John.Nerge
@brooklynpark.org
City of Brooklyn Park
Minnesota, USA
4 May 2015

Building and Maintaining Flexible, Focused GIS Web Apps

A guide to keeping your GIS house in order

Brooklyn Park
Unique. United. Undiscovered.

Contents

Introduction

At the City of Brooklyn Park, we use a mix of ArcGIS and Esri Partner software to publish apps that are quick to build and easy to use. Integrating with existing city systems reduces the need to create new data manually. The app designs are reusable, so building new apps is quick and effective. Since the apps focus on a specific group, adoption is healthy and support is minimal.

This paper focuses on our methods and strategies for maintaining a healthy and efficient GIS. Doing so is, in our opinion, a prerequisite to successfully maintaining a focused GIS app collection. Without a solid foundation, sustaining a modern GIS is an unrealistic objective.

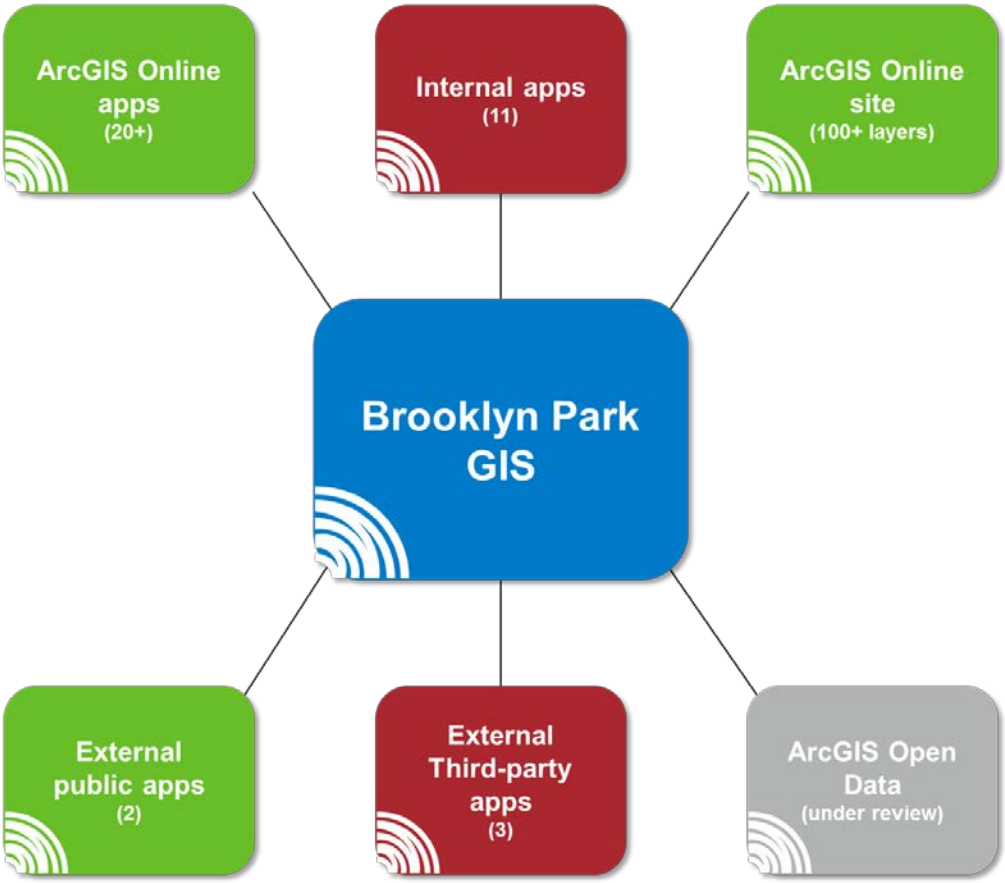
We could write a paper on each of these topics, but our goal is to provide a brief, holistic view of our techniques and approaches.

Welcome to Brooklyn Park

With a population of 77,000, Brooklyn Park is the sixth largest city in Minnesota. We are a proud and active community with numerous youth, adult, and senior organizations, including important citizen boards that enrich our community. As part of our Community Engagement Initiative, we always make residents and stakeholders our primary focus when it comes to important decision making. Our mandate is to engage Brooklyn Park residents so they can share their diverse viewpoints openly and honestly.

Brooklyn Park GIS

Our GIS staff is in our IT division and includes one GIS Coordinator and one GIS Intern. Our primary mission is to support the city's departments by designing GIS apps that quickly deliver consistent data for internal and public use. Currently, we host 11 internal and 5 external GIS apps on premise. We supplement this work with ArcGIS Online, where we've connected and configured our organization's account with 100 local and regional map layers and published over 20 apps.


ArcGIS Online apps (20+)
Internal apps (11)
ArcGIS Online site (100+ layers)
Brooklyn Park GIS
External public apps (2)
External Third-party apps (3)
ArcGIS Open Data (under review)

Our GIS app collection

We have 12 part-time GIS users throughout our organization. They're our main data creators and do an excellent job with it. We assist them with cartography, spatial analysis, and data mining. We also get spatial data from county and state agencies.

Our GIS software packages include:

- ArcGIS for Desktop®, Server®, and Online®
- SQL Server® enterprise geodatabases
- Geocortex Essentials®
- Limited use of Cartegraph SIGNview®
- Limited use of Autodesk AutoCAD®

Data Integration and Alignment

All of our large information systems are SQL server-based. Most of the data have either an address or parcel ID. Because of this, everyone at our organization that does data entry is a potential GIS data creator. We use their data to create and update GIS datasets with a mix of tabular joins, geocoding, and using longitudes and latitudes from parcel centroids. We also connect to live data using Geocortex Essentials' database connection capabilities.

Our main information systems are:

- New World Logos® for parcel management and financial data
- New World Aegis® for police records
- CAMAvision® for assessing information

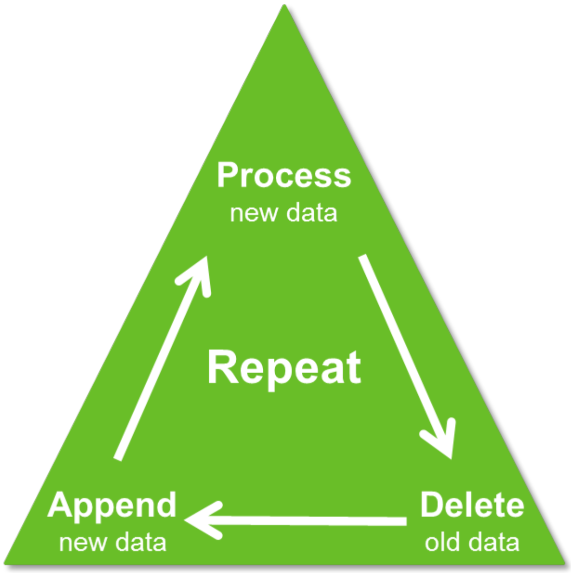
When choosing and implementing software, we strive to follow all of our vendors' suggested best practices. Since we use Esri technology, we want to stay aligned with their long-term plans. So far, doing so has been very beneficial. Transitioning from one version to another has caused minimal issues. We also follow their recommended, configuration-based approach instead of doing custom programming. We discuss the benefits of doing this later.

The Power of Python

Our main data maintenance tool is Python. We use it to automate data updates, generate daily email alerts for our inspectors, perform database tuning, and more. We can't overstress its value.

Scripting for Reuse

We follow a basic workflow when writing a data update script: process, delete, append, repeat.



Process, delete, append, repeat (it even rhymes)

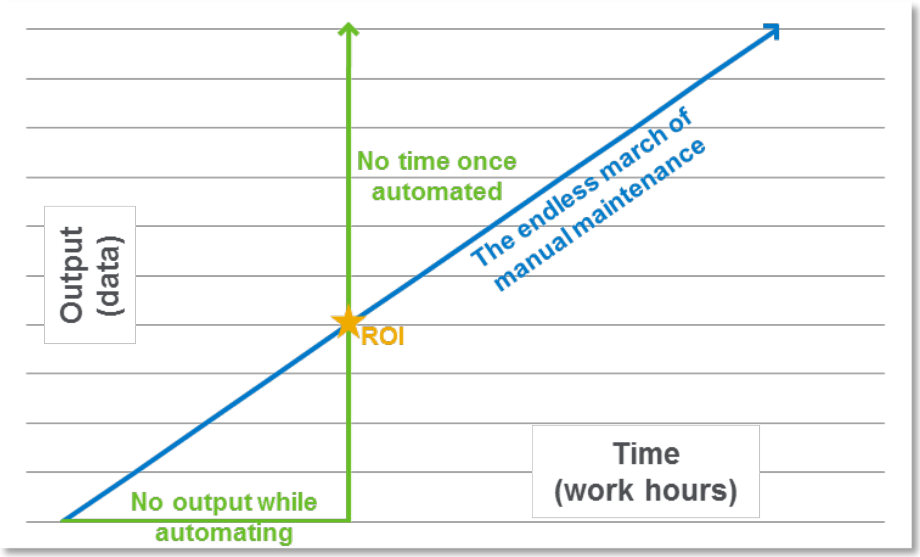
We process the new data, which usually means creating a SQL view in one of our systems to make the data GIS-friendly. Depending on the data source, we either append a cut of the view (Table Select) to existing GIS data, or we geocode the records using an address, parcel ID, or street intersection. Then we purge the old data (Delete Features) and upload (Append) the new data. Because this process is so common, we can cut and paste lines from existing scripts into new scripts for new data we want to create and maintain. We could write more complex scripts that use temporal, spatial, and attribute queries, but doing so would make it harder to reuse code pieces. Instead, we build that logic into our SQL views whenever possible.

A Zero Maintenance Model

When we add a new dataset to our collection, we always include a maintenance plan for it. Whenever possible, we automate the plan. Our goal is to have a zero maintenance model in which the majority of our GIS data is self-maintaining and requires no human work hours to stay up to date.

Currently, only 25% of our 160 main GIS datasets require regular, manual maintenance. 75 datasets don't need any regular maintenance (district boundaries, office locations, etc.). 45 datasets have automated maintenance that runs anywhere from quarterly to daily. The last 40 datasets need regular, manual maintenance. Most of these are utility layers, which need updates whenever there's new construction (i.e. utility layers). As such, there's no practical way to automate maintenance since someone has to create the new data as we install, maintain, and replace our city's infrastructure. Even so, we're exploring ways to streamline manual maintenance and automate pieces of it.

The upfront cost to automation is work hours with no output. There's a period of time where manual maintenance is more cost-effective. However, once you automate, your output rapidly catches up.



Automation quickly makes up for lost time and surpasses manual work

The gap between the two methods is your return on investment (ROI) gap. We quantify our ROI gaps using this formula:

$$\frac{time_a}{time_m} * frequency = ROI\ gap$$

Where $time_a$ is the total amount of time to automate the task, $time_m$ is the amount of time it takes to do the task manually, and $frequency$ is how often you perform the task.

As an example, let's say you spend two hours automating a task that takes you twenty minutes to do manually, and you do this task every other month. The formula would be:

$$\frac{120\ minutes}{20\ minutes} * 2\ months = 12\ months$$

In other words, it will take a year (six iterations) of running the automated process to close the ROI gap. While that seems like a long time, keep in mind that everything after that first year is pure work output without any human work hours. Your saved time is also available for doing other work.

When full automation isn't an option, we still cut out as many manual tasks as possible. We also practice gradual automation in which we automate one piece of a workflow each time we have to do it. In fact, most of our automated tasks started this way. Rome wasn't built in a day, but the Romans still built every day!

To Configure or Customize

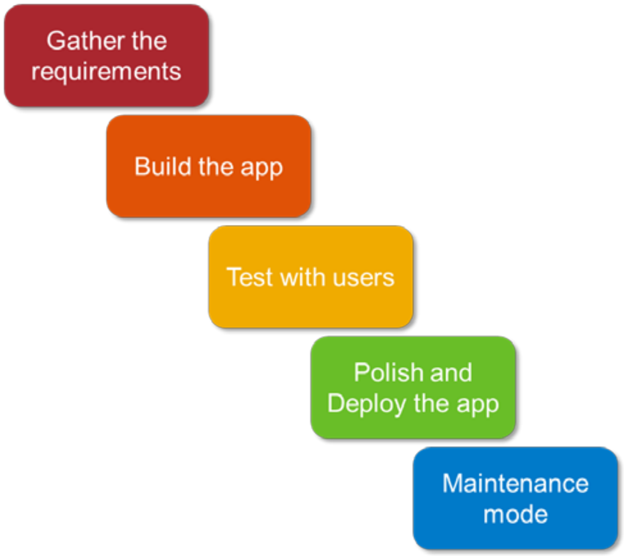
Our mayor likes to say that we should be either first or best in whatever we do. Even though being cutting edge is exciting and comes with bragging rights, we rarely try to be first. It's a lot of work, and it's even more work to stay first. Instead, we strive to be the best.

Every new project begins with a review of similar, exemplary projects. In the information age, there are very few original ideas. The odds are that multiple people have already done your project and found good solutions for it. Why not use them? Furthermore, they likely made decisions that you can make better because of newer technology and your personal insight about your data.

If we do custom programming, then we're doing a process manually that a vendor normally does for us (and then manually updating it whenever they release a new software version or API). In most cases, the added functionality isn't so critical that it's strategically worth custom programming. We've seen rapid changes to ArcGIS Online in the last year, so much so that we've had to go back and update our apps to use new tools and settings that we didn't have before. These changes have encouraged us to wait for Esri to do the work instead of building things ourselves. Custom programming would have been a waste of time because the newest release does it for us. That reveals another pitfall of customization: you lose the output and your work hours because an upgrade has replaced both. All of that hard work to be first doesn't add up to a valuable effort.

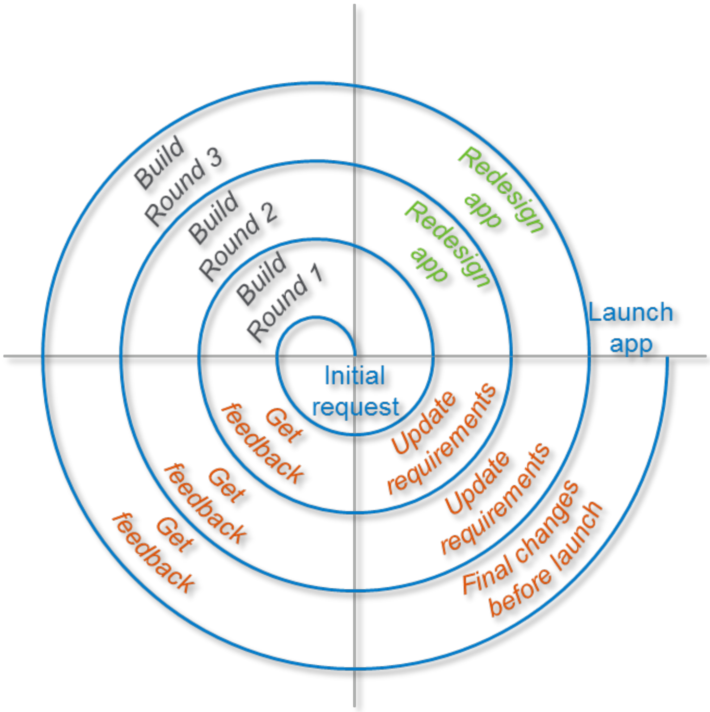
Waterfalls aren’t Flexible

Most of us are very familiar with the waterfall model of design. It usually looks something like this.



The classically sequential (and ineffective) waterfall design model

The main problem with the waterfall model is that it doesn’t work. There’s no flexibility built into it for new requirements that invariably arise. When's the last time your users knew everything they wanted from square one? We prefer Boehm’s spiral model, which emphasizes iterative design.

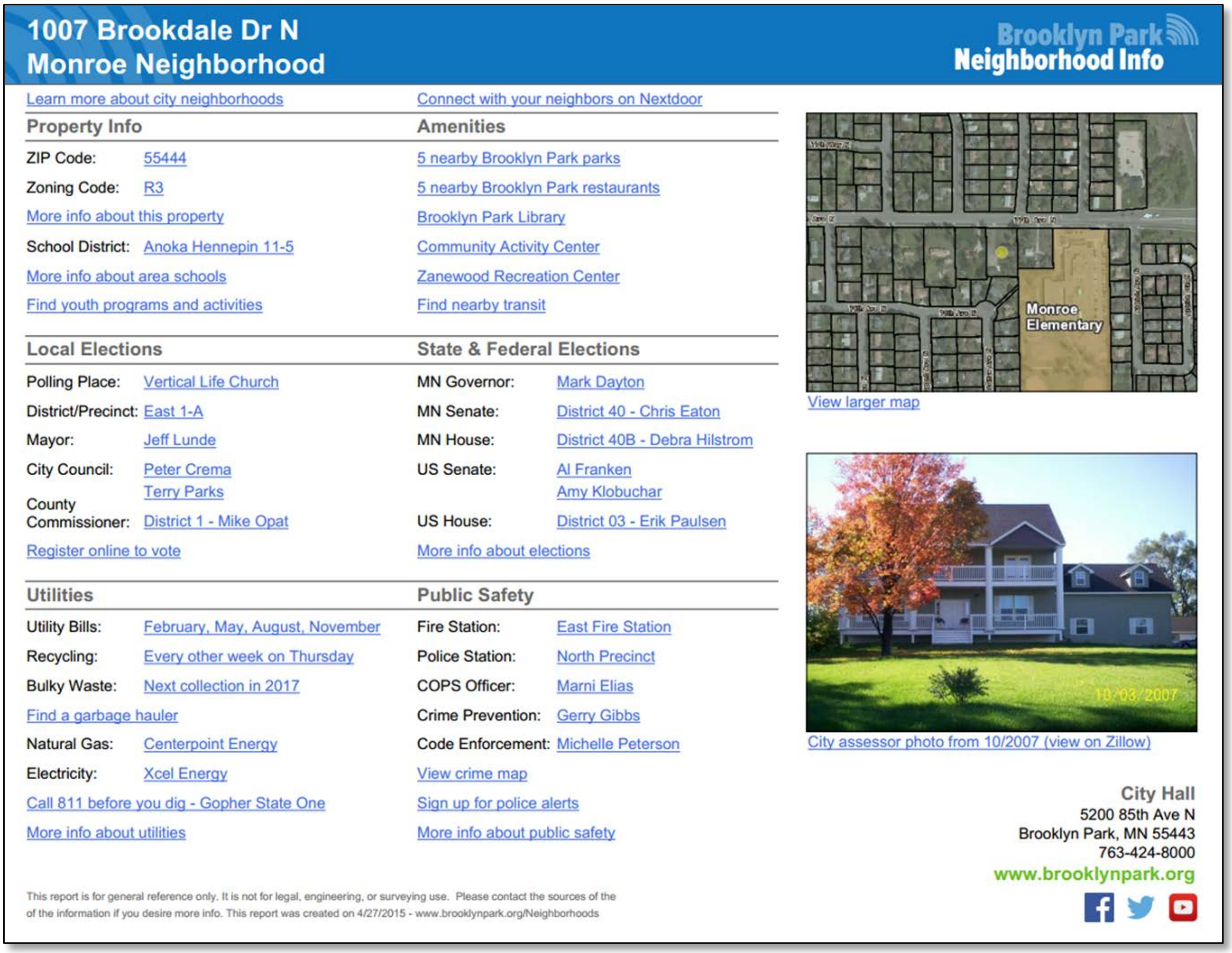


Spiraling toward a better final design

During each build round, we focus on developing only the parts of the application for which our customers have determined requirements. In most cases, we discover new requirements based on what we've built so far. We can adjust the app relatively easily because we haven't fully developed it yet. With each round, we spiral closer and closer to a final design. The app benefits from this process because we took the time to look back at each piece to see if we should change it to fit better within the new requirements.

Case in Point: Neighborhood Info

We combined of all of these methods to design and build our Neighborhood Info app. It's our one-stop shop that delivers all of our content to staff, residents, and businesses. The report includes nearly 50 address-centric data points, all of which link to more information.



1007 Brookdale Dr N
Monroe Neighborhood

Learn more about city neighborhoods

Connect with your neighbors on Nextdoor

Property Info		Amenities
ZIP Code:	55444	5 nearby Brooklyn Park parks
Zoning Code:	R3	5 nearby Brooklyn Park restaurants
More info about this property		Brooklyn Park Library
School District:	Anoka Hennepin 11-5	Community Activity Center
More info about area schools		Zanewood Recreation Center
Find youth programs and activities		Find nearby transit

Local Elections		State & Federal Elections	
Polling Place:	Vertical Life Church	MN Governor:	Mark Dayton
District/Precinct:	East 1-A	MN Senate:	District 40 - Chris Eaton
Mayor:	Jeff Lunde	MN House:	District 40B - Debra Hilstrom
City Council:	Peter Crema Terry Parks	US Senate:	Al Franken Amy Klobuchar
County Commissioner:	District 1 - Mike Opat	US House:	District 03 - Erik Paulsen
Register online to vote		More info about elections	

Utilities		Public Safety	
Utility Bills:	February, May, August, November	Fire Station:	East Fire Station
Recycling:	Every other week on Thursday	Police Station:	North Precinct
Bulky Waste:	Next collection in 2017	COPS Officer:	Marni Elias
Find a garbage hauler		Crime Prevention:	Gerry Gibbs
Natural Gas:	Centerpoint Energy	Code Enforcement:	Michelle Peterson
Electricity:	Xcel Energy	View crime map	
Call 811 before you dig - Gopher State One		Sign up for police alerts	
More info about utilities		More info about public safety	

View larger map

City assessor photo from 10/2007 (view on Zillow)

City Hall
5200 85th Ave N
Brooklyn Park, MN 55443
763-424-8000
www.brooklynpark.org

This report is for general reference only. It is not for legal, engineering, or surveying use. Please contact the sources of the of the information if you desire more info. This report was created on 4/27/2015 - www.brooklynpark.org/Neighborhoods

Our interactive neighborhood info report

Before building anything, we studied our exemplars. We found similar apps created by a neighboring city, Saint Louis Park, the cities of Los Angeles, Boston, and Saint Louis, and Fairfax County, VA. We got valuable feedback about how their residents use their apps and thought twice about the kinds of devices our app needed to run on. By first doing this study of past designers, we were able to take the best things from their apps and combine them with some ideas of our own. As we just did in this paragraph, we remind you to give credit to your exemplars where and when it is due!

We used SQL server to integrate data from our systems and tie them to our parcels. A scheduled Python script updates the data monthly. Instead of rebuilding functionality, we link to ready to consume services like Google Maps' transit finder, Zillow, and various pages on our city website. We also link to configured ArcGIS Online maps and apps, all of which connect to our live, enterprise geodatabase.

We built the application in phases and got feedback from staff and outside groups after each phase. We adjusted the app and its tools as the requirements changed. The final product is a streamlined app featuring a single page report in both an online, interactive format and a ready to print format that substitutes hyperlinks with physical addresses and URLs. Our existing maintenance plans feed into this application, meaning we didn't create any additional ongoing work. For all intents and purposes, it's a self-sustaining city service.

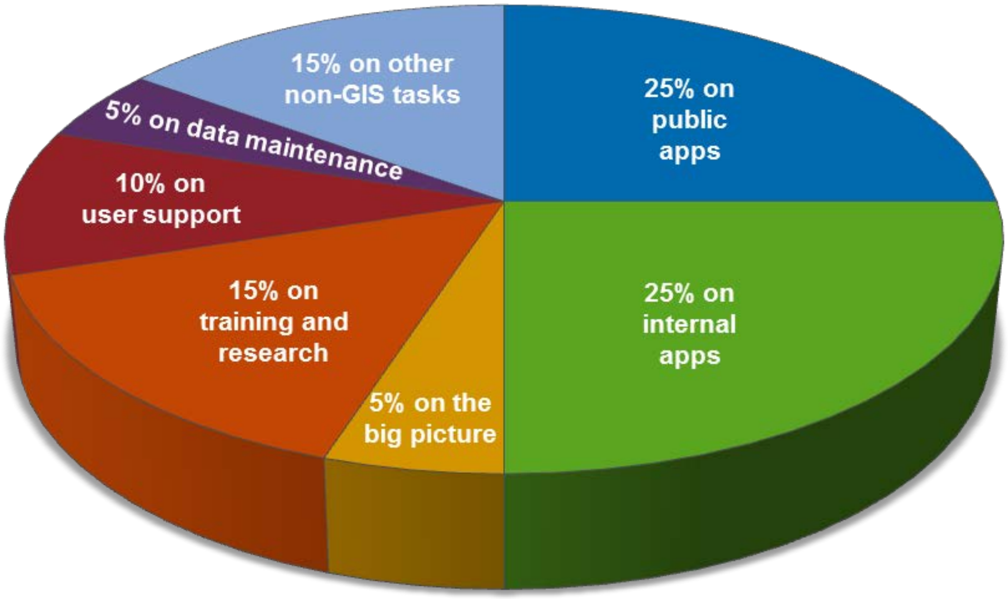
We've had great success with this app so far. We're averaging about 30 visits a day, a metric we can measure thanks to a Python script that reads and processes our web server log every morning. Anecdotally, we've received very positive feedback from our residents. We have a six month outreach plan for the app, so we expect usage to grow.

The End Result

All of this work ensures we have a flexible, integrated, and minimal maintenance GIS. With that in place, we can build apps to consume our GIS data and make it broadly. Our apps are flexible because the underlying system they connect to is flexible. When configuring apps, we apply the same build for reuse strategy that we use with our Python scripts. We study our exemplars and use an iterative design approach, both of which ensure we're focusing our apps on our target audience. Internally, we've had a growing adoption rate of using our GIS apps because we've built them to meet the department's unique needs.

Along with our Neighborhood Info app, we track usage for our other GIS apps. Currently, about 15% of our staff uses one of our GIS apps daily. We regularly receive praise and thanks for making our staff's jobs easier and more effective. On ArcGIS Online, we've had over 4,600 app views and 37,000 layer views since officially launching the site five months ago. We don't have metrics on how we've reduced phone calls requesting information, but the most popular apps and layers are the ones that provide information we normally get calls for, such as zoning codes and city office locations.

It would be a lie to say that setting all of this up is anything but a huge undertaking. It takes a lot of time and thought to develop a comprehensive GIS strategy, and during that time you don't have a strong, tangible product to present to your supervisors. Though the upfront cost is notable, the long term benefits are worth it. Beyond having a stable environment in which to build apps, we also have a strategic direction to drive and defend our business decisions. The end result for our work load is that we only spend about half of our time on application development.



Work load distribution for Brooklyn Park's GIS Coordinator

We spend 15% of our time helping GIS users and doing data maintenance that we can't automate. Another 15% of our time goes to non-GIS tasks (i.e. other IT-related work). We even have 20% of our time left over to attend training, do research, and keep our big picture goals aligned so that we can keep up with the industry and best practices.

We didn't establish our strategy overnight. These standards took three years to formalize. We developed them gradually and non-linearly using the same spiral model we use for application design. We're also constantly adjusting our standards as we discover better ways to do things from new technology and other organizations. For example, it doesn't make sense to automate everything. Sometimes our ROI gap formula shows that a time frame isn't financially viable. In those cases, the time to get ROI is often longer than the expected life span of the process. New technology may help in the future, but for now some manual work is still the most efficient method. As you develop your own strategy, remember to keep your end goal in mind, which is to have a smoothly operating GIS, whatever that means for your organization now and in the future.

Your Friends in IT

As a final thought, we want to stress the importance of aligning your GIS goals with those of your IT division. Our other IT staff don't do any GIS work, but we still have close ties to them. They provide essential support for our databases, servers, and web service management. They're also the stewards of admin rights and channel customer feedback to us. Best of all, staying closely aligned with IT means we get access to the newest technology, which is extremely valuable when trying to develop modern GIS apps. If you're looking for a starting point for modernizing your GIS, then begin with solidifying your professional relationship with your IT division.

Sources and Inspirations

A major influence on our strategy is Frederick P. Brook Jr.'s book, *The Design of Design*. We also paid close attention to the Port of Rotterdam's Esri UC 2014 plenary presentation on making everything accessible within three clicks. We use that benchmark as a measurement when assessing our apps' designs and focus.

Several of the images in this report borrow from existing models. Our spiral model is a simplified version of Barry Boehm's more comprehensive design. We based our automation vs. manual work chart on a similar chart published to howtogeek.com. We provide links to these resources below.

Source Links

Boehm's Spiral Model: http://en.wikipedia.org/wiki/Spiral_model

How-To Geek's repetitive tasks chart: http://www.howtogeek.com/102420/geeks-versus-non-geeks-when-doing-repetitive-tasks-funny-chart/

Contact Info

John Nerge, GISP

5200 85th Avenue

Brooklyn Park, MN 55443

1.763.493.8196

john.nerge@brooklynpark.org

Our Links

Map gallery: www.brooklynpark.org/maps

ArcGIS Online site: brooklynpark.maps.arcgis.com

Neighborhood Info app: https://cityview.brooklynpark.org/neighborhoodinfo