8-2010

Critical Success and Failure Factors in a Charitable Organization's Systems Development and Adoption

Abdulrahman A. Mirza
Information Systems Department, King Saud University, amirza@ksu.edu.sa

Critical Success and Failure Factors in a Charitable Organization's Systems Development and Adoption

Abdulrahman A. Mirza
Information Systems Department, King Saud University
amirza@ksu.edu.sa

ABSTRACT

In the current knowledge society, adoption of information technology (IT) innovation initiatives has become a necessity for the success of most organizations. The decision to adopt information technology solutions however must be made on well-defined user requirements, and not on mere high-expectations. In this paper we present a case study of an ambitious Saudi Arabian charitable organization that decided to start the development of its information system based on anticipated modes of operation and not actual operational and user requirements. Even after the deployment of the information system, the shape of organizational operations was still transforming. Once functional operations of the foundation were finally stabilized it was realized that the developed system, to a large extent, had failed to meet the actual user needs and its intended user adoption did not fully materialize.

Keywords

Requirements definition, non-profit organization, cause-fail analysis, critical success factors, case study, software development, Saudi Arabia.

INTRODUCTION

One of the main objectives of the King Abdullah Foundation for Developmental Housing, established in 2002 as a private charitable organization, is to develop housing communities for the most-needy citizens in the different regions of Saudi Arabia. Another important objective is to provide residents of those communities with developmental programs that would help them become self-sufficient and eventually effective contributing members of the society.

The company that won the contract to develop the foundation's integrated information system (IS) has sales and marketing offices in Saudi Arabia, however, its software development operations are conducted offshore in Egypt. Even though this fact may have created some communication problems and some delays in solving some specific problems, it did not generally have an adverse affect on the low-level adoption of the developed system. We present in this paper the main success and failure factors behind the less than desired adoption and utilization of the developed information system. Only two out of the eight developed modules of this integrated IS were effectively utilized.

We shall start with a background section briefly describing the foundation's activities. We then look at related research regarding the role of IT in non-profit organizations, and the issue of software systems requirements definition and its critical role in producing successful systems. The next two sections present the foundation's organizational structure and the drivers behind its decision to adopt information technology as a strategic solution. We then briefly introduce the different modules of the foundation's developed information system. In the following section, we describe causes leading to the failure in the adoption of certain components of the IS, followed by success factors of utilized subsystems. A brief conclusion section will end the case study with a discussion of lessons learned and some closing remarks.

BACKGROUND

Over the past few years, the King Abdullah Foundation for Developmental Housing has been actively developing modern housing communities in the most-needy remote locations within the country. The first phase of the foundation's projects which is nearing completion covers ten different locations with a total of nearly 2,000 housing units divided among those

projects. In addition to developed housing units, other developments within each community include boys and girls schools, places of worship, health centers, community and training centers, and, administrative offices.

All data belonging to the more than twelve thousand surveyed families around the country, for the purpose of determining the most-needy areas of the nation, were saved in SPSS files. Once projects were decided for specific locations including the most appropriate number of housing units to build in that location, a process was then started through which needy deserving families were selected and offered rent-free developmental housing.

As families move into their new homes, developmental programs are immediately offered. The first program deals with educating families on how to use the different facilities within their new homes. This is a very important program since many of the families may not have had the luxury of air conditioning, stove ovens, refrigerators, or even running tab water. Other developmental programs are eventually offered to families which include social, educational, vocational, and small loan programs.

RELATED RESEARCH

IT in Non-Profit Organizations

Non-profit organizations typically have limited budgets and resources, and commonly rely on volunteer work. Even so, computer and information technologies still play an important role for these types of organizations (Ouellette, 1996). IT departments of charitable organizations just like in their for-profit counterparts, and in spite of their limited resources, are faced with pressures to produce great results and value (Slofstra, 1996). The wide-spread IT acceptance within nonprofits has brought with it a great potential for change. IT has the ability to reconfigure a non-profit's internal and external structures and working relationships, as well as improving its organizational learning and knowledge management systems (Burt & Taylor, 2000; Castells, 1996).

A non-profit organization typically has a strategic objective which mostly involves some social mission: the creation of public value (Moore, 2000; Bryson, 1995; Bryce, 1992). Leaders of nonprofits are encouraged to "obtain a better overall appreciation of IT's full potential, along with a willingness and capacity to more directly link the acquisition and utilization of IT to the organizational mission" (Hackler & Saxton, 2007). According to a research study of 1,572 non-profit organizations in the U.S., 98% of such organizations utilize personal computers in their offices, 97% have Internet connections with 56% of them through high-speed broadband connections. Leaders of nonprofits in the study perceive that the most valuable aspect of IT is in increasing the number of clients served (27%) and in improving communications between staff members (26%). To a lower extent, 11% believe the value is in providing staff with more time to work on other projects, and for reducing operational costs (6%) (Hackler & Saxton, 2007).

As far as technology pitfalls that need to be avoided by non-profit organizations, Gleason (2007) identifies five areas of caution: heavy dependence on used equipment; failure to designate proper budgets for IT infrastructure; failure to clarify operational needs; ignoring decisions incorporated in the organization's systems; and, relying on home grown software. In-house developed software can become almost impossible to maintain, especially if the system developer had departed the organization. It is more advisable to establish a relationship with a software contractor who can provide necessary services when they are needed (Gleason, 2007).

Requirements Definition

Software projects are typically characterized as being late, over budget, unpredictable (Reel, 1999), do not accomplish what the users want, and produce systems that are often ineffectively utilized by those who paid for them (Sommerville & Sawyer, 1977). Based on many studies, the rates of software project failure range between 50% and 80% (Saqib, 2007). The same can be said about offshore software development with a 50% failure rate and a 33% customer satisfaction rate (Rottman, 2006).

One of the most commonly blamed factors for the failure of software projects are inadequately explored or described requirements (El-Ansary, 2002; Verner & Cerpa, 2005). Defining business requirements is considered to be the most important task of software development. However, it is also the poorest performed (Goldsmith, 2004), and the most difficult (Hoffman & Lehner, 2001). Out of the ten identified signs of IS project failure by Reel (1999), seven occur before a system design is developed or a single line of code is written.

Requirements are described as those conditions that must be met in order for the developed software system to be acceptable by the users, customers, and other stakeholders. Requirements may be capabilities, functions, constraints, or other properties that must be satisfied by the system (Sodhi & Sodhi, 2001). A requirements definition document is typically produced after a

contract is won and includes a description of the client's problem statement and the tasks to be accomplished in order to derive the desired software solution (Lecky-Thompson, 2005).

Among the more common problems with requirements definition are those described by Sommerville & Sawyer (1977):

- Requirements do not represent the actual needs of the customer
- Requirements are incomplete or conflicting
- Requirements are difficult and expensive to update after they have been agreed upon.
- Customers, requirements analysts, and software engineers who develop the system have difficulties understanding each other.

The most common method of requirements gathering is through interviews with clients and through the collection of any available customer forms or described processes. The appropriate communication and understanding between developers and clients is a must for proper requirements definition (Holtzblatt & Beyer, 1995). Representatives from the client side must include the potential users of the system, as they are the ones who possess the true and detailed knowledge of how things work. The three main factors of software projects success involve executive management support, a clear statement of requirements, and user involvement in the requirements definition process (Herlea, 1999).

ORGANIZATIONAL STRUCTURE OF THE FOUNDATION

The foundation follows a functional organizational structure. In addition to the CEO's office, there are seven departments that manage the entire operations of the foundation. The first is the management department which includes the financial, investment, administrative communications, and human resources divisions. The second is the engineering department which is concerned with designing the appropriate housing and community models and managing and overseeing the housing construction projects. The IT department handles all information systems and technology requirements of the foundation. The residential community management department overlooks resident concerns and maintenance problems. The last three departments are each of a strictly social responsibility, one for conducting research studies on the most needy areas of the country, the next is concerned with determining the most needy families in a selected location, and the last is concerned with running developmental programs for community residents.

DECISION TO ADOPT INFORMATION TECHNOLOGY

The foundation came to life during an era of great technological advancements within the country, where a growing societal acceptance of the Internet and modern modes of telecommunications was taking place. The government had just around the same period put forth a national strategic plan for the dissemination of information and communications technologies (ICT). Soon after that, the ministry of communications and information technology (http://www.mcit.gov.sa/english) and the communications and information technology commission (http://www.citc.gov.sa) were both established. The spread of the use of mobile devices, personal computers, and the Internet within the country has been far exceeding the world cumulative annual growth rate (CICT, 2005). As a result of such developments, information technology was immediately recognized by the board of trustees as a critical and strategic means for supporting and managing the day-to-day activities of the foundation.

A consulting firm was hired to develop an IT plan for the foundation. This plan included the suggested infrastructure of the local area network, an organizational chart of the IT department, a foundation website, and a comprehensive information system. This information system covered both financial and human resources systems, as well as specialized information systems that were based around the foundation's organizational structure. Each module of the system was basically proposed as representing a single department or division of the foundation. Brief bullet point descriptions of each subsystem were created. These descriptions, however, were mainly derived from the foundation bylaws document.

The suggested information system initially included twelve different modules. These were later expanded to fourteen modules based on recommendations of the foundation's social and engineering consultants. Soon after the selection of a software contractor to develop the information system, but before making any formal agreements, the foundation's initial IT manager decided to seek employment elsewhere. This meant that the IS development project came to a halt. Seven months later, the author of this paper was hired as a part-time consultant, and within a couple of months hired on a full-time basis as the foundation's CIO. Among the main tasks that he set-out to immediately accomplish was the continuation of the IS development project, as it was perceived to be a priority for the foundation.

MODULES OF THE DEVELOPED INFORMATION SYSTEM

The initially determined fourteen modules were narrowed down to only eight. These included the modules for housing projects management, beneficiaries management, housing and residents management, developmental programs management, investments management, managerial communications (correspondence system), archiving, and the system management module (Table 1)

A system analyst from the software contractor flew into the country for two weeks and met with departmental heads in order to obtain the detailed requirements of the different subsystems. Meetings typically included the departmental head, system analyst, and the foundation's CIO. Department heads talked about their needs and the types of functionality that they expected out of the system. A list of needed reports, to be generated by the system, were also provided by each department. Additionally, some departments provided specific forms that represented important information to be captured through the system. In many cases, however, those forms were generic and in some cases adopted from general charitable organizations "standard" forms.

No.	Module Name
1	Housing projects management
2	Beneficiaries management
3	Housing and residents management
4	Managerial communications system
5	Developmental programs management
6	System management
7	Investments management
8	Archiving system

Table 1. Final list of IS modules selected for development

Even though each system mostly serves a single department, the foundation's CIO stressed to the system analyst the importance of systems integration since in the end all departments of the foundation aim to provide a valuable service to their end customers, the needy families.

LOW-LEVEL UTILIZATION OF DEVELOPED INFORMATION SYSTEM

Even though the software contractor delivered the software system with minimal delays, with an acceptable number of missing or non-operational functionality, and with an acceptable user-friendly interface, the information system in its current status is considered on the most part to be a failure. The only subsystems that have been utilized to a great extent are the system management and beneficiaries management modules. The system management module is used for carrying-out system administrative services such as establishing new accounts, granting privileges, and creating certain data types as needed for different subsystems. Other subsystems were hardly utilized at all.

FACTORS BEHIND THE GENERAL FAILURE OF THE FOUNDATION'S INFORMATION SYSTEM

Among the ten signs of IS failure as described by Reel (1999) are the ill definition of project scope, resistance of system users, loss of sponsorship, and the change in business needs. All four signs described here in addition to other failure factors were clearly visible in the case of the foundation's IS implementation, and hence leading to the failure of the system for the most part. We take a look at these factors in this section.

Change in Management and System Expectations

In the case of engineering department requirements, it seems that the department head did not consider all the possible variations that may occur in the various construction projects. As an example, he indicated that there were 16 different activities that make up the total components of a construction project. When asked by the CIO about the possibility that there

may be more, less, or different activities, he said no! This might have been true for housing units being developed at a particular period using a particular method. Hence, the system was developed with a limited ability to track a static and predefined number and set of construction activities.

A new department head later expressed that the housing construction activities were not set and that they could vary based on the actual type of building being constructed and type of contract signed with the construction contractor. Additionally, other important aspects of tracking payments to contractors were missing. Payments made to external consultants who also play an important role in the supervision of housing construction projects were not considered in the system at all. The new department manager also demanded a much improved and detailed document archiving system. Without these important components of the system, he claimed that the system is useless to his department. Hence, the problem of requirements not representing the actual needs of the customer as well as being incomplete (Somerville & Sawyer, 1977).

Failure to involve all Stakeholders in the Requirements Definition Process

A good number of the departments of the foundation were, and in some cases still are, a one-man show. The only departments that had supporting staff were the engineering, beneficiaries, and general management departments. The engineering department had two engineers working at the time when detailed system requirements were being gathered by the software contractor. One of the main problems with the engineering management subsystem is that none of the department engineers were asked to participate in the requirements definition process. This resulted in missing important system functionality that the first department manager had failed to include as requirements. According to the engineers who were asked to utilize the system after it was deployed, it was not flexible enough to handle their day-to-day operational requirements, and hence refrained from using it. The department manager who defined the initial system requirements had defined them for activities that needed to be carried out by the department engineers, and hence, as it turned out later, the specified requirement did not completely represent the needs of the end-users (Somerville & Sawyer, 1977). Both the CIO and system analyst relied strictly on the manager's defined requirements and unfortunately did not communicate with the main would-be users of the system. One of the main success factors for software project success described by Herlea (1999) is user involvement in the requirements definition process. This was surely missing in this case.

Ill-defined Scope and Changes in Business Needs

The housing and residents management subsystem is a very innovative module. Its requirements were mostly envisioned by one of the foundation's architectural full-time consultants. It is used for creating housing models, including the definitions of each model in the number and types of rooms as well as the different components within each room (e.g., A/C, sink, oven, door knob, etc.). Beneficiaries that have been selected are assigned a unit number that is also defined through this subsystem. Additionally, the subsystem provides the ability to track maintenance requests by tenants. This includes specific detailed information about the maintenance request including the actual component of the housing unit needing maintenance, action taken, cost of repair, and the name of the maintenance engineer, etc.

Naturally, this system should enable the foundation to track all maintenance requests, costs, problem components, and even problem tenants. Unfortunately with the conclusion of the first housing community project, a managerial decision was made such that the foundation will not be responsible for maintaining the inside of housing units. Tenants themselves would need to bare the cost, in an effort to instill upon them the value of responsibility, and to make them aware that maintenance repairs can be costly, and hence be more apt to take good care of the assigned housing unit. Tenants would need to find their own maintenance engineers as the housing communities did not hire any. This meant however, that the main functionalities of the subsystem will hardly be utilized. The main benefits of the subsystem will be limited to mainly realizing which unit the beneficiary lives in and what are the different model designs. This case definitely reflects the problem of ill-defined scope and change in business needs (Reel, 1999).

Passive Requirements Analyst Role

The software company's system analyst unfortunately did not do much to help department heads in utilizing the full potential that information systems can offer. The company did on the most part deliver what was agreed upon in the requirements analysis document. However, the analyst could have done a lot more to help the foundation's personnel in recognizing additional functionality that could have resulted in a much more useful system. A very probable and good reason for this is that the company had found itself in an already set price contract and hence wanted to get away with getting as little as possible done. The software contractor's system analyst was more concerned about understanding certain issues rather than trying to help the customer expanding on requirements or discovering new ones, and hence the problem of inadequately

explored requirements (El-Ansary, 2002; Verner & Cerpa, 2005). The limited contract value for the development of the total project did not leave much space for the developer to work on more innovative solutions.

Passive CEO and Department Manager Roles

Both CEO and departmental heads for the most part did not take the implemented system seriously. Once the system was deployed, a memo was written from the CIO to all department heads requesting that they start using the system. It was explicitly explained that they had to perform acceptance testing to verify that developed systems meet their previously defined requirements. Training had been conducted by the developing company to all would be users of the system. Departments were asked to give their feedback within three weeks after the end of training. The three week period expired and only the beneficiaries department and managerial communications division had performed their requested tasks. All other departments did not use the system at all.

The CIO repeatedly informed the CEO that most departments were reluctant to use the system, however, rather than directing each department's manager to take the matter seriously, the CEO continued placing the burden on the CIO. Naturally the CIO did not have the executive capacity to enforce other managers to answer his requests. This situation is clearly identified with lack of the critical success factor of executive management support (Herlea, 1999), and the problem of management loss of sponsorship (Reel, 1999). Even though upper management initially pushed for the development of organizational information systems and realized its importance, it did not realize that continuous follow-up and support for the system was needed all the way through.

User Apathy towards the System

The would-be system users had become comfortable performing their tasks in a certain way, and were in the most part not too excited about doing things any other way. Others were not involved in the requirements definition process, and hence were pointing-out that the system is missing important functionality and that it does not meet their complete needs. With the lack of strong efforts by the CEO and most department heads in pushing the use of the newly developed system, it was natural that the foundation personnel would not make a great effort on their own will. This is a good example of the problem of user-resistance to new systems (Reel, 1999).

User Acceptance Testing Carried-out by CIO

As most departments and divisions were unwilling or incapable of using their developed software modules, the CIO had to take on the responsibility of testing the system himself. The software contractor had provided a three months period during which the foundation can report any missing functionality or system bugs. Armed with the systems requirements document, the CIO made sure that customer requirements were implemented in the system and that the system operated correctly. A good number of bugs were discovered in the system as a result of performed acceptance testing. This process however, being carried-out by a non-expert in the subject matter of tested subsystems, did not capture missing functionality not expressed through system requirements. This is a good example again of the problem of resistance by system users (Reel, 1999) who were not willing to use the system and hence not testing it for completeness. The CEO and department managers failed to play a more visible role in making sure that end-users actually performed their required tasks.

MOST SUCCESSFUL SYSTEM MODULE AND FACTORS OF SUCCESS

In addition to the system management module, mainly for the use by the system administrator, the beneficiaries management subsystem is considered to be the main successful implementation. The beneficiaries department relied heavily on this subsystem in determining its qualifying families. System data is continuously modified with up to date information and data of new prospective beneficiaries. Reports are used for official purposes and are used for presenting information to the CEO and other official committees. Critical success factors for this system include the following:

- Even though no actual family selection process had been carried-out, the main activities to be conducted by the beneficiaries department were for the greater part well defined before the start of the requirements analysis process.
- Determined rules and regulations for the selection of beneficiaries were arrived at as a result of a collective effort. These regulations which were translated into system requirements were arrived at by the managers of the three social departments: the beneficiaries, the research and studies, and developmental programs departments, and had the feedback and approval of the foundation's CEO.

- Beneficiary data was readily available in SPSS files and migrated to the IS database through a specially developed data migration tool. This meant that necessary data for the quick utilization of the system did not require great collection and population efforts in order to start taking immediate advantage of the system.
- The head of the beneficiaries department was greatly anticipating the completion of the proposed information system, and was a great advocate for it.
- Even though the department's social worker, who would be using the application, did not participate in the requirements definition process, the strong direction by the department manager to use the module lead to his quick acceptance of the developed subsystem.
- Deriving the names of qualified beneficiaries and creating reports regarding selected and rejected families would require a much greater effort if any other method for arriving at those results was to be used. Hence there was an urgent need for using automated processes for carrying-out the required tasks of the department. For other departments, the urgent need for system utilization did not exist.
- Active participation by the system's end-user in the acceptance testing process and immediate recognition of missing functionality and needed modifications.
- Active support by the CIO in immediately relaying discovered buggy and missing functionality, and requested modifications to the software developer.

CONCLUSION

Among the very important lessons derived from this case study is that software projects should not be jumped into without careful consideration. It is very easy for organizations these days to recognize the value and importance of information technology in general and information systems in particular. However, organizations must first determine their exact needs and get all users involved in the requirements definition process before setting out in pursuit of a silver bullet solution that would solve all their operational requirements.

It is very important in the requirements definition process not to over-estimate actual system needs. This does not mean that a system should not anticipate modifications in organizational processes, but, decisions to anticipate such modifications and additions must be based on solid evidence that such activities could occur with high probability.

The King Abdullah Foundation for Developmental Housing is currently awaiting the deployment of a new release of its information system. It is anticipated this time around that the implemented system will be much more useful to all users. This is based on the great efforts made by all stakeholders in the development of much improved requirement definitions. This was only possible after the stabilization of operational activities within each department and the foundation in general. Additionally, concerning specific areas where specialized knowledge was needed and not possessed by the foundation, such as the administrative correspondence system, experts were hired to supervise the requirements definition process. Last but not least, as a critical driver for the expected IS adoption success, the second time around, is the much more proactive role played by department managers, CIO, and CEO.

REFERENCES

1. CITC (2005) Communications and Information Technology Commission, Saudi Arabia – Towards the Information Society, November, *Available at: http://www.yesser.gov.sa/documents/ksa-to-information-society.pdf*, [Retrieved online on April 15, 2007].
2. El-Ansary, A. (2002) Behavioral Pattern Analysis: towards a new representation of systems requirements based on actions and events, *In Proceedings of the 2002 ACM Symposium on Applied Computing*, Madrid, Spain, March 11 - 14, SAC '02. ACM, New York, NY, pp. 984-991.
3. Gleason, D. (2007) Five technology pitfalls for nonprofits: finding cost-effective solutions, *Nonprofit World*, Vol. 25, No. 3, pp. 22-24.
4. Goldsmith, R. (2004) Discovering real system requirements for software project success, Artech House, Inc., Norwood, MA.
5. Hackler, D. and Saxton, G.D. (2007) The Strategic Use of Information Technology by Nonprofit Organizations: Increasing Capacity and Untapped Potential*, Public Administration Review*, Vol. 67 No. 3, pp. 474–487.
6. Herlea, D. E. (1999) User participation in requirements negotiation. *SIGGROUP Bull.* 20, 1 (Apr. 1999), pp. 30-35.

7. Holtzblatt, K. and Beyer, H. R. (1995) Requirements gathering: the human factor, *Comm. ACM*, vol. 38, no. 5, May, pp. 31-32.
8. Hoffman, H.F., Lehner, F. (2001) Requirements engineering as a success factor in software projects, *IEEE Software*, Vol. 18, No. 4, pp. 58-66.
9. Lecky-Thompson, G. (2005) Corporate Software Project Management, Charles River Media, Inc., Hingham, MA.
10. Ouellette, T. (1996) Nonprofits rely on IS innovation, *Computerworld*, Dec. 16, 1996, p. 6.
11. Reel, J.S. (1999) Critical success factors in software projects, *IEEE Software*, Vol. 16, No. 3, May/June, pp. 18-23.
12. Rottman, J. (2006) Successfully outsourcing embedded software development, *IEEE Computer*, Vol. 39, No. 1, pp. 55-61.
13. Saqib, M. (2007) Causes of software project failure, Ezine Articles, Feb. 13, *Available at: http://ezinearticles.com/?Causes-Of-Software-Project-Failure&id=453814* [Retrieved online on January 2009].
14. Sodhi, J. and Sodhi, P. (2001) IT Project Management Handbook, Management Concepts, Vienna, VA., USA.
15. Solfstra, M. (1996) At United Way, charity begins in IS, *Computing Canada*, Vol. 22, No. 8, pp 1-2.
16. Sommerville, I. and Sawyer, P. (1997) Requirements Engineering: A Good Practice Guide, John Wiley & Sons, New York, NY., USA.
17. Verner, J. M., Cerpa, N. (2005) Australian Software Development: What Software Project Management Practices Lead to Success? *2005 Australian Software Engineering Conference (ASWEC'05)*, pp. 70-77.