Industrial Placement Report 2014-2015

## Executive Summary

## Introduction

This report documents the roles and responsibilities of an Indicater Intern. It outlines the technical aspects of day-to-day tasks and operations and how the skills developed in a BSc in Computer Science at the University of Reading complement this job role. This report also analyses the positive and negatives of the internship and the skills applied and developed during the placement.

## Information on the company/organisation you are working for

Indicater was established in 2000 by Mike Day and Lou Willcock. The company based in Henley provides web-based back of house online management systems for the hospitality industry. This software management system provides over 30 modules of critical business management tools for performing business operations and managing company data such as stocktaking, sales, payroll, recipe management and online booking systems.

The founders of Indicater created and managed a contract catering business before the launch of Indicater. This meant that the initial launch of Indicater focused on provided management systems for the contract catering sector but has since expanded to supply management systems for other sectors of the hospitality industry such as the NHS, schools and universities, restaurants and hotels and stadiums and pubs.

Currently Indicater provides software solutions for over 2500 outlets. The company has developed strong business relations over the years with well achieved and respected businesses within the hospitality industry such as ‘Rhubarb’, ‘G4S’, ‘Crown’, ‘Centerplate’ and Innventure.

## Department/area you have worked in and how it fits into the company

## Your role and responsibilities and where you were able to add value

During my role as an Indicater Intern I worked in a small team of software developers and a project manager based at the Indicater headoffice in Henley. My role within the technical team was mainly centred in the testing department although the position incorporated other areas of work such as monthly software releases, editing webpages and documenting new processes and amendments to the system.

The senior technician manager Julian was very focused on developing the testing department to reinforce the delivery of cleaner, more efficient software solutions to the demands and deadlines of the company’s clients.

Like the majority of software companies the testing department is at the heart of delivering clean and effective software solutions to a client. The intensive testing of software before it is delivered to the client is essential to a company’s reputation, good client relations and future business prospects. This is the where any issues with the new development or the system that might have been introduced by the deployed development can be fixed.

In the first week of Indicater I was introduced to the technical team and my line manager Julian Bedford. Julian had started at Indicater a few weeks previous to my arrival and was also finding his roots. The technical team was built up of a handful of developers, one of which worked in the Henley office and the others that worked from another office. After the first couple of weeks at Indicater a new developer and project manager also joined the Indicater technical team.

With such a turn-around in staff new systems and practices of work were being instantiated by the technical manager Julian and development team. This meant that the other developers, client accountants and Indicater employees were also adapting to these new concepts and standards of work. Whilst the technical team where moulding into the new structure of workflow I had time to familiarise myself with the structure of the system and the technologies used to maintain the system and monitor development workflow.

As a tester my main roles and responsibilities where to work through jobs that had been allocated to me via a workflow management system. Each job would have a ticket which details the information about its development. This information would typically contain details such as a list of files that have been updated during this development, explanations of any modifications in code not incorporated within the original specification and time logged for each stage of the workflow.

After sourcing the necessary knowledge from the jobs ticket I would then need to use version control software (GIT) to synchronous my local repository with the main development repository and retrieve the latest version of the files needed to deploy the development. Once these files are collected they could then be copied over to the desired server ready for testing.

Next a manual sanity test of the updated code would be carried out to test that the development is functioning correctly and any code associated with the updated code is still executing correctly. Alongside the sanity test the results of the testing is documented on the jobs ticket in the workflow management system. The development would be tested in multiple environments with a variety of data base configurations to provide for a range of systems and clients. This testing is to ensure that the software is displayed and functioning correctly on different web browsers and hardware setups.

If a development fails testing it would be reassign back to the developer and the jobs ticket updated with evidence of the issue/bug that the tester has encountered. As I was trusted with high level permissions and had access to the source code I would often debug the issue further and try to provide extra technical information for the developers.

Code Releases:

Testing strategies: Regression, sanity, unit?

**BLACK BOX:** The internal functionalities of the program are unknown. The tester only knows the inputs and the expected outputs.

**WHITE BOX:** The internal functionality of the software is known by the tester. This is regarding with the testing if internal mechanisms within a program. testing?

Day-to-day standard incorporated the following steps:

1. Gather information and details about the job from online workflow management system.
2. Check files and materials are up-to-date
3. Gather files and materials
4. Determine which is the best site/sites to test the job on.
5. Deploy amended code and materials to target test site
6. Test targeted change and surrounding affected functionality.
7. Conclude whether or not the new code amendments function as expected and no issues are encountered. If it does not meet the requirements of its specification or its description from the ticket then it is reassigned to the developer who submitted it with testing notes to provide extra information of the bugs and issues found in testing. As the code base was nearly always ready available I would debug these bugs myself and provide further information as where in code the issue lies and what effects this is having on the system, what functionality is failing.
8. Provide and document relevant evidence of the test case that displays the new code or materials has functioned as expected, the performance of the test, any concerns or issues encountered and where further developments could be made.

I was placed in charge of deploying new developments and fixes to testing environments where they would be sanity tested ready for the next code release.

There was a quick turn-around in staff of the technical team just previous to my arrival at Indicater. This meant that most of the technical team where new members finding their feet including the senior technical manager. To my benefit the new technical team members had all come from the same previous business whichh was a larger scale business than Indicater. This was great advantage to myself as the new team where establishing and developing a testing department and methodology standards of work.

* Substantial employee growth previous to starting placement.
* Involved in the setting up the testing sites and developments.
* New team, new methods, new structure of work.
* Importance of documentation at this point was crucial as on a regular basis I would be carrying processes that no other developer has yet to carry out.

In the past due to a weak testing environment clients had received properly tested software which provided many problems for them. This reinforced the need for development of the testing environments.

Indicater software is quite large and is very costly on resources for its general maintenance. When testing developments or bug fixes this would be done by sanity checking the pages where the updated code would reside and the depended functionality around that development. Regression testing would be carried out for much larger deployments and unit testing f.----or compiled back end server programs. (Basically for complied versions of the code)

Mention the system tools developed to help with day to day tasks within the work place. This consists of the deployment and Zip programs and the GCR (Git Commit Reader) analysis tool. Comment on the simplicity but power of these tools. How I was encouraged to automate work where I could and develop tools aid daily tasks.

Diagram of the workflow designed at Indicater by the end of placement.

## Evaluation of your placement experience: knowledge/training gained, new

* Advanced documentation skills, extracting relevant information from the documentation (Confluence).
* Advanced technical skills. Working with Database driven sites. Git version control software. Project tracking software (Jira).
* MSRS (Microsoft reporting services) RDL files (Report definition language).
* C#, used to build deployment program, zip files program. System development tools.
* Verbal communication – presenting findings to the team. Diluting a problem and describing it to the client accountants so they can relay essential problems or processes about the system back the client user. I f the client accountants have a better understanding of the system so when they are recording a bug or development presented by the client users then the can ask more relevant information and pinpoint the demands of that particular task.
* Black, white box testing. Software lifecycle, Sanity, regression and unit testing.
* Mention about the use of flowcharts to help understand the process of the system.

BSc Computer Science Course modules most relevant to the placement:

* Databases
* Advanced Databases
* Software engineering
* Project management
* Programming
* Java
* Advanced Databases
* HCI and Applications

## What were the positives/negatives of the experience and what you will take

## away/have learnt from the experience

Positives:

* Exposure to all areas of the system.
* Increased responsibilities
* Small and focused team.
* Fully engaged within the project life cycle and team
* High level of permissions given.
* Big contribution to the development of the testing environment.

Negatives:

* Documentation. There wasn’t any.
* Some of the technologies used to firs create the system and proving problematic when attempting to develop the system further.
* Unrealistic time constraints and project management has unreachable milestones and deadlines.
* As I worked in a small team of developers it was sometimes quite hard to get some help and quite often you would be left to figure it out or document enough details to return to the issue.

Conclusion:

The best practice at a technical company are not always being implemented due to the restrictions of the demanding change of rapidly evolving software and implementation of the system. The quality of the software implementation dictates the ease and availability of modern developments to the system.

The importance of social legal and ethical practices within the working environment and customers confidentiality. Customer details disclosure and security. (sensitive data). Website certificates (SSL)

How online services like Indicater are affected by modifications of internet standards.

## References

www.indicater.com/clients