

# TEAM AMERICA & TOM

For our EPICS project we teamed with the local NPR affiliate, WFYI, to improve the search engine on their website. To complete this project we had to educate ourselves in PHP and MySQL, and research how to properly score search results.

## *Epics Final Report*

### Team Members:

Max Brodbeck

Phil Dwyer

Tom Pritchard

Adam Sanders

4/24/14

Spring Semester 2014

## Table of Contents

### Chapter 1: Introduction

Problem Statement	pg. 3
Motivation and Rationale	pg. 3
Description of Customers and Developers	pg. 3
Overall Approach and Process Model	pg. 3

### Chapter 2: Requirements Specifications

Functional Features	pg. 4
Assumptions and Constraints	pg. 4

### Chapter 3: Architecture

System Services	pg.5
System Structure & Functionality	pg.5
Architectural Decomposition and Style	pg.5
System Platforms	pg.5

### Chapter 4: Implementation

Selection of Implementation Language(s)	pg.6
Coding Standards and Comments Used	pg.6
Implementation Process and Distribution of Work	pg.6
Organization of the code base	pg.6

### Chapter 5: Quality Assurance & Testing

Describe the selection of testing objectives	pg.7
Explain Basic Test Approach and Methods Used	pg.7
Defect Detection	pg.7
Documented Sample Runs	pg.7

### Chapter 6: Project Organization & Management

Team Organizational Structure	pg.8
Management Process	pg.8
Technologies and Tools Used	pg.8
WSR	pg.9-17

### Appendix

Source Code	pg.18-20
Customer Evaluation	pg. 20-21
Peer Evaluations	pg.21-22
Presentation Slides	pg. 22-24

## Chapter 1: Introduction

### Problem Statement:

The goal of this project is to improve the ordering of results provided by the wfyi.org search feature. The goal of the ordering is to increase the relevancy and accuracy of the results based on the user's query string.

The current search feature of WFYI's Web site uses a single table to store a subset of aggregated data from all of the assets on the site. The assets include news articles, TV and radio programs, TV and radio program episodes, events, pages and contests. The details of the asset types aren't as important as understanding that each asset has a common set of meta-data that is written to the search table. This common meta- data is: title, heading (sub-title), URL, Meta Description, content and date. Results will be ranked by matching the user query against the various meta-data fields and assigning a weight value based on criteria such as an exact match, multiple matches, partial matches and no matches of string values a rank will be assigned to each search result. The rank is what will be used to order the results for the user output.

The deliverable for this project will be the implementation of a PHP class. The skeleton for this class will be provided and will include function headers corresponding to the expected method signature of the class. The core function of this class is to assign a weight score to a set of results that are relevant to the given user query.

### Motivation and Rationale:

We believed that WFYI would be a good client to work with based on the feedback Professor Linos gave about them from previous years. Some of us also had interests in the functionality of search engines, and most of us were interested in learning PHP.

### Description of Customers and Developers

We worked with Chris Bowman of WFYI, Indianapolis's NPR affiliate. Our team was composed of four members, Max, Phil, Tom, and Adam. Max is a junior Computer Science major from Lima, Ohio. He is the starting center for the Butler Football team. Phil is a junior Computer Science major from Rochester, New York. In between semesters, Phil works at a coffee shop. Tom is a foreign exchange from the United Kingdom. He is employed at Developer Town. Adam is a sophomore Computer Science major from St. Louis, Missouri. Adam is interning at AT&T this summer at St. Louis.

### Overall Approach and Process Model:

We decided to spend the first few weeks on our project learning PHP. We began work on the searching algorithm after spring break, and would meet every Sunday and possibly Wednesdays if necessary to work on the code. We developed weekly goals for our team to strive towards. Also, we had major milestones that we set up for us to work efficiently on this project. In addition, we would have occasional meetings with Chris to clarify any questions we would have or work out any issues that came up when implementing the code.

## Chapter 2: Requirements Specification

### Functional Features:

This project involved improving the search engine for the WFYI website. Our improvements were meant to return better search results by utilizing a scoring system that compared the user's query to the title and text of an article and then applied our scoring algorithm.

### Assumptions and Constraints:

We initially underestimated the difficulty of the project, which as a result cut down on our testing time for our algorithm. Our virtual environment was more difficult to step up than we initially thought, in fact one of our team members, Adam, was unable to ever set up the environment on his computer. The virtual environment start up set us back a week or two initially, but luckily we were able to catch up with having a few Wednesday meetings.

We also had a late change in our coding style. At first, we were using a mainly PHP style where we would just get a very basic query to trim and rank in a result array. As we started testing this, we noticed we were having difficulties in our query string when the string contained two or more words separated by spaces. After the initial word in the string, the query would not recognize the rest of the string. We researched alternatives and solutions to this problem for about a week. At first, we tried to use the PHP string function `explode()` on the query string, which breaks up the string into an array of smaller strings containing each word in the original string. We were running into problems with this, so we decided to look into the alternatives. The alternative we chose was to introduce a mainly MySQL style where the majority of the work is done by MySQL. The process entails a query that uses like statements to find relevant tuples in the database and sorts them using our scoring system. This process has worked much better than the first approach.

## Chapter 3: Requirements Specification

### System Services:

We created an algorithm that will take a query from a user and compare it to the items in the database. Each item is given an initial score of one, and this number is then multiplied by a certain number depending on the percent match of either the title or article text. This score can be lowered by the age of that item, so that the newest and most relevant items are at the top of the results.

### System Structure & Functionality:

We have a PHP class file that connects to a pre-determined database. This class file will return an array of relevant results to the website, but that is not our concern for this project as we were instructed to focus on the scoring algorithm to return the proper results.

### Architectural Decomposition and Style:

Our architectural decomposition is a client/server relationship, because our code is hosted on a server which is accessible by the users of the WFYI site.

### System Platforms:

Our system that our PHP class file runs on is an Apache server that runs PHP 5.4.10 and MySQL 5.1. In our virtual environment for testing, we actually run PHP 5.3.10 and MySQL 5.1. Despite having an older version, our class file tests perfectly.

## Chapter 4: Implementation

### Selection of Implementation Languages:

We did not really have a choice in languages, as our program had to be implemented into an existing system. We wrote our code in PHP, and within the code we generated MySQL queries.

### Coding Standards and Comments Used:

We used underscore-case for our code, because the preexisting code we added onto used underscore-case. We also used single quotes everywhere possible. We also wrote comments for every function, as well as anything else in the code that we felt needed further explanation.

### Implementation Process and Distribution of Work:

The implantation of our code fell more to Chris than to our team. We sent him the code, and he implemented it on the WFYI servers. He gave us some feedback which we used to further refine our algorithm, and after making these changes we sent the code back to him.

We functioned fairly well as a team. We would have weekly meetings, and during these meetings the work was fairly well distributed. Phil and Adam primarily focused on background tasks like documentation and diagrams initially, though they did help to refine the algorithm. Tom did the majority of the actual coding. Max helped to organize the meetings, communicate with our client, Chris, and helped out wherever else was needed on the team, whether it was coding, documentation, or refinement.

### Organization of the Code Base:

Our code was all contained in a single PHP file. This file would act as an intermediary between the already existing SQL database and the WFYI website.

## Chapter 5: Quality Assurance and Testing

### Selection of Testing Objectives:

The selection of our testing objectives was fairly simple, as our project was a search engine. We wanted to make sure that relevant results would be returned, and that they would be returned fairly quickly. We also asked Chris for a list of the most commonly searched terms on the website so that we could make sure typical searches on the website would return the most relevant results.

### Basic Testing Approach:

Our testing approach was fairly simple. We would change a search term in our main file, which would then interact with the PHP file we created and a SQL database Chris provided for us. Then we would simply run the main file, and see whether or not the returned results were relevant. Also, we would print the score for each result to make sure they were sorting properly in a descending order. We included date as well, so that we can make sure the aging element of score is working.

### Defect Detection:

Our testing helped us to refine our algorithm, and see that initially certain things, like multi-word queries would not function properly. We also saw in initial versions of the code, certain scores were not being applied properly, but through our testing process we were able to correct these bugs. In all testing, the aging element would not work in our sample databases because the timestamps for each item are basically the same, so therefore the aging element treats every item as the same date.

### Documented Sample Runs:

This is an example of a typical sample run; we would edit a text string in our main program. This string would be used by the PHP file to query the database. This page would then be displayed as a webpage so that we could see that our algorithm was working correctly. The ability to see the score allowed us to detect initial errors in code, like improper results for multi-word strings. However, through testing and refinement most of the defects were fixed. A larger version of this picture can be viewed on our wiki.

#### SQL used:

```
select * from search where upper(Title) like '%NO%' or upper(Title) like '%LIMITS%' or upper(Heading) like '%NO%' or upper(Heading) like '%LIMITS%' or upper(MetaDescription) like '%NO%' or upper(MetaDescription) like '%LIMITS%' or upper(Content) like '%NO%' or upper(Content) like '%LIMITS%';
```

Results for: *No Limits*

Count: 2855

- **Title: No Limits**  
Score: 5582.1598780826  
Title %: 33.33333333333333  
Content %: 1.4962593516209  
Time Modifier: 0.97361985919736
- **Title: I Had No Idea!**  
Score: 4306.8728772001  
Title %: 26.086956521739  
Content %: 4.444444444444444  
Time Modifier: 0.9736199417704
- **Title: No Limits - May 5, 2011**  
Score: 4180.172069767  
Title %: 25  
Content %: 3.7037037037037  
Time Modifier: 0.9736199417704
- **Title: No Limits - May 12, 2011**  
Score: 4009.1206624528  
Title %: 24.24242424242424  
Content %: 2.2522522522523  
Time Modifier: 0.9736199417704
- **Title: No Limits - March 24, 2011**  
Score: 3944.3287604217  
Title %: 22.857142857143  
Content %: 7.0707070707071  
Time Modifier: 0.9736199417704
- **Title: No Limits - March 3, 2011**  
Score: 3934.9039970054  
Title %: 23.529411764706  
Content %: 3.4934497816594  
Time Modifier: 0.9736199417704
- **Title: No Limits - IUPUI Symposium - January 24, 2013**  
Score: 3927.7357653487  
Title %: 21.818181818182  
Content %: 11.267605633803  
Time Modifier: 0.9736199417704
- **Title: No Limits - March 10, 2011**  
Score: 3898.8676533634  
Title %: 22.857142857143  
Content %: 5.6818181818182

## Chapter 6: Project Organization and Management

### Team Organizational Structure:

Max was our team leader as well as the communication lead. He also helped to organize meetings, and helped wherever he was needed. Tom was the coding lead, as he did most of the work in our PHP file. He also functioned as a testing lead, as he would usually be the one to test the strings that the rest of the team would provide to him. Adam helped to come up with the scoring for the algorithm, and worked on documentation and diagrams. He also functioned as the back-up team lead, in the few occasions where Max was unable to make a meeting or presentation to Professor Linos. Phil helped with documentation. He also functioned as the assistant technical lead.

### Management Process:

We would have weekly meetings on Sundays. During these meetings would work on and later refine our code and algorithm. The majority of our intragroup communication was done either face to face, or via text-message. However, our communication with our client Chris was done primarily through email, save for a few face to face meetings, as the communication and team lead Max handled most of the correspondence with Chris. We put our weekly status reports on our Google Drive, where both Chris and Professor Linos had access to them. After spring break, we would also send Chris a weekly progress update.

### Technologies and Tools Used:

We had to set up a virtual environment on our computers so that we would all have the same versions of PHP and MySQL, as well as the same Apache server. Our virtual environment was set up through virtual box, and within this virtual box we ran vagrant.

### Weekly Status Reports:



## WEEKLY STATUS REPORT (WSR)

[2/16/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 2/10-2/17

**I. RED FLAGS:** None so far.

**II. ISSUES:** We still need to learn PHP and MySQL. I believe Tom has some experience with PHP, which will be beneficial.

**III. ACCOMPLISHMENTS (dates):**

2/10

- Created Team America and Tom.
- Assigned Project with WFYI.
- Phil created wiki.

2/11

- Scheduled our first meeting with Chris on 2/19 at 4 PM.
- Sent out group text to reread project description and find guides on PHP.

2/16

- Max created a google drive for the group.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Read up on PHP tutorials and guides (Everyone) – On-going
- Come up with questions for first interview with Chris by Wednesday (Everyone)
- Interview with Chris (Everyone) – 2/19 @ 4 PM

## WEEKLY STATUS REPORT (WSR)

[2/23/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 2/17-2/24

**I. RED FLAGS:** None so far.

**II. ISSUES:** We still need to learn PHP and MySQL. I believe Tom has some experience with PHP, which will be beneficial. We need to learn PHP quickly to stay on task with schedule because after spring break, we should start developing our search algorithm.

**III. ACCOMPLISHMENTS (dates):**

2/19

- Had our first meeting with Chris.
- Planned out the semester with basic goals.
- Added Chris to the google drive, so he can see progress.
- Talked to Sorenson and Gupta for PHP books. Got one book with about thirty pages on PHP.

2/20

- Chris provided us on the google drive folder with more information and skeletons of the code that we will work on.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Read up on PHP tutorials and guides (Everyone) – On-going
- Figure out what we will use to test our code (Everyone) – On-going
- Need to find a time for the group that we can all sit down and work on PHP. (Everyone) – On-going

## WEEKLY STATUS REPORT (WSR)

[3/3/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 2/25-3/3

**I. RED FLAGS:** None so far.

**II. ISSUES:** We still need to learn PHP and MySQL. I believe Tom has some experience with PHP, which will be beneficial. We need to learn PHP quickly to stay on task with schedule because after spring break, we should start developing our search algorithm.

**III. ACCOMPLISHMENTS (dates):**

2/26

- Had a group meeting. Tom set up a virtual machine environment to test our code.

2/27

- Updated our wiki page on Moodle to include more of our documents on our google drive folder.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Read up on PHP tutorials and guides (Everyone) – On-going
- Need to find a time for the group to develop a pseudocode for the search engine (Everyone) – Monday or Wednesday
- Need to find a time for the group that we can all sit down and work on PHP. (Everyone) – On-going
- Need to schedule a meeting or at least an update email to Chris on how things are going (Max) – Sometime this week

## WEEKLY STATUS REPORT (WSR)

[3/17/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 3/10-3/17

**I. RED FLAGS:** None so far.

**II. ISSUES:** Scheduling team meetings is a little difficult with our busy schedules.

**III. ACCOMPLISHMENTS (dates):**

3/9-3/16

- Group ready to start coding the algorithm. Finished tutorials on PHP and MySQL.

3/5

- Developed some pseudocode for the algorithm and posted it to the google drive.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Meeting with group and try to set up times on when to code algorithm.  
(3/17)
- Come up with some questions this week when coding to ask Chris in our next meeting – Sometime this week
- Need to schedule a meeting or at least an update email to Chris on how things are going (Max) – Sometime this week

## WEEKLY STATUS REPORT (WSR)

[3/24/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 3/17-3/24

**I. RED FLAGS:** None so far.

**II. ISSUES:** Scheduling team meetings is a little difficult with our busy schedules.

**III. ACCOMPLISHMENTS (dates):**

3/19

- Had a group meeting to set up our virtual environment to test our code. Also, set up meetings to start code.

3/23

- Group started preliminary coding the ranking system. Came up with questions for Chris. Max emailed Chris. Phil and Adam worked on use case diagrams, use case scenarios, and sequence diagrams.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Waiting on Chris to answer questions. (Whenvever)
- After questions answered, finish preliminary coding. (Probably weekend)

## WEEKLY STATUS REPORT (WSR)

[3/31/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard

**FROM:** Max Brodbeck

**SUBJECT:** Status report for week 3/24-3/31

**I. RED FLAGS:** None so far.

**II. ISSUES:** None at this moment.

**III. ACCOMPLISHMENTS (dates):**

3/24

- Chris emailed back answers to our questions and scheduled meeting time for this upcoming week.

3/30

- Finished Preliminary code for our rank system and completed a state diagram.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Meeting with Chris on Wednesday at 4 PM to discuss further milestones and direction of project. (4/2)

## WEEKLY STATUS REPORT (WSR)

[4/6/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 3/31-4/6

**I. RED FLAGS:** None so far.

**II. ISSUES:** Getting errors in our PHP files trying to connect to our MySQL database.

**III. ACCOMPLISHMENTS (dates):**

3/24

- Had our meeting with Chris. We discussed about scheduling, direction of the project with the progress we have so far, and how has the project as a whole been going.

4/6

- Working on getting our test environment working to test our scoring algorithm and optimize our code. Having an error connecting to our MySQL Database in our main.php file.
- Emailed Chris about our error in our test environment.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Waiting on Chris to email back on issue. (Whenever)
- Continue research on our error (Whenever)
- Work on our test environment (Sometime during week or Sunday)

## WEEKLY STATUS REPORT (WSR)

[4/13/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard  
**FROM:** Max Brodbeck  
**SUBJECT:** Status report for week 4/6-4/13

**I. RED FLAGS:** None so far.

**II. ISSUES:** Having errors in our strings that are being searched. Doesn't like the having spaces in between the words. Working on solutions now.

**III. ACCOMPLISHMENTS (dates):**

4/7

- Chris helped us connecting our php file to database.

4/13

- Our code is working now and sorting our queries with our scoring system. Our problems now exist in our search strings and it is not liking having spaces in between words. We are working on solutions right now. Emailed Chris on update and new error.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Keep improving our code and optimizing our system. (Most likely Sunday)



## WEEKLY STATUS REPORT (WSR)

[4/20/14]

---

**TO:** Phil Dwyer, Adam Sanders, Thomas Pritchard

**FROM:** Max Brodbeck

**SUBJECT:** Status report for week 4/13-4/20

**I. RED FLAGS:** None so far.

**II. ISSUES:** Our scoring system is a little off, but we are working on solutions as we speak.

**III. ACCOMPLISHMENTS (dates):**

4/16

- We tried to tackle our issue with multiple word strings with our heavily based PHP style. We decided that we should look into a different direction and after a few hours of research, we are taking a heavy SQL style now.

4/20

- We made huge progress on our new SQL style code. We are now having some issues with our scoring system that we are currently working on.

**IV. ACTION ITEMS FOR FOLLOWING WEEK (dates):**

- Fix scoring system and optimize code. (Wednesday or Sunday)
- Prepare for our final presentation next Monday (Wednesday or Sunday)

## Appendix

### Source Code:

```
<?php
class wfyi_site_search {

    private $db;    // the PDO object used to query the search table

    /*
    * __construct(PDO $_db)
    * $_db is a initialized PDO object conencted to the site database
    */
    public function __construct(PDO $_db) {
        $this->db = $_db;
    }

    /*
    * getSearchResults($query)
    * return an array of results based on $_qString, or false if no results
    * results are sorted by their weight score
    * $query is a user supplied string
    */
    public function getSearchResults($query) {
        $sql = 'select * from search';
        $connection = mysql_connect('localhost','user','password');
        if(!$connection) {
            die('Database connection failed: ' . mysql_error());
        } else {
            $db_select = mysql_select_db('wfyisearch', $connection);
            if (!$db_select) {
                die('Database selection failed: ' . mysql_error());
            }
        }
        // We preform a bit of filtering
        $query = strtoupper($query);
        $query = strip_tags($query);
        $query = trim($query);

        $sql = $this->buildSQL($query);
        $data = mysql_query($sql);

        $index = 0;
        while($result = mysql_fetch_array( $data ))
        {
            similar_text($query, $result['Title'], $title_perc);
            similar_text($query, $result['MetaDescription'], $meta_perc);
            similar_text($query, $result['Content'], $content_perc);
            $strength = 1;
            $strength = $strength + ($title_perc * 100);
            $strength = $strength + ($meta_perc * 10);
            $strength = $strength + ($content_perc * 10);

            // Title
            // Meta-
            // Content
        }
    }

    public function buildSQL($query) {
        $sql = "select * from search where (title like '%$query%' or meta_description like '%$query%' or content like '%$query%') order by weight desc";
    }
}
```

Description

```

        $strength = $strength + ($content_perc * 10); // Content
        if ($result['TableName'] == 'wfyi_programs') {
            $strength = $strength + ($strength / 1.5);
        } elseif ($result['TableName'] == 'wfyi_articles') {
            $strength = $strength + ($strength / 4);
        } else {
            $weeks_since_creation = (time() - $result['Date']) / 604800;
            $time_multiplier = pow(0.95, $weeks_since_creation);
            $strength = $strength * $time_multiplier;
        }

        if ($title_perc > 20) {
            if ($result['TableName'] == 'wfyi_programs_television_episodes' ||
$result['TableName'] == 'wfyi_programs_radio_episodes') {
                $strength = $strength + ($strength / 1.5);
            }
        }

        $results[$index] = array(
            'Title' => $result['Title'],

            'Score' => $strength,

            'Title%' => $title_perc,

            'Meta%' => $meta_perc,

            'Content%' => $content_perc,

            'TimeMod' => $time_multiplier);
        $index = $index + 1;
    }
    // Organise output
    $this->arraySortByColumn($results, 'Score');
    return $results;
}

/*
 * buildSQL($query)
 */
private function buildSQL($query) {
    $strip_words = array("and", "or", "the");
    foreach($strip_words as $stripped) {
        $query = preg_replace("#\b" . $stripped . "\b#is", "", $query);
    }
    $query = preg_replace("#" . $stripped . "\s#is", "", $query);
    $query = preg_replace("#\s" . $stripped . "#is", "", $query);
}

    $words = explode(" ", $query);
    $words_count = count($words);
    $index = 0;

    $sql = "select * from search where ";

```

```

        foreach (array('Title', 'Heading', 'MetaDescription', 'Content') as $column) {
            foreach ($words as $word) {
                $sql .= "upper($column) like '%$word%' ";
                if ($index !== $words_count) {
                    $sql .= "or ";
                }
            }
        }
        $sql = substr($sql, 0, -3);
        $sql .= ' ';
        echo '<b>SQL used:</b><br><br><code>' . $sql . '</code><br><br>';
        return $sql;
    }

    /*
    * arraySortByColumn(&$arr, $col, $dir = SORT_ASC)
    * sorts multi-dimensional arrays by a column in the sub-array
    * arr is passed by reference so it just updates it
    */
    private function arraySortByColumn(&$arr, $col, $dir = SORT_DESC) {
        $sort_col = array();
        foreach ($arr as $key=> $row) {
            $sort_col[$key] = $row[$col];
        }

        array_multisort($sort_col, $dir, $arr);
    }
}
?>

```

### Customer Evaluation Form:

## CLIENT QUESTIONNAIRE EPICS@BUTLER PROJECT EVALUATION

NAME OF CLIENT AGENCY:\_\_\_\_\_

EVALUATOR'S NAME/TITLE: \_\_\_\_\_

SEMESTER/YEAR WORKING WITH TEAM:\_\_\_\_\_

Please rate the degree to which you are satisfied with the following:

Very Satisfied		Neutral		Very Dissatisfied
1	2	3	4	5
a. Communication with student team				
1	2	3	4	5
b. Responsiveness of team to customer's needs and interests				
1	2	3	4	5

c. Professionalism of the team

1                      2                      3                      4                      5

d. Amount of time team devoted to this project

1                      2                      3                      4                      5

e. Skill level of team

1                      2                      3                      4                      5

f. Quality of the work

1                      2                      3                      4                      5

g. Degree of work completion

1                      2                      3                      4                      5

h. Overall experience with Butler students

1                      2                      3                      4                      5

Also, please provide some feedback about your work with the EPICS software engineering team. You may respond to the following questions by writing your answers in the spaces below.

Thank you.

1. Did working with the group enable you to improve your business process? If so, how?
2. What 1-3 things could be done to improve the team and their project?
3. What would you like the students to know?
3. What would you like the faculty advisor to know?

#### Peer Evaluations:

##### Progress Report for Team America and Tom (WFYI Project)

Max – Team Lead

- Scheduled Team Meetings and time to work on rank system
- Handles all communication between client and schedules meetings with client
- Weekly Status Reports
- Learned PHP and MySQL
- Contributed to preliminary code with outlining requirements and pseudocode
- Reviewed and questioned preliminary code

Phil – Grade A

- Completed inception stage diagrams
- Documented pseudocode
- Learned PHP and MySQL
- Has been very flexible scheduling meetings

- Helped with testing of preliminary code and setting up virtual Environments on laptops

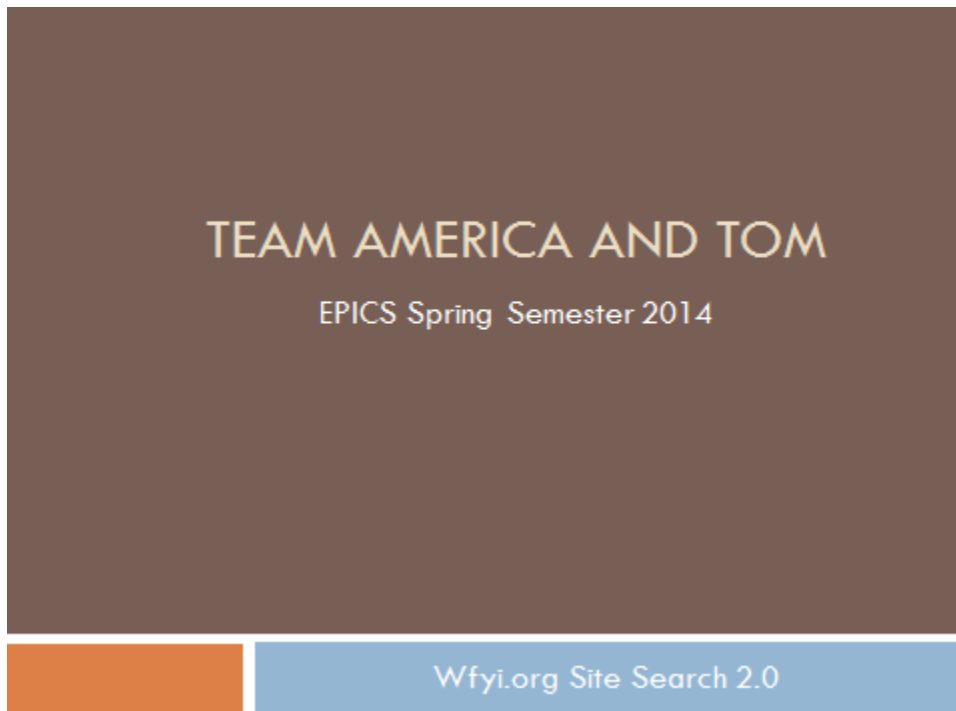
Adam – Grade A

- Completed inception stage diagrams
- Has been very flexible scheduling meetings
- Learned PHP and MySQL
- Contributed to the rank system in mathematical standards regarding the scoring system for each result
- Contributed to preliminary code

Tom – Grade A

- The mastermind behind setting up the virtual environment on our laptops using Vagrant
- Main coder for the preliminary code
- Learned PHP and MySQL
- Overall, Tom is the technical lead in this project.

Presentation Slides:



## Introduction

---

- **Prior Experiences**
  - No PHP or MySQL Experience
- **Role Explanation**
  - Max – Team Lead
  - Tom – Technical Lead
  - Phil – Algorithm/Documentation
  - Adam – Algorithm/Documentation (Vice Team Lead)

## Milestones

---

- **Stage 1: Learning PHP and MySQL**
  - How did we teach ourselves?
- **Stage 2: Setting up VM**
  - Tom's Idea
- **Stage 3: Coding**
  - Difficulties/Errors
  - Solutions/Alternatives
  - Testing/PHP Log



## Conclusion