|  |
| --- |
| University of Reading |
| Indicater Placement Report 2014-15 |
|  |

|  |
| --- |
| Thomas Robert Bedford  10-5-2015 |

Contents

[Executive Summary 2](#_Toc431747795)

[Introduction 2](#_Toc431747796)

[About Indicater 2](#_Toc431747797)

[Department worked in and its role within the company 3](#_Toc431747798)

[Roles and Responsibilities 3](#_Toc431747799)

[Adding Value to the Department 4](#_Toc431747800)

[Evaluation: knowledge/training gained, new skills 5](#_Toc431747801)

[Communication and Documentation 5](#_Toc431747802)

[Technical 5](#_Toc431747803)

[Computer Science modules most beneficial to the job role 5](#_Toc431747804)

[Positives/negatives of the placement role 6](#_Toc431747805)

[Conclusion 7](#_Toc431747806)

[References 7](#_Toc431747807)

Indicater Placement Report 2014-2015

## Executive Summary

P

P

P

P

P

p

## 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.

## About Indicater

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 lead the initial launch of Indicaters focus 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 worked in and its role within the company

During my role as an Indicater Intern I worked in a small team of software developers and a project manager based at the head office 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.

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.

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.

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

## Roles and Responsibilities

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 in relation to the 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 to test 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 I would reassign the job back to the developer and document evidence of the issue/bug that has been exposed in testing. As I was trusted with high level permissions and had access to much of the source code I would often debug the issue further and try to provide extra information of the problem for the developers.

Typically on a monthly bases there would be a software release where all new development code is deployed to the live servers. This would require a full sanity check of client sites and services to check for any configuration issues or bugs introduced to the system. This would be simple black box testing where functionality within client sites are tested for expected and unexpected outputs.

## Adding Value to the Department

Many aspects of the testing job where mundane repetitive processes that would be carried out manually. This was costly on company time and resources and can easily introduce human errors whist testing or releasing new code.

After the first month at Indicater I was beginning to settle into the job and manage my work load comfortably. As the release of new developments sometimes required me to gather, back up and release hundreds of files from multiple directories I began to explore the possibility of automating sections of this manual work. It was here where I developed a few simple tools that would automate these processes for me.

Firstly I developed a simple batch driven program that could be passed a text document containing a list of files as a parameter. This list of files would detail the full file paths of the files. The program would create a text based log file that would detail information about the programs execution such as files deployed successfully, files not deployed with error details and useful information such as the source and target of the package being built. This was a very simple but useful software tool which could be used for building and deploying release packages. The development of this tool saved company time and resources by removing these lengthy manual processes.

The next tool I developed was a zip program. This again was a simple batch driven program that would be passed a text document as a parameter. When carrying out monthly releases full site directories would need to be backed up. Some of these directories where a substantial size and the proportion of the directory contained documents and files that were auto-generated upon use of the sites functionality. The Zip program was concerned with backing up just the source code, libraries and dependencies only to reduce the amount of storage needed for the back up’s.

Before collecting the files for the release package there was much work needed to analyse which files were associated with a particular release package. This too was done manually proving extremely time consuming and again quite often introduced human errors. The senior technician later began to develop an analysis tool that would gather file deployment lists based on references to GIT commit messages. The program dubbed the GIT commit analyser was a very powerful tool for managing software releases. The program was later developed further by Julian and myself to incorporate the tools I had developed in previous weeks. This allowed the program to analyse, collect, back up and release deployment packages.

Nearer to the end of the placement I had the chance to get involved with developing automated testing scripts for websites. Again this removed the need for mundane manual work and saved company resources and time. This was a great advance to the testing department as know new code could be tested on multiple clients with speed and ease at the click of button.

Amongst developing software tools to aid daily activities and processes I was involved with building and managing new test and integration environments for the developers. This allowed more controlled testing to be carried and easier tracking of code versions.

Throughout the placement I assisted in the creation and documentation of new testing environments. Documentation was regularly updated with the systems current status.

## Evaluation: knowledge/training gained, new skills

Categorized below are the skills and abilities learnt and developed during my role as an Indicater Intern.

### Communication and Documentation

* Analysing and extracting information from documentation.
* Documenting and expressing relevant information.
* Getting involved in meetings and presenting findings to other team members.
* Verbal communication is more punctual, fluent and engaging.
* Better knowledge of technical terminology and acronyms.
* Documenting system status and development standards using company collaborative and workflow management software.
* Diluting technical problems so the client accountants can explain issues to a client.

### Technical

* Working with version control software (GIT).
* MSRS (Microsoft reporting services) RDL files (Report definition language).
* Used C# to build the software tools. Deployment program, zip files program and automated testing scripts.
* Improved understanding and use of black and white box testing. Sanity, regression and unit testing.
* Clearer understanding of the Software lifecycle. Waterfall and agile workflows.
* Technical diagrams such as flowcharts to help understand and illustrate program and business logic.
* Working with database driven sites has allowed me develop my SQL (Structured Query Language).
* Developing automated testing scripts (Selenium).
* Back end server process
* Modelling and displaying data.

### Computer Science modules most beneficial to the job role

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

## Positives/negatives of the placement role

Positives:

* High level of permission provided exposure to a large area of the system.
* Directly involved with important large projects.
* Small and focused team.
* Encouraged to incorporate new technologies and languages within work.
* Fully engaged within the project life cycle and team.
* Included in the discussion and creation of development specifications.
* Given time to settle in and find your comfort level.
* Experience with common technologies and standards within the computer industry.

Negatives:

* Parts of the systems documentation is very unclear and in some cases non-existent.
* Some of the languages and technologies used to first create the system proved problematic when attempting to develop further with new technologies.
* A few times there seem to be unrealistic time constraints and deadlines.
* Although working in a small team of developers allows you that exposure it also proved quite difficult to get help with tasks.
* Some technologies in use are dated and prove troublesome.
* There was a large turn-around in staff just before I arrived at Indicater which meant the first few months the standard of work was still being established.
* With such a small testing team I didn’t get much time in other areas of the technical workplace.

## Conclusion

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.

The testing department was undeveloped for a company of its volume when I arrived at Indicater. This was good as it allowed me to get heavily involved with the development of the testing environment. I help to build test and integration sites, document new features of the system and even standardize working practices. The Indicater codebase is very large and very costly on resources for its general maintenance. With new developments coming in and the demands of the clients re-writing parts of the system seem unrealistic in the near future.

During my placement I learnt the importance of supporting documentation and how crucial it is to the future development of the system. A number of times I would be carrying out processes that had not yet been carried out by other developers which reinforces the need for documentation. Another example was having to carry out changes to the system and the documentation, if any is not always logical or makes sense.

I have a better appreciation for the legal and ethical aspects of a computer company. At so many points in the development process the technical team needed to analyse the proposed solution to establish whether the development is legally or ethically viable. The storage, access and display of business sensitive data.

Online services like Indicater are affected by modifications of internet standards. In my placement I discovered how frequently these standards are modified and the havoc they can cause a business if they are not up to standard.

Most importantly from my time at Indicater I understand that when creating a specification for a development the client rarely understands or clearly knows what it is that they actually won’t.

## References

www.indicater.com