TABLE OF CONTENTS

[1 INTRODUCTION 2](#_Toc510486337)

[2 AT&T NETWORK OUTAGE – JANUARY 15TH 1990 3](#_Toc510486338)

[3 ARIANE 5 FLIGHT 501 – JUNE 4TH 1996 6](#_Toc510486339)

[4 NOVEMBER 2000 – NATIONAL CANCER INSTITUTE, PANAMA CITY 8](#_Toc510486340)

[5 CONCLUSION 11](#_Toc510486341)

[REFERENCES 12](#_Toc510486342)

# 1 INTRODUCTION

When starting a project of any area, there are a few phases to manage the process of it. Each and all of these phases are individually important and should never be neglected.

The mentioned phases for an Information Systems project are: Initiation, Planning, Execution, Monitoring and Controlling, and Closing the project. On the Initiation phase, the focus of the project will be analysed, as well as define the mission, the objectives and strategies of the organization. When planning, evaluate the feasibility factors like economic, technical, operational, schedule, legal and contractual and political. Assess the technical project risks involving the project size, project structure, development group and the user group. Then the project is executed, following the project plan. The project performance is measured against the project plan on the Control phase. And finally, the project is closed when the final paper work is completed and signed off by all stakeholders (Fuller, et all., 2008)

Still according to Fuller et all (2008), there are many factors that can bring success of any IS project such as receiving active support of top management, having the best people on the implementation team, well timeline management, involving users in the process, capturing the user requirements right the first time and the use of the appropriate methodology.

On the other hand, there are also several reasons for a IS project to fail. Bocij, et all., (2008) outputs that from 1500 IT projects across the UK in all industrial sectors, 84% failed to hit their targets on budget, schedule and scope, 45% failed to complete on time and 54% failed to deliver on the planned- for functionality.

Lyytinen and Hirsheim (1987) researched the reasons of project failing, they are as follow: technical failure from poor technical quality; data failure from poor data design, processing errors and poor data management, poor user procedures and poor data quality control at the input stage; user failure that could involve an unwillingness to train staff or user management failure to allow their staff full involvement in the systems development process; organizational failure; failure in the business environment.

In the following sections, a few failed IS will be analysed to the cause and to the consequences of its failure.

# 2 AT&T NETWORK OUTAGE – JANUARY 15TH 1990

AT&T is a worldwide company leader in communications, media and entertainment, and technology. As mentioned on its website, in 2017, the company’s consolidated revenue was more than $160.5 billion. However, in 1990 there was a huge collapse on its system. The company who by the time had built a reputation and a large advertising campaign base on its reliability and security suffered with a bug on the software causing several problems for 9 hours.

According to Dennis Burke (1995), at 2:25pm on Monday, January 15th, an alarming number of red warning signals from all the different parts of their world-wide network began to be noticed. That caused almost 50% of the calls placed through AT&T to fail to go through.

Larry Seese, AT&T’s director of technology development at the time, said “The fault was in the code” (Neumann, P., 1990). Firstly let’s understand how the software worked. The whole system was based on a 114 computer-operated electronic switches (4ESS) spread across the United States. Each switch, which was capable of manage up to 700,000 calls per hour, were linked via a cascading network called Common Channel Signalling System No.7. A switch would verify a list of 14 different possible routes to complete a call, when a telephone call was received by the network. While the telephone number was passed to a parallel signalling network, alternate routes were checked to ascertain if the destination switch was able to deliver the call to its local company. If the destination switch was busy, a busy signal was sent from the original switch and the line was released. If it was available, a reservation would be made by a computer at the destination switch and ordered it to pass the call. This whole process took four to six seconds (Burke, D., 1995).

A team of 100 technicians identified that the problem started in New York City, while the New York switch had performed a routine self-test that indicated it was reaching its limits. As a measure of maintenance, the switch performed a reset operation and sent a message as a “congestion signal” over all the 114 switches that it would take no more calls until further notice. This reset process last only four seconds and when it finished, it dispersed a signal that it had had backed up during the off-line time. Thus, a cascade effect started when another switch received a message that a call from New York was on its way and began to update its records to show that the New York switch was back on line. But a second message from the New York switch arrived and because the first message had not yet been handled, the second message should have been saved until later. A software defect made the second message to be written and the software in the receiving switch detected the overwrite and activated a backup link while it reset itself, but “another pair of closely timed messages triggered the same response in the backup processor causing it to shut down also”. The problem repeated throughout the 114 switches within the network, which blocked over 50 million calls in the nine hours it took to stabilize the system (Burke, D., 1995).

"We made an improvement in the way we react to those messages so we can react more quickly. The first common channel signalling system 7 initial address message (caused by a call attempt) that switch B receives from switch A alerts B that A is back in service. Switch B then resets its internal logic to indicate that A is back in service," said Seese. This improvement is referred to an upgrade on the software, made months before, to speed the processing of certain types of messages.

A pseudocode by Burke (1995), show how the code is read.

|  |  |
| --- | --- |
|  |  |

The defect was a C program that featured a *break* statement within an if clause, nested with a switch clause. On line 7, if the switch is not empty, the program should have dropped out the if clause, processed the incoming message, and set up the pointer to the database on line 11. However, because of the break statement in the else clause on line 10, the program dropped out the case statement entirely and began doing optional parameter work on line 13.

The upgraded code had been tested rigorously, but the biggest problem with this bug, besides the one-line feature, was the use of C programs and compilers. Kao, et all., (1994) says that complete networks (C program) may fail, thus they recommend a more aggressive approach by having a fully independent alternative.

# 3 ARIANE 5 FLIGHT 501 – JUNE 4TH 1996

CNES’s 1996 were marked with a terrible failure on the launching of the maiden flight of the Ariane 5. About 40 seconds after the ignition of the flight, at an altitude of approximately 3700m, the launcher changed its path, broke up and exploded.

According to Lions, (1996) who did a report by the Inquiry Board, the weather was acceptable for launching on the morning of 4th June 1996 and there was no obstacle to the transfer of the launcher to the launch pad. There was only one uncertainty about the fulfilment of the flight was the visibility criteria.

The rocket performed an absolutely normal flight until approximately 37 seconds after it was lift-off. It presented failure of the back-up Inertial Reference System which was followed by failure of the active Inertial Reference System (SRI). This failure caused the launcher to veer abruptly and activate the self-destruction procedure (Lions, 1996).

The Flight Control System of the Ariane 5 was of a standard design, which means they reutilized the same software that was used on Ariane 4. After the analysis of the failure, it was found that what caused the chain of technical events was an exception during the execution of a data conversion from 64-bit floating point to 16-bit signed integer value, which means that the floating point number that was converted had a greater value than what could be represented by a 16-bit signed integer causing an overflow. This resulted in an Operand Error.

There was redundancy at equipment level, regarding two SRIs and two On-Board Computers (OBC), in order to improve reliability. If the OBC detects that the active SRI has failed, it immediately switches to the “hot” stand-by SRI, provided that this unit is functioning properly. A number of other units in the Flight Control System were also duplicated. When the problem occurred, the OBC could not switch to the back-up SRI because it had already ceased to function during previous data cycle (72 miliseconds period) for the same reason as SRI 2.

“The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose. (Lions, 1996)”. This alignment function; which was a requirement for Ariane 4 but was not required for Ariane 5; consisted in being operative for 50 seconds after starting of the Flight Mode of the SRIs, consequently, when lift-off occurred, the function continued for approximately 40 seconds. “The Operand Error occurred due to an unexpected high value of an internal alignment function result called BH, Horizontal Bias, related to the horizontal velocity sensed by the platform. This value is calculated as an indicator for alignment precision over time. The value of BH was much higher than expected because the early part of the trajectory of Ariane 5 differs from that of Ariane 4 and results in considerably higher horizontal velocity values. (Lions, 1996)”.

Some variables of the codes were protected for this type of conversion after analysis of every operation involving seven variables were at risk of leading to an Operand Error; but not this one and other two, because a maximum workload target of 80% had been set for the SRI computer. There was no reference to justification of the decision to not protect these variables in the source code. “The reason for the three remaining variables, including the one denoting horizontal bias, being unprotected was that further reasoning indicated that they were either physically limited or that there was a large margin of safety (Lions, 1996)”. In the case, the decision turned out to be the faulty. This decision to protect some of the variables and to not protect others was taken by the project partners at several contractual levels jointly.

It took 10 years and $7 billion to produce Ariane 5, which was capable of throwing a pair of three-ton satellites into orbit giving Europe great supremacy in the commercial space business (Gleick, 1996). Even though the program was tested and analysed, it was done wrongly. It started with the decision of reutilize the software from Ariane 4, which was a good software at the time, but neglected on Ariane 5 because it had a higher velocity causing a stack overflow on the system. The destroyed rocket and its cargo had a value of $500 million (Arnold, 2000). In addition to the material loss, there were also deaths from the crew on the failed launch.

# 4 NOVEMBER 2000 – NATIONAL CANCER INSTITUTE, PANAMA CITY

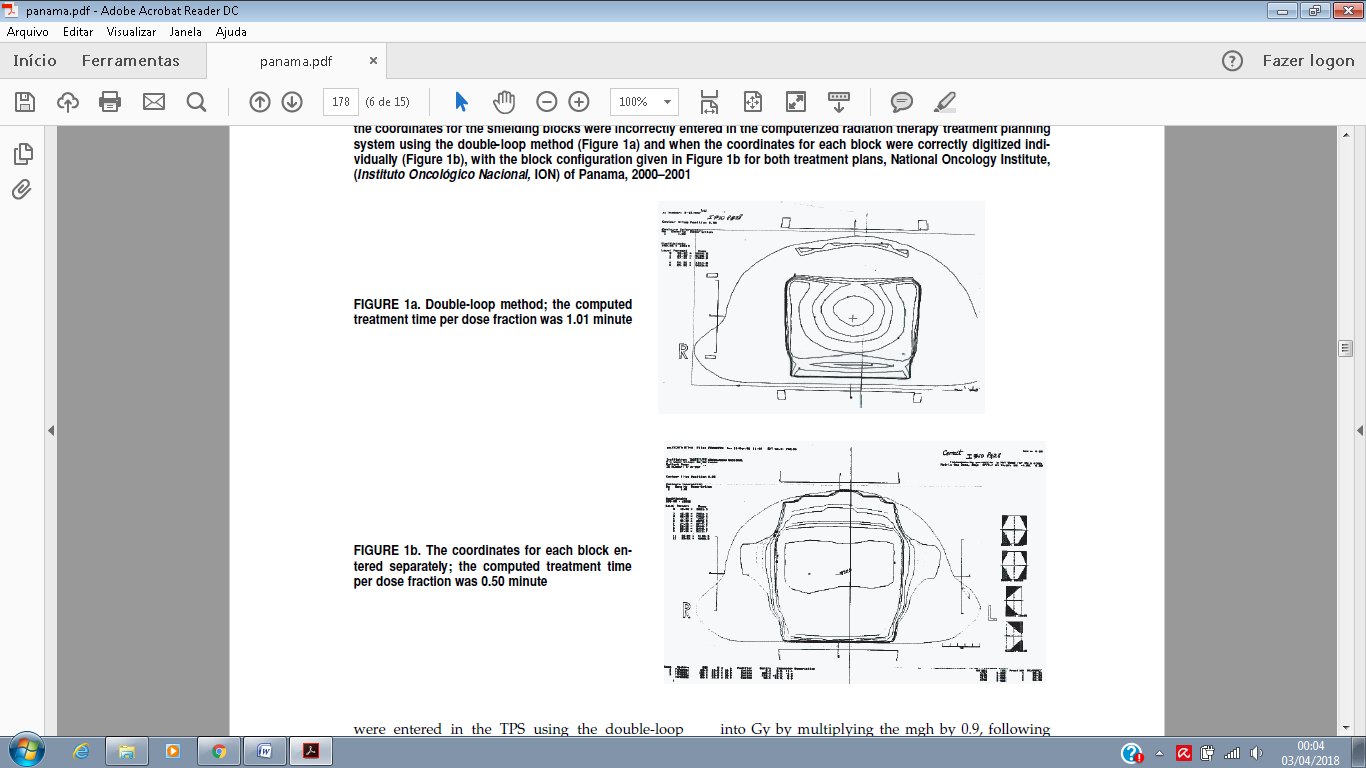
In late 2000 and early 2001, the National Oncology Institute (Instituto Oncológico Nacional, ION) in Panama, suffered with a software failure which caused the death of 23 victims and other 5 victims to critical health conditions. The victims were overexposed to radiation of the cobalt-60 unit for cancer treatment.

In March 2001, the director of ION was notified of the serious overreactions in patients who were taking radiation therapy to treat cancer. His investigation used information on radiation therapy treatment techniques and data from copies of patient charts which was provided by the clinical and medical physics staff at the ION.

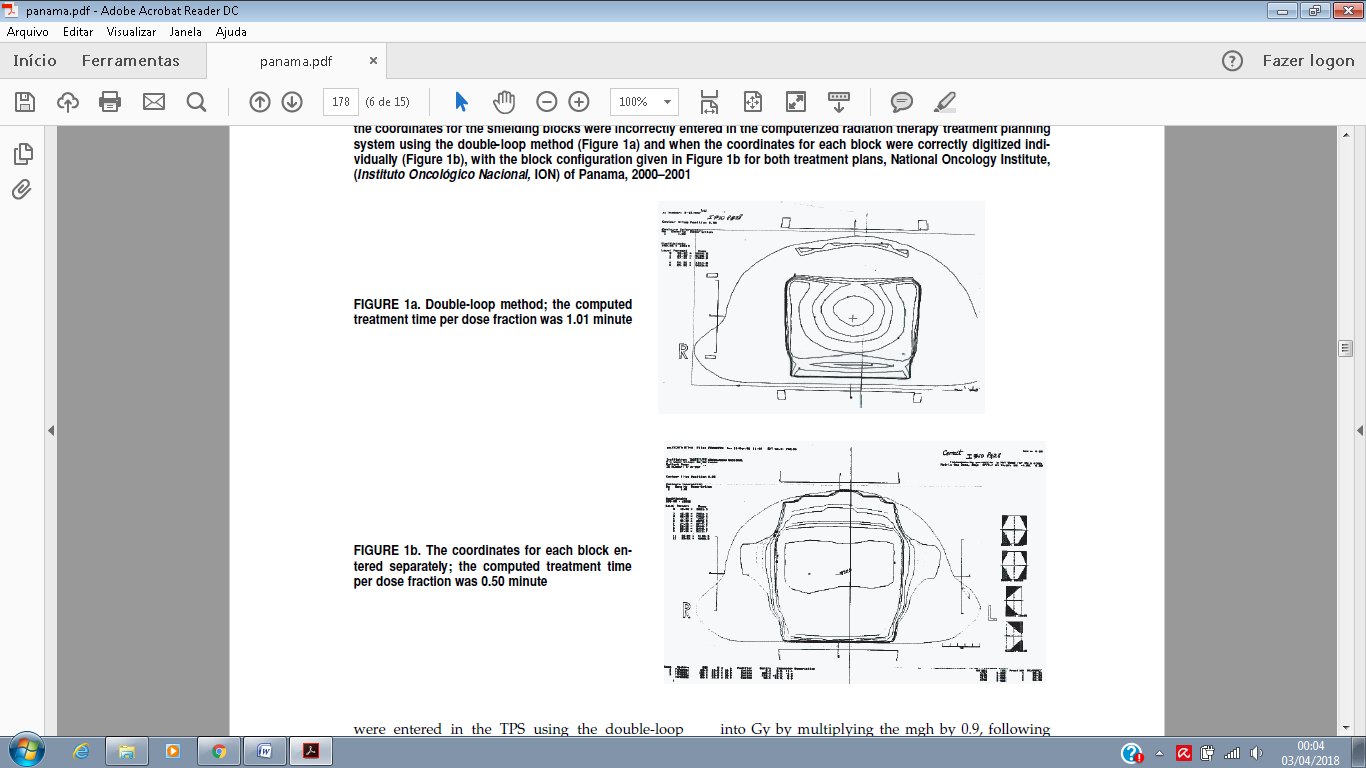
The dosimetry calculations performed by the ION medical physicists were done with a computerized radiation therapy treatment planning system (TPS) manufactured by Multidata Systems International Corporation (Saint Louis, Missouri, United States of America), with the isodose distributions and treatment times being generated using the “External Beam” Version 2.1.1 software for that TPS (Borrás, 2006).

Experts from M.D Anderson Cancer Center, in Houston, were invited to help investigate the problem. They found out that the algorithm used in the TPS software gave treatment times differing by a factor of about two. According to the TPS manual, the instructions to be entered in the system were the coordinates of the perimeter of the unshielded field and then the software would digitize one block at a time, to a maximum of four blocks. However, this process took a lot of time, so some of the radiation oncologists at the ION wanted to treat some cervix cancer patients with five blocks. Thus, those physicists found a way around the Multidata TPS software block entry limitations by digitizing all the blocks in a continuous way, as if it were a single block. After done that, the computer program was performing a double loop when the outside and inside perimeters of the field were entered into the system, which made the computed treatment times double dose those obtained when the outside perimeter was digitized in one direction and the inside perimeter in the reverse direction. “The multidata system did not alert the user that an improper data sequence had been entered (Borrás, 2006)”. This mistake caused an error that ranged from 10 to more 100% more the recommended dose of radiation for those cases.

The figure 1 shows a treatment for a patient with rectum four-field treatment plan using the double-loop method; the computed treatment time per dose fraction was 1.01 minute.



The figure 2 shows the same type of treatment but with the coordinates for each block entered separately; the computed treatment time per dose fraction was 0.50 minute.



When comparing the two treatments, we can clearly see the difference of the shape of the isodoses generated by the double-loop method and the required dose. However, the physicists did not notice the difference; otherwise they would have uncovered the problem with the TPS sooner.

Three Panamanian medical physicists went on trial on charges of second-degree murder. They may be held responsible for input changes into the software that led directly to the death of 21 patients and other with serious health problems, even though they thought they were helping patients. Multidata Systems International, the St. Louis Company, faced off charges that could damage or maybe destroy the company itself if the company is found guilty. The company would be forced to pay damages sought by the victims, which could vary $14 million to $28 million. But the company itself had about $2 million in annual sales and about 15 employees (McCormick, 2004).

# 5 CONCLUSION

After analysing the three projects that failed, we can clearly see that skipping or ignoring one of the phases can led to serious problems. Sometimes, the errors can be minimum, like a single like on the code which can generate a bug that will cause serious problems for company.

On the case of AT&T for example we could see that the whole project was tested properly, but the project was not analysed correctly, otherwise they would have used a different type of program. The C program used on the project had the same features for all the 114 switches causing a cascade effect of the problem, whether if they had used an independent program the problem would be contained into one single switch, that is if we consider that they have not found the code error, which had also an important role on the bug.

On the Ariane 5 failure, the analysis process also failed especially when they decided to copy the same code of a previous, successful project without any changes or analysis of the previous project, when clearly a new project would have features that an old one would not have, like the velocity. This is a parameter that should definitely be considered. Like said previously, the project was tested, but some of the code could not being tested due to impossibility to recreate the real life situation without lifting-off.

On the National Cancer Intitute case, the project failed due to people (user failure), when the doctors decided to input information that was not supposed to be input with the objective of accelerate the process without even testing previously and properly.

Some projects can fail with only material loss like the AT&T failure, but others can also involve the life of thirds, for example on the Ariane 5 and the Panama failure, where people actually died.

# REFERENCES

Arnold, D. (2000). *The Explosion of the Ariane 5*. [online] Www-users.math.umn.edu. Available at: http://www-users.math.umn.edu/~arnold/disasters/ariane.html [Accessed 2 Apr. 2018].

Bocij, P., Greasley, A. and Hickie, S. (2008). *Business information systems*. 4th ed. Harlow [u.a.]: Pearson.

Borrás, C. (2006). Overexposure of radiation therapy patients in Panama: problem recognition and follow-up measures. *Revista Panamericana de Salud Pública*, 20(2-3).

Burke, D. (1995). *All Circuits are Busy Now: The 1990 AT&T Long Distance Network Collapse*. [online] Users.csc.calpoly.edu. Available at: http://users.csc.calpoly.edu/~jdalbey/SWE/Papers/att\_collapse.html [Accessed 1 Apr. 2018].

Fuller, M., George, J. and Valacich, J. (2008). *Information systems project management*. 1st ed. Upper Saddle River, NJ: Pearson Prentice Hall.

Gleick, J. (1996). *A Bug and a Crash by James Gleick*. [online] Around.com. Available at: https://around.com/ariane.html [Accessed 2 Apr. 2018].

Kao, B., Garcia-Molina, H. and Barbara, D. (1994). Aggressive transmissions of short messages over redundant paths. *IEEE Transactions on Parallel and Distributed Systems*, [online] 5(1), pp.102-109. Available at: http://i.stanford.edu/pub/cstr/reports/cs/tr/92/1431/CS-TR-92-1431.pdf [Accessed 1 Apr. 2018].

Lions, J. (1996). *ARIANE 5 Failure - Full Report*. [online] Sunnyday.mit.edu. Available at: http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html [Accessed 2 Apr. 2018].

Lyytinen and Hirsheim (1987). Information Systems Failures – a survey and classification of the empirical literature. In: Bocij, P., Greasley, A. and Hickie, S. (2008). *Business information systems*. 4th ed. Harlow [u.a.]: Pearson.

McCormick, J. (2004). *'We Did Nothing Wrong'*. [online] Baselinemag.com. Available at: http://www.baselinemag.com/c/a/Projects-Processes/We-Did-Nothing-Wrong [Accessed 2 Apr. 2018].

Neumann, P. (1990). *Telephone World - The Crash of the AT&T Network (1990)*. [online] Phworld.org. Available at: http://phworld.org/history/attcrash.htm [Accessed 1 Apr. 2018].

Skrzycki, C., Phillips, D., Skrzycki, C. and Phillips, D. (1991). *HUMAN ERROR LED TO OUTAGE, ATT SAYS*. [online] Washington Post. Available at: https://www.washingtonpost.com/archive/politics/1991/09/19/human-error-led-to-outage-att-says/546c404b-e957-4068-b2dd-4a920f67ae6a/?utm\_term=.0c9c156ec02d [Accessed 1 Apr. 2018].