



MIEIC – October 2017

Software Failures

ESOF

Class 1 - Group F

Diogo Peixoto Pereira – up201504326

Maria Eduarda Santos Cunha – up201506524

Pedro Miguel Ferraz Nogueira da Silva – up201505460



Index

1. US Prisoners Released Early.....	3
2. Amazon 1p Glitch	5
3. Bar Exam Failure	7
4. “No Silver Bullet” Reading Notes	9



1. US Prisoners Released Early

Since 2002 and during a period of 13 years, inmates have been released early due to an update introduced in 2002 followed by court ruling about applying good time credits¹. The total number of inmates released totals 3200 thanks to a decrease in their sentences.

On average, the sentences were calculated and decreased by a total of 49 days and 3% of the inmates released were given excessive good time credits making their sentences greatly smaller.

This bug was discovered in 2012 when a prisoner's family discovered he was getting released earlier than expected. Although the officials started working on the problem right away, it was only when a new IT boss for the DoC² was appointed that the problem received the right amount of attention.

Videos: Inmates lives after being released:

<https://youtu.be/WRlELk2xXtc>

Important Dates: 2002 - System update causes glitch.

2012 - DoC discovers the mistake.

2015 December 22 - News released to the press.

Consequences: Inmates have been released earlier than expected, during a period of 13 years.

Impact: The early release of the inmates had an impact in the life of them and their families; Later the police officers and correctional officials had to locate them and try to discern whether it was necessary for them to return to prison.

Flaw: Glitch in “good time” credits given to prisoners.

Source:

<http://edition.cnn.com/2015/12/22/us/washington-state-inmates-early-release-sentencing-errors/index.html>

<http://www.bbc.com/news/technology-35167191>

¹good time credits – Credits awarded to inmates for good behaviour.

²DoC - Department of Corrections is a governmental agency responsible for overseeing the incarceration of persons convicted of crimes within a particular jurisdiction.



Traditionally the Department of Corrections is responsible for handling all aspects of State executions.



2. Amazon 1p Glitch

Two weeks before 2014's Christmas, customers could not believe their eyes, as they saw the price of thousands of items going down to 1p on retailing website Amazon.

This was caused due to a technical glitch in the software developed by RepricerExpress that lasted for about an hour. This software automatically reprices items of stock if a cheaper

version becomes available online. Although customers were in for a treat, many small businesses walked into the new year with the wrong foot as they were faced with heavy losses and some even the very real possibility of bankruptcy.

According to the Guardian, a seller on the website went as far as saying the glitch could have cost him more than £100,000. This seller was warned by a competitor that all his listings were on for a penny and was, therefore, able to take his store offline. At that point, £30,000 worth of merchandise had already been ordered by customers.

Amazon has since said that the response to this incident was fast, as they were able to cancel most of orders placed on these affected items, without any penalty to the sellers.



Some buyers surely couldn't believe their eyes.
Or their luck.

Sky News screenshot by Chris Matyszczyk/CNET

amazon.com

Important Dates: December 2014 – Day of glitch.



Consequences: Prices of thousands of products crashed to 1p.

Impact: Poor ratings for sellers who cancelled their shipments; Thousands of dollars lost in merch, including up to the point of bankruptcy; Loss in trust on Amazon; Up to this day, users still attempt to explore glitches on the website to get products for much cheaper or even for free.

Flaw: Glitch in software by RepricerExpress

Videos: Retail Expert on Amazon's Glitch:

<https://www.youtube.com/watch?v=9ieRPtDxKBk>

Purchase Made During Glitch:

https://www.youtube.com/watch?v=Ihmq_paRhwY

Source:

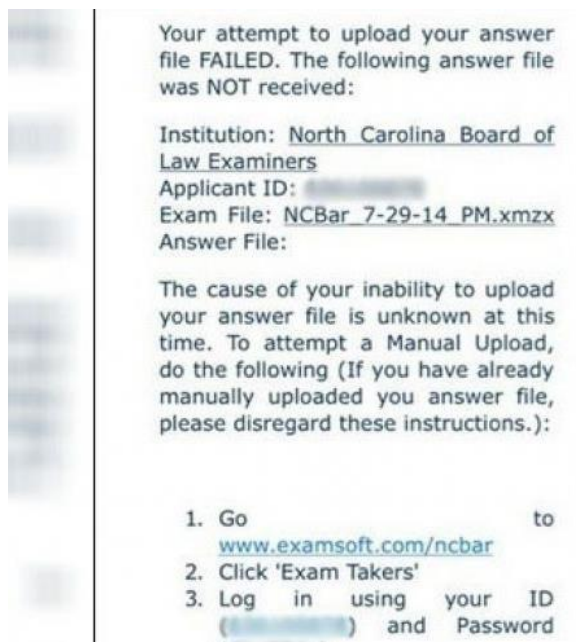
<https://www.theguardian.com/money/2014/dec/14/amazon-glitch-prices-penny-repricerexpress>

<http://www.cityam.com/assets/uploads/content/2016/09/amazon-1p-57edacd31a986.jpg>

<http://www.telegraph.co.uk/finance/newsbysector/retailandconsumer/11292999/Amazon-glitch-leads-to-rush-over-1p-bargains.html>



3. Bar Exam Failure



In July of 2014, thousands of law students were not able to submit their answers to the bar exam due to a software glitch, which resulted in many lawsuits against ExamSoft Worldwide Inc., responsible for the software students were using at the time.

“On the long list of things about which exam takers should be worried, wondering whether they will be able to turn in their

exams for grading should be at the very bottom”, according to a lawsuit filed in Washington state. “It is hard to imagine anything more basic in an exam than being able to turn it in for grading.”

As it turns out, students who weren’t able to have their answers submitted in due time, although it was the software’s fault, failed the bar exam.

To this day, the company has not told the public why the incident happened and refuses to let more on. Also, 90\$ refunds were made on the purchased software, which cost between 125\$ and 150\$.

After this problem occurred, the company has improved its technology and communication with its users, setting up a new website with better information about the exam.



Important Dates: July of 2014.



Consequences: Answers to bar exam not submitted.

Impact: Students failing the bar exam or obtaining lower grades; Several lawsuits.

Flaw: not disclosed, but company assures it had nothing to do with the large amount of people using the software at the same time, as was initially assumed, since, in previous years, there had been many more exam takers at once.

Source:

<http://www.mercurynews.com/2014/08/12/bar-exam-software-failure-sets-off-wave-of-lawsuits/>

<https://www.nytimes.com/2015/05/21/business/dealbook/after-glitches-in-test-law-graduates-who-took-bar-exam-offered-a-small-consolation.html?mcubz=1>

<http://www.foxnews.com/us/2014/07/30/bar-exam-fail-law-grads-unable-to-submit-completed-tests-due-to-software-glitch.html>



4. “No Silver Bullet” Reading Notes

Software construction implies both essential and accidental tasks. The former meaning “the fashioning of the complex conceptual structures that compose the abstract software entity” and the latter “the representation of these abstract entities in programming languages and the mapping of these onto machine languages within space and speed constraints”.

Accidental tasks’ hardships have been significantly diminished by the development of high-level languages, the creation of the time-sharing concept (preserving immediacy and allowing us to keep an overview of complexity) and unified programming environments. Therefore, the main focus becomes dealing with essential tasks.

In the same manner that ancient folklore considered silver bullets the solution to the danger of werewolves, we hope for a straightforward answer to nowadays beast – software projects, apparently innocent, but capable of turning into a monster of missed schedules, overrun budgets and flawed products.

Try as we might, “There is no single development, in either technology or management technique, which by itself promises even one order-of-magnitude improvement within a decade in productivity, in reliability, in simplicity” as software’s nature itself proves to be inconsistent with the possibility of startling breakthroughs.

The problems consist of the fact that:

- “Software entities are more complex for their size than perhaps any other human construct”, given that no two parts are alike and, if they are similar, they, in turn, get made into subroutines, open or closed.

Scaling up a software entity is no mere repetition of elements in larger size, but the interaction between elements in nonlinear fashion, increasing complexity much more.

So, surge the difficulties of team communication, consequent flawed products, blown budgets and schedule delays. There is also unreliability due to all possible states of the program, a struggle with invoking complex functions and the dawning possibility of ripple effects when adding new functions to the program. Conceptual integrity is but an illusion.

- Much of the complexity we “must master is arbitrary complexity, forced without rhyme or reason by the many human institutions and systems to which” our interfaces must conform.



- The pressures for change as software is infinitely malleable and, by comparison to the call-back of a car or the change of a building, cheaper to work with.
- Software is not tangible by definition. As many efforts as we put into creating a visual layout to describe all that is the software behind a project, it is simply impossible to get an overview of what that software is and comprehends.

As a result, the mind is deprived “of some of its most powerful conceptual tools” and, if the process of designing within it is hindered, let alone the communication of the design among minds.

Hopes for a silver bullet are (a)advances in high-level languages, like Ada, which maintains the philosophy of modularization, abstract data types and hierarchical structuring, (b)object-oriented programming, (c)artificial intelligence, which is a confusing topic as, if it is developed by someone, is, inherently, no longer what it was proposed to be – “once we see how the program works and understand the problem, we will not think of it as AI anymore”, (d)expert systems, based on a generalized inference engine and a rule base, although these are axiomatically dependent of an expert and, as a consequence, serve only “to put at the service of the inexperienced programmer the experience and accumulated wisdom of the best programmers”, (e)“automatic” programming, (f)graphical programming, widely critiqued by the author as something that is not embedded in space surely cannot be expected to be represented in two dimensions, (g)program verification, which is more laborious than fruitful, (h)better programming environments and tools, but they can’t promise much more than freedom from syntactic or simple semantic errors and, at last, (i)improved workstations. Nevertheless, the “composition and editing of programs and documents is fully supported by today’s speeds”.

Our conclusion is that the author’s expectations lay low as far as most of these go. The exception is an Ada like language, which he predicts, once its effectiveness and philosophy is assessed, it will make the most difference. Despite these low expectations, the author considers that there are some “attacks” that address the essence of the software problem:

- The exploration of the market to prevent spending time working on what can already be bought and is well documented and maintained;
- The use of “rapid prototyping as a part of planned iteration in establishing software requirements”;
- The organic software growth, progressively adding more functions to systems as needed, by running, using and testing;



- The search for the great conceptual designers of the new generation, which “produce structures that are faster, smaller, simpler, cleaner, and produced with less effort”.