UNIVERSITY OF WINDSOR

ELECTRICAL ENGINEERING CO-OP

WEB APPLICATION DEVELOPMENT BEST PRACTICES

ESSEX ENERGY CORPORATION

SOFTWARE DEVELOPMENT

WINDSOR, ON

Submitted to: Mr. Robert Galka

Submitted by: Eric Parker

Submitted on: September 12, 2016

University of Windsor
Faculty of Electrical and Computer Engineering
Windsor, ON N9B 3P4

September 12, 2016

Mr. Robert Galka
Technical Services Supervisor
2199 Blackacre Dr.
Oldcastle, ON N0R 1L0

Dear Mr. Galka,

Please accept this report entitled “Web Application Development Best Practices” as my submission to fulfil my work term requirements.

It was an honour to complete my second work term at Essex Energy Corporation in the software development department. It was my pleasure to have the opportunity to participate in the company’s growth and development. During my time at Essex Energy, I was fortunate enough to develop a very strong understanding of both web application development techniques and power distribution systems. I gained valuable work experience through contributing to a number of different projects on both the software development and engineering teams. Such projects awarded me the opportunity to work with several very intelligent and highly skilled people from which I learned a great deal.

This work term has provided me with valuable work experience, a deeper passion for computer programming, and a better understanding of electrical generation/distribution systems. My submitted report outlines the importance of producing and maintaining quality web applications that can connect to database servers to enable businesses to grow through constant tracking and maintenance of business data.

Finally, I would like to thank you, my manager, for sharing your experience and knowledge with me throughout my work term. I enjoyed learning from you and I am very grateful that you were willing to share information with me and answer my questions.

Sincerely,

Eric Parker

TABLE OF CONTENTS

LIST OF FIGURES

Page

Executive Summary:

The purpose of this report is to exemplify one way businesses can easily implement data driven web applications for internal or customer use. Such web development techniques are currently being used by Essex Energy Corporation to manage vast amounts of outage management data as well as data collected from power distribution asset inspections (among others) and make such information available over the internet through the creation and use of Windows Communication Foundation (WCF) web services. Regardless of which business sector a business is in, vast amounts of data are collected each year by almost every single company in the world. Often for legal or customer service purposes, this information must be stored and maintained thus creating a need for software development and IT departments within companies. A very popular way to store company information is through the use of relational databases which store data in a number of tables that may be linked by one or more fields. These databases can be made available online through the use of representational state transfer (REST) web services and consumed in web browsers or mobile applications for simple and intuitive database interaction. This method of storing and serving data is deployed by thousands of companies worldwide and will be discussed in detail in this report.

This report also provides readers with a brief description about some of the necessary tasks that need to be completed by power distribution utilities. Since Essex Energy is associated with Essex Powerlines (a regulated local distribution company), the two companies work together to solve problems. Some of the necessary tasks that need to be completed are outlined in this report, as well as an introduction on how to automate said repetitive corporate tasks (using a programming language called Python) for the sake of efficiency.

Introduction:

This report outlines the web application development techniques and database management procedures used at Essex Energy Corporation to manage, maintain, and serve large amounts of dynamic asset inspection data. In general, the storage of business data is imperative to the success of any business regardless of which business sector the company operates in. As the world becomes increasingly driven by data, companies need to be able to make business information available for consumption when needed. This is why many successful companies employ web developers and IT personnel to create and maintain web applications that make dynamic company information available. This allows other employees or customers who lack programming expertise to view such information simply by navigating through the company website. At Essex Energy, one of the main software development projects required the software development team to create a web application that used stored database procedures to pull data from the company's "PMDev" database and make this information available on a website; done so in an aesthetically pleasing and intuitive way so that electrical engineers could read from and add to the database simply by filling out information on a website and clicking on buttons.

Part of the duties of a regulated distribution company is to inspect power distribution assets (such as transformers, switching cubicles, utility poles, etc.) regularly to ensure they are operating correctly and safely as required by the government of Canada. Thus for legal reasons, it is important that the results of each inspection are stored. At Essex Energy, all inspection data is stored in the "PMDev" database and thousands of inspections are completed annually. The task faced by the software development team was to create a web application capable of connecting to and modifying information contained within this database to meet the needs of the electrical engineers performing asset inspections. A complete investigation of the software development team's solution to this problem is presented within this report, including web application development best practices, analysis of some of the problems encountered during the project's development, and software management techniques learned throughout the project's development that should be implemented in all future software development projects to eliminate bugs from and improve the quality of the code being produced.

As the work term progressed, demands within the company shifted focus from developing web applications to maintaining and testing existing software applications. As such, this report also provides readers with a brief explanation of the software testing techniques used at Essex Energy, as well as a description of software testing best practices that should employed on future projects at Essex Energy.

Also included in this report is a summary of some of the tasks regularly completed by power engineers at Essex Energy. Since some of these tasks were very tedious and time consuming, computer programs for process automation were developed in an effort to improve efficiency. The results of such efforts are also outlined in this report, including an explanation on how to automate computer tasks and what types of tasks are ideal for automation.

Overview

Essex Energy is a division of Essex Power Corporation that consists of approximately 30 employees. Essex Power Corporation is the parent company to three local corporations who perform services related to mass power distribution. The three divisions of Essex Power Corporation are: Essex Powerlines Corporation, Essex Power Services Corporation, and Essex Energy Corporation. Essex Powerlines Corporation is a regulated power distribution company responsible for providing power to most commercial and residential electrical loads in Amherstburg, Lasalle, Tecumseh, and Leamington. The company performs regular maintenance on all power distribution assets (including transmission lines, transformers, utility poles, etc.) owned by the company, services such assets in the event of an outage to restore power to customers as quickly as possible, and tracks and maintains account and billing information for customers. Essex Power Services Corporation is a regulated service company that provides maintenance, construction, and other third party services to customers, particularly for municipal shareholders working with very high voltages such as feeder station setup and maintenance. Essex Energy Corporation is the only unregulated division of Essex Power Corporation, and it serves mainly as a consulting engineering firm providing expertise in the areas of energy management and conservation as well as distributed generation. Essex Energy also has a software development division who is responsible for producing engineering analysis software for use by engineers and executives within the company to track asset information and perform engineering analysis based on real time data.

The role of a software developer within a small startup company is dynamic; responsibilities and priorities can change daily depending on the demands present within the company. A software developer at a small company is responsible for everything code related from bug fixes to User Interface (UI) design. Smaller companies also demand more from web developers because they may have to design everything both client-side including the UI and the controllers handling the information entered by users, and server-side including creating procedures and business logic that interact directly with the database. These types of developers are comfortable working with both back-end (server-side) technologies including databases and database servers as well as server-side languages including C#, PHP, or Python, and front-end (client-side) technologies including HTML, CSS, JavaScript, AngularJS, etc.

It should be noted that most communication over the internet is done in the form of Hypertext Transfer Protocol (HTTP) requests. HTTP is the foundation of data communication for the World Wide Web, and it is how websites and files can be shared over the internet. All computers contain a unique Internet Protocol (IP) address that serves as an address so web servers know where to send information when such information is requested. Computers that connect to the internet can be sorted into two categories: web servers and web clients. Web servers are computers connected to the internet that run special server software that enables them to share files over the internet with other computers. Every domain name (such as *google.com*) points to the IP address of the computer that acts as the server for that domain. Contrarily, web clients are computers connected to the internet via a web browser (such as *Google Chrome* or *Internet Explorer*) who request files from web servers by visiting websites. In doing so, web clients send their IP address to the web server and the web server responds by “mailing” the files that make up the website to the address of the client. These notions make up the fundamental concepts ultimately allowing for file communication over the internet.

I. Understanding Big Data Problems

As the world continues to become increasingly driven by data, it puts intense pressures on businesses to combine robust data tracking and data storage techniques with a strong internet and mobile presence. Since any given website or mobile device may have millions of daily consumers, greater demands are placed on software developers than ever before. With this in mind, software development is always managed with scalability in mind. Scalability is a very important term in the field of software development, and can be defined (generally speaking) as the ability of a company to maintain or even increase its level of performance or efficiency when experiencing increased consumer demand. In the field of web and mobile development, this can mean higher website traffic, more mobile application downloads, or an increase in database requests.

Since the demand for a website or mobile application can never be known in advance, software development teams must incorporate development techniques that ensure they have the capability to scale upwards given the opportunity. From a hardware perspective, this means that server computers must be physically equipped with the means to satisfy high demands, however this can be accomplished simply by buying and installing better hardware on the server machines. The real challenge is faced by the software development team; the original source code for a web application or mobile application must be designed efficiently so that the code demands as little hardware resources as possible, thus allowing for thousands of requests to be made at the same time. To put this concept into perspective, consider a hypothetical inefficient algorithm implemented by the company Twitter. Due to this inefficient algorithm, an extra 40 Bytes of memory is required for each tweet that is posted. This value may seem extremely insignificant, but consider that about 250 million tweets are posted on Twitter each day. This means Twitter will need to store about 10 GB of data extra each day, or 3.65 TB annually just due to the minute inefficiency present within the tweet algorithm. With this example in mind, it is easy to see how scalability is a big issue that needs to be addressed when new projects in computer science are taken on.

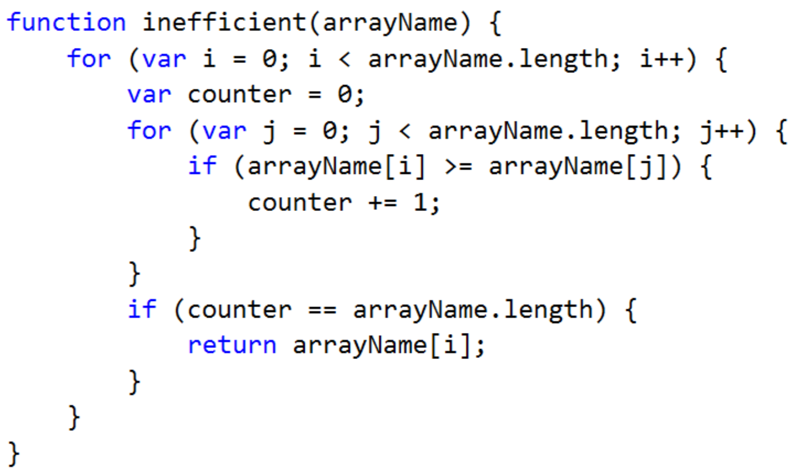
It is important to note that scalability is only one of many issues faced by web developers; not only do algorithms need to be as efficient as possible, but there should be as few functions as necessary to get the required task completed, and the required functions need to operate without bugs in order to complete the desired task. In other words, if you have already written a function to perform a specific task in one file, it is considered very bad practice to reuse (i.e. copy and paste) that function into another file in the same project.

Yet another code management issue plays a very important role in producing quality computer programs: bug fixes. It is not uncommon for programmers to be working on a task and then become frustrated because they cannot solve a particular problem as quickly as they would like to, thus decide to move on to their next task leaving behind a faulty code that still needs to be fixed at another time. In theory, this shouldn't be an issue. The programmer should be able to take a break and return to the code at a later time to work through the problem. However, in practice, this procedure can be detrimental to the productivity of the software development team; since programmers often work on many bug fixes and develop new code for a number of different files, developers may forget where they left unfixed bugs, and (if somehow overlooked during software testing) it may not be found until a customer calls in asking why a specific feature isn't working properly.

II. Software Development Best Practices

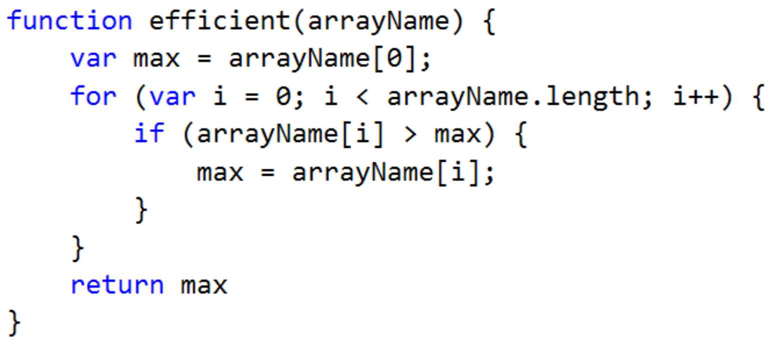
In order to address the issues presented in the *Understanding Big Data Problems* section of this report, software developers follow a series of software development techniques known as “best practices” in order to ensure they are producing quality code. It should be noted that this report covers only a very small portion of the set of all software development best practices, and it does so in a very general manner attempting to cover only a few of the most fundamental concepts in software development. It should also be noted that different programming languages have different best practices due to the way each language is structured differently. However, this report covers very general best practices that should be adopted by all programmers regardless of what language they have chosen to work with. Specific examples of these best practices will be implemented using JavaScript and AngularJS since that was the programming language used by Essex Energy to develop its newest web applications.

The first problem discussed in the previous section of this report described the importance of creating efficient algorithms to enhance the scalability of web applications. Consider the following examples shown in Figure 1 and Figure 2. Each snippet of code defines a JavaScript function that accepts an array as an argument and is designed to return the maximum value in that array.



```
function inefficient(arrayName) {
    for (var i = 0; i < arrayName.length; i++) {
        var counter = 0;
        for (var j = 0; j < arrayName.length; j++) {
            if (arrayName[i] >= arrayName[j]) {
                counter += 1;
            }
        }
        if (counter == arrayName.length) {
            return arrayName[i];
        }
    }
}
```

Figure 1: An Inefficient Method to Find the Largest Value in an Array



```
function efficient(arrayName) {
    var max = arrayName[0];
    for (var i = 0; i < arrayName.length; i++) {
        if (arrayName[i] > max) {
            max = arrayName[i];
        }
    }
    return max
}
```

Figure 2: An Efficient Method to Find the Largest Value in an Array

Consider the functions written in both Figure 1 and Figure 2. Both methods achieve the same result, and both do so very quickly; unnoticeably different when the functions are passed arrays with very small number of items. However, further analysis of each individual function indicates one clear winner. In Figure 1, the use of two nested “for” loops are used to compare each entry to every single other entry in the array. In order for a value to be the maximum value in the array, it must be greater than or equal to every other value in the array, meaning the “counter” variable will be equal to the length of the array. In the worst case scenario, the maximum value is in the last index of the array, meaning for an array of length N, this method will iterate N times for each index, resulting in a total of N^2 iterations.

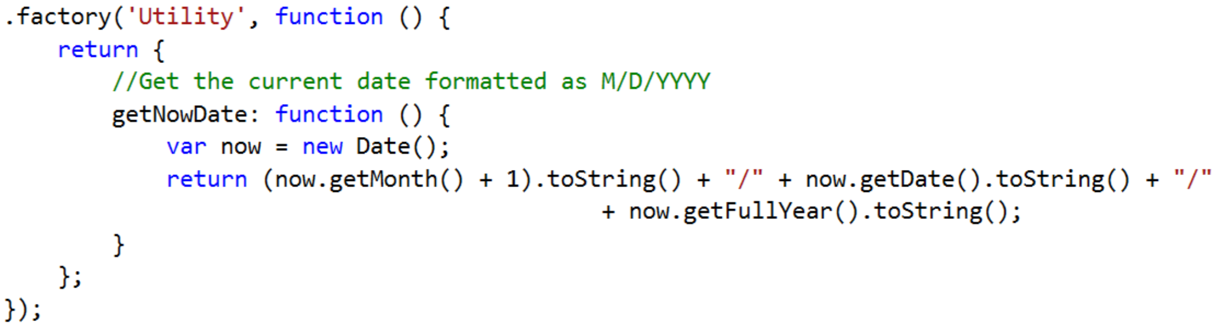
Contrarily, the function presented in Figure 2 sets a variable named “max” to the value of the first item in the array. It then loops through the rest of the array, comparing the current maximum value to the next value in the array. If the next value in the array is larger than the current maximum value, the maximum value is redefined to equal the larger value. For an array of length N, the loop iterates a maximum of N times.

Although the performance of each function is almost identical for very small arrays, as the length of the array passed to each function approaches infinite, the function shown in Figure 2 will far outperform the function presented in Figure 1. This is just a very simple example of how certain algorithms for the same function can outperform others significantly as the amount of data that the function has to work with increases. As a web developer, this concept is particularly important considering that most of the data used within web applications is pulled from databases that can contain immense quantities of data.

Another software development best practice that needs to be discussed is how to address the issues created when developers reuse (i.e. copy and paste) code. As discussed in the previous section, reusing code is considered a very bad practice. This can be for a number of reasons; not only does the extra code physically take up more memory (which increases the time it takes to serve such files over the internet and decreases application scalability), but once the code is copied, each occurrence of the code will need to be tested and maintained separately, meaning that each instance of the code must be remembered, modified, and tested if required.

Economically, it multiplies the cost required to produce that code by the number of times someone pastes it. This estimation excludes the fact that it is very likely that the programmer who pasted the code may have forgotten how many times (and in how many places) he or she pasted it, so if there is a bug or specification change, they may forget to implement the fix everywhere only to find out later that they need to fix it again. In general, code reuse should be avoided wherever possible.

The solution to this problem depends very much on the situation and the programming language being used. For example, in object oriented programming, if one class of objects contains methods that the programmer wishes to use with another class of objects, inheritance can be used. However, this too has its limitations if the programmer wishes to inherit just a single method. Thus, sometimes the best solution is specific to a particular programming language. For example, one way to solve this problem using AngularJS is through the use of Angular services. Angular services allow the programmer to write one function and make this function available anywhere it is needed, thus preventing the need to reuse (copy and paste) code from one file into another. Consider the following example shown in Figure 3 below.



```
.factory('Utility', function () {
    return {
        //Get the current date formatted as M/D/YYYY
        getNowDate: function () {
            var now = new Date();
            return (now.getMonth() + 1).toString() + "/" + now.getDate().toString() + "/"
                                                + now.getFullYear().toString();
        }
    };
});
```

Figure 3: An Example of an Angular Service that Returns the Current Date

The code shown in Figure 3 shows the creation of an Angular service called "Utility". AngularJS uses a software design pattern called *dependency injection*. Using the `.factory(…)` Angular method creates a dependency object that can be injected into and used in the controller for each web page as required simply by calling the method like so: `Utility.getNowDate()`. Using this technique, any number of custom functions can be written and injected anywhere into the project without ever having to copy and paste code. Now, if changes need to be made to the method, the changes only need to be made and tested once. This type of dependency injection is an extremely powerful feature of AngularJS and should be used in almost every possible circumstance to discourage developers from copying and pasting code.

Lastly, the final web development best practices that will be discussed in this report involve dealing with bugs and bug fixes. No matter how great an idea for a computer program is, so much of the success of software is dependent on how few problems there are with the program and how efficiently the development team can implement the code; both of which are the result of lack of bugs within a program. A software bug can be defined as an error or fault in a computer program that causes it to behave in unintended ways or produce an incorrect of unexpected result. Even if the code's syntax is correct, runtime errors can easily be created when programmers fail to overlook minor details imperative to the function's success. For programmers, bugs are a part of daily life; it is very uncommon for a programmer to produce perfect code the first time creating a function. However, it is how bugs are dealt with that is the focus of these next best practices.

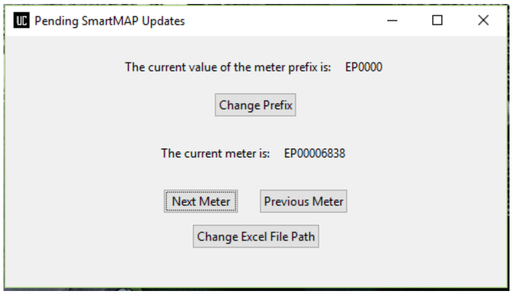
The first bug-related best practice is to keep and maintain a bug database; that is, an organized list of all known bugs in the code. Many programmers think they can keep a bug list in their head, but in practice this rarely works out to anyone's advantage even on a team consisting of one person. If all known bugs are documented, including how to reproduce the bug and the expected behaviour versus the observed (incorrect) behaviour, any programmer working on the project can find, understand, and reproduce the bug and then begin fixing the issue accordingly. Economically, this makes sense as well; a bug database prevents known bugs from being forgotten and found again. This not only costs more to pay the programmer or tester each time they find the issue again, but the later the problem is dealt with, the less information is known about the issue thus requiring more time to fix the issue. This concept leads to the final and even more important software development best practice.

In all fields of software development, all bugs should be completely eliminated before writing new code. In the early days of commercial software development, companies were insistent on keeping on schedule rather than ensuring that the code being produced was free of bugs. They preferred meeting deadlines and dealing with bug fixes later rather than ensuring the code currently being produced was of good quality. This method ultimately proved useless as many large software providers experienced years of delays beyond the expected launch dates because of these bug fixes that were being completed after the fact. This is why large corporations including Microsoft adopted what is known as a "zero defects methodology" which means at any given time, the highest priority is to eliminate bugs before writing any new code. In general, the longer a developer waits before fixing a bug, the costlier it is to fix (in time and money). This concept is true even for trivial errors because if code is tested by a developer just moments after they've completed writing it, the code is still fresh in their mind and they will often be able to locate and fix the problem with ease. Contrarily, returning to the same problem after a few days or even months of working on another project will require them to find the error again even though they've probably forgotten much about the code. From a manager's perspective, another advantage to fixing bugs before writing new code is that it is much easier to predict how long it will take to write new code than it is to predict how long a bug fix will take. This enables managers to give executives more accurate timelines as to when a feature might be ready, and enables the managers to ship bug free code whenever necessary.

III. Process Automation Using Python

Besides deploying quality code, another indispensable skill that programmers should acquire is the ability to write scripts capable of automating tedious but necessary computer tasks. At some point or another, almost every employee within any corporate company will be faced with a very tedious task that needs to be completed. Such tasks may include responsibilities such as sorting through large Microsoft Excel files or renaming files on a computer to fit a certain format; very repetitive and time consuming tasks that need to be completed for one reason or another. Such tasks are perfect candidates for computer automation because of the speed at which computers operate.

At any regulated power distribution company, there is no shortage of such tasks. Since Essex Energy often worked in collaboration with Essex Powerlines Corporation, the software development department at Essex Energy was often asked to attempt to automate said tasks. A computer programming language that contains many modules created specifically for process automation is Python. Used in conjunction with the book "Automate the Boring Stuff with Python", the software development department was able to automate many processes for the power engineers including programs that manipulate Microsoft Excel files and PDFs, send emails, rename folders and files, and even leverage the computer's clipboard (Figure 4).



Pending SmartMAP Updates

The current value of the meter prefix is: EP0000

The current meter is: EP00006838

Figure 4: An Example Graphical User Interface (GUI) Python Automation Program

Conclusion:

In conclusion, the storage of business data is imperative to the success of any business regardless of which business sector the company operates in. Companies need to be able to make business information available online for consumption when necessary thus demonstrating just how important the duties taken on by a company's software development and IT departments are. Since the success of any business is dependent on data in one way or another, it is important that software developers implement web development best practices while programming to ensure that quality code is consistently produced on schedule. Perhaps the most critical development practices that should be deployed by every developer include implimenting the most efficient algorithms possible for any given task, avoiding reuse (copying and pasting) of code at all times, and eliminating all bugs before writing new code as well as tracking all known bugs in an organized bug database.

Although many of these practices are already in place at Essex Energy, it is recommended that all of the web development best practices discussed in this report be strictly enforced for the sake of producing quality code. Particularly, a "zero defects methodology" (where the highest priority is to eliminate bugs before writing any new code) should be adopted by all developers and managers, and a bug database should be created and maintained for all software development projects (especially for the sake of co-op students). It is also recommended that programmers attempt to automate all tedious tasks faced by employees of Essex Energy for the sake of efficiency.

REFERENCES

"Automate the Boring Stuff with Python." *Automatetheboringstuff.com*. Last modified June 2015.

https://automatetheboringstuff.com/

"Dependency injection." *Wikipedia.com*. Last modified September 2016.

https://en.wikipedia.org/wiki/Dependency_injection

"Don't Copy Code. Oh, and Inheritance and Composition are Bad, Too." *Propelorm.org*. Last modified March 2011.

http://propelorm.org/blog/2011/03/03/don-t-copy-code-oh-and-inheritance-and-composition-are-bad-too.html

"How do you show that one algorithm is more efficient than another algorithm?" *stackoverflow.com*. Last modified January 2010.

http://stackoverflow.com/questions/2028359/how-do-you-show-that-one-algorithm-is-more-efficient-than-another-algorithm

"How Twitter Stores 250 Million Tweets A Day Using MySQL." *Highscalability.com*. Last modified December 2011.

http://highscalability.com/blog/2011/12/19/how-twitter-stores-250-million-tweets-a-day-using-mysql.html

"The Joel Test: 12 Steps to Better Code." *Joelonsoftware.com*. Last modified August, 2000.

http://www.joelonsoftware.com/articles/fog0000000043.html