Report 1

Work Integrated Learning – MonicIT

Francesco Ferraioli – n8323143



Executive Summary

This report will give a short overview of my first thirty days of industry-based work as a software engineer. In the first thirty days of placement, I found myself working for MonicIT, a small software company in Brisbane CBD. At MonicIT I was able to move quite quickly up in the company, starting off as a tester for their android app to one of the main developers of the company. This meant that I found myself participating in a variety of different tasks. Being a small company I was able to get a lot of mentoring from the senior developers. I gained a lot of experience in my first thirty days of my placement at MonicIT, not only technical skills, but I also learned the everyday duties and tasks of professionals in my field.

Table of Contents

1.0 Work Place 4

1.1 Introduction 4

1.2 Employer Organisation 4

1.3 Work Environment Conditions 4

2.0 Work Activities 4

2.1 Major Activity and Reflection 1 5

2.2 Major Activity and Reflection 2 5

3.0 Conclusion 5

References 5

Appendicies 5

Appendix A: Work Log 5

Appendix B: Reflective Notes 6

# 1.0 Work Place

## 1.1 Introduction

Before commencing my work at MonicIT, I had little to no experience besides my studies at QUT in my Bachelor of Software Engineering. Midway through my fifth semester of the degree I saw an opportunity for a tester position at MonicIT and I decided to take it. At that stage I had completed more than half of my degree and I felt that I had already gained many vital skills and this provided me with the confidence of fulfilling the position. Many of the subjects I undertook at QUT involved working in teams as well as an individual. These subjects gave me the experience I needed to know what is required when working in a team and taught me the crucial skills of teamwork. They taught me not only that communication is key, but also how to communicate effectively. The individual aspects to these subjects helped me to become a stronger problem solver, which was certainly something I needed if I wanted to be a member of a team in the industry. By that time I had received outstanding grades for those subjects, with a GPA of 6.6875.

## 1.2 Employer Organisation

My first thirty days of industry-based work were undertaken at MonicIT. MonicIT was founded in 2008. They started off by providing customized software to companies but later began focusing on software for limousine companies as the founders saw some real potential in the market. After a limousine company owner approached them with interest for software to assist them to run their business, they did some more research and found that many limousine companies were in need of this kind of software. After that, they decided to spend all their resources on creating this software for limousine companies. Just like all their other software, Limousine Management System (LMS), was written as a Ruby on Rails web app. MonicIT decided to take this software a step further and began creating an android app for limousine drivers to use that would communicate with the LMS.

My main responsibilities and duties at MonicIT during my first thirty days of placement revolved around ensuring that the android app worked as expected. This involved running tests on the android app against the desired functionality and if bugs were found, to report them in a professional manner to the developers to fix. As the app and LMS were closely linked, this also meant that the functionality of LMS was also tested and sometimes bugs were found there too. Initially the testing involved only manual testing, but then moved to automated tests which involved me writing scripts to run the User Interface.

## 1.3 Work Environment Conditions

Professional Indemnity insurance was designed to protect workers against legal costs and claims for damages to third parties, which may arise out of an act, omission or breach of professional duty in the course of your business (Aon, 2014). Damage can include but is not limited to, any loss of data that belongs to the third party, and certainly includes any financial costs that may arise for the company. A majority of the web applications that are being implemented can be considered e-commerce applications (Schafer, Konstan, & Riedl , 1999). E-commerce applications deal very heavily with financial transactions. These applications have to be not only impeccably correct, but also heavily secure. Any breach of these can result in damage to the third party (clients). A breach could be not securing a transaction well enough that it can be seen or even altered, causing confidentiality and possibly even integrity breaches. It is vital for software engineers to have professional indemnity insurance, especially those working on e-commerce applications. The application MonicIT was developing, LMS, frequently dealt with financial transactions, paying for trips through the application and is certainly classed as an e-commerce application. For my placement at MonicIT, it was crucial for an inexperienced professional as myself to have this.

Workplace Health and Safety is the discipline that deals with protecting the health and safety of all stakeholders in the workplace (Davitt, 2012). Workplace health and safety is not a major issue for software companies but certainly needs to be considered. MonicIT’s office is located on the top floor of one of the highest building in the Brisbane CBD. Workplace health and safety was certainly a part of the building and MonicIT, as a customer of the building, needed to abide by any of defined rules regarding workplace health and safety that were set for the building. The building had stairs to use in case of a fire and in every room was a diagram detailing the meeting place in case of a fire and how to direct yourself there. It was important for workers at MonicIT to be aware of the procedures in case or emergencies. Fire detectors were present in every room of the building and were tested regularly.

Intellectual Property is used to describe the results of creative and innovative endeavours (Dutfield, 2008). Intellectual Property is a vital part of any company and certainly crucial for software companies. Without Intellectual Property a developer could take any work that they does to be their own. At MonicIT it was made clear from the start that any work that is done using MonicIT’s resources were the property of MonicIT. MonicIT ensured that Intellectual Property was part of every contract that an employee signs with them. For developers, especially it is important to know and understand this and MonicIT made it very clear in the contract. Even for a position like mine, the contract stated very clearly the agreed terms regarding Intellectual Property.

Quality Assurance is any systematic process of checking to see whether a product or service being developed is meeting specified requirements (Rouse, 2014). It is crucial for any company to have quality assurance when creating a product for customers. MonicIT valued quality assurance highly. My main duties during my first thirty days of placement at MonicIT revolved around ensuring that the product was meeting the requirements. This is vital for customer satisfaction and thus revenue for the company.

# 2.0 Work Activities

Work Activities

## 2.1 Major Activity and Reflection 1

Major Activity and Reflection 1

## 2.2 Major Activity and Reflection 2

Major Activity and Reflection 2

# 3.0 Conclusion

Conclusion

# References

Aon. (2014). *Professional indemnity insurance.* Retrieved August 12, 2014, from AUSDANCE: http://ausdance.org.au/articles/details/what-is-professional-indemnity-insurance

Dutfield, G. (2008). Global intellectual property law.

Davitt, M. (2012, April). Workplace health and safety act 2012.

Schafer, B., Konstan, J., & Riedl , J. (1999, November). E-Commerce Recommendation Applications . Minneapolis.

Rouse, M. (2014). *Quality Assurance (QA).* Retrieved August 12, 2014, from Tech Target: http://searchsoftwarequality.techtarget.com/definition/quality-assurance

# Appendicies

## Appendix A: Work Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Day | Date | Hours worked | Type of work | Description of work |
| 1 | Monday, 11/03/2013 | 5 | Manual Testing / Documenting | Manual Tests on Android App Documenting Results Found Bug and Reported bug: Reset of the icon counters (App) |
| 2 | Wednesday, 13/03/2013 | 8 | Manual Testing / Documenting | Manual Tests on Android App Documenting Results Found bug: Date selection Found bug: Found 500 response (Web) |
| 3 | Thursday, 14/03/2013 | 8 | Manual Testing / Documenting | Documenting bugs Writing steps to reproduce bugs |
| 4 | Monday, 18/03/2013 | 5 | Manual Testing / Documenting | Manual Tests on Android App Documenting results Proposed Enhancement to functionality |
| 5 | Wednesday, 20/03/2013 | 8 | Manual Testing / Documenting | Manual Tests on Android App Documenting results Proposed Enhancement for user friendliness |
| 6 | Thursday, 21/03/2013 | 8 | Manual Testing / Documenting | Manual Tests on Android App Documenting results Found bug with invoicing |
| 7 | Monday, 25/03/2013 | 5 | Manual Testing / Documenting | Cross Platform Testing of Android App |
| 8 | Wednesday, 27/03/2013 | 8 | Manual Testing / Documenting | Documenting current features |
| 9 | Thursday, 28/03/2013 | 8 | Manual Testing / Documenting | Documenting current features |
| 10 | Monday, 01/04/2013 | 5 | Autonomous Testing / Documenting | Began learning how to write autonomous tests Testing Tool: TestWise Testing Language: RWebSpec/Selinium |
| 11 | Wednesday, 03/04/2013 | 8 | Autonomous Testing / Documenting | Continued learning how to write autonomous tests |
| 12 | Thursday, 04/04/2013 | 8 | Autonomous Testing / Documenting | Continued learning how to write autonomous tests |
| 13 | Monday, 08/04/2013 | 5 | Autonomous Testing / Documenting | Continued learning how to write autonomous tests |
| 14 | Wednesday, 10/04/2013 | 8 | Autonomous Testing / Documenting | Writing my first autonomous tests |
| 15 | Thursday, 11/04/2013 | 8 | Autonomous Testing / Documenting | Writing autonomous tests to run against webapp |
| 16 | Monday, 15/04/2013 | 5 | Autonomous Testing / Documenting | Writing more simple autonomous tests |
| 17 | Wednesday, 17/04/2013 | 8 | Autonomous Testing / Documenting | Writing autonomous tests to expose bugs |
| 18 | Thursday, 18/04/2013 | 8 | Autonomous Testing / Documenting | Writing autonomous tests for validation |
| 19 | Monday, 22/04/2013 | 5 | Autonomous Testing / Documenting | Writing more complex autonomous tests Authentication Authorization |
| 20 | Wednesday, 24/04/2013 | 8 | Autonomous Testing / Documenting | Exposing known bugs through autonomous tests |
| 21 | Thursday, 25/04/2013 | 8 | Autonomous Testing / Documenting | Finding new bugs through autonomous tests |
| 22 | Monday, 29/04/2013 | 5 | Autonomous Testing / Documenting | Fixing test issue where test where giving false negatives |
| 23 | Wednesday, 01/05/2013 | 8 | Implementation / Autonomous Testing | Began learning Ruby on Rails for the new project |
| 24 | Thursday, 02/05/2013 | 8 | Implementation / Autonomous Testing | Wrote HTML for the web app |
| 25 | Monday, 06/05/2013 | 5 | Implementation / Autonomous Testing | Wrote HTML for the web app |
| 26 | Wednesday, 08/05/2013 | 8 | Implementation / Autonomous Testing | Wrote HTML for the web app |
| 27 | Thursday, 09/05/2013 | 8 | Implementation / Autonomous Testing | Began writing JavaScript for interacting with HTML |
| 28 | Monday, 13/05/2013 | 5 | Implementation / Autonomous Testing | Wrote my first ajax call used to update fields on the database via the webapp |
| 29 | Wednesday, 15/05/2013 | 8 | Implementation / Autonomous Testing | Finished implementing ajax call to update fields on the database via the webapp |
| 30 | Thursday, 16/05/2013 | 8 | Implementation / Autonomous Testing | Wrote tests for the ajax call functionality |

## Appendix B: Reflective Notes

|  |  |
| --- | --- |
| Situation | The android application has to send invoices |
| Task | Undertake manual test against the invoice functionality of the android app |
| Action | 1. Brainstormed normal test cases 2. Brainstormed edge test cases 3. Wrote steps for tests 4. Wrote the expected results 5. Undertook the tests 6. Documented results |
| Result | Some of the tests passed – the outcome was what was expected   * Invoice had the right amount * Invoice included correct items   Some of the tests failed – the outcome was different than expected   * Some invoices were issued to the incorrect person * Some invoices didn’t send at all |
| Learnt | * Testing functionality after implementation is crucial * Write down all the steps taken so that if an error occurs, the issue can be reproduced * Invoices are extremely important and must be tested thoroughly * Good practice to brainstorm test ideas and test steps before commencing testing * Good practice to write down the expected results next to each step * Good practice to think of a good set of normal test cases as well as many edge cases |

|  |  |
| --- | --- |
| Situation | The web application implements authentication and authorization |
| Task | Write autonomous tests to test authentication and authorization functionality |
| Action | 1. Brainstorm test cases for authentication and authorization (normal and edge) 2. Write down the steps for each test 3. Write down the expected outcome for each step 4. Turn the worded steps into scripts that can be run against the User Interface 5. Run the tests |
| Result | * After writing the test cases in RWebSpec some of the test passed and some didn’t * Some of failing test cases failed due to test bugs. The tests were not written correctly and caused failure. Most of them were due to incorrect ID names for the elements. * Other failing test cases were failed due to browser speeds versus driver speed. The driver would be too fast and would try to interact with an element when the browser has not yet finished loading it. |
| Learnt | * It is good practice to test your test cases as you write them to minimize test bugs * Browsers can be slower than the driver and to be aware of this when implementing tests. This can be done by making the driver wait for something to happen that is known to happen when the page is being loaded, before trying to access elements. * Authentication and Authorization is vital and needs to be tested thoroughly * Authentication includes changing and restoring of passwords |

|  |  |
| --- | --- |
| Situation | There are known bugs in the system |
| Task | Write autonomous tests to expose the bugs |
| Action | 1. List down the known bugs 2. Write down the steps to expose each bug 3. Think of any other way that any of the bugs may be exposed and write the steps for those 4. Turn the worded steps into scripts that can be run against the User Interface 5. Run the tests and ensure that if they fail, they fail because they are exposing the bugs |
| Result | * After writing the test cases all of the test failed * Some tests failed because of the bug * Some tests failed because of a different bug in the web app * Some tests failed because of a test bug |
| Learnt | * When exposing bugs with autonomous test, it is essential to ensure that they fail because of the bug, not due to a test bug * Exposing bugs with autonomous tests is essential in Agile as these tests will run every build and these can be used to ensure the bug doesn’t exist even in the future. |

|  |  |
| --- | --- |
| Situation | My colleague had implemented a feature but hadn’t had the time to write tests for the feature |
| Task | Write autonomous tests for the functionality |
| Action | 1. Take a look at the feature my colleague had implemented 2. Undertake manual tests to understand the feature 3. Write down normal test cases for the feature 4. Write down edge test cases for the feature 5. Turn the worded steps into scripts that can be run against the User Interface 6. Run the tests |
| Result | * The tests came back all green. This occurred because I was careful about the way I implemented the tests and ran the tests as I was implementing them and only once they were complete, would I truly be testing the functionality and not the test. |
| Learnt | * In Agile, a feature has not finished to be implemented until there is at least a test for it * Firstly running tests manually is not bad practice when writing test scripts * How important tests are in development * How important it is to run tests as I am implementing them to remove any possible test bugs |