**Product Increment:**

Tagged in bitbucket as Sprint\_6\_Product\_Increment

**Test results and analysis:**

* Task 1: Add/Edit/Delete Notifications
  + Only project owners can add/edit/delete notifications
    - **PASS**
  + When a project owner clicks add notification the form only shows valid options for a notification.
    - **PASS**
  + After a project owner adds a notification it should automatically change the point color based on the type of notification
    - **PASS**
  + If a project owner clicks on a point that has a notification it should bring up the edit/delete notification form and show only valid options. If the current risk is equal to notification zone it displays “you will be sent the notification later if the conditions stay the same”.
    - **PASS**
  + Editing a notification changes the points color and automatically deselects the point
    - **PASS**
  + Deleting the notification changes the point color back to the series color and automatically deselects the point
    - **PASS**
* Task 2: Web statistics
  + Unit testing of date range form
    - **PASS**
* Task 3: Performance testing
  + - Tested 50 random zip codes for how long it took to calculate evaporation rate and generate the graph for a guest user (See Documentation/Test Plan/generateGraphPerformanceTest.txt for full results)
      * Average = 178.48 ms
      * Min = 147 ms
      * Max = 205 ms
      * Median = 183 ms
    - Tested 100 random zip codes for how long it took to retrieve a piece of weather data from NOAA (See Documentation/Test Plan/random zips performance test/ for full results)
      * What we found out was that on average the time was pretty quick. But every once in awhile it would take way too long. Because of this we save weather data, which could completely eliminate having to go to NOAA and get the weather data. It also eliminates having to ask NOAA for the lat/long of the zip code which helps save time.
* Task 4: Compatibility testing
  + IE, Firefox, Chrome, Safari
    - All functionality works the same, subtle differences in general GUI
      * **PASS**
* Task 5: Recovery testing
  + If an admin chooses to reset the database
    - Drop **most** database tables
      * **PASS**
    - Recreate a new admin
      * **PASS**
    - Email all users
      * **PASS**
* Task 6: Deleting a series
  + Deletes all notifications associated with it
    - **PASS**
* Task 7: Deleting a project
  + Deletes all series and notifications associated with it
    - **PASS**
* Task 8: Database interaction classes
  + Unit testing database interaction classes (Change in risk notifications, admin, Future notification, Series)
    - **PASS**
* Task 9: Black box testing of emails
  + Email graph, Future notifications, Change in risk notifications, validate user, forgot password, add user to project, etc.
    - **PASS**
* Task 10: On project load graph gets all series/notifications and correctly displays them
  + **PASS**
* Task 11: Data Mitigation Testing to ITS server/Conformance testing
  + Went through all requirements to see that they function properly
    - **PASS**
* Anomalies
  + **NONE**

**Sprint Review**

* Monday April 24th 1:00 – 2:00
* Zach Smith, Bryan Allen, Mark Grinter, Anne Werner
* Discussion:
  + We demoed what was accomplished in sprint 6
  + Discussed the following topics
    - Emails
    - Add/edit/delete notifications
    - Add/delete series
    - Admin
      * Web usage statistics
      * Reset database
    - GUI
    - About page
    - Privacy policy
    - Footer on graph page
    - Add series form (new form)
    - Forgot Password
    - Success messages
    - Acceptance Testing
* Decisions:
  + See follow up actions
  + Email formats were good
  + Add/edit/delete notifications were good
  + Add/delete series were good (besides add series button text)
* Follow up actions:
  + Update about page
  + Instruction manual
  + Create database reset email to all users if database is reset
  + Add instructions on enter zip form.
  + Add Footer
  + Admin Statistics on one page
  + Change Add series button text to modify conditions
  + Add pop out for the zip code form that says, “Enter the zip code of your project to view the forecast of shrinkage crack risk
  + Pop out for if you want to do more create an account
  + Move the date to next to day on graph, keep month at bottom
  + Add remaining log stats

**Sprint Retrospective**

* Everything is going well. Client had no complaints.

**Overview**

Due to the time left in the semester we decided to have a 3-week sprint. We had 2 problems with ITS during this sprint. The first problem was they had a firewall up blocking us from sending emails from the plastic crack email address. This caused around 10 hours of wasted development time trying to implement the email system. Another problem was there was a problem with PHP and sessions. This caused 1 hour of wasted development time. We underestimated the time it would take to make a good GUI. We thought this would be easy but there was a lot of time spent on this. We also underestimated the time it would take to integrate the database into the application.

**Update Sprint Backlog:**

See Backlog.xlsx, Tab: Sp6

**Create Sprint Burndown Charts:**

See Burndown.xlsx, Tab: Sp6

**Create Sprint Effort and Velocity Charts:**

See “Effort and Velocity.xlsx”, Tab: Sp6

**Update Product Effort Charts:**

See “Effort and Velocity.xlsx”, Tab: Product

**Update Product Backlog**

See Backlog.xlsx, Tab: Current