* **What the mapping group does**

1. a collective of 15 of the greatest, awesomest, collaborativest, most efficientest GIS and data management professionals in the whole wide world.
2. We work with esri arcmap every day. Do all of our analysis with arcmap. Store data in sde database.
3. Go through mileages. Talk our public products: maps, txdot\_roadways, statewide planning map.
4. Talk about county road project requiring data collection

* **County road and current data collection**

1. County road updates come from inventory submissions by individual counties
2. Bi-annual process which involves office review of submissions and field work when necessary (bad aerial, new road, vegetation, etc.)
3. Trimble pathfinder xh with antenna requires laptop and mxd with preloaded data.

Trimble unit

Antenna

Laptop

Cables

Usb adaptor

Vehicle desk

Airbag shut off

1. Cumbersome, lots of equipment. Limited amount of trimble units
2. Regional Data Collection project to use employees from across the state to help collect data is the field. Create datasets for project reference, maintenance purposes.
3. Project can be expanded for all types of data we can use for analysis
4. Juno SB handheld has older interface. Clunky and not intuitive. Only device we can afford to purchase on mass scale for everyone. Troublesome for less GIS experienced individually insofar as unloading, QCing, and posting data
5. What can we do? What are our options? There has got to be an easier way, which cost less money, to get the data back into our MXD for use in map making, analysis, inventory updates, etc. Smart devices are everywhere. People are attached to them, even emotionally invested. Capable of mapping grade location information. How do we collect and make that information accessible to our mapping group?

* **Tech application ideas**

1. Android vs iOS vs windows? App store?
2. Simplicity is golden, HTML 5. Accessible by any device with the internet. Simple language, self-taught. Lots of examples on the internet. Whole project put together in a couple months.
3. Basic language education, self-taught, reference examples prevalent on the internet to put together a simple page
4. Geolocation API allows GPS location from just 2-3 functions.
5. Local Storage:

“With HTML5, web pages can store data locally within the user's browser.

Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used ONLY when asked for. It is also possible to store large amounts of data, without affecting the website's performance.” –w3 schools

1. Quick cache.manifest to save interface of site. Use GPS to save locait info until you have service again
2. Input text boxes, radio buttons, dropdown options, and other basic form information with list of location coordinates to save attribute to local storage. Text based way to save. (Reduces error, will get into that later)
3. Take advantage of the internet connection. Upload data from storage to a database for use immediately in the office. Can still be in the field collecting data and office can already be analyzing. Let you know what you missed, what doesn’t turn out right.
4. This idea was laid out and experimented with. Guess what, we did it.
5. Primitive layout established for testing. Immediately, as putting this application together, the possibilities started to become blatantly apparent and endless.
6. Design customization was common sense. Permit only specific form/attribute information options. Limit dropdown and text field requirements to ensure data dictionary and schema compliance.
7. Points vs lines. Discovered there is no reason we can’t collect both. Choose the type of feature. We want different attributes for feature type, we need different schema for each. Soo…..
8. Make attribute options dependent upon other options. Make subtypes like in arcgis so data is compliant.
9. Geolocation position information includes many additional metadata traits. Accuracy is one of them. Since the application simply pulls the Geolocation position and we just save it locally as we wish, accuracy requirements were implemented to help ensure quality. If the accuracy rating is better than (let’s say) *10 meters* then save the data to local storage.
10. Why can’t we have a map for reference? Google map service available for testing purposes. Only a few lines of code as well. Of course, plenty of examples all over the internet.
11. So we have already implemented a few cool ideas to customize this application and design it specifically for data collection. We can pull information and save it locally to a device. But how to get the data back to office, uploaded to a useable database?
12. Self-education scenario revealed a simple php script can connect the application to a database.
13. Initial issues arose with our security and firewall. Application is not worth compromising the firewall and creating holes within it to push data onto the network. But how do we get the data onto the network?
14. Much easier to post it on a server outside of our network and then retrieve it at will. Not a problem to step outside the firewall.
15. Reviewed multiple types of databases to satisfy this requirement. Doesn’t need much since the database only standing as a temporary space to unload data from a device and save it until it is retrieved from the office.
16. Postgres database was chosen due to it being open source—Lots of documentation (like ArcGIS) and tools available. Python compliant and fast which is great for getting the data from the Postgres database to the SDE database we use for our entire inventory.

* **Testing results and accuracy - truly mapping grade?**

1. We designed this application because we want mapping grade location information at ease. Is it accurate enough? We tested the application on multiple smart devices: iphones, ipad, and many android phones.
2. Simultaneously collected data on the Trimble SB handheld and Trimble Pathfinder setup. The pathfinder is rated to be much more precise than the handheld and we knew that going in. Assume it as a standard to measure location error. Where specific projects and data may require that increased accuracy, we weren’t looking for a reason to throw it away. Just looking for the benefits of convenience and cost to collect mass data simultaneously.
3. Overlay all collected data in ArcMap to identify errors, inaccuracies, and inconsistencies at the office acceptable standard scale 1:3000 ft
4. Lines and points at different speeds to develop any trend of error and/or window of accuracy based on speed
5. Points. Instant pull versus multiple points average trimble
6. Signal loss
7. Storage capacity
8. Device specific trends of error. Iphone privacy settings. Browser specific performance.
9. We have a couple screen shots from the overlay.
10. LINEWORK: this roadway was collected at a moderate 35-40mph. can you see we have multiple lines on this roadway? Red, blue and orange. Tough to tell? That’s a good problem. Red is the Juno Handheld, blue is the pathfinder (the most accurate by capability), and the multiple orange lines are various smart devices.
11. The various devices consistently proved more accurate relative the pathfinder. Different devices hovered around the pathfinder linework in different ways. The juno trended an offset. This kind of offset can be fixed with post-collection office correction but adding steps is what we are trying to avoid. We are trying to streamline and simplify to remove chance for error.
12. POINTS: some major revelations and ideas came from the point testing. the pathfinder was not set up for point collection when we performed the field work. Used aerial based location comparison alongside the juno comparison.
13. When sitting still (#4) the juno would collect multiple point locations and average them together for increased accuracy based on precision analysis. The application currently pulls single instantaneous location information. This proved the juno more accurate when sitting still for a period of time and letting the juno collect multiple points. As soon we attempted to incorporate movement the juno accuracy tumbled as it was then forced to utilize a single instantaneous location versus the application which acquires directional data and proved more consistency amongst multiple devices (#1, 2, 3). This proved we should add speed recognition to the application to handle stand-still collection differently from moving collection.
14. SIGNAL LOSS: one major issue which arose in testing was signal strength. Multiple times the juno took several minutes to connect--- a lot of waiting around and holding the unit out the window to try and get a signal. Versus the pathfinder which didn’t have any issues gaining a signal and neither did the applications.
15. Maintaining a signal was the real issue. Both trimble units lost signal in the middle of collecting linework at some point during testing. The pathfinder did so multiple times. This wastes time, money, manpower because the entire collection must be restarted. The application never had an signal issues of any kind.
16. LOCAL STORAGE CAPACITY: never any issues with storage capacity. Was able to record straight data for over 5 minutes without any issues. 1 character=1 byte. Typical storage capacity is ~5MB which is 5 million characters.

* **Best practices and specifics noticed**

1. Default browser for device brand. Safari for apple, chrome for android
2. Safari “Private browsing” settings
3. Fluid consistency of speed while collecting a line
4. Do not allow the device to go idle or to sleep

* **Benefits:**

1. Cost. – no licenses, no extra software purchases. Can be put together with just the cost of a few days or weeks of putting it together and self-education. Exponential speed so once you have an idea of how to work with it then speed picks up.

Juno-$1390

Pathfinder-$2000

PostgreSQL database-$337/3year ($222/1 year)

Ipad-<$300 Android-$50

1. Hardware reduction. Lose or break the device? No problem!
2. Training and Error

Juno- Non-intuitive. Requires specific data dictionary. Slow, old, windows 95. Problem with non-GIS educated people working with it

Pathfinder- same non-GIS user problems navigating within arcmap. Overwhelming and bulky.

Both trimble methods require post office processing which has embarrassingly yielded issues.

Application is intuitive, simply, easy to figure out. Upload data and its done. No extra training required or room for error.

1. Battery requirements

Trimble units require pre-charged batteries. Spare batteries cost can cost a couple hundred dollars. If batteries didn’t charge or don’t last the whole day then the day is lost.

Smart devices all have available car chargers, usually cost 20$ or less

1. Utilized by more people simultaneously

Due to limited costs and budgets, acquiring mass numbers of hardware devices is restrictive to speedy and efficient collection. The plethora of devices available dissolves this issue

1. Instant upload and office use. No lose data if device malfunctions or breaks. Can be in the field still while data is being used. If you missed something you can get a phone call and get what you missed before returning to the office
2. Points while collecting lines. Multitask reduces time and cost to run a roadway multiple times. Points will be snapped already

* **Future improvements**

1. Security- login and password
2. General improvement of layout/interface
3. Hide/show map with alternative basemaps
4. Smart device and user metadata
5. ‘idle’ or ‘sleep’ prevention
6. Stand-still point recognition and collection methodology like juno
7. Connect to RTK (real time kinematic) position network for location correction and increased accuracy
8. Collected data review, delete specific records

* conclusion