Web Development Life Cycle

Planning

Analysis

Iterative Process

Testing

Design

Implementation

Maintenance
Technology Used

- ArcGIS Online
- Web AppBuilder
- ArcGIS Arcade
- JSON/GeoJSON
- ArcGIS JavaScript API
- JavaScript/HTML5/CSS
- jQuery
- DataTables
- FME
- Scrum Methodology
- Visual Studio Team Services -
  - Source Control
DEMO

WEATHERVIEW APPLICATION
Planning

- Requirements gathering and scope document creation

- Objectives
  - Commercial-Off-The-Shelf (COTS) Solution
    - Web AppBuilder for ArcGIS
    - Utilize ArcGIS Online web map
  - Responsive
    - Mobile Friendly – Tablet and Smartphone Accessibility
    - Ensure application works on all platforms
Analysis

- Pros and cons of previous application
- Analyze the proposed application

**Pros**
- County forecast table
- RWIS data
- AWOS data
- Stationary popup (infowindow)

**Cons**
- Fully custom (maintenance nightmare)
- Expensive to upgrade and maintain ($11,000 annual fee)
- Not responsive (tabular mobile)
- Utilized Google Maps API
- And more…!
Old WeatherView Application

- 2010 Deployment
- C# / VB programming language used
Old Weather Feeds

- VB Script to process data from SQL Server to Oracle Spatial
- Job was never monitored
- Bad/Missing IDs for relational tables
- Data was old or not accurate
- No Spatial ‘Buddy Check’ for sensors
- No way to share RWIS Images
- Data Feeds failed often due to RDMS bottlenecks.
Design

• Features
  • Mobile/tablet friendly
  • Easy maintenance
  • Quick to upgrade at no additional cost
  • High-Availability REST services
  • Stable environment
  • Cloud based

• Final Data Feeds
  1. RWIS Atmospheric
  2. RWIS Surface
  3. RWIS Subsurface
  4. RWIS Deep Probe
  5. RWIS Traffic
  6. RWIS Camera Images
  7. AWOS Data
     • Metar Feed
     • NOAA/FAA GeoJSON
  8. County Forecasts
Implementation

- Code Development
  - ArcGIS API for JavaScript
  - jQuery
  - DataTables
  - Dojo
  - JavaScript/HTML5/CSS

- Web AppBuilder Framework
  - Prototype
  - Unit Testing
  - Validate application functionality
• **Python Script**
  1. Cleans out directory on FME Server
  2. Copies RWIS Images from vendor FTP directory to FME Server
  3. Generates simple text file of file names
  4. Copies RWIS Images to WeatherView Code Base Folder
  5. Separate FME Server job runs nightly to clear out images in the WeatherView Code Base Folder
Data Sources

- **RWIS (Roadway Weather Info System)**
  - Comprises automatic weather stations that have Atmospheric, Surface, Sub-surface and Deep Probe sensors. Most RWIS sites also have cameras that can provide several camera positions.
  - There are 69 RWIS Sensors in Iowa.
  - Vendor provides data feed to SQL Server at Iowa DOT.

- **AWOS (Automated Weather Observing System)**
  - Automated weather stations at most airports that provide a myriad of weather data including temp, pressure, cloud cover, precipitation, visibility and more.
  - We pull data for 43 AWOS sensors in Iowa.
  - Vendor provides data feed to SQL Server at Iowa DOT.
New AWOS and RWIS Feeds

data.iowadot.gov
Testing and Debugging

- Ensure application works on all platforms
- Test on multiple devices
  - iPhone, iPad, Galaxy, Note, etc.
- Test on multiple OS
  - iOS
  - Android
  - Windows
- Test on multiple browsers
  - Chrome
  - IE
  - Firefox
Deployment and Maintenance Cycle
- Iterative updates
- User feedback through WAB Share widget
- Continue refining backend processes
Monitor Application Status

- Utilize GeoCortex Analytics to monitor application status and usage.
Questions?

Thank You!

Cody.barrett@iowadot.us

@Codyr_barrett