

Software engineering

Profession



INTRODUCTION

The goal of this paper is to analyze about three major software projects namely

- The London Ambulance System
- The Virtual Case File
- The Automatic Baggage System

By analyzing these software projects and the software engineering principles followed, the key factors responsible for the software projects failure can be understood. Each of these projects has failed miserably as they didn't follow proper software engineering principles. In this term paper the following projects have been studied and reason for their failures are identified.

Finally there is a comparison of all the three software projects studied. The methodology followed in writing this term paper is reading the following reference materials available in the internet and extracting the key points for the failures of the software projects. The papers referenced for writing the following term paper are 1. H. Goldstein. Who Killed the Virtual Case File? IEEE Spectrum, Sept. 2005, pp. 24-35. 2. Statement of Glenn A. Fine, Inspector General, US Dept. of Justice, 27 July 2005. 3. A.

Finkelstein and J. Dowell. A Comedy of Errors: the London Ambulance Service Case Study. 4. Report of the Inquiry into the London Ambulance Service (February 1993), by A. Finkelstein, 5. Richard de Neufville. "The Baggage System at Denver: Prospects and Lessons," Journal of Air 6. Barry Shore. "Systematic Biases and Culture in Project Failures," Project Management Journal

CONCLUSION: The conclusion after studying these three

papers, for any software projects the good principles of software engineering should be followed. The software development process should be properly planned with achievable and realistic deadlines. All the three projects had poor planning with unrealistic deadlines. • Great importance should be given to the requirements gathering phase and it should not be changed during the middle of the development • Developers should develop the projects with proper coding standards so that there is no issue during the integration of different modules. • Time critical projects should require critical and solid reasoning as well as good anticipation of problems and perform risk management. The schedule of the software projects should have good portion of time in testing the software product developed. • Finally, as far as possible keep the complexity of the system to manageable levels and tested effectively. LONDON AMBULANCE SYSTEM In October 1992 the Computer Aided Despatch (CAD) system developed by Systems Options was deployed for the London Ambulance System (LAS). The goal of the software system was to automate the process of the ambulance service for the London Ambulance System (LAS) in the city of London, United Kingdom.

The implemented project was a major failure due to variety of factors. The Each component of good state of the art has been ignored, each guideline of the Software engineering has ignored by the management and authorities' neglected basic management principles. The working of the LAS can be summarized as: the system gets request by phone calls and sends ambulance based on nature, availability of resources. The automatic vehicle locating system (AVLS) and mobile data terminals (MDT) was used to perform automatic communication with ambulances.

Some of the major reasons for the failure of the London ambulance system can be stated as:

- The deadline given for the completion of the project was six months. The project of such big magnitude cannot be completed within a small deadline.
- The software was not fully developed and incomplete. The individual modules were tested, but the software was not tested fully as a integrated system.
- The resilience of the hardware under a full load condition had not been tested before the deployment of the software. The flash cut over strategy was used to implement the system which was a high risk and moreover it didn't have any backup systems to revert on failure.

Inappropriate and unjustified assumptions were made during the specification process of the project. Some of the few assumptions that were made are : Complete accuracy and reliability of the hardware system. Perfect location and status information. Cooperation of all operators and ambulance crew members.

Lack of consultation with the prospective users of the system and subject matter experts. The Software requirement specification was excessively prescriptive, incomplete and not formally signed off. The London Ambulance system underestimated the difficulties involved in the project during the project blastoff phase. Inadequate staff training. The crew members were not fully trained on the operation of the new software and their prior experience was not used in the newly developed software. The Report of the Inquiry into the London Ambulance Service by Anthony Finkelstein also gives us more <https://assignbuster.com/software-engineering/>

information about the failure of the system. Some of the are listed below as follows: It states that “ the CAD system implemented in 1992 was over ambitious and was developed and implemented against an impossible timetable”. In addition, the LAS Committee got the wrong impression, that the software contractor had prior experience in emergency systems; this was misleading in awarding the contract to systems options. Project management throughout the development and implementation process was inadequate and at times ambiguous. A major project like this requires a full time, professional, experience project management which was lacking. The computer system did not fail in a technical sense, the increase in calls on October 26 and 27 1992 was due to unidentified duplicate calls and call backs from the public in response to ambulance delays. “ On 4th November 1992 the system did fail. This was caused by a minor programming error that caused the system to crash”.

VIRTUAL CASE FILE SYSTEM

The primary goal of the Virtual case file (VCF) system was to automate the process of FBI paper based workenvironment, allow agents and intelligence analysts to share vital investigative information, and replace the obsolete Automated Case Support (ACS) system.

In ACS tremendous time is spend in processing paperwork, faxing and Fedexing standardized memo. Virtual case file (VCF) system was aimed at centralizing the IT operations and removes the redundancy present in various databases across the FBI system. In September 2000 the FBI Informationtechnologyupgrade project was underway. It was divided into three parts.

- The Information Presentation Component
- The Transportation Network Component
- User Application Component

The first part involved distribution of new Dell computers, scanners, printers and servers.

The second part would provide secure wide area networks, allowing agents to share information with their supervisors and each other. The third part is the virtual case file. The Virtual Case File system project was awarded to a US government contractor, ScienceApplications International Corporation (SAIC). The FBI used cost plus - award fee contracts. This project was of great importance because the FBI lacked the ability to know what it knew; there was no effective mechanism for capturing or sharing its institutional knowledge. This project was initially led by former IBM Executive Bob E. Dies. On 3th December 2003, SAIC delivered the VCF to FBI, only to have it declared dead on arrival. The major reasons for the failure of the VCF system can be summarized as: The project lacked clearly defined schedules and proper deadlines, there was no formal project schedules outlined for the project and poor communication between development teams that was dividing into eight teams to speed up the project completion. The software engineering principle of reusing the existing components was ignored. SAIC was developing a E - mail like system even though FBI was already using an off - the - shelf software package. The deployment strategy followed in implementing the system was flash -cutover. It is a risky way a deploying a system as the system would be changed in a single shot. The project violated the first rule of software planning of keeping it simple. The

requirement document was so exhaustive that rather of describing the function what it should perform it also stated how the functions should be implemented. Developers coded the module to make individuals features work but were not concerned about the integration of the whole system together.

There was no coding standards followed and hence there was difficulty in the integration process. The design requirement were poorly designed and kept on constantly changing through the development phase. The high level documents including the system architecture and system requirements were neither complete nor consistent. Lack of plan to guide hardware purchases, network deployments, and software development. Appointment of person with no prior experience in management to manage a critical project such as this was grave mistake, appointment of Depew as VCF project manager. Project lacked transparency in the work within the SAIC and between SAIC and the FBI. Infrastructure including both the hardware and network was not in place to test thoroughly the developed virtual case file system by SAIC which was essentially needed for flash cut off deployment. The requirement and design documentation were incomplete, imprecise, requirement and design tracings have gaps and the maintenance of software was costlier. According to the report by Harry Goldstein, " there was 17 ' functional deficiencies' in the deployed Virtual Case File System".

It didn't have the ability to search for individuals by specialty and job title. All these above factors contributed to the failure of the Virtual Case File System which wasted a lot of public tax payers' money.

AUTOMATIC BAGGAGE SYSTEM

The automatic baggage system designed for the Denver International Airport is a classic example of a software failure system in the 1990's. With a greater airport capacity, the city of Denver wanted to construct the state of art automatic baggage handling system. Covering a land area of 140 square kilometer the Denver airport has 88 airport gates with 3 concourses.

The fully automated baggage system was unique in its complexity because of the massive size of the airport and its novel technology. The three other airports that have such systems are the San Francisco International Airport, International airport in Frankfurt and the Franz Joseph Strauss Airport in Munich. This project is far more complex than any other projects, because it has 12 times as many carts as in exiting comparable system . The contract for this automatic baggage system was given to BAE automated systems. In 1995 after many delays, the baggage system project was deployed, which was a major failure.

The baggage carts derailed, luggage was torn and the system completely failed. But the system was redesigned with lesser complexity and opened 16 months later.

GOALS OF THE PROJECT

The system calls for replacing the traditional slow conveyor belts with telecars that roll freely on underground tracks. It was designed to carry up to 70 bags per minute to and from baggage check-in and checkout at speed up to 24 miles/hour. This would allow the airlines to receive checked baggage at their aircraft within 20 minutes. The automatic baggage system was a critical

because the aircraft turnaround time was to be reduced to as little as 30 minutes.

The faster turnaround time meant more quickly the operations and it increases the productivity. The installers are quoted as having planned “ a design that will allow baggage to be transported anywhere within the terminal within 10 minutes”. PROJECT SCOPE: The International airport at Denver three concourses and initially it aimed at automating all the three concourses. But later the concourse B was alone designed to be made automatic. The project was later redefined to handle only outbound baggage. It does not deal with the transfer of bags.

STAKE HOLDERS

The major stake holders in the project can be identified as: • The Denver International Airport Management. • The BAE Automated Systems. • The Airline Management. The project blastoff according to Robertson & Robertson states that during this phase it has to identify all the stakeholders and ask their inputs for the requirements. In the ABS System the Airline Management was not made to involve in the blastoff meetings to provide their inputs and excluded from the discussions. As well as the risk should be analyzed properly during the blast off which was also a draw back in this system.

This was a perfect example of failure to perform risk management. The cost estimation of the project was incorrect as it exceeded the estimated cost during the development. So, Aspects in which the project blastoffs were not addressed can be summarized as follows:

- The underestimation of complexity
- Poor stakeholder management
- Poor Design
- Failure to perform risk management

There were only three “intense” working sessions to discuss the scope of the project and the agreement between the airport management and BAE automated systems.

Although BAE automated systems had been working in the construction of the baggage system in concourse B for United Airlines, the three working sessions are not sufficient to collect all the requirements for the construction of the automated baggage systems. This shows clearly a poor software engineering principle because requirements are the key base factors for the project to be built upon. Reports indicate that the two-year deadline for the construction of the automatic baggage system is inadequate. The reports that showed that the project required more than two years are as follows: “The complexity was too high for the system to be built successfully” by The Baggage System at Denver: Prospects and Lesson – Dr. R. de Neufville *Journal of Air Transport Management*, Vol. 1, No. 4, Dec, pp. 229-236, 1994. None of the bidders quoted to finish the project within two years. Experts from Munich airport advised that a much simpler system had taken two full years to complete and it was system tested thoroughly six months before the opening of the Munich airport. Despite all this information the decision to continue with a project was not based on the sound engineering principles.

ABS REQUIREMENT DESIGN AND IMPLEMENTATION

The Automatic Baggage System constructed by the Airport Management was a decision taken two years before the opening of the new Denver International Airport. Initially the concourse B meant for United Airlines was supposed to be constructed by the BAE Automated Systems and all other airlines had to construct their own baggage handling mechanism. Later the responsibility was taken by the Denver Airport Management to construct the Automatic Baggage System.

The integrated nature of the ABS system meant that airport looks after its own facility and has a central control. The BAE plan to construct for the concourse B was expanded to the other three concourses which was a major change in the strategy of the airport construction. Moreover the airport management believed that an automated baggage system would be more cost effective than manual system given the size of the massive airport. During the development phase the requirements kept on changing which added additional complexity to the project. Though in the contract there was clearly statement no change in requirement would be accommodated, they accepted the changes to meet the stakeholder needs. For example the addition of the ski equipment racks and the addition of maintenance track to allow carts to be serviced without being removed from the rails and able to handle oversized baggage. The baggage system and the airport building shared physical space and services such as the electrical supply. Hence the designers of the physical building and the designers of the baggage system needed to work as one integrated team with lot of interdependency.

Since the construction of the airport was started initially the building designers made general allowances in the place where they thought the baggage system would come into place. Hence the designers of the automatic baggage system have to work with the constraints that have already been placed. For example sharp turns were supposed to be made due to the constraints placed and these were one of the major factors for the bags to be ejected from the carts. The design of the automatic baggage system “ Systematic Biases and Culture in Project Failures”, a Project Management Journal is as follows. Luggage was to be first loaded onto the conveyor belts, much as it is in conventional baggage handling system. These conveyors would then deposit the luggage in the carts that were controlled by computers. The luggage would travel at 17 miles per hour to its destinations, as much as one mile away. •The automatic baggage system would include around 4000 baggage carts travelling throughout the airport under the control of 100 computers with processing power up to 1400 bags per minute. However the design with the above architecture failed as it was not able to handle variable load.

It was also suffering from various problems they are identified as: The software was sending carts out at the wrong times, causing jams and in many cases sending carts to the wrong locations. The baggage system continued to unload bags even though they were jammed on the conveyor belt. The fully automated system may never be able to deliver bags consistently within the times and at the capacity originally promised. In another case the bags from the aircraft can only be unloaded and loaded

into the unloading conveyor belt is moving, this belt moves only when there are empty carts.

Empty carts will only arrive after they have deposited previous loads; this is a cascade of queues. Achieving high reliability also depends on the mechanical and the computers that controlled the baggage carts' reliability. Errors may occur during reading or transmitting information about the destinations. There may be various scenarios during which these errors can take place. Some of them are listed as below.

1. The baggage handler may place the bag on the conveyor with the label hidden.
2. The baggage may have two labels on it. one from the previous flight.
3. The labels may be mutilated or dirty.
4. The label may not lie in the direction of the view of the laser reader.
5. The laser may malfunction or the laser guns stop reading the labels.

The reading of information is vital in the automatic baggage system since the whole system is dependent on the information transmitted from reading of the labels and this information must be transmitted by radio to devices on each of the baggage carts. There is no available evidence of effective alternative testing of the capability of the system to provide reliable delivery to all destinations under variable patterns of load.

This variable demand made in the system is famously called as the line balancing problem. That is, it is crucial to control the capacity of the system so that all lines of flow have balanced service. This problem can be avoided by eliminating situations where some lines get little or no service, to avoid

the possibility that some connections simply do not function or in other words control the emptiness. This failure also was because the entire system was developed within a two year deadline and hence the automatic baggage system was not testing completely with variable loads.

Lack of testing also is a major reason for this failure. These all are the major factors that led to the failure of the automatic baggage system in Denver international airport. Subsequently a much less complex system was design and implemented sixteen months later. This newly designed system had the following functionality as follows:

- Serve only one concourse, the concourse B for United Airlines.
- Operate on half the planned capacity on each track.
- Handle only outbound baggage at the start.
- Not deal with transfer bags.

COMPARISON OF ABS, VCF and LAS PROJECTS

All the management teams of the three projects wanted the software system to be built quickly without taking into consideration of the system requirement. Hence all the system had unrealistic deadline to be met. Because of these unrealistic deadlines the system didn't follow proper software engineering standards and principles. In all the three projects during the project blastoff phase the requirements gathering activity was not proper and incomplete, due to which the requirements kept on changing during the development phase. Lack of consultation with the stake holders and prospective users. All the three projects Software requirement specification was excessively prescriptive, incomplete and not formally signed off. All the three systems were not properly tested before deployment

due to lack of time and tight schedules. The timeline was not reasonable for any of the projects. There was poor communication between the developers, customers and the clients in all the projects. The identification of the stakeholders and collecting requirements from the stakeholders and subject matter experts was not proper and incomplete.

ASPECTS | ABS | VCF | LAS | |

DEPLOYMENT STRATEGY | It was deployed in a single phase| Flash Cutover strategy was used in| Flash Cutover strategy was used | | | with a major failure of the | replacing the ACS System | in replacing the existing System | | | system | | |

PROJECT SCHEDULE/DEADLINE | Had a very tight schedule of two | Over ambitious schedule | Had a very tight deadline, two | | | years to implement | | years(1990 - 1992) | |

PROJECT PLANNING | Poor Planning, The system was | Poor Planning and constantly | Good Engineering practice were | | | decided to be developed two years| changing milestones | Ignored | | | before the completion of the | | | | airport | | |

SOFTWARE REQUIREMENT SPECIFICATION | Kept on changing to meet the | Slowly changing design | On the fly code changes and | | | needs of the stakeholders | requirements | requirement changes | |

PROJECT BLASTOFF | There was only three intense | The project blastoff phase didn't | It left out the view of the | | | session to collect the | collect all the requirements | customers and subject matter | | | requirements which is inadequate | properly | experts | |

REUSABILITY | This system didn't have any back | They already had e-mail like | The existing communication | | | up system to reuse | system which could have been | devises in the ambulance system | | | reused but new mail system was | | | | written | | |

CODING/TESTING | The system was not tested with | The software system followed the | Backup dispatch system not tested| | | variable load | spiral developmental model and not| and the overall software

<https://assignbuster.com/software-engineering/>

not | | | | tested as a whole | system tested | | SYSTEM DESIGN | The system design was too complex| The system was not base lined and | The System design was incomplete | | | | kept on changing | | | BUGS | System was unable to detect bugs | 59 issues and sub issues were | 81 Know Bugs in the Deployed | | | | identified | System | | ASSUMPTIONS/ | It was dependent on computers | No major assumptions were made in | Perfect location information and | | DEPENDENCY | that controlled the baggage cars | this project | dependent on the MDT | | | | communications |

PERSONAL REFLECTION

After reading all the three projects I now understand that development of software not necessary has to be coding the software properly but there are various aspects apart from coding like requirement gathering, risk analysis, testing. The requirements gather should plays a vital role in software development and it has to be properly made in consultation with all the stakeholders, customers of the software. Understanding the complexity of the software being developed. Proper planning and schedule of events for the development activities. Deadlines for the software development should be realistic and achievable Use of any of the software engineering models for the development like waterfall model, Bohms' spiral model, incremental work flow model or agile software development. Last but not the least the software developed should be thoroughly tested for finding out flaws in the development and fixing them.

REFERENCES

1. H. Goldstein. Who Killed the Virtual Case File? IEEE Spectrum, Sept. 2005, pp. 24-35.

2. Statement of Glenn A. Fine, Inspector General, US Dept. of Justice, 27 July 2005.
3. A. Finkelstein and J. Dowell. A Comedy of Errors: the London Ambulance Service Case Study. Proc. 8th Int. Workshop on Software Specification and Design (IWSSD96), pp. 2-4, Velen, Germany, 1996.
4. Report of the Inquiry into the London Ambulance Service (February 1993), International Workshop on Software Specification and Design Case Study. Electronic Version Prepared by A. Finkelstein, with kind permission from the Communications Directorate, South West Thames RegionalHealthAuthority.
5. Richard de Neufville. “ The Baggage System at Denver: Prospects and Lessons,” Journal of Air Transport Management, Vol. 1, No. 4, Dec. 1994, pp. 229-236.
6. Barry Shore. “ Systematic Biases and Culture in Project Failures,” Project Management Journal, Vol. 39, No. 4, 2008, pp. 5-16.