Using Gamification to Encourage Continuing Computer Science and Information Technology Education in South African Corporate IT Departments

Laurie Butgereit[1,2]
[1]Nelson Mandela Metropolitan University
[2] Blue Label Telecoms
South Africa
laurie.butgereit(at)nmmu.ac.za

Abstract

Cervero argues that the goal of professional practice is wise action and that knowledge which is acquired from practice is necessary in order to achieve the goal of wise action. In corporate IT departments, however, it is often hard to provide the necessary time for ongoing continuing education of employees. When employees in a corporate IT department are not kept up-to-date with technology, then the corporate is at risk of having outdated software which runs slowly and may be subject to security breeches. This paper describes the use of gamified treasure hunts to encourage employees in a busy IT department to learn about new programming techniques, new utilities and tools, and general security issues. The treasure hunts combined both virtual and tangible objects. The treasure hunts would start with posters containing a QR code and would end with a tangible treasure box full of chocolates.

1 Introduction

Corporate IT departments consist of professionals at varying levels of expertise and experience. Cervero argues that the goal of professional practice is wise action. He continues by stating that "knowledge acquired from practice is necessary to achieve the goal of wise action." He further believes that "the primary goal of continuing education should be to improve professionals' abilities to engage in wise action" [1].

Often, however, in a busy corporate environment, there is no time to provide continuing education and training to the employees. In many cases, it appears easier to hire new expertise than to train up existing employees with the required new knowledge.

Gamification is the use of game mechanics in a non-gaming environment in order to solve problems and engage users [2, 3]. Gamification is not to be confused with game based learning. Game based learning takes educational content and inserts it into a game [5]. Gamification, on the other hand, takes the mechanics of games which make them engaging and often addictive and inserts these mechanics into other environments in order to solve problems in an engaging way.

Gamification can now be seen in many environments including business, health, and education. Airlines which offer frequent flyer miles are, in fact, using common game mechanics such as points, badges and leaderboards in order to encourage loyalty amongst its customers. Mobile health applications which track the distance that a person has jogged or walked and then allow the person to upload this information to a website or social media site and compete against other people around the world is another example of gamification to solve problems.

In formal education, teachers have been using types of game mechanics to engage their pupils long before the word *gamification* was coined. Star charts in primary school, badges on school uniforms, certficates awarded at ceremonies, different colored gowns at graduation, and privledges for good behaviour are all game mechanics which have been inserted into formal education.

Corporate IT departments, however, often do not have time for formal education. Not only is time scarce, trying to find a *common* time where all employees can attend a seminar or lecture is even more difficult. In addition, the corporate may not have a classroom or meeting room which is large enough for everybody to attend at the same time even if a common time could be found.

This paper describes the use of fun treasure hunts where employees are enticed to take part in a search for a physical treasure box full of chocolates which is hidden somewhere in the office. Clues to the location of the physical treasure box can be found by writing specific types of programs such as accessing a NoSQL database or such as decrypting a file using a symmetric key. The treasure hunts embody a number of game mechanics such as a story line and competition to engage employees in a fun manner. In addition, the treasure hunts are designed in such a way that employees can follow the treasure hunt in their own time when it is convenient to themselves.

978-1-943436-06-4 / copyright ISCA, CATA 2017
March 20-22, 2017, Honolulu, Hawaii, USA

2 Continuing Education

As mentioned in Section 1, Cervero holds that the primary goal of continuing education is to improve a professional person's ability to engage in wise action and that the knowledge acquired from practice is necessary to achieve the goal of wise action [1].

Although IT does not have mandatory continuing education requirements for its employees, the fall out from employees not keeping up-to-date with latest techniques and utilities can be seen in the popular press. Security breeches are often possible through out-dated software and obsolete programming techniques. For example, employees who are not aware how easy it is to mount a brute-force attack or a dictionary attack often don't know how to design websites against such an attack.

In order for a corporate IT department to generate and maintain computer code which is efficient and secure, continuing education of the members of that department is necessary.

3 Treasure Hunts

The company under research is one of the largest vendors of secure tokens in South Africa. The secure tokens can be used to purchase cell phone airtime and data bundles, to purchase electricity, and to purchase types of mobile money. In addition to these secure tokens, the company also offers facilities to purchase topup airtime and data bundles, topup electricity, bill payments, traffic fine payments, and event ticketing. The IT department is primarily a Java programming shop. Employees range in qualifications with some employees having the equivalent of a secondary school diploma with a possible commercial web development certificate with progress towards a Java certification and with some employees having post graduate degrees.

The treasure hunts were originally implemented as a three month research project to see if important knowledge could be provided to the employees in a way that was fun and engaging. As will be seen in Section 9 of this paper, the treasure hunts have been extremely successful and the project has now been running continually for a year.

The treasure hunts are weekly events. On Fridays cryptic posters are present in the office in the early mornings. These posters often have one or two images and a QR (quick response) code for further information. The posters are unsigned and appear anonymously on people's desks, on the scrum boards, in the kitchens, by the coffee machines, and on empty tables. A sample poster which introduced a treasure hunt based on the Cassandra NoSQL database can be see in Figure 1 .

Figure 1: Sample poster to introduce Cassandra

The QR code would link to a short lesson about the topic. The lesson would then supply a challenge of some sort. In the case of the treasure hunt about Cassandra, the challenge was simply to connect to a local Cassandra database and access some particular data inside the database. In order to do this, the employee would need to download libraries and/or utilities and learn about Cassandra along the way. Hidden inside the Cassandra database was the location of a little treasure box full of chocolate as can be seen in Figure 2 .

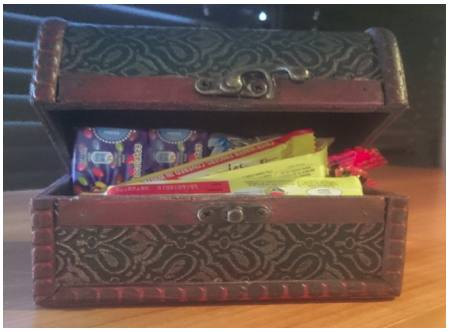

Figure 2: Treasure box of chocolates

During the course of the year that this project has been running, the topics of the treasure hunts have been centered in four main domains. These four domains are 1) Java language, Java programming techniques, and support tools such as Subversion, Maven, Ant, Jenkins, etc 2) Java Enterprise Edition 3) NoSQL databases 4) Security (include historical encryption, steganography, and modern public key/private key encryption). These domains were intermixed so that the topics were always new and exciting.

The following four sections of this paper give an overview of the various types of treasure hunts in the four

different domains. After those four sections, the results are provided followed by concluding remarks.

4 Java and Support Tools

A number of the employees in the company under research did not have Java certifications and were working towards achieving such certifications. These employees had limited professional experience and were not familiar with supporting tools such as Ant, Subversion, Git, Maven, Jenkins, etc. In addition, these less experienced programmers were not familiar with various data structures such as lists and queues and were also not familiar with programming techniques such as recursion.

A few of these treasure hunts include:

1. Modifying a Maven `pom.xml` file to automatically download a specific Java library from a repository maintained by the company under research. When specific methods were executed, the location of the treasure box would be revealed.
2. Creating a Jenkins Continuous Integration project via the Jenkins website. The project must bring a specific project down from a central subversion repository (again maintained by the company under research), build the project, and execute a specific project option. The location of the treasure box would be revealed on the Jenkins console output.
3. Checking out a specific Subversion project, compile and run. The location of the treasure box would be revealed. To ensure that the participants knew how to commit source files back to a Subversion repository, the treasure hunt encouraged them to move the treasure box to a new location, update the appropriate source files and commit the changes back to the repository.

These types of treasure hunts were especially popular with the younger less experienced members of the department.

5 Java Enterprise Edition

Many of the programmers in the company under research were not familiar with any JEE (Java Enterprise Edition) objects. The treasure hunts centered on the programmers learning about open source JEE containers such as Tomcat, Wildfly, and Glassfish and learning about the various JEE objects such as Servlets, Restful interfaces, and generalized webapps.

In general, the JEE based treasure hunts had the following steps:

1. Download and install a specific JEE container.
2. Download and build a specific project which was already partially built and hosted in a combination of maven repositories and subversion repositories
3. After building and deploying this project locally on the participant's own workstation, the participant would find the location of the treasure box.
4. To encourage fun and excitement, however, the participant was then encouraged move the physical treasure box to a new location, to change his or her version of the project to give the new location of the treasure box, and to leave a QR code at the original location of the treasure box to point treasure seekers in the correct direction.

These steps ensured that the participants could build, deploy and modify various types of JEE projects.

6 NoSQL Databases

The company under research primarily used SQL. There was no knowledge about any of the more modern NoSQL databases. The term NoSQL, in fact, is hard to define. In his book “NoSQL Distilled”, Martin Fowler explains that the current usage of the term *NoSQL* was coined in 2009 as a Twitter hashtag for a meetup in San Francisco to discuss various new database technologies [9]. According to Fowler, the term “caught on like wildfire, but it’s never been a term that’s had much in the way of a strong definition.” In general the term has come to identify a number of databases which have the following characteristics:

1. They do not use SQL (although they may have their own query language)
2. They are primarily open source (although they may have a commercial version available)
3. They are designed to run on clusters
4. They are schema-less

Researchers have grouped the current NoSQLs which are available currently into four broad categories: key-values stores, columnar databases, graph databases, and document databases [7]. For the treasure hunts, one typical NoSQL database was chosen out of each category and two treasure hunts in consecutive weeks were conducted for each of the databases. The chosen databases were Redis (key-value store), Cassandra (columnar database), Neo4J (graph database), and MongoDB (document database). The treasure hunts were designed to show some of the advantages of each of the category of NoSQL database. For these treasure hunts, a central server was used to host the various NoSQL server which was the basis of the treasure hunt.

In general, the NoSQL based treasure hunts had the following steps:

1. Access the NoSQL data base on the central server.
2. Find some information which would lead the location of the physical treasure box.
3. Move the physical treasure box.
4. Update the central NoSQL data base to hold the new location of the treasure box.

To emphasize some of the specific features of each NoSQL some slight twist would be added. For example, in the case of the Cassandra database (which is a columnar database and ideally suited for big data applications), data was downloaded from the World Bank and pre-loaded into the Cassandra database. Participants had to execute various types of calculations on the data in order to find the treasure box.

In the case of Redis and MongoDB (where it is easy to store data of different types), the physical location of the treasure box was given in the form of images and movies. MongoDB had an additional treasure hunt emphasizing the native geolocation features that the database provided. In the case of Neo4J, interlinking data was scraped from various consumer complaint websites and preloaded into the database.

7 Security

Recent news headlines about computer breeches and cyber-heists prompted a large number of security related treasure hunts. The general format of these security related treasure hunts was to hack some sort of file or website – the underlying belief being that if programmers saw how easy it was to launch various types of attacks such as brute force, dictionary, and rainbow attacks, then the programmers would be more prepared to code against such attacks.

Treasure hunts included:

1. Programming some historical encryption schemes such as Caesar Cypher, Vigenère Cypher, and AutoKey Cypher and to launch attacks against data which was encrypted using these algorithms.
2. Launching a brute force attack on a symmetrically encrypted datafile using a combination of Jasypt and BouncyCastle. In order to ensure success of the treasure hunt in a limited period of time, the participants were given some indication of the key such as it being a numeric less than six characters or that it was an alphanumeric less than four characters. The encrypted file gave the location of the treasure box.
3. Launching a dictionary attack against a provided website using a combination of Selenium and a provided dictionary. Upon successful login, the provided website gave the location of the treasure box.
4. Launching a modified dictionary attack against a provided website which encouraged the use of a recursive algorithm to generate all combinations of upper case and lower case letters within the word.
5. Launching a further modified dictionary attack against a provided website which encouraged the use of a further recursive algorithm to general all combination of upper case, lower case and 133t speak within the word.
6. Launching a rainbow attack against a provided website given a rainbow table and an unsalted password file.
7. Launching brute force attacks against encrypted PDF documents

All of these treasure hunts ensured that the participants understood how easy it was to launch these attacks.

8 Costs to Company

In order to create an engaging, fun treasure hunt, approximately one full man-day was required per treasure hunt. This time included not only designing and creating the treasure hunt but also included time to download specific software (such as NoSQL database servers) and to install this software on servers which the employees could access. In addition to this manpower cost, a server (or virtual server) was required for many of the treasure hunts.

9 Results

As mentioned in Section 3, the original research project for this was initially designed for three months. However, it has proved to be extremely successful and the project has been extended and has been running now for over a year.

As with all research projects, evaluation is critical. The evaluation of the treasure hunts was done via links to SurveyMonkey. SurveyMonkey is a free online survey tool. Links to these surveys were often included in the body of the treasure hunt and were also emailed out to a programmer mailing list independent of the actual treasure hunts.

The survey questions varied, but always included the following two:

1. Was the treasure hunt too easy? Too hard? Just right?
2. Please provide topics for new treasure hunts.

Input from the programmers was critical to inform subsequent treasure hunts. This ensured that the treasure hunts helped the programmers. The surveys gave shy and

inexperienced programmers an opportunity to say what they needed to learn.

In addition, management expressed their positive feelings about the treasure hunts remarking that the programmers are attaining wider knowledge of tools and techniques.

More important, however, was the general conversation around the coffee machine after the security treasure hunts. Many of the participants were completely unaware at how easy it was to launch such attacks. General discussion ensued about how to modify websites of the company under research to protect against such attacks.

10 Conclusion

In his 1965 paper “Cramming More Components onto Integrated Circuits” [6] the contents of which have been shortened to “Moore's Law”, Gordon Moore held that computer hardware processing speeds and power double every two years. Less well known, however, is “Wirth's Law” (named after Swiss professor of computer science Niklaus Wirth) which states that “software gets slower faster than hardware gets faster” [8]. In other words, despite hardware getting faster, the end product is getting slower.

To help mitigate against this, it is important that corporate IT departments continue to provide ongoing education and training to their employees. If this ongoing education and training is in the form of actual practice, then the knowledge acquired from this practice will lead to wise action [1].

This paper described the use of gamified treasure hunts to encourage employees in a corporate IT department to learn about new utilities, tools, and programming techniques. The treasure hunts were specifically created in a fun and entertaining manner to encourage participation.

Comments from the participants were gleaned from SurveyMonkey and from general informal comments from the participants after the treasure hunts were completed. These comments informed subsequent treasure hunt topics. The participants showed enthusiasm for the treasure hunts and shared that they enjoyed learning about new tools, utilities, and techniques. The original research project was for a limited three month period, but the results from the participants encouraged company management to ask that the treasure hunts continue as a normal weekly event in the company.

The treasure hunts have now become an established part of the environment in the company under research.

References

[1] Ronald M. Cervero. The importance of practical knowledge and implications for continuing education. *Journal of continuing Education in the Health Professions* 10: 1: 85-94, 1990.

[2] Sebastian Deterding, Dan Dixon, Rilla Khaled and Lennart Nacke. From game design elements to gamefulness: defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments9-152011.

[3] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara and Dan Dixon. Gamification. using game-design elements in non-gaming contextsIn CHI'11 Extended Abstracts on Human Factors in Computing Systems2425-24282011.

[4] Gill Furze and Pat Pearcey. Continuing education in nursing: a review of the literature. *Journal of advanced nursing* 29: 2: 355-363, 1999.

[5] Ian Glover. Play as you learn: gamification as a technique for motivating learners. 2013.

[6] Gordon E. Moore. Cramming more components onto integrated circuits (reprint). *Proceedings of the IEEE* 86: 1: 82-85, 1998.

[7] Ameya Nayak, Anil Poriya and Dikshay Poojary. Type of NOSQL Databases and its Comparison with Relational Databases. *International Journal of Applied Information Systems* 5: 4: 16-19, 2013.

[8] Philip E. Ross. 5 Commandments [technology laws and rules of thumb]. *Spectrum, IEEE* 40: 12: 30-35, 2003.

[9] Pramod J. Sadalage and Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence.* Pearson Education, 2012.