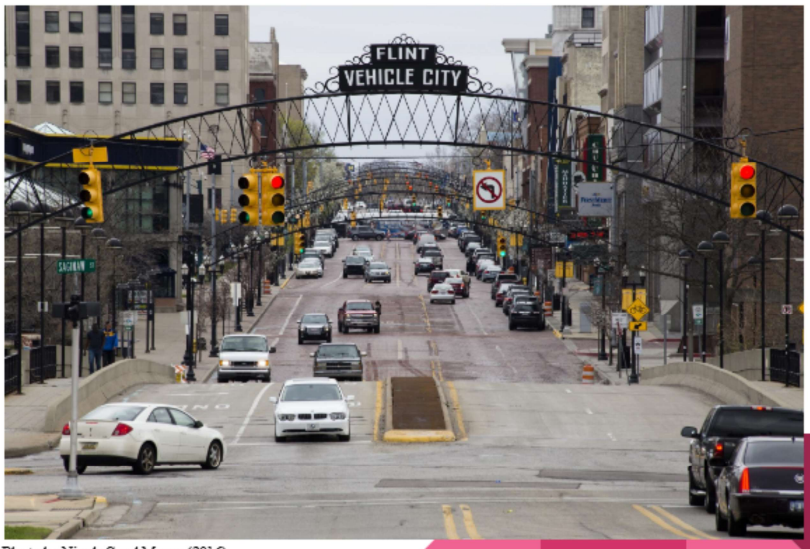


Photo by Nicole Casal Moore (2016)

I know there are a lot of misconceptions about the Flint water crisis, so I wanted to go over the basics. There IS NOT lead inherently in Michigan water. Instead, thousands of Flint's service lines are made of lead. Putting in new lead pipes was outlawed in 1986, but a large amount still are used today. This is because records of old pipes are inaccurate or non-existent and it's extremely expensive to replace them. Flint (the birthplace of General Motors) had a booming automobile industry but started to struggle around the time of the auto industry bailout, which resulted in plants getting shut. So in April 2014, the city decided that paying for Detroit water was too expensive and switched water sources until a more permanent solution could be completed in 2017.

Lead Service Lines

- By May, residents complained about the brown water coming from their taps
- In August, E. coli and coliform (fecal bacteria) were detected in the water

Photo by Kelsey J. Pieper, Min Tang, and Marc A. Edwards (2014)

After the switch to Flint River water, residents began to complain about brown water coming from their taps. A couple months later, E.coli and coliform were detected in the water, likely because of insufficient chlorine in the water mains. As you can see from the picture, the structure of existing service lines are lead and then a layer of rust that builds up over time when the metal particles from the water coat the pipe with a layer of rust. Drinking water starts at treatment plants that disinfect and/or filter, then it goes to main arteries called water mains that come out of the plants and run alongside roads. Those mains break off into smaller “service lines or service leads” that connect to the residents.

Rust and Lead Start to Contaminate Water

- Then a leaked memo from the Environmental Protection Agency was found that warned of dangerous lead levels in the water
- In October 2015 the city switched their water supply back to Detroit water

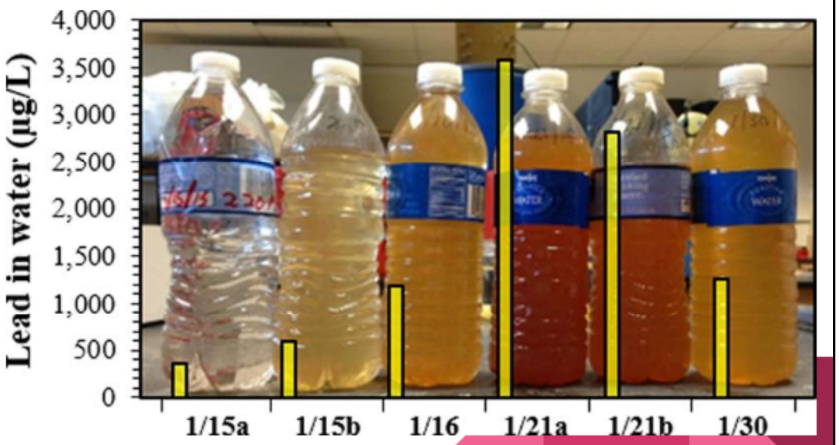


Photo by Kelsey J. Pieper, Min Tang, and Marc A. Edwards (2014)

The combination of Flint River water being slightly more acidic than Detroit water, and no corrosion preventative aka “orthophosphates” caused the layer of rust to slowly flake off of the lead pipes. This is why residents were seeing yellowish orange water. Once the rust was off, that left bare lead pipe and the lead would start leeching off into the water and ultimately flow straight to the residents’ tap. After over a year, the city finally switched back to Detroit water, but the damage to the pipes had already been done.

Effects of the Crisis

- It took months for the people of Flint's concerns to be heard
- 12 people died from Legionnaires Disease
- Hundreds of thousands of people were exposed to high lead levels
- It ended up costing an estimated amount of hundreds of millions of dollars
- Multiple people are/were on trial for letting this happen

Photo by Dennis Pajot / Getty Images

The crisis had a direct effect on the health of the residents of Flint. It took months before the governor declared a state of emergency in the city. All the while, people died from Legionnaires Disease, a severe form of pneumonia, due to the city's failure to maintain sufficient chlorine in its water mains. Flint should have put orthophosphates in their water, but why didn't they? The higher ups that were in charge of Flint's water ignored warning signs of an issue and wanted to save money. In the end, the crisis ended up costing hundreds of millions of dollars.

What legislation has come from this?

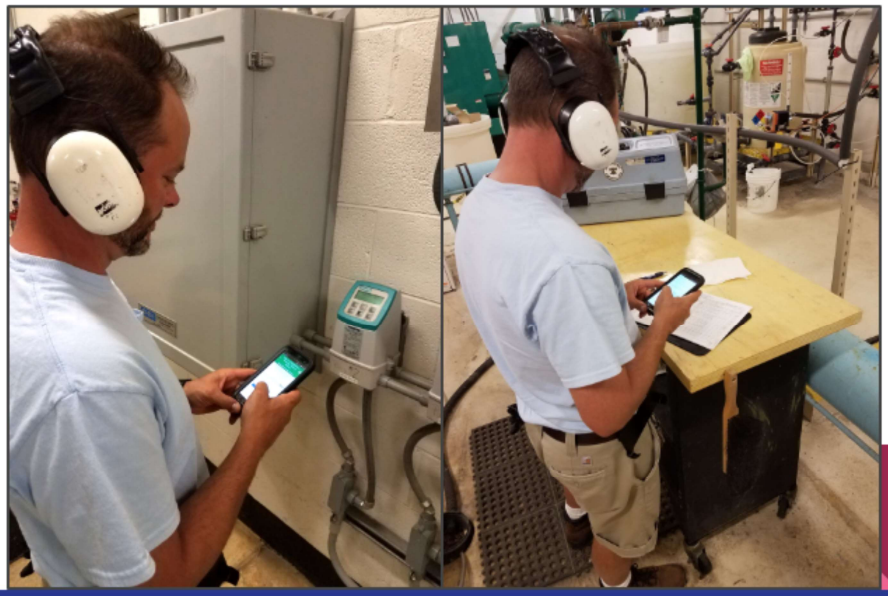
- Lead action level of 15 ppb will change to 12 ppb in 2025
- A preliminary Distribution System Materials Inventory (DSMI) must be submitted to the MDEQ by January 1, 2020
- Water supplies with lead service lines must be replaced at a rate of 5%/year

Source: The Berkey

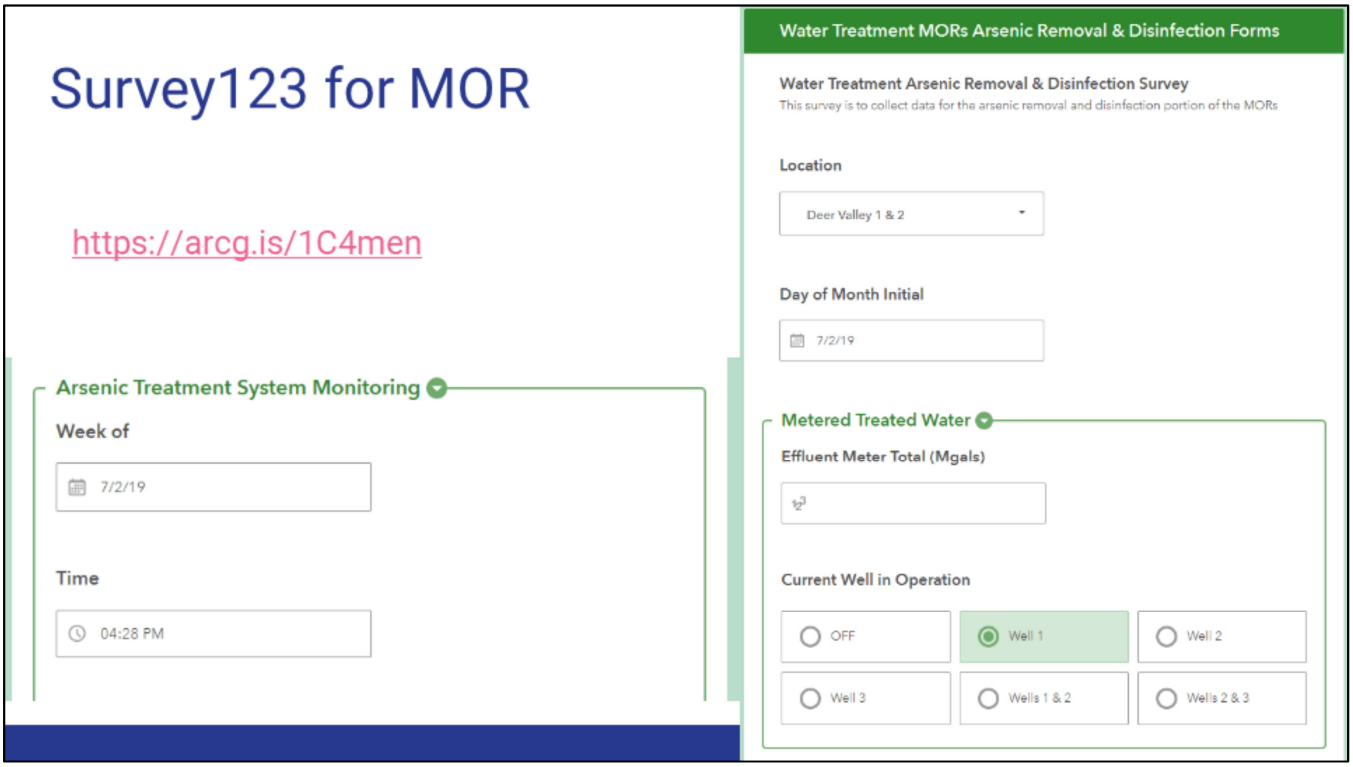
Other effects of the crisis include the laws, legislation, and regulations that were created as a result and are in practice today. The laws about the maximum contaminant level of lead (level at which action will be taken) will go from 15 parts per billion to 12 in 2025. Preliminary service lead inventories and a list of what materials are in our service leads are required by the end of this year with a final more detailed copy due by 2025. This is a measure to document all known lead service lines. More on this later. By 2025 communities also need to be replacing their lead service lines completely. But it's more crucial than ever to be using GIS to its full potential in situations like this. It can help with locating pipes, cost and material.

Independence Township's Survey123 Solution

- We are collecting data by using Survey123 at the wellhouses
- Independence uses Python to pull data from the feature service and generate a monthly operating report (MOR)

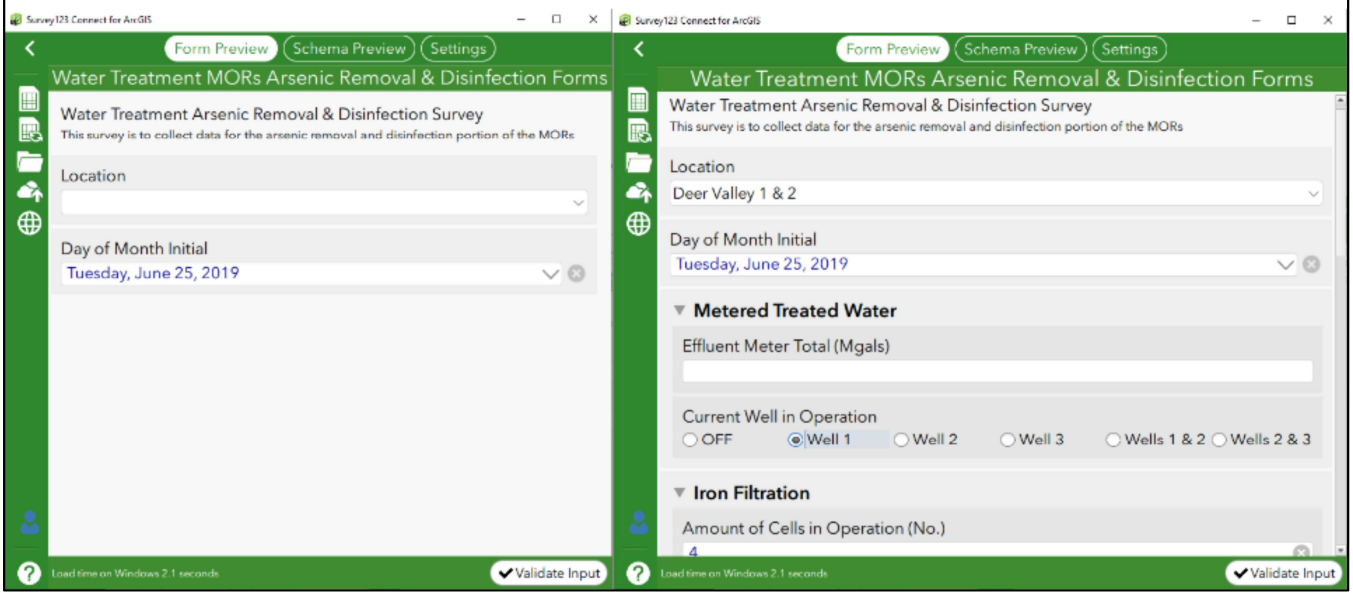


Even though Independence Township doesn't have lead pipes, how did we adapt to the changing regulations? Daily readings are being collected on cell phones with Survey123 everyday at the wellhouses. This is purely to compile data for the monthly operating report or "MOR". We run a Python script that pulls data from the feature service and places it in an MOR. Before we implemented this solution, the crew would write down 20 columns of data for each day of the month, resulting in roughly 6,000 data points collected across all of the wellhouses for the month. Then one person would have to look at all that handwritten data and manually type up a Monthly Operating Report to send the the Michigan Department of Environmental Quality (now known as EGLE). This time consuming process would take almost 8 hours every month. Here is a picture of Tom from the water treatment crew using Survey123.



I'll walk you through the survey now. As you can see, I'm using the relevant column to only show data when necessary. Choosing a location will bring up the proper questions for that specific wellhouse.In Survey123 you can use the now() function to populate the date. You can also use groups to outline each section visually. When I choose "OFF" under current well in operation, it shortens the survey to only include relevant questions. When I choose "Well 1", more options are shown and when I do a backwash, even more. No lab, partial lab, and full lab have different fields to fill out for each scenario. When we perform a full lab, certain wellhouses will show an additional arsenic treatment system monitoring group. We autofill the date, time, and well number as much as possible to save time. Then send the survey off and complete the survey at the next location. Sending the survey then populates the feature service through the phones data connection, or if there is no service, you can just save them on your phone and send them later.

Survey123 for Connect Application



I created the survey on the desktop application “Survey123 for Connect” last year in version 2.5, but since then I’ve updated to version 3.3.51. Here is a snapshot of the application I used.

Paper to Digital: Benefits

Are you performing a Partial Lab or a Full Lab?
○ Partial Lab ◉ Full Lab ○ No Lab

[Raw] Iron as Fe (Mg/l)

[Raw] pH

[Treated] Chlorine POE Free (Mg/l)

[Treated] Chlorine POE Total (Mg/l)

[Treated] Iron as Fe (Mg/l)

[Treated] pH

Are you performing a Partial Lab or a Full Lab?
◉ Partial Lab ○ Full Lab ○ No Lab

[Treated] Chlorine POE Free (Mg/l)

[Treated] Chlorine POE Total (Mg/l)

[Treated] Iron as Fe (Mg/l)

Each of the field titles in the survey are worded exactly the same way as on the paper version. This makes for a smooth transition from paper to digital. One of the advantages of the relevancy feature in my survey is that a crew member can not accidentally type information into the wrong cell or another wellhouses row. For example, choosing partial lab hides questions that would be asked in a full lab so there is no room for error.

Survey123 in Connect (Excel)

	A	B	C	L	M	N	O	P	Q	R	S
1	type	name	label	relevant	calculation	choice_filter	repeat_count	label::language1	hint::language1	media::audio	media::image
2	note	survey_name	Water Treatment Arsenic Removal & D								
3	select_one list_loc	location	Location								
4	start	start_time	Start Time		now()						
5	end	end_time	End Time		now()						
6	date	day_of_month	Day of Month Initial		now()						
7	begin group	h_metered_treated_wat	Metered Treated Water	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
8	decimal	h_effluent_meter_total_	Effluent Meter Total (Mgals)	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
9	select_one list_hor	h_current_well_in_oper	Current Well in Operation	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
10	end group	h_metered_treated_wat	Metered Treated Water	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
11	begin group	h_iron_filtration	Iron Filtration	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
12	integer	h_amount_of_cells_in_	Amount of Cells in Operation (No.)	selected(${h_current_well_in_n}${h_current_well_in_operation},'OFF','OFF','4')							
13	decimal	h_head_loss_through_f	Head Loss Through Filter (PSID)	selected(${h_current_well_in_n}${h_current_well_in_operation},'OFF','2','')							
14	decimal	h_flow_since_bw	Plant Flow-thru Since RWM	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
15	select_one list_ny	h_backwash_n_y	Did you perform a backwash?	selected(${h_current_well_in_operation},'1') or selected(${h_current_well_in_operation},'2') or selected(${h_current_well_in_operation},'3') or selected(${h_current_well_in_operation},'4') or selected(${h_current_well_in_operation},'1,2') or selec							
16	integer	h_amount_of_cells_bac	Amount of Cells Backwashed (Iron)	selected(${h_backwash_n_y}${h_backwash_n_y},'OFF','2','4')							
17	decimal	h_backwash_waste_wa	Backwash Waste Water (Mgals)	selected(${h_backwash_n_y},'YLS')							
18	end group	h_iron_filtration	Iron Filtration	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
19	begin group	h_chlorination	Chlorination	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
20	decimal	h_125_naocl_chl_tank_	12.5% NAOCL Available Chlorine Tank	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
21	decimal	h_12p_naocl_chl_tank	12.5% NAOCL Available Chlorine Tank	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
22	end group	h_chlorination	Chlorination	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
23	begin group	h_chemical_analysis_r	Chemical Analysis/Residuals	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
24	select_one list_feb	h_partial_full_feb	Are you performing a Partial Feb or a F	selected(${h_current_well_in_operation},'1') or selected(${h_current_well_in_operation},'2') or selected(${h_current_well_in_operation},'3') or selected(${h_current_well_in_operation},'4') or selected(${h_current_well_in_operation},'1,2') or selec							
25	decimal	h_raw_iron_as_fe_mgl	[Raw] Iron as Fe (Mg/l)	selected(${h_partial_full_feb},'Full')							
26	decimal	h_raw_ph	[Raw] pH	selected(${h_partial_full_feb},'Full')							
27	decimal	h_treated_chlorine_poc	[Treated] Chlorine HOCL Free (Mg/l)	selected(${h_partial_full_feb},'Full') or selected(${h_partial_full_feb},'Partial')							
28	decimal	h_treated_chlorine_poc	[Treated] Chlorine POC Total (Mg/l)	selected(${h_partial_full_feb},'Full') or selected(${h_partial_full_feb},'Partial')							
29	decimal	h_treated_iron_as_fe_m	[Treated] Iron as Fe (Mg/l)	selected(${h_partial_full_feb},'Full') or selected(${h_partial_full_feb},'Partial')							
30	decimal	h_treated_ph	[Treated] pH	selected(${h_partial_full_feb},'Full')							
31	end group	h_chemical_analysis_r	Chemical Analysis/Residuals	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
32	begin group	h_arsenic_monitoring	Arsenic Treatment System Monitoring	${location}='deer_valley1and2' or ${location}='deerwood1' or ${location}='deerwood_manor1if or ${location}='lake_oakland_woods2and3'							
33	date	h_week_of	Week of	selected(${h_partial_full_feb},'now')							
34	time	h_time	Time	selected(${h_partial_full_feb},'now')							
35	select_one list_name	h_initial	Initial	selected(${h_partial_full_feb},'Full')							
36	decimal	h_total_flow_mgals	Total Flow (Mgals)	selected(${h_partial_full_feb},'Full')							
37	integer	h_well_in_op_arsenic	Well in Operation Number	selected(${h_partial_full_feb},${h_current_well_in_operation})							
38	decimal	h_raw_temp	[Raw] Temperature (Cel)	selected(${h_partial_full_feb},'Full')							
39	decimal	h_raw_ph_arsenic	[Raw] pH	selected(${h_partial_full_feb},${h_raw_ph})							
40	decimal	h_raw_arsenic_ppb	[Raw] Arsenic PPB	selected(${h_partial_full_feb},'Full')							
41	decimal	h_poe_temp	[POE] Temperature (Cel)	selected(${h_partial_full_feb},'Full')							
42	decimal	h_poe_ph	[POE] pH	selected(${h_partial_full_feb},${h_treated_ph})							
43	decimal	h_poe_arsenic_ppb	[POE] Arsenic PPB	selected(${h_partial_full_feb},'Full')							

survey | choices | settings | types

This picture shows Survey123 in Connect in the excel view to show a sample of the background structure of the survey. This is how I created the relevancy and autofill features and variables. From the picture you can see that we are using the “now()” calculation to automatically enter the date. I have set it up to only have the crew input data when its relevant. I set up groups to keep a visual queue of your progress in the survey. Certain wellhouses have the same water filters and are similar structure wise, so they have the same paperwork and can be grouped together with the same variable names. Deer Valley, Deerwood, Deerwood Manors, and Lake Oakland Woods for example share a horizontal filter in the wellhouse and have similar reports.

Python Script

- You can query data from your survey and set up a date range to pull data from a specific month
- You can load workbooks for a specific location to generate its MOR

```python
# Query location and Date /since the first of the month/
mor_query_he = mor_lyr.query(where="location LIKE 'hillview' AND day_of_month >= date '" + str(firstofmonth) +
mor_query_cg = mor_lyr.query(where="location LIKE 'clarkston_gardens' AND day_of_month >= date '" + str(firstof
mor_query_ch = mor_lyr.query(where="location LIKE 'chestnut_hills_farms' AND day_of_month >= date '" + str(firs
mor_query_dv = mor_lyr.query(where="location LIKE 'deer_valley1and2' AND day_of_month >= date '" + str(firstofm
mor_query_dw = mor_lyr.query(where="location LIKE 'deerwoodII' AND day_of_month >= date '" + str(firstofmonth)
mor_query_dm = mor_lyr.query(where="location LIKE 'deerwood_manorsIII' AND day_of_month >= date '" + str(firsto
mor_query_lo = mor_lyr.query(where="location LIKE 'lake_oakland_woods2and3' AND day_of_month >= date '" + str(f
mor_query_ph = mor_lyr.query(where="location LIKE 'pelton_heights2' AND day_of_month >= date '" + str(firstofmo
mor_query_wc = mor_lyr.query(where="location LIKE 'waldon_creek1' AND day_of_month >= date '" + str(firstofmont
mor_query_bc = mor_lyr.query(where="location LIKE 'bay_court1and2' AND day_of_month >= date '" + str(firstofmon
ars_query_dm = mor_lyr.query(where="location LIKE 'deerwood_manorsIII' AND h_week_of IS NOT NULL AND day_of_mon
ars_query_dw = mor_lyr.query(where="location LIKE 'deerwoodII' AND h_week_of IS NOT NULL AND day_of_month >= da
ars_query_dv = mor_lyr.query(where="location LIKE 'deer_valley1and2' AND h_week_of IS NOT NULL AND day_of_month
ars_query_bc = mor_lyr.query(where="location LIKE 'bay_court1and2' AND v_week_of IS NOT NULL AND day_of_month >
ars_query_lo = mor_lyr.query(where="location LIKE 'lake_oakland_woods2and3' AND h_week_of IS NOT NULL AND day_o
ars_query_wc = mor_lyr.query(where="location LIKE 'waldon_creek1' AND v_week_of IS NOT NULL AND day_of_month >=
ars_query_ph = mor_lyr.query(where="location LIKE 'pelton_heights2' AND v_week_of IS NOT NULL AND day_of_month
```

```python
# Variable to load workbooks
book_3175 = openpyxl.load_workbook(filename='D://Python/WSSN_3175_MORs.xlsm', read_only=False, keep_vba=True)
book_1773 = openpyxl.load_workbook(filename='D://Python/WSSN_1773_MORs.xlsm', read_only=False, keep_vba=True)
book_3342 = openpyxl.load_workbook(filename='D://Python/WSSN_3342_MORs.xlsm', read_only=False, keep_vba=True)
book_ars1773 = openpyxl.load_workbook('D://Python/ArsenicTestResults_1773.xlsx')
book_ars3342 = openpyxl.load_workbook('D://Python/ArsenicTestResults_3342.xlsx')
```

```python
# Variable for location tab in sheets
sheet_he = book_3175['Hillview Estates']
sheet_cg = book_3342['Clarkston Gardens']
sheet_ch = book_3342['Chestnut Hill Farms']
sheet_dv = book_1773['DeerValley']
sheet_dw = book_1773['Deerwood']
sheet_dm = book_1773['Manors']
sheet_lo = book_3342['Lake Oakland']
sheet_ph = book_3342['Pelton Heights 2']
sheet_wc = book_3342['Waldon Creek 1']
sheet_bc = book_3342['Bay Court 1&2']
sheet_arsdm = book_ars1773['Manors']
sheet_arsdw = book_ars1773['Deerwood']
sheet_arsdv = book_ars1773['Deervalley']
sheet_arsbc = book_ars3342['Bay Court']
sheet_arslo = book_ars3342['Lake Oakland']
sheet_arswc = book_ars3342['Waldon Creek']
sheet_arsph = book_ars3342['Pelton']
```

Before implementing the survey, it would take me 4 hours every month to export a csv from Survey123 and copy and paste in the values to the MOR for Tom who types up our data. Now, it takes me 5 mins to run the Python script and double check that everything ran correctly. The top picture of python code is a query for location and date, which grabs a specific wellhouse and then all of the data from the previous month. The middle picture loads up the workbooks, which are labeled by WSSN (Water Supply Serial Number), a unique identifier for water systems. We also do arsenic sheets for any wellhouse that is filtering out arsenic, so that workbook is loaded up as well. On the bottom I make a variable for each location for the excel sheets.

Populating Monthly Data Using Python

- Python script for Deer Valley wellhouse is used to populate its MOR
- We Use Jupyter Notebooks
- Saved to an Excel sheet

```python
# write query to excel - deer valley
for f in mor_query_dv:
    unix_time = str(f.attributes['day_of_month'])[0: 10]
    unix_time = float(unix_time)
    date = datetime.datetime.fromtimestamp(unix_time, tz_utc)
    row = int(date.strftime('%d')) + 11
    column_cell = 'B'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_effluent_meter_total_mgals'])
    column_cell = 'F'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_current_well_in_operation'])
    column_cell = 'G'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_amount_of_cells_in_oper_no'])
    column_cell = 'H'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_head_loss_through_filter_psid'])
    column_cell = 'I'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_amount_of_cells_backwashed_fe'])
    column_cell = 'J'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_backwash_waste_water_mgals'])
    column_cell = 'K'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_125_naocl_chl_tank_vol_gal'])
    column_cell = 'L'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_125_naocl_chl_tank_fill_gal'])
    column_cell = 'R'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_raw_iron_as_fe_mgl'])
    column_cell = 'S'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_raw_ph'])
    column_cell = 'T'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_treated_chlorine_poe_free_mgl'])
    column_cell = 'U'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_treated_chlorine_poe_tot_mgl'])
    column_cell = 'V'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_treated_iron_as_fe_mgl'])
    column_cell = 'W'
    sheet_dv[column_cell+str(row)] = (f.attributes['h_treated_ph'])

book_1773.save("D://Python/testsave1.xlsm")
```

The python script pulls data directly from the feature service that was set up by Survey123 and we can pull up variables from the attributes command. From there, we can iterate the data via the date it was collected while the categories remain the same for all of the reports. This is just the Deer Valley portion of the script. There are 9 other iterations that run to complete the other wellhouses.

MOR

	METERED TREATED WATER				IRON FILTRATION				CHLORINATION					CHEMICAL ANALYSIS/RESIDUALS					
														Raw/Pre-Filter		Treated/ Post-Filter			
Day Of Month Initial	Effluent Meter Total	Total Water To System	Total Water To System	Current Well In Oper.	Amt. Of Cells In Oper.	Head Loss Thru Filter	Amt. Cells Back-Washed	Back Wash Waste Water	12.5% NAOCL Available Chlorine Tank Volume/ Fill Level		Total Available Chlorine Used	Total Available Chlorine Used	Total Available Chlorine Applied	Iron as Fe	pH	Chlorine POE MG/L		Iron as Fe	pH
	M. GALS.	M.GALS.	M.LBS.	NO.	IRON	PSID	IRON	M.GALS.	GALS.		GALS.	LBS	MG/L	MG/L		FREE	TOTAL	MG/L	
1	2	3	4	5	6	7	8	9	10		11	12	13	14	15	16		17	18
1	121.317	0.076	0.634	2	4	7.94			31.86		2.92	3.650	**5.759**			0.19	0.22	0.01	
2	121.393	0.014	0.117	2	4	9.45			28.94	39.95	0.55	0.688	**5.888**	1.31	7.51	0.18	0.21	0.02	7.47
3	121.407	0.000	0.000	OFF	OFF	0.00			39.40		0.00	0.000	**0.000**						
4	121.407	0.018	0.150	OFF	OFF	0.00			39.40		0.66	0.825	**5.496**						
5	121.425	0.051	0.428	2	4	7.30			38.74		2.15	2.683	**6.268**			0.15	0.18	0.02	
6		0.051	0.428								2.15	2.683	**6.268**						
7		0.051	0.428								2.15	2.683	**6.268**						
8	121.579	0.013	0.108	2	4	11.41			32.30		0.50	0.625	**5.765**	1.33	7.51	0.72	0.78	0.03	7.46
9	121.592	0.000	0.000	OFF	OFF	0.00			31.80		0.00	0.000	**0.000**						
10	121.592	0.103	0.859	OFF	OFF	0.00	4	0.0320	31.80		5.34	6.675	**7.770**						
11	121.695	0.034	0.284	2	4	1.31			26.46	39.67	1.10	1.375	**4.849**			0.17	0.23	0.01	
12	121.729	0.041	0.345	2	4	1.00			38.57		1.34	1.675	**4.859**			0.08	0.10	0.00	
13		0.041	0.345								1.34	1.675	**4.859**						
14		0.041	0.345								1.34	1.675	**4.859**						
15	121.853	0.100	0.834	2	4	1.06			34.55		3.35	4.188	**5.021**			0.07	0.10	0.00	
16	121.953	0.008	0.067	2	4	1.92			31.20		0.22	0.275	**4.122**	1.29	7.48	0.47	0.52	0.00	7.47

This is an example of an MOR that was generated from the process. The water treatment crew filled out surveys on their phone for the month, then I ran the python script to generate this report using that data. Then Tom QA's the data and sends it off to the MDEQ. So that means that wording is the same on the survey as the reports so for example, groups are titled "meter treated water" on both, and questions are called "effluent meter total" on both. I built a macro on the MOR excel sheet that automatically calculates certain data like the columns in blue show, acting exactly like a function in excel would. For example, converting gallons to pounds above. However, the macro takes care of irregularities that a function would not be able to account for like weekends, for example.

Populating More Monthly Data

- Arsenic Results Python Script
- Arsenic Filtration required in applicable wellhouses MORs
- This substance needs its own report

```
# write query to excel - arsenic 1773 manors
for row, f in enumerate(ars_query_dm):
    unix_time = str(f.attributes['h_week_of'])[0: 10]
    unix_time = float(unix_time)
    date = datetime.datetime.fromtimestamp(unix_time, tz_utc)
    week = date.strftime('%x')
    column_cell = 'A'
    sheet_arsdm[column_cell+str(row+17)] = (week)
    column_cell = 'B'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_time'])
    column_cell = 'C'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_initial'])
    column_cell = 'D'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_well_in_op_arsenic'])
    column_cell = 'E'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_total_flow_mgals'])
    column_cell = 'F'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_raw_temp'])
    column_cell = 'G'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_raw_ph_arsenic'])
    column_cell = 'H'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_raw_arsenic_ppb'])
    column_cell = 'O'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_poe_temp'])
    column_cell = 'P'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_poe_ph'])
    column_cell = 'Q'
    sheet_arsdm[column_cell+str(row+17)] = (f.attributes['h_poe_arsenic_ppb'])

book_ars1773.save("D://Python/testsave4.xlsx")
```

A part of the MOR is an arsenic report, which records how much arsenic was originally in the water and how much remained after it was filtered. The wellhouses that have arsenic filtration are required to include this information in their MORs. The arsenic report is included in the survey, but it is saved to its own excel workbook.

Arsenic Treatment System Report

					Raw Water			Filter #1 Eff			Filter #2 Eff			Distribution System POE			
Week Of	Time	Initial	Well in	Total Flow	Temp.	PH	Arsenic	Temp.	PH	Arsenic	Temp.	PH	Arsenic	Temp.	PH	Arsenic	Removal
			Oper. #	M. Gals	°C		PPB			PPB			PPB	°C		PPB	Arsenic PPB
4/2/2019	08:50	BH	2	2.640	12.1	7.51	10.0							12.5	7.47	0.0	10.0
4/8/2019	08:50	BH	2	2.826	12.1	7.51	9.5							12.5	7.46	0.0	9.5
4/16/2019	09:42	BH	2	0.361	12.1	7.48	10.0							12.4	7.47	0.0	10.0
4/25/2019	08:34	BH	2	0.754	11.9	7.51	10.0							12.1	7.47	0.0	10.0
																	0.0

Remarks:

Average Arsenic Removal PPB: 9.875

This is the finalized arsenic report that was pulled from the survey. The raw water going into the plant had between 9.5 and 10 parts per billion of arsenic, but after going through special filters, the water going into the distribution system has 0 parts per billion of arsenic.

What's Next for Independence Township

- Tablets, Excel Online, and Python
- Digital data will be used for statistical analysis
- Preliminary DSMI form can be found at www.michigan.gov/deq leadcopper
- Using Survey123 for cross-connection and backflow prevention

2. Complete the table below.

Estimated Number of Service Connections by Service Line Material

A service line includes any section of pipe from the water main to the building plumbing at the first shut-off valve inside the building, or 18 inches inside the building, whichever is shorter.

Any Portion Contains Lead	Contains Galvanized Previously Connected to Lead*	Unknown			Contains neither Lead, nor Galvanized Previously Connected to Lead	Total**
		Likely Contains Lead	Likely Does Not Contain Lead	Material(s) Unknown		

*If a galvanized line is still connected to lead, it is a lead service line and must be counted in the first column.
**The total number should equal the total number of potable water service lines in your water supply (residential, commercial, industrial, other).

The application I created using Survey123 was doing a great job populating the MORs. In the future, I would like to make not only the MORs digital but all the other data that the water treatment crew collects. This is where tablets with a stylus can be used to replace the paper and clipboards we currently use. Excel Online - with an established data connection - would be an improvement on the current workflow. Python scripting will also still be used to generate MORs or any other reports at the end of the month. The Independence water crew has become more comfortable with technology. And they are ready to tackle the next challenge. In Michigan, the Distribution System Materials Inventory is due to the MDEQ by January 1st, 2020. This is a short form that needs to be filled out to record the total estimated number of service lines made out of each material, sources for that estimate, and level of confidence. By 2025, a full inventory will need to be submitted with attributes for each service line, such as the address, pipe material, and data sources for this information. And that's where GIS will be advantageous.

There are more regulations and substances that we are required by the State of Michigan to test for, and they will always be adding more, but never taking any current substances off. We still have the same amount of time in an 8 hour work day though. This creates a need for applications such as this one to help save on time and other resources. The next substance we will probably have to test for is PFAS. PFAS is from firefighting foam and manufacturing processes and is being found in water supplies across Michigan and other areas around the country. Something that I noticed with our water treatment crew recording a hefty amount of data every day, was that shaving off seconds ends up saving hours over the course of a year. Auto filling a handful of fields and removing irrelevant questions from the surveys helped us save on hours of labor costs while ensuring proper data collection. In conclusion, the new legislation along with new and more efficient methods of data collection to keep up with the ever-changing laws and regulations, helps Michigan communities avoid crises like Flint's. They also keep those in charge in check if they decide to not follow them. Applications like the one we implemented help us keep up with the ever-growing list of contaminants we need to test for in the water. And can be translated to all kinds of other data collection workflows to provide the benefits shown here.

Sources

2018 Rule Revision Summary [PDF]. (2018, July). MDEQ.

Denchak, M. (2019, May 22). Flint Water Crisis: Everything You Need to Know. Retrieved from https://www.nrdc.org/stories/flint-water-crisis-everything-you-need-know

Flint Michigan Water Crisis: Its Timeline and Latest Update. (n.d.). Retrieved from https://theberkey.com/blogs/water-filter/flint-michigan-water-crisis-its-timeline-and-latest-update

Kelsey J. Pieper, Min Tang, and Marc A. Edwards
Environmental Science & Technology 2017 51 (4), 2007-2014
DOI: 10.1021/acs.est.6b04034

Moore, Nicole Casal. "Google-Funded Flint Water App Helps Residents Find Lead Risk, Resources." *University of Michigan News*, 8 Dec. 2016, news.umich.edu/google-funded-flint-water-app-helps-residents-find-lead-risk-resources/.

The Flint water crisis begins. (2018, May 29). Retrieved from https://www.history.com/this-day-in-history/the-flint-water-crisis-begins

https://www.history.com/this-day-in-history/the-flint-water-crisis-begins

https://www.michigan.gov/documents/deq/deq-dwmad-cws-2018_Rule_Revision_Summary_631411_7.pdf

https://www.google.com/maps
https://news.umich.edu/google-funded-flint-water-app-helps-residents-find-lead-risk-resources/
Min Tang and Kelsey Pieper
Flintwaterstudy.org
https://www.nrdc.org/stories/flint-water-crisis-everything-you-need-know
https://theberkey.com/blogs/water-filter/flint-michigan-water-crisis-its-timeline-and-latest-update

Presenter Info

Thank you to my GIS team, the DPW director, the water treatment crew, and Independence Township

Randy Winowiecki

GIS Technician

rwinowiecki@indtwp.com

A big thank you to Doreen Groth, Mark Brown, David McKee, Tom Graham, Tim McArthur, Bruce Houck, and the Township of Independence.