# Work Term Report 3
# COMP 3999A

**FOR THE SUMMER 2014 PLACEMENT AT THE OFFICE OF THE AUDITOR GENERAL OF CANADA**

**Ankush Varshneya (100853074)** AnkushVarshneya@cmail.carleton.ca

Supervisor: Alain Roy (Director of IT Applications)

Ankush Varshneya
415 Cresthaven Drive
Nepean ON,
K2G 6Z4

Saturday, August 16, 2014

Dr. Irwin Reichstein
School Of Computer Science
Carleton University
Ottawa, Ontario
KIS 5B6

**Work Term Report 3 COMP 3999A**

Dear Sir,

This is my third work term report and was prepared in connection with course COMP 3999A for my Co-op placement at the Office of the Auditor General of Canada (OAG). I worked as a Developer in the IT Applications team lead by Alain Roy. Most of my term was spent on the Product Code Management (PCM) Project lead by project manager Erin Larsen.

The Office of the Auditor General of Canada (OAG) is an independent audit office and a world leader in legislative and environmental auditing. They conduct independent audits and studies that provide objective information, advice, and assurance to Parliament, territorial legislatures, boards of crown corporations, government, and Canadians. The IT Application provides various software tools which helps the office to carry out all necessary tasks.

This report highlights some of the key concepts I learned during my placement as well as my contributions to OAG's products. During this term I mainly worked on front-end related tasks. Primarily I was assigned to implement maintenance related screens for administrators to manage the Product Code Management (PCM) application. To complement my learning I was given various small task which in turn helped me learn a lot about the system, hence making it easier to work on my main goal.

This report has been prepared and written by me and has not received any previous academic credit at this or any other institution. I would like to thank Alain Roy (Director of IT Applications) for helping in reviewing this report.

Sincerely,
Ankush Varshneya
Student number 100853074

# 2 ABSTRACT (EXECUTIVE SUMMARY)

I was hired by the Office of the Auditor General of Canada as a Developer to help implement maintenance related screen for a future release. This report begins with an introduction to the Office of the Auditor General of Canada, and then goes on to document the key concepts I learned about while developing there.

During this term I was primarily assigned the responsibility of implementing maintenance related screens for administrators to manage the PCM application. This process involved looking at the system requirements and coming up with conceptual User Interface designs (UI mockups) tailored to the requirements, as well as various reviews with the targeted users and other parties.

In addition to implementing features related to the PCM application, this document also includes details about some of the other tasks I was assigned. Initially, I was asked to link some modules to an application called OAG Campus. Besides developing the maintenance related screen, I also worked on importing some data to the database which helped me learn about the system from a back-end prospective and hence complemented with my primary task.

This report also highlights my work experience, day to day activities, some of the tools I used, how I applied my academic knowledge in my position and further my knowledge by getting work experience, different type of problems faced at the work, and the type of approaches I used to tackle those problems.

Lastly this document elaborates on how I related my work experience to my academic studies, my accomplishments, and any abbreviations used in my work term report.

# 3 ACKNOWLEDGMENTS

First and foremost, I would like to thank Bill Moeller, the software architect, for helping me understand my tasks and for helping me understand the underlying technologies used by the Office of the Auditor General of Canada and lastly for guiding me throughout the term.

I would like to thank Joe Tacconelli for helping me debug some of the problems I encountered during my development phase.

I would also like to thank Erin Larson for helping me throughout the term with my primary task and helping me understand a bit more about the system as well as some of the other things I needed to know about OAG.

I would also like to thank Eugenio Lo for pointing me at the right part of the database, when I needed to enter data for one of my tasks.

Lastly I would like to thank the business users, Alain Roy, Patrick Dumoulin, Erin Larson and other members of her team for helping me review and influencing my mock-ups and design idea as well as providing me with the needed requirements.

# 4 TABLE OF CONTENTS

# 5 INTRODUCTION

The Office of the Auditor general of Canada is an independent audit office with head office in Ottawa and four regional offices in Vancouver, Edmonton, Montréal and Halifax. The office is headed by the Auditor General of Canada, Michael Ferguson, who audits federal government departments and agencies, most Crown corporations, and many other federal organizations, and reports directly to Parliament. The Auditor General of Canada is also the auditor for the governments of Nunavut, the Yukon, and the Northwest Territories, and reports directly to their legislative assemblies.

Being a company focused on audits, the OAG is naturally keen on knowing as to how its own resources are being spent. To really be organized in this aspect, OAG uses an internal software called MIS2000 where employees use time codes as a means to show what activity and project they are working on. For example a time code may be in the format XXXXXX-YY where XXXXXX is a code for the project and YY is an activity. So If I were working on developing I would be using one code and if I were in meetings I would be using another code to time myself and so on. This approach really helps the company to better track its resources, as well it helps one to be able to see what they had spent time on and reflect accordingly.

In the IT application department, employees are usually assigned to a project for an application, for example I was assigned to the Product Cost Managements project. Each project consists of a project leader and other employees such as developers. Everyone in the project team reports to the project leader on progress related to the project. An employee is usually swapped in between projects as needed by project leaders. Every employee in the IT application department also reports to the director (DX), Alain Roy. All the DXs of various departments throughout the company report to the Principal (PX), the PX of the IT is Jean-Charles Parisé. The PXs report to an Assistant Auditor General (AAG), The AAG for IT is Lyn Sachs. All the AAGs report to the Auditor general of Canada, Michael Ferguson.

The IT application department makes and maintains applications for the entire office to help in various tasks such as audits. The Product Cost Management application is a web application, designed to replace components of an existing legacy terminal based application called MIS2000. This application is used to track projects and their budgets which are organized by product Lines, Product Categories, and Topics.

# 6 WORK EXPERIENCES

OAG follows agile project management methodologies, as a result many project teams consisting of cross-functional teams of the project leader, developers and key business users meet on a daily basis in stand-up SCRUM meetings. These meetings are arranged by the project manager, where the developers discuss the progress of the project as a whole with the business, as well as gather the users' feedback. Developers also discuss their progress with the project leader and may also use this time to get opinion and help from their fellow developers. Additionally, new feature implementations and improvements are also discussed here. These meetings are a great opportunity to find out what's going on around you, what your fellow colleagues are working on, it gives you a great understanding of the project and it's a great opportunity to follow up on any questions you may have. Since developers are sometimes switched around from project to project, these meetings are a great place to find out about what's going on in the project you may have just been assigned to.

Moreover, the project manager also arranges a meeting at the end of a sprint, every 2 weeks, where senior management, clients, developers and other interested parties meet to get updates on the project. This is less of a discussion group and more meant as an information session to get a broad level overview of the project and address any key concerns of the clients.

In addition to meetings organized by the project leader, a developer may also organize meetings with the key parties to discuss requirements and concerns related to the component of the project they may be working on. This is what I did at times to discuss mockups and gather feedback throughout the term so that I could provide my level best output to the project.

Lastly, a biweekly stand-up is scheduled by the director of IT application department, which is exclusive to all developers managed by that director. This is where developers from different projects discuss their progress and issues with the director and with each other. This is a key place to find out what is happening in other projects and the entire development team as a whole.

## 6.1 TOOLS

At the Office of the Auditor General of Canada, they always use the latest and greatest technologies which means that projects that originated at different time periods may use different technologies that were current to the time they were instantiated. The Product Cost Management application involves heavy use of .NET VB programming and the ASP.NET MVC 4 architecture. Its IDE consists of Microsoft Visual Studio and it uses Microsoft SQL Server as a back-end store, which is managed through the Microsoft SQL Server Management Studio. IIS also referred to as Internet Information Services, is used as a web server to deploy the application.

Microsoft Team Foundation Server (TFS) is another key component used by the application. It is used as a means of version control and as a product tracker. TFS is also used for compiling builds and deploying them.

From a front-end prospective HTML5 is used to display the UI and jQuery/JavaScript are used for client side scripting. Twitter bootstrap and CSS3 are used for styling the web application. In contrast to

previous work terms I found the front – end to be really flexible and easy to use, it was easy to add virtually any jQuery plug-ins to easily enhance the functionality. I utilized the DataTable plug-in throughout the maintenance related screens to address a lot of the requirements requested by the business users for the user interphase.

From a back-end prospective, the application uses a Service Oriented Architecture (SOA) and Windows Communication Foundation framework (WCF), which is used for building a service oriented application in a secure manner. .Net Domain object and .Net Services in combination with the entity framework are used to communicate with the back-end database.

.Net MVC (Model View Control) 4 is the architecture used to bind the front-end to the back-end. First, the entity framework is used to query the database and then the .Net Domain framework makes domain objects out of the data it is given. The .Net Services are used to send the domain objects to the front-end, through service call methods. The convention is that the Model part of "MVC" makes calls to the service and translates the domain objects to a Model used by the front-end. Using HTML Helpers, a tool included in .NET MVC, the View can bind the UI components to the properties of the Model; so if a UI component is used to get user input then that input will get sent to the model, which using the service calls, will call the back-end to save the user information. The controller part of MVC is used to route web pages to the correct view as well as to instantiate the model with the correct settings needed by the view. The URL convention used by the PCM application is as follows: www.NOTAREALWEBSITE.com/LANGUAGE/NAME_OF_CONTROLLER/NAME_OF_VIEW. The name of the controller is used to route to the correct controller, the name of the view is used to tell the controller that view it needs to route to and what model to execute. The language is used as an internationalization mechanism, it tells the controller what language to render the views in.

From talking to other students and fellow developers, other newer projects use a similar layout to the one mentioned above with the exception that some teams use .NET C# instead of .NET VB.

## 6.2 OBJECTIVES AND CONTRIBUTIONS

### 6.2.1 Maintenance Screens for administrative tasks

The main objective for me in this term was to add maintenance related screens to the Product cost management application. This gave me some exposure to .NET framework and various other Microsoft technologies. I also learned a bit about the formal development cycle present at OAG, and to see as to how they use agile methodologies to manage their projects.

#### *6.2.1.1 Requirement and Design Phase*

Initially I was given a set of draft requirements and some low fidelity prototypes to get me started. From there I arranged meetings with the key user group of the application as well as the project manager and other key members of the project team to try and gather a bit more requirements, solidify existing requirements, and gather use cases of the screens. Using this information I tried to make some medium fidelity prototypes via mockups using Balsamiq Mockups software.

Using this mockups I met up with the software architect, Bill Moeller, the Usability Architect, Patrick Dumoulin, the project leader, Erin Larson as well as the business users to further discuss and gain some valuable information about the project as well as to refine the vision for the maintenance screens. The Usability Architect helped me understand some of the common design methodologies used across OAG products and helped me understand how those should be applied to what I was working on. The software architect gave me an understanding of the tools available and how feasible my ideas were and also gave me an understanding of how I would need to interact with the back-end, in terms of service calls, to accomplish my mock-ups.

#### *6.2.1.2 Pre-Development Phase*

Before beginning development, I was originally given a walk down about the tools used by the project and how they work. In addition to talking with my fellow developers I also researched some of the material and even tried out some examples in tutorials to get an understanding of the tools and how they work together, as mentioned in the tools section above.

The PCM project uses other projects to function, which are called: OAG Service, BL, OAG.tt and DAL. OAG services include the SOA and WCF components and are used by .NET MVC to access the back-end, this is the top layer of the back-end. The BL is what contains the back-end business logic, this is where the logic used by the service call resides, and it gets and manipulates information from the DAL and passes it to the service accordingly. Using LINQ statements, one can even query the information in the DAL. The DAL, also called Data Access Layer, is a low level layer in the back-end, it is what communicates with the database and manipulates it directly; and is what makes the business logic "work". OAG.tt is what contains the Entity framework which is used to translate the information collected by the DAL and BL into Domain Objects which are part of .NET Domain, this is used by OAG Services to send information to .NET MVC. I also learned about how .NET MVC connects the back-end services to the front-end which is detailed in the tools section above.

#### *6.2.1.3 Development Phase*

After getting an understanding of the way the project works I proceeded to start developing the maintenance screens using the UI mock-ups and requirements. For each of the maintenance screens I followed a similar development pattern as listed in the next few paragraphs.

Primary, I analysed the mock up and requirements and made a list of what functionality will be needed. For example, most of the screens required a way to retrieve data, create data and amend pre-existing data. After the list was made, I asked the software architect whether the needed calls were available in the back-end or not; some screens had the needed back-end code while some had to be made by back-end developers. There were four types of maintenance screens I had worked on at the time of writing this report namely Product Lines screen, Product Categories screen, Topics screen, and Product Line rate screen.

Furthermore, then I started to piece together the MVC part of the screens. First, I developed the Model part; this is where I made service calls to the back-end to get the needed properties and implement the needed methods as part of the model. This involved reverse engineering the mock-ups to see what data I needed and to organize it accordingly. Some models required calls to multiple services while some models were simple and used a single service.

Additionally, I created the View part, based on the model and mock-ups. This is where the majority of my development effort went into. Initially I used the view as a high fidelity prototype to show off to the business; I would organize review meetings and use my view to solicit feedback from users on a couple of occasions I would even do live coding in front of the business, to quickly show off new concepts. The view was highly based on HTML5 programming and I used CSS3 to easily style the view based on user feedback.

Finally, I developed the controller which as mentioned in the tools section, is used to route web pages to the correct view as well as to instantiate the model with the correct settings needed by the view. Thus, this gave me somewhat of a finished product.

After finishing the above process for one of the maintenance screens and ironing out all the kinks, I was able to use this as a template to easily create the other three maintenance screen. The Product Lines screen, Product Categories screen and Topics screen were all very similar so I really just had to make one and the other two screens were more or less the same code with some minor code changes. In contrast, the Product Line rate screen was very different from the other three both in the UI looks and the underlying structure so it was made from scratch.

#### *6.2.1.4 Test Phase*

As of writing this report, the test phase is still going on. I have already completed basic tests and have yet to begin implementing thorough test cases. See the challenges and solutions section below to see what I encountered so far while testing and how I tried to solve it.

#### *6.2.1.5 Delivery of Product Phase*

A new release containing the maintenance screens I built, is scheduled to be released in mid-August, pending the completion of the test phase.

### 6.2.2 Other Tasks

In parallel to working on the maintenance screens, I was also assigned some other minor tasks which helped to complement my learning.

Initially I was asked to link some modules to an application called OAG Campus. Looking at the courses I got to learn the type of courses available to an employee at OAG, as well as get some insight into OAG's Learning Management System (LMS) platform.

Furthermore, I was asked to go through some existing test cases for the Product Cost Management application. This gave me a thorough walkthrough of the application and allowed me to learn a lot about the application and how it works. While conducting the test cases, I had to talk to some of my fellow employees and thus I got to know about a lot of people around the office.

Alongside developing the maintenance related screen I also worked on importing some data, used by the PCM application, to the database. With the help of fellow developer, Eugenio Lo, I was able to understand what the significance of the data and where to insert it into the database. This task helped me learn about the system from a back-end prospective and hence complemented with my primary task.

# 7 REFLECTION

Reflecting back on the term, it was a fantastic learning experience for me. I learned about a formal software cycle that is present at the OAG. Attending weekly stand-up SCRUM meetings helped me with getting the bigger picture of how the PCM project works and how the IT Application department functions at large. It also taught me about how agile project management is used at a company such as OAG.

I learned a great deal about the .NET Visual Basics Platform, how a .NET application works and how it is structured as well as how everything is layered from the database all the way up to the UI presented to the user.

Working with the front-end has taught me how .NET MVC can be used to build fluid applications as well as how it can be connected to the back-end through service calls. I really learned a lot about HTML5, CSS3 and jQuery and all the plug-ins that it can use to easily make a flexible UI which would have made for a daunting task if it was implemented in some of the traditional applications I worked on in a previous work term. Working on the maintenance screens gave me an experience with developing a component from scratch; in contrast to a previous work term, where I was only adding on top of a component rather than developing it completely from scratch.

In addition to some of the debugging tools found in Microsoft Visual Studio, I also learned about tools such as Microsoft SQL Server, which can be used to interact with the database outside of the actual application, as well as the developer tools in Internet Explorer which are used for client side debugging directly in the browser.

## 7.1 CHALLENGES AND SOLUTIONS

While working on my primary task, understanding the technologies used was somewhat of a challenge. With the help of Bill Moeller, I was able to get a good idea of how the technologies work together. With the help of online tutorials and looking at some of the other parts of the project I was able to get a good understanding of how to go forward with my task.

Looking at examples throughout the application involved a large variety of code written by many writers. Looking at such a variety of code helped me explore a variety of coding styles and practices, which in turn influenced my own style by a great degree. Sometimes certain code segments lead me to interact with other developers, which allowed me to explore how other developers contribute to the OAG products and hence helped me greatly increase my productivity and knowledge.

Looking at some of the requirements and trying to implement UI features using JavaScript/jQuery was very tough at times. For example, on one of the requirements was, to be able to sort columns of a HTML table, I tried implementing it on my own only to run into a dead end. Finally a fellow developer, Joe Tacconelli, suggested that I use the Datatable Plug-in. Using the plug-in, not only helped me to implement the user requirement but I was also able to use some of my own extensions and others available in the plug-in to have the UI on my maintenance screens be very intuitive and to the users liking.

During the testing phase of my main task I ran into a couple of defects. Using the Microsoft SQL Server Management Studio and the debugging tools in Microsoft Visual Studio, I was able to solve a good chunk of the defects. However, because of one of the bugs as well as to meet a user requirement I had to finally refactor some of the code within the models. This refactoring exercise allowed me to review the work I had done so far and even allowed me to detect a potential problem with the way my code was arranged.

One of the defects was due to me not knowing about certain requirements. This made me realize the importance of the data gathering phase and how when not done properly, it may lead to problems in the future.

As of writing this report, I still have one major bug I am working on, which had to do with the dataTable plug-in not deploying on one of the production servers.

## 7.2 Relation to Academic Studies and Career Development

This placement helps me enhance the learning I received at university. It helped me see some of the software engineering concepts such as agile project management in action in real life projects

During the Requirement and Design Phase, I found myself using some of the skills I acquired through a Human-Computer-Interaction class I took at the university. Through meetings with the business I was able to conduct user studies which I learned about in class. I made medium and high fidelity prototypes through Balsamiq Mock-ups, as well as the View of my application. This helped me see how concepts I learned about in class are implemented in real life which complements my learning received in university.

During the development phase, I found myself using a lot of concepts I learned about in a web application class at university. For example, when working with jQuery some variable was not being set when it should have been; I remembered learning about closures in java script and using that logic, I understood the flow of my jQuery and understood where my error was coming from. Additionally, I used the basic structure I learned about websites in the web application course and was able to apply it to my work, thus I was easily able to understand as to how the tools work as described in the tools section above.

While developing the maintenance screens I looked at and worked with a lot of pre-existing material, which is really hard to come by in a course offered at university. Since pre-existing code has been written by many developers, going through it helps one to really understand the new styles and gives you a variety of coding practices and it helps you to build your own style in the process.

Solving defects on my own screens touched upon various other components of the application, which really helped me in learning trouble shooting tips, which are easier to learn in a work scenario. In contrast, the courses I have been through in university required you to write your own code with the exception of 1 or 2 courses, where there is very little code written by the professor or teaching assistants, which doesn't offer as large a variety.

While doing group work for courses, the work ends up being divided unequally at times. For example in a group, one member may have expertise in mock-ups and user requirement gathering, hence they may end up doing a good chunk of that work while another member focuses on another aspect. Here at OAG, I was able to get a good experience with each of the aspects. For example in past courses I did not do much requirement gathering or product reviews with users as they were done by other members of my group. So I lacked those skills even though they may have been taught at a conceptual level in the course. Since I got a lot of exposure to it while working at OAG it really helped me to go back and piece together the conceptual part I learned in class to the practical skills I acquired in this work term. When I first began going in to the development phase of my term it was really overwhelming at times, as I was dealing with such a large system based on a language that was foreign to me. But with some help from fellow developers, looking at examples, and connecting skills acquired from past work terms and classes, I was quickly able to understand the technologies used on the project. Towards the end of the development phase I felt very comfortable and confident working on the project.

In conclusion, I was able to make use of some of the concepts and techniques I acquired from my studies to be able to better adapt myself to the work environment. After gaining the confident I am now able to better trust myself even when I work on different areas of the product. Working at OAG has given me a great experience as well as a new appreciation for this type of career.

# 8 ABBREVIATIONS

IT – Information Technology
PCM – Product Code Management
UI – User Interface
OAG – The Office of Auditor General of Canada
DX – Director
PX – Principal
AAG – Assistant Auditor General
VB – Visual Basics
.NET – a frame work developed by Microsoft
ASP – Active Server Pages
IDE – Integrated Development Environment
SQL – Structured Query Language
IIS – Internet Information Services
TFS – Team Foundation Server
HTML – Hyper Text Markup Language
CSS – Cascading Style Sheets
SOA – Service Oriented Architecture
WCF – Windows Communication Foundation
MVC – Model View Control
URL – Uniform Resource Locator
BL – Business Logic
OAG.tt – A project used by Product Code Management project
DAL – Data Access Layer
LINQ – Language integrated Query
LMS – Learning Management System
MIS2000 – A legacy application used at the Office of Auditor General of Canada