

Quality issues in system development 13719

[Technology](#), [Internet](#)



Quality Issues In System Development

The period between the 1970's and 1980's was a time of great advancement in

computer hardware technology which took an industry still in it's infancy, to a

level of much sophistication and which ultimately revolutionised the information

storage and processing needs of every other industry and that of the entire

world. However, it was also during this period when the shortcomings of

implementing such technology became apparent. A significant number of

development projects failed which resulted with disastrous consequences, not

only of an economic nature, but social aswell. Seemingly, although hardware

technology was readily available and ever improving, what was inhibiting the

industry was in the methods of implementing large systems. Consequently, all

kinds of limited approaches materialized that avoided the costs and risks

inherent in big-systems developments.

Times have changed, and with it our understanding and experience as how

<https://assignbuster.com/quality-issues-in-system-development-13719/>

best to develop large systems. Today's large systems yield greater benefits for less cost than those of previous decades. Large systems provide better, more timely information, the ability to integrate and correlate internal and external information, the ability to integrate and facilitate streamlined business processes. Unfortunately, not every system that information workers develop are well implemented; this means that the computer system which was originally intended to make a company more efficient, productive and cost-effective, is in the end doing the exact opposite - namely, wasting time, money and valuable manpower. So even with all the lessons learned from the 70's and 80's, our vastly superior methodologies and knowledge of the 90's is still proving to be fallible, as suggested in the following examples.

System Development Failures

In Britain, 1993, an incident occurred which forced the London

Ambulance Service to abandon its emergency system after it performed

<https://assignbuster.com/quality-issues-in-system-development-13719/>

disastrously on delivery, causing delays in answering calls. An independent inquiry ordered by British government agencies found that the ambulance service

had accepted a suspiciously low bid from a small and inexperienced supplier. The

inquiry report, released in February 1993, determined that the system was far

too small to cope with the data load. For an emergency service, the system error

would not only cause the loss of money, but more essentially, fail to dispatch ambulances correctly and promptly upon the arising of critical situations. Thus,

the implications of such a failure are apparently obvious, both socially and economically. Since the failures, the ambulance service has reverted to a paper-

based system that will remain in place for the foreseeable future.

Another failure was the collapse of the Taurus trading system of the

London Stock Exchange. Taurus would have replaced the shuffling of six sorts of

paper among three places over two weeks - which is how transactions in shares

are settled in London-with a computerized system able to settle trades in three

days. The five-year Taurus development effort, which sources estimated cost hundreds of millions of dollars, was termed a disaster, and the project was abandoned in March 1993. Exchange officials have acknowledged that the failure

put the future of the Exchange in danger.

Why did they fail?

What went wrong with these systems? The real failure in the case of the

London Stock Exchange was managerial, both at the exchange and among member

firms. The exchange's bosses gave the project managers too much rope, allowing

them to fiddle with specifications and bring in too many outside consultants and

computer firms. Its new board, having heavy-weight and diverse membership,

proved too remote from the project. Member firms that spent years griping about

Taurus's cost and delays did not communicate their doubts concerning the project.

The Bank of England, a strong Taurus supporter, failed to ask enough questions,

despite having had to rescue the exchange's earlier attempt to computerize

settlement of the gilts market. According to Meredith , an expert in project

management issues, many system development catastrophes begin with the selection

of a low bidder to do a project, even though most procurement rules state that

cost should be only one of several criteria of designation. The software failure

occurs because the companies involved did not do a risk assessment prior to

starting a project. In addition, many companies do not study the problems

experienced in earlier software development projects, so they cannot apply that

data when implementing new projects.

Another source of problems is the failure to measure the quality of output during the development process. Information workers as yet have not fully understood the relationship that exists between information and development. It is shown that information should be viewed as one of the essential know-how resources. The value and necessity of information for development is argued. An attempt is made to classify the various areas where information is needed for development, as well as the information systems and infrastructures available or required to provide for the different needs. There are a number of reasons why information has not yet played a significant role in development. One reason is that planners, developers and governments do not yet acknowledge the role of information as a basic resource. Another is that the quality of existing

information services is such that they cannot yet make an effective contribution

to information provision for development.

Avoiding development failure

Companies blame their unfinished system projects on such factors as poor technology, excessive budgets, and lack of employee interest. Yet, all these factors can be easily avoided. All that is needed to develop and implement successful systems is a strong corporate commitment and a basic formula which

has proven effective time after time. By following the guidelines below, any system workers can install and implement a successful, efficient system quickly

and with minimal disruption to the workplace. Understand your workplace- every

company must fully understand its existing environment in order to successfully

change it. Define a vision for the future- This objective view will help the

company develop a clear vision of the future. Share the vision- In order for the

system to be successful, all those who are involved in its development must fully buy into the process and end-product. This will also help further define specific goals and expectations. Organize a steering committee-This committee,

which must be headed by the executive who is most affected by the success or

failure of the project, has to be committed and involved throughout all stages.

Develop a plan-The project plan should represent the path to the vision and finely detail the major stages of the project, while still allowing room for refinement along the way. Select a Team of users- A sampling of company employees is important to help create, and then test, the system. In the

Laboratory systems failure case . That means both the vendor and laboratory should identify what users know and what they need to know to get the best out

of the LIS. They must also develop a formal training plan before selecting a system. Create a prototype-Before investing major dollars into building the system, consider investing in the development of a prototype or mock system

which physically represents the end product. This is similar in concept to an architect's model, which allows one to actually touch and feel the end product

before it is created. Have the users actually develop the system- It is the end-

users who will directly benefit from the system, so why not let them have a hand

in developing it? In the DME is DBA case , the fault that the Open Software

Foundation(OSF) make it's Distributed Management Environment system fail is the

OSF tried to go from theory to perfect product without the real-world trial and

error that is so critical to technology development. Build the solution-With a

model in place, building the solution is relatively easy for the programmer.

Users continue to play an important role at this stage ensuring smooth

communication and accurate user requirement. Implement the system-

Testing the

system, training and learning new procedures can now begin. Because the majority

of time up until now has been spend planning and organizing, implementation

should be smooth and natural, and most importantly quick.

The Role of SAA and ACS in the Assurance of Quality

The Standards Association of Australia was established in 1922 as the

Australian Commonwealth Engineering Standards Association. Their original focus

was on engineering, subsequently it expanded to include manufacturing standards,

product specifications and quality assurance and consumer-related standards. The

role the SAA play is in quality certification. According to SAA, a standard is a published document which sets out the minimum requirements necessary to ensure

that a material, product, or method will do the job it is intended to do. For

systems development, both the Standards Association of Australia and Australian

Computer Society give the guides and standards to develop a system and to

control the quality of a system and to prevent failure from occurring. They also

make the standard of the system developed connectable world wide.

When software development projects fail, they usually fail in a big way.

For large development projects, the cost is typically astronomical, both in terms of dollars spent and human resources consumed, some with even further

reaching implications effecting adversely the whole of a society. Too often, mistakes made in developing one project are perpetuated in subsequent ones. As

with the error which occurred in the London Stock Exchange system, what they

should have done was find out how the system allowed the error to happen and fix

it, then learn from it for making better developments for future information systems.

Bibliography:

1. Fail-safe Advice, Software Magazine, George Black, 3/93 2. All fall down, The

Economist, Anonymous, 20/3/93 3. DME is DBA(Dead Before Arrival), Data

Communications, Robin Layland, 2/94 4. There's No Excuse for Failure,
Canadian

Manager, Selim El Raheb, 9/92 5. Laboratory Systems failure: The enemy
may be us,

Computers in

Healthcare, Stanley J. Geyer, M. D., 9/93 6. Australian Standard Software

quality management system, Standards

Australia